Broadcast Psi-calculi

Palle Raabjerg Johannes Åman Pohjola Tjark Weber

March 17, 2025

Abstract

We provide an Isabelle/HOL-Nominal formalisation of the definitions, theorems and proofs in the paper *Broadcast Psi-calculi with an Application to Wireless Protocols* by Borgström et al., which extends the Psi-calculi framework with primitives for broadcast communication in order to model wireless protocols.

1 Introduction

We provide an Isabelle/HOL-Nominal formalisation of the definitions, theorems and proofs in the paper *Broadcast Psi-calculi with an Application to Wireless Protocols* [4, 5], which extends the Psi-calculi framework [2, 3, 1] with primitives for broadcast communication in order to model wireless protocols.

The file `Broadcast_Thms.thy` contains a collection of the relevant definitions and theorems, with comments relating them directly to the paper.

2 Formalisation

theory *Broadcast-Chain*
  imports *Psi-Calculi.Chain*
begin

lemma *pair-perm-fresh-contr*:
  fixes *a*::*'a* **and** *b*::*'a*
  assumes
    at: *at TYPE*(*'a*)
  and
    prems: *b* ♯ *pi* (*a*, *b*) ∈ *set pi*
  shows *False*
  ⟨*proof*⟩

lemma *pair-perm-fresh-contr'*:
  fixes *a*::*'a* **and** *b*::*'a*
  assumes

at: *at TYPE*('*a*)
and
prems: *a* ♯ *pi* (*a*, *b*) ∈ *set pi*
shows *False*
⟨*proof*⟩

lemma *list-set-supp*:
fixes *l* :: ('*d*::*fs-name*) *list*
shows *supp* (*set l*) = (*supp l* :: *name set*)
⟨*proof*⟩

lemma *name-set-supp*:
assumes *finite a*
shows *supp a* = (*a*::*name set*)
⟨*proof*⟩

lemma *supp-idem*:
fixes *l* :: ('*d*::*fs-name*)
shows *supp*((*supp l*)::*name set*) = (*supp*(*l*)::*name set*)
⟨*proof*⟩

lemma *fresh-supp*:
fixes *a* :: *name*
and *X* :: ('*d*::*fs-name*)
shows *a* ♯ ((*supp X*)::*name set*) = *a* ♯ *X*
⟨*proof*⟩

lemma *fresh-chain-supp*:
fixes *A* :: *name list*
and *X* :: ('*d*::*fs-name*)
shows *A* ♯∗ ((*supp X*)::*name set*) = *A* ♯∗ *X*
⟨*proof*⟩

lemma *fresh-chain-fin-union*:
fixes *X*::('*d*::*fs-name set*)
and *Y*::('*d*::*fs-name set*)
and *A*::*name list*
assumes *f1*: *finite X*
and *f2*: *finite Y*
shows *A*♯∗(*X*∪*Y*) = (*A*♯∗*X* ∧ *A*♯∗ *Y*)
⟨*proof*⟩

lemma *fresh-subset*:
fixes *S* :: *name set*
and *S'* :: *name set*
and *a* :: *name*
assumes *a* ♯ *S*
and *S'* ⊆ *S*
and *finite S*

shows $a \sharp S'$
⟨*proof*⟩

lemma *fresh-subset'*:
fixes $S :: 'd::fs\text{-}name\ set$
and $S' :: 'd::fs\text{-}name\ set$
and $a :: name$
assumes $a \sharp S$
and $S' \subseteq S$
and *finite* S

shows $a \sharp S'$
⟨*proof*⟩

lemma *fresh-star-subset'*:
fixes $S :: 'd::fs\text{-}name\ set$
and $S' :: 'd::fs\text{-}name\ set$
and $A :: name\ list$
assumes $A \sharp* S$
and $S' \subseteq S$
and *finite* S

shows $A \sharp* S'$
⟨*proof*⟩

lemma *fresh-star-subset*:
fixes $S :: name\ set$
and $S' :: name\ set$
and $A :: name\ list$
assumes $A \sharp* S$
and $S' \subseteq S$
and *finite* S

shows $A \sharp* S'$
⟨*proof*⟩

lemma *times-set-fresh*:
fixes $a :: name$
and $S :: name\ list$
and $S' :: name\ list$
assumes $a \sharp set\ S$
and $a \sharp set\ S'$
shows $a \sharp set\ S \times set\ S'$
⟨*proof*⟩

lemma *times-set-fresh-star*:
fixes $A :: name\ list$
and $S :: name\ list$

and S' :: *name list*
assumes $A \sharp* set\ S$
and $A \sharp* set\ S'$
shows $A \sharp* (set\ S \times set\ S')$
⟨*proof*⟩

lemma *supp-list-set*:
fixes M::$'d$::*fs-name list*
shows $(supp\ M) = ((supp(set\ M))$::*name set*$)$
⟨*proof*⟩

lemma *fresh-list-set*:
fixes M::$'d$::*fs-name list*
and A::*name list*
shows $A \sharp* set\ M = A \sharp* M$
⟨*proof*⟩

lemma *permSupp*:
fixes Ψ :: *name prm*
and Ψ' :: $'d$::*fs-name*

shows $(supp(\Psi \cdot \Psi')$::*name set*$) \subseteq ((supp\ \Psi) \cup (supp\ \Psi'))$
⟨*proof*⟩

end
theory *Broadcast-Frame*
imports *Psi-Calculi.Frame*
begin

locale *assertionAux* = *Frame.assertionAux SCompose SImp SBottom SChanEq*
for *SCompose* :: $'b$::*fs-name* ⇒ $'b$ ⇒ $'b$ (**infixr** ‹⊗› *80*)
and *SImp* :: $'b$ ⇒ $'c$::*fs-name* ⇒ *bool* (‹- ⊢ -› [*70*, *70*] *70*)
and *SBottom* :: $'b$ (‹⊥› *90*)
and *SChanEq* :: ($'a$::*fs-name* ⇒ $'a$ ⇒ $'c$) (‹- ↔ -› [*80*, *80*] *80*)
+
fixes *SOutCon* :: $'a$::*fs-name* ⇒ $'a$ ⇒ $'c$ (‹- ≼ -› [*80*, *80*] *80*)
and *SInCon* :: $'a$::*fs-name* ⇒ $'a$ ⇒ $'c$ (‹- ≽ -› [*80*, *80*] *80*)

assumes *statEqvt'''*[*eqvt*]: $\bigwedge p$::*name prm*. $p \cdot (M \preceq N) = (p \cdot M) \preceq (p \cdot N)$
and *statEqvt''''*[*eqvt*]: $\bigwedge p$::*name prm*. $p \cdot (M \succeq N) = (p \cdot M) \succeq (p \cdot N)$

begin

lemma *chanInConSupp*:
fixes M :: $'a$
and N :: $'a$

shows $(supp(M \succeq N)$::*name set*$) \subseteq ((supp\ M) \cup (supp\ N))$
⟨*proof*⟩

lemma *chanOutConSupp*:
fixes $M :: 'a$
and $N :: 'a$

shows $(supp(M \preceq N)::name\ set) \subseteq ((supp\ M) \cup (supp\ N))$
⟨*proof*⟩

lemma *freshInCon*[*intro*]:
fixes $x :: name$
and $M :: 'a$
and $N :: 'a$

assumes $x \sharp M$
and $x \sharp N$

shows $x \sharp M \succeq N$
⟨*proof*⟩

lemma *freshInConChain*[*intro*]:
fixes $xvec :: name\ list$
and $Xs :: name\ set$
and $M :: 'a$
and $N :: 'a$

shows $[\![xvec \sharp* M; xvec \sharp* N]\!] \Longrightarrow xvec \sharp* (M \succeq N)$
and $[\![Xs \sharp* M; Xs \sharp* N]\!] \Longrightarrow Xs \sharp* (M \succeq N)$
⟨*proof*⟩

lemma *freshOutCon*[*intro*]:
fixes $x :: name$
and $M :: 'a$
and $N :: 'a$

assumes $x \sharp M$
and $x \sharp N$

shows $x \sharp M \preceq N$
⟨*proof*⟩

lemma *freshOutConChain*[*intro*]:
fixes $xvec :: name\ list$
and $Xs :: name\ set$
and $M :: 'a$
and $N :: 'a$

shows $[\![xvec \sharp* M; xvec \sharp* N]\!] \Longrightarrow xvec \sharp* (M \preceq N)$
and $[\![Xs \sharp* M; Xs \sharp* N]\!] \Longrightarrow Xs \sharp* (M \preceq N)$
⟨*proof*⟩

lemma *chanOutConClosed*:
fixes $\Psi :: 'b$
and $M :: 'a$
and $N :: 'a$
and $p :: name\ prm$

assumes $\Psi \vdash M \preceq N$

shows $(p \cdot \Psi) \vdash (p \cdot M) \preceq (p \cdot N)$
⟨*proof*⟩

lemma *chanInConClosed*:
fixes $\Psi :: 'b$
and $M :: 'a$
and $N :: 'a$
and $p :: name\ prm$

assumes $\Psi \vdash M \succeq N$

shows $(p \cdot \Psi) \vdash (p \cdot M) \succeq (p \cdot N)$
⟨*proof*⟩

end

locale *assertion* = *assertionAux SCompose SImp SBottom SChanEq SOutCon SInCon* + *assertion SCompose SImp SBottom SChanEq*
for *SCompose* :: *'b*::*fs-name* ⇒ *'b* ⇒ *'b*
and *SImp* :: *'b* ⇒ *'c*::*fs-name* ⇒ *bool*
and *SBottom* :: *'b*
and *SChanEq* :: *'a*::*fs-name* ⇒ *'a* ⇒ *'c*
and *SOutCon* :: *'a*::*fs-name* ⇒ *'a* ⇒ *'c*
and *SInCon* :: *'a*::*fs-name* ⇒ *'a* ⇒ *'c* +

assumes *chanOutConSupp*: *SImp* Ψ (*SOutCon M N*) ⟹ (((*supp N*)::*name set*) ⊆ ((*supp M*)::*name set*))
and *chanInConSupp*: *SImp* Ψ (*SInCon N M*) ⟹ (((*supp N*)::*name set*) ⊆ ((*supp M*)::*name set*))

begin

notation *SOutCon* (‹- ≼ -› [*90*, *90*] *90*)
notation *SInCon* (‹- ≽ -› [*90*, *90*] *90*)

end

end
theory *Semantics*
imports *Broadcast-Chain Broadcast-Frame*

begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Semantics* from [1]. The nominal datatypes *('a,'b,'c) residual* and *'a action* have been extended with constructors for broadcast input and output. This leads to a different semantics.

nominal-datatype $('a, 'b, 'c)$ *boundOutput* =
  BOut $'a$::*fs-name* $('a, 'b$::*fs-name*, $'c$::*fs-name*) *psi* (‹- $\prec''$ -› [*110*, *110*] *110*)
  | *BStep* «*name*» $('a, 'b, 'c)$ *boundOutput* (‹(|ν-|)-› [*110*, *110*] *110*)

primrec *BOresChain* :: *name list* ⇒ ($'a$::*fs-name*, $'b$::*fs-name*, $'c$::*fs-name*) *boundOutput* ⇒
  $('a, 'b, 'c)$ *boundOutput*
  where
    Base: *BOresChain* [] $B = B$
  | *Step*: *BOresChain* $(x\#xs)$ B = (|νx|)(*BOresChain* xs B)

abbreviation
  BOresChainJudge (‹(|ν*-|)-› [*80*, *80*] *80*) **where** (|ν*$xvec$|)$B \equiv$ *BOresChain* $xvec$ B

lemma *BOresChainEqvt*[*eqvt*]:
  fixes *perm* :: *name prm*
    and *lst* :: *name list*
    and B :: ($'a$::*fs-name*, $'b$::*fs-name*, $'c$::*fs-name*) *boundOutput*

shows $perm \cdot$ ((|ν*$xvec$|)B) = (|ν*($perm \cdot xvec$)|)($perm \cdot B$)
  ⟨*proof*⟩

lemma *BOresChainSimps*[*simp*]:
  fixes *xvec* :: *name list*
    and N :: $'a$::*fs-name*
    and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
    and N' :: $'a$
    and P' :: $('a, 'b, 'c)$ *psi*
    and B :: $('a, 'b, 'c)$ *boundOutput*
    and B' :: $('a, 'b, 'c)$ *boundOutput*

shows ((|ν*$xvec$|)$N \prec' P = N' \prec' P') = (xvec = [] \wedge N = N' \wedge P = P')$
  and $(N' \prec' P' =$ (|ν*$xvec$|)$N \prec' P) = (xvec = [] \wedge N = N' \wedge P = P')$
  and $(N' \prec' P' = N \prec' P) = (N = N' \wedge P = P')$
  and ((|ν*$xvec$|)$B =$ (|ν*$xvec$|)$B') = (B = B')$
  ⟨*proof*⟩

lemma *outputFresh*[*simp*]:
  fixes *Xs* :: *name set*
    and *xvec* :: *name list*
    and N :: $'a$::*fs-name*
    and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*

shows $(Xs \sharp* (N \prec' P)) = ((Xs \sharp* N) \wedge (Xs \sharp* P))$
and $(xvec \sharp* (N \prec' P)) = ((xvec \sharp* N) \wedge (xvec \sharp* P))$
⟨*proof*⟩

lemma *boundOutputFresh*:
fixes x :: *name*
and $xvec$ :: *name list*
and B :: $('a::fs\text{-}name, 'b::fs\text{-}name, 'c::fs\text{-}name)$ *boundOutput*

shows $(x \sharp (\!|\nu*xvec|\!) B)) = (x \in set\ xvec \vee x \sharp B)$
⟨*proof*⟩

lemma *boundOutputFreshSet*:
fixes Xs :: *name set*
and $xvec$ :: *name list*
and B :: $('a::fs\text{-}name, 'b::fs\text{-}name, 'c::fs\text{-}name)$ *boundOutput*
and $yvec$ :: *name list*
and x :: *name*

shows $Xs \sharp* ((\!|\nu*xvec|\!) B) = (\forall x \in Xs.\ x \in set\ xvec \vee x \sharp B)$
and $yvec \sharp* ((\!|\nu*xvec|\!) B) = (\forall x \in (set\ yvec).\ x \in set\ xvec \vee x \sharp B)$
and $Xs \sharp* ((\!|\nu x|\!) B) = Xs \sharp* [x].B$
and $xvec \sharp* ((\!|\nu x|\!) B) = xvec \sharp* [x].B$
⟨*proof*⟩

lemma *BOresChainSupp*:
fixes $xvec$ :: *name list*
and B :: $('a::fs\text{-}name, 'b::fs\text{-}name, 'c::fs\text{-}name)$ *boundOutput*

shows $(supp((\!|\nu*xvec|\!) B)::name\ set) = (supp\ B) - (supp\ xvec)$
⟨*proof*⟩

lemma *boundOutputFreshSimps*[*simp*]:
fixes Xs :: *name set*
and $xvec$ :: *name list*
and B :: $('a::fs\text{-}name, 'b::fs\text{-}name, 'c::fs\text{-}name)$ *boundOutput*
and $yvec$ :: *name list*
and x :: *name*

shows $Xs \sharp* xvec \Longrightarrow (Xs \sharp* ((\!|\nu*xvec|\!) B)) = (Xs \sharp* B)$
and $yvec \sharp* xvec \Longrightarrow yvec \sharp* ((\!|\nu*xvec|\!) B) = yvec \sharp* B$
and $xvec \sharp* ((\!|\nu*xvec|\!) B)$
and $x \sharp xvec \Longrightarrow x \sharp (\!|\nu*xvec|\!) B = x \sharp B$
⟨*proof*⟩

lemma *boundOutputChainAlpha*:
fixes p :: *name prm*
and $xvec$ :: *name list*

and B :: ($'a$::*fs-name*, $'b$::*fs-name*, $'c$::*fs-name*) *boundOutput*
and *yvec* :: *name list*

assumes *xvecFreshB*: $(p \cdot xvec) \sharp* B$
and S: *set* $p \subseteq$ *set xvec* $\times$ *set* $(p \cdot xvec)$
and $(set\ xvec) \subseteq (set\ yvec)$

shows $(\langle\!|\nu * yvec|\!\rangle B) = (\langle\!|\nu * (p \cdot yvec)|\!\rangle (p \cdot B))$
⟨*proof*⟩

lemma *boundOutputChainAlpha'*:
fixes p :: *name prm*
and *xvec* :: *name list*
and B :: ($'a$::*fs-name*, $'b$::*fs-name*, $'c$::*fs-name*) *boundOutput*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes *xvecFreshB*: *xvec* $\sharp*$ B
and S: *set* $p \subseteq$ *set xvec* $\times$ *set yvec*
and *yvec* $\sharp*$ $(\langle\!|\nu * zvec|\!\rangle B)$

shows $(\langle\!|\nu * zvec|\!\rangle B) = (\langle\!|\nu * (p \cdot zvec)|\!\rangle (p \cdot B))$
⟨*proof*⟩

lemma *boundOutputChainAlpha''*:
fixes p :: *name prm*
and *xvec* :: *name list*
and M :: $'a$::*fs-name*
and P :: ($'a$::*fs-name*, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
and *yvec* :: *name list*

assumes $(p \cdot xvec) \sharp* M$
and $(p \cdot xvec) \sharp* P$
and *set* $p \subseteq$ *set xvec* $\times$ *set* $(p \cdot xvec)$
and $(set\ xvec) \subseteq (set\ yvec)$

shows $(\langle\!|\nu * yvec|\!\rangle M \prec' P) = (\langle\!|\nu * (p \cdot yvec)|\!\rangle (p \cdot M) \prec' (p \cdot P))$
⟨*proof*⟩

lemma *boundOutputChainSwap*:
fixes x :: *name*
and y :: *name*
and N :: $'a$::*fs-name*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
and *xvec* :: *name list*

assumes $y \sharp N$
and $y \sharp P$
and $x \in (set\ xvec)$

shows $(\!|\nu * xvec|\!)N \prec' P = (\!|\nu * ([(x,\ y)] \cdot xvec)|\!)([(x\ ,y)] \cdot N) \prec' ([(x,\ y)] \cdot P)$
⟨*proof*⟩

lemma *alphaBoundOutput*:
fixes x :: *name*
and y :: *name*
and B :: $('a::fs\text{-}name,\ 'b::fs\text{-}name,\ 'c::fs\text{-}name)$ *boundOutput*

assumes $y \sharp B$

shows $(\!|\nu x|\!)B = (\!|\nu y|\!)([(x,\ y)] \cdot B)$
⟨*proof*⟩

lemma *boundOutputEqFresh*:
fixes B :: $('a::fs\text{-}name,\ 'b::fs\text{-}name,\ 'c::fs\text{-}name)$ *boundOutput*
and C :: $('a,\ 'b,\ 'c)$ *boundOutput*
and x :: *name*
and y :: *name*

assumes $(\!|\nu x|\!)B = (\!|\nu y|\!)C$
and $x \sharp B$

shows $y \sharp C$
⟨*proof*⟩

lemma *boundOutputEqSupp*:
fixes B :: $('a::fs\text{-}name,\ 'b::fs\text{-}name,\ 'c::fs\text{-}name)$ *boundOutput*
and C :: $('a,\ 'b,\ 'c)$ *boundOutput*
and x :: *name*
and y :: *name*

assumes $(\!|\nu x|\!)B = (\!|\nu y|\!)C$
and $x \in supp\ B$

shows $y \in supp\ C$
⟨*proof*⟩

lemma *boundOutputChainEq*:
fixes *xvec* :: *name list*
and B :: $('a::fs\text{-}name,\ 'b::fs\text{-}name,\ 'c::fs\text{-}name)$ *boundOutput*
and *yvec* :: *name list*
and B' :: $('a,\ 'b,\ 'c)$ *boundOutput*

assumes $(\!|\nu * xvec|\!)B = (\!|\nu * yvec|\!)B'$
and $xvec \sharp * yvec$
and $length\ xvec = length\ yvec$

shows $\exists p.\ (set\ p) \subseteq (set\ xvec) \times set\ (yvec) \wedge distinctPerm\ p \wedge\ B = p \cdot B' \wedge$

$(set\ (map\ fst\ p)) \subseteq (supp\ B) \wedge xvec\ \sharp*\ B' \wedge yvec\ \sharp*\ B$
⟨*proof*⟩

lemma *boundOutputChainEqLength*:
fixes *xvec* :: *name list*
and *M* :: *'a::fs-name*
and *P* :: *('a, 'b::fs-name, 'c::fs-name) psi*
and *yvec* :: *name list*
and *N* :: *'a::fs-name*
and *Q* :: *('a, 'b::fs-name, 'c::fs-name) psi*

assumes $(\!|\nu*xvec|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' Q$

shows *length xvec* = *length yvec*
⟨*proof*⟩

lemma *boundOutputChainEq'*:
fixes *xvec* :: *name list*
and *M* :: *'a::fs-name*
and *P* :: *('a, 'b::fs-name, 'c::fs-name) psi*
and *yvec* :: *name list*
and *N* :: *'a*
and *Q* :: *('a::fs-name, 'b::fs-name, 'c::fs-name) psi*

assumes $(\!|\nu*xvec|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' Q$
and *xvec* $\sharp*$ *yvec*

shows $\exists p.\ (set\ p) \subseteq (set\ xvec) \times set\ (yvec) \wedge distinctPerm\ p \wedge\ M = p \cdot N \wedge\ P = p \cdot Q \wedge xvec\ \sharp*\ N \wedge xvec\ \sharp*\ Q \wedge yvec\ \sharp*\ M \wedge yvec\ \sharp*\ P$
⟨*proof*⟩

lemma *boundOutputChainEq''*:
fixes *xvec* :: *name list*
and *M* :: *'a::fs-name*
and *P* :: *('a, 'b::fs-name, 'c::fs-name) psi*
and *yvec* :: *name list*
and *N* :: *'a*
and *Q* :: *('a::fs-name, 'b::fs-name, 'c::fs-name) psi*

assumes $(\!|\nu*xvec|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' Q$
and *xvec* $\sharp*$ *yvec*
and *distinct xvec*
and *distinct yvec*

obtains *p* **where** $(set\ p) \subseteq (set\ xvec) \times set\ (p \cdot xvec)$ **and** *distinctPerm p* **and** *yvec* = $p \cdot xvec$ **and** $N = p \cdot M$ **and** $Q = p \cdot P$ **and** *xvec* $\sharp*$ *N* **and** *xvec* $\sharp*$ *Q* **and** $(p \cdot xvec)\ \sharp*\ M$ **and** $(p \cdot xvec)\ \sharp*\ P$
⟨*proof*⟩

lemma *boundOutputEqSupp'*:
fixes x :: *name*
and *xvec* :: *name list*
and M :: $'a$::*fs-name*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
and y :: *name*
and *yvec* :: *name list*
and N :: $'a$
and Q :: ($'a$, $'b$, $'c$) *psi*

assumes *Eq*: $(\!|\nu x|\!)((\!|\nu * xvec|\!)M \prec' P) = (\!|\nu y|\!)((\!|\nu * yvec|\!)N \prec' Q)$
and $x \neq y$
and $x \sharp yvec$
and $x \sharp xvec$
and $y \sharp xvec$
and $y \sharp yvec$
and $xvec \sharp* yvec$
and $x \in supp\ M$

shows $y \in supp\ N$
⟨*proof*⟩

lemma *boundOutputChainOpenIH*:
fixes *xvec* :: *name list*
and x :: *name*
and B :: ($'a$::*fs-name*, $'b$::*fs-name*, $'c$::*fs-name*) *boundOutput*
and *yvec* :: *name list*
and y :: *name*
and B' :: ($'a$, $'b$, $'c$) *boundOutput*

assumes *Eq*: $(\!|\nu * xvec|\!)((\!|\nu x|\!)B) = (\!|\nu * yvec|\!)((\!|\nu y|\!)B')$
and *L*: *length xvec* = *length yvec*
and *xFreshB'*: $x \sharp B'$
and *xFreshxvec*: $x \sharp xvec$
and *xFreshyvec*: $x \sharp yvec$

shows $(\!|\nu * xvec|\!)B = (\!|\nu * yvec|\!)([(x, y)] \cdot B')$
⟨*proof*⟩

lemma *boundOutputPar1Dest*:
fixes *xvec* :: *name list*
and M :: $'a$::*fs-name*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
and *yvec* :: *name list*
and N :: $'a$
and Q :: ($'a$, $'b$, $'c$) *psi*
and R :: ($'a$, $'b$, $'c$) *psi*

assumes $(\!|\nu * xvec|\!)M \prec' P = (\!|\nu * yvec|\!)N \prec' (Q \parallel R)$

and $xvec \sharp* R$
and $yvec \sharp* R$

obtains T **where** $P = T \parallel R$ **and** $(\!|\nu*xvec|\!)M \prec' T = (\!|\nu*yvec|\!)N \prec' Q$
⟨*proof*⟩

lemma *boundOutputPar1Dest'*:
fixes $xvec$ :: *name list*
and M :: $'a$::*fs-name*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
and $yvec$ :: *name list*
and N :: $'a$
and Q :: ($'a$, $'b$, $'c$) *psi*
and R :: ($'a$, $'b$, $'c$) *psi*

assumes $(\!|\nu*xvec|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' (Q \parallel R)$
and $xvec \sharp* yvec$

obtains T p **where** *set* $p \subseteq$ *set* $xvec \times$ *set* $yvec$ **and** $P = T \parallel (p \cdot R)$ **and**
$(\!|\nu*xvec|\!)M \prec' T = (\!|\nu*yvec|\!)N \prec' Q$
⟨*proof*⟩

lemma *boundOutputPar2Dest*:
fixes $xvec$ :: *name list*
and M :: $'a$::*fs-name*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
and $yvec$ :: *name list*
and N :: $'a$
and Q :: ($'a$, $'b$, $'c$) *psi*
and R :: ($'a$, $'b$, $'c$) *psi*

assumes $(\!|\nu*xvec|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' (Q \parallel R)$
and $xvec \sharp* Q$
and $yvec \sharp* Q$

obtains T **where** $P = Q \parallel T$ **and** $(\!|\nu*xvec|\!)M \prec' T = (\!|\nu*yvec|\!)N \prec' R$
⟨*proof*⟩

lemma *boundOutputPar2Dest'*:
fixes $xvec$ :: *name list*
and M :: $'a$::*fs-name*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*
and $yvec$ :: *name list*
and N :: $'a$
and Q :: ($'a$, $'b$, $'c$) *psi*
and R :: ($'a$, $'b$, $'c$) *psi*

assumes $(\!|\nu*xvec|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' (Q \parallel R)$
and $xvec \sharp* yvec$

obtains T p **where** $set\ p \subseteq set\ xvec \times set\ yvec$ **and** $P = (p \cdot Q) \parallel T$ **and** $(\!|\nu{*}xvec|\!)M \prec' T = (\!|\nu{*}yvec|\!)N \prec' R$
⟨*proof*⟩

lemma *boundOutputApp*:
fixes *xvec* :: *name list*
and *yvec* :: *name list*
and B :: ('*a*::*fs-name*, '*b*::*fs-name*, '*c*::*fs-name*) *boundOutput*

shows $(\!|\nu{*}(xvec@yvec)|\!)B = (\!|\nu{*}xvec|\!)((\!|\nu{*}yvec|\!)B)$
⟨*proof*⟩

lemma *openInjectAux*:
fixes *xvec1* :: *name list*
and x :: *name*
and *xvec2* :: *name list*
and *yvec* :: *name list*

assumes $length(xvec1@x\#xvec2) = length\ yvec$

shows $\exists\, yvec1\ y\ yvec2.\ yvec = yvec1@y\#yvec2 \wedge length\ xvec1 = length\ yvec1 \wedge length\ xvec2 = length\ yvec2$
⟨*proof*⟩

lemma *boundOutputOpenDest*:
fixes *yvec* :: *name list*
and M :: '*a*::*fs-name*
and P :: ('*a*, '*b*::*fs-name*, '*c*::*fs-name*) *psi*
and *xvec1* :: *name list*
and x :: *name*
and *xvec2* :: *name list*
and N :: '*a*
and Q :: ('*a*, '*b*, '*c*) *psi*

assumes *Eq*: $(\!|\nu{*}(xvec1@x\#xvec2)|\!)M \prec' P = (\!|\nu{*}yvec|\!)N \prec' Q$
and $x \sharp xvec1$
and $x \sharp yvec$
and $x \sharp N$
and $x \sharp Q$
and *distinct yvec*

obtains *yvec1 y yvec2* **where** *yvec*=*yvec1*@*y*#*yvec2* **and** *length xvec1* = *length yvec1* **and** *length xvec2* = *length yvec2*
and $(\!|\nu{*}(xvec1@xvec2)|\!)M \prec' P = (\!|\nu{*}(yvec1@yvec2)|\!)([(x, y)] \cdot N) \prec' ([(x, y)] \cdot Q)$
⟨*proof*⟩

lemma *boundOutputOpenDest'*:
fixes *yvec* :: *name list*
and *M* :: *'a::fs-name*
and *P* :: *('a, 'b::fs-name, 'c::fs-name) psi*
and *xvec1* :: *name list*
and *x* :: *name*
and *xvec2* :: *name list*
and *N* :: *'a*
and *Q* :: *('a, 'b, 'c) psi*

assumes *Eq*: $(\!|\nu*(xvec1@x\#xvec2)|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' Q$
and $x \sharp xvec1$
and $x \sharp yvec$
and $x \sharp N$
and $x \sharp Q$

obtains *yvec1 y yvec2* **where** *yvec=yvec1@y#yvec2* **and** *length xvec1 = length yvec1* **and** *length xvec2 = length yvec2*
and $(\!|\nu*(xvec1@xvec2)|\!)M \prec' P = (\!|\nu*(yvec1@[(x, y)] \cdot yvec2)|\!)([(x, y)] \cdot N) \prec' ([(x, y)] \cdot Q)$
⟨*proof*⟩

lemma *boundOutputScopeDest*:
fixes *xvec* :: *name list*
and *M* :: *'a::fs-name*
and *P* :: *('a, 'b::fs-name, 'c::fs-name) psi*
and *yvec* :: *name list*
and *N* :: *'a*
and *x* :: *name*
and *Q* :: *('a, 'b, 'c) psi*

assumes $(\!|\nu*xvec|\!)M \prec' P = (\!|\nu*yvec|\!)N \prec' (\!|\nu z|\!)Q$
and $z \sharp xvec$
and $z \sharp yvec$

obtains *R* **where** $P = (\!|\nu z|\!)R$ **and** $(\!|\nu*xvec|\!)M \prec' R = (\!|\nu*yvec|\!)N \prec' Q$
⟨*proof*⟩

nominal-datatype *('a, 'b, 'c) residual* =
RIn 'a::fs-name 'a ('a, 'b::fs-name, 'c::fs-name) psi
| *RBrIn 'a::fs-name 'a ('a, 'b::fs-name, 'c::fs-name) psi*
| *ROut 'a ('a, 'b, 'c) boundOutput*
| *RBrOut 'a ('a, 'b, 'c) boundOutput*
| *RTau ('a, 'b, 'c) psi*

nominal-datatype *'a action* = *In 'a::fs-name 'a* (‹-(|-|)› [*90*, *90*] *90*)
| *BrIn 'a::fs-name 'a* (‹¿-(|-|)› [*90*, *90*] *90*)
| *Out 'a::fs-name name list 'a* (‹-(|ν*-|)⟨-⟩› [*90*, *90*, *90*] *90*)

```
  | BrOut 'a::fs-name name list 'a (‹¡-(⦇ν*-⦈)⟨-⟩› [90, 90, 90] 90)
  | Tau                                (‹τ› 90)

nominal-primrec bn :: ('a::fs-name) action ⇒ name list
  where
    bn (M(⦇N⦈)) = []
  | bn (¿M(⦇N⦈)) = []
  | bn (M(⦇ν*xvec⦈)⟨N⟩) = xvec
  | bn (¡M(⦇ν*xvec⦈)⟨N⟩) = xvec
  | bn (τ) = []
  ⟨proof⟩

lemma bnEqvt[eqvt]:
  fixes p :: name prm
    and α :: ('a::fs-name) action

shows (p ∙ bn α) = bn(p ∙ α)
  ⟨proof⟩

nominal-primrec create-residual :: ('a::fs-name) action ⇒ ('a, 'b::fs-name, 'c::fs-name)
psi ⇒ ('a, 'b, 'c) residual (‹- ≺ -› [80, 80] 80)
  where
    (M(⦇N⦈)) ≺ P = RIn M N P
  | (¿M(⦇N⦈)) ≺ P = RBrIn M N P
  | M(⦇ν*xvec⦈)⟨N⟩ ≺ P = ROut M ((⦇ν*xvec⦈)(N ≺' P))
  | (¡M(⦇ν*xvec⦈)⟨N⟩) ≺ P = RBrOut M ((⦇ν*xvec⦈)(N ≺' P))
  | τ ≺ P = (RTau P)
  ⟨proof⟩

nominal-primrec subject :: ('a::fs-name) action ⇒ 'a option
  where
    subject (M(⦇N⦈)) = Some M
  | subject (¿M(⦇N⦈)) = Some M
  | subject (M(⦇ν*xvec⦈)⟨N⟩) = Some M
  | subject (¡M(⦇ν*xvec⦈)⟨N⟩) = Some M
  | subject (τ) = None
  ⟨proof⟩

nominal-primrec object :: ('a::fs-name) action ⇒ 'a option
  where
    object (M(⦇N⦈)) = Some N
  | object (¿M(⦇N⦈)) = Some N
  | object (M(⦇ν*xvec⦈)⟨N⟩) = Some N
  | object (¡M(⦇ν*xvec⦈)⟨N⟩) = Some N
  | object (τ) = None
  ⟨proof⟩

lemma optionFreshChain[simp]:
  fixes xvec :: name list
```

and X :: *name set*

shows $xvec \sharp* (Some\ x) = xvec \sharp* x$
and $X \sharp* (Some\ x) = X \sharp* x$
and $xvec \sharp* None$
and $X \sharp* None$
⟨*proof*⟩

lemmas [*simp*] = *fresh-some fresh-none*

lemma *actionFresh*[*simp*]:
fixes x :: *name*
and α :: ('*a*::*fs-name*) *action*

shows $(x \sharp \alpha) = (x \sharp (subject\ \alpha) \wedge x \sharp (bn\ \alpha) \wedge x \sharp (object\ \alpha))$
⟨*proof*⟩

lemma *actionFreshChain*[*simp*]:
fixes X :: *name set*
and α :: ('*a*::*fs-name*) *action*
and *xvec* :: *name list*

shows $(X \sharp* \alpha) = (X \sharp* (subject\ \alpha) \wedge X \sharp* (bn\ \alpha) \wedge X \sharp* (object\ \alpha))$
and $(xvec \sharp* \alpha) = (xvec \sharp* (subject\ \alpha) \wedge xvec \sharp* (bn\ \alpha) \wedge xvec \sharp* (object\ \alpha))$
⟨*proof*⟩

lemma *subjectEqvt*[*eqvt*]:
fixes p :: *name prm*
and α :: ('*a*::*fs-name*) *action*

shows $(p \cdot subject\ \alpha) = subject(p \cdot \alpha)$
⟨*proof*⟩

lemma *okjectEqvt*[*eqvt*]:
fixes p :: *name prm*
and α :: ('*a*::*fs-name*) *action*

shows $(p \cdot object\ \alpha) = object(p \cdot \alpha)$
⟨*proof*⟩

lemma *create-residualEqvt*[*eqvt*]:
fixes p :: *name prm*
and α :: ('*a*::*fs-name*) *action*
and P :: ('*a*, '*b*::*fs-name*, '*c*::*fs-name*) *psi*

shows $(p \cdot (\alpha \prec P)) = (p \cdot \alpha) \prec (p \cdot P)$
⟨*proof*⟩

lemma *residualFresh*:

fixes x :: *name*
and α :: $'a$::*fs-name action*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*

shows $(x \sharp (\alpha \prec P)) = (x \sharp (subject\ \alpha) \wedge (x \in (set(bn(\alpha))) \vee (x \sharp object(\alpha) \wedge x \sharp P)))$
⟨*proof*⟩

lemma *residualFresh2*[*simp*]:
fixes x :: *name*
and α :: ($'a$::*fs-name*) *action*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*

assumes $x \sharp \alpha$
and $x \sharp P$

shows $x \sharp \alpha \prec P$
⟨*proof*⟩

lemma *residualFreshChain2*[*simp*]:
fixes *xvec* :: *name list*
and X :: *name set*
and α :: ($'a$::*fs-name*) *action*
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*

shows $[\![xvec \sharp* \alpha; xvec \sharp* P]\!] \Longrightarrow xvec \sharp* (\alpha \prec P)$
and $[\![X \sharp* \alpha; X \sharp* P]\!] \Longrightarrow X \sharp* (\alpha \prec P)$
⟨*proof*⟩

lemma *residualFreshSimp*[*simp*]:
fixes x :: *name*
and M :: $'a$::*fs-name*
and N :: $'a$
and P :: ($'a$, $'b$::*fs-name*, $'c$::*fs-name*) *psi*

shows $x \sharp (M(\!|N|\!) \prec P) = (x \sharp M \wedge x \sharp N \wedge x \sharp P)$
and $x \sharp (¿\ M(\!|N|\!) \prec P) = (x \sharp M \wedge x \sharp N \wedge x \sharp P)$
and $x \sharp (M(\!|\nu*xvec|\!)\langle N\rangle \prec P) = (x \sharp M \wedge x \sharp ((\!|\nu*xvec|\!)(N \prec' P)))$
and $x \sharp (¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P) = (x \sharp M \wedge x \sharp ((\!|\nu*xvec|\!)(N \prec' P)))$
and $x \sharp (\tau \prec P) = (x \sharp P)$
⟨*proof*⟩

lemma *residualInject'*:

shows $(\alpha \prec P = RIn\ M\ N\ Q) = (P = Q \wedge \alpha = M(\!|N|\!))$
and $(\alpha \prec P = RBrIn\ M\ N\ Q) = (P = Q \wedge \alpha = ¿M(\!|N|\!))$
and $(\alpha \prec P = ROut\ M\ B) = (\exists\, xvec\ N.\ \alpha = M(\!|\nu*xvec|\!)\langle N\rangle \wedge B = (\!|\nu*xvec|\!)(N \prec' P))$

and $(\alpha \prec P = RBrOut\ M\ B) = (\exists\, xvec\ N.\ \alpha = ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \wedge B = (\!|\nu{*}xvec|\!)(N \prec' P))$
and $(\alpha \prec P = RTau\ Q) = (\alpha = \tau \wedge P = Q)$
and $(RIn\ M\ N\ Q = \alpha \prec P) = (P = Q \wedge \alpha = M(\!|N|\!))$
and $(RBrIn\ M\ N\ Q = \alpha \prec P) = (P = Q \wedge \alpha = ¿M(\!|N|\!))$
and $(ROut\ M\ B = \alpha \prec P) = (\exists\, xvec\ N.\ \alpha = M(\!|\nu{*}xvec|\!)\langle N\rangle \wedge B = (\!|\nu{*}xvec|\!)(N \prec' P))$
and $(RBrOut\ M\ B = \alpha \prec P) = (\exists\, xvec\ N.\ \alpha = ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \wedge B = (\!|\nu{*}xvec|\!)(N \prec' P))$
and $(RTau\ Q = \alpha \prec P) = (\alpha = \tau \wedge P = Q)$
⟨*proof*⟩

lemma *residualFreshChainSimp*[*simp*]:
fixes *xvec* :: *name list*
and *X* :: *name set*
and *M* :: *'a*::*fs-name*
and *N* :: *'a*
and *yvec* :: *name list*
and *P* :: (*'a*, *'b*::*fs-name*, *'c*::*fs-name*) *psi*

shows $xvec \sharp* (M(\!|N|\!) \prec P) = (xvec \sharp* M \wedge xvec \sharp* N \wedge xvec \sharp* P)$
and $xvec \sharp* (¿M(\!|N|\!) \prec P) = (xvec \sharp* M \wedge xvec \sharp* N \wedge xvec \sharp* P)$
and $xvec \sharp* (M(\!|\nu{*}yvec|\!)\langle N\rangle \prec P) = (xvec \sharp* M \wedge xvec \sharp* ((\!|\nu{*}yvec|\!)(N \prec' P)))$
and $xvec \sharp* (¡M(\!|\nu{*}yvec|\!)\langle N\rangle \prec P) = (xvec \sharp* M \wedge xvec \sharp* ((\!|\nu{*}yvec|\!)(N \prec' P)))$
and $xvec \sharp* (\tau \prec P) = (xvec \sharp* P)$
and $X \sharp* (M(\!|N|\!) \prec P) = (X \sharp* M \wedge X \sharp* N \wedge X \sharp* P)$
and $X \sharp* (¿M(\!|N|\!) \prec P) = (X \sharp* M \wedge X \sharp* N \wedge X \sharp* P)$
and $X \sharp* (M(\!|\nu{*}yvec|\!)\langle N\rangle \prec P) = (X \sharp* M \wedge X \sharp* ((\!|\nu{*}yvec|\!)(N \prec' P)))$
and $X \sharp* (¡M(\!|\nu{*}yvec|\!)\langle N\rangle \prec P) = (X \sharp* M \wedge X \sharp* ((\!|\nu{*}yvec|\!)(N \prec' P)))$
and $X \sharp* (\tau \prec P) = (X \sharp* P)$
⟨*proof*⟩

lemma *residualFreshChainSimp2*[*simp*]:
fixes *xvec* :: *name list*
and *X* :: *name set*
and *M* :: *'a*::*fs-name*
and *N* :: *'a*
and *yvec* :: *name list*
and *P* :: (*'a*, *'b*::*fs-name*, *'c*::*fs-name*) *psi*

shows $xvec \sharp* (RIn\ M\ N\ P) = (xvec \sharp* M \wedge xvec \sharp* N \wedge xvec \sharp* P)$
and $xvec \sharp* (RBrIn\ M\ N\ P) = (xvec \sharp* M \wedge xvec \sharp* N \wedge xvec \sharp* P)$
and $xvec \sharp* (ROut\ M\ B) = (xvec \sharp* M \wedge xvec \sharp* B)$
and $xvec \sharp* (RBrOut\ M\ B) = (xvec \sharp* M \wedge xvec \sharp* B)$
and $xvec \sharp* (RTau\ P) = (xvec \sharp* P)$
and $X \sharp* (RIn\ M\ N\ P) = (X \sharp* M \wedge X \sharp* N \wedge X \sharp* P)$
and $X \sharp* (RBrIn\ M\ N\ P) = (X \sharp* M \wedge X \sharp* N \wedge X \sharp* P)$
and $X \sharp* (ROut\ M\ B) = (X \sharp* M \wedge X \sharp* B)$
and $X \sharp* (RBrOut\ M\ B) = (X \sharp* M \wedge X \sharp* B)$

and $X \sharp* (RTau\ P) = (X \sharp* P)$
⟨*proof*⟩

lemma *freshResidual3*[*dest*]:
fixes x :: *name*
and α :: ('*a*::*fs-name*) *action*
and P :: ('*a*, '*b*::*fs-name*, '*c*::*fs-name*) *psi*

assumes $x \sharp bn\ \alpha$
and $x \sharp \alpha \prec P$

shows $x \sharp \alpha$ **and** $x \sharp P$
⟨*proof*⟩

lemma *freshResidualChain3*[*dest*]:
fixes *xvec* :: *name list*
and α :: ('*a*::*fs-name*) *action*
and P :: ('*a*, '*b*::*fs-name*, '*c*::*fs-name*) *psi*

assumes *xvec* $\sharp*$ $(\alpha \prec P)$
and *xvec* $\sharp*$ *bn* α

shows *xvec* $\sharp*$ α **and** *xvec* $\sharp*$ P
⟨*proof*⟩

lemma *freshResidual4*[*dest*]:
fixes x :: *name*
and α :: ('*a*::*fs-name*) *action*
and P :: ('*a*, '*b*::*fs-name*, '*c*::*fs-name*) *psi*

assumes $x \sharp \alpha \prec P$

shows $x \sharp$ *subject* α
⟨*proof*⟩

lemma *freshResidualChain4*[*dest*]:
fixes *xvec* :: *name list*
and α :: ('*a*::*fs-name*) *action*
and P :: ('*a*, '*b*::*fs-name*, '*c*::*fs-name*) *psi*

assumes *xvec* $\sharp*$ $(\alpha \prec P)$

shows *xvec* $\sharp*$ *subject* α
⟨*proof*⟩

lemma *alphaOutputResidual*:
fixes M :: '*a*::*fs-name*
and *xvec* :: *name list*
and N :: '*a*

and P :: ('a, 'b::*fs-name*, 'c::*fs-name*) *psi*
and p :: *name prm*

assumes $(p \cdot xvec) \sharp* N$
and $(p \cdot xvec) \sharp* P$
and $set\ p \subseteq set\ xvec \times set(p \cdot xvec)$
and $set\ xvec \subseteq set\ yvec$

shows $M(\!|\nu* yvec|\!)\langle N\rangle \prec P = M(\!|\nu*(p \cdot yvec)|\!)\langle(p \cdot N)\rangle \prec (p \cdot P)$
and $¡M(\!|\nu* yvec|\!)\langle N\rangle \prec P = ¡M(\!|\nu*(p \cdot yvec)|\!)\langle(p \cdot N)\rangle \prec (p \cdot P)$
⟨*proof*⟩

lemmas[*simp del*] = *create-residual.simps*

lemma *residualInject''*:

assumes $bn\ \alpha = bn\ \beta$

shows $(\alpha \prec P = \beta \prec Q) = (\alpha = \beta \wedge P = Q)$
⟨*proof*⟩

lemmas *residualInject* = *residual.inject create-residual.simps residualInject' residualInject''*

lemma *bnFreshResidual*[*simp*]:
fixes α :: ('a::*fs-name*) *action*

shows $(bn\ \alpha) \sharp* (\alpha \prec P) = bn\ \alpha \sharp* (subject\ \alpha)$
⟨*proof*⟩

lemma *actionCases*[*case-names cInput cBrInput cOutput cBrOutput cTau*]:
fixes α :: ('a::*fs-name*) *action*

assumes $\bigwedge M\ N.\ \alpha = M(\!|N|\!) \Longrightarrow Prop$
and $\bigwedge M\ N.\ \alpha = ¿M(\!|N|\!) \Longrightarrow Prop$
and $\bigwedge M\ xvec\ N.\ \alpha = M(\!|\nu* xvec|\!)\langle N\rangle \Longrightarrow Prop$
and $\bigwedge M\ xvec\ N.\ \alpha = ¡M(\!|\nu* xvec|\!)\langle N\rangle \Longrightarrow Prop$
and $\alpha = \tau \Longrightarrow Prop$

shows *Prop*
⟨*proof*⟩

lemma *actionPar1Dest*:
fixes α :: ('a::*fs-name*) *action*
and P :: ('a, 'b::*fs-name*, 'c::*fs-name*) *psi*
and β :: ('a::*fs-name*) *action*
and Q :: ('a, 'b, 'c) *psi*
and R :: ('a, 'b, 'c) *psi*

assumes $\alpha \prec P = \beta \prec (Q \parallel R)$
and $bn\ \alpha \sharp* bn\ \beta$

obtains $T\ p$ **where** $set\ p \subseteq set(bn\ \alpha) \times set(bn\ \beta)$ **and** $P = T \parallel (p \cdot R)$ **and** $\alpha \prec T = \beta \prec Q$
⟨*proof*⟩

lemma *actionPar2Dest*:
fixes $\alpha :: ('a::fs\text{-}name)\ action$
and $P :: ('a, 'b::fs\text{-}name, 'c::fs\text{-}name)\ psi$
and $\beta :: ('a::fs\text{-}name)\ action$
and $Q :: ('a, 'b, 'c)\ psi$
and $R :: ('a, 'b, 'c)\ psi$

assumes $\alpha \prec P = \beta \prec (Q \parallel R)$
and $bn\ \alpha \sharp* bn\ \beta$

obtains $T\ p$ **where** $set\ p \subseteq set(bn\ \alpha) \times set(bn\ \beta)$ **and** $P = (p \cdot Q) \parallel T$ **and** $\alpha \prec T = \beta \prec R$
⟨*proof*⟩

lemma *actionScopeDest*:
fixes $\alpha :: ('a::fs\text{-}name)\ action$
and $P :: ('a, 'b::fs\text{-}name, 'c::fs\text{-}name)\ psi$
fixes $\beta :: ('a::fs\text{-}name)\ action$
and $x :: name$
and $Q :: ('a, 'b, 'c)\ psi$

assumes $\alpha \prec P = \beta \prec (\!|\nu x|\!)\, Q$
and $x \sharp bn\ \alpha$
and $x \sharp bn\ \beta$

obtains R **where** $P = (\!|\nu x|\!) R$ **and** $\alpha \prec R = \beta \prec Q$
⟨*proof*⟩

lemma *emptyFreshName*:
fixes $x :: name$
and $M :: 'a::fs\text{-}name$

assumes $supp\ M = (\{\}::name\ set)$

shows $x \sharp M$
⟨*proof*⟩

lemma *emptyFresh*:
fixes $xvec :: name\ list$
and $M :: 'a::fs\text{-}name$

assumes $supp\ M = (\{\}::name\ set)$

shows *xvec* $\sharp*$ *M*
⟨*proof*⟩

lemma *permEmptyEq*:
fixes *p* :: *name prm*
and *M* :: *'a::fs-name*

assumes *suppE*: *supp M* = ({}::*name set*)

shows (*p* ∙ *M*) = *M*
⟨*proof*⟩

abbreviation
outputJudge (‹-⟨-⟩› [*110*, *110*] *110*) **where** $M\langle N\rangle \equiv M(\!|\nu*([])|\!)\langle N\rangle$

abbreviation
brOutputJudge (‹¡-⟨-⟩› [*110*, *110*] *110*) **where** $¡M\langle N\rangle \equiv ¡M(\!|\nu*([])|\!)\langle N\rangle$

declare [[*unify-trace-bound=100*]]

locale *env* = *substPsi substTerm substAssert substCond* +
assertion SCompose' SImp' SBottom' SChanEq' SOutCon' SInCon'
for *substTerm* :: (*'a::fs-name*) ⇒ *name list* ⇒ *'a::fs-name list* ⇒ *'a*
and *substAssert* :: (*'b::fs-name*) ⇒ *name list* ⇒ *'a::fs-name list* ⇒ *'b*
and *substCond* :: (*'c::fs-name*) ⇒ *name list* ⇒ *'a::fs-name list* ⇒ *'c*
and *SCompose'* :: *'b* ⇒ *'b* ⇒ *'b*
and *SImp'* :: *'b* ⇒ *'c* ⇒ *bool*
and *SBottom'* :: *'b*
and *SChanEq'* :: *'a* ⇒ *'a* ⇒ *'c*
and *SOutCon'* :: *'a* ⇒ *'a* ⇒ *'c*
and *SInCon'* :: *'a* ⇒ *'a* ⇒ *'c*
begin
notation *SCompose'* (**infixr** ‹⊗› *90*)
notation *SImp'* (‹- ⊢ -› [*85*, *85*] *85*)
notation *FrameImp* (‹- $\vdash_F$ -› [*85*, *85*] *85*)
abbreviation
FBottomJudge (‹$\bot_F$› *90*) **where** $\bot_F \equiv (FAssert\ SBottom')$
notation *SChanEq'* (‹- ↔ -› [*90*, *90*] *90*)
notation *SOutCon'* (‹- ≼ -› [*90*, *90*] *90*)
notation *SInCon'* (‹- ≽ -› [*90*, *90*] *90*)
notation *substTerm* (‹-[-::=-]› [*100*, *100*, *100*] *100*)
notation *subs* (‹-[-::=-]› [*100*, *100*, *100*] *100*)
notation *AssertionStatEq* (‹- ≃ -› [*80*, *80*] *80*)
notation *FrameStatEq* (‹- $\simeq_F$ -› [*80*, *80*] *80*)
notation *SBottom'* (‹**1**› *190*)
abbreviation *insertAssertion'* (‹*insertAssertion*›) **where** *insertAssertion'* ≡ *assertionAux.insertAssertion* (⊗)

inductive *semantics* :: $'b \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow ('a, 'b, 'c)\ residual \Rightarrow bool$
($\langle$- $\rhd$ - $\longmapsto$ -$\rangle$ [50, 50, 50] 50)
where
cInput: $[\![\Psi \vdash M \leftrightarrow K;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;\ xvec \sharp* Tvec;$
$length\ xvec = length\ Tvec;$
$xvec \sharp* \Psi;\ xvec \sharp* M;\ xvec \sharp* K]\!] \Longrightarrow \Psi \rhd M(\!|\lambda* xvec\ N|\!).P \longmapsto K(\!|(N[xvec::=Tvec])|\!) \prec P[xvec::=Tvec]$
| *cBrInput*:$[\![\Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;\ xvec \sharp* Tvec;$
$length\ xvec = length\ Tvec;$
$xvec \sharp* \Psi;\ xvec \sharp* M;\ xvec \sharp* K]\!] \Longrightarrow \Psi \rhd M(\!|\lambda* xvec\ N|\!).P \longmapsto ¿K(\!|(N[xvec::=Tvec])|\!) \prec P[xvec::=Tvec]$
| *Output*: $[\![\Psi \vdash M \leftrightarrow K]\!] \Longrightarrow \Psi \rhd M\langle N\rangle.P \longmapsto K\langle N\rangle \prec P$
| *BrOutput*: $[\![\Psi \vdash M \preceq K]\!] \Longrightarrow \Psi \rhd M\langle N\rangle.P \longmapsto ¡K\langle N\rangle \prec P$
| *Case*: $[\![\Psi \rhd P \longmapsto Rs;\ (\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P]\!] \Longrightarrow \Psi \rhd Cases\ Cs \longmapsto Rs$
| *cPar1*: $[\![(\Psi \otimes \Psi_Q) \rhd P \longmapsto \alpha \prec P';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* \alpha;\ A_Q \sharp* P';\ distinct(bn\ \alpha);$
$bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* \Psi_Q;\ bn\ \alpha \sharp* Q;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* (subject\ \alpha)]\!] \Longrightarrow$
$\Psi \rhd P \parallel Q \longmapsto \alpha \prec (P' \parallel Q)$
| *cPar2*: $[\![(\Psi \otimes \Psi_P) \rhd Q \longmapsto \alpha \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;\ A_P \sharp* \alpha;\ A_P \sharp* Q';\ distinct(bn\ \alpha);$
$bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* \Psi_P;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* Q;\ bn\ \alpha \sharp* (subject\ \alpha)]\!] \Longrightarrow$
$\Psi \rhd P \parallel Q \longmapsto \alpha \prec (P \parallel Q')$
| *cComm1*: $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu* xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* xvec;$
$A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;\ A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';$
$A_Q \sharp* Q;\ A_Q \sharp* K;\ A_Q \sharp* Q';\ A_Q \sharp* xvec;\ distinct\ xvec;$
$xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;\ xvec \sharp* M;$
$xvec \sharp* Q;\ xvec \sharp* K]\!] \Longrightarrow$
$\Psi \rhd P \parallel Q \longmapsto \tau \prec (\!|\nu* xvec|\!)(P' \parallel Q')$
| *cComm2*: $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu* xvec|\!)\langle N\rangle \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* xvec;$
$A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;\ A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';$
$A_Q \sharp* Q;\ A_Q \sharp* K;\ A_Q \sharp* Q';\ A_Q \sharp* xvec;\ distinct\ xvec;$
$xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;\ xvec \sharp* M;$
$xvec \sharp* Q;\ xvec \sharp* K]\!] \Longrightarrow$
$\Psi \rhd P \parallel Q \longmapsto \tau \prec (\!|\nu* xvec|\!)(P' \parallel Q')$
| *cBrMerge*: $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$

$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$;
$A_P \sharp* M$; $A_Q \sharp* M$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$;
$A_Q \sharp* Q$; $A_Q \sharp* Q']\!] \Longrightarrow$
$\Psi \rhd P \parallel Q \longmapsto ¿M(\!|N|\!) \prec (P' \parallel Q')$

| *cBrComm1*: $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$;
$A_P \sharp* M$; $A_Q \sharp* M$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$;
$A_Q \sharp* Q$; $A_Q \sharp* Q'$; $A_Q \sharp* xvec$; *distinct* $xvec$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $xvec \sharp* M]\!] \Longrightarrow$
$\Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec (P' \parallel Q')$
| *cBrComm2*: $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$;
$A_P \sharp* M$; $A_Q \sharp* M$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$;
$A_Q \sharp* Q$; $A_Q \sharp* Q'$; $A_Q \sharp* xvec$; *distinct* $xvec$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $xvec \sharp* M]\!] \Longrightarrow$
$\Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec (P' \parallel Q')$
| *cBrClose*: $[\![\Psi \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P'$;
$x \in supp\ M$;
distinct $xvec$; $xvec \sharp* \Psi$; $xvec \sharp* P$;
$xvec \sharp* M$;
$x \sharp \Psi$; $x \sharp xvec]\!] \Longrightarrow$
$\Psi \rhd (\!|\nu x|\!)P \longmapsto \tau \prec (\!|\nu x|\!)((\!|\nu*xvec|\!)P')$
| *cOpen*: $[\![\Psi \rhd P \longmapsto M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'$; $x \in supp\ N$; $x \sharp xvec$; $x \sharp yvec$; $x \sharp M$; $x \sharp \Psi$;
distinct $xvec$; *distinct* $yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* yvec$; $yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M]\!] \Longrightarrow$
$\Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|\nu*(xvec@x\#yvec)|\!)\langle N\rangle \prec P'$
| *cBrOpen*: $[\![\Psi \rhd P \longmapsto ¡M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'$; $x \in supp\ N$; $x \sharp xvec$; $x \sharp yvec$; $x \sharp M$; $x \sharp \Psi$;
distinct $xvec$; *distinct* $yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* yvec$; $yvec \sharp* \Psi$; $yvec \sharp* P$;

yvec $\sharp*$ *M*$\rrbracket \Longrightarrow$
$\Psi \rhd (\!|\nu x|\!)P \longmapsto ¡M(\!|\nu*(xvec@x\#yvec)|\!)\langle N\rangle \prec P'$
| *cScope*: $\llbracket \Psi \rhd P \longmapsto \alpha \prec P'$; $x \sharp \Psi$; $x \sharp \alpha$; *bn* $\alpha \sharp* \Psi$; *bn* $\alpha \sharp* P$; *bn* $\alpha \sharp*$ (*subject* α); *distinct*(*bn* α)$\rrbracket \Longrightarrow \Psi \rhd (\!|\nu x|\!)P \longmapsto \alpha \prec ((\!|\nu x|\!)P')$
| *Bang*: $\llbracket \Psi \rhd P \parallel !P \longmapsto Rs$; *guarded* $P\rrbracket \Longrightarrow \Psi \rhd !P \longmapsto Rs$

abbreviation
semanticsBottomJudge ($\langle$- $\longmapsto$ -$\rangle$ [50, 50] 50) **where** $P \longmapsto Rs \equiv \mathbf{1} \rhd P \longmapsto Rs$

equivariance *env.semantics*

nominal-inductive2 *env.semantics*
avoids *cInput*: *set xvec*
| *cBrInput*: *set xvec*
| *cPar1*: *set* $A_Q \cup$ *set*(*bn* α)
| *cPar2*: *set* $A_P \cup$ *set*(*bn* α)
| *cComm1*: *set* $A_P \cup$ *set* $A_Q \cup$ *set xvec*
| *cComm2*: *set* $A_P \cup$ *set* $A_Q \cup$ *set xvec*
| *cBrMerge*: *set* $A_P \cup$ *set* A_Q
| *cBrComm1*: *set* $A_P \cup$ *set* $A_Q \cup$ *set xvec*
| *cBrComm2*: *set* $A_P \cup$ *set* $A_Q \cup$ *set xvec*
| *cBrClose*: $\{x\} \cup$ *set xvec*
| *cOpen*: $\{x\} \cup$ *set xvec* $\cup$ *set yvec*
| *cBrOpen*: $\{x\} \cup$ *set xvec* $\cup$ *set yvec*
| *cScope*: $\{x\} \cup$ *set*(*bn* α)
⟨*proof*⟩

lemma *nilTrans1*:
fixes Ψ :: *'b*
and *M* :: *'a*
and *xvec* :: *name list*
and *N* :: *'a*
and *P* :: (*'a*, *'b*, *'c*) *psi*

assumes $\Psi \rhd \mathbf{0} \longmapsto M(\!|\nu*xvec|\!)\langle N\rangle \prec P$

shows *False*
⟨*proof*⟩

lemma *nilTrans1'*:
fixes Ψ :: *'b*
and *M* :: *'a*
and *xvec* :: *name list*
and *N* :: *'a*
and *P* :: (*'a*, *'b*, *'c*) *psi*

assumes $\Psi \rhd \mathbf{0} \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P$

shows *False*
⟨*proof*⟩

lemma *nilTrans2*:
fixes Ψ :: *'b*
and *Rs* :: (*'a*, *'b*, *'c*) *residual*

assumes Ψ ▷ **0** ⟼ *Rs*

shows *False*
⟨*proof*⟩

lemma *nilTrans3*:
fixes Ψ :: *'b*
and *M* :: *'a*
and *M'* :: *'a*
and *xvec* :: *name list*
and *yvec* :: *name list*
and *N* :: *'a*
and *N'* :: *'a*
and *P* :: (*'a*, *'b*, *'c*) *psi*
and *P'* :: (*'a*, *'b*, *'c*) *psi*

assumes Ψ ▷ *M*⦇λ∗*xvec N*⦈.*P* ⟼ *M'*⦇ν∗*yvec*⦈⟨*N'*⟩ ≺ *P'*

shows *False*
⟨*proof*⟩

lemma *nilTrans3'*:
fixes Ψ :: *'b*
and *M* :: *'a*
and *M'* :: *'a*
and *xvec* :: *name list*
and *yvec* :: *name list*
and *N* :: *'a*
and *N'* :: *'a*
and *P* :: (*'a*, *'b*, *'c*) *psi*
and *P'* :: (*'a*, *'b*, *'c*) *psi*

assumes Ψ ▷ *M*⦇λ∗*xvec N*⦈.*P* ⟼ ¡*M'*⦇ν∗*yvec*⦈⟨*N'*⟩ ≺ *P'*

shows *False*
⟨*proof*⟩

lemma *nilTrans4*:
fixes Ψ :: *'b*
and *Rs* :: (*'a*, *'b*, *'c*) *residual*

assumes Ψ ▷ *M*⦇λ∗*xvec N*⦈.*P* ⟼τ ≺ *P'*

shows *False*
⟨*proof*⟩

lemma *nilTrans5*:
fixes Ψ :: $'b$
fixes Ψ' :: $'b$
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*

assumes $\Psi \rhd \{\!|\Psi'|\!\} \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P$

shows *False*
⟨*proof*⟩

lemma *nilTrans5'*:
fixes Ψ :: $'b$
fixes Ψ' :: $'b$
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*

assumes $\Psi \rhd \{\!|\Psi'|\!\} \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P$

shows *False*
⟨*proof*⟩

lemma *nilTrans6*:
fixes Ψ :: $'b$
and *Rs* :: $('a, 'b, 'c)$ *residual*

assumes $\Psi \rhd \{\!|\Psi'|\!\} \longmapsto Rs$

shows *False*
⟨*proof*⟩

lemma *nilTrans*[*dest*]:
fixes Ψ :: $'b$
and *Rs* :: $('a, 'b, 'c)$ *residual*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and K :: $'a$
and *yvec* :: *name list*
and N' :: $'a$

and P' :: $('a, 'b, 'c)$ *psi*
and *CsP* :: $('c \times ('a, 'b, 'c)$ *psi*$)$ *list*
and Ψ' :: $'b$

shows $\Psi \rhd \mathbf{0} \longmapsto Rs \Longrightarrow False$
and $\Psi \rhd M(\!|\lambda * xvec\ N|\!).P \longmapsto K(\!|\nu * yvec|\!)\langle N'\rangle \prec P' \Longrightarrow False$
and $\Psi \rhd M(\!|\lambda * xvec\ N|\!).P \longmapsto ¡K(\!|\nu * yvec|\!)\langle N'\rangle \prec P' \Longrightarrow False$
and $\Psi \rhd M(\!|\lambda * xvec\ N|\!).P \longmapsto \tau \prec P' \Longrightarrow False$
and $\Psi \rhd M\langle N\rangle.P \longmapsto K(\!|N'|\!) \prec P' \Longrightarrow False$
and $\Psi \rhd M\langle N\rangle.P \longmapsto ¿K(\!|N'|\!) \prec P' \Longrightarrow False$
and $\Psi \rhd M\langle N\rangle.P \longmapsto \tau \prec P' \Longrightarrow False$
and $\Psi \rhd \{\!|\Psi'|\!\} \longmapsto Rs \Longrightarrow False$
⟨*proof*⟩

lemma *residualEq*:
fixes α :: $'a$ *action*
and P :: $('a, 'b, 'c)$ *psi*
and β :: $'a$ *action*
and Q :: $('a, 'b, 'c)$ *psi*

assumes $\alpha \prec P = \beta \prec Q$
and $bn\ \alpha \sharp* (bn\ \beta)$
and $distinct(bn\ \alpha)$
and $distinct(bn\ \beta)$
and $bn\ \alpha \sharp* (\alpha \prec P)$
and $bn\ \beta \sharp* (\beta \prec Q)$

obtains p **where** $set\ p \subseteq set(bn\ \alpha) \times set(bn(p \cdot \alpha))$ **and** $distinctPerm\ p$ **and** $\beta = p \cdot \alpha$ **and** $Q = p \cdot P$ **and** $bn\ \alpha \sharp* \beta$ **and** $bn\ \alpha \sharp* Q$ **and** $bn(p \cdot \alpha) \sharp* \alpha$ **and** $bn(p \cdot \alpha) \sharp* P$
⟨*proof*⟩

lemma *semanticsInduct*[*consumes 3*, *case-names cAlpha cInput cBrInput cOutput cBrOutput cCase cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2 cBrClose cOpen cBrOpen cScope cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*
and *Prop* :: $'f$::*fs-name* ⇒ $'b$ ⇒ $('a, 'b, 'c)$ *psi* ⇒ $'a$ *action* ⇒ $('a, 'b, 'c)$ *psi* ⇒ *bool*
and C :: $'f$::*fs-name*

assumes $\Psi \rhd P \longmapsto \alpha \prec P'$
and $bn\ \alpha \sharp* (subject\ \alpha)$
and $distinct(bn\ \alpha)$
and *rAlpha*: $\bigwedge \Psi\ P\ \alpha\ P'\ p\ C.\ [\![bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* (subject\ \alpha);$
$bn\ \alpha \sharp* C;\ bn\ \alpha \sharp* (bn(p \cdot \alpha));$
$set\ p \subseteq set(bn\ \alpha) \times set(bn(p \cdot \alpha));\ distinctPerm\ p;$

$(bn(p \cdot \alpha))\ \sharp*\ \alpha;\ (bn(p \cdot \alpha))\ \sharp*\ P';\ Prop\ C\ \Psi\ P\ \alpha\ P'\rrbracket \Longrightarrow$

$Prop\ C\ \Psi\ P\ (p \cdot \alpha)\ (p \cdot P')$

and *rInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C.$
$\llbracket \Psi \vdash M \leftrightarrow K;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec\ \sharp*\ \Psi;$
$xvec\ \sharp*\ M;\ xvec\ \sharp*\ K;\ xvec\ \sharp*\ C\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda*xvec\ N|\!).P)$
$(K(\!|(N[xvec::=Tvec])|\!))\ (P[xvec::=Tvec])$

and *rBrInput*:$\bigwedge \Psi\ K\ M\ xvec\ N\ Tvec\ P\ C.$
$\llbracket \Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec\ \sharp*\ \Psi;$
$xvec\ \sharp*\ M;\ xvec\ \sharp*\ K;\ xvec\ \sharp*\ C\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda*xvec\ N|\!).P)$
$(¿K(\!|(N[xvec::=Tvec])|\!))\ (P[xvec::=Tvec])$

and *rOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ \llbracket \Psi \vdash M \leftrightarrow K\rrbracket \Longrightarrow Prop\ C\ \Psi\ (M\langle N\rangle.P)\ (K\langle N\rangle)\ P$

and *rBrOutput*:$\bigwedge \Psi\ M\ K\ N\ P\ C.\ \llbracket \Psi \vdash M \preceq K\rrbracket \Longrightarrow Prop\ C\ \Psi\ (M\langle N\rangle.P)\ (¡K\langle N\rangle)\ P$

and *rCase*: $\bigwedge \Psi\ P\ \alpha\ P'\ \varphi\ Cs\ C.\ \llbracket \Psi \rhd P \longmapsto \alpha \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ P\ \alpha\ P';\ (\varphi,\ P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P\rrbracket \Longrightarrow$

$Prop\ C\ \Psi\ (Cases\ Cs)\ \alpha\ P'$

and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ \alpha\ P'\ A_Q\ Q\ C.$
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P';\ extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ \alpha\ P';$
$A_Q\ \sharp*\ P;\ A_Q\ \sharp*\ Q;\ A_Q\ \sharp*\ \Psi;\ A_Q\ \sharp*\ \alpha;\ A_Q\ \sharp*\ P';\ A_Q\ \sharp*\ C;\ distinct(bn\ \alpha);\ bn\ \alpha\ \sharp*\ Q;$
$bn\ \alpha\ \sharp*\ \Psi;\ bn\ \alpha\ \sharp*\ \Psi_Q;\ bn\ \alpha\ \sharp*\ P;\ bn\ \alpha\ \sharp*\ subject\ \alpha;\ bn\ \alpha\ \sharp*\ C\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ \alpha\ (P' \parallel Q)$

and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ \alpha\ Q'\ A_P\ P\ C.$
$\llbracket \Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ \alpha\ Q';$
$A_P\ \sharp*\ P;\ A_P\ \sharp*\ Q;\ A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ \alpha;\ A_P\ \sharp*\ Q';\ A_P\ \sharp*\ C;\ distinct(bn\ \alpha);\ bn\ \alpha\ \sharp*\ Q;$
$bn\ \alpha\ \sharp*\ \Psi;\ bn\ \alpha\ \sharp*\ \Psi_P;\ bn\ \alpha\ \sharp*\ P;\ bn\ \alpha\ \sharp*\ subject\ \alpha;\ bn\ \alpha\ \sharp*\ C\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ \alpha\ (P \parallel Q')$

and *rComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ K\ xvec\ Q'\ A_Q\ C.$
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ (M(\!|N|\!))\ P';$
$extractFrame\ P = \langle A_P,\ \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu*xvec|\!)\langle N\rangle \prec Q';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ (K(\!|\nu*xvec|\!)\langle N\rangle)\ Q';$
$extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ distinct\ A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;$
$A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ \Psi_Q;\ A_P\ \sharp*\ P;\ A_P\ \sharp*\ M;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ P';$

$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* K$; $A_Q \sharp* Q'$;
distinct xvec;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$; $xvec \sharp* M$;
$xvec \sharp* Q$; $xvec \sharp* K$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ (τ) $((\!|\nu* xvec|\!)(P' \parallel Q'))$

and *rComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ K\ Q'\ A_Q\ C.$
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu* xvec|\!)\langle N\rangle \prec P'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_Q)$ P $(M(\!|\nu* xvec|\!)\langle N\rangle)$ P';
extractFrame $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q $(K(\!|N|\!))$ Q';
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* K$; $A_Q \sharp* Q'$;
distinct xvec;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$; $xvec \sharp* M$;
$xvec \sharp* Q$; $xvec \sharp* K$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ (τ) $((\!|\nu* xvec|\!)(P' \parallel Q'))$

and *rBrMerge*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_Q)$ P $(¿M(\!|N|\!))$ P';
extractFrame $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q $(¿M(\!|N|\!))$ Q';
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* M$; $A_Q \sharp* M$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$;
$A_Q \sharp* Q$; $A_Q \sharp* Q'$; $A_P \sharp* C$; $A_Q \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ $(¿M(\!|N|\!))$ $(P' \parallel Q')$

and *rBrComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ xvec\ Q'\ A_Q\ C.$
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_Q)$ P $(¿M(\!|N|\!))$ P';
extractFrame $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu* xvec|\!)\langle N\rangle \prec Q'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q $(¡M(\!|\nu* xvec|\!)\langle N\rangle)$ Q';
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* Q'$; *distinct xvec*;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ $(¡M(\!|\nu* xvec|\!)\langle N\rangle)$ $(P' \parallel Q')$

and *rBrComm2*: $\bigwedge\Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto {¡}M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ ({¡}M(\!|\nu{*}xvec|\!)\langle N\rangle)\ P';$
$extractFrame\ P = \langle A_P,\ \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto {¿}M(\!|N|\!) \prec Q';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ ({¿}M(\!|N|\!))\ Q';$
$extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ distinct\ A_Q;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* xvec;\ A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;$
$A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* Q;\ A_Q \sharp* Q';\ distinct\ xvec;$
$A_P \sharp* M;\ A_Q \sharp* M;\ xvec \sharp* M;$
$A_Q \sharp* xvec;\ xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;$
$xvec \sharp* Q;\ A_P \sharp* C;\ A_Q \sharp* C;\ xvec \sharp* C[\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ ({¡}M(\!|\nu{*}xvec|\!)\langle N\rangle)\ (P' \parallel Q')$

and *rBrClose*: $\bigwedge\Psi\ P\ M\ xvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto {¡}M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P';$
$\bigwedge C.\ Prop\ C\ \Psi\ P\ ({¡}M(\!|\nu{*}xvec|\!)\langle N\rangle)\ P';$
$x \in supp\ M;$
$distinct\ xvec;\ xvec \sharp* \Psi;\ xvec \sharp* P;$
$xvec \sharp* M;$
$x \sharp \Psi;\ x \sharp xvec[\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ (\tau)\ ((\!|\nu x|\!)((\!|\nu{*}xvec|\!)P'))$

and *rOpen*: $\bigwedge\Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle \prec P';\ x \in supp\ N;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ (M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle)\ P';$
$x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;\ x \sharp yvec;\ xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;\ distinct\ xvec;\ distinct\ yvec;$
$yvec \sharp* \Psi;\ yvec \sharp* P;\ yvec \sharp* M;\ yvec \sharp* C;\ x \sharp C;\ xvec \sharp* C[\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ (M(\!|\nu{*}(xvec@x\#yvec)|\!)\langle N\rangle)\ P'$

and *rBrOpen*: $\bigwedge\Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto {¡}M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle \prec P';\ x \in supp\ N;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ ({¡}M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle)\ P';$
$x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;\ x \sharp yvec;\ xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;\ distinct\ xvec;\ distinct\ yvec;$
$yvec \sharp* \Psi;\ yvec \sharp* P;\ yvec \sharp* M;\ yvec \sharp* C;\ x \sharp C;\ xvec \sharp* C[\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ ({¡}M(\!|\nu{*}(xvec@x\#yvec)|\!)\langle N\rangle)\ P'$

and *rScope*: $\bigwedge\Psi\ P\ \alpha\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto \alpha \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ P\ \alpha\ P';$
$x \sharp \Psi;\ x \sharp \alpha;\ bn\ \alpha \sharp* \Psi;$
$bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* (subject\ \alpha);\ x \sharp C;\ bn\ \alpha \sharp* C;\ distinct(bn\ \alpha)[\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ \alpha\ ((\!|\nu x|\!)P')$

and *rBang*: $\bigwedge\Psi\ P\ \alpha\ P'\ C.$
$[\![\Psi \rhd P \parallel !P \longmapsto \alpha \prec P';\ guarded\ P;\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel !P)\ \alpha\ P'[\!] \Longrightarrow$
$Prop\ C\ \Psi\ (!P)\ \alpha\ P'$

shows $Prop\ C\ \Psi\ P\ \alpha\ P'$
⟨*proof*⟩

lemma *outputInduct*[*consumes 1*, *case-names cOutput cCase cPar1 cPar2 cOpen cScope cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and *Prop* :: $'f$::*fs-name* $\Rightarrow$ $'b \Rightarrow ('a, 'b, 'c)$ *psi* $\Rightarrow$ $'a \Rightarrow ('a, 'b, 'c)$ *boundOutput* $\Rightarrow$ *bool*
and C :: $'f$::*fs-name*

assumes $\Psi \rhd P \longmapsto ROut\ M\ B$
and *rOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ [\![\Psi \vdash M \leftrightarrow K]\!] \Longrightarrow Prop\ C\ \Psi\ (M\langle N\rangle.P)\ K\ (N \prec' P)$
and *rCase*: $\bigwedge \Psi\ P\ M\ B\ \varphi\ Cs\ C.$
$[\![\Psi \rhd P \longmapsto (ROut\ M\ B);\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ B;\ (\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (Cases\ Cs)\ M\ B$
and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ \ P'\ A_Q\ Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ ((\!|\nu * xvec|\!)N \prec' P');$
$A_Q \sharp * P;\ A_Q \sharp * Q;\ A_Q \sharp * \Psi;\ A_Q \sharp * M;$
$A_Q \sharp * xvec;\ A_Q \sharp * N;\ A_Q \sharp * P';\ A_Q \sharp * C;\ xvec \sharp * Q;$
$xvec \sharp * \Psi;\ xvec \sharp * \Psi_Q;\ xvec \sharp * P;\ xvec \sharp * M;\ xvec \sharp * C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu * xvec|\!)N \prec' (P' \parallel Q))$
and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ xvec\ N\ \ Q'\ A_P\ P\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ ((\!|\nu * xvec|\!)N \prec' Q');$
$A_P \sharp * P;\ A_P \sharp * Q;\ A_P \sharp * \Psi;\ A_P \sharp * M;$
$A_P \sharp * xvec;\ A_P \sharp * N;\ A_P \sharp * Q';\ A_P \sharp * C;\ xvec \sharp * P;$
$xvec \sharp * \Psi;\ xvec \sharp * \Psi_P;\ xvec \sharp * Q;\ xvec \sharp * M;\ xvec \sharp * C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu * xvec|\!)N \prec' (P \parallel Q'))$
and *rOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|\nu * (xvec @ yvec)|\!)\langle N\rangle \prec P';\ x \in supp\ N;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ ((\!|\nu * (xvec @ yvec)|\!)N \prec' P');$
$x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;\ x \sharp yvec;\ xvec \sharp * \Psi;\ xvec \sharp * P;\ xvec \sharp * M;$
$xvec \sharp * yvec;\ yvec \sharp * \Psi;\ yvec \sharp * P;\ yvec \sharp * M;\ yvec \sharp * C;\ x \sharp C;\ xvec \sharp * C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ M\ ((\!|\nu * (xvec @ x \# yvec)|\!)N \prec' P')$
and *rScope*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ ((\!|\nu * xvec|\!)N \prec' P');$
$x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;\ x \sharp N;\ xvec \sharp * \Psi;\ xvec \sharp * P;\ xvec \sharp * M;$
$x \sharp C;\ xvec \sharp * C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ M\ ((\!|\nu * xvec|\!)N \prec' (\!|\nu x|\!)P')$
and *rBang*: $\bigwedge \Psi\ P\ M\ B\ C.$
$[\![\Psi \rhd P \parallel\ !P \longmapsto (ROut\ M\ B);\ guarded\ P;\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel$

$!P)\ M\ B\rrbracket \Longrightarrow$

$\quad\quad Prop\ C\ \Psi\ (!P)\ M\ B$

shows $Prop\ C\ \Psi\ P\ M\ B$

$\langle proof\rangle$

lemma *brOutputInduct*[*consumes 1*, *case-names cBrOutput cCase cPar1 cPar2 cBrComm1 cBrComm2 cBrOpen cScope cBang*]:

fixes Ψ :: $'b$

and P :: $('a,\ 'b,\ 'c)\ psi$

and M :: $'a$

and B :: $('a,\ 'b,\ 'c)\ boundOutput$

and $Prop$ :: $'f{::}fs\text{-}name \Rightarrow\ 'b \Rightarrow ('a,\ 'b,\ 'c)\ psi \Rightarrow$
$'a \Rightarrow ('a,\ 'b,\ 'c)\ boundOutput \Rightarrow bool$

and C :: $'f{::}fs\text{-}name$

assumes $\Psi \rhd P \longmapsto RBrOut\ M\ B$

and *rBrOutput*: $\bigwedge\Psi\ M\ K\ N\ P\ C.\ \llbracket\Psi \vdash M \preceq K\rrbracket \Longrightarrow Prop\ C\ \Psi\ (M\langle N\rangle.P)\ K\ (N \prec'\ P)$

and *rCase*: $\bigwedge\Psi\ P\ M\ B\ \varphi\ Cs\ C.$
$\llbracket\Psi \rhd P \longmapsto(RBrOut\ M\ B);\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ B;\ (\varphi,\ P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ (Cases\ Cs)\ M\ B$

and *rPar1*: $\bigwedge\Psi\ \Psi_Q\ P\ M\ xvec\ N\ \ P'\ A_Q\ Q\ C.$
$\llbracket\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P';\ extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ ((\!|\nu{*}xvec|\!)N \prec'\ P');$
$A_Q \sharp{*}\ P;\ A_Q \sharp{*}\ Q;\ A_Q \sharp{*}\ \Psi;\ A_Q \sharp{*}\ M;$
$A_Q \sharp{*}\ xvec;\ A_Q \sharp{*}\ N;\ A_Q \sharp{*}\ P';\ A_Q \sharp{*}\ C;\ xvec \sharp{*}\ Q;$
$xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ \Psi_Q;\ xvec \sharp{*}\ P;\ xvec \sharp{*}\ M;\ xvec \sharp{*}\ C\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu{*}xvec|\!)N \prec'\ (P' \parallel Q))$

and *rPar2*: $\bigwedge\Psi\ \Psi_P\ Q\ M\ xvec\ N\ \ Q'\ A_P\ P\ C.$
$\llbracket\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ ((\!|\nu{*}xvec|\!)N \prec'\ Q');$
$A_P \sharp{*}\ P;\ A_P \sharp{*}\ Q;\ A_P \sharp{*}\ \Psi;\ A_P \sharp{*}\ M;$
$A_P \sharp{*}\ xvec;\ A_P \sharp{*}\ N;\ A_P \sharp{*}\ Q';\ A_P \sharp{*}\ C;\ xvec \sharp{*}\ P;$
$xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ \Psi_P;\ xvec \sharp{*}\ Q;\ xvec \sharp{*}\ M;\ xvec \sharp{*}\ C\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu{*}xvec|\!)N \prec'\ (P \parallel Q'))$

and *rBrComm1*: $\bigwedge\Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ xvec\ Q'\ A_Q\ C.$
$\llbracket\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';$
$extractFrame\ P = \langle A_P,\ \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec Q';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ ((\!|\nu{*}xvec|\!)N \prec'\ Q');$
$extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ distinct\ A_Q;$
$A_P \sharp{*}\ \Psi;\ A_P \sharp{*}\ \Psi_Q;\ A_P \sharp{*}\ P;\ A_P \sharp{*}\ N;\ A_P \sharp{*}\ P';$
$A_P \sharp{*}\ Q;\ A_P \sharp{*}\ Q';\ A_P \sharp{*}\ A_Q;\ A_P \sharp{*}\ xvec;\ A_Q \sharp{*}\ \Psi;\ A_Q \sharp{*}\ \Psi_P;$
$A_Q \sharp{*}\ P;\ A_Q \sharp{*}\ N;\ A_Q \sharp{*}\ P';\ A_Q \sharp{*}\ Q;\ A_Q \sharp{*}\ Q';\ distinct\ xvec;$
$A_P \sharp{*}\ M;\ A_Q \sharp{*}\ M;\ xvec \sharp{*}\ M;$
$A_Q \sharp{*}\ xvec;\ xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ \Psi_P;\ xvec \sharp{*}\ \Psi_Q;\ xvec \sharp{*}\ P;$

$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ (\langle\!|\nu * xvec|\!\rangle N \prec' (P' \parallel Q'))$

and *rBrComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M\langle\!|\nu * xvec|\!\rangle\langle N\rangle \prec P'; \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ (\langle\!|\nu * xvec|\!\rangle N \prec' P')$;
$extractFrame\ P = \langle A_P, \Psi_P\rangle$; $distinct\ A_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M\langle\!|N|\!\rangle \prec Q'$;
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle$; $distinct\ A_Q$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* Q'$; $distinct\ xvec$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ (\langle\!|\nu * xvec|\!\rangle N \prec' (P' \parallel Q'))$

and *rBrOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto ¡M\langle\!|\nu * (xvec @ yvec)|\!\rangle\langle N\rangle \prec P'; x \in supp\ N; \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ (\langle\!|\nu * (xvec @ yvec)|\!\rangle N \prec' P')$;
$x \sharp \Psi$; $x \sharp M$; $x \sharp xvec$; $x \sharp yvec$; $xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$;
$xvec \sharp* yvec$; $yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $yvec \sharp* C$; $x \sharp C$;
$xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (\langle\!|\nu x|\!\rangle P)\ M\ (\langle\!|\nu * (xvec @ x \# yvec)|\!\rangle N \prec' P')$

and *rScope*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto ¡M\langle\!|\nu * xvec|\!\rangle\langle N\rangle \prec P'; \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ (\langle\!|\nu * xvec|\!\rangle N \prec' P')$;
$x \sharp \Psi$; $x \sharp M$; $x \sharp xvec$; $x \sharp N$; $xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$;
$x \sharp C$; $xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (\langle\!|\nu x|\!\rangle P)\ M\ (\langle\!|\nu * xvec|\!\rangle N \prec' \langle\!|\nu x|\!\rangle P')$

and *rBang*: $\bigwedge \Psi\ P\ M\ B\ C.$
$[\![\Psi \rhd P \parallel !P \longmapsto (RBrOut\ M\ B); guarded\ P; \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel !P)\ M\ B]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (!P)\ M\ B$

shows $Prop\ C\ \Psi\ P\ M\ B$
⟨*proof*⟩

lemma *boundOutputBindObject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and $yvec$ :: $name\ list$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and y :: $name$

assumes $\Psi \rhd P \longmapsto \alpha \prec P'$
and $bn\ \alpha \sharp* subject\ \alpha$
and $distinct(bn\ \alpha)$
and $y \in set(bn\ \alpha)$

shows $y \in supp(object\ \alpha)$
⟨*proof*⟩

lemma *alphaBoundOutputChain'*:
fixes *yvec* :: *name list*
and *xvec* :: *name list*
and B :: $('a, 'b, 'c)$ *boundOutput*

assumes *length xvec* = *length yvec*
and *yvec* $\sharp*$ B
and *yvec* $\sharp*$ *xvec*
and *distinct yvec*

shows $(\!|\nu*xvec|\!)B = (\!|\nu*yvec|\!)([xvec\ yvec] \cdot_v B)$
⟨*proof*⟩

lemma *alphaBoundOutputChain''*:
fixes *yvec* :: *name list*
and *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*

assumes *length xvec* = *length yvec*
and *yvec* $\sharp*$ N
and *yvec* $\sharp*$ P
and *yvec* $\sharp*$ *xvec*
and *distinct yvec*

shows $(\!|\nu*xvec|\!)(N \prec' P) = (\!|\nu*yvec|\!)(([xvec\ yvec] \cdot_v N) \prec' ([xvec\ yvec] \cdot_v P))$
⟨*proof*⟩

lemma *alphaDistinct*:
fixes *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and *yvec* :: *name list*
and M :: $'a$
and Q :: $('a, 'b, 'c)$ *psi*

assumes $\alpha \prec P = \beta \prec Q$
and *distinct*$(bn\ \alpha)$
and $\bigwedge x.\ x \in set(bn\ \alpha) \Longrightarrow x \in supp(object\ \alpha)$
and $bn\ \alpha\ \sharp*\ bn\ \beta$
and $bn\ \alpha\ \sharp*\ (object\ \beta)$
and $bn\ \alpha\ \sharp*\ Q$

shows *distinct*$(bn\ \beta)$
⟨*proof*⟩

lemma *boundOutputDistinct*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*

assumes $\Psi \rhd P \longmapsto \alpha \prec P'$

shows *distinct*$(bn\ \alpha)$
⟨*proof*⟩

lemma *inputDistinct*:
fixes Ψ :: $'b$
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and Rs :: $('a, 'b, 'c)$ *residual*

assumes $\Psi \rhd M(\!|\lambda * xvec\ N|\!).P \longmapsto Rs$

shows *distinct xvec*
⟨*proof*⟩

lemma *outputInduct'*[*consumes 2*, *case-names cAlpha cOutput cCase cPar1 cPar2 cOpen cScope cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *yvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and *Prop* :: $'f$::*fs-name* ⇒ $'b$ ⇒ $('a, 'b, 'c)$ *psi* ⇒ $'a$ ⇒ *name list* ⇒ $'a$ ⇒ $('a, 'b, 'c)$ *psi* ⇒ *bool*
and C :: $'f$::*fs-name*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *xvec* $\sharp*$ M
and *rAlpha*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ p\ C.$ ⟦*xvec* $\sharp*$ Ψ; *xvec* $\sharp*$ P; *xvec* $\sharp*$ M; *xvec* $\sharp*$ C; *xvec* $\sharp*$ $(p \cdot xvec)$; *set* $p \subseteq$ *set xvec* $\times$ *set*$(p \cdot xvec)$; *distinctPerm* p; $(p \cdot xvec)$ $\sharp*$ N; $(p \cdot xvec)$ $\sharp*$ P'; *Prop* C Ψ P M *xvec* N P'⟧ ⟹ *Prop* C Ψ P M $(p \cdot xvec)$ $(p \cdot N)$ $(p \cdot P')$
and *rOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.$ ⟦$\Psi \vdash M \leftrightarrow K$⟧ ⟹ *Prop* C Ψ $(M\langle N\rangle.P)$ K $([])$ N P
and *rCase*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ \varphi\ Cs\ C.$ ⟦$\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$; $\bigwedge C.$ *Prop* C Ψ P M *xvec* N P'; $(\varphi, P) \in$ *set Cs*; $\Psi \vdash \varphi$; *guarded* P⟧ ⟹ *Prop* C Ψ $(Cases\ Cs)$ M *xvec* N P'

and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ \ P'\ A_Q\ Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ xvec\ N\ P';$
$A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* M;$
$A_Q \sharp* xvec;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* C;\ xvec \sharp* Q;$
$xvec \sharp* \Psi;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ xvec\ N\ (P' \parallel Q)$
and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ xvec\ N\ \ Q'\ A_P\ P\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ \ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ xvec\ N\ Q';$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;\ A_P \sharp* M;$
$A_P \sharp* xvec;\ A_P \sharp* N;\ A_P \sharp* Q';\ A_P \sharp* C;\ xvec \sharp* Q;$
$xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ xvec\ N\ (P \parallel Q')$
and *rOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|\nu * (xvec@yvec)|\!)\langle N\rangle \prec P';\ x \in supp\ N;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ (xvec@yvec)\ N\ P';$
$x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;\ x \sharp yvec;\ xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;$
$yvec \sharp* \Psi;\ yvec \sharp* P;\ yvec \sharp* M;\ yvec \sharp* C;\ x \sharp C;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ M\ (xvec@x\#yvec)\ N\ P'$
and *rScope*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ xvec\ N\ P';$
$x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;\ x \sharp N;\ xvec \sharp* \Psi;$
$xvec \sharp* P;\ xvec \sharp* M;\ x \sharp C;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ M\ xvec\ N\ ((\!|\nu x|\!)P')$
and *rBang*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ C.$
$[\![\Psi \rhd P \parallel\ !P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ guarded\ P;\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel\ !P)\ M\ xvec\ N\ P']\!] \Longrightarrow$
$Prop\ C\ \Psi\ (!P)\ M\ xvec\ N\ P'$
shows $Prop\ C\ \Psi\ P\ M\ xvec\ N\ P'$
⟨*proof*⟩

lemma *brOutputInduct'*[*consumes 2*, *case-names cAlpha cBrOutput cCase cPar1 cPar2 cBrComm1 cBrComm2 cBrOpen cScope cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and $yvec$ :: $name\ list$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and $Prop$ :: $'f::fs\text{-}name \Rightarrow 'b \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow 'a \Rightarrow name\ list \Rightarrow 'a \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow bool$
and C :: $'f::fs\text{-}name$

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and *rAlpha*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ p\ C.\ [\![xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;\ \ xvec$

$\sharp*\ C$; $xvec\ \sharp*\ (p \cdot xvec)$;
$set\ p \subseteq set\ xvec \times set(p \cdot xvec)$; $distinctPerm\ p$;
$(p \cdot xvec)\ \sharp*\ N$; $(p \cdot xvec)\ \sharp*\ P'$; $Prop\ C\ \Psi\ P\ M\ xvec\ N\ P'\rrbracket \Longrightarrow$
$Prop\ C\ \Psi\ P\ M\ (p \cdot xvec)\ (p \cdot N)\ (p \cdot P')$

and *rBrOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ [\![\Psi \vdash M \preceq K]\!] \Longrightarrow Prop\ C\ \Psi\ (M\langle N\rangle.P)\ K\ ([])\ N\ P$

and *rCase*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ \varphi\ Cs\ C.\ [\![\Psi \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ xvec\ N\ P';\ (\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (Cases\ Cs)\ M\ xvec\ N\ P'$

and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ \ P'\ A_Q\ Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ xvec\ N\ P';$
$A_Q\ \sharp*\ P;\ A_Q\ \sharp*\ Q;\ A_Q\ \sharp*\ \Psi;\ A_Q\ \sharp*\ M;$
$A_Q\ \sharp*\ xvec;\ A_Q\ \sharp*\ N;\ A_Q\ \sharp*\ P';\ A_Q\ \sharp*\ C;\ xvec\ \sharp*\ Q;$
$xvec\ \sharp*\ \Psi;\ xvec\ \sharp*\ \Psi_Q;\ xvec\ \sharp*\ P;\ xvec\ \sharp*\ M;\ xvec\ \sharp*\ C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ xvec\ N\ (P' \parallel Q)$

and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ xvec\ N\ \ Q'\ A_P\ P\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ xvec\ N\ Q';$
$A_P\ \sharp*\ P;\ A_P\ \sharp*\ Q;\ A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ M;$
$A_P\ \sharp*\ xvec;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ Q';\ A_P\ \sharp*\ C;\ xvec\ \sharp*\ Q;$
$xvec\ \sharp*\ \Psi;\ xvec\ \sharp*\ \Psi_P;\ xvec\ \sharp*\ P;\ xvec\ \sharp*\ M;\ xvec\ \sharp*\ C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ xvec\ N\ (P \parallel Q')$

and *rBrComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ xvec\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec Q';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ xvec\ N\ Q';$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ \Psi_Q;\ A_P\ \sharp*\ P;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ P';$
$A_P\ \sharp*\ Q;\ A_P\ \sharp*\ Q';\ A_P\ \sharp*\ A_Q;\ A_P\ \sharp*\ xvec;\ A_Q\ \sharp*\ \Psi;\ A_Q\ \sharp*\ \Psi_P;$
$A_Q\ \sharp*\ P;\ A_Q\ \sharp*\ N;\ A_Q\ \sharp*\ P';\ A_Q\ \sharp*\ Q;\ A_Q\ \sharp*\ Q';\ distinct\ xvec;$
$A_P\ \sharp*\ M;\ A_Q\ \sharp*\ M;\ xvec\ \sharp*\ M;$
$A_Q\ \sharp*\ xvec;\ xvec\ \sharp*\ \Psi;\ xvec\ \sharp*\ \Psi_P;\ xvec\ \sharp*\ \Psi_Q;\ xvec\ \sharp*\ P;$
$xvec\ \sharp*\ Q;\ A_P\ \sharp*\ C;\ A_Q\ \sharp*\ C;\ xvec\ \sharp*\ C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ xvec\ N\ (P' \parallel Q')$

and *rBrComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ xvec\ N\ P';$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ \Psi_Q;\ A_P\ \sharp*\ P;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ P';$
$A_P\ \sharp*\ Q;\ A_P\ \sharp*\ Q';\ A_P\ \sharp*\ A_Q;\ A_P\ \sharp*\ xvec;\ A_Q\ \sharp*\ \Psi;\ A_Q\ \sharp*\ \Psi_P;$
$A_Q\ \sharp*\ P;\ A_Q\ \sharp*\ N;\ A_Q\ \sharp*\ P';\ A_Q\ \sharp*\ Q;\ A_Q\ \sharp*\ Q';\ distinct\ xvec;$
$A_P\ \sharp*\ M;\ A_Q\ \sharp*\ M;\ xvec\ \sharp*\ M;$

$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (P \parallel Q)\ M\ xvec\ N\ (P' \parallel Q')$

and *rOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ C$.
$[\![\Psi \rhd P \longmapsto\!\!¡ M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'$; $x \in supp\ N$; $\bigwedge C$. *Prop* $C\ \Psi\ P\ M\ (xvec@yvec)\ N\ P'$;
$x \sharp \Psi$; $x \sharp M$; $x \sharp xvec$; $x \sharp yvec$; $xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$;
$yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $yvec \sharp* C$; $x \sharp C$; $xvec \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ M\ (xvec@x\#yvec)\ N\ P'$

and *rScope*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ x\ C$.
$[\![\Psi \rhd P \longmapsto\!\!¡ M(\!|\nu*xvec|\!)\langle N\rangle \prec P'$; $\bigwedge C$. *Prop* $C\ \Psi\ P\ M\ xvec\ N\ P'$;
$x \sharp \Psi$; $x \sharp M$; $x \sharp xvec$; $x \sharp N$; $xvec \sharp* \Psi$;
$xvec \sharp* P$; $xvec \sharp* M$; $x \sharp C$; $xvec \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ M\ xvec\ N\ ((\!|\nu x|\!)P')$

and *rBang*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ C$.
$[\![\Psi \rhd P \parallel\ !P \longmapsto\!\!¡ M(\!|\nu*xvec|\!)\langle N\rangle \prec P'$; *guarded* P; $\bigwedge C$. *Prop* $C\ \Psi\ (P \parallel\ !P)\ M\ xvec\ N\ P']\!] \Longrightarrow$
Prop $C\ \Psi\ (!P)\ M\ xvec\ N\ P'$

shows *Prop* $C\ \Psi\ P\ M\ xvec\ N\ P'$
⟨*proof*⟩

lemma *inputInduct*[*consumes 1*, *case-names cInput cCase cPar1 cPar2 cScope cBang*]:
fixes Ψ :: *'b*
and P :: *('a, 'b, 'c) psi*
and M :: *'a*
and N :: *'a*
and P' :: *('a, 'b, 'c) psi*
and *Prop* :: *'f::fs-name* ⇒ *'b* ⇒ *('a, 'b, 'c) psi* ⇒ *'a* ⇒ *'a* ⇒ *('a, 'b, 'c) psi* ⇒ *bool*
and C :: *'f::fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and *rInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C$.
$[\![\Psi \vdash M \leftrightarrow K$; *distinct xvec*; *set xvec* $\subseteq$ *supp N*;
length xvec = *length Tvec*; $xvec \sharp* \Psi$;
$xvec \sharp* M$; $xvec \sharp* K$; $xvec \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (M(\!|\lambda*xvec\ N|\!).P)$
$K\ (N[xvec::=Tvec])\ (P[xvec::=Tvec])$

and *rCase*: $\bigwedge \Psi\ P\ M\ N\ P'\ \varphi\ Cs\ C$. $[\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$; $\bigwedge C$. *Prop* $C\ \Psi\ P\ M\ N\ P'$; $(\varphi, P) \in set\ Cs$; $\Psi \vdash \varphi$; *guarded* $P]\!] \Longrightarrow$
Prop $C\ \Psi\ (Cases\ Cs)\ M\ N\ P'$

and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_Q\ Q\ C$.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\bigwedge C$. *Prop* $C\ (\Psi \otimes \Psi_Q)\ P\ M\ N\ P'$; *distinct* A_Q;
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* M$; $A_Q \sharp* N$;
$A_Q \sharp* P'$; $A_Q \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (P \parallel Q)\ M\ N\ (P' \parallel Q)$

and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ N\ Q'\ A_P\ P\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|N|\!) \prec Q';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;$
$distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ N\ Q';\ distinct\ A_P;$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;\ A_P \sharp* M;\ A_P \sharp* N;$
$A_P \sharp* Q';\ A_P \sharp* C] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ N\ (P \parallel Q')$
and *rScope*: $\bigwedge \Psi\ P\ M\ N\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ N\ P';\ x \sharp \Psi;\ x \sharp$
$M;\ x \sharp N;\ x \sharp C] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ M\ N\ ((\!|\nu x|\!)P')$
and *rBang*: $\bigwedge \Psi\ P\ M\ N\ P'\ C.$
$[\![\Psi \rhd P \parallel !P \longmapsto M(\!|N|\!) \prec P';\ guarded\ P;\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel$
$!P)\ M\ N\ P']\!] \Longrightarrow Prop\ C\ \Psi\ (!P)\ M\ N\ P'$
shows $Prop\ C\ \Psi\ P\ M\ N\ P'$
⟨*proof*⟩

lemma *brInputInduct*[*consumes 1*, *case-names cBrInput cCase cPar1 cPar2 cBrMerge cScope cBang*]:
fixes Ψ :: *'b*
and P :: *('a, 'b, 'c) psi*
and M :: *'a*
and N :: *'a*
and P' :: *('a, 'b, 'c) psi*
and *Prop* :: *'f::fs-name* ⇒ *'b* ⇒ *('a, 'b, 'c) psi* ⇒
'a ⇒ *'a* ⇒ *('a, 'b, 'c) psi* ⇒ *bool*
and C :: *'f::fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and *rBrInput*: $\bigwedge \Psi\ K\ M\ xvec\ N\ Tvec\ P\ C.$
$[\![\Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec \sharp* \Psi;$
$xvec \sharp* M;\ xvec \sharp* K;\ xvec \sharp* C] \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda* xvec\ N|\!).P)$
$K\ (N[xvec::=Tvec])\ (P[xvec::=Tvec])$
and *rCase*: $\bigwedge \Psi\ P\ M\ N\ P'\ \varphi\ Cs\ C.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \bigwedge C.\ Prop\ C\ \Psi$
$P\ M\ N\ P';\ (\varphi,\ P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (Cases\ Cs)\ M\ N\ P'$
and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_Q\ Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;$
$distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ N\ P';\ distinct\ A_Q;$
$A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* M;\ A_Q \sharp* N;$
$A_Q \sharp* P';\ A_Q \sharp* C] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ N\ (P' \parallel Q)$
and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ N\ Q'\ A_P\ P\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;$
$distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ N\ Q';\ distinct\ A_P;$

$A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \Psi$; $A_P \sharp* M$; $A_P \sharp* N$;
$A_P \sharp* Q'$; $A_P \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ M N $(P \parallel Q')$

and *rBrMerge*: $\bigwedge \Psi$ Ψ_Q P M N P' A_P Ψ_P Q Q' A_Q C.
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_Q)$ P M N P';
extractFrame $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q M N Q';
extractFrame $Q = \langle A_Q, \Psi_Q \rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* M$; $A_Q \sharp* M$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$;
$A_Q \sharp* Q$; $A_Q \sharp* Q'$; $A_P \sharp* C$; $A_Q \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ M N $(P' \parallel Q')$

and *rScope*: $\bigwedge \Psi$ P M N P' x C.
$\llbracket \Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; $\bigwedge C$. *Prop* C Ψ P M N P'; $x \sharp \Psi$; $x \sharp M$; $x \sharp N$; $x \sharp C$$\rrbracket \Longrightarrow$
Prop C Ψ $((\!|\nu x|\!)P)$ M N $((\!|\nu x|\!)P')$

and *rBang*: $\bigwedge \Psi$ P M N P' C.
$\llbracket \Psi \rhd P \parallel !P \longmapsto ¿M(\!|N|\!) \prec P'$; *guarded* P; $\bigwedge C$. *Prop* C Ψ $(P \parallel !P)$ M N $P$$\rrbracket \Longrightarrow$ *Prop* C Ψ $(!P)$ M N P'

shows *Prop* C Ψ P M N P'

⟨*proof*⟩

lemma *tauInduct*[*consumes 1*, *case-names cCase cPar1 cPar2 cComm1 cComm2 cBrClose cScope cBang*]:

fixes Ψ :: *'b*
and P :: *('a, 'b, 'c) psi*
and Rs :: *('a, 'b, 'c) residual*
and *Prop* :: *'f::fs-name* ⇒ *'b* ⇒ *('a, 'b, 'c) psi* ⇒ *('a, 'b, 'c) psi* ⇒ *bool*
and C :: *'f::fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto \tau \prec P'$

and *rCase*: $\bigwedge \Psi$ P P' φ *Cs* C. $\llbracket \Psi \rhd P \longmapsto \tau \prec P'$; $\bigwedge C$. *Prop* C Ψ P P'; $(\varphi, P) \in$ *set Cs*; $\Psi \vdash \varphi$; *guarded* $P$$\rrbracket \Longrightarrow$
Prop C Ψ (*Cases Cs*) P'

and *rPar1*: $\bigwedge \Psi$ Ψ_Q P P' A_Q Q C.
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto \tau \prec P'$; *extractFrame* $Q = \langle A_Q, \Psi_Q \rangle$; *distinct* A_Q;
$\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_Q)$ P P';
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$;
$A_Q \sharp* P'$; $A_Q \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ $(P' \parallel Q)$

and *rPar2*: $\bigwedge \Psi$ Ψ_P Q Q' A_P P C.
$\llbracket \Psi \otimes \Psi_P \rhd Q \longmapsto \tau \prec Q'$; *extractFrame* $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;

$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ Q';$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;$
$A_P \sharp* Q';\ A_P \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (P \parallel Q')$

and *rComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ K\ xvec\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!(N)\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;$
$distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!(\nu*xvec)\!)\langle N\rangle \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* xvec;\ A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;$
$A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* Q;\ A_Q \sharp* K;\ A_Q \sharp* Q';$
$A_Q \sharp* xvec;\ xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;\ xvec \sharp* M;$
$xvec \sharp* Q;\ xvec \sharp* K;\ A_P \sharp* C;\ A_Q \sharp* C;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ ((\!(\nu*xvec)\!)(P' \parallel Q'))$

and *rComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ K\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!(\nu*xvec)\!)\langle N\rangle \prec P';\ \ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!(N)\!) \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* xvec;\ A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;$
$A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* Q;\ A_Q \sharp* K;\ A_Q \sharp* Q';$
$A_Q \sharp* xvec;\ xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;\ xvec \sharp* M;$
$xvec \sharp* Q;\ xvec \sharp* K;\ A_P \sharp* C;\ A_Q \sharp* C;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ ((\!(\nu*xvec)\!)(P' \parallel Q'))$

and *rBrClose*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ x\ C.$
$[\![\Psi \rhd P \longmapsto ¡M(\!(\nu*xvec)\!)\langle N\rangle \prec P';$
$x \in supp\ M;$
$distinct\ xvec;\ xvec \sharp* \Psi;\ xvec \sharp* P;$
$xvec \sharp* M;$
$x \sharp \Psi;\ x \sharp xvec]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!(\nu x)\!)P)\ ((\!(\nu x)\!)((\!(\nu*xvec)\!)P'))$

and *rScope*: $\bigwedge \Psi\ P\ P'\ x\ C.$
$[\![\Psi \rhd P \longmapsto \tau \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ P\ P';\ x \sharp \Psi;\ x \sharp C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!(\nu x)\!)P)\ ((\!(\nu x)\!)P')$

and *rBang*: $\bigwedge \Psi\ P\ P'\ C.$
$[\![\Psi \rhd P \parallel !P \longmapsto \tau \prec P';\ guarded\ P;\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel !P)\ P']\!]$
$\Longrightarrow Prop\ C\ \Psi\ (!P)\ P'$

shows $Prop\ C\ \Psi\ P\ P'$

⟨*proof*⟩

lemma *semanticsFrameInduct*[*consumes 3*, *case-names cAlpha cInput cBrInput cOutput cBrOutput cCase cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2 cBrClose cOpen cBrOpen cScope cBang*]:

fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Rs :: $('a, 'b, 'c)$ *residual*
and A_P :: *name list*
and Ψ_P :: $'b$
and *Prop* :: $'f$::*fs-name* $\Rightarrow$ $'b \Rightarrow ('a, 'b, 'c)$ *psi* $\Rightarrow$
$('a, 'b, 'c)$ *residual* $\Rightarrow$ *name list* $\Rightarrow$ $'b \Rightarrow$ *bool*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto Rs$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and *distinct* A_P
and *rAlpha*: $\bigwedge \Psi\ P\ A_P\ \Psi_P\ p\ Rs\ C.\ [\![A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* (p \cdot A_P);\ A_P \sharp* Rs;\ A_P \sharp* C;$
$set\ p \subseteq set\ A_P \times set(p \cdot A_P);\ distinctPerm\ p;$
$Prop\ C\ \Psi\ P\ Rs\ A_P\ \Psi_P]\!] \Longrightarrow Prop\ C\ \Psi\ P\ Rs\ (p \cdot A_P)\ (p \cdot \Psi_P)$
and *rInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C.$
$[\![\Psi \vdash M \leftrightarrow K;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec \sharp* \Psi;$
$xvec \sharp* M;\ xvec \sharp* K;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda* xvec\ N|\!).P)$
$(K(\!|(N[xvec::=Tvec])|\!) \prec (P[xvec::=Tvec]))\ ([])\ (\mathbf{1})$
and *rBrInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C.$
$[\![\Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec \sharp* \Psi;$
$xvec \sharp* M;\ xvec \sharp* K;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda* xvec\ N|\!).P)$
$(¡K(\!|(N[xvec::=Tvec])|\!) \prec (P[xvec::=Tvec]))\ ([])\ (\mathbf{1})$
and *rOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ \Psi \vdash M \leftrightarrow K \Longrightarrow Prop\ C\ \Psi\ (M\langle N \rangle.P)\ (K\langle N \rangle \prec P)\ ([])\ (\mathbf{1})$
and *rBrOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ \Psi \vdash M \preceq K \Longrightarrow Prop\ C\ \Psi\ (M\langle N \rangle.P)\ (¡K\langle N \rangle \prec P)\ ([])\ (\mathbf{1})$
and *rCase*: $\bigwedge \Psi\ P\ Rs\ \varphi\ Cs\ A_P\ \Psi_P\ C.\ [\![\Psi \rhd P \longmapsto Rs;\ extractFrame\ P = \langle A_P, \Psi_P \rangle;\ distinct\ A_P;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ Rs\ A_P\ \Psi_P;$
$(\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P;\ \Psi_P \simeq \mathbf{1};$
$(supp\ \Psi_P) = (\{\}::name\ set);$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Rs;\ A_P \sharp* C]\!] \Longrightarrow Prop\ C\ \Psi\ (Cases\ Cs)\ Rs\ ([])\ (\mathbf{1})$
and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ \alpha\ P'\ A_Q\ Q\ A_P\ \Psi_P\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P';$
$extractFrame\ P = \langle A_P, \Psi_P \rangle;\ distinct\ A_P;$
$extractFrame\ Q = \langle A_Q, \Psi_Q \rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ (\alpha \prec P')\ A_P\ \Psi_P;\ distinct(bn\ \alpha);$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;\ A_P \sharp* \alpha;\ A_P \sharp* P';\ A_P \sharp* A_Q;\ A_P \sharp* \Psi_Q;$
$A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* \alpha;\ A_Q \sharp* P';\ A_Q \sharp* \Psi_P;$
$bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* Q;\ bn\ \alpha \sharp* subject\ \alpha;\ bn\ \alpha \sharp* \Psi_P;$
$bn\ \alpha \sharp* \Psi_Q;$

$A_P \sharp* C$; $A_Q \sharp* C$; $bn\ \alpha \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (\alpha \prec (P' \parallel Q))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ \alpha\ Q'\ A_P\ P\ A_Q\ \Psi_Q\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q'$; $extractFrame\ P = \langle A_P, \Psi_P \rangle$; $distinct\ A_P$;
$extractFrame\ Q = \langle A_Q, \Psi_Q \rangle$; $distinct\ A_Q$;
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ (\alpha \prec Q')\ A_Q\ \Psi_Q$; $distinct(bn\ \alpha)$;
$A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \Psi$; $A_P \sharp* \alpha$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* \Psi_Q$;
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* \alpha$; $A_Q \sharp* Q'$; $A_Q \sharp* \Psi_P$;
$bn\ \alpha \sharp* \Psi$; $bn\ \alpha \sharp* P$; $bn\ \alpha \sharp* Q$; $bn\ \alpha \sharp* subject\ \alpha$; $bn\ \alpha \sharp* \Psi_P$; $bn\ \alpha \sharp* \Psi_Q$;
$A_P \sharp* C$; $A_Q \sharp* C$; $bn\ \alpha \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (\alpha \prec (P \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ K\ xvec\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P'$; $extractFrame\ P = \langle A_P, \Psi_P \rangle$; $distinct\ A_P$;
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ ((M(\!|N|\!)) \prec P')\ A_P\ \Psi_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu * xvec|\!)\langle N \rangle \prec Q'$; $extractFrame\ Q = \langle A_Q, \Psi_Q \rangle$; $distinct\ A_Q$;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$;
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ (K(\!|\nu * xvec|\!)\langle N \rangle \prec Q')\ A_Q\ \Psi_Q$; $distinct\ xvec$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* K$; $A_Q \sharp* Q'$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$; $xvec \sharp* M$;
$xvec \sharp* Q$; $xvec \sharp* K$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (\tau \prec (\!|\nu * xvec|\!)(P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ K\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N \rangle \prec P'$; $extractFrame\ P = \langle A_P, \Psi_P \rangle$; $distinct\ A_P$;
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ (M(\!|\nu * xvec|\!)\langle N \rangle \prec P')\ A_P\ \Psi_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q'$; $extractFrame\ Q = \langle A_Q, \Psi_Q \rangle$; $distinct\ A_Q$;
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ (K(\!|N|\!) \prec Q')\ A_Q\ \Psi_Q$;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; $distinct\ xvec$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* K$; $A_Q \sharp* Q'$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$; $xvec \sharp* M$;
$xvec \sharp* Q$; $xvec \sharp* K$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (\tau \prec (\!|\nu * xvec|\!)(P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrMerge*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; $\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ (¿M(\!|N|\!) \prec P')\ A_P\ \Psi_P$;
$extractFrame\ P = \langle A_P, \Psi_P \rangle$; $distinct\ A_P$;

$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; $\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q $(¿M(\!|N|\!) \prec Q')$ A_Q Ψ_Q;
extractFrame $Q = \langle A_Q, \Psi_Q \rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* M$; $A_Q \sharp* M$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$;
$A_Q \sharp* Q$; $A_Q \sharp* Q'$; $A_P \sharp* C$; $A_Q \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ $(¿M(\!|N|\!) \prec (P' \parallel Q'))$ $(A_P@A_Q)$ $(\Psi_P \otimes \Psi_Q)$

and *rBrComm1*:$\bigwedge \Psi$ Ψ_Q P M N P' A_P Ψ_P Q *xvec* Q' A_Q C.
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_Q)$ P $((¿M(\!|N|\!)) \prec P')$ A_P Ψ_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N \rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q \rangle$; *distinct* A_Q;
$\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q $(¡M(\!|\nu*xvec|\!)\langle N \rangle \prec Q')$ A_Q Ψ_Q; *distinct* *xvec*;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* Q'$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ $(¡M(\!|\nu*xvec|\!)\langle N \rangle \prec (P' \parallel Q'))$ $(A_P@A_Q)$ $(\Psi_P \otimes \Psi_Q)$

and *rBrComm2*:$\bigwedge \Psi$ Ψ_Q P M *xvec* N P' A_P Ψ_P Q Q' A_Q C.
$\llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N \rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_Q)$ P $(¡M(\!|\nu*xvec|\!)\langle N \rangle \prec P')$ A_P Ψ_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q \rangle$; *distinct* A_Q;
$\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q $(¿M(\!|N|\!) \prec Q')$ A_Q Ψ_Q; *distinct* *xvec*;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* Q'$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ $(¡M(\!|\nu*xvec|\!)\langle N \rangle \prec (P' \parallel Q'))$ $(A_P@A_Q)$ $(\Psi_P \otimes \Psi_Q)$

and *rBrClose*: $\bigwedge \Psi$ P M *xvec* N P' A_P Ψ_P x C.
$\llbracket \Psi \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N \rangle \prec P'$;
$x \in supp\ M$;
$\bigwedge C$. *Prop* C Ψ P $(¡M(\!|\nu*xvec|\!)\langle N \rangle \prec P')$ A_P Ψ_P;
extractFrame $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* xvec$;
distinct *xvec*; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* P$;
$xvec \sharp* M$;
$x \sharp \Psi$; $x \sharp xvec$; $x \sharp A_P$;

$A_P \sharp* C$; $xvec \sharp* C$; $x \sharp C$]] $\Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ (\tau \prec ((\!|\nu x|\!)((\!|\nu* xvec|\!)P')))\ (x\#A_P)\ \Psi_P$

and *rOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ A_P\ \Psi_P\ C$.
[[$\Psi \rhd P \longmapsto M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* $C\ \Psi\ P\ (M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P')\ A_P\ \Psi_P$; $x \in supp\ N$; $x \sharp \Psi$; $x \sharp M$;
$x \sharp A_P$; $x \sharp xvec$; $x \sharp yvec$; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* xvec$; $A_P \sharp* yvec$; $xvec \sharp* yvec$; *distinct* $xvec$; *distinct* $yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$; $yvec \sharp* \Psi_P$;
$yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $A_P \sharp* C$; $x \sharp C$; $xvec \sharp* C$; $yvec \sharp* C$]] $\Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ (M(\!|\nu*(xvec@x\#yvec)|\!)\langle N\rangle \prec P')\ (x\#A_P)\ \Psi_P$

and *rBrOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ A_P\ \Psi_P\ C$.
[[$\Psi \rhd P \longmapsto ¡M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* $C\ \Psi\ P\ (¡M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P')\ A_P\ \Psi_P$; $x \in supp\ N$; $x \sharp \Psi$; $x \sharp M$;
$x \sharp A_P$; $x \sharp xvec$; $x \sharp yvec$; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* xvec$; $A_P \sharp* yvec$; $xvec \sharp* yvec$; *distinct* $xvec$; *distinct* $yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$; $yvec \sharp* \Psi_P$;
$yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $A_P \sharp* C$; $x \sharp C$; $xvec \sharp* C$; $yvec \sharp* C$]] $\Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ (¡M(\!|\nu*(xvec@x\#yvec)|\!)\langle N\rangle \prec P')\ (x\#A_P)\ \Psi_P$

and *rScope*: $\bigwedge \Psi\ P\ \alpha\ P'\ x\ A_P\ \Psi_P\ C$.
[[$\Psi \rhd P \longmapsto \alpha \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* $C\ \Psi\ P\ (\alpha \prec P')\ A_P\ \Psi_P$;
$x \sharp \Psi$; $x \sharp \alpha$; $x \sharp A_P$; $A_P \sharp* \Psi$; $A_P \sharp* P$;
$A_P \sharp* \alpha$; $A_P \sharp* P'$; *distinct*$(bn\ \alpha)$;
$bn\ \alpha \sharp* \Psi$; $bn\ \alpha \sharp* P$; $bn\ \alpha \sharp* subject\ \alpha$; $bn\ \alpha \sharp* \Psi_P$;
$A_P \sharp* C$; $x \sharp C$; $bn\ \alpha \sharp* C$]] $\Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ (\alpha \prec ((\!|\nu x|\!)P'))\ (x\#A_P)\ \Psi_P$

and *rBang*: $\bigwedge \Psi\ P\ Rs\ A_P\ \Psi_P\ C$.
[[$\Psi \rhd P \parallel\ !P \longmapsto Rs$; *guarded* P; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* $C\ \Psi\ (P \parallel\ !P)\ Rs\ A_P\ (\Psi_P \otimes \mathbf{1})$; $\Psi_P \simeq \mathbf{1}$; $supp\ \Psi_P = (\{\}::name\ set)$;
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* Rs$; $A_P \sharp* C$]] $\Longrightarrow$ *Prop* $C\ \Psi\ (!P)\ Rs\ ([])$ (**1**)

shows *Prop* $C\ \Psi\ P\ Rs\ A_P\ \Psi_P$
⟨*proof*⟩

lemma *semanticsFrameInduct'*[*consumes 5*, *case-names cAlpha cFrameAlpha cInput cBrInput cOutput cBrOutput cCase cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2 cBrClose cOpen cBrOpen cScope cBang*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*

and Rs :: $('a, 'b, 'c)$ *residual*
and A_P :: *name list*
and Ψ_P :: $'b$
and *Prop* :: $'f$::*fs-name* $\Rightarrow$ $'b \Rightarrow ('a, 'b, 'c)$ *psi* $\Rightarrow$ $'a$ *action* $\Rightarrow$
$('a, 'b, 'c)$ *psi* $\Rightarrow$ *name list* $\Rightarrow$ $'b \Rightarrow$ *bool*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto \alpha \prec P'$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and *distinct* A_P
and $bn\ \alpha \sharp* subject\ \alpha$
and *distinct*$(bn\ \alpha)$
and *rAlpha*: $\bigwedge \Psi\ P\ \alpha\ P'\ p\ A_P\ \Psi_P\ C.$ $[\![bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* subject\ \alpha;\ bn\ \alpha \sharp* \Psi_P;$
$bn\ \alpha \sharp* C;\ bn\ \alpha \sharp* (p \cdot \alpha);\ A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* \alpha;\ A_P \sharp* P';\ A_P \sharp* C;$
$set\ p \subseteq set(bn\ \alpha) \times set(bn(p \cdot \alpha));\ distinctPerm\ p;$
$bn(p \cdot \alpha) \sharp* \alpha;\ (bn(p \cdot \alpha)) \sharp* P';\ Prop\ C\ \Psi\ P\ \alpha\ P'\ A_P\ \Psi_P]\!] \Longrightarrow$
$Prop\ C\ \Psi\ P\ (p \cdot \alpha)\ (p \cdot P')\ A_P\ \Psi_P$
and *rFrameAlpha*: $\bigwedge \Psi\ P\ A_P\ \Psi_P\ p\ \alpha\ P'\ C.$ $[\![A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* (p \cdot A_P);\ A_P \sharp* \alpha;\ A_P \sharp* P';\ A_P \sharp* C;$
$set\ p \subseteq set\ A_P \times set(p \cdot A_P);\ distinctPerm\ p;\ A_P \sharp* subject\ \alpha;$
$Prop\ C\ \Psi\ P\ \alpha\ P'\ A_P\ \Psi_P]\!] \Longrightarrow Prop\ C\ \Psi\ P\ \alpha\ P'\ (p \cdot A_P)\ (p \cdot \Psi_P)$
and *rInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C.$
$[\![\Psi \vdash M \leftrightarrow K;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec \sharp* \Psi;$
$xvec \sharp* M;\ xvec \sharp* K;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda* xvec\ N|\!).P)$
$(K(\!|(N[xvec::=Tvec])|\!))\ (P[xvec::=Tvec])\ ([])\ (\mathbf{1})$
and *rBrInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C.$
$[\![\Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec \sharp* \Psi;$
$xvec \sharp* M;\ xvec \sharp* K;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda* xvec\ N|\!).P)$
$(¡K(\!|(N[xvec::=Tvec])|\!))\ (P[xvec::=Tvec])\ ([])\ (\mathbf{1})$
and *rOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ \Psi \vdash M \leftrightarrow K \Longrightarrow Prop\ C\ \Psi\ (M\langle N \rangle.P)\ (K\langle N \rangle)\ P\ ([])\ (\mathbf{1})$
and *rBrOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ \Psi \vdash M \preceq K \Longrightarrow Prop\ C\ \Psi\ (M\langle N \rangle.P)\ (¡K\langle N \rangle)\ P\ ([])\ (\mathbf{1})$
and *rCase*: $\bigwedge \Psi\ P\ \alpha\ P'\ \varphi\ Cs\ A_P\ \Psi_P\ C.$ $[\![\Psi \rhd P \longmapsto \alpha \prec P';\ extractFrame\ P = \langle A_P, \Psi_P \rangle;\ distinct\ A_P;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ \alpha\ P'\ A_P\ \Psi_P;$
$(\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P;\ \Psi_P \simeq \mathbf{1};\ (supp\ \Psi_P) = (\{\}::name\ set);$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* \alpha;\ A_P \sharp* P';\ A_P \sharp* C]\!] \Longrightarrow Prop\ C\ \Psi\ (Cases\ Cs)\ \alpha\ P'\ ([])\ (\mathbf{1})$

and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ \alpha\ P'\ A_Q\ Q\ A_P\ \Psi_P\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P'$;
extractFrame $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_Q)\ P\ \alpha\ P'\ A_P\ \Psi_P$;
$A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \Psi$; $A_P \sharp* \alpha$; $A_P \sharp* P'$; $A_P \sharp* A_Q$; $A_P \sharp* \Psi_Q$;
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* \alpha$; $A_Q \sharp* P'$; $A_Q \sharp* \Psi_P$;
$bn\ \alpha \sharp* \Psi$; $bn\ \alpha \sharp* P$; $bn\ \alpha \sharp* Q$; $bn\ \alpha \sharp*$ *subject* α; $bn\ \alpha \sharp* \Psi_P$; $bn\ \alpha \sharp* \Psi_Q$;
$A_P \sharp* C$; $A_Q \sharp* C$; $bn\ \alpha \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (P \parallel Q)\ \alpha\ (P' \parallel Q)\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ \alpha\ Q'\ A_P\ P\ A_Q\ \Psi_Q\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q'$;
extractFrame $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_P)\ Q\ \alpha\ Q'\ A_Q\ \Psi_Q$;
$A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \Psi$; $A_P \sharp* \alpha$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* \Psi_Q$;
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* \alpha$; $A_Q \sharp* Q'$; $A_Q \sharp* \Psi_P$;
$bn\ \alpha \sharp* \Psi$; $bn\ \alpha \sharp* P$; $bn\ \alpha \sharp* Q$; $bn\ \alpha \sharp*$ *subject* α; $bn\ \alpha \sharp* \Psi_P$; $bn\ \alpha \sharp* \Psi_Q$;
$A_P \sharp* C$; $A_Q \sharp* C$; $bn\ \alpha \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (P \parallel Q)\ \alpha\ (P \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ K\ xvec\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_Q)\ P\ (M(\!|N|\!))\ P'\ A_P\ \Psi_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu * xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; *distinct xvec*;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_P)\ Q\ (K(\!|\nu * xvec|\!)\langle N\rangle)\ Q'\ A_Q\ \Psi_Q$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* K$; $A_Q \sharp* Q'$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$; $xvec \sharp* M$;
$xvec \sharp* Q$; $xvec \sharp* K$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (P \parallel Q)\ (\tau)\ ((\!|\nu * xvec|\!)(P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ K\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_Q)\ P\ (M(\!|\nu * xvec|\!)\langle N\rangle)\ P'\ A_P\ \Psi_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_P)\ Q\ (K(\!|N|\!))\ Q'\ A_Q\ \Psi_Q$;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; *distinct xvec*;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;

$A_Q \sharp\!* P$; $A_Q \sharp\!* N$; $A_Q \sharp\!* P'$; $A_Q \sharp\!* Q$; $A_Q \sharp\!* K$; $A_Q \sharp\!* Q'$;
$A_Q \sharp\!* xvec$; $xvec \sharp\!* \Psi$; $xvec \sharp\!* \Psi_P$; $xvec \sharp\!* \Psi_Q$; $xvec \sharp\!* P$; $xvec \sharp\!* M$;
$xvec \sharp\!* Q$; $xvec \sharp\!* K$; $A_P \sharp\!* C$; $A_Q \sharp\!* C$; $xvec \sharp\!* C$$[\![\Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (\tau)\ ((\!|\nu{*}xvec|\!)(P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrMerge*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ (¿M(\!|N|\!))\ P'\ A_P\ \Psi_P;$
$extractFrame\ P = \langle A_P, \Psi_P \rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ (¿M(\!|N|\!))\ Q'\ A_Q\ \Psi_Q;$
$extractFrame\ Q = \langle A_Q, \Psi_Q \rangle;\ distinct\ A_Q;$
$A_P \sharp\!* \Psi$; $A_P \sharp\!* \Psi_Q$; $A_P \sharp\!* P$; $A_P \sharp\!* N$; $A_P \sharp\!* P'$;
$A_P \sharp\!* Q$; $A_P \sharp\!* Q'$; $A_P \sharp\!* A_Q$; $A_P \sharp\!* M$; $A_Q \sharp\!* M$;
$A_Q \sharp\!* \Psi$; $A_Q \sharp\!* \Psi_P$; $A_Q \sharp\!* P$; $A_Q \sharp\!* N$; $A_Q \sharp\!* P'$;
$A_Q \sharp\!* Q$; $A_Q \sharp\!* Q'$; $A_P \sharp\!* C$; $A_Q \sharp\!* C$$[\![\Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (¿M(\!|N|\!))\ (P' \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ xvec\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P \rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ (¿M(\!|N|\!))\ P'\ A_P\ \Psi_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N \rangle \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q \rangle;\ distinct\ A_Q;$
$distinct\ xvec;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ (¡M(\!|\nu{*}xvec|\!)\langle N \rangle)\ Q'\ A_Q\ \Psi_Q;$
$A_P \sharp\!* \Psi$; $A_P \sharp\!* \Psi_Q$; $A_P \sharp\!* P$; $A_P \sharp\!* N$; $A_P \sharp\!* P'$;
$A_P \sharp\!* Q$; $A_P \sharp\!* Q'$; $A_P \sharp\!* A_Q$; $A_P \sharp\!* xvec$; $A_Q \sharp\!* \Psi$; $A_Q \sharp\!* \Psi_P$;
$A_Q \sharp\!* P$; $A_Q \sharp\!* N$; $A_Q \sharp\!* P'$; $A_Q \sharp\!* Q$; $A_Q \sharp\!* Q'$;
$A_Q \sharp\!* xvec$; $xvec \sharp\!* \Psi$; $xvec \sharp\!* \Psi_P$; $xvec \sharp\!* \Psi_Q$; $xvec \sharp\!* P$;
$xvec \sharp\!* Q$; $A_P \sharp\!* C$; $A_Q \sharp\!* C$; $xvec \sharp\!* C$;
$A_P \sharp\!* M$; $A_Q \sharp\!* M$; $xvec \sharp\!* M$$[\![\Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (¡M(\!|\nu{*}xvec|\!)\langle N \rangle)\ (P' \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N \rangle \prec P';\ extractFrame\ P = \langle A_P, \Psi_P \rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ (¡M(\!|\nu{*}xvec|\!)\langle N \rangle)\ P'\ A_P\ \Psi_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q \rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ (¿M(\!|N|\!))\ Q'\ A_Q\ \Psi_Q;$
$distinct\ xvec;$
$A_P \sharp\!* \Psi$; $A_P \sharp\!* \Psi_Q$; $A_P \sharp\!* P$; $A_P \sharp\!* N$; $A_P \sharp\!* P'$;
$A_P \sharp\!* Q$; $A_P \sharp\!* Q'$; $A_P \sharp\!* A_Q$; $A_P \sharp\!* xvec$; $A_Q \sharp\!* \Psi$; $A_Q \sharp\!* \Psi_P$;
$A_Q \sharp\!* P$; $A_Q \sharp\!* N$; $A_Q \sharp\!* P'$; $A_Q \sharp\!* Q$; $A_Q \sharp\!* Q'$;
$A_Q \sharp\!* xvec$; $xvec \sharp\!* \Psi$; $xvec \sharp\!* \Psi_P$; $xvec \sharp\!* \Psi_Q$; $xvec \sharp\!* P$;
$xvec \sharp\!* Q$; $A_P \sharp\!* C$; $A_Q \sharp\!* C$; $xvec \sharp\!* C$;
$A_P \sharp\!* M$; $A_Q \sharp\!* M$; $xvec \sharp\!* M$$[\![\Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ (¡M(\!|\nu{*}xvec|\!)\langle N \rangle)\ (P' \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrClose*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ x\ C.$
$[\![\Psi \rhd P \longmapsto {¡}M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P';$
$x \in supp\ M;$
$\bigwedge C.\ Prop\ C\ \Psi\ P\ ({¡}M(\!|\nu{*}xvec|\!)\langle N\rangle)\ P'\ A_P\ \Psi_P;$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$A_P \sharp{*} \Psi;\ A_P \sharp{*} P;\ A_P \sharp{*} M;\ A_P \sharp{*} N;\ A_P \sharp{*} P';\ A_P \sharp{*} xvec;$
$distinct\ xvec;\ xvec \sharp{*} \Psi;\ xvec \sharp{*} \Psi_P;\ xvec \sharp{*} P;$
$xvec \sharp{*} M;$
$x \sharp \Psi;\ x \sharp xvec;\ x \sharp A_P;$
$A_P \sharp{*} C;\ xvec \sharp{*} C;\ x \sharp C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ (\tau)\ ((\!|\nu x|\!)((\!|\nu{*}xvec|\!)P'))\ (x\#A_P)\ \Psi_P$

and *rOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ A_P\ \Psi_P\ y\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ \Psi\ P\ (M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle)\ P'\ A_P\ \Psi_P;\ x \in supp\ N;\ x \sharp \Psi;\ x \sharp M;$
$x \sharp A_P;\ x \sharp xvec;\ x \sharp yvec;\ A_P \sharp{*} \Psi;\ A_P \sharp{*} P;\ A_P \sharp{*} M;\ A_P \sharp{*} N;\ A_P \sharp{*} P';$
$A_P \sharp{*} xvec;\ A_P \sharp{*} yvec;\ xvec \sharp{*} yvec;\ distinct\ xvec;\ distinct\ yvec;$
$xvec \sharp{*} \Psi;\ xvec \sharp{*} P;\ xvec \sharp{*} M;\ xvec \sharp{*} \Psi_P;$
$yvec \sharp{*} \Psi;\ yvec \sharp{*} P;\ yvec \sharp{*} M;\ A_P \sharp{*} C;\ x \sharp C;\ xvec \sharp{*} C;\ yvec \sharp{*} C;$
$y \neq x;\ y \sharp \Psi;\ y \sharp P;\ y \sharp M;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp N;\ y \sharp P';\ y \sharp A_P;\ y \sharp \Psi_P;\ y \sharp C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ (M(\!|\nu{*}(xvec@y\#yvec)|\!)\langle([(x,\ y)] \cdot N)\rangle)\ ([(x,\ y)] \cdot P')\ (x\#A_P)\ \Psi_P$

and *rBrOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ A_P\ \Psi_P\ y\ C.$
$[\![\Psi \rhd P \longmapsto {¡}M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ \Psi\ P\ ({¡}M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle)\ P'\ A_P\ \Psi_P;\ x \in supp\ N;\ x \sharp \Psi;\ x \sharp M;$
$x \sharp A_P;\ x \sharp xvec;\ x \sharp yvec;\ A_P \sharp{*} \Psi;\ A_P \sharp{*} P;\ A_P \sharp{*} M;\ A_P \sharp{*} N;\ A_P \sharp{*} P';$
$A_P \sharp{*} xvec;\ A_P \sharp{*} yvec;\ xvec \sharp{*} yvec;\ distinct\ xvec;\ distinct\ yvec;$
$xvec \sharp{*} \Psi;\ xvec \sharp{*} P;\ xvec \sharp{*} M;\ xvec \sharp{*} \Psi_P;$
$yvec \sharp{*} \Psi;\ yvec \sharp{*} P;\ yvec \sharp{*} M;\ A_P \sharp{*} C;\ x \sharp C;\ xvec \sharp{*} C;\ yvec \sharp{*} C;$
$y \neq x;\ y \sharp \Psi;\ y \sharp P;\ y \sharp M;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp N;\ y \sharp P';\ y \sharp A_P;\ y \sharp \Psi_P;\ y \sharp C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ ({¡}M(\!|\nu{*}(xvec@y\#yvec)|\!)\langle([(x,\ y)] \cdot N)\rangle)\ ([(x,\ y)] \cdot P')\ (x\#A_P)\ \Psi_P$

and *rScope*: $\bigwedge \Psi\ P\ \alpha\ P'\ x\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \longmapsto \alpha \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ \Psi\ P\ \alpha\ P'\ A_P\ \Psi_P;$
$x \sharp \Psi;\ x \sharp \alpha;\ x \sharp A_P;\ A_P \sharp{*} \Psi;\ A_P \sharp{*} P;$
$A_P \sharp{*} \alpha;\ A_P \sharp{*} P';$
$bn\ \alpha \sharp{*} \Psi;\ bn\ \alpha \sharp{*} P;\ bn\ \alpha \sharp{*} subject\ \alpha;\ bn\ \alpha \sharp{*} \Psi_P;$
$A_P \sharp{*} C;\ x \sharp C;\ bn\ \alpha \sharp{*} C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ \alpha\ ((\!|\nu x|\!)P')\ (x\#A_P)\ \Psi_P$

and *rBang*: $\bigwedge \Psi\ P\ \alpha\ P'\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \parallel\ !P \longmapsto \alpha \prec P';$ *guarded* P; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C.$ *Prop* $C\ \Psi\ (P \parallel\ !P)\ \alpha\ P'\ A_P\ (\Psi_P \otimes \mathbf{1})$; $\Psi_P \simeq \mathbf{1}$; *supp* $\Psi_P = (\{\}$::*name set*$)$;
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* \alpha$; $A_P \sharp* P'$; $A_P \sharp* C]\!] \Longrightarrow$ *Prop* $C\ \Psi\ (!P)\ \alpha\ P'\ ([])\ (\mathbf{1})$
shows *Prop* $C\ \Psi\ P\ \alpha\ P'\ A_P\ \Psi_P$
⟨*proof*⟩

lemma *inputFrameInduct*[*consumes 3*, *case-names cAlpha cInput cCase cPar1 cPar2 cScope cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and *Prop* :: $'f$::*fs-name* ⇒ $'b$ ⇒ $('a, 'b, 'c)$ *psi* ⇒
$'a$ ⇒ $'a$ ⇒ $('a, 'b, 'c)$ *psi* ⇒ *name list* ⇒ $'b$ ⇒ *bool*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *distinct* A_P
and *rAlpha*: $\bigwedge \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P\ p\ C.$ $[\![A_P \sharp* \Psi; A_P \sharp* P; A_P \sharp* M; A_P \sharp* N; A_P \sharp* P'; A_P \sharp* (p \cdot A_P); A_P \sharp* C;$
set $p \subseteq$ *set* $A_P \times$ *set*$(p \cdot A_P)$; *distinctPerm* p;
Prop $C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P]\!] \Longrightarrow$ *Prop* $C\ \Psi\ P\ M\ N\ P'\ (p \cdot A_P)\ (p \cdot \Psi_P)$
and *rInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C.$
$[\![\Psi \vdash M \leftrightarrow K$; *distinct xvec*; *set xvec* $\subseteq$ *supp* N;
length xvec = *length Tvec*; *xvec* $\sharp* \Psi$;
xvec $\sharp* M$; *xvec* $\sharp* K$; *xvec* $\sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (M(\!|\lambda* xvec\ N|\!).P)$
$K\ (N[xvec::=Tvec])\ (P[xvec::=Tvec])\ ([])\ (\mathbf{1})$
and *rCase*: $\bigwedge \Psi\ P\ M\ N\ P'\ \varphi\ Cs\ A_P\ \Psi_P\ C.$ $[\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P; $\bigwedge C.$ *Prop* $C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P$;
$(\varphi, P) \in$ *set Cs*; $\Psi \vdash \varphi$; *guarded* P; $\Psi_P \simeq \mathbf{1}$; *supp* $\Psi_P) = (\{\}$::*name set*$)$;
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* C]\!] \Longrightarrow$ *Prop* $C\ \Psi$ (*Cases Cs*) $M\ N\ P'\ ([])\ (\mathbf{1})$
and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_Q\ Q\ A_P\ \Psi_P\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P'$;
extractFrame $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_Q)\ P\ M\ N\ P'\ A_P\ \Psi_P$;
$A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \Psi$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* A_Q$; $A_P \sharp* \Psi_Q$;
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* M$; $A_Q \sharp* N$; $A_Q \sharp* P'$; A_Q

$\sharp* \Psi_P$;
$A_P \sharp* C$; $A_Q \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (P \parallel Q)\ M\ N\ (P' \parallel Q)\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$
and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ N\ Q'\ A_P\ P\ A_Q\ \Psi_Q\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|N|\!) \prec Q'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\bigwedge C.$ *Prop* $C\ (\Psi \otimes \Psi_P)\ Q\ M\ N\ Q'\ A_Q\ \Psi_Q$;
$A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \Psi$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* \Psi_Q$;
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* M$; $A_Q \sharp* N$; $A_Q \sharp* Q'$; $A_Q \sharp* \Psi_P$;
$A_P \sharp* C$; $A_Q \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (P \parallel Q)\ M\ N\ (P \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$
and *rScope*: $\bigwedge \Psi\ P\ M\ N\ P'\ x\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C.$ *Prop* $C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P$; $x \sharp \Psi$; $x \sharp M$; $x \sharp N$;
$x \sharp A_P$; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* C$; $x \sharp C]\!] \Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ M\ N\ ((\!|\nu x|\!)P')\ (x \# A_P)\ \Psi_P$
and *rBang*: $\bigwedge \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \parallel !P \longmapsto M(\!|N|\!) \prec P'$; *guarded* P; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C.$ *Prop* $C\ \Psi\ (P \parallel !P)\ M\ N\ P'\ A_P\ (\Psi_P \otimes \mathbf{1})$; $\Psi_P \simeq \mathbf{1}$; (*supp* Ψ_P) = ({}::*name set*);
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* C]\!] \Longrightarrow$
Prop $C\ \Psi\ (!P)\ M\ N\ P'\ ([])\ (\mathbf{1})$
shows *Prop* $C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P$
⟨*proof*⟩

lemma *brinputFrameInduct*[*consumes 3*, *case-names cAlpha cBrInput cCase cPar1 cPar2 cBrMerge cScope cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and *Prop* :: $'f$::*fs-name* ⇒ $'b$ ⇒ $('a, 'b, 'c)$ *psi* ⇒
$'a$ ⇒ $'a$ ⇒ $('a, 'b, 'c)$ *psi* ⇒ *name list* ⇒ $'b$ ⇒ *bool*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto ¿ M(\!|N|\!) \prec P'$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *distinct* A_P
and *rAlpha*: $\bigwedge \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P\ p\ C.\ [\![A_P \sharp* \Psi; A_P \sharp* P; A_P \sharp* M; A_P \sharp* N; A_P \sharp* P'; A_P \sharp* (p \cdot A_P); A_P \sharp* C;$
set $p \subseteq$ *set* $A_P \times$ *set*$(p \cdot A_P)$; *distinctPerm* p;
Prop $C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P]\!] \Longrightarrow$ *Prop* $C\ \Psi\ P\ M\ N\ P'\ (p \cdot A_P)\ (p \cdot \Psi_P)$

and *rBrInput*: $\bigwedge \Psi\ M\ K\ xvec\ N\ Tvec\ P\ C.$
$[\![\Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;\ xvec\ \sharp*\ \Psi;$
$xvec\ \sharp*\ M;\ xvec\ \sharp*\ K;\ xvec\ \sharp*\ C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (M(\!|\lambda* xvec\ N|\!).P)$
$K\ (N[xvec::=Tvec])\ (P[xvec::=Tvec])\ ([])\ (\mathbf{1})$

and *rCase*: $\bigwedge \Psi\ P\ M\ N\ P'\ \varphi\ Cs\ A_P\ \Psi_P\ C.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P;$
$(\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P;\ \Psi_P \simeq \mathbf{1};$
$(supp\ \Psi_P) = (\{\}::name\ set);$
$A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ P;\ A_P\ \sharp*\ M;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ P';\ A_P\ \sharp*\ C]\!] \Longrightarrow Prop\ C\ \Psi\ (Cases\ Cs)\ M\ N\ P'\ ([])\ (\mathbf{1})$

and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_Q\ Q\ A_P\ \Psi_P\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ N\ P'\ A_P\ \Psi_P;$
$A_P\ \sharp*\ P;\ A_P\ \sharp*\ Q;\ A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ M;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ P';\ A_P\ \sharp*\ A_Q;\ A_P\ \sharp*\ \Psi_Q;$
$A_Q\ \sharp*\ P;\ A_Q\ \sharp*\ Q;\ A_Q\ \sharp*\ \Psi;\ A_Q\ \sharp*\ M;\ A_Q\ \sharp*\ N;\ A_Q\ \sharp*\ P';\ A_Q\ \sharp*\ \Psi_P;$
$A_P\ \sharp*\ C;\ A_Q\ \sharp*\ C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ N\ (P' \parallel Q)\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ N\ Q'\ A_P\ P\ A_Q\ \Psi_Q\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ N\ Q'\ A_Q\ \Psi_Q;$
$A_P\ \sharp*\ P;\ A_P\ \sharp*\ Q;\ A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ M;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ Q';\ A_P\ \sharp*\ A_Q;\ A_P\ \sharp*\ \Psi_Q;$
$A_Q\ \sharp*\ P;\ A_Q\ \sharp*\ Q;\ A_Q\ \sharp*\ \Psi;\ A_Q\ \sharp*\ M;\ A_Q\ \sharp*\ N;\ A_Q\ \sharp*\ Q';\ A_Q\ \sharp*\ \Psi_P;$
$A_P\ \sharp*\ C;\ A_Q\ \sharp*\ C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ N\ (P' \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrMerge*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ N\ P'\ A_P\ \Psi_P;$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ \bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ N\ Q'\ A_Q\ \Psi_Q;$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_P\ \sharp*\ \Psi;\ A_P\ \sharp*\ \Psi_Q;\ A_P\ \sharp*\ P;\ A_P\ \sharp*\ N;\ A_P\ \sharp*\ P';$
$A_P\ \sharp*\ Q;\ A_P\ \sharp*\ Q';\ A_P\ \sharp*\ A_Q;\ A_P\ \sharp*\ M;\ A_Q\ \sharp*\ M;$
$A_Q\ \sharp*\ \Psi;\ A_Q\ \sharp*\ \Psi_P;\ A_Q\ \sharp*\ P;\ A_Q\ \sharp*\ N;\ A_Q\ \sharp*\ P';$
$A_Q\ \sharp*\ Q;\ A_Q\ \sharp*\ Q';\ A_P\ \sharp*\ C;\ A_Q\ \sharp*\ C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ N\ (P' \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rScope*: $\bigwedge \Psi\ P\ M\ N\ P'\ x\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$

$\bigwedge C.\ Prop\ C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P;\ x \sharp \Psi;\ x \sharp M;\ x \sharp N;$
$x \sharp A_P;\ A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* C;\ x \sharp C] \Longrightarrow$
$Prop\ C\ \Psi\ (\langle\!|\nu x|\!\rangle P)\ M\ N\ (\langle\!|\nu x|\!\rangle P')\ (x\#A_P)\ \Psi_P$

and *rBang*: $\bigwedge \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \parallel\ !P \longmapsto ¿M\langle\!|N|\!\rangle \prec P';\ guarded\ P;\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\bigwedge C.\ Prop\ C\ \Psi\ (P \parallel\ !P)\ M\ N\ P'\ A_P\ (\Psi_P \otimes \mathbf{1});\ \Psi_P \simeq \mathbf{1};\ (supp\ \Psi_P) = (\{\}::name\ set);$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;\ A_P \sharp* P';\ A_P \sharp* C] \Longrightarrow$
$Prop\ C\ \Psi\ (!P)\ M\ N\ P'\ ([])\ (\mathbf{1})$

shows $Prop\ C\ \Psi\ P\ M\ N\ P'\ A_P\ \Psi_P$

⟨*proof*⟩

lemma *outputFrameInduct*[*consumes 3*, *case-names cAlpha cOutput cCase cPar1 cPar2 cOpen cScope cBang*]:

fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and B :: $('a, 'b, 'c)\ boundOutput$
and A_P :: *name list*
and Ψ_P :: $'b$
and *Prop* :: $'f::fs\text{-}name \Rightarrow 'b \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow$
$'a \Rightarrow ('a, 'b, 'c)\ boundOutput \Rightarrow name\ list \Rightarrow 'b \Rightarrow bool$
and C :: $'f::fs\text{-}name$

assumes *Trans*: $\Psi \rhd P \longmapsto ROut\ M\ B$
and *FrP*: $extractFrame\ P = \langle A_P, \Psi_P\rangle$
and $distinct\ A_P$
and *rAlpha*: $\bigwedge \Psi\ P\ M\ A_P\ \Psi_P\ p\ B\ C.\ [\![A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* (p \cdot A_P);\ A_P \sharp* B;\ A_P \sharp* C;$
$set\ p \subseteq set\ A_P \times set(p \cdot A_P);\ distinctPerm\ p;$
$Prop\ C\ \Psi\ P\ M\ B\ A_P\ \Psi_P] \Longrightarrow Prop\ C\ \Psi\ P\ M\ B\ (p \cdot A_P)\ (p \cdot \Psi_P)$
and *rOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ \Psi \vdash M \leftrightarrow K \Longrightarrow Prop\ C\ \Psi\ (M\langle N\rangle.P)\ K\ (N \prec' P)\ ([])\ (\mathbf{1})$
and *rCase*: $\bigwedge \Psi\ P\ M\ B\ \varphi\ Cs\ A_P\ \Psi_P\ C.\ [\![\Psi \rhd P \longmapsto (ROut\ M\ B);\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ B\ A_P\ \Psi_P;$
$(\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P;\ \Psi_P \simeq \mathbf{1};\ (supp\ \Psi_P) = (\{\}::name\ set);$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* B;\ A_P \sharp* C] \Longrightarrow Prop\ C\ \Psi\ (Cases\ Cs)\ M\ B\ ([])\ (\mathbf{1})$
and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_Q\ Q\ A_P\ \Psi_P\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M\langle\!|\nu* xvec|\!\rangle\langle N\rangle \prec P';$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ (\langle\!|\nu* xvec|\!\rangle N \prec' P')\ A_P\ \Psi_P;$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;\ A_P \sharp* M;\ A_P \sharp* xvec;\ A_P \sharp* N;\ A_P \sharp* P';\ A_P \sharp* A_Q;\ A_P \sharp* \Psi_Q;$

$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* M$; $A_Q \sharp* xvec$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* \Psi_P$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$;
$A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ M $((\!|\nu* xvec|\!)N \prec' (P' \parallel Q))$ $(A_P @ A_Q)$ $(\Psi_P \otimes \Psi_Q)$

and *rPar2*: $\bigwedge \Psi$ Ψ_P Q M $xvec$ N Q' A_P P A_Q Ψ_Q C.
$\llbracket \Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|\nu* xvec|\!)\langle N \rangle \prec Q'$;
extractFrame $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
extractFrame $Q = \langle A_Q, \Psi_Q \rangle$; *distinct* A_Q;
$\bigwedge C$. *Prop* C $(\Psi \otimes \Psi_P)$ Q M $((\!|\nu* xvec|\!)N \prec' Q')$ A_Q Ψ_Q;
$A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \Psi$; $A_P \sharp* M$; $A_P \sharp* xvec$; $A_P \sharp* N$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* \Psi_Q$;
$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* M$; $A_Q \sharp* xvec$; $A_Q \sharp* N$; $A_Q \sharp* Q'$; $A_Q \sharp* \Psi_P$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$;
$A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $(P \parallel Q)$ M $((\!|\nu* xvec|\!)N \prec' (P \parallel Q'))$ $(A_P @ A_Q)$ $(\Psi_P \otimes \Psi_Q)$

and *rOpen*: $\bigwedge \Psi$ P M $xvec$ $yvec$ N P' x A_P Ψ_P C.
$\llbracket \Psi \rhd P \longmapsto M(\!|\nu*(xvec @ yvec)|\!)\langle N \rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* C Ψ P M $((\!|\nu*(xvec @ yvec)|\!)N \prec' P')$ A_P Ψ_P; $x \in supp\ N$; $x \sharp \Psi$; $x \sharp M$;
$x \sharp A_P$; $x \sharp xvec$; $x \sharp yvec$; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* xvec$; $A_P \sharp* yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$;
$yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $A_P \sharp* C$; $x \sharp C$; $xvec \sharp* C$; $yvec \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $((\!|\nu x|\!)P)$ M $((\!|\nu*(xvec @ x\#yvec)|\!)N \prec' P')$ $(x\#A_P)$ Ψ_P

and *rScope*: $\bigwedge \Psi$ P M $xvec$ N P' x A_P Ψ_P C.
$\llbracket \Psi \rhd P \longmapsto M(\!|\nu* xvec|\!)\langle N \rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* C Ψ P M $((\!|\nu* xvec|\!)N \prec' P')$ A_P Ψ_P;
$x \sharp \Psi$; $x \sharp M$; $x \sharp xvec$; $x \sharp N$; $x \sharp A_P$; $A_P \sharp* \Psi$; $A_P \sharp* P$;
$A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* xvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$;
$A_P \sharp* C$; $x \sharp C$; $xvec \sharp* C$$\rrbracket \Longrightarrow$
Prop C Ψ $((\!|\nu x|\!)P)$ M $((\!|\nu* xvec|\!)N \prec' ((\!|\nu x|\!)P'))$ $(x\#A_P)$ Ψ_P

and *rBang*: $\bigwedge \Psi$ P M B A_P Ψ_P C.
$\llbracket \Psi \rhd P \parallel !P \longmapsto ROut\ M\ B$; *guarded* P; *extractFrame* $P = \langle A_P, \Psi_P \rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* C Ψ $(P \parallel !P)$ M B A_P $(\Psi_P \otimes \mathbf{1})$; $\Psi_P \simeq \mathbf{1}$; $supp\ \Psi_P = (\{\}::name\ set)$;
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* C$$\rrbracket \Longrightarrow$ *Prop* C Ψ $(!P)$ M B $([])$ $(\mathbf{1})$

shows *Prop* C Ψ P M B A_P Ψ_P
⟨*proof*⟩

lemma *broutputFrameInduct*[*consumes 3*, *case-names cAlpha cBrOutput cCase cPar1 cPar2 cBrComm1 cBrComm2 cBrOpen cScope cBang*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and M :: *'a*
and B :: (*'a*, *'b*, *'c*) *boundOutput*
and A_P :: *name list*
and Ψ_P :: *'b*
and *Prop* :: *'f*::*fs-name* ⇒ *'b* ⇒ (*'a*, *'b*, *'c*) *psi* ⇒
'a ⇒ (*'a*, *'b*, *'c*) *boundOutput* ⇒ *name list* ⇒ *'b* ⇒ *bool*
and C :: *'f*::*fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto RBrOut\ M\ B$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *distinct* A_P
and *rAlpha*: $\bigwedge \Psi\ P\ M\ A_P\ \Psi_P\ p\ B\ C.\ [\![A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* (p \cdot A_P);\ A_P \sharp* B;\ A_P \sharp* C;$
$set\ p \subseteq set\ A_P \times set(p \cdot A_P);\ distinctPerm\ p;$
$Prop\ C\ \Psi\ P\ M\ B\ A_P\ \Psi_P]\!] \Longrightarrow Prop\ C\ \Psi\ P\ M\ B\ (p \cdot A_P)\ (p \cdot \Psi_P)$
and *rBrOutput*: $\bigwedge \Psi\ M\ K\ N\ P\ C.\ \Psi \vdash M \preceq K \Longrightarrow Prop\ C\ \Psi\ (M\langle N\rangle.P)\ K\ (N \prec' P)\ ([])\ (\mathbf{1})$
and *rCase*: $\bigwedge \Psi\ P\ M\ B\ \varphi\ Cs\ A_P\ \Psi_P\ C.\ [\![\Psi \rhd P \longmapsto (RBrOut\ M\ B);\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;\ \bigwedge C.\ Prop\ C\ \Psi\ P\ M\ B\ A_P\ \Psi_P;$
$(\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P;\ \Psi_P \simeq \mathbf{1};\ (supp\ \Psi_P) = (\{\}::name\ set);$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* B;\ A_P \sharp* C]\!] \Longrightarrow Prop\ C\ \Psi\ (Cases\ Cs)\ M\ B\ ([])\ (\mathbf{1})$
and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_Q\ Q\ A_P\ \Psi_P\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P';$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ ((\!|\nu * xvec|\!)N \prec' P')\ A_P\ \Psi_P;$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;\ A_P \sharp* M;\ A_P \sharp* xvec;\ A_P \sharp* N;\ A_P \sharp* P';\ A_P \sharp* A_Q;\ A_P \sharp* \Psi_Q;$
$A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* M;\ A_Q \sharp* xvec;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* \Psi_P;$
$xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* Q;\ xvec \sharp* M;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;$
$A_P \sharp* C;\ A_Q \sharp* C;\ xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu * xvec|\!)N \prec' (P' \parallel Q))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$
and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ M\ xvec\ N\ Q'\ A_P\ P\ A_Q\ \Psi_Q\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';$
$extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ ((\!|\nu * xvec|\!)N \prec' Q')\ A_Q\ \Psi_Q;$
$A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* \Psi;\ A_P \sharp* M;\ A_P \sharp* xvec;\ A_P \sharp* N;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* \Psi_Q;$

$A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* M$; $A_Q \sharp* xvec$; $A_Q \sharp* N$; $A_Q \sharp* Q'$; $A_Q \sharp* \Psi_P$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$;
$A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu * xvec|\!)N \prec' (P \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ xvec\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿ M (\!|N|\!) \prec P'$; $extractFrame\ P = \langle A_P, \Psi_P \rangle$; $distinct\ A_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡ M (\!|\nu * xvec|\!) \langle N \rangle \prec Q'$; $extractFrame\ Q = \langle A_Q, \Psi_Q \rangle$; $distinct\ A_Q$;
$distinct\ xvec$;
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_P)\ Q\ M\ ((\!|\nu * xvec|\!)N \prec' Q')\ A_Q\ \Psi_Q$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* Q'$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu * xvec|\!)N \prec' (P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡ M (\!|\nu * xvec|\!) \langle N \rangle \prec P'$; $extractFrame\ P = \langle A_P, \Psi_P \rangle$; $distinct\ A_P$;
$\bigwedge C.\ Prop\ C\ (\Psi \otimes \Psi_Q)\ P\ M\ ((\!|\nu * xvec|\!)N \prec' P')\ A_P\ \Psi_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿ M (\!|N|\!) \prec Q'$; $extractFrame\ Q = \langle A_Q, \Psi_Q \rangle$; $distinct\ A_Q$;
$distinct\ xvec$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* Q'$;
$A_Q \sharp* xvec$; $xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$;
$xvec \sharp* Q$; $A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel Q)\ M\ ((\!|\nu * xvec|\!)N \prec' (P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrOpen*: $\bigwedge \Psi\ P\ M\ xvec\ yvec\ N\ P'\ x\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \longmapsto ¡ M (\!|\nu * (xvec @ yvec)|\!) \langle N \rangle \prec P'$; $extractFrame\ P = \langle A_P, \Psi_P \rangle$; $distinct\ A_P$;
$\bigwedge C.\ Prop\ C\ \Psi\ P\ M\ ((\!|\nu * (xvec @ yvec)|\!)N \prec' P')\ A_P\ \Psi_P$; $x \in supp\ N$; $x \sharp \Psi$; $x \sharp M$;
$x \sharp A_P$; $x \sharp xvec$; $x \sharp yvec$; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* xvec$; $A_P \sharp* yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* \Psi_P$;
$yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $A_P \sharp* C$; $x \sharp C$; $xvec \sharp* C$; $yvec \sharp* C]\!] \Longrightarrow$
$Prop\ C\ \Psi\ ((\!|\nu x|\!)P)\ M\ ((\!|\nu * (xvec @ x \# yvec)|\!)N \prec' P')\ (x \# A_P)\ \Psi_P$

and *rScope*: $\bigwedge \Psi\ P\ M\ xvec\ N\ P'\ x\ A_P\ \Psi_P\ C.$

$[\![\Psi \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* $C\ \Psi\ P\ M\ ((\!|\nu{*}xvec|\!)N \prec' P')\ A_P\ \Psi_P$;
$x \sharp \Psi$; $x \sharp M$; $x \sharp xvec$; $x \sharp N$; $x \sharp A_P$; $A_P \sharp{*} \Psi$; $A_P \sharp{*} P$;
$A_P \sharp{*} M$; $A_P \sharp{*} N$; $A_P \sharp{*} P'$; $A_P \sharp{*} xvec$;
$xvec \sharp{*} \Psi$; $xvec \sharp{*} P$; $xvec \sharp{*} M$; $xvec \sharp{*} \Psi_P$;
$A_P \sharp{*} C$; $x \sharp C$; $xvec \sharp{*} C]\!] \Longrightarrow$
Prop $C\ \Psi\ ((\!|\nu x|\!)P)\ M\ ((\!|\nu{*}xvec|\!)N \prec' ((\!|\nu x|\!)P'))\ (x\#A_P)\ \Psi_P$

and *rBang*: $\bigwedge \Psi\ P\ M\ B\ A_P\ \Psi_P\ C$.
$[\![\Psi \rhd P \parallel\ !P \longmapsto RBrOut\ M\ B$; *guarded* P; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\bigwedge C$. *Prop* $C\ \Psi\ (P \parallel\ !P)\ M\ B\ A_P\ (\Psi_P \otimes \mathbf{1})$; $\Psi_P \simeq \mathbf{1}$; *supp* $\Psi_P = (\{\}::name\ set)$;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} P$; $A_P \sharp{*} M$; $A_P \sharp{*} C]\!] \Longrightarrow$ *Prop* $C\ \Psi\ (!P)\ M\ B\ ([])\ (\mathbf{1})$

shows *Prop* $C\ \Psi\ P\ M\ B\ A_P\ \Psi_P$

⟨*proof*⟩

lemma *tauFrameInduct*[*consumes 3*, *case-names cAlpha cCase cPar1 cPar2 cComm1 cComm2 cBrClose cScope cBang*]:
fixes Ψ :: *'b*
and P :: *('a, 'b, 'c) psi*
and P' :: *('a, 'b, 'c) psi*
and *Prop* :: *'f::fs-name* ⇒ *'b* ⇒ *('a, 'b, 'c) psi* ⇒ *('a, 'b, 'c) psi* ⇒ *name list* ⇒ *'b* ⇒ *bool*
and C :: *'f::fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto \tau \prec P'$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *distinct* A_P
and *rAlpha*: $\bigwedge \Psi\ P\ P'\ A_P\ \Psi_P\ p\ C$. $[\![A_P \sharp{*} \Psi$; $A_P \sharp{*} P$; $A_P \sharp{*} P'$; $A_P \sharp{*} (p \cdot A_P)$; $A_P \sharp{*} C$;
set $p \subseteq$ *set* $A_P \times$ *set* $(p \cdot A_P)$; *distinctPerm* p;
Prop $C\ \Psi\ P\ P'\ A_P\ \Psi_P]\!] \Longrightarrow$ *Prop* $C\ \Psi\ P\ P'\ (p \cdot A_P)\ (p \cdot \Psi_P)$
and *rCase*: $\bigwedge \Psi\ P\ P'\ \varphi\ Cs\ A_P\ \Psi_P\ C$. $[\![\Psi \rhd P \longmapsto \tau \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P; $\bigwedge C$. *Prop* $C\ \Psi\ P\ P'\ A_P\ \Psi_P$;
$(\varphi, P) \in$ *set* Cs; $\Psi \vdash \varphi$; *guarded* P; $\Psi_P \simeq \mathbf{1}$; $(supp\ \Psi_P) = (\{\}::name\ set)$;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} P$; $A_P \sharp{*} P'$; $A_P \sharp{*} C]\!] \Longrightarrow$ *Prop* $C\ \Psi\ (Cases\ Cs)\ P'\ ([])\ (\mathbf{1})$
and *rPar1*: $\bigwedge \Psi\ \Psi_Q\ P\ P'\ A_Q\ Q\ A_P\ \Psi_P\ C$.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto \tau \prec P'$;
extractFrame $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
extractFrame $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\bigwedge C$. *Prop* $C\ (\Psi \otimes \Psi_Q)\ P\ P'\ A_P\ \Psi_P$;
$A_P \sharp{*} P$; $A_P \sharp{*} Q$; $A_P \sharp{*} \Psi$; $A_P \sharp{*} P'$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} \Psi_Q$;
$A_Q \sharp{*} P$; $A_Q \sharp{*} Q$; $A_Q \sharp{*} \Psi$; $A_Q \sharp{*} P'$; $A_Q \sharp{*} \Psi_P$;
$A_P \sharp{*} C$; $A_Q \sharp{*} C]\!] \Longrightarrow$

$$\mathit{Prop}\ C\ \Psi\ (P \parallel Q)\ (P' \parallel Q)\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$$

and *rPar2*: $\bigwedge \Psi\ \Psi_P\ Q\ Q'\ A_P\ P\ A_Q\ \Psi_Q\ C.$
$[\![\Psi \otimes \Psi_P \rhd Q \longmapsto \tau \prec Q';$
$\mathit{extractFrame}\ P = \langle A_P, \Psi_P \rangle;\ \mathit{distinct}\ A_P;$
$\mathit{extractFrame}\ Q = \langle A_Q, \Psi_Q \rangle;\ \mathit{distinct}\ A_Q;$
$\bigwedge C.\ \mathit{Prop}\ C\ (\Psi \otimes \Psi_P)\ Q\ Q'\ A_Q\ \Psi_Q;$
$A_P \sharp^* P;\ A_P \sharp^* Q;\ A_P \sharp^* \Psi;\ A_P \sharp^* Q';\ A_P \sharp^* A_Q;\ A_P \sharp^* \Psi_Q;$
$A_Q \sharp^* P;\ A_Q \sharp^* Q;\ A_Q \sharp^* \Psi;\ A_Q \sharp^* Q';\ A_Q \sharp^* \Psi_P;$
$A_P \sharp^* C;\ A_Q \sharp^* C]\!] \Longrightarrow$
$\mathit{Prop}\ C\ \Psi\ (P \parallel Q)\ (P \parallel Q')\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rComm1*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ K\ \mathit{xvec}\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P';\ \mathit{extractFrame}\ P = \langle A_P, \Psi_P \rangle;$
$\mathit{distinct}\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu * \mathit{xvec}|\!)\langle N \rangle \prec Q';\ \mathit{extractFrame}\ Q = \langle A_Q, \Psi_Q \rangle;\ \mathit{distinct}\ A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;\ \mathit{distinct}\ \mathit{xvec};$
$A_P \sharp^* \Psi;\ A_P \sharp^* \Psi_Q;\ A_P \sharp^* P;\ A_P \sharp^* M;\ A_P \sharp^* N;\ A_P \sharp^* P';$
$A_P \sharp^* Q;\ A_P \sharp^* Q';\ A_P \sharp^* A_Q;\ A_P \sharp^* \mathit{xvec};\ A_Q \sharp^* \Psi;\ A_Q \sharp^* \Psi_P;$
$A_Q \sharp^* P;\ A_Q \sharp^* N;\ A_Q \sharp^* P';\ A_Q \sharp^* Q;\ A_Q \sharp^* K;\ A_Q \sharp^* Q';$
$A_Q \sharp^* \mathit{xvec};\ \mathit{xvec} \sharp^* \Psi;\ \mathit{xvec} \sharp^* \Psi_P;\ \mathit{xvec} \sharp^* \Psi_Q;\ \mathit{xvec} \sharp^* P;\ \mathit{xvec} \sharp^* M;$
$\mathit{xvec} \sharp^* Q;\ \mathit{xvec} \sharp^* K;\ A_P \sharp^* C;\ A_Q \sharp^* C;\ \mathit{xvec} \sharp^* C]\!] \Longrightarrow$
$\mathit{Prop}\ C\ \Psi\ (P \parallel Q)\ ((\!|\nu * \mathit{xvec}|\!)(P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rComm2*: $\bigwedge \Psi\ \Psi_Q\ P\ M\ \mathit{xvec}\ N\ P'\ A_P\ \Psi_P\ Q\ K\ Q'\ A_Q\ C.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * \mathit{xvec}|\!)\langle N \rangle \prec P';\ \mathit{extractFrame}\ P = \langle A_P, \Psi_P \rangle;\ \mathit{distinct}\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q';\ \mathit{extractFrame}\ Q = \langle A_Q, \Psi_Q \rangle;\ \mathit{distinct}\ A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;\ \mathit{distinct}\ \mathit{xvec};$
$A_P \sharp^* \Psi;\ A_P \sharp^* \Psi_Q;\ A_P \sharp^* P;\ A_P \sharp^* M;\ A_P \sharp^* N;\ A_P \sharp^* P';$
$A_P \sharp^* Q;\ A_P \sharp^* Q';\ A_P \sharp^* A_Q;\ A_P \sharp^* \mathit{xvec};\ A_Q \sharp^* \Psi;\ A_Q \sharp^* \Psi_P;$
$A_Q \sharp^* P;\ A_Q \sharp^* N;\ A_Q \sharp^* P';\ A_Q \sharp^* Q;\ A_Q \sharp^* K;\ A_Q \sharp^* Q';$
$A_Q \sharp^* \mathit{xvec};\ \mathit{xvec} \sharp^* \Psi;\ \mathit{xvec} \sharp^* \Psi_P;\ \mathit{xvec} \sharp^* \Psi_Q;\ \mathit{xvec} \sharp^* P;\ \mathit{xvec} \sharp^* M;$
$\mathit{xvec} \sharp^* Q;\ \mathit{xvec} \sharp^* K;\ A_P \sharp^* C;\ A_Q \sharp^* C;\ \mathit{xvec} \sharp^* C]\!] \Longrightarrow$
$\mathit{Prop}\ C\ \Psi\ (P \parallel Q)\ ((\!|\nu * \mathit{xvec}|\!)(P' \parallel Q'))\ (A_P @ A_Q)\ (\Psi_P \otimes \Psi_Q)$

and *rBrClose*: $\bigwedge \Psi\ P\ M\ \mathit{xvec}\ N\ P'\ A_P\ \Psi_P\ x\ C.$
$[\![\Psi \rhd P \longmapsto ¡M(\!|\nu * \mathit{xvec}|\!)\langle N \rangle \prec P';$
$x \in \mathit{supp}\ M;$
$\mathit{extractFrame}\ P = \langle A_P, \Psi_P \rangle;\ \mathit{distinct}\ A_P;$
$A_P \sharp^* \Psi;\ A_P \sharp^* P;\ A_P \sharp^* M;\ A_P \sharp^* N;\ A_P \sharp^* P';\ A_P \sharp^* \mathit{xvec};$
$\mathit{distinct}\ \mathit{xvec};\ \mathit{xvec} \sharp^* \Psi;\ \mathit{xvec} \sharp^* \Psi_P;\ \mathit{xvec} \sharp^* P;$
$\mathit{xvec} \sharp^* M;$
$x \sharp \Psi;\ x \sharp \mathit{xvec};\ x \sharp A_P;$
$A_P \sharp^* C;\ \mathit{xvec} \sharp^* C;\ x \sharp C]\!] \Longrightarrow$
$\mathit{Prop}\ C\ \Psi\ ((\!|\nu x|\!)P)\ ((\!|\nu x|\!)((\!|\nu * \mathit{xvec}|\!)P'))\ (x \# A_P)\ \Psi_P$

and *rScope*: $\bigwedge \Psi\ P\ P'\ x\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \longmapsto \tau \prec P';\ \mathit{extractFrame}\ P = \langle A_P, \Psi_P \rangle;\ \mathit{distinct}\ A_P;$
$\bigwedge C.\ \mathit{Prop}\ C\ \Psi\ P\ P'\ A_P\ \Psi_P;\ x \sharp \Psi;$

$x \sharp A_P$; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* P'$;
$A_P \sharp* C$; $x \sharp C]\!] \Longrightarrow$
Prop $C\ \Psi\ (([\!|\nu x|\!])P)\ (([\!|\nu x|\!])P')\ (x\#A_P)\ \Psi_P$

and *rBang*: $\bigwedge \Psi\ P\ P'\ A_P\ \Psi_P\ C.$
$[\![\Psi \rhd P \parallel\ !P \longmapsto \tau \prec P'$; *guarded* P; *extractFrame* $P = \langle A_P, \Psi_P\rangle$;
distinct A_P;
$\bigwedge C.$ *Prop* $C\ \Psi\ (P \parallel\ !P)\ P'\ A_P\ (\Psi_P \otimes \mathbf{1})$; $\Psi_P \simeq \mathbf{1}$; *supp* $\Psi_P =$ ({}::*name set*);
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* P'$; $A_P \sharp* C]\!] \Longrightarrow$ *Prop* $C\ \Psi\ (!P)\ P'$ ([]) (**1**)
shows *Prop* $C\ \Psi\ P\ P'\ A_P\ \Psi_P$
⟨*proof*⟩

lemma *inputFreshDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and $x \sharp P$
and $x \sharp N$

shows $x \sharp P'$
⟨*proof*⟩

lemma *brinputFreshDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and $x \sharp P$
and $x \sharp N$

shows $x \sharp P'$
⟨*proof*⟩

lemma *inputFreshChainDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*

and *xvec* :: *name list*

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and $xvec \sharp* P$
and $xvec \sharp* N$

shows $xvec \sharp* P'$
⟨*proof*⟩

lemma *brinputFreshChainDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and *xvec* :: *name list*

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and $xvec \sharp* P$
and $xvec \sharp* N$

shows $xvec \sharp* P'$
⟨*proof*⟩

lemma *outputFreshDerivativeN*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|\nu*xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and *distinct xvec*
and $x \sharp P$
and $x \sharp xvec$

shows $x \sharp N$
⟨*proof*⟩

lemma *broutputFreshDerivativeN*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*

and x :: *name*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and *distinct xvec*
and $x \sharp P$
and $x \sharp xvec$

shows $x \sharp N$
⟨*proof*⟩

lemma *outputFreshDerivativeP*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and *distinct xvec*
and $x \sharp P$
and $x \sharp xvec$

shows $x \sharp P'$
⟨*proof*⟩

lemma *broutputFreshDerivativeP*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and *distinct xvec*
and $x \sharp P$
and $x \sharp xvec$

shows $x \sharp P'$
⟨*proof*⟩

lemma *outputFreshDerivative*:
fixes Ψ :: $'b$

and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *xvec* $\sharp*$ M
and *distinct xvec*
and $x \sharp P$
and $x \sharp$ *xvec*

shows $x \sharp N$
and $x \sharp P'$
⟨*proof*⟩

lemma *broutputFreshDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *xvec* $\sharp*$ M
and *distinct xvec*
and $x \sharp P$
and $x \sharp$ *xvec*

shows $x \sharp N$
and $x \sharp P'$
⟨*proof*⟩

lemma *outputFreshChainDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and *yvec* :: *name list*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *xvec* $\sharp*$ M
and *distinct xvec*
and *yvec* $\sharp*$ P

and $yvec \sharp* xvec$

shows $yvec \sharp* N$
and $yvec \sharp* P'$
⟨*proof*⟩

lemma *broutputFreshChainDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and $xvec$ :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and $yvec$ :: *name list*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and *distinct xvec*
and $yvec \sharp* P$
and $yvec \sharp* xvec$

shows $yvec \sharp* N$
and $yvec \sharp* P'$
⟨*proof*⟩

lemma *tauFreshDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto \tau \prec P'$
and $x \sharp P$

shows $x \sharp P'$
⟨*proof*⟩

lemma *tauFreshChainDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and $xvec$ :: *name list*

assumes $\Psi \rhd P \longmapsto \tau \prec P'$
and $xvec \sharp* P$

shows $xvec \sharp* P'$

⟨*proof*⟩

lemma *freeFreshDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto \alpha \prec P'$
and $bn\ \alpha \sharp* subject\ \alpha$
and $distinct(bn\ \alpha)$
and $x \sharp \alpha$
and $x \sharp P$

shows $x \sharp P'$
⟨*proof*⟩

lemma *freeFreshChainDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*
and *xvec* :: *name list*

assumes $\Psi \rhd P \longmapsto \alpha \prec P'$
and $bn\ \alpha \sharp* subject\ \alpha$
and $distinct(bn\ \alpha)$
and $xvec \sharp* P$
and $xvec \sharp* \alpha$

shows $xvec \sharp* P'$
⟨*proof*⟩

lemma *Input*:
fixes Ψ :: $'b$
and M :: $'a$
and K :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and *Tvec* :: $'a$ *list*

assumes $\Psi \vdash M \leftrightarrow K$
and *distinct xvec*
and *set xvec* $\subseteq$ *supp N*
and *length xvec* = *length Tvec*

shows $\Psi \rhd M(\!|\lambda* xvec\ N|\!).P \longmapsto K(\!|N[xvec::=Tvec]|\!) \prec P[xvec::=Tvec]$
⟨*proof*⟩

lemma *BrInput*:
fixes Ψ :: $'b$
and M :: $'a$
and K :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and *Tvec* :: $'a$ *list*

assumes $\Psi \vdash K \succeq M$
and *distinct xvec*
and *set xvec* $\subseteq$ *supp N*
and *length xvec* = *length Tvec*

shows $\Psi \rhd M(\!|\lambda * xvec\ N|\!).P \longmapsto ¿K(\!|N[xvec ::= Tvec]|\!) \prec P[xvec ::= Tvec]$
⟨*proof*⟩

lemma *residualAlpha*:
fixes p :: *name prm*
and α :: $'a$ *action*
and P :: ($'a$, $'b$, $'c$) *psi*

assumes $bn(p \cdot \alpha) \sharp * object\ \alpha$
and $bn(p \cdot \alpha) \sharp * P$
and $bn\ \alpha \sharp * subject\ \alpha$
and $bn(p \cdot \alpha) \sharp * subject\ \alpha$
and *set* $p \subseteq set(bn\ \alpha) \times set(bn(p \cdot \alpha))$

shows $\alpha \prec P = (p \cdot \alpha) \prec (p \cdot P)$
⟨*proof*⟩

lemma *Par1*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and P :: ($'a$, $'b$, $'c$) *psi*
and α :: $'a$ *action*
and P' :: ($'a$, $'b$, $'c$) *psi*
and A_Q :: *name list*
and Q :: ($'a$, $'b$, $'c$) *psi*

assumes *Trans*: $\Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P'$
and *extractFrame* $Q = \langle A_Q, \Psi_Q \rangle$
and $bn\ \alpha \sharp * Q$
and $A_Q \sharp * \Psi$
and $A_Q \sharp * P$
and $A_Q \sharp * \alpha$

shows $\Psi \rhd P \parallel Q \longmapsto \alpha \prec (P' \parallel Q)$
⟨*proof*⟩

lemma *Par2*:
fixes Ψ :: $'b$
and Ψ_P :: $'b$
and Q :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and Q' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and P :: $('a, 'b, 'c)$ *psi*

assumes *Trans*: $\Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q'$
and *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and *bn* $\alpha \sharp* P$
and $A_P \sharp* \Psi$
and $A_P \sharp* Q$
and $A_P \sharp* \alpha$

shows $\Psi \rhd P \parallel Q \longmapsto \alpha \prec (P \parallel Q')$
⟨*proof*⟩

lemma *Open*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and *yvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes *Trans*: $\Psi \rhd P \longmapsto M(\!|\nu*(xvec @ yvec)|\!)\langle N \rangle \prec P'$
and $x \in supp\ N$
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp xvec$
and $x \sharp yvec$

shows $\Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|\nu*(xvec @ x \# yvec)|\!)\langle N \rangle \prec P'$
⟨*proof*⟩

lemma *BrOpen*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and *yvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes *Trans*: $\Psi \rhd P \longmapsto ¡ M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'$
and $x \in supp\ N$
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp xvec$
and $x \sharp yvec$

shows $\Psi \rhd (\!|\nu x|\!)P \longmapsto ¡ M(\!|\nu*(xvec@x\#yvec)|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *Scope*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and α :: $'a\ action$
and P' :: $('a, 'b, 'c)\ psi$
and x :: *name*

assumes $\Psi \rhd P \longmapsto \alpha \prec P'$
and $x \sharp \Psi$
and $x \sharp \alpha$

shows $\Psi \rhd (\!|\nu x|\!)P \longmapsto \alpha \prec (\!|\nu x|\!)P'$
⟨*proof*⟩

lemma *inputSwapFrameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and $x \sharp P$
and $y \sharp P$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto ([(x, y)] \cdot M)(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *brinputSwapFrameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and $x \sharp P$
and $y \sharp P$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto ¿([(x, y)] \cdot M)(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *inputPermFrameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and Xs :: *name set*
and Ys :: *name set*

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and S: *set* $p \subseteq Xs \times Ys$
and $Xs \sharp* P$
and $Ys \sharp* P$

shows $(p \cdot \Psi) \rhd P \longmapsto (p \cdot M)(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *brinputPermFrameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and Xs :: *name set*
and Ys :: *name set*

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and S: *set* $p \subseteq Xs \times Ys$
and $Xs \sharp* P$
and $Ys \sharp* P$

shows $(p \cdot \Psi) \rhd P \longmapsto ¿(p \cdot M)(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *inputSwapSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$

and $P' :: ('a, 'b, 'c)\ psi$
and $x :: name$
and $y :: name$

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and $x \sharp P$
and $y \sharp P$
and $x \sharp \Psi$
and $y \sharp \Psi$

shows $\Psi \rhd P \longmapsto ([(x, y)] \cdot M)(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *brinputSwapSubject*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $M :: 'a$
and $N :: 'a$
and $P' :: ('a, 'b, 'c)\ psi$
and $x :: name$
and $y :: name$

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and $x \sharp P$
and $y \sharp P$
and $x \sharp \Psi$
and $y \sharp \Psi$

shows $\Psi \rhd P \longmapsto ¿([(x, y)] \cdot M)(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *inputPermSubject*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $M :: 'a$
and $N :: 'a$
and $P' :: ('a, 'b, 'c)\ psi$
and $p :: name\ prm$
and $Xs :: name\ set$
and $Ys :: name\ set$

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and S: $set\ p \subseteq Xs \times Ys$
and $Xs \sharp* P$
and $Ys \sharp* P$
and $Xs \sharp* \Psi$
and $Ys \sharp* \Psi$

shows $\Psi \rhd P \longmapsto (p \cdot M)(\!|N|\!) \prec P'$

⟨*proof*⟩

lemma *brinputPermSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and Xs :: *name set*
and Ys :: *name set*

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and S: *set* $p \subseteq Xs \times Ys$
and $Xs \sharp* P$
and $Ys \sharp* P$
and $Xs \sharp* \Psi$
and $Ys \sharp* \Psi$

shows $\Psi \rhd P \longmapsto ¿(p \cdot M)(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *inputSwapFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and $x \sharp P$
and $y \sharp P$
and $x \sharp M$
and $y \sharp M$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto M(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *brinputSwapFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and $x \sharp P$
and $y \sharp P$
and $x \sharp M$
and $y \sharp M$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *inputPermFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and Xs :: *name set*
and Ys :: *name set*

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and S: *set* $p \subseteq Xs \times Ys$
and $Xs \sharp* P$
and $Ys \sharp* P$
and $Xs \sharp* M$
and $Ys \sharp* M$

shows $(p \cdot \Psi) \rhd P \longmapsto M(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *brinputPermFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and Xs :: *name set*
and Ys :: *name set*

assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and S: *set* $p \subseteq Xs \times Ys$
and $Xs \sharp* P$
and $Ys \sharp* P$
and $Xs \sharp* M$
and $Ys \sharp* M$

shows $(p \cdot \Psi) \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *inputAlpha*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and *xvec* :: *name list*

assumes $\Psi \rhd P \longmapsto M(\!|N|\!) \prec P'$
and *set* $p \subseteq$ (*set xvec*) $\times$ (*set* $(p \cdot xvec)$)
and *distinctPerm* p
and *xvec* $\sharp*$ P
and $(p \cdot xvec) \sharp* P$

shows $\Psi \rhd P \longmapsto M(\!|(p \cdot N)|\!) \prec (p \cdot P')$
⟨*proof*⟩

lemma *brinputAlpha*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and *xvec* :: *name list*

assumes $\Psi \rhd P \longmapsto ¿ M(\!|N|\!) \prec P'$
and *set* $p \subseteq$ (*set xvec*) $\times$ (*set* $(p \cdot xvec)$)
and *distinctPerm* p
and *xvec* $\sharp*$ P
and $(p \cdot xvec) \sharp* P$

shows $\Psi \rhd P \longmapsto ¿ M(\!|(p \cdot N)|\!) \prec (p \cdot P')$
⟨*proof*⟩

lemma *frameFresh*[*dest*]:
fixes x :: *name*
and A_F :: *name list*
and Ψ_F :: $'b$

assumes $x \sharp A_F$
and $x \sharp \langle A_F, \Psi_F \rangle$

shows $x \sharp \Psi_F$
⟨*proof*⟩

lemma *outputSwapFrameSubject*:
fixes Ψ :: $'b$

and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and $x \sharp P$
and $y \sharp P$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto([(x, y)] \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *broutputSwapFrameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and $x \sharp P$
and $y \sharp P$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto ¡([(x, y)] \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *outputPermFrameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and S: *set* $p \subseteq$ *set yvec* $\times$ *set zvec*
and $yvec \sharp* P$
and $zvec \sharp* P$

shows $(p \cdot \Psi) \rhd P \longmapsto(p \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$

⟨*proof*⟩

lemma *broutputPermFrameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and p :: *name prm*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes $\Psi \rhd P \longmapsto ¡ M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and S: *set* $p \subseteq$ *set yvec* $\times$ *set zvec*
and *yvec* $\sharp *$ P
and *zvec* $\sharp *$ P

shows $(p \cdot \Psi) \rhd P \longmapsto ¡ (p \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *outputSwapSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *xvec* $\sharp *$ M
and $x \sharp P$
and $y \sharp P$
and $x \sharp \Psi$
and $y \sharp \Psi$

shows $\Psi \rhd P \longmapsto ([(x, y)] \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *broutputSwapSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto ¡ M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *xvec* $\sharp *$ M

and $x \sharp P$
and $y \sharp P$
and $x \sharp \Psi$
and $y \sharp \Psi$

shows $\Psi \rhd P \longmapsto ¡([(x, y)] \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *outputPermSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and p :: *name prm*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and S: *set* $p \subseteq$ *set yvec* $\times$ *set zvec*
and *yvec* $\sharp*$ P
and *zvec* $\sharp*$ P
and *yvec* $\sharp*$ Ψ
and *zvec* $\sharp*$ Ψ

shows $\Psi \rhd P \longmapsto (p \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *broutputPermSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and p :: *name prm*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and S: *set* $p \subseteq$ *set yvec* $\times$ *set zvec*
and *yvec* $\sharp*$ P
and *zvec* $\sharp*$ P
and *yvec* $\sharp*$ Ψ
and *zvec* $\sharp*$ Ψ

shows $\Psi \rhd P \longmapsto ¡(p \cdot M)(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *outputSwapFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*

and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and $x \sharp P$
and $y \sharp P$
and $x \sharp M$
and $y \sharp M$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *broutputSwapFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and x :: *name*
and y :: *name*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and $x \sharp P$
and $y \sharp P$
and $x \sharp M$
and $y \sharp M$

shows $([(x, y)] \cdot \Psi) \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *outputPermFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and p :: *name prm*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and S: *set p* $\subseteq$ *set yvec* $\times$ *set zvec*
and *yvec* $\sharp*$ P
and *zvec* $\sharp*$ P
and *yvec* $\sharp*$ M
and *zvec* $\sharp*$ M

shows $(p \cdot \Psi) \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *broutputPermFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and B :: $('a, 'b, 'c)$ *boundOutput*
and p :: *name prm*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes $\Psi \rhd P \longmapsto\!\!¡ M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and S: *set* $p \subseteq$ *set yvec* $\times$ *set zvec*
and *yvec* $\sharp*$ P
and *zvec* $\sharp*$ P
and *yvec* $\sharp*$ M
and *zvec* $\sharp*$ M

shows $(p \cdot \Psi) \rhd P \longmapsto\!\!¡ M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *Comm1*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$
and Q :: $('a, 'b, 'c)$ *psi*
and K :: $'a$
and *xvec* :: *name list*
and Q' :: $('a, 'b, 'c)$ *psi*
and A_Q :: *name list*

assumes $\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and $\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu * xvec|\!)\langle N\rangle \prec Q'$
and *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$
and $\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$
and $A_P \sharp* \Psi$
and $A_P \sharp* P$
and $A_P \sharp* Q$
and $A_P \sharp* M$
and $A_P \sharp* A_Q$
and $A_Q \sharp* \Psi$
and $A_Q \sharp* P$

and $A_Q \sharp* Q$
and $A_Q \sharp* K$
and $xvec \sharp* P$

shows $\Psi \rhd P \parallel Q \longmapsto \tau \prec (\!|\nu * xvec|\!)(P' \parallel Q')$
⟨*proof*⟩

lemma *Comm2*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$
and Q :: $('a, 'b, 'c)$ *psi*
and K :: $'a$
and Q' :: $('a, 'b, 'c)$ *psi*
and A_Q :: *name list*

assumes $\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and $\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q'$
and *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$
and $\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$
and $A_P \sharp* \Psi$
and $A_P \sharp* P$
and $A_P \sharp* Q$
and $A_P \sharp* M$
and $A_P \sharp* A_Q$
and $A_Q \sharp* \Psi$
and $A_Q \sharp* P$
and $A_Q \sharp* Q$
and $A_Q \sharp* K$
and $xvec \sharp* Q$

shows $\Psi \rhd P \parallel Q \longmapsto \tau \prec (\!|\nu * xvec|\!)(P' \parallel Q')$
⟨*proof*⟩

lemma *BrMerge*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*

and Ψ_P :: $'b$
and Q :: $('a, 'b, 'c)$ *psi*
and Q' :: $('a, 'b, 'c)$ *psi*
and A_Q :: *name list*

assumes $\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and $\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$
and *extractFrame* $Q = \langle A_Q, \Psi_Q \rangle$
and $A_P \sharp* \Psi$
and $A_P \sharp* P$
and $A_P \sharp* Q$
and $A_P \sharp* M$
and $A_P \sharp* A_Q$
and $A_Q \sharp* \Psi$
and $A_Q \sharp* P$
and $A_Q \sharp* Q$
and $A_Q \sharp* M$

shows $\Psi \rhd P \parallel Q \longmapsto ¿M(\!|N|\!) \prec (P' \parallel Q')$
⟨*proof*⟩

lemma *BrComm1*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$
and Q :: $('a, 'b, 'c)$ *psi*
and *xvec* :: *name list*
and Q' :: $('a, 'b, 'c)$ *psi*
and A_Q :: *name list*

assumes $\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and $\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N \rangle \prec Q'$
and *extractFrame* $Q = \langle A_Q, \Psi_Q \rangle$
and $A_P \sharp* \Psi$
and $A_P \sharp* P$
and $A_P \sharp* Q$
and $A_P \sharp* M$
and $A_P \sharp* A_Q$
and $A_Q \sharp* \Psi$
and $A_Q \sharp* P$
and $A_Q \sharp* Q$
and $A_Q \sharp* M$

and $xvec \sharp* P$

shows $\Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec (P' \parallel Q')$
⟨*proof*⟩

lemma *BrComm2*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$
and Q :: $('a, 'b, 'c)$ *psi*
and Q' :: $('a, 'b, 'c)$ *psi*
and A_Q :: *name list*

assumes $\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and $\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$
and *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$
and $A_P \sharp* \Psi$
and $A_P \sharp* P$
and $A_P \sharp* Q$
and $A_P \sharp* M$
and $A_P \sharp* A_Q$
and $A_Q \sharp* \Psi$
and $A_Q \sharp* P$
and $A_Q \sharp* Q$
and $A_Q \sharp* M$
and $xvec \sharp* Q$

shows $\Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec (P' \parallel Q')$
⟨*proof*⟩

lemma *BrClose*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and x :: *name*

assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $x \in supp\ M$
and $x \sharp \Psi$

shows $\Psi \rhd (\!|\nu x|\!)P \longmapsto \tau \prec (\!|\nu x|\!)((\!|\nu{*}xvec|\!)P')$
⟨*proof*⟩

lemma *semanticsCasesAux*[*consumes 1*, *case-names cInput cBrInput cOutput cBrOutput cCase cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2 cBrClose cOpen cBrOpen cScope cBang*]:
fixes cP :: ('a, 'b, 'c) *psi*
and cRs :: ('a, 'b, 'c) *residual*
and C :: 'f::*fs-name*
and x :: *name*

assumes $\Psi \rhd cP \longmapsto cRs$
and *rInput*: $\bigwedge M\ K\ xvec\ N\ Tvec\ P.\ [\![cP = M(\!|\lambda{*}xvec\ N|\!).P;\ cRs = K(\!|(N[xvec::=Tvec])|\!) \prec P[xvec::=Tvec];$
$\Psi \vdash M \leftrightarrow K;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;$
$xvec \sharp{*}\ Tvec;\ xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ M;\ xvec \sharp{*}\ K;$
$xvec \sharp{*}\ C]\!] \Longrightarrow Prop$
and *rBrInput*: $\bigwedge M\ K\ xvec\ N\ Tvec\ P.\ [\![cP = M(\!|\lambda{*}xvec\ N|\!).P;\ cRs = ¡K(\!|(N[xvec::=Tvec])|\!) \prec P[xvec::=Tvec];$
$\Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;$
$length\ xvec = length\ Tvec;$
$xvec \sharp{*}\ Tvec;\ xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ M;\ xvec \sharp{*}\ K;$
$xvec \sharp{*}\ C]\!] \Longrightarrow Prop$
and *rOutput*: $\bigwedge M\ K\ N\ P.\ [\![cP = M\langle N\rangle.P;\ cRs = K\langle N\rangle \prec P;\ \Psi \vdash M \leftrightarrow K]\!] \Longrightarrow Prop$
and *rBrOutput*: $\bigwedge M\ K\ N\ P.\ [\![cP = M\langle N\rangle.P;\ cRs = ¡K\langle N\rangle \prec P;\ \Psi \vdash M \preceq K]\!] \Longrightarrow Prop$
and *rCase*: $\bigwedge Cs\ P\ \varphi.\ [\![cP = Cases\ Cs;\ \Psi \rhd P \longmapsto cRs;\ (\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P]\!] \Longrightarrow Prop$

and *rPar1*: $\bigwedge \Psi_Q\ P\ \alpha\ P'\ Q\ A_Q.\ [\![cP = P \parallel Q;\ cRs = \alpha \prec (P' \parallel Q);$
$(\Psi \otimes \Psi_Q) \rhd P \longmapsto (\alpha \prec P');\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_Q \sharp{*}\ P;\ A_Q \sharp{*}\ Q;\ A_Q \sharp{*}\ \Psi;\ A_Q \sharp{*}\ \alpha;\ A_Q \sharp{*}\ C;\ A_Q \sharp{*}\ P';\ bn\ \alpha \sharp{*}\ \Psi;\ bn\ \alpha \sharp{*}\ \Psi_Q;$
$bn\ \alpha \sharp{*}\ Q;\ bn\ \alpha \sharp{*}\ P;\ bn\ \alpha \sharp{*}\ subject\ \alpha;\ bn\ \alpha \sharp{*}\ C;\ distinct(bn\ \alpha)]\!] \Longrightarrow$
$Prop$
and *rPar2*: $\bigwedge \Psi_P\ Q\ \alpha\ Q'\ P\ A_P.\ [\![cP = P \parallel Q;\ cRs = \alpha \prec (P \parallel Q');$
$(\Psi \otimes \Psi_P) \rhd Q \longmapsto \alpha \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$A_P \sharp{*}\ P;\ A_P \sharp{*}\ Q;\ A_P \sharp{*}\ \Psi;\ A_P \sharp{*}\ \alpha;\ A_P \sharp{*}\ C;$
$A_P \sharp{*}\ Q';\ bn\ \alpha \sharp{*}\ \Psi;\ bn\ \alpha \sharp{*}\ \Psi_P;\ bn\ \alpha \sharp{*}\ P;\ bn\ \alpha \sharp{*}\ Q;\ bn\ \alpha \sharp{*}\ subject\ \alpha;\ bn\ \alpha \sharp{*}\ C;\ distinct(bn\ \alpha)]\!] \Longrightarrow Prop$
and *rComm1*: $\bigwedge \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ K\ xvec\ Q'\ A_Q.$
$[\![cP = P \parallel Q;\ cRs = \tau \prec (\!|\nu{*}xvec|\!)P' \parallel Q';$
$\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;$

distinct A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu{*}xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; $A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} M$; $A_P \sharp{*} N$;
$A_P \sharp{*} P'$; $A_P \sharp{*} Q$; $A_P \sharp{*} Q'$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} xvec$; $A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$;
$A_Q \sharp{*} P$; $A_Q \sharp{*} K$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$; $A_Q \sharp{*} Q$; $A_Q \sharp{*} Q'$; $A_Q \sharp{*} xvec$;
$xvec \sharp{*} \Psi$; $xvec \sharp{*} \Psi_P$; $xvec \sharp{*} \Psi_Q$; $xvec \sharp{*} P$; $xvec \sharp{*} M$; $xvec \sharp{*} Q$;
$xvec \sharp{*} K$; $A_P \sharp{*} C$; $A_Q \sharp{*} C$; $xvec \sharp{*} C$; *distinct* $xvec$$]\!] \Longrightarrow$ *Prop*

and *rComm2*: $\bigwedge \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ K\ Q'\ A_Q$.
$[\![cP = P \parallel Q$; $cRs = \tau \prec (\!|\nu{*}xvec|\!)P' \parallel Q'$;
$\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; $A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} M$; $A_P \sharp{*} N$;
$A_P \sharp{*} P'$; $A_P \sharp{*} Q$; $A_P \sharp{*} Q'$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} xvec$; $A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$;
$A_Q \sharp{*} P$; $A_Q \sharp{*} K$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$; $A_Q \sharp{*} Q$; $A_Q \sharp{*} Q'$; $A_Q \sharp{*} xvec$;
$xvec \sharp{*} \Psi$; $xvec \sharp{*} \Psi_P$; $xvec \sharp{*} \Psi_Q$; $xvec \sharp{*} P$; $xvec \sharp{*} M$; $xvec \sharp{*} Q$;
$xvec \sharp{*} K$; $A_P \sharp{*} C$; $A_Q \sharp{*} C$; $xvec \sharp{*} C$; *distinct* $xvec$$]\!] \Longrightarrow$ *Prop*

and *rBrMerge*: $\bigwedge \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q$.
$[\![cP = (P \parallel Q)$; $cRs = ¿M(\!|N|\!) \prec (P' \parallel Q')$;
$\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} N$; $A_P \sharp{*} P'$;
$A_P \sharp{*} Q$; $A_P \sharp{*} Q'$; $A_P \sharp{*} M$; $A_P \sharp{*} A_Q$;
$A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$; $A_Q \sharp{*} P$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$;
$A_Q \sharp{*} Q$; $A_Q \sharp{*} Q'$; $A_Q \sharp{*} M$; $A_P \sharp{*} C$; $A_Q \sharp{*} C$$]\!] \Longrightarrow$ *Prop*

and *rBrComm1*: $\bigwedge \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ xvec\ Q'\ A_Q$.
$[\![cP = P \parallel Q$; $cRs = ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec (P' \parallel Q')$;
$\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} N$;
$A_P \sharp{*} P'$; $A_P \sharp{*} Q$; $A_P \sharp{*} Q'$; $A_P \sharp{*} M$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} xvec$; $A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$;
$A_Q \sharp{*} P$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$; $A_Q \sharp{*} M$; $A_Q \sharp{*} Q$; $A_Q \sharp{*} Q'$; $A_Q \sharp{*} xvec$;
$xvec \sharp{*} \Psi$; $xvec \sharp{*} \Psi_P$; $xvec \sharp{*} \Psi_Q$; $xvec \sharp{*} P$; $xvec \sharp{*} Q$; $xvec \sharp{*}$

M;
$A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$; $distinct\ xvec]\!] \Longrightarrow Prop$
and *rBrComm2*: $\bigwedge \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q$.
$[\![cP = P \parallel Q$; $cRs = ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec (P' \parallel Q')$;
$\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$; $extractFrame\ P = \langle A_P, \Psi_P\rangle$; $distinct\ A_P$;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; $extractFrame\ Q = \langle A_Q, \Psi_Q\rangle$; $distinct\ A_Q$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$;
$A_P \sharp* P'$; $A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* M$; $A_P \sharp* A_Q$; $A_P \sharp* xvec$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$;
$A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* M$; $A_Q \sharp* Q$; $A_Q \sharp* Q'$; $A_Q \sharp* xvec$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* \Psi_Q$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* M$;
$A_P \sharp* C$; $A_Q \sharp* C$; $xvec \sharp* C$; $distinct\ xvec]\!] \Longrightarrow Prop$
and *rBrClose*: $\bigwedge P\ M\ xvec\ N\ P'\ x$.
$[\![cP = (\!|\nu x|\!)P$; $cRs = \tau \prec (\!|\nu x|\!)((\!|\nu{*}xvec|\!)P')$;
$x \in supp\ M$;
$\Psi \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$;
$distinct\ xvec$; $xvec \sharp* \Psi$; $xvec \sharp* P$;
$xvec \sharp* M$;
$x \sharp \Psi$; $x \sharp xvec$;
$xvec \sharp* C$; $x \sharp C]\!] \Longrightarrow Prop$
and *rOpen*: $\bigwedge P\ M\ xvec\ yvec\ N\ P'\ x$.
$[\![cP = (\!|\nu x|\!)P$; $cRs = M(\!|\nu{*}(xvec@x\#yvec)|\!)\langle N\rangle \prec P'$;
$\Psi \rhd P \longmapsto M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle \prec P'$; $x \in supp\ N$; $x \sharp xvec$; $x \sharp yvec$; $x \sharp M$; $x \sharp \Psi$; $distinct\ xvec$; $distinct\ yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* yvec$; $yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $xvec \sharp* C$; $x \sharp C$; $yvec \sharp* C]\!] \Longrightarrow$
$Prop$
and *rBrOpen*: $\bigwedge P\ M\ xvec\ yvec\ N\ P'\ x$.
$[\![cP = (\!|\nu x|\!)P$; $cRs = ¡M(\!|\nu{*}(xvec@x\#yvec)|\!)\langle N\rangle \prec P'$;
$\Psi \rhd P \longmapsto ¡M(\!|\nu{*}(xvec@yvec)|\!)\langle N\rangle \prec P'$; $x \in supp\ N$; $x \sharp xvec$; $x \sharp yvec$; $x \sharp M$; $x \sharp \Psi$; $distinct\ xvec$; $distinct\ yvec$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* yvec$; $yvec \sharp* \Psi$; $yvec \sharp* P$; $yvec \sharp* M$; $xvec \sharp* C$; $x \sharp C$; $yvec \sharp* C]\!] \Longrightarrow$
$Prop$
and *rScope*: $\bigwedge P\ \alpha\ P'\ x$. $[\![cP = (\!|\nu x|\!)P$; $cRs = \alpha \prec (\!|\nu x|\!)P'$;
$\Psi \rhd P \longmapsto \alpha \prec P'$; $x \sharp \Psi$; $x \sharp \alpha$; $x \sharp C$; $bn\ \alpha \sharp* \Psi$; $bn\ \alpha \sharp* P$; $bn\ \alpha \sharp* subject\ \alpha$; $bn\ \alpha \sharp* C$; $distinct(bn\ \alpha)]\!] \Longrightarrow Prop$
and *rBang*: $\bigwedge P$. $[\![cP = !P$; $\Psi \rhd P \parallel !P \longmapsto cRs$; $guarded\ P]\!] \Longrightarrow Prop$
shows *Prop*
⟨*proof*⟩

nominal-primrec
inputLength :: ('*a*::*fs-name*, '*b*::*fs-name*, '*c*::*fs-name*) *psi* ⇒ *nat*
and *inputLength'* :: ('*a*::*fs-name*, '*b*::*fs-name*, '*c*::*fs-name*) *input* ⇒ *nat*
and *inputLength''* :: ('*a*::*fs-name*, '*b*::*fs-name*, '*c*::*fs-name*) *psiCase* ⇒ *nat*

where
  $inputLength\ (\mathbf{0}) = 0$
| $inputLength\ (M\langle N\rangle.P) = 0$
| $inputLength\ (M(\!|I) = inputLength'\ I$
| $inputLength\ (Case\ C) = 0$
| $inputLength\ (P \parallel Q) = 0$
| $inputLength\ ((\!|\nu x|\!)P) = 0$
| $inputLength\ (\{\!|\Psi|\!\}) = 0$
| $inputLength\ (!P) = 0$

| $inputLength'\ (Trm\ M\ P) = 0$
| $inputLength'\ (\nu\ y\ I) = 1 + (inputLength'\ I)$

| $inputLength''\ (\bot_c) = 0$
| $inputLength''\ (\Box\Phi \Rightarrow P\ C) = 0$
  $\langle proof\rangle$

nominal-primrec $boundOutputLength :: ('a, 'b, 'c)\ boundOutput \Rightarrow nat$
  where
    $boundOutputLength\ (BOut\ M\ P) = 0$
  | $boundOutputLength\ (BStep\ x\ B) = (boundOutputLength\ B) + 1$
      $\langle proof\rangle$

nominal-primrec $residualLength :: ('a, 'b, 'c)\ residual \Rightarrow nat$
  where
    $residualLength\ (RIn\ M\ N\ P) = 0$
  | $residualLength\ (RBrIn\ M\ N\ P) = 0$
  | $residualLength\ (ROut\ M\ B) = boundOutputLength\ B$
  | $residualLength\ (RBrOut\ M\ B) = boundOutputLength\ B$
  | $residualLength\ (RTau\ P) = 0$
  $\langle proof\rangle$

lemma $inputLengthProc[simp]$:
  shows $inputLength(M(\!|\lambda*xvec\ N|\!).P) = length\ xvec$
  $\langle proof\rangle$

lemma $boundOutputLengthSimp[simp]$:
  shows $residualLength(M(\!|\nu*xvec|\!)\langle N\rangle \prec P) = length\ xvec$
    and $residualLength(¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P) = length\ xvec$
  $\langle proof\rangle$

lemma $boundOuputLengthSimp2[simp]$:
  shows $residualLength(\alpha \prec P) = length(bn\ \alpha)$
  $\langle proof\rangle$

lemmas $[simp\ del] = inputLength\text{-}inputLength'\text{-}inputLength''.simps\ residualLength.simps$
$boundOutputLength.simps$

lemma *constructPerm*:
fixes *xvec* :: *name list*
and *yvec* :: *name list*

assumes *length xvec* = *length yvec*
and *xvec* ♯∗ *yvec*
and *distinct xvec*
and *distinct yvec*

obtains *p* **where** *set p* ⊆ *set xvec* × *set*(*p* ⋅ *xvec*) **and** *distinctPerm p* **and** *yvec* = *p* ⋅ *xvec*
⟨*proof*⟩

lemma *distinctApend*[*simp*]:
fixes *xvec* :: *name list*
and *yvec* :: *name list*

shows (*set xvec* ∩ *set yvec* = {}) = *xvec* ♯∗ *yvec*
⟨*proof*⟩

lemma *lengthAux*:
fixes *xvec* :: *name list*
and *y* :: *name*
and *yvec* :: *name list*

assumes *length xvec* = *length*(*y*#*yvec*)

obtains *z zvec* **where** *xvec* = *z*#*zvec* **and** *length zvec* = *length yvec*
⟨*proof*⟩

lemma *lengthAux2*:
fixes *xvec* :: *name list*
and *yvec* :: *name list*
and *zvec* :: *name list*

assumes *length xvec* = *length*(*yvec*@*y*#*zvec*)

obtains *xvec1 x xvec2* **where** *xvec*=*xvec1*@*x*#*xvec2* **and** *length xvec1* = *length yvec* **and** *length xvec2* = *length zvec*
⟨*proof*⟩

lemma *semanticsCases*[*consumes 19*, *case-names cInput cBrInput cOutput cBrOutput cCase cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2 cBrClose cOpen cBrOpen cScope cBang*]:
fixes Ψ :: *'b*
and *cP* :: (*'a*, *'b*, *'c*) *psi*
and *cRs* :: (*'a*, *'b*, *'c*) *residual*
and *C* :: *'f*::*fs-name*
and *x1* :: *name*

and $x2$ :: *name*
and $x3$ :: *name*
and $x4$ :: *name*
and $xvec1$ :: *name list*
and $xvec2$ :: *name list*
and $xvec3$ :: *name list*
and $xvec4$ :: *name list*
and $xvec5$ :: *name list*
and $xvec6$ :: *name list*
and $xvec7$ :: *name list*
and $xvec8$ :: *name list*
and $xvec9$ :: *name list*

assumes $\Psi \rhd cP \longmapsto cRs$
and *length xvec1* = *inputLength cP* **and** *distinct xvec1*
and *length xvec6* = *inputLength cP* **and** *distinct xvec6*
and *length xvec2* = *residualLength cRs* **and** *distinct xvec2*
and *length xvec3* = *residualLength cRs* **and** *distinct xvec3*
and *length xvec4* = *residualLength cRs* **and** *distinct xvec4*
and *length xvec5* = *residualLength cRs* **and** *distinct xvec5*
and *length xvec7* = *residualLength cRs* **and** *distinct xvec7*
and *length xvec8* = *residualLength cRs* **and** *distinct xvec8*
and *length xvec9* = *residualLength cRs* **and** *distinct xvec9*
and *rInput*: $\bigwedge M\ K\ N\ Tvec\ P.\ (\llbracket xvec1 \sharp* \Psi;\ xvec1 \sharp* cP;\ xvec1 \sharp* cRs\rrbracket \Longrightarrow cP = M(\!|\lambda* xvec1\ N|\!).P \wedge\ cRs = K(\!|(N[xvec1::=Tvec])|\!) \prec P[xvec1::=Tvec] \wedge \Psi \vdash M \leftrightarrow K \wedge distinct\ xvec1 \wedge set\ xvec1 \subseteq supp\ N \wedge length\ xvec1 = length\ Tvec \wedge xvec1 \sharp* Tvec \wedge xvec1 \sharp* \Psi \wedge xvec1 \sharp* M \wedge xvec1 \sharp* K) \Longrightarrow Prop$
and *rBrInput*: $\bigwedge M\ K\ N\ Tvec\ P.\ (\llbracket xvec6 \sharp* \Psi;\ xvec6 \sharp* cP;\ xvec6 \sharp* cRs\rrbracket \Longrightarrow cP = M(\!|\lambda* xvec6\ N|\!).P \wedge\ cRs = ¿K(\!|(N[xvec6::=Tvec])|\!) \prec P[xvec6::=Tvec] \wedge \Psi \vdash K \succeq M \wedge distinct\ xvec6 \wedge set\ xvec6 \subseteq supp\ N \wedge length\ xvec6 = length\ Tvec \wedge xvec6 \sharp* Tvec \wedge xvec6 \sharp* \Psi \wedge xvec6 \sharp* M \wedge xvec6 \sharp* K) \Longrightarrow Prop$
and *rOutput*: $\bigwedge M\ K\ N\ P.\ \llbracket cP = M\langle N\rangle.P;\ cRs = K\langle N\rangle \prec P;\ \Psi \vdash M \leftrightarrow K\rrbracket \Longrightarrow Prop$
and *rBrOutput*: $\bigwedge M\ K\ N\ P.\ \llbracket cP = M\langle N\rangle.P;\ cRs = ¡K\langle N\rangle \prec P;\ \Psi \vdash M \preceq K\rrbracket \Longrightarrow Prop$
and *rCase*: $\bigwedge Cs\ P\ \varphi.\ \llbracket cP = Cases\ Cs;\ \Psi \rhd P \longmapsto cRs;\ (\varphi,\ P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P\rrbracket \Longrightarrow Prop$
and *rPar1*: $\bigwedge \Psi_Q\ P\ \alpha\ P'\ Q\ A_Q.\ (\llbracket xvec2 \sharp* \Psi;\ xvec2 \sharp* cP;\ xvec2 \sharp* cRs\rrbracket \Longrightarrow cP = P \parallel Q \wedge cRs = \alpha \prec (P' \parallel Q) \wedge xvec2 = bn\ \alpha \wedge \Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P' \wedge extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle \wedge distinct\ A_Q \wedge A_Q \sharp* P \wedge A_Q \sharp* Q \wedge A_Q \sharp* \Psi \wedge A_Q \sharp* \alpha \wedge A_Q \sharp* P' \wedge A_Q \sharp* C) \Longrightarrow Prop$
and *rPar2*: $\bigwedge \Psi_P\ Q\ \alpha\ Q'\ P\ A_P.\ (\llbracket xvec3 \sharp* \Psi;\ xvec3 \sharp* cP;\ xvec3 \sharp* cRs\rrbracket \Longrightarrow$

$cP = P \parallel Q \wedge cRs = \alpha \prec (P \parallel Q') \wedge xvec3 = bn\ \alpha \wedge$
$\Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q' \wedge extractFrame\ P = \langle A_P, \Psi_P \rangle \wedge distinct\ A_P \wedge$
$A_P \sharp* P \wedge A_P \sharp* Q \wedge A_P \sharp* \Psi \wedge A_P \sharp* \alpha \wedge A_P \sharp* Q' \wedge A_P \sharp* C) \Longrightarrow Prop$

and *rComm1*: $\bigwedge \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ K\ xvec\ Q'\ A_Q.$
$[\![cP = P \parallel Q;\ cRs = \tau \prec (\!|\nu * xvec|\!) P' \parallel Q';$
$\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P \rangle;$ *distinct* $A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu * xvec|\!)\langle N \rangle \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q \rangle;$ *distinct* $A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;\ A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;$
$A_P \sharp* P';\ A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* xvec;\ A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;$
$A_Q \sharp* P;\ A_Q \sharp* K;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* Q;\ A_Q \sharp* Q';\ A_Q \sharp* xvec;$
$xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* Q;$
$xvec \sharp* K;\ A_P \sharp* C;\ A_Q \sharp* C;\ xvec \sharp* C;\ distinct\ xvec] \Longrightarrow Prop$

and *rComm2*: $\bigwedge \Psi_Q\ P\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q\ K\ Q'\ A_Q.$
$[\![cP = P \parallel Q;\ cRs = \tau \prec (\!|\nu * xvec|\!) P' \parallel Q';$
$\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N \rangle \prec P';\ extractFrame\ P = \langle A_P, \Psi_P \rangle;$ *distinct* $A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q \rangle;$ *distinct* $A_Q;$
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;\ A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* M;\ A_P \sharp* N;$
$A_P \sharp* P';\ A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* xvec;\ A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;$
$A_Q \sharp* P;\ A_Q \sharp* K;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* Q;\ A_Q \sharp* Q';\ A_Q \sharp* xvec;$
$xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* \Psi_Q;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* Q;$
$xvec \sharp* K;\ A_P \sharp* C;\ A_Q \sharp* C;\ xvec \sharp* C;\ distinct\ xvec] \Longrightarrow Prop$

and *rBrMerge*: $\bigwedge \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q.$
$[\![cP = (P \parallel Q);\ cRs = ¿M(\!|N|\!) \prec (P' \parallel Q');$
$\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P \rangle;$ *distinct* $A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q \rangle;$ *distinct* $A_Q;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* M;\ A_Q \sharp* M;$
$A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;\ A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';$
$A_Q \sharp* Q;\ A_Q \sharp* Q';\ A_P \sharp* C;\ A_Q \sharp* C] \Longrightarrow Prop$

and *rBrComm1*: $\bigwedge \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q.$
$([\![xvec7 \sharp* \Psi;\ xvec7 \sharp* cP;\ xvec7 \sharp* cRs]\!] \Longrightarrow$
$cP = P \parallel Q \wedge cRs = ¡M(\!|\nu * xvec7|\!)\langle N \rangle \prec (P' \parallel Q') \wedge$

$\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P' \wedge \mathit{extractFrame}\ P = \langle A_P, \Psi_P\rangle \wedge$
$\mathit{distinct}\ A_P \wedge$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu * \mathit{xvec7}|\!)\langle N\rangle \prec Q' \wedge \mathit{extractFrame}\ Q = \langle A_Q, \Psi_Q\rangle \wedge \mathit{distinct}\ A_Q \wedge$
$A_P \sharp* \Psi \wedge A_P \sharp* \Psi_Q \wedge A_P \sharp* P \wedge A_P \sharp* N \wedge$
$A_P \sharp* P' \wedge A_P \sharp* Q \wedge A_P \sharp* Q' \wedge A_P \sharp* A_Q \wedge A_P \sharp* \mathit{xvec7} \wedge A_Q \sharp* \Psi \wedge A_Q \sharp* \Psi_P \wedge$
$A_Q \sharp* P \wedge A_Q \sharp* N \wedge A_Q \sharp* P' \wedge A_Q \sharp* Q \wedge A_Q \sharp* Q' \wedge A_Q \sharp* \mathit{xvec7} \wedge$
$\mathit{xvec7} \sharp* \Psi \wedge \mathit{xvec7} \sharp* \Psi_P \wedge \mathit{xvec7} \sharp* \Psi_Q \wedge \mathit{xvec7} \sharp* P \wedge \mathit{xvec7} \sharp* Q \wedge$
$A_P \sharp* M \wedge A_Q \sharp* M \wedge \mathit{xvec7} \sharp* M \wedge$
$A_P \sharp* C \wedge A_Q \sharp* C \wedge \mathit{distinct}\ \mathit{xvec7}) \Longrightarrow \mathit{Prop}$

and *rBrComm2*: $\bigwedge \Psi_Q\ P\ M\ N\ P'\ A_P\ \Psi_P\ Q\ Q'\ A_Q.$
$(\!|\mathit{xvec8} \sharp* \Psi;\ \mathit{xvec8} \sharp* cP;\ \mathit{xvec8} \sharp* cRs|\!] \Longrightarrow$
$cP = P \parallel Q \wedge cRs = ¡M(\!|\nu * \mathit{xvec8}|\!)\langle N\rangle \prec (P' \parallel Q') \wedge$
$\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu * \mathit{xvec8}|\!)\langle N\rangle \prec P' \wedge \mathit{extractFrame}\ P = \langle A_P, \Psi_P\rangle \wedge \mathit{distinct}\ A_P \wedge$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q' \wedge \mathit{extractFrame}\ Q = \langle A_Q, \Psi_Q\rangle \wedge \mathit{distinct}\ A_Q \wedge$
$A_P \sharp* \Psi \wedge A_P \sharp* \Psi_Q \wedge A_P \sharp* P \wedge A_P \sharp* N \wedge$
$A_P \sharp* P' \wedge A_P \sharp* Q \wedge A_P \sharp* Q' \wedge A_P \sharp* A_Q \wedge A_P \sharp* \mathit{xvec8} \wedge A_Q \sharp* \Psi \wedge A_Q \sharp* \Psi_P \wedge$
$A_Q \sharp* P \wedge A_Q \sharp* N \wedge A_Q \sharp* P' \wedge A_Q \sharp* Q \wedge A_Q \sharp* Q' \wedge A_Q \sharp* \mathit{xvec8} \wedge$
$\mathit{xvec8} \sharp* \Psi \wedge \mathit{xvec8} \sharp* \Psi_P \wedge \mathit{xvec8} \sharp* \Psi_Q \wedge \mathit{xvec8} \sharp* P \wedge \mathit{xvec8} \sharp* Q \wedge$
$A_P \sharp* M \wedge A_Q \sharp* M \wedge \mathit{xvec8} \sharp* M \wedge$
$A_P \sharp* C \wedge A_Q \sharp* C \wedge \mathit{distinct}\ \mathit{xvec8}) \Longrightarrow \mathit{Prop}$

and *rBrClose*: $\bigwedge P\ M\ N\ \mathit{xvec}\ P'.$
$(\!|\mathit{x3} \sharp \Psi;\ \mathit{x3} \sharp cP;\ \mathit{x3} \sharp cRs|\!] \Longrightarrow$
$cP = (\!|\nu \mathit{x3}|\!)P) \wedge cRs = \tau \prec (\!|\nu \mathit{x3}|\!)((\!|\nu * \mathit{xvec}|\!)P') \wedge$
$\mathit{x3} \in \mathit{supp}\ M \wedge$
$\Psi \rhd P \longmapsto ¡M(\!|\nu * \mathit{xvec}|\!)\langle N\rangle \prec P' \wedge$
$\mathit{distinct}\ \mathit{xvec} \wedge \mathit{xvec} \sharp* \Psi \wedge \mathit{xvec} \sharp* P \wedge$
$\mathit{xvec} \sharp* M \wedge \mathit{xvec} \sharp* C \wedge$
$\mathit{x3} \sharp \Psi \wedge \mathit{x3} \sharp \mathit{xvec}) \Longrightarrow \mathit{Prop}$

and *rOpen*: $\bigwedge P\ M\ \mathit{xvec}\ y\ \mathit{yvec}\ N\ P'.$
$(\!|\mathit{xvec4} \sharp* \Psi;\ \mathit{xvec4} \sharp* cP;\ \mathit{xvec4} \sharp* cRs;\ \mathit{x1} \sharp \Psi;\ \mathit{x1} \sharp cP;\ \mathit{x1} \sharp cRs;\ \mathit{x1} \sharp \mathit{xvec4}|\!] \Longrightarrow$
$cP = (\!|\nu \mathit{x1}|\!)P \wedge cRs = M(\!|\nu * (\mathit{xvec}@\mathit{x1}\#\mathit{yvec})|\!)\langle N\rangle \prec P' \wedge \mathit{xvec4} = \mathit{xvec}@y\#\mathit{yvec} \wedge$
$\Psi \rhd P \longmapsto M(\!|\nu * (\mathit{xvec}@\mathit{yvec})|\!)\langle N\rangle \prec P' \wedge \mathit{x1} \in \mathit{supp}\ N \wedge \mathit{x1} \sharp \mathit{xvec} \wedge \mathit{x1} \sharp \mathit{yvec} \wedge$
$\mathit{distinct}\ \mathit{xvec} \wedge \mathit{distinct}\ \mathit{yvec} \wedge \mathit{xvec} \sharp* \Psi \wedge \mathit{xvec} \sharp* P \wedge \mathit{xvec} \sharp* M \wedge \mathit{xvec} \sharp* \mathit{yvec} \wedge$
$\mathit{yvec} \sharp* \Psi \wedge \mathit{yvec} \sharp* P \wedge \mathit{yvec} \sharp* M) \Longrightarrow \mathit{Prop}$

and *rBrOpen*: $\bigwedge P\ M\ \mathit{xvec}\ y\ \mathit{yvec}\ N\ P'.$
$(\!|\mathit{xvec9} \sharp* \Psi;\ \mathit{xvec9} \sharp* cP;\ \mathit{xvec9} \sharp* cRs;\ \mathit{x4} \sharp \Psi;\ \mathit{x4} \sharp cP;\ \mathit{x4} \sharp$

cRs; $x4 \sharp xvec9$$\rrbracket \Longrightarrow$
$cP = (\!|\nu x4|\!)P \wedge cRs = {¡}M(\!|\nu*(xvec@x4\#yvec)|\!)\langle N\rangle \prec P' \wedge xvec9{=}xvec@y\#yvec \wedge$
$\Psi \rhd P \longmapsto {¡}M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P' \wedge x4 \in supp\ N \wedge x4 \sharp xvec \wedge x4 \sharp yvec \wedge$
$distinct\ xvec \wedge distinct\ yvec \wedge xvec \sharp* \Psi \wedge xvec \sharp* P \wedge xvec \sharp* M \wedge xvec \sharp* yvec \wedge$
$yvec \sharp* \Psi \wedge yvec \sharp* P \wedge yvec \sharp* M) \Longrightarrow Prop$
and *rScope*: $\bigwedge P\ \alpha\ P'.$ ($\llbracket xvec5 \sharp* \Psi$; $xvec5 \sharp* cP$; $xvec5 \sharp* cRs$; $x2 \sharp \Psi$; $x2 \sharp cP$; $x2 \sharp cRs$; $x2 \sharp xvec5\rrbracket \Longrightarrow$
$cP = (\!|\nu x2|\!)P \wedge cRs = \alpha \prec (\!|\nu x2|\!)P' \wedge\ xvec5 = bn\ \alpha \wedge$
$\Psi \rhd P \longmapsto \alpha \prec P' \wedge x2 \sharp \Psi \wedge x2 \sharp \alpha \wedge bn\ \alpha \sharp* subject\ \alpha \wedge distinct(bn\ \alpha)) \Longrightarrow Prop$
and *rBang*: $\bigwedge P.\ \llbracket cP = !P;$
$\Psi \rhd P \parallel !P \longmapsto cRs;\ guarded\ P\rrbracket \Longrightarrow Prop$
shows *Prop*
⟨*proof*⟩

lemma *resResidEq*:
fixes *xvec* :: *name list*
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and N' :: $'a$

assumes ${¡}M(\!|\nu*xvec|\!)\langle N\rangle \prec P = {¡}M(\!|\nu*xvec|\!)\langle N'\rangle \prec Q$
and $xvec \sharp* M$

shows ${¡}M(\!|\nu*xvec|\!)\langle N\rangle = {¡}M(\!|\nu*xvec|\!)\langle N'\rangle$
⟨*proof*⟩

lemma *parCases*[*consumes 5*, *case-names cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and T :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*
and M :: $'a$
and N :: $'a$

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto \alpha \prec T$
and $bn\ \alpha \sharp* \Psi$
and $bn\ \alpha \sharp* P$
and $bn\ \alpha \sharp* Q$
and $bn\ \alpha \sharp* subject\ \alpha$
and *rPar1*: $\bigwedge P'\ A_Q\ \Psi_Q.\ \llbracket \Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P';$ *extractFrame* $Q = \langle A_Q,$

$\Psi_Q\rangle$; *distinct* A_Q;
$A_Q \sharp* \Psi$; $A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* \alpha$; $A_Q \sharp* P'$; $A_Q \sharp* C]\!] \Longrightarrow$ *Prop* α $(P' \parallel Q)$

and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P$. $[\![\Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* \alpha$; $A_P \sharp* Q'$; $A_P \sharp* C]\!] \Longrightarrow$ *Prop* α $(P \parallel Q')$

and *rComm1*: $\bigwedge \Psi_Q\ M\ N\ P'\ A_P\ \Psi_P\ K\ xvec\ Q'\ A_Q$.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu* xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; *distinct xvec*; $\alpha = \tau$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* Q$; $A_P \sharp* xvec$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* C$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* K$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* xvec$; $A_Q \sharp* Q'$; $A_Q \sharp* C$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* K$; $xvec \sharp* Q$; $xvec \sharp* \Psi_Q$; $xvec \sharp* C]\!] \Longrightarrow$
Prop (τ) $((\!|\nu* xvec|\!)(P' \parallel Q'))$

and *rComm2*: $\bigwedge \Psi_Q\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ K\ Q'\ A_Q$.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu* xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; *distinct xvec*; $\alpha = \tau$;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* Q$; $A_P \sharp* xvec$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* C$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* K$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* xvec$; $A_Q \sharp* Q'$; $A_Q \sharp* C$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* K$; $xvec \sharp* Q$; $xvec \sharp* \Psi_Q$; $xvec \sharp* C]\!] \Longrightarrow$
Prop (τ) $((\!|\nu* xvec|\!)(P' \parallel Q'))$

and *rBrMerge*: $\bigwedge \Psi_Q\ M\ N\ P'\ A_P\ \Psi_P\ Q'\ A_Q$.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$;
$A_P \sharp* Q$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* M$; $A_Q \sharp* M$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$;
$A_Q \sharp* Q$; $A_Q \sharp* Q'$; $A_P \sharp* C$; $A_Q \sharp* C$; $\alpha = ¿M(\!|N|\!)]\!] \Longrightarrow$
Prop $(¿M(\!|N|\!))$ $(P' \parallel Q')$

and *rBrComm1*: $\bigwedge \Psi_Q\ M\ N\ P'\ A_P\ \Psi_P\ xvec\ Q'\ A_Q$.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu* xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
distinct xvec;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* Q$; $A_P \sharp* xvec$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* C$;

$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* xvec$; $A_Q \sharp* Q'$; $A_Q \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$; ¡$M\langle\!|\nu*xvec|\!\rangle\langle N\rangle = \alpha$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* \Psi_Q$⟧ ⟹
Prop (¡$M\langle\!|\nu*xvec|\!\rangle\langle N\rangle$) ($P' \parallel Q'$)

and *rBrComm2*: $\bigwedge \Psi_Q\ M\ xvec\ N\ P'\ A_P\ \Psi_P\ Q'\ A_Q$.
⟦$\Psi \otimes \Psi_Q \rhd P \longmapsto$¡$M\langle\!|\nu*xvec|\!\rangle\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto$¿$M\langle\!|N|\!\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
distinct xvec;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* Q$; $A_P \sharp* xvec$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* C$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* xvec$; $A_Q \sharp* Q'$; $A_Q \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$; ¡$M\langle\!|\nu*xvec|\!\rangle\langle N\rangle = \alpha$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* \Psi_Q$⟧ ⟹
Prop (¡$M\langle\!|\nu*xvec|\!\rangle\langle N\rangle$) ($P' \parallel Q'$)

shows *Prop* α T
⟨*proof*⟩

lemma *parInputCases*[*consumes 1*, *case-names cPar1 cPar2*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and Q :: (*'a*, *'b*, *'c*) *psi*
and M :: *'a*
and N :: *'a*
and R :: (*'a*, *'b*, *'c*) *psi*
and C :: *'f*::*fs-name*

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto M\langle\!|N|\!\rangle \prec R$
and *rPar1*: $\bigwedge P'\ A_Q\ \Psi_Q$. ⟦$\Psi \otimes \Psi_Q \rhd P \longmapsto M\langle\!|N|\!\rangle \prec P'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_Q \sharp* \Psi$; $A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* M$; $A_Q \sharp* N$; $A_Q \sharp* C$⟧ ⟹ *Prop* ($P' \parallel Q$)
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P$. ⟦$\Psi \otimes \Psi_P \rhd Q \longmapsto M\langle\!|N|\!\rangle \prec Q'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* M$; $A_P \sharp* N$; $A_P \sharp* C$⟧ ⟹ *Prop* ($P \parallel Q'$)
shows *Prop* R
⟨*proof*⟩

lemma *parBrInputCases*[*consumes 1*, *case-names cPar1 cPar2 cBrMerge*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and Q :: (*'a*, *'b*, *'c*) *psi*
and M :: *'a*
and N :: *'a*

and $R :: ('a, 'b, 'c)\ psi$
and $C :: 'f::fs\text{-}name$

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto ¿M(\!|N|\!) \prec R$
and *rPar1*: $\bigwedge P'\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;\ A_Q \sharp* N;\ A_Q \sharp* C]\!] \Longrightarrow Prop\ (P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P.\ [\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_P \sharp* N;\ A_P \sharp* C]\!] \Longrightarrow Prop\ (P \parallel Q')$
and *rBrMerge*: $\bigwedge \Psi_Q\ P'\ A_P\ \Psi_P\ Q'\ A_Q.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* N;\ A_P \sharp* P';$
$A_P \sharp* Q;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* M;\ A_Q \sharp* M;$
$A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;\ A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';$
$A_Q \sharp* Q;\ A_Q \sharp* Q';\ A_P \sharp* C;\ A_Q \sharp* C]\!] \Longrightarrow$
$Prop\ (P' \parallel Q')$
shows *Prop R*
⟨*proof*⟩

lemma *parOutputCases*[*consumes 5*, *case-names cPar1 cPar2*]:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $M :: 'a$
and $xvec :: name\ list$
and $N :: 'a$
and $R :: ('a, 'b, 'c)\ psi$
and $C :: 'f::fs\text{-}name$

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec R$
and $xvec \sharp* \Psi$
and $xvec \sharp* P$
and $xvec \sharp* Q$
and $xvec \sharp* M$
and *rPar1*: $\bigwedge P'\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;\ A_Q \sharp* xvec;\ A_Q \sharp* N;\ A_Q \sharp* C;\ A_Q \sharp* xvec;\ distinct\ xvec]\!] \Longrightarrow Prop\ (P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P.\ [\![\Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_P \sharp* xvec;\ A_P \sharp* N;\ A_P \sharp* C;\ A_P \sharp* xvec;\ distinct\ xvec]\!] \Longrightarrow Prop\ (P \parallel Q')$
shows *Prop R*

⟨*proof*⟩

lemma *parBrOutputCases*[*consumes 5*, *case-names cPar1 cPar2 cBrComm1 cBrComm2*]:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $M :: 'a$
and $xvec :: name\ list$
and $N :: 'a$
and $R :: ('a, 'b, 'c)\ psi$
and $C :: 'f::fs\text{-}name$

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec R$
and $xvec \sharp* \Psi$
and $xvec \sharp* P$
and $xvec \sharp* Q$
and $xvec \sharp* M$
and *rPar1*: $\bigwedge P'\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;\ A_Q \sharp* xvec;\ A_Q \sharp* N;\ A_Q \sharp* C;\ A_Q \sharp* xvec;\ distinct\ xvec]\!] \Longrightarrow Prop\ (P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P.\ [\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_P \sharp* xvec;\ A_P \sharp* N;\ A_P \sharp* C;\ A_P \sharp* xvec;\ distinct\ xvec]\!] \Longrightarrow Prop\ (P \parallel Q')$
and *rBrComm1*: $\bigwedge \Psi_Q\ P'\ A_P\ \Psi_P\ Q'\ A_Q.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$distinct\ xvec;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* N;\ A_P \sharp* P';\ A_P \sharp* Q;\ A_P \sharp* xvec;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* C;$
$A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;\ A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* Q;\ A_Q \sharp* xvec;\ A_Q \sharp* Q';\ A_Q \sharp* C;$
$A_P \sharp* M;\ A_Q \sharp* M;\ xvec \sharp* M;$
$xvec \sharp* \Psi;\ xvec \sharp* \Psi_P;\ xvec \sharp* P;\ xvec \sharp* Q;\ xvec \sharp* \Psi_Q]\!] \Longrightarrow$
$Prop\ (P' \parallel Q')$
and *rBrComm2*: $\bigwedge \Psi_Q\ P'\ A_P\ \Psi_P\ Q'\ A_Q.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;\ distinct\ A_Q;$
$distinct\ xvec;$
$A_P \sharp* \Psi;\ A_P \sharp* \Psi_Q;\ A_P \sharp* P;\ A_P \sharp* N;\ A_P \sharp* P';\ A_P \sharp* Q;\ A_P \sharp* xvec;\ A_P \sharp* Q';\ A_P \sharp* A_Q;\ A_P \sharp* C;$
$A_Q \sharp* \Psi;\ A_Q \sharp* \Psi_P;\ A_Q \sharp* P;\ A_Q \sharp* N;\ A_Q \sharp* P';\ A_Q \sharp* Q;\ A_Q \sharp* xvec;\ A_Q \sharp* Q';\ A_Q \sharp* C;$

$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* \Psi_Q]\!] \Longrightarrow$
$Prop\ (P' \parallel Q')$
shows $Prop\ R$
⟨*proof*⟩

lemma *theEqvt*[*eqvt-force*]:
fixes $p :: name\ prm$
and $\alpha :: 'a\ action$

assumes $\alpha \neq \tau$

shows $(p \cdot the(subject\ \alpha)) = the(p \cdot (subject\ \alpha))$
⟨*proof*⟩

lemma *theSubjectFresh*[*simp*]:
fixes $\alpha :: 'a\ action$
and $x :: name$

assumes $\alpha \neq \tau$

shows $x \sharp the(subject\ \alpha) = x \sharp subject\ \alpha$
⟨*proof*⟩

lemma *theSubjectFreshChain*[*simp*]:
fixes α :: $'a\ action$
and $xvec :: name\ list$

assumes $\alpha \neq \tau$

shows $xvec \sharp* the(subject\ \alpha) = xvec \sharp* subject\ \alpha$
⟨*proof*⟩

lemma *inputObtainPrefix*:
fixes $\Psi :: 'b$
and P :: $('a, 'b, 'c)\ psi$
and P' :: $('a, 'b, 'c)\ psi$
and A_P :: $name\ list$
and $\Psi_P :: 'b$
and N :: $'a$
and K :: $'a$
and B :: $name\ list$

assumes $\Psi \rhd P \longmapsto K(\!|N|\!) \prec P'$
and $extractFrame\ P = \langle A_P, \Psi_P \rangle$
and $distinct\ A_P$
and $B \sharp* P$
and $A_P \sharp* \Psi$
and $A_P \sharp* B$

and $A_P \sharp* P$
and $A_P \sharp* K$

obtains M **where** $\Psi \otimes \Psi_P \vdash K \leftrightarrow M$ **and** $B \sharp* M$
⟨*proof*⟩

lemma *outputObtainPrefix*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $P' :: ('a, 'b, 'c)\ psi$
and $A_P :: name\ list$
and $\Psi_P :: 'b$
and $N :: 'a$
and $K :: 'a$
and $xvec :: name\ list$
and $B :: name\ list$

assumes $\Psi \rhd P \longmapsto ROut\ K\ (\langle\!|\nu * xvec|\!\rangle N \prec' P')$
and $extractFrame\ P = \langle A_P, \Psi_P\rangle$
and $distinct\ A_P$
and $xvec \sharp* K$
and $distinct\ xvec$
and $B \sharp* P$
and $A_P \sharp* \Psi$
and $A_P \sharp* B$
and $A_P \sharp* P$
and $A_P \sharp* K$

obtains M **where** $\Psi \otimes \Psi_P \vdash K \leftrightarrow M$ **and** $B \sharp* M$
⟨*proof*⟩

lemma *inputRenameSubject*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $M :: 'a$
and $N :: 'a$
and $P' :: ('a, 'b, 'c)\ psi$
and $A_P :: name\ list$
and $\Psi_P :: 'b$

assumes $\Psi \rhd P \longmapsto M\langle\!|N|\!\rangle \prec P'$
and $extractFrame\ P = \langle A_P, \Psi_P\rangle$
and $distinct\ A_P$
and $\Psi \otimes \Psi_P \vdash M \leftrightarrow K$
and $A_P \sharp* \Psi$
and $A_P \sharp* P$
and $A_P \sharp* M$
and $A_P \sharp* K$

shows $\Psi \rhd P \longmapsto K(\!|N|\!) \prec P'$
⟨*proof*⟩

lemma *outputRenameSubject*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$

assumes $\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *distinct* A_P
and $\Psi \otimes \Psi_P \vdash M \leftrightarrow K$
and $A_P \sharp* \Psi$
and $A_P \sharp* P$
and $A_P \sharp* M$
and $A_P \sharp* K$

shows $\Psi \rhd P \longmapsto K(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
⟨*proof*⟩

lemma *parCasesSubject*[*consumes 7*, *case-names cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and R :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*
and *yvec* :: *name list*

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto \alpha \prec R$
and *bn* $\alpha \sharp* \Psi$
and *bn* $\alpha \sharp* P$
and *bn* $\alpha \sharp* Q$
and *bn* $\alpha \sharp*$ *subject* α
and *yvec* $\sharp* P$
and *yvec* $\sharp* Q$
and *rPar1*: $\bigwedge P'\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P';$ *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_Q \sharp* \Psi$; $A_Q \sharp* P$; $A_Q \sharp* \alpha$; $A_Q \sharp* C]\!] \Longrightarrow$ *Prop* α $(P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P.\ [\![\Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q';$ *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$A_P \sharp* \Psi$; $A_P \sharp* Q$; $A_P \sharp* \alpha$; $A_P \sharp* C]\!] \Longrightarrow$ *Prop* α $(P \parallel Q')$
and *rComm1*: $\bigwedge \Psi_Q\ M\ N\ P'\ A_P\ \Psi_P\ K\ xvec\ Q'\ A_Q.$

$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu{*}xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; *yvec* $\sharp{*}$ M; *yvec* $\sharp{*}$ K; *distinct xvec*; $\alpha = \tau$;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} M$; $A_P \sharp{*} N$; $A_P \sharp{*} P'$; $A_P \sharp{*} Q$; $A_P \sharp{*}$ *xvec*; $A_P \sharp{*} Q'$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} C$;
$A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$; $A_Q \sharp{*} P$; $A_Q \sharp{*} K$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$; $A_Q \sharp{*} Q$; $A_Q \sharp{*}$ *xvec*; $A_Q \sharp{*} Q'$; $A_Q \sharp{*} C$;
xvec $\sharp{*}$ Ψ; *xvec* $\sharp{*}$ Ψ_P; *xvec* $\sharp{*}$ P; *xvec* $\sharp{*}$ M; *xvec* $\sharp{*}$ K; *xvec* $\sharp{*}$ Q; *xvec* $\sharp{*}$ Ψ_Q; *xvec* $\sharp{*}$ $C]\!] \Longrightarrow$
Prop (τ) $((\!|\nu{*}xvec|\!)(P' \parallel Q'))$

and *rComm2*: $\bigwedge \Psi_Q$ M *xvec* N P' A_P Ψ_P K Q' A_Q.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K$; *yvec* $\sharp{*}$ M; *yvec* $\sharp{*}$ K; *distinct xvec*; $\alpha = \tau$;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} M$; $A_P \sharp{*} N$; $A_P \sharp{*} P'$; $A_P \sharp{*} Q$; $A_P \sharp{*}$ *xvec*; $A_P \sharp{*} Q'$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} C$;
$A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$; $A_Q \sharp{*} P$; $A_Q \sharp{*} K$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$; $A_Q \sharp{*} Q$; $A_Q \sharp{*}$ *xvec*; $A_Q \sharp{*} Q'$; $A_Q \sharp{*} C$;
xvec $\sharp{*}$ Ψ; *xvec* $\sharp{*}$ Ψ_P; *xvec* $\sharp{*}$ P; *xvec* $\sharp{*}$ M; *xvec* $\sharp{*}$ K; *xvec* $\sharp{*}$ Q; *xvec* $\sharp{*}$ Ψ_Q; *xvec* $\sharp{*}$ $C]\!] \Longrightarrow$
Prop (τ) $((\!|\nu{*}xvec|\!)(P' \parallel Q'))$

and *rBrMerge*: $\bigwedge \Psi_Q$ M N P' A_P Ψ_P Q' A_Q.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} N$; $A_P \sharp{*} P'$;
$A_P \sharp{*} Q$; $A_P \sharp{*} Q'$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} M$; $A_Q \sharp{*} M$;
$A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$; $A_Q \sharp{*} P$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$;
$A_Q \sharp{*} Q$; $A_Q \sharp{*} Q'$; $A_P \sharp{*} C$; $A_Q \sharp{*} C$; $\alpha = ¿M(\!|N|\!)]\!] \Longrightarrow$
Prop $(¿M(\!|N|\!))$ $(P' \parallel Q')$

and *rBrComm1*: $\bigwedge \Psi_Q$ M N P' A_P Ψ_P *xvec* Q' A_Q.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
distinct xvec;
$A_P \sharp{*} \Psi$; $A_P \sharp{*} \Psi_Q$; $A_P \sharp{*} P$; $A_P \sharp{*} N$; $A_P \sharp{*} P'$; $A_P \sharp{*} Q$; $A_P \sharp{*}$ *xvec*; $A_P \sharp{*} Q'$; $A_P \sharp{*} A_Q$; $A_P \sharp{*} C$;
$A_Q \sharp{*} \Psi$; $A_Q \sharp{*} \Psi_P$; $A_Q \sharp{*} P$; $A_Q \sharp{*} N$; $A_Q \sharp{*} P'$; $A_Q \sharp{*} Q$; $A_Q \sharp{*}$ *xvec*; $A_Q \sharp{*} Q'$; $A_Q \sharp{*} C$;
$A_P \sharp{*} M$; $A_Q \sharp{*} M$; *xvec* $\sharp{*}$ M; $¡M(\!|\nu{*}xvec|\!)\langle N\rangle = \alpha$;
xvec $\sharp{*}$ Ψ; *xvec* $\sharp{*}$ Ψ_P; *xvec* $\sharp{*}$ P; *xvec* $\sharp{*}$ Q; *xvec* $\sharp{*}$ $\Psi_Q]\!] \Longrightarrow$
Prop $(¡M(\!|\nu{*}xvec|\!)\langle N\rangle)$ $(P' \parallel Q')$

and *rBrComm2*: $\bigwedge \Psi_Q$ M *xvec* N P' A_P Ψ_P Q' A_Q.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;

$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
distinct xvec;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* N$; $A_P \sharp* P'$; $A_P \sharp* Q$; $A_P \sharp* xvec$; $A_P \sharp* Q'$; $A_P \sharp* A_Q$; $A_P \sharp* C$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* N$; $A_Q \sharp* P'$; $A_Q \sharp* Q$; $A_Q \sharp* xvec$; $A_Q \sharp* Q'$; $A_Q \sharp* C$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$; $¡M(\!|\nu*xvec|\!)\langle N\rangle = \alpha$;
$xvec \sharp* \Psi$; $xvec \sharp* \Psi_P$; $xvec \sharp* P$; $xvec \sharp* Q$; $xvec \sharp* \Psi_Q$]] $\Longrightarrow$
Prop $(¡M(\!|\nu*xvec|\!)\langle N\rangle)$ $(P' \parallel Q')$

shows *Prop* α R
⟨*proof*⟩

lemma *inputCases*[*consumes 1*, *case-names cInput cBrInput*]:
fixes Ψ :: $'b$
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*

assumes *Trans*: $\Psi \rhd M(\!|\lambda*xvec\ N|\!).P \longmapsto \alpha \prec P'$
and *rInput*: $\bigwedge K$ *Tvec*. [[$\Psi \vdash M \leftrightarrow K$; *set xvec* $\subseteq$ *supp* N; *length xvec* = *length Tvec*; *distinct xvec*]] $\Longrightarrow$ *Prop* $(K(\!|N[xvec::=Tvec]|\!))$ $(P[xvec::=Tvec])$
and *rBrInput*: $\bigwedge K$ *Tvec*. [[$\Psi \vdash K \succeq M$; *set xvec* $\subseteq$ *supp* N; *length xvec* = *length Tvec*; *distinct xvec*]] $\Longrightarrow$ *Prop* $(¿K(\!|N[xvec::=Tvec]|\!))$ $(P[xvec::=Tvec])$

shows *Prop* α P'
⟨*proof*⟩

lemma *outputCases*[*consumes 1*, *case-names cOutput cBrOutput*]:
fixes Ψ :: $'b$
and M :: $'a$
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*

assumes $\Psi \rhd M\langle N\rangle.P \longmapsto \alpha \prec P'$
and $\bigwedge K.\ \Psi \vdash M \leftrightarrow K \Longrightarrow$ *Prop* $(K\langle N\rangle)$ P
and $\bigwedge K.\ \Psi \vdash M \preceq K \Longrightarrow$ *Prop* $(¡K\langle N\rangle)$ P

shows *Prop* α P'
⟨*proof*⟩

lemma *caseCases*[*consumes 1*, *case-names cCase*]:
fixes Ψ :: $'b$

and $Cs :: ('c \times ('a, 'b, 'c)\ psi)\ list$
and $\alpha :: 'a\ action$
and $P' :: ('a, 'b, 'c)\ psi$

assumes *Trans*: $\Psi \rhd (Cases\ Cs) \longmapsto Rs$
and *rCase*: $\bigwedge\varphi\ P.\ [\![\Psi \rhd P \longmapsto Rs;\ (\varphi, P) \in set\ Cs;\ \Psi \vdash \varphi;\ guarded\ P]\!] \Longrightarrow Prop$

shows *Prop*
⟨*proof*⟩

lemma *resCases*[*consumes* 7, *case-names cOpen cBrOpen cRes cBrClose*]:
fixes $\Psi :: 'b$
and $x :: name$
and $P :: ('a, 'b, 'c)\ psi$
and $\alpha :: 'a\ action$
and $P' :: ('a, 'b, 'c)\ psi$
and $C :: 'f::fs\text{-}name$

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto \alpha \prec P'$
and $x \sharp \Psi$
and $x \sharp \alpha$
and $x \sharp P'$
and $bn\ \alpha \sharp* \Psi$
and $bn\ \alpha \sharp* P$
and $bn\ \alpha \sharp* subject\ \alpha$
and *rOpen*: $\bigwedge M\ xvec\ yvec\ y\ N\ P'.\ [\![\Psi \rhd P \longmapsto M(\!|\nu*(xvec@yvec)|\!)\langle([(x, y)] \cdot N)\rangle \prec ([(x, y)] \cdot P');\ y \in supp\ N;$
$x \sharp N;\ x \sharp P';\ x \neq y;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp M;$
$distinct\ xvec;\ distinct\ yvec;$
$xvec \sharp* \Psi;\ y \sharp \Psi;\ yvec \sharp* \Psi;\ xvec \sharp* P;\ y \sharp P;$
$yvec \sharp* P;\ xvec \sharp* M;\ y \sharp M;$
$yvec \sharp* M;\ xvec \sharp* yvec]\!] \Longrightarrow$
$Prop\ (M(\!|\nu*(xvec@y\#yvec)|\!)\langle N\rangle)\ P'$
and *rBrOpen*: $\bigwedge M\ xvec\ yvec\ y\ N\ P'.\ [\![\Psi \rhd P \longmapsto ¡M(\!|\nu*(xvec@yvec)|\!)\langle([(x, y)] \cdot N)\rangle \prec ([(x, y)] \cdot P');\ y \in supp\ N;$
$x \sharp N;\ x \sharp P';\ x \neq y;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp M;$
$distinct\ xvec;\ distinct\ yvec;$
$xvec \sharp* \Psi;\ y \sharp \Psi;\ yvec \sharp* \Psi;\ xvec \sharp* P;\ y \sharp P;$
$yvec \sharp* P;\ xvec \sharp* M;\ y \sharp M;$
$yvec \sharp* M;\ xvec \sharp* yvec]\!] \Longrightarrow$
$Prop\ (¡M(\!|\nu*(xvec@y\#yvec)|\!)\langle N\rangle)\ P'$
and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto \alpha \prec P']\!] \Longrightarrow Prop\ \alpha\ (\!|\nu x|\!)P')$
and *rBrClose*: $\bigwedge M\ xvec\ N\ P'.$
$[\![\Psi \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P';$
$x \in supp\ M;$
$distinct\ xvec;\ xvec \sharp* \Psi;\ xvec \sharp* P;$
$xvec \sharp* M;$
$x \sharp \Psi;\ x \sharp xvec;$

$xvec \sharp* C]] \Longrightarrow Prop\ (\tau)\ ((\!|\nu x|\!)((\!|\nu* xvec|\!)P'))$
shows $Prop\ \alpha\ P'$
⟨*proof*⟩

lemma *resCases'*[*consumes* 7, *case-names cOpen cBrOpen cRes cBrClose*]:
fixes Ψ :: $'b$
and x :: *name*
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto \alpha \prec P'$
and $x \sharp \Psi$
and $x \sharp \alpha$
and $x \sharp P'$
and $bn\ \alpha \sharp* \Psi$
and $bn\ \alpha \sharp* P$
and $bn\ \alpha \sharp* subject\ \alpha$
and *rOpen*: $\bigwedge M\ xvec\ yvec\ y\ N\ P'.\ [\![\Psi \rhd ([(x, y)] \cdot P) \longmapsto M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P';\ y \in supp\ N;$
$x \sharp N;\ x \sharp P';\ x \neq y;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp M;$
$distinct\ xvec;\ distinct\ yvec;$
$xvec \sharp* \Psi;\ y \sharp \Psi;\ yvec \sharp* \Psi;\ xvec \sharp* P;\ y \sharp P;$
$yvec \sharp* P;\ xvec \sharp* M;\ y \sharp M;$
$yvec \sharp* M;\ xvec \sharp* yvec]\!] \Longrightarrow$
$Prop\ (M(\!|\nu*(xvec@y\#yvec)|\!)\langle N\rangle)\ P'$
and *rBrOpen*: $\bigwedge M\ xvec\ yvec\ y\ N\ P'.\ [\![\Psi \rhd ([(x, y)] \cdot P) \longmapsto ¡M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P';\ y \in supp\ N;$
$x \sharp N;\ x \sharp P';\ x \neq y;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp M;$
$distinct\ xvec;\ distinct\ yvec;$
$xvec \sharp* \Psi;\ y \sharp \Psi;\ yvec \sharp* \Psi;\ xvec \sharp* P;\ y \sharp P;$
$yvec \sharp* P;\ xvec \sharp* M;\ y \sharp M;$
$yvec \sharp* M;\ xvec \sharp* yvec]\!] \Longrightarrow$
$Prop\ (¡M(\!|\nu*(xvec@y\#yvec)|\!)\langle N\rangle)\ P'$
and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto \alpha \prec P']\!] \Longrightarrow Prop\ \alpha\ ((\!|\nu x|\!)P')$
and *rBrClose*: $\bigwedge M\ xvec\ N\ P'.$
$[\![\Psi \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P';$
$x \in supp\ M;$
$distinct\ xvec;\ xvec \sharp* \Psi;\ xvec \sharp* P;$
$xvec \sharp* M;$
$x \sharp \Psi;\ x \sharp xvec;$
$xvec \sharp* C]\!] \Longrightarrow Prop\ (\tau)\ ((\!|\nu x|\!)((\!|\nu* xvec|\!)P'))$

shows $Prop\ \alpha\ P'$
⟨*proof*⟩

lemma *resInputCases'*[*consumes* 4, *case-names cRes*]:
fixes Ψ :: $'b$

and x :: *name*
and M :: $'a$
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and R :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|N|\!) \prec R$
and *2*: $x \sharp \Psi$
and $x \sharp (M(\!|N|\!))$
and *4*: $x \sharp R$
and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P']\!] \Longrightarrow Prop\ ((\!|\nu x|\!)P')$

shows *Prop* R
⟨*proof*⟩

lemma *resBrInputCases'*[*consumes 4*, *case-names cRes*]:
fixes Ψ :: $'b$
and x :: *name*
and M :: $'a$
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and R :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto ¿M(\!|N|\!) \prec R$
and *2*: $x \sharp \Psi$
and $x \sharp (¿M(\!|N|\!))$
and *4*: $x \sharp R$
and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P']\!] \Longrightarrow Prop\ ((\!|\nu x|\!)P')$

shows *Prop* R
⟨*proof*⟩

lemma *resOutputCases'''*[*consumes 7*, *case-names cOpen cRes*]:
fixes Ψ :: $'b$
and x :: *name*
and M :: $'a$
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and R :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|\nu*(zvec1\ @\ zvec2)|\!)\langle N\rangle \prec R$
and *1*: $x \sharp \Psi$
and $x \sharp (M(\!|\nu*(zvec1\ @\ zvec2)|\!)\langle N\rangle)$
and *3*: $x \sharp R$
and $(zvec1\ @\ zvec2) \sharp* \Psi$
and $(zvec1\ @\ zvec2) \sharp* P$

and $(zvec1 \ @ \ zvec2) \sharp* M$
and *rOpen*: $\bigwedge M \ xvec \ yvec \ y \ N \ P'. \ [\![\Psi \rhd ([(x, y)] \cdot P) \longmapsto M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'; \ y \in supp \ N; \ x \sharp N; \ x \sharp P'; \ x \neq y; \ y \sharp xvec; \ y \sharp yvec; \ y \sharp M; \ distinct \ xvec; \ distinct \ yvec; \ xvec \sharp* \Psi; \ y \sharp \Psi; \ yvec \sharp* \Psi; \ xvec \sharp* P; \ y \sharp P; \ yvec \sharp* P; \ xvec \sharp* M; \ y \sharp M; \ yvec \sharp* M; \ xvec \sharp* yvec]\!] \Longrightarrow Prop \ P'$
and *rScope*: $\bigwedge P'. \ [\![\Psi \rhd P \longmapsto M(\!|\nu*(zvec1 \ @ \ zvec2)|\!)\langle N\rangle \prec P']\!] \Longrightarrow Prop \ ((\!|\nu x|\!)P')$

shows *Prop R*
⟨*proof*⟩

lemma *resOutputCases''*[*consumes 7*, *case-names cOpen cRes*]:
fixes Ψ :: *'b*
and x :: *name*
and z :: *name*
and M :: *'a*
and N :: *'a*
and P :: $('a, 'b, 'c) \ psi$
and R :: $('a, 'b, 'c) \ psi$
and C :: *'f::fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|\nu*(zvec1@y\#zvec2)|\!)\langle N\rangle \prec R$
and *1*: $x \sharp \Psi$
and $x \sharp (M(\!|\nu*(zvec1@y\#zvec2)|\!)\langle N\rangle)$
and *3*: $x \sharp R$
and $(zvec1@y\#zvec2) \sharp* \Psi$
and $(zvec1@y\#zvec2) \sharp* P$
and $(zvec1@y\#zvec2) \sharp* M$
and *rOpen*: $\bigwedge M \ xvec \ yvec \ y \ N \ P'. \ [\![\Psi \rhd ([(x, y)] \cdot P) \longmapsto M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P'; \ y \in supp \ N; \ x \sharp N; \ x \sharp P'; \ x \neq y; \ y \sharp xvec; \ y \sharp yvec; \ y \sharp M; \ distinct \ xvec; \ distinct \ yvec; \ xvec \sharp* \Psi; \ y \sharp \Psi; \ yvec \sharp* \Psi; \ xvec \sharp* P; \ y \sharp P; \ yvec \sharp* P; \ xvec \sharp* M; \ y \sharp M; \ yvec \sharp* M; \ xvec \sharp* yvec]\!] \Longrightarrow Prop \ P'$
and *rScope*: $\bigwedge P'. \ [\![\Psi \rhd P \longmapsto M(\!|\nu*(zvec1@y\#zvec2)|\!)\langle N\rangle \prec P']\!] \Longrightarrow Prop \ ((\!|\nu x|\!)P')$

shows *Prop R*
⟨*proof*⟩

abbreviation
statImpJudge (‹- ↪ -› [*80*, *80*] *80*)
where $\Psi \hookrightarrow \Psi' \equiv AssertionStatImp \ \Psi \ \Psi'$

lemma *statEqTransition*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Rs :: $('a, 'b, 'c)$ *residual*
and Ψ' :: $'b$

assumes $\Psi \rhd P \longmapsto Rs$
and $\Psi \simeq \Psi'$

shows $\Psi' \rhd P \longmapsto Rs$
⟨*proof*⟩

lemma *brInputTermSupp*:
fixes Ψ :: $'b$::*fs-name*
and P :: $('a, 'b, ('c$::*fs-name*$))$ *psi*
and P' :: $('a, 'b, 'c)$ *psi*
and N :: $'a$::*fs-name*
and K :: $'a$::*fs-name*

assumes $\Psi \rhd P \longmapsto ¿K(\!|N|\!) \prec P'$

shows (*supp* K) $\subseteq$ ((*supp* P)::*name set*)
⟨*proof*⟩

lemma *brOutputTermSupp*:
fixes Ψ :: $'b$::*fs-name*
and P :: $('a, 'b, ('c$::*fs-name*$))$ *psi*
and P' :: $('a, 'b, 'c)$ *psi*
and N :: $'a$::*fs-name*
and K :: $'a$::*fs-name*
and *xvec* :: *name list*

assumes $\Psi \rhd P \longmapsto RBrOut\ K\ ((\!|\nu * xvec|\!)N \prec' P')$

shows (*supp* K) $\subseteq$ ((*supp* P)::*name set*)
⟨*proof*⟩

lemma *actionPar1Dest'*:
fixes α :: $('a$::*fs-name*$)$ *action*
and P :: $('a, 'b$::*fs-name*$, 'c$::*fs-name*$)$ *psi*
and β :: $('a$::*fs-name*$)$ *action*
and Q :: $('a, 'b, 'c)$ *psi*
and R :: $('a, 'b, 'c)$ *psi*

assumes $\alpha \prec P = \beta \prec (Q \parallel R)$
and *bn* $\alpha \sharp* R$
and *bn* $\beta \sharp* R$

obtains T **where** $P = T \parallel R$ **and** $\alpha \prec T = \beta \prec Q$
⟨*proof*⟩

lemma *actionPar2Dest'*:
fixes α :: ('*a*::*fs-name*) *action*
and P :: ('*a*, '*b*::*fs-name*, '*c*::*fs-name*) *psi*
and β :: ('*a*::*fs-name*) *action*
and Q :: ('*a*, '*b*, '*c*) *psi*
and R :: ('*a*, '*b*, '*c*) *psi*

assumes $\alpha \prec P = \beta \prec (Q \parallel R)$
and *bn* $\alpha \sharp* Q$
and *bn* $\beta \sharp* Q$

obtains T **where** $P = Q \parallel T$ **and** $\alpha \prec T = \beta \prec R$
⟨*proof*⟩

lemma *expandNonTauFrame*:
fixes Ψ :: '*b*
and P :: ('*a*, '*b*, '*c*) *psi*
and α :: '*a action*
and P' :: ('*a*, '*b*, '*c*) *psi*
and A_P :: *name list*
and Ψ_P :: '*b*
and C :: '*f*::*fs-name*
and C' :: '*g*::*fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto \alpha \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and *distinct* A_P
and *bn* $\alpha \sharp* subject\ \alpha$
and $A_P \sharp* P$
and $A_P \sharp* \alpha$
and $A_P \sharp* C$
and $A_P \sharp* C'$
and *bn* $\alpha \sharp* P$
and *bn* $\alpha \sharp* C'$
and $\alpha \neq \tau$

obtains p Ψ' A_P' Ψ_P' **where** *set* $p \subseteq set(bn\ \alpha) \times set(bn(p \cdot \alpha))$ **and** $(p \cdot \Psi_P) \otimes \Psi' \simeq \Psi_P'$ **and** *distinctPerm* p **and**
extractFrame $P' = \langle A_P', \Psi_P' \rangle$ **and** $A_P' \sharp* P'$ **and** $A_P' \sharp* \alpha$ **and** $A_P' \sharp* (p \cdot \alpha)$ **and**
$A_P' \sharp* C$ **and** $(bn(p \cdot \alpha)) \sharp* C'$ **and** $(bn(p \cdot \alpha)) \sharp* \alpha$ **and** $(bn(p \cdot \alpha)) \sharp* P'$ **and** *distinct* A_P'
⟨*proof*⟩

lemma *expandTauFrame*:
fixes Ψ :: '*b*

and P :: $('a, 'b, 'c)$ *psi*
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$
and C :: $'f$::*fs-name*

assumes $\Psi \rhd P \longmapsto \tau \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and *distinct* A_P
and $A_P \sharp* P$
and $A_P \sharp* C$

obtains Ψ' A_P' Ψ_P' **where** *extractFrame* $P' = \langle A_P', \Psi_P' \rangle$ **and** $\Psi_P \otimes \Psi' \simeq \Psi_P'$ **and** $A_P' \sharp* C$ **and** $A_P' \sharp* P'$ **and** *distinct* A_P'
⟨*proof*⟩

lemma *expandFrame*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$
and C :: $'f$::*fs-name*
and C' :: $'g$::*fs-name*

assumes *Trans*: $\Psi \rhd P \longmapsto \alpha \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P \rangle$
and *distinct* A_P
and *bn* $\alpha \sharp*$ *subject* α
and *distinct*(*bn* α)
and $A_P \sharp* \alpha$
and $A_P \sharp* P$
and $A_P \sharp* C$
and $A_P \sharp* C'$
and *bn* $\alpha \sharp* P$
and *bn* $\alpha \sharp* C'$

obtains p Ψ' A_P' Ψ_P' **where** *set* $p \subseteq$ *set*(*bn* α) $\times$ *set*(*bn*$(p \cdot \alpha)$) **and** $(p \cdot \Psi_P) \otimes \Psi' \simeq \Psi_P'$ **and** *distinctPerm* p **and**
extractFrame $P' = \langle A_P', \Psi_P' \rangle$ **and** $A_P' \sharp* P'$ **and** $A_P' \sharp* \alpha$ **and** $A_P' \sharp* (p \cdot \alpha)$ **and**
$A_P' \sharp* C$ **and** $(bn(p \cdot \alpha)) \sharp* C'$ **and** $(bn(p \cdot \alpha)) \sharp* \alpha$ **and** $(bn(p \cdot \alpha)) \sharp* P'$ **and** *distinct* A_P'
⟨*proof*⟩

abbreviation
frameImpJudge ($\langle$- $\hookrightarrow_F$ -$\rangle$ [80, 80] 80)
where $F \hookrightarrow_F G \equiv$ *FrameStatImp* F G

lemma *FrameStatEqImpCompose*:
fixes $F :: 'b\ frame$
and $G :: 'b\ frame$
and $H :: 'b\ frame$
and $I :: 'b\ frame$

assumes $F \simeq_F G$
and $G \hookrightarrow_F H$
and $H \simeq_F I$

shows $F \hookrightarrow_F I$
⟨*proof*⟩

lemma *transferNonTauFrame*:
fixes $\Psi_F :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $\alpha :: 'a\ action$
and $P' :: ('a, 'b, 'c)\ psi$
and $A_F :: name\ list$
and $A_G :: name\ list$
and $\Psi_G :: 'b$

assumes $\Psi_F \rhd P \longmapsto \alpha \prec P'$
and $extractFrame\ P = \langle A_P, \Psi_P \rangle$
and $distinct\ A_P$
and $distinct(bn\ \alpha)$
and $\langle A_F, \Psi_F \otimes \Psi_P \rangle \hookrightarrow_F \langle A_G, \Psi_G \otimes \Psi_P \rangle$
and $A_F \sharp* P$
and $A_G \sharp* P$
and $A_F \sharp* subject\ \alpha$
and $A_G \sharp* subject\ \alpha$
and $A_P \sharp* A_F$
and $A_P \sharp* A_G$
and $A_P \sharp* \Psi_F$
and $A_P \sharp* \Psi_G$
and $\alpha \neq \tau$

shows $\Psi_G \rhd P \longmapsto \alpha \prec P'$
⟨*proof*⟩

lemma *transferTauFrame*:
fixes $\Psi_F :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $P' :: ('a, 'b, 'c)\ psi$
and $A_F :: name\ list$
and $A_G :: name\ list$
and $\Psi_G :: 'b$

assumes $\Psi_F \rhd P \longmapsto\tau \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *distinct* A_P
and $\langle A_F, \Psi_F \otimes \Psi_P\rangle \hookrightarrow_F \langle A_G, \Psi_G \otimes \Psi_P\rangle$
and $A_F \sharp* P$
and $A_G \sharp* P$
and $A_P \sharp* A_F$
and $A_P \sharp* A_G$
and $A_P \sharp* \Psi_F$
and $A_P \sharp* \Psi_G$

shows $\Psi_G \rhd P \longmapsto\tau \prec P'$
⟨*proof*⟩

lemma *transferFrame*:
fixes Ψ_F :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*
and A_F :: *name list*
and A_G :: *name list*
and Ψ_G :: $'b$

assumes $\Psi_F \rhd P \longmapsto\alpha \prec P'$
and *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *distinct* A_P
and $\langle A_F, \Psi_F \otimes \Psi_P\rangle \hookrightarrow_F \langle A_G, \Psi_G \otimes \Psi_P\rangle$
and $A_F \sharp* P$
and $A_G \sharp* P$
and $A_F \sharp* \mathit{subject}\ \alpha$
and $A_G \sharp* \mathit{subject}\ \alpha$
and $A_P \sharp* A_F$
and $A_P \sharp* A_G$
and $A_P \sharp* \Psi_F$
and $A_P \sharp* \Psi_G$

shows $\Psi_G \rhd P \longmapsto\alpha \prec P'$
⟨*proof*⟩

lemma *parCasesInputFrame*[*consumes* 7, *case-names cPar1 cPar2*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and T :: $('a, 'b, 'c)$ *psi*
and C :: $'d$::*fs-name*

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto M(\!|N|\!) \prec T$
and $extractFrame(P \parallel Q) = \langle A_{PQ}, \Psi_{PQ}\rangle$
and $distinct\ A_{PQ}$
and $A_{PQ} \sharp* \Psi$
and $A_{PQ} \sharp* P$
and $A_{PQ} \sharp* Q$
and $A_{PQ} \sharp* M$
and *rPar1*: $\bigwedge P'\ A_P\ \Psi_P\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;$
$distinct\ A_P;\ distinct\ A_Q;\ A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;$
$A_P \sharp* \Psi_Q;\ A_Q \sharp* \Psi_P;\ A_P \sharp* A_Q;\ A_{PQ} = A_P@A_Q;\ \Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow Prop\ (P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|N|\!) \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;$
$distinct\ A_P;\ distinct\ A_Q;\ A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;$
$A_P \sharp* \Psi_Q;\ A_Q \sharp* \Psi_P;\ A_P \sharp* A_Q;\ A_{PQ} = A_P@A_Q;\ \Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow Prop\ (P \parallel Q')$
shows *Prop T*
⟨*proof*⟩

lemma *parCasesBrInputFrame*[*consumes 7*, *case-names cPar1 cPar2 cBrMerge*]:
fixes Ψ :: *'b*
and P :: *('a, 'b, 'c) psi*
and Q :: *('a, 'b, 'c) psi*
and M :: *'a*
and *xvec* :: *name list*
and N :: *'a*
and T :: *('a, 'b, 'c) psi*
and C :: *'d*::*fs-name*

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto ¿M(\!|N|\!) \prec T$
and $extractFrame(P \parallel Q) = \langle A_{PQ}, \Psi_{PQ}\rangle$
and $distinct\ A_{PQ}$
and $A_{PQ} \sharp* \Psi$
and $A_{PQ} \sharp* P$
and $A_{PQ} \sharp* Q$
and $A_{PQ} \sharp* M$
and *rPar1*: $\bigwedge P'\ A_P\ \Psi_P\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;$
$distinct\ A_P;\ distinct\ A_Q;\ A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;$
$A_P \sharp* \Psi_Q;\ A_Q \sharp* \Psi_P;\ A_P \sharp* A_Q;\ A_{PQ} = A_P@A_Q;\ \Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow Prop\ (P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ P = \langle A_P, \Psi_P\rangle;\ extractFrame\ Q = \langle A_Q, \Psi_Q\rangle;$
$distinct\ A_P;\ distinct\ A_Q;\ A_P \sharp* \Psi;\ A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;$

$A_P \sharp* \Psi_Q$; $A_Q \sharp* \Psi_P$; $A_P \sharp* A_Q$; $A_{PQ} = A_P @ A_Q$; $\Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow$ *Prop* $(P \parallel Q')$

and *rBrMerge*: $\bigwedge \Psi_Q\ P'\ A_P\ \Psi_P\ Q'\ A_Q$. $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P; $\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q'$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q; $A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* A_Q$; $A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* Q$; $A_{PQ} = A_P @ A_Q$; $\Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow$ *Prop* $(P' \parallel Q')$

shows *Prop* T

⟨*proof*⟩

lemma *parCasesOutputFrame*[*consumes 11*, *case-names cPar1 cPar2*]:

fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and T :: $('a, 'b, 'c)$ *psi*
and C :: $'d$::*fs-name*

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec T$
and *xvec* $\sharp*$ Ψ
and *xvec* $\sharp*$ P
and *xvec* $\sharp*$ Q
and *xvec* $\sharp*$ M
and *extractFrame*$(P \parallel Q) = \langle A_{PQ}, \Psi_{PQ}\rangle$
and *distinct* A_{PQ}
and $A_{PQ} \sharp* \Psi$
and $A_{PQ} \sharp* P$
and $A_{PQ} \sharp* Q$
and $A_{PQ} \sharp* M$
and *rPar1*: $\bigwedge P'\ A_P\ \Psi_P\ A_Q\ \Psi_Q$. $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_P; *distinct* A_Q; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* M$; $A_Q \sharp* \Psi$; $A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* M$; $A_P \sharp* \Psi_Q$; $A_Q \sharp* \Psi_P$; $A_P \sharp* A_Q$; $A_{PQ} = A_P @ A_Q$; $\Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow$ *Prop* $(P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P\ A_Q\ \Psi_Q$. $[\![\Psi \otimes \Psi_P \rhd Q \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec Q'$; *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_P; *distinct* A_Q; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* M$; $A_Q \sharp* \Psi$; $A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* M$; $A_P \sharp* \Psi_Q$; $A_Q \sharp* \Psi_P$; $A_P \sharp* A_Q$; $A_{PQ} = A_P @ A_Q$; $\Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow$ *Prop* $(P \parallel Q')$

shows *Prop T*
⟨*proof*⟩

lemma *parCasesBrOutputFrame*[*consumes 11*, *case-names cPar1 cPar2 cBrComm1 cBrComm2*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and Q :: (*'a*, *'b*, *'c*) *psi*
and M :: *'a*
and *xvec* :: *name list*
and N :: *'a*
and T :: (*'a*, *'b*, *'c*) *psi*
and C :: *'d*::*fs-name*

assumes *Trans*: $\Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec T$
and $xvec \sharp* \Psi$
and $xvec \sharp* P$
and $xvec \sharp* Q$
and $xvec \sharp* M$
and $extractFrame(P \parallel Q) = \langle A_{PQ}, \Psi_{PQ}\rangle$
and *distinct* A_{PQ}
and $A_{PQ} \sharp* \Psi$
and $A_{PQ} \sharp* P$
and $A_{PQ} \sharp* Q$
and $A_{PQ} \sharp* M$
and *rPar1*: $\bigwedge P'\ A_P\ \Psi_P\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P';$ *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$;
distinct A_P; *distinct* A_Q; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* M$; $A_Q \sharp* \Psi$; $A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* M$;
$A_P \sharp* \Psi_Q$; $A_Q \sharp* \Psi_P$; $A_P \sharp* A_Q$; $A_{PQ} = A_P @ A_Q$; $\Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow$ *Prop* $(P' \parallel Q)$
and *rPar2*: $\bigwedge Q'\ A_P\ \Psi_P\ A_Q\ \Psi_Q.\ [\![\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';$ *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$;
distinct A_P; *distinct* A_Q; $A_P \sharp* \Psi$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* M$; $A_Q \sharp* \Psi$; $A_Q \sharp* P$; $A_Q \sharp* Q$; $A_Q \sharp* M$;
$A_P \sharp* \Psi_Q$; $A_Q \sharp* \Psi_P$; $A_P \sharp* A_Q$; $A_{PQ} = A_P @ A_Q$; $\Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow$ *Prop* $(P \parallel Q')$
and *rBrComm1*: $\bigwedge \Psi_Q\ P'\ A_P\ \Psi_P\ Q'\ A_Q.$
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';$ *extractFrame* $P = \langle A_P, \Psi_P\rangle$; *distinct* A_P;
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec Q';$ *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$; *distinct* A_Q;
distinct xvec;
$A_P \sharp* \Psi$; $A_P \sharp* \Psi_Q$; $A_P \sharp* P$; $A_P \sharp* Q$; $A_P \sharp* A_Q$;
$A_Q \sharp* \Psi$; $A_Q \sharp* \Psi_P$; $A_Q \sharp* P$; $A_Q \sharp* Q$;
$A_P \sharp* M$; $A_Q \sharp* M$; $xvec \sharp* M$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* Q$;
$A_{PQ} = A_P @ A_Q$; $\Psi_{PQ} = \Psi_P \otimes \Psi_Q]\!] \Longrightarrow$
Prop $(P' \parallel Q')$

and *rBrComm2*: $\bigwedge \Psi_Q\ P'\ A_P\ \Psi_P\ Q'\ A_Q$.
$[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;$
$distinct\ A_P;$
$\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ distinct$
$A_Q;$
$distinct\ xvec;$
$A_P \sharp{*}\ \Psi;\ A_P \sharp{*}\ \Psi_Q;\ A_P \sharp{*}\ P;\ A_P \sharp{*}\ Q;\ A_P \sharp{*}\ A_Q;$
$A_Q \sharp{*}\ \Psi;\ A_Q \sharp{*}\ \Psi_P;\ A_Q \sharp{*}\ P;\ A_Q \sharp{*}\ Q;$
$A_P \sharp{*}\ M;\ A_Q \sharp{*}\ M;\ xvec \sharp{*}\ M;$
$xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ P;\ xvec \sharp{*}\ Q;$
$A_{PQ} = A_P@A_Q;\ \Psi_{PQ} = \Psi_P \otimes \Psi_Q[\!] \Longrightarrow$
$Prop\ (P' \parallel Q')$
shows *Prop T*
⟨*proof*⟩

inductive *bangPred* :: ('*a*, '*b*, '*c*) *psi* ⇒ ('*a*, '*b*, '*c*) *psi* ⇒ *bool*
where
aux1: *bangPred P* (!*P*)
| *aux2*: *bangPred P* (*P* ∥ !*P*)

lemma *bangInduct*[*consumes 1*, *case-names cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2 cBang*]:
fixes Ψ :: '*b*
and *P* :: ('*a*, '*b*, '*c*) *psi*
and *Rs* :: ('*a*, '*b*, '*c*) *residual*
and *Prop* :: '*d*::*fs-name* ⇒ '*b* ⇒ ('*a*, '*b*, '*c*) *psi* ⇒ ('*a*, '*b*, '*c*) *residual* ⇒ *bool*
and *C* :: '*d*

assumes $\Psi \rhd !P \longmapsto Rs$
and *rPar1*: $\bigwedge \alpha\ P'\ C.\ [\![\Psi \rhd P \longmapsto \alpha \prec P';\ bn\ \alpha \sharp{*}\ \Psi;\ bn\ \alpha \sharp{*}\ P;\ bn\ \alpha \sharp{*}$
$subject\ \alpha;\ bn\ \alpha \sharp{*}\ C;\ distinct(bn\ \alpha)[\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (\alpha \prec (P' \parallel !P))$
and *rPar2*: $\bigwedge \alpha\ P'\ C.\ [\![\Psi \rhd !P \longmapsto \alpha \prec P';\ bn\ \alpha \sharp{*}\ \Psi;\ bn\ \alpha \sharp{*}\ P;\ bn\ \alpha \sharp{*}$
$subject\ \alpha;\ bn\ \alpha \sharp{*}\ C;\ distinct(bn\ \alpha);$
$\bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (\alpha \prec P')[\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (\alpha$
$\prec (P \parallel P'))$
and *rComm1*: $\bigwedge M\ N\ P'\ K\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P';\ \Psi \rhd !P$
$\longmapsto K(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (K(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'');\ \Psi \vdash M \leftrightarrow$
$K;$
$xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ P;\ xvec \sharp{*}\ M;\ xvec \sharp{*}\ K;$
$xvec \sharp{*}\ C;\ distinct\ xvec[\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (\tau \prec (\!|\nu{*}xvec|\!)(P' \parallel P''))$
and *rComm2*: $\bigwedge M\ xvec\ N\ P'\ K\ P''\ C.\ [\![\Psi \rhd P \longmapsto M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P';\ \Psi \rhd$
$!P \longmapsto K(\!|N|\!) \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (K(\!|N|\!) \prec P'');\ \Psi \vdash M \leftrightarrow K;$
$xvec \sharp{*}\ \Psi;\ xvec \sharp{*}\ P;\ xvec \sharp{*}\ M;\ xvec \sharp{*}\ K;$
$xvec \sharp{*}\ C;\ distinct\ xvec[\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (\tau \prec (\!|\nu{*}xvec|\!)(P' \parallel P''))$
and *rBrMerge*: $\bigwedge M\ N\ P'\ P''\ C.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \Psi \rhd !P \longmapsto ¿M(\!|N|\!)$
$\prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (¿M(\!|N|\!) \prec P'')[\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel !P)\ (¿M(\!|N|\!) \prec (P' \parallel P''))$
and *rBrComm1*: $\bigwedge M\ N\ P'\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \Psi \rhd !P$
$\longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P'');$

$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* C$; *distinct xvec*$\rrbracket \Longrightarrow$ *Prop* $C\ \Psi\ (P \parallel !P)\ (¡M(\!|\nu * xvec|\!)\langle N\rangle \prec (P' \parallel P''))$
and *rBrComm2*: $\bigwedge M\ N\ P'\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ \Psi \rhd !P \longmapsto ¿M(\!|N|\!) \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (¿M(\!|N|\!) \prec P'')$;
$xvec \sharp* \Psi$; $xvec \sharp* P$; $xvec \sharp* M$; $xvec \sharp* C$; *distinct xvec*$\rrbracket \Longrightarrow$ *Prop* $C\ \Psi\ (P \parallel !P)\ (¡M(\!|\nu * xvec|\!)\langle N\rangle \prec (P' \parallel P''))$
and *rBang*: $\bigwedge Rs\ C.\ [\![\Psi \rhd P \parallel !P \longmapsto Rs;\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel !P)\ Rs;\ guarded\ P]\!] \Longrightarrow Prop\ C\ \Psi\ (!P)\ Rs$
shows *Prop* $C\ \Psi\ (!P)\ Rs$
⟨*proof*⟩

lemma *bangInputInduct*[*consumes 1*, *case-names cPar1 cPar2 cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and *Prop* :: $'b \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow 'a \Rightarrow 'a \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow bool$

assumes $\Psi \rhd !P \longmapsto M(\!|N|\!) \prec P'$
and *rPar1*: $\bigwedge P'.\ \Psi \rhd P \longmapsto M(\!|N|\!) \prec P' \Longrightarrow Prop\ \Psi\ (P \parallel !P)\ M\ N\ (P' \parallel !P)$
and *rPar2*: $\bigwedge P'.\ [\![\Psi \rhd !P \longmapsto M(\!|N|\!) \prec P';\ Prop\ \Psi\ (!P)\ M\ N\ P']\!] \Longrightarrow Prop\ \Psi\ (P \parallel !P)\ M\ N\ (P \parallel P')$
and *rBang*: $\bigwedge P'.\ [\![\Psi \rhd P \parallel !P \longmapsto M(\!|N|\!) \prec P';\ Prop\ \Psi\ (P \parallel !P)\ M\ N\ P';\ guarded\ P]\!] \Longrightarrow Prop\ \Psi\ (!P)\ M\ N\ P'$
shows *Prop* $\Psi\ (!P)\ M\ N\ P'$
⟨*proof*⟩

lemma *brbangInputInduct*[*consumes 1*, *case-names cPar1 cPar2 cBrMerge cBang*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and *Prop* :: $'b \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow 'a \Rightarrow 'a \Rightarrow ('a, 'b, 'c)\ psi \Rightarrow bool$

assumes $\Psi \rhd !P \longmapsto ¿M(\!|N|\!) \prec P'$
and *rPar1*: $\bigwedge P'.\ \Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P' \Longrightarrow Prop\ \Psi\ (P \parallel !P)\ M\ N\ (P' \parallel !P)$
and *rPar2*: $\bigwedge P'.\ [\![\Psi \rhd !P \longmapsto ¿M(\!|N|\!) \prec P';\ Prop\ \Psi\ (!P)\ M\ N\ P']\!] \Longrightarrow Prop\ \Psi\ (P \parallel !P)\ M\ N\ (P \parallel P')$
and *rBrMerge*: $\bigwedge P'\ P''\ C.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \Psi \rhd !P \longmapsto ¿M(\!|N|\!) \prec P'';\ \bigwedge C.\ Prop\ \Psi\ (!P)\ M\ N\ P']\!] \Longrightarrow Prop\ \Psi\ (P \parallel !P)\ M\ N\ (P' \parallel P'')$
and *rBang*: $\bigwedge P'.\ [\![\Psi \rhd P \parallel !P \longmapsto ¿M(\!|N|\!) \prec P';\ Prop\ \Psi\ (P \parallel !P)\ M\ N\ P';\ guarded\ P]\!] \Longrightarrow Prop\ \Psi\ (!P)\ M\ N\ P'$
shows *Prop* $\Psi\ (!P)\ M\ N\ P'$
⟨*proof*⟩

lemma *bangOutputInduct*[*consumes 1*, *case-names cPar1 cPar2 cBang*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and M :: *'a*
and *xvec* :: *name list*
and N :: *'a*
and P' :: (*'a*, *'b*, *'c*) *psi*
and *Prop* :: *'d*::*fs-name* ⇒ *'b* ⇒ (*'a*, *'b*, *'c*) *psi* ⇒ *'a* ⇒ (*'a*, *'b*, *'c*) *boundOutput* ⇒ *bool*
and C :: *'d*

assumes $\Psi \rhd !P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *rPar1*: $\bigwedge xvec\ N\ P'\ C.\ [\![\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ xvec\ \sharp*\ \Psi;\ xvec\ \sharp*\ P;\ xvec\ \sharp*\ M;\ xvec\ \sharp*\ C;\ distinct\ xvec]\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ M\ (([\!|\nu * xvec|\!])N \prec' (P' \parallel !P))$
and *rPar2*: $\bigwedge xvec\ N\ P'\ C.\ [\![\Psi \rhd !P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ M\ (([\!|\nu * xvec|\!])N \prec' P');\ xvec\ \sharp*\ \Psi;\ xvec\ \sharp*\ P;\ xvec\ \sharp*\ M;\ xvec\ \sharp*\ C;\ distinct\ xvec]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel !P)\ M\ (([\!|\nu * xvec|\!])N \prec' (P \parallel P'))$
and *rBang*: $\bigwedge B\ C.\ [\![\Psi \rhd P \parallel !P \longmapsto (ROut\ M\ B);\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel !P)\ M\ B;\ guarded\ P]\!] \Longrightarrow Prop\ C\ \Psi\ (!P)\ M\ B$

shows $Prop\ C\ \Psi\ (!P)\ M\ (([\!|\nu * xvec|\!])N \prec' P')$
⟨*proof*⟩

lemma *bangTauInduct*[*consumes 1*, *case-names cPar1 cPar2 cComm1 cComm2 cBang*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and P' :: (*'a*, *'b*, *'c*) *psi*
and *Prop* :: *'d*::*fs-name* ⇒ *'b* ⇒ (*'a*, *'b*, *'c*) *psi* ⇒ (*'a*, *'b*, *'c*) *psi* ⇒ *bool*
and C :: *'d*

assumes $\Psi \rhd !P \longmapsto \tau \prec P'$
and *rPar1*: $\bigwedge P'\ C.\ \Psi \rhd P \longmapsto \tau \prec P' \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (P' \parallel !P)$
and *rPar2*: $\bigwedge P'\ C.\ [\![\Psi \rhd !P \longmapsto \tau \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ P']\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (P \parallel P')$
and *rComm1*: $\bigwedge M\ N\ P'\ K\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P';\ \Psi \rhd !P \longmapsto K(\!|\nu * xvec|\!)\langle N\rangle \prec P'';\ \Psi \vdash M \leftrightarrow K;$
$xvec\ \sharp*\ \Psi;\ xvec\ \sharp*\ P;\ xvec\ \sharp*\ M;\ xvec\ \sharp*\ K;\ xvec\ \sharp*\ C]\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ ((\!|\nu * xvec|\!)(P' \parallel P''))$
and *rComm2*: $\bigwedge M\ N\ P'\ K\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P';\ \Psi \rhd !P \longmapsto K(\!|N|\!) \prec P'';\ \Psi \vdash M \leftrightarrow K;$
$xvec\ \sharp*\ \Psi;\ xvec\ \sharp*\ P;\ xvec\ \sharp*\ M;\ xvec\ \sharp*\ K;\ xvec\ \sharp*\ C]\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ ((\!|\nu * xvec|\!)(P' \parallel P''))$
and *rBang*: $\bigwedge P'\ C.\ [\![\Psi \rhd P \parallel !P \longmapsto \tau \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel !P)\ P';\ guarded\ P]\!] \Longrightarrow Prop\ C\ \Psi\ (!P)\ P'$

shows $Prop\ C\ \Psi\ (!P)\ P'$

⟨*proof*⟩

lemma *bangInduct'*[*consumes 2*, *case-names cAlpha cPar1 cPar2 cComm1 cComm2 cBrMerge cBrComm1 cBrComm2 cBang*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and α :: *'a action*
and P' :: (*'a*, *'b*, *'c*) *psi*
and *Prop* :: *'d*::*fs-name* ⇒ *'b* ⇒ (*'a*, *'b*, *'c*) *psi* ⇒ *'a action* ⇒ (*'a*, *'b*, *'c*) *psi* ⇒ *bool*
and C :: *'d*::*fs-name*

assumes $\Psi \rhd\, !P \longmapsto \alpha \prec P'$
and $bn\ \alpha \sharp* subject\ \alpha$
and *rAlpha*: $\bigwedge \alpha\ P'\ p\ C.\ [\![bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* subject\ \alpha;\ bn\ \alpha \sharp* C;$
$set\ p \subseteq set(bn\ \alpha) \times set(bn(p \cdot \alpha));\ distinctPerm\ p;$
$bn(p \cdot \alpha) \sharp* \alpha;\ bn(p \cdot \alpha) \sharp* P';\ Prop\ C\ \Psi\ (P \parallel !P)\ \alpha\ P']\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel !P)\ (p \cdot \alpha)\ (p \cdot P')$
and *rPar1*: $\bigwedge \alpha\ P'\ C.$
$[\![\Psi \rhd P \longmapsto \alpha \prec P';\ bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* subject\ \alpha;\ bn\ \alpha \sharp* C;\ distinct(bn\ \alpha)]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel !P)\ \alpha\ (P' \parallel !P)$
and *rPar2*: $\bigwedge \alpha\ P'\ C.$
$[\![\Psi \rhd\, !P \longmapsto \alpha \prec P';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ \alpha\ P';$
$bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* subject\ \alpha;\ bn\ \alpha \sharp* C;\ distinct(bn\ \alpha)]\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel !P)\ \alpha\ (P \parallel P')$
and *rComm1*: $\bigwedge M\ N\ P'\ K\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P';\ \Psi \rhd\, !P \longmapsto K(\!|\nu*xvec|\!)\langle N\rangle \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (K(\!|\nu*xvec|\!)\langle N\rangle)\ P'';\ \Psi \vdash M \leftrightarrow K;$
$xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* K;\ xvec \sharp* C;\ distinct\ xvec]\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (\tau)\ ((\!|\nu*xvec|\!)(P' \parallel P''))$
and *rComm2*: $\bigwedge M\ xvec\ N\ P'\ K\ P''\ C.\ [\![\Psi \rhd P \longmapsto M(\!|\nu*xvec|\!)\langle N\rangle \prec P';\ \Psi \rhd\, !P \longmapsto K(\!|N|\!) \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (K(\!|N|\!))\ P'';\ \Psi \vdash M \leftrightarrow K;$
$xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* K;\ xvec \sharp* C;\ distinct\ xvec]\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (\tau)\ ((\!|\nu*xvec|\!)(P' \parallel P''))$
and *rBrMerge*: $\bigwedge M\ N\ P'\ P''\ C.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \Psi \rhd\, !P \longmapsto ¿M(\!|N|\!) \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (¿M(\!|N|\!))\ P']\!] \Longrightarrow$
$Prop\ C\ \Psi\ (P \parallel !P)\ (¿M(\!|N|\!))\ (P' \parallel P'')$
and *rBrComm1*: $\bigwedge M\ N\ P'\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ \Psi \rhd\, !P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (¡M(\!|\nu*xvec|\!)\langle N\rangle)\ P'';$
$xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* C;\ distinct\ xvec]\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (¡M(\!|\nu*xvec|\!)\langle N\rangle)\ (P' \parallel P'')$
and *rBrComm2*: $\bigwedge M\ N\ P'\ xvec\ P''\ C.\ [\![\Psi \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P';\ \Psi \rhd\, !P \longmapsto ¿M(\!|N|\!) \prec P'';\ \bigwedge C.\ Prop\ C\ \Psi\ (!P)\ (¿M(\!|N|\!))\ P'';$
$xvec \sharp* \Psi;\ xvec \sharp* P;\ xvec \sharp* M;\ xvec \sharp* C;\ distinct\ xvec]\!] \Longrightarrow Prop\ C\ \Psi\ (P \parallel !P)\ (¡M(\!|\nu*xvec|\!)\langle N\rangle)\ (P' \parallel P'')$
and *rBang*: $\bigwedge \alpha\ P'\ C.$
$[\![\Psi \rhd P \parallel !P \longmapsto \alpha \prec P';\ guarded\ P;\ \bigwedge C.\ Prop\ C\ \Psi\ (P \parallel !P)\ \alpha$

P'; *guarded* P; *distinct*(*bn* α)$\rrbracket \Longrightarrow$
Prop C Ψ $(!P)$ α P'
shows *Prop* C Ψ $(!P)$ α P'
⟨*proof*⟩

lemma *brCommInAuxTooMuch*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and R :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and R' :: $('a, 'b, 'c)$ *psi*
and A_R :: *name list*
and Ψ_R :: $'b$
and A_P :: *name list*
and Ψ_P :: $'b$
and A_Q :: *name list*

assumes *RTrans*: $\Psi \otimes \Psi_Q \rhd R \longmapsto ¿ M(\!|N|\!) \prec R'$
and *FrR*: *extractFrame* $R = \langle A_R, \Psi_R\rangle$
and *distinct* A_R
and *QimpP*: $\langle A_Q, (\Psi \otimes \Psi_Q) \otimes \Psi_R\rangle \hookrightarrow_F \langle A_P, (\Psi \otimes \Psi_P) \otimes \Psi_R\rangle$
and $A_R \sharp* A_P$
and $A_R \sharp* A_Q$
and $A_R \sharp* \Psi$
and $A_R \sharp* \Psi_P$
and $A_R \sharp* \Psi_Q$
and $A_R \sharp* (\Psi \otimes \Psi_Q)$
and $A_R \sharp* R$
and $A_R \sharp* M$
and $A_P \sharp* R$
and $A_P \sharp* M$
and $A_Q \sharp* R$
and $A_Q \sharp* M$

shows $\Psi \otimes \Psi_P \rhd R \longmapsto ¿ M(\!|N|\!) \prec R'$
⟨*proof*⟩

lemma *brCommInAux*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and R :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$
and R' :: $('a, 'b, 'c)$ *psi*
and A_R :: *name list*
and Ψ_R :: $'b$
and A_P :: *name list*
and Ψ_P :: $'b$

and A_Q :: *name list*

assumes *RTrans*: $\Psi \otimes \Psi_Q \rhd R \longmapsto ¿M(\!|N|\!) \prec R'$
and *FrR*: *extractFrame* $R = \langle A_R, \Psi_R \rangle$
and *distinct* A_R
and *QimpP*: $\langle A_Q, (\Psi \otimes \Psi_Q) \otimes \Psi_R \rangle \hookrightarrow_F \langle A_P, (\Psi \otimes \Psi_P) \otimes \Psi_R \rangle$
and $A_R \sharp* A_P$
and $A_R \sharp* A_Q$
and $A_R \sharp* \Psi$
and $A_R \sharp* \Psi_P$
and $A_R \sharp* \Psi_Q$
and $A_R \sharp* R$
and $A_R \sharp* M$
and $A_P \sharp* R$
and $A_P \sharp* M$
and $A_Q \sharp* R$
and $A_Q \sharp* M$

shows $\Psi \otimes \Psi_P \rhd R \longmapsto ¿M(\!|N|\!) \prec R'$
⟨*proof*⟩

lemma *brCommOutAuxTooMuch*:
fixes Ψ :: *'b*
and Ψ_Q :: *'b*
and R :: (*'a*, *'b*, *'c*) *psi*
and M :: *'a*
and *xvec* :: *name list*
and N :: *'a*
and R' :: (*'a*, *'b*, *'c*) *psi*
and A_R :: *name list*
and Ψ_R :: *'b*
and A_P :: *name list*
and Ψ_P :: *'b*
and A_Q :: *name list*

assumes *RTrans*: $\Psi \otimes \Psi_Q \rhd R \longmapsto RBrOut\ M\ (\!|\nu * xvec|\!)N \prec' R'$
and *FrR*: *extractFrame* $R = \langle A_R, \Psi_R \rangle$
and *distinct* A_R
and *QimpP*: $\langle A_Q, (\Psi \otimes \Psi_Q) \otimes \Psi_R \rangle \hookrightarrow_F \langle A_P, (\Psi \otimes \Psi_P) \otimes \Psi_R \rangle$
and $A_R \sharp* A_P$
and $A_R \sharp* A_Q$
and $A_R \sharp* \Psi$
and $A_R \sharp* \Psi_P$
and $A_R \sharp* \Psi_Q$
and $A_R \sharp* (\Psi \otimes \Psi_Q)$
and $A_R \sharp* R$
and $A_R \sharp* M$
and $A_P \sharp* R$
and $A_P \sharp* M$

and $A_Q \sharp* R$
and $A_Q \sharp* M$

shows $\Psi \otimes \Psi_P \rhd R \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec R'$
⟨*proof*⟩

lemma *brCommOutAux*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and R :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and R' :: $('a, 'b, 'c)$ *psi*
and A_R :: *name list*
and Ψ_R :: $'b$
and A_P :: *name list*
and Ψ_P :: $'b$
and A_Q :: *name list*

assumes *RTrans*: $\Psi \otimes \Psi_Q \rhd R \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec R'$
and *FrR*: *extractFrame* $R = \langle A_R, \Psi_R\rangle$
and *distinct* A_R
and *QimpP*: $\langle A_Q, (\Psi \otimes \Psi_Q) \otimes \Psi_R\rangle \hookrightarrow_F \langle A_P, (\Psi \otimes \Psi_P) \otimes \Psi_R\rangle$
and $A_R \sharp* A_P$
and $A_R \sharp* A_Q$
and $A_R \sharp* \Psi$
and $A_R \sharp* \Psi_P$
and $A_R \sharp* \Psi_Q$
and $A_R \sharp* R$
and $A_R \sharp* M$
and $A_P \sharp* R$
and $A_P \sharp* M$
and $A_Q \sharp* R$
and $A_Q \sharp* M$

shows $\Psi \otimes \Psi_P \rhd R \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec R'$
⟨*proof*⟩

lemma *comm1Aux*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and R :: $('a, 'b, 'c)$ *psi*
and K :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and R' :: $('a, 'b, 'c)$ *psi*
and A_R :: *name list*
and Ψ_R :: $'b$

and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and L :: $'a$
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*
and Ψ_P :: $'b$
and A_Q :: *name list*

assumes *RTrans*: $\Psi \otimes \Psi_Q \rhd R \longmapsto K\langle\!|\nu * xvec|\!\rangle\langle N\rangle \prec R'$
and *FrR*: *extractFrame* $R = \langle A_R, \Psi_R\rangle$
and *PTrans*: $\Psi \otimes \Psi_R \rhd P \longmapsto M\langle\!|L|\!\rangle \prec P'$
and *MeqK*: $\Psi \otimes \Psi_Q \otimes \Psi_R \vdash M \leftrightarrow K$
and *PeqQ*: $\langle A_Q, (\Psi \otimes \Psi_Q) \otimes \Psi_R\rangle \hookrightarrow_F \langle A_P, (\Psi \otimes \Psi_P) \otimes \Psi_R\rangle$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *FrQ*: *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$
and *distinct* A_P
and *distinct* A_R
and $A_R \sharp* A_P$
and $A_R \sharp* A_Q$
and $A_R \sharp* \Psi$
and $A_R \sharp* P$
and $A_R \sharp* Q$
and $A_R \sharp* R$
and $A_R \sharp* K$
and $A_P \sharp* \Psi$
and $A_P \sharp* R$
and $A_P \sharp* P$
and $A_P \sharp* M$
and $A_Q \sharp* R$
and $A_Q \sharp* M$

obtains K' **where** $\Psi \otimes \Psi_P \rhd R \longmapsto K'\langle\!|\nu * xvec|\!\rangle\langle N\rangle \prec R'$ **and** $\Psi \otimes \Psi_P \otimes \Psi_R \vdash M \leftrightarrow K'$ **and** $A_R \sharp* K'$
⟨*proof*⟩

lemma *comm2Aux*:
fixes Ψ :: $'b$
and Ψ_Q :: $'b$
and R :: $('a, 'b, 'c)$ *psi*
and K :: $'a$
and N :: $'a$
and R' :: $('a, 'b, 'c)$ *psi*
and A_R :: *name list*
and Ψ_R :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and P' :: $('a, 'b, 'c)$ *psi*
and A_P :: *name list*

and Ψ_P :: *'b*
and A_Q :: *name list*

assumes *RTrans*: $\Psi \otimes \Psi_Q \rhd R \longmapsto K(\!|N|\!) \prec R'$
and *FrR*: *extractFrame* $R = \langle A_R, \Psi_R\rangle$
and *PTrans*: $\Psi \otimes \Psi_R \rhd P \longmapsto M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and *MeqK*: $\Psi \otimes \Psi_Q \otimes \Psi_R \vdash M \leftrightarrow K$
and *QimpP*: $\langle A_Q, (\Psi \otimes \Psi_Q) \otimes \Psi_R\rangle \hookrightarrow_F \langle A_P, (\Psi \otimes \Psi_P) \otimes \Psi_R\rangle$
and *FrP*: *extractFrame* $P = \langle A_P, \Psi_P\rangle$
and *FrQ*: *extractFrame* $Q = \langle A_Q, \Psi_Q\rangle$
and *distinct* A_P
and *distinct* A_R
and $A_R \sharp* A_P$
and $A_R \sharp* A_Q$
and $A_R \sharp* \Psi$
and $A_R \sharp* P$
and $A_R \sharp* Q$
and $A_R \sharp* R$
and $A_R \sharp* K$
and $A_P \sharp* \Psi$
and $A_P \sharp* R$
and $A_P \sharp* P$
and $A_P \sharp* M$
and $A_Q \sharp* R$
and $A_Q \sharp* M$
and $A_R \sharp* xvec$
and $xvec \sharp* M$

obtains K' **where** $\Psi \otimes \Psi_P \rhd R \longmapsto K'(\!|N|\!) \prec R'$ **and** $\Psi \otimes \Psi_P \otimes \Psi_R \vdash M \leftrightarrow K'$ **and** $A_R \sharp* K'$
⟨*proof*⟩

end

end
theory *Simulation*
imports *Semantics*
begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Simulation* from [1].

context *env* **begin**

definition
simulation :: *'b* ⇒ (*'a*, *'b*, *'c*) *psi* ⇒
(*'b* × (*'a*, *'b*, *'c*) *psi* × (*'a*, *'b*, *'c*) *psi*) *set* ⇒
(*'a*, *'b*, *'c*) *psi* ⇒ *bool* (‹- ▷ - ⇝[-] -› [*80*, *80*, *80*, *80*] *80*)
where
$\Psi \rhd P \rightsquigarrow[Rel]\ Q \equiv \forall \alpha\ Q'.\ \Psi \rhd Q \longmapsto \alpha \prec Q' \longrightarrow bn\ \alpha \sharp* \Psi \longrightarrow bn\ \alpha \sharp* P$

$\longrightarrow (\exists\, P'.\ \Psi \rhd P \longmapsto \alpha \prec P' \wedge (\Psi,\ P',\ Q') \in Rel)$

abbreviation
simulationNilJudge (‹- ⇝[-] -› [*80*, *80*, *80*] *80*) **where** $P \rightsquigarrow [Rel]\ Q \equiv SBottom' \rhd P \rightsquigarrow [Rel]\ Q$

lemma *simI*[*consumes 1*, *case-names cSim*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and *Rel* :: (*'b* × (*'a*, *'b*, *'c*) *psi* × (*'a*, *'b*, *'c*) *psi*) *set*
and Q :: (*'a*, *'b*, *'c*) *psi*
and C :: *'d*::*fs-name*

assumes *Eqvt*: *eqvt Rel*
and *Sim*: $\bigwedge \alpha\ Q'.\ [\![\Psi \rhd Q \longmapsto \alpha \prec Q';\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* Q;\ bn\ \alpha \sharp* \Psi;$ *distinct*(*bn* α);
$bn\ \alpha \sharp* (subject\ \alpha);\ bn\ \alpha \sharp* C]\!] \Longrightarrow \exists\, P'.\ \Psi \rhd P \longmapsto \alpha \prec P' \wedge (\Psi,\ P',\ Q') \in Rel$

shows $\Psi \rhd P \rightsquigarrow [Rel]\ Q$
⟨*proof*⟩

lemma *simI2*[*case-names cSim*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and *Rel* :: (*'b* × (*'a*, *'b*, *'c*) *psi* × (*'a*, *'b*, *'c*) *psi*) *set*
and Q :: (*'a*, *'b*, *'c*) *psi*
and C :: *'d*::*fs-name*

assumes *Sim*: $\bigwedge \alpha\ Q'.\ [\![\Psi \rhd Q \longmapsto \alpha \prec Q';\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* \Psi;\ distinct(bn\ \alpha)]\!]$
$\Longrightarrow \exists\, P'.\ \Psi \rhd P \longmapsto \alpha \prec P' \wedge (\Psi,\ P',\ Q') \in Rel$

shows $\Psi \rhd P \rightsquigarrow [Rel]\ Q$
⟨*proof*⟩

lemma *simIChainFresh*[*consumes 4*, *case-names cSim*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and *Rel* :: (*'b* × (*'a*, *'b*, *'c*) *psi* × (*'a*, *'b*, *'c*) *psi*) *set*
and Q :: (*'a*, *'b*, *'c*) *psi*
and *xvec* :: *name list*
and C :: *'d*::*fs-name*

assumes *Eqvt*: *eqvt Rel*
and *xvec* $\sharp*\ \Psi$
and *xvec* $\sharp*\ P$
and *xvec* $\sharp*\ Q$
and *Sim*: $\bigwedge \alpha\ Q'.\ [\![\Psi \rhd Q \longmapsto \alpha \prec Q';\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* Q;\ bn\ \alpha \sharp* \Psi;$
$bn\ \alpha \sharp* subject\ \alpha;\ distinct(bn\ \alpha);\ bn\ \alpha \sharp* C;\ xvec \sharp* \alpha;\ xvec$

$\sharp* \ Q'[\!] \Longrightarrow$
$\exists P'.\ \Psi \rhd P \longmapsto \alpha \prec P' \wedge (\Psi,\ P',\ Q') \in Rel$

shows $\Psi \rhd P \rightsquigarrow [Rel]\ Q$
⟨*proof*⟩

lemma *simIFresh*[*consumes* 4, *case-names cSim*]:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$
and Q :: $('a, 'b, 'c)\ psi$
and x :: *name*
and C :: $'d$::*fs-name*

assumes *Eqvt*: *eqvt Rel*
and $x \sharp \Psi$
and $x \sharp P$
and $x \sharp Q$
and $\bigwedge \alpha\ Q'.\ [\![\Psi \rhd Q \longmapsto \alpha \prec Q';\ bn\ \alpha \sharp* P;\ bn\ \alpha \sharp* Q;\ bn\ \alpha \sharp* \Psi;$
$bn\ \alpha \sharp* subject\ \alpha;\ distinct(bn\ \alpha);\ bn\ \alpha \sharp* C;\ x \sharp \alpha;\ x \sharp Q'[\!]] \Longrightarrow$
$\exists P'.\ \Psi \rhd P \longmapsto \alpha \prec P' \wedge (\Psi,\ P',\ Q') \in Rel$

shows $\Psi \rhd P \rightsquigarrow [Rel]\ Q$
⟨*proof*⟩

lemma *simE*:
fixes F :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$
and Q :: $('a, 'b, 'c)\ psi$

assumes $\Psi \rhd P \rightsquigarrow [Rel]\ Q$

shows $\bigwedge \alpha\ Q'.\ [\![\Psi \rhd Q \longmapsto \alpha \prec Q';\ bn\ \alpha \sharp* \Psi;\ bn\ \alpha \sharp* P]\!] \Longrightarrow \exists P'.\ \Psi \rhd P \longmapsto \alpha$
$\prec P' \wedge (\Psi,\ P',\ Q') \in Rel$
⟨*proof*⟩

lemma *simClosedAux*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$
and Q :: $('a, 'b, 'c)\ psi$
and p :: *name prm*

assumes *EqvtRel*: *eqvt Rel*
and *PSimQ*: $\Psi \rhd P \rightsquigarrow [Rel]\ Q$

shows $(p \cdot \Psi) \rhd (p \cdot P) \rightsquigarrow [Rel]\ (p \cdot Q)$
⟨*proof*⟩

lemma *simClosed*:
  fixes Ψ :: $'b$
    and P :: $('a, 'b, 'c)$ *psi*
    and *Rel* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
    and Q :: $('a, 'b, 'c)$ *psi*
    and p :: *name prm*

assumes *EqvtRel*: *eqvt Rel*

shows $\Psi \rhd P \rightsquigarrow[Rel]\ Q \Longrightarrow (p \cdot \Psi) \rhd (p \cdot P) \rightsquigarrow[Rel]\ (p \cdot Q)$
  and $P \rightsquigarrow[Rel]\ Q \Longrightarrow (p \cdot P) \rightsquigarrow[Rel]\ (p \cdot Q)$
  ⟨*proof*⟩

lemma *reflexive*:
  fixes *Rel* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
    and Ψ :: $'b$
    and P :: $('a, 'b, 'c)$ *psi*

assumes $\{(\Psi, P, P) \mid \Psi\ P.\ True\} \subseteq Rel$

shows $\Psi \rhd P \rightsquigarrow[Rel]\ P$
  ⟨*proof*⟩

lemma *transitive*:
  fixes Ψ :: $'b$
    and P :: $('a, 'b, 'c)$ *psi*
    and *Rel* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
    and Q :: $('a, 'b, 'c)$ *psi*
    and *Rel'* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
    and R :: $('a, 'b, 'c)$ *psi*
    and *Rel''* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*

assumes *PSimQ*: $\Psi \rhd P \rightsquigarrow[Rel]\ Q$
  and *QSimR*: $\Psi \rhd Q \rightsquigarrow[Rel']\ R$
  and *Eqvt*: *eqvt Rel''*
  and *Set*: $\{(\Psi, P, R) \mid \Psi\ P\ R.\ \exists Q.\ (\Psi, P, Q) \in Rel \wedge (\Psi, Q, R) \in Rel'\} \subseteq Rel''$

shows $\Psi \rhd P \rightsquigarrow[Rel'']\ R$
  ⟨*proof*⟩

lemma *statEqSim*:
  fixes Ψ :: $'b$
    and P :: $('a, 'b, 'c)$ *psi*
    and *Rel* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
    and Q :: $('a, 'b, 'c)$ *psi*
    and Ψ' :: $'b$

assumes *PSimQ*: $\Psi \rhd P \rightsquigarrow[Rel]\ Q$

and *eqvt Rel′*
and $\Psi \simeq \Psi'$
and *C1*: $\bigwedge \Psi'' \; R \; S \; \Psi'''. \; [\![(\Psi'', R, S) \in Rel; \Psi'' \simeq \Psi''']\!] \Longrightarrow (\Psi''', R, S) \in Rel'$

shows $\Psi' \rhd P \rightsquigarrow [Rel'] \; Q$
⟨*proof*⟩

lemma *monotonic*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c) \; psi$
and $A :: ('b \times ('a, 'b, 'c) \; psi \times ('a, 'b, 'c) \; psi) \; set$
and $Q :: ('a, 'b, 'c) \; psi$
and $B :: ('b \times ('a, 'b, 'c) \; psi \times ('a, 'b, 'c) \; psi) \; set$

assumes $\Psi \rhd P \rightsquigarrow [A] \; Q$
and $A \subseteq B$

shows $\Psi \rhd P \rightsquigarrow [B] \; Q$
⟨*proof*⟩

end

end
theory *Bisimulation*
imports *Simulation*
begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Bisimulation* from [1].

context *env* **begin**

lemma *monoCoinduct*: $\bigwedge x \; y \; xa \; xb \; xc \; P \; Q \; \Psi.$
$x \leq y \Longrightarrow$
$(\Psi \rhd Q \rightsquigarrow [\{(xc, xb, xa). \; x \; xc \; xb \; xa\}] \; P) \longrightarrow$
$(\Psi \rhd Q \rightsquigarrow [\{(xb, xa, xc). \; y \; xb \; xa \; xc\}] \; P)$
⟨*proof*⟩

coinductive-set *bisim* :: $('b \times ('a, 'b, 'c) \; psi \times ('a, 'b, 'c) \; psi) \; set$
where
step: $[\![(insertAssertion \; (extractFrame \; P)) \; \Psi \simeq_F (insertAssertion \; (extractFrame \; Q) \; \Psi);$
$\Psi \rhd P \rightsquigarrow [bisim] \; Q;$
$\forall \Psi'. \; (\Psi \otimes \Psi', \; P, \; Q) \in bisim; \; (\Psi, \; Q, \; P) \in bisim]\!] \Longrightarrow (\Psi, \; P, \; Q) \in bisim$
monos *monoCoinduct*

abbreviation
bisimJudge ($\langle$- $\rhd$ - $\sim$ -$\rangle$ [70, 70, 70] 65) **where** $\Psi \rhd P \sim Q \equiv (\Psi, \; P, \; Q) \in bisim$
abbreviation

bisimNilJudge (‹- ∼ -› [*70*, *70*] *65*) **where** $P \sim Q \equiv SBottom' \rhd P \sim Q$

lemma *bisimCoinductAux*[*consumes 1*]:
fixes $F :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes $(\Psi, P, Q) \in X$
and $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi \wedge$
$(\Psi \rhd P \rightsquigarrow [(X \cup bisim)]\ Q) \wedge$
$(\forall \Psi'.\ (\Psi \otimes \Psi', P, Q) \in X \vee (\Psi \otimes \Psi', P, Q) \in bisim) \wedge$
$((\Psi, Q, P) \in X \vee (\Psi, Q, P) \in bisim)$

shows $(\Psi, P, Q) \in bisim$
⟨*proof*⟩

lemma *bisimCoinduct*[*consumes 1*, *case-names cStatEq cSim cExt cSym*]:
fixes $F :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes $(\Psi, P, Q) \in X$
and $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow insertAssertion\ (extractFrame\ R)\ \Psi' \simeq_F insertAssertion\ (extractFrame\ S)\ \Psi'$
and $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow \Psi' \rhd R \rightsquigarrow [(X \cup bisim)]\ S$
and $\bigwedge \Psi'\ R\ S\ \Psi''.\ (\Psi', R, S) \in X \Longrightarrow (\Psi' \otimes \Psi'', R, S) \in X \vee (\Psi' \otimes \Psi'', R, S) \in bisim$
and $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow (\Psi', S, R) \in X \vee (\Psi', S, R) \in bisim$

shows $(\Psi, P, Q) \in bisim$
⟨*proof*⟩

lemma *bisimWeakCoinductAux*[*consumes 1*]:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes $(\Psi, P, Q) \in X$
and $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi \wedge$
$\Psi \rhd P \rightsquigarrow [X]\ Q \wedge$
$(\forall \Psi'.\ (\Psi \otimes \Psi', P, Q) \in X) \wedge (\Psi, Q, P) \in X$

shows $(\Psi, P, Q) \in bisim$

⟨*proof*⟩

lemma *bisimWeakCoinduct*[*consumes 1*, *case-names cStatEq cSim cExt cSym*]:
fixes $F :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes $(\Psi, P, Q) \in X$
and $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi$
and $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow \Psi \rhd P \rightsquigarrow[X]\ Q$
and $\bigwedge \Psi\ P\ Q\ \Psi'.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi \otimes \Psi', P, Q) \in X$
and $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi, Q, P) \in X$

shows $(\Psi, P, Q) \in bisim$
⟨*proof*⟩

lemma *bisimE*:
fixes $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $\Psi :: 'b$
and $\Psi' :: 'b$

assumes $(\Psi, P, Q) \in bisim$

shows $insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi$
and $\Psi \rhd P \rightsquigarrow[bisim]\ Q$
and $(\Psi \otimes \Psi', P, Q) \in bisim$
and $(\Psi, Q, P) \in bisim$
⟨*proof*⟩

lemma *bisimI*:
fixes $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $\Psi :: 'b$

assumes $insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi$
and $\Psi \rhd P \rightsquigarrow[bisim]\ Q$
and $\forall \Psi'.\ (\Psi \otimes \Psi', P, Q) \in bisim$
and $(\Psi, Q, P) \in bisim$

shows $(\Psi, P, Q) \in bisim$
⟨*proof*⟩

lemma *bisimReflexive*:
fixes $\Psi :: 'b$

and $P :: ('a, 'b, 'c)\ psi$

shows $\Psi \rhd P \sim P$
⟨*proof*⟩

lemma *bisimClosed*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $p :: name\ prm$

assumes *PBisimQ*: $\Psi \rhd P \sim Q$

shows $(p \cdot \Psi) \rhd (p \cdot P) \sim (p \cdot Q)$
⟨*proof*⟩

lemma *bisimEqvt*[*simp*]:
shows *eqvt bisim*
⟨*proof*⟩

lemma *statEqBisim*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $\Psi' :: 'b$

assumes $\Psi \rhd P \sim Q$
and $\Psi \simeq \Psi'$

shows $\Psi' \rhd P \sim Q$
⟨*proof*⟩

lemma *bisimTransitive*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $R :: ('a, 'b, 'c)\ psi$

assumes *PQ*: $\Psi \rhd P \sim Q$
and *QR*: $\Psi \rhd Q \sim R$

shows $\Psi \rhd P \sim R$
⟨*proof*⟩

lemma *weakTransitiveCoinduct*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$

and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes p: $(\Psi, P, Q) \in X$
and *Eqvt*: $eqvt\ X$
and *rStatEq*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi$
and *rSim*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow \Psi \rhd P \rightsquigarrow[(\{(\Psi, P, Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi, P', Q') \in X \wedge \Psi \rhd Q' \sim Q\})]\ Q$
and *rExt*: $\bigwedge \Psi\ P\ Q\ \Psi'.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi \otimes \Psi', P, Q) \in X$
and *rSym*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi, Q, P) \in X$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

lemma *weakTransitiveCoinduct'*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes p: $(\Psi, P, Q) \in X$
and *Eqvt*: $eqvt\ X$
and *rStatEq*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi$
and *rSim*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow \Psi \rhd P \rightsquigarrow[(\{(\Psi, P, Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi, P', Q') \in X \wedge \Psi \rhd Q' \sim Q\})]\ Q$
and *rExt*: $\bigwedge \Psi\ P\ Q\ \Psi'.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi \otimes \Psi', P, Q) \in X$
and *rSym*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi, Q, P) \in \{(\Psi, P, Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi, P', Q') \in X \wedge \Psi \rhd Q' \sim Q\}$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

lemma *weakTransitiveCoinduct''*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes p: $(\Psi, P, Q) \in X$
and *Eqvt*: $eqvt\ X$

and *rStatEq*: $\bigwedge\Psi\ P\ Q.\ (\Psi,\ P,\ Q) \in X \Longrightarrow \mathit{insertAssertion}\ (\mathit{extractFrame}\ P)\ \Psi \simeq_F \mathit{insertAssertion}\ (\mathit{extractFrame}\ Q)\ \Psi$
and *rSim*: $\bigwedge\Psi\ P\ Q.\ (\Psi,\ P,\ Q) \in X \Longrightarrow \Psi \rhd P \rightsquigarrow[(\{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi,\ P',\ Q') \in X \wedge \Psi \rhd Q' \sim Q\})]\ Q$
and *rExt*: $\bigwedge\Psi\ P\ Q\ \Psi'.\ (\Psi,\ P,\ Q) \in \{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi,\ P',\ Q') \in X \wedge \Psi \rhd Q' \sim Q\} \Longrightarrow (\Psi \otimes \Psi',\ P,\ Q) \in \{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi,\ P',\ Q') \in X \wedge \Psi \rhd Q' \sim Q\}$
and *rSym*: $\bigwedge\Psi\ P\ Q.\ (\Psi,\ P,\ Q) \in \{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi,\ P',\ Q') \in X \wedge \Psi \rhd Q' \sim Q\} \Longrightarrow (\Psi,\ Q,\ P) \in \{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi,\ P',\ Q') \in X \wedge \Psi \rhd Q' \sim Q\}$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

lemma *transitiveCoinduct*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes $\Psi :: 'b$
and $P :: ('a,\ 'b,\ 'c)\ psi$
and $Q :: ('a,\ 'b,\ 'c)\ psi$
and $X :: ('b \times ('a,\ 'b,\ 'c)\ psi \times ('a,\ 'b,\ 'c)\ psi)\ set$

assumes *p*: $(\Psi,\ P,\ Q) \in X$
and *Eqvt*: *eqvt X*
and *rStatEq*: $\bigwedge\Psi'\ R\ S.\ (\Psi',\ R,\ S) \in X \Longrightarrow \mathit{insertAssertion}\ (\mathit{extractFrame}\ R)\ \Psi' \simeq_F \mathit{insertAssertion}\ (\mathit{extractFrame}\ S)\ \Psi'$
and *rSim*: $\bigwedge\Psi'\ R\ S.\ (\Psi',\ R,\ S) \in X \Longrightarrow \Psi' \rhd R \rightsquigarrow[(\{(\Psi',\ R,\ S) \mid \Psi'\ R\ R'\ S'\ S.\ \Psi' \rhd R \sim R' \wedge ((\Psi',\ R',\ S') \in X \vee \Psi' \rhd R' \sim S') \wedge \Psi' \rhd S' \sim S\})]\ S$
and *rExt*: $\bigwedge\Psi'\ R\ S\ \Psi''.\ (\Psi',\ R,\ S) \in X \Longrightarrow (\Psi' \otimes \Psi'',\ R,\ S) \in X \vee \Psi' \otimes \Psi'' \rhd R \sim S$
and *rSym*: $\bigwedge\Psi'\ R\ S.\ (\Psi',\ R,\ S) \in X \Longrightarrow (\Psi',\ S,\ R) \in X \vee \Psi' \rhd S \sim R$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

lemma *transitiveCoinduct'*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes $\Psi :: 'b$
and $P :: ('a,\ 'b,\ 'c)\ psi$
and $Q :: ('a,\ 'b,\ 'c)\ psi$
and $X :: ('b \times ('a,\ 'b,\ 'c)\ psi \times ('a,\ 'b,\ 'c)\ psi)\ set$

assumes p: $(\Psi,\ P,\ Q) \in X$
and *Eqvt*: *eqvt* X
and *rStatEq*: $\bigwedge \Psi\ P\ Q.\ (\Psi,\ P,\ Q) \in X \Longrightarrow$ *insertAssertion* (*extractFrame* P) Ψ $\simeq_F$ *insertAssertion* (*extractFrame* Q) Ψ
and *rSim*: $\bigwedge \Psi\ P\ Q.\ (\Psi,\ P,\ Q) \in X \Longrightarrow \Psi \rhd P \rightsquigarrow [(\{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi,\ P',\ Q') \in (X \cup bisim) \wedge \Psi \rhd Q' \sim Q\})]\ Q$
and *rExt*: $\bigwedge \Psi\ P\ Q\ \Psi'.\ (\Psi,\ P,\ Q) \in X \Longrightarrow (\Psi \otimes \Psi',\ P,\ Q) \in X \vee \Psi \otimes \Psi' \rhd P \sim Q$
and *rSym*: $\bigwedge \Psi\ P\ Q.\ (\Psi,\ P,\ Q) \in X \Longrightarrow (\Psi,\ Q,\ P) \in \{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge ((\Psi,\ P',\ Q') \in (X \cup bisim)) \wedge \Psi \rhd Q' \sim Q\}$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

lemma *bisimSymmetric*:
fixes $\Psi :: 'b$
and $P :: ('a,\ 'b,\ 'c)\ psi$
and $Q :: ('a,\ 'b,\ 'c)\ psi$

assumes $\Psi \rhd P \sim Q$

shows $\Psi \rhd Q \sim P$
⟨*proof*⟩

lemma *eqvtTrans*[*intro*]:
assumes *eqvt* X

shows *eqvt* $\{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge ((\Psi,\ P',\ Q') \in X \vee \Psi \rhd P' \sim Q') \wedge \Psi \rhd Q' \sim Q\}$
⟨*proof*⟩

lemma *eqvtWeakTrans*[*intro*]:
assumes *eqvt* X

shows *eqvt* $\{(\Psi,\ P,\ Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi,\ P',\ Q') \in X \wedge \Psi \rhd Q' \sim Q\}$
⟨*proof*⟩

inductive-set
rel-trancl :: $('b \times ('a,'b,'c)\ psi \times ('a,'b,'c)\ psi)\ set \Rightarrow ('b \times ('a,'b,'c)\ psi \times ('a,'b,'c)\ psi)\ set$ ($\langle$(-*)$\rangle$ [1000] 999)
for $r :: 'b \times ('a,'b,'c)\ psi \times ('a,'b,'c)\ psi)\ set$
where
r-into-rel-trancl [*intro*, *Pure.intro*]: $(\Psi,\ P,Q) : r \Longrightarrow (\Psi,P,Q) : r^\star$
| *rel-trancl-into-rel-trancl* [*Pure.intro*]: $(\Psi,P,Q) : r^\star \Longrightarrow (\Psi,Q,R) : r \Longrightarrow (\Psi,P,R) : r^\star$

lemma *rel-trancl-transitive*:
assumes $(\Psi,P,Q) \in Rel^\star$
and $(\Psi,Q,R) \in Rel^\star$
shows $(\Psi,P,R) \in Rel^\star$
⟨*proof*⟩

lemma *rel-trancl-eqvt*:
assumes *eqvt X*
shows *eqvt*$(X^\star)$
⟨*proof*⟩

lemma *bisimStarWeakCoinduct*[*consumes 2*, *case-names cStatEq cSim cExt cSym*]:
fixes $F :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes $(\Psi, P, Q) \in X$
and *eqvt X*
and *rStatEq*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow insertAssertion\ (extractFrame\ R)\ \Psi' \simeq_F insertAssertion\ (extractFrame\ S)\ \Psi'$
and *rSim*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow \Psi' \rhd R \rightsquigarrow[X^\star]\ S$
and *rExt*: $\bigwedge \Psi'\ R\ S\ \Psi''.\ (\Psi', R, S) \in X \Longrightarrow (\Psi' \otimes \Psi'', R, S) \in X$
and *rSym*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow (\Psi', S, R) \in X$

shows $(\Psi, P, Q) \in bisim$
⟨*proof*⟩

lemma *weakTransitiveStarCoinduct*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $X :: ('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes *p*: $(\Psi, P, Q) \in X$
and *Eqvt*: *eqvt X*
and *rStatEq*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi$
and *rSim*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow \Psi \rhd P \rightsquigarrow[(\{(\Psi, P, Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi, P', Q') \in X \wedge \Psi \rhd Q' \sim Q\})^\star]\ Q$
and *rExt*: $\bigwedge \Psi\ P\ Q\ \Psi'.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi \otimes \Psi', P, Q) \in X$
and *rSym*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi, Q, P) \in X$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

lemma *weakTransitiveStarCoinduct'*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and Q :: (*'a*, *'b*, *'c*) *psi*
and X :: (*'b* × (*'a*, *'b*, *'c*) *psi* × (*'a*, *'b*, *'c*) *psi*) *set*

assumes *p*: $(\Psi, P, Q) \in X$
and *Eqvt*: *eqvt X*
and *rStatEq*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow$ *insertAssertion* (*extractFrame P*) $\Psi \simeq_F$ *insertAssertion* (*extractFrame Q*) Ψ
and *rSim*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow \Psi \rhd P \rightsquigarrow [(\{(\Psi, P, Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi, P', Q') \in X \wedge \Psi \rhd Q' \sim Q\})^\star]\ Q$
and *rExt*: $\bigwedge \Psi\ P\ Q\ \Psi'.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi \otimes \Psi', P, Q) \in X$
and *rSym*: $\bigwedge \Psi\ P\ Q.\ (\Psi, P, Q) \in X \Longrightarrow (\Psi, Q, P) \in \{(\Psi, P, Q) \mid \Psi\ P\ P'\ Q'\ Q.\ \Psi \rhd P \sim P' \wedge (\Psi, P', Q') \in X \wedge \Psi \rhd Q' \sim Q\}$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

lemma *transitiveStarCoinduct*[*case-names cStatEq cSim cExt cSym*, *case-conclusion bisim step*, *consumes 2*]:
fixes Ψ :: *'b*
and P :: (*'a*, *'b*, *'c*) *psi*
and Q :: (*'a*, *'b*, *'c*) *psi*
and X :: (*'b* × (*'a*, *'b*, *'c*) *psi* × (*'a*, *'b*, *'c*) *psi*) *set*

assumes *p*: $(\Psi, P, Q) \in X$
and *Eqvt*: *eqvt X*
and *rStatEq*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow$ *insertAssertion* (*extractFrame R*) $\Psi' \simeq_F$ *insertAssertion* (*extractFrame S*) Ψ'
and *rSim*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow \Psi' \rhd R \rightsquigarrow [(\{(\Psi', R, S) \mid \Psi'\ R\ R'\ S'\ S.\ \Psi' \rhd R \sim R' \wedge ((\Psi', R', S') \in X \vee \Psi' \rhd R' \sim S') \wedge \Psi' \rhd S' \sim S\})^\star]\ S$
and *rExt*: $\bigwedge \Psi'\ R\ S\ \Psi''.\ (\Psi', R, S) \in X \Longrightarrow (\Psi' \otimes \Psi'', R, S) \in X \vee \Psi' \otimes \Psi'' \rhd R \sim S$
and *rSym*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in X \Longrightarrow (\Psi', S, R) \in X \vee \Psi' \rhd S \sim R$

shows $\Psi \rhd P \sim Q$
⟨*proof*⟩

end

end
theory *Sim-Pres*
imports *Simulation*
begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Sim_Pres* from [1].

context *env* **begin**

lemma *inputPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and *Rel* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
and Q :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$

assumes *PRelQ*: $\bigwedge Tvec.$ *length xvec* = *length Tvec* $\Longrightarrow (\Psi, P[xvec::=Tvec], Q[xvec::=Tvec]) \in Rel$

shows $\Psi \rhd M(\!|\lambda * xvec\ N|\!).P \rightsquigarrow[Rel]\ M(\!|\lambda * xvec\ N|\!).Q$
⟨*proof*⟩

lemma *outputPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and *Rel* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
and Q :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$

assumes *PRelQ*: $(\Psi, P, Q) \in Rel$

shows $\Psi \rhd M\langle N\rangle.P \rightsquigarrow[Rel]\ M\langle N\rangle.Q$
⟨*proof*⟩

lemma *casePres*:
fixes Ψ :: $'b$
and *CsP* :: $('c \times ('a, 'b, 'c)$ *psi*$)$ *list*
and *Rel* :: $('b \times ('a, 'b, 'c)$ *psi* $\times ('a, 'b, 'c)$ *psi*$)$ *set*
and *CsQ* :: $('c \times ('a, 'b, 'c)$ *psi*$)$ *list*
and M :: $'a$
and N :: $'a$

assumes *PRelQ*: $\bigwedge \varphi\ Q.\ (\varphi, Q) \in set\ CsQ \Longrightarrow \exists P.\ (\varphi, P) \in set\ CsP \wedge guarded\ P \wedge (\Psi, P, Q) \in Rel$
and *Sim*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R, S) \in Rel \Longrightarrow \Psi' \rhd R \rightsquigarrow[Rel]\ S$

and $Rel \subseteq Rel'$

shows $\Psi \rhd Cases\ CsP \rightsquigarrow[Rel']\ Cases\ CsQ$
⟨*proof*⟩

lemma *resPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$
and Q :: $('a, 'b, 'c)\ psi$
and x :: $name$
and Rel' :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes *PSimQ*: $\Psi \rhd P \rightsquigarrow[Rel]\ Q$
and $eqvt\ Rel'$
and $x \sharp \Psi$
and $Rel \subseteq Rel'$
and *C1*: $\bigwedge \Psi'\ R\ S\ yvec.\ [\![(\Psi', R, S) \in Rel; yvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (\!|\nu * yvec|\!)R, (\!|\nu * yvec|\!)S) \in Rel'$

shows $\Psi \rhd (\!|\nu x|\!)P \rightsquigarrow[Rel']\ (\!|\nu x|\!)Q$
⟨*proof*⟩

lemma *resChainPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$
and Q :: $('a, 'b, 'c)\ psi$
and $xvec$ :: $name\ list$

assumes *PSimQ*: $\Psi \rhd P \rightsquigarrow[Rel]\ Q$
and $eqvt\ Rel$
and $xvec \sharp* \Psi$
and *C1*: $\bigwedge \Psi'\ R\ S\ yvec.\ [\![(\Psi', R, S) \in Rel; yvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (\!|\nu * yvec|\!)R, (\!|\nu * yvec|\!)S) \in Rel$

shows $\Psi \rhd (\!|\nu * xvec|\!)P \rightsquigarrow[Rel]\ (\!|\nu * xvec|\!)Q$
⟨*proof*⟩

lemma *parPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$
and Q :: $('a, 'b, 'c)\ psi$
and R :: $('a, 'b, 'c)\ psi$
and Rel' :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes *PRelQ*: $\bigwedge A_R\ \Psi_R.\ [\![extractFrame\ R = \langle A_R, \Psi_R\rangle; A_R \sharp* \Psi; A_R \sharp* P; A_R \sharp* Q]\!] \Longrightarrow (\Psi \otimes \Psi_R, P, Q) \in Rel$

and *Eqvt*: *eqvt Rel*
and *Eqvt'*: *eqvt Rel'*

and *StatImp*: $\bigwedge \Psi'\ S\ T.\ (\Psi',\ S,\ T) \in Rel \Longrightarrow insertAssertion\ (extractFrame\ T)\ \Psi' \hookrightarrow_F insertAssertion\ (extractFrame\ S)\ \Psi'$
and *Sim*: $\bigwedge \Psi'\ S\ T.\ (\Psi',\ S,\ T) \in Rel \Longrightarrow \Psi' \rhd S \rightsquigarrow [Rel]\ T$
and *Ext*: $\bigwedge \Psi'\ S\ T\ \Psi''.\ [\![(\Psi',\ S,\ T) \in Rel]\!] \Longrightarrow (\Psi' \otimes \Psi'',\ S,\ T) \in Rel$

and *C1*: $\bigwedge \Psi'\ S\ T\ A_U\ \Psi_U\ U.\ [\![(\Psi' \otimes \Psi_U,\ S,\ T) \in Rel;\ extractFrame\ U = \langle A_U,\ \Psi_U\rangle;\ A_U \sharp* \Psi';\ A_U \sharp* S;\ A_U \sharp* T]\!] \Longrightarrow (\Psi',\ S \parallel U,\ T \parallel U) \in Rel'$
and *C2*: $\bigwedge \Psi'\ S\ T\ xvec.\ [\![(\Psi',\ S,\ T) \in Rel';\ xvec \sharp* \Psi']\!] \Longrightarrow (\Psi',\ (\!|\nu* xvec|\!)S,\ (\!|\nu* xvec|\!)\ T) \in Rel'$
and *C3*: $\bigwedge \Psi'\ S\ T\ \Psi''.\ [\![(\Psi',\ S,\ T) \in Rel;\ \Psi' \simeq \Psi']\!] \Longrightarrow (\Psi'',\ S,\ T) \in Rel$

shows $\Psi \rhd P \parallel R \rightsquigarrow [Rel']\ Q \parallel R$
⟨*proof*⟩

unbundle *no relcomp-syntax*

lemma *bangPres*:
fixes Ψ :: *'b*
and P :: *('a, 'b, 'c) psi*
and Q :: *('a, 'b, 'c) psi*
and R :: *('a, 'b, 'c) psi*
and *Rel* :: *('b × ('a, 'b, 'c) psi × ('a, 'b, 'c) psi) set*
and *Rel'* :: *('b × ('a, 'b, 'c) psi × ('a, 'b, 'c) psi) set*

assumes $(\Psi,\ P,\ Q) \in Rel$
and *eqvt Rel'*
and *guarded P*
and *guarded Q*
and *cSim*: $\bigwedge \Psi'\ S\ T.\ (\Psi',\ S,\ T) \in Rel \Longrightarrow \Psi' \rhd S \rightsquigarrow [Rel]\ T$
and *cExt*: $\bigwedge \Psi'\ S\ T\ \Psi''.\ (\Psi',\ S,\ T) \in Rel \Longrightarrow (\Psi' \otimes \Psi'',\ S,\ T) \in Rel$
and *cSym*: $\bigwedge \Psi'\ S\ T.\ (\Psi',\ S,\ T) \in Rel \Longrightarrow (\Psi',\ T,\ S) \in Rel$
and *StatEq*: $\bigwedge \Psi'\ S\ T\ \Psi''.\ [\![(\Psi',\ S,\ T) \in Rel;\ \Psi' \simeq \Psi']\!] \Longrightarrow (\Psi'',\ S,\ T) \in Rel$
and *Closed*: $\bigwedge \Psi'\ S\ T\ p.\ (\Psi',\ S,\ T) \in Rel \Longrightarrow ((p::name\ prm) \cdot \Psi',\ p \cdot S,\ p \cdot T) \in Rel$
and *Assoc*: $\bigwedge \Psi'\ S\ T\ U.\ (\Psi',\ S \parallel (T \parallel U),\ (S \parallel T) \parallel U) \in Rel$
and *ParPres*: $\bigwedge \Psi'\ S\ T\ U.\ (\Psi',\ S,\ T) \in Rel \Longrightarrow (\Psi',\ S \parallel U,\ T \parallel U) \in Rel$
and *FrameParPres*: $\bigwedge \Psi'\ \Psi_U\ S\ T\ U\ A_U.\ [\![(\Psi' \otimes \Psi_U,\ S,\ T) \in Rel;\ extractFrame\ U = \langle A_U,\ \Psi_U\rangle;\ A_U \sharp* \Psi';\ A_U \sharp* S;\ A_U \sharp* T]\!] \Longrightarrow (\Psi',\ U \parallel S,\ U \parallel T) \in Rel$
and *ResPres*: $\bigwedge \Psi'\ S\ T\ xvec.\ [\![(\Psi',\ S,\ T) \in Rel;\ xvec \sharp* \Psi']\!] \Longrightarrow (\Psi',\ (\!|\nu* xvec|\!)S,\ (\!|\nu* xvec|\!)\ T) \in Rel$
and *ScopeExt*: $\bigwedge xvec\ \Psi'\ S\ T.\ [\![xvec \sharp* \Psi';\ xvec \sharp* T]\!] \Longrightarrow (\Psi',\ (\!|\nu* xvec|\!)(S \parallel T),\ ((\!|\nu* xvec|\!)S) \parallel T) \in Rel$
and *Trans*: $\bigwedge \Psi'\ S\ T\ U.\ [\![(\Psi',\ S,\ T) \in Rel;\ (\Psi',\ T,\ U) \in Rel]\!] \Longrightarrow (\Psi',\ S,\ U) \in Rel$

and *Compose*: $\bigwedge\Psi'\ S\ T\ U\ O.\ [\![(\Psi',\ S,\ T) \in Rel;\ (\Psi',\ T,\ U) \in Rel';\ (\Psi',\ U,\ O) \in Rel]\!] \Longrightarrow (\Psi',\ S,\ O) \in Rel'$
and *C1*: $\bigwedge\Psi\ S\ T\ U.\ [\![(\Psi,\ S,\ T) \in Rel;\ guarded\ S;\ guarded\ T]\!] \Longrightarrow (\Psi,\ U \parallel !S,\ U \parallel !T) \in Rel'$
and *Der*: $\bigwedge\Psi'\ S\ \alpha\ S'\ T.\ [\![\Psi' \rhd !S \longmapsto\alpha \prec S';\ (\Psi',\ S,\ T) \in Rel;\ bn\ \alpha\ \sharp*\ \Psi';\ bn\ \alpha\ \sharp*\ S;\ bn\ \alpha\ \sharp*\ T;\ guarded\ T;\ bn\ \alpha\ \sharp*\ subject\ \alpha]\!] \Longrightarrow$
$\exists\ T'\ U\ O.\ \ \Psi' \rhd !T \longmapsto\alpha \prec T' \wedge (\Psi',\ S',\ U \parallel !S) \in Rel \wedge (\Psi',\ T',\ O \parallel !T) \in Rel \wedge$
$(\Psi',\ U,\ O) \in Rel \wedge ((supp\ U)::name\ set) \subseteq supp\ S' \wedge$
$((supp\ O)::name\ set) \subseteq supp\ T'$

shows $\Psi \rhd R \parallel !P \rightsquigarrow[Rel']\ R \parallel !Q$
⟨*proof*⟩

unbundle *relcomp-syntax*

end

end
theory *Sim-Struct-Cong*
imports *Simulation HOL−Library.Multiset*
begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Sim_Struct_Cong* from [1].

lemma *partitionListLeft*:
assumes $xs@ys{=}xs'@y\#ys'$
and $y \in set\ xs$
and $distinct(xs@ys)$

obtains *zs* **where** $xs = xs'@y\#zs$ **and** $ys'{=}zs@ys$
⟨*proof*⟩

lemma *partitionListRight*:

assumes $xs@ys{=}xs'@y\#ys'$
and $y \in set\ ys$
and $distinct(xs@ys)$

obtains *zs* **where** $xs' = xs@zs$ **and** $ys{=}zs@y\#ys'$
⟨*proof*⟩

context *env* **begin**

lemma *resOutputCases''''*[*consumes 8*, *case-names cOpen cRes*]:
fixes Ψ :: $'b$
and x :: *name*
and *zvec* :: *name list*

and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto \alpha \prec P'$
and *1*: $x \sharp \Psi$
and *2*: $x \sharp \alpha$
and *3*: $x \sharp P'$
and *4*: $bn\ \alpha \sharp* \Psi$
and *5*: $bn\ \alpha \sharp* P$
and *6*: $bn\ \alpha \sharp* subject\ \alpha$
and $\alpha = M(\!|\nu* zvec|\!)\langle N\rangle$
and *rOpen*: $\bigwedge M\ xvec\ yvec\ y\ N\ P'.\ [\![\Psi \rhd ([(x, y)] \cdot P) \longmapsto M(\!|\nu*(xvec@yvec)|\!)\langle N\rangle \prec P';\ y \in supp\ N;\ x \sharp N;\ x \sharp P';\ x \neq y;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp M;\ distinct\ xvec;\ distinct\ yvec;\ xvec \sharp* \Psi;\ y \sharp \Psi;\ yvec \sharp* \Psi;\ xvec \sharp* P;\ y \sharp P;\ yvec \sharp* P;\ xvec \sharp* M;\ y \sharp M;\ yvec \sharp* M;\ xvec \sharp* yvec]\!] \Longrightarrow Prop\ (M(\!|\nu*(xvec@y\#yvec)|\!)\langle N\rangle)\ P'$
and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto \alpha \prec P']\!] \Longrightarrow Prop\ \alpha\ ((\!|\nu x|\!)P')$

shows *Prop* $\alpha\ P'$
⟨*proof*⟩

lemma *resOutputCases''''''*[*consumes 7*, *case-names cOpen cRes*]:
fixes Ψ :: $'b$
and x :: *name*
and *zvec* :: *name list*
and P :: $('a, 'b, 'c)$ *psi*
and P' :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|\nu* zvec|\!)\langle N\rangle \prec P'$
and *1*: $x \sharp \Psi$
and $x \sharp M(\!|\nu* zvec|\!)\langle N\rangle$
and *3*: $x \sharp P'$
and $zvec \sharp* \Psi$
and $zvec \sharp* P$
and $zvec \sharp* M$
and *rOpen*: $\bigwedge M'\ xvec\ yvec\ y\ N'\ P'.\ [\![\Psi \rhd ([(x, y)] \cdot P) \longmapsto M'(\!|\nu*(xvec@yvec)|\!)\langle N'\rangle \prec P';\ y \in supp\ N';\ x \sharp N';\ x \sharp P';\ x \neq y;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp M';\ distinct\ xvec;\ distinct\ yvec;\ xvec \sharp* \Psi;\ y \sharp \Psi;\ yvec \sharp* \Psi;\ xvec \sharp* P;\ y \sharp P;\ yvec \sharp* P;\ xvec \sharp* M';\ y \sharp M';\ yvec \sharp* M';\ xvec \sharp* yvec;\ M'(\!|\nu*(xvec@y\#yvec)|\!)\langle N'\rangle = M(\!|\nu* zvec|\!)\langle N\rangle]\!] \Longrightarrow$

$$Prop\ P'$$

and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto M(\!|\nu * zvec|\!)\langle N\rangle \prec P']\!] \Longrightarrow Prop\ ((\!|\nu x|\!)P')$

shows $Prop\ P'$
⟨*proof*⟩

lemma *resBrOutputCases'*[*consumes* 7, *case-names cBrOpen cRes*]:
fixes Ψ :: $'b$
and x :: *name*
and *zvec* :: *name list*
and P :: $('a, 'b, 'c)$ *psi*
and P' :: $('a, 'b, 'c)$ *psi*
and C :: $'f$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto ¡M(\!|\nu * zvec|\!)\langle N\rangle \prec P'$
and *1*: $x \sharp \Psi$
and $x \sharp ¡M(\!|\nu * zvec|\!)\langle N\rangle$
and *3*: $x \sharp P'$
and $zvec \sharp* \Psi$
and $zvec \sharp* P$
and $zvec \sharp* M$
and *rBrOpen*: $\bigwedge M'\ xvec\ yvec\ y\ N'\ P'.\ [\![\Psi \rhd ([(x, y)] \cdot P) \longmapsto ¡M'(\!|\nu * (xvec @ yvec)|\!)\langle N'\rangle$
$\prec P'$; $y \in supp\ N'$;
$$x \sharp N';\ x \sharp P';\ x \neq y;\ y \sharp xvec;\ y \sharp yvec;\ y \sharp M';$$
distinct xvec; *distinct yvec*;
$$xvec \sharp* \Psi;\ y \sharp \Psi;\ yvec \sharp* \Psi;\ xvec \sharp* P;\ y \sharp P;$$
$yvec \sharp* P$; $xvec \sharp* M'$; $y \sharp M'$;
$$yvec \sharp* M';\ xvec \sharp* yvec;\ ¡M'(\!|\nu * (xvec @ y \# yvec)|\!)\langle N'\rangle$$
$= ¡M(\!|\nu * zvec|\!)\langle N\rangle]\!] \Longrightarrow$
$$Prop\ P'$$

and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto ¡M(\!|\nu * zvec|\!)\langle N\rangle \prec P']\!] \Longrightarrow Prop\ ((\!|\nu x|\!)P')$

shows $Prop\ P'$
⟨*proof*⟩

lemma *brOutputFreshSubject*:
fixes x::*name*
assumes $\Psi \rhd P \longmapsto ¡M(\!|\nu * xvec|\!)\langle N\rangle \prec P'$
and $xvec \sharp* M$
and $x \sharp P$
shows $x \sharp M$
⟨*proof*⟩

lemma *brInputFreshSubject*:
fixes x::*name*
assumes $\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P'$
and $x \sharp P$
shows $x \sharp M$
⟨*proof*⟩

lemma *resComm*:
fixes Ψ :: *'b*
and x :: *name*
and y :: *name*
and *Rel* :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$
and P :: $('a, 'b, 'c)\ psi$

assumes $x \sharp \Psi$
and $y \sharp \Psi$
and *eqvt Rel*
and *R1*: $\bigwedge \Psi'\ Q.\ (\Psi', Q, Q) \in Rel$
and *R2*: $\bigwedge \Psi'\ a\ b\ Q.\ [\![a \sharp \Psi'; b \sharp \Psi']\!] \Longrightarrow (\Psi', (\!|\nu a|\!)((\!|\nu b|\!)\, Q), (\!|\nu b|\!)((\!|\nu a|\!)\, Q)) \in Rel$
and *R3*: $\bigwedge \Psi'\ xvec\ yvec\ Q.\ [\![xvec \sharp* \Psi'; mset\ xvec = mset\ yvec]\!] \Longrightarrow (\Psi', (\!|\nu* xvec|\!)\, Q, (\!|\nu* yvec|\!)\, Q) \in Rel$

shows $\Psi \rhd (\!|\nu x|\!)((\!|\nu y|\!)P) \rightsquigarrow [Rel]\ (\!|\nu y|\!)((\!|\nu x|\!)P)$
⟨*proof*⟩

lemma *parAssocLeft*:
fixes Ψ :: *'b*
and P :: $('a, 'b, 'c)\ psi$
and Q :: $('a, 'b, 'c)\ psi$
and R :: $('a, 'b, 'c)\ psi$
and *Rel* :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes *eqvt Rel*
and *C1*: $\bigwedge \Psi'\ S\ T\ U.\ (\Psi, (S \parallel T) \parallel U, S \parallel (T \parallel U)) \in Rel$
and *C2*: $\bigwedge xvec\ \Psi'\ S\ T\ U.\ [\![xvec \sharp* \Psi'; xvec \sharp* S]\!] \Longrightarrow (\Psi', (\!|\nu* xvec|\!)((S \parallel T) \parallel U), S \parallel ((\!|\nu* xvec|\!)(T \parallel U))) \in Rel$
and *C3*: $\bigwedge xvec\ \Psi'\ S\ T\ U.\ [\![xvec \sharp* \Psi'; xvec \sharp* U]\!] \Longrightarrow (\Psi', ((\!|\nu* xvec|\!)(S \parallel T)) \parallel U, (\!|\nu* xvec|\!)(S \parallel (T \parallel U))) \in Rel$
and *C4*: $\bigwedge \Psi'\ S\ T\ xvec.\ [\![(\Psi', S, T) \in Rel; xvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (\!|\nu* xvec|\!)S, (\!|\nu* xvec|\!)\, T) \in Rel$

shows $\Psi \rhd (P \parallel Q) \parallel R \rightsquigarrow [Rel]\ P \parallel (Q \parallel R)$
⟨*proof*⟩

lemma *parNilLeft*:
fixes Ψ :: *'b*
and P :: $('a, 'b, 'c)\ psi$
and *Rel* :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes *eqvt Rel*
and *C1*: $\bigwedge Q.\ (\Psi, Q \parallel \mathbf{0}, Q) \in Rel$

shows $\Psi \rhd (P \parallel \mathbf{0}) \rightsquigarrow [Rel]\ P$
⟨*proof*⟩

lemma *parNilRight*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes *eqvt Rel*
and *C1*: $\bigwedge Q.\ (\Psi, Q, (Q \parallel \mathbf{0})) \in Rel$

shows $\Psi \rhd P \rightsquigarrow [Rel]\ (P \parallel \mathbf{0})$
⟨*proof*⟩

lemma *resNilLeft*:
fixes x :: *name*
and Ψ :: $'b$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

shows $\Psi \rhd (\!|\nu x|\!)\mathbf{0} \rightsquigarrow [Rel]\ \mathbf{0}$
⟨*proof*⟩

lemma *resNilRight*:
fixes x :: *name*
and Ψ :: $'b$
and Rel :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

shows $\Psi \rhd \mathbf{0} \rightsquigarrow [Rel]\ (\!|\nu x|\!)\mathbf{0}$
⟨*proof*⟩

lemma *inputPushResLeft*:
fixes Ψ :: $'b$
and x :: *name*
and M :: $'a$
and $xvec$ :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)\ psi$

assumes *eqvt Rel*
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp xvec$
and $x \sharp N$
and *C1*: $\bigwedge Q.\ (\Psi, Q, Q) \in Rel$

shows $\Psi \rhd (\!|\nu x|\!)(M(\!|\lambda * xvec\ N|\!).P) \rightsquigarrow [Rel]\ M(\!|\lambda * xvec\ N|\!).(\!|\nu x|\!)P$
⟨*proof*⟩

lemma *inputPushResRight*:
fixes Ψ :: $'b$
and x :: *name*

and M :: $'a$
and $xvec$ :: $name\ list$
and N :: $'a$
and P :: $('a, 'b, 'c)\ psi$

assumes $eqvt\ Rel$
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp xvec$
and $x \sharp N$
and $C1$: $\bigwedge Q.\ (\Psi, Q, Q) \in Rel$

shows $\Psi \rhd M(\!|\lambda * xvec\ N|\!).(|\nu x|\!)P \rightsquigarrow [Rel]\ (|\nu x|\!)(M(\!|\lambda * xvec\ N|\!).P)$
⟨*proof*⟩

lemma *outputPushResLeft*:
fixes Ψ :: $'b$
and x :: $name$
and M :: $'a$
and N :: $'a$
and P :: $('a, 'b, 'c)\ psi$

assumes $eqvt\ Rel$
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp N$
and $C1$: $\bigwedge Q.\ (\Psi, Q, Q) \in Rel$

shows $\Psi \rhd (|\nu x|\!)(M\langle N\rangle.P) \rightsquigarrow [Rel]\ M\langle N\rangle.(|\nu x|\!)P$
⟨*proof*⟩

lemma *broutputNoBind*:
fixes Ψ :: $'b$
and M :: $'a$
and N :: $'a$
and P :: $('a, 'b, 'c)\ psi$
and α :: $'a\ action$
and P' :: $('a, 'b, 'c)\ psi$

assumes $\Psi \rhd M\langle N\rangle.P \longmapsto (¡K(|\nu * xvec|\!)\langle N'\rangle) \prec P'$
shows $xvec = []$
⟨*proof*⟩

lemma *broutputObjectEq*:
fixes Ψ :: $'b$
and M :: $'a$
and N :: $'a$
and P :: $('a, 'b, 'c)\ psi$
and α :: $'a\ action$

and $P' :: ('a, 'b, 'c)\ psi$

assumes $\Psi \rhd M\langle N\rangle.P \longmapsto (¡K(\!|\nu * xvec|\!)\langle N'\rangle) \prec P'$
shows $N = N'$
⟨*proof*⟩

lemma *brOutputOutputCases*[*consumes 1*, *case-names cBrOutput*]:
fixes $\Psi :: 'b$
and $M :: 'a$
and $N :: 'a$
and $P :: ('a, 'b, 'c)\ psi$
and $\alpha :: 'a\ action$
and $P' :: ('a, 'b, 'c)\ psi$

assumes *Trans*: $\Psi \rhd M\langle N\rangle.P \longmapsto (¡K(\!|\nu * xvec|\!)\langle N'\rangle) \prec P'$
and *rBrOutput*: $\bigwedge K.\ \Psi \vdash M \preceq K \Longrightarrow Prop\ (¡K\langle N\rangle)\ P$

shows $Prop\ (¡K(\!|\nu * xvec|\!)\langle N'\rangle)\ P'$
⟨*proof*⟩

lemma *outputPushResRight*:
fixes $\Psi :: 'b$
and $x :: name$
and $M :: 'a$
and $N :: 'a$
and $P :: ('a, 'b, 'c)\ psi$

assumes *eqvt Rel*
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp N$
and *C1*: $\bigwedge Q.\ (\Psi, Q, Q) \in Rel$

shows $\Psi \rhd M\langle N\rangle.(\!|\nu x|\!)P \rightsquigarrow [Rel]\ (\!|\nu x|\!)(M\langle N\rangle.P)$
⟨*proof*⟩

lemma *casePushResLeft*:
fixes $\Psi :: 'b$
and $x :: name$
and $Cs :: ('c \times ('a, 'b, 'c)\ psi)\ list$

assumes *eqvt Rel*
and $x \sharp \Psi$
and $x \sharp map\ fst\ Cs$
and *C1*: $\bigwedge Q.\ (\Psi, Q, Q) \in Rel$

shows $\Psi \rhd (\!|\nu x|\!)(Cases\ Cs) \rightsquigarrow [Rel]\ Cases\ (map\ (\lambda(\varphi, P).\ (\varphi, (\!|\nu x|\!)P))\ Cs)$
⟨*proof*⟩

lemma *casePushResRight*:
fixes Ψ :: $'b$
and x :: *name*
and Cs :: $('c \times ('a, 'b, 'c)\ psi)\ list$

assumes *eqvt Rel*
and $x \sharp \Psi$
and $x \sharp map\ fst\ Cs$
and *C1*: $\bigwedge Q.\ (\Psi, Q, Q) \in Rel$

shows $\Psi \rhd Cases\ (map\ (\lambda(\varphi, P).\ (\varphi, (\!|\nu x|\!)P))\ Cs) \rightsquigarrow [Rel]\ (\!|\nu x|\!)(Cases\ Cs)$
⟨*proof*⟩

lemma *resInputCases*[*consumes 5*, *case-names cRes*]:
fixes Ψ :: $'b$
and x :: *name*
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and C :: $'d$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|N|\!) \prec P'$
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp N$
and $x \sharp P'$
and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto M(\!|N|\!) \prec P']\!] \Longrightarrow Prop\ ((\!|\nu x|\!)P')$

shows *Prop* P'
⟨*proof*⟩

lemma *resBrInputCases*[*consumes 5*, *case-names cRes*]:
fixes Ψ :: $'b$
and x :: *name*
and P :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and N :: $'a$
and P' :: $('a, 'b, 'c)\ psi$
and C :: $'d$::*fs-name*

assumes *Trans*: $\Psi \rhd (\!|\nu x|\!)P \longmapsto ¿M(\!|N|\!) \prec P'$
and $x \sharp \Psi$
and $x \sharp M$
and $x \sharp N$
and $x \sharp P'$
and *rScope*: $\bigwedge P'.\ [\![\Psi \rhd P \longmapsto ¿M(\!|N|\!) \prec P']\!] \Longrightarrow Prop\ ((\!|\nu x|\!)P')$

shows *Prop* P'

⟨*proof*⟩

lemma *swap-supp*:
fixes x :: *name*
and y :: *name*
and N :: $'a$

assumes $y \in supp\ N$

shows $x \in supp\ ([(x, y)] \cdot N)$
⟨*proof*⟩

lemma *swap-supp'*:
fixes x :: *name*
and y :: *name*
and N :: $'a$

assumes $x \in supp\ N$

shows $y \in supp\ ([(x, y)] \cdot N)$
⟨*proof*⟩

lemma *brOutputFreshSubjectChain*:
fixes Ψ :: $'b$
and Q :: $('a, 'b, 'c)\ psi$
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$
and Q' :: $('a, 'b, 'c)\ psi$
and *yvec* :: *name list*

assumes $\Psi \rhd Q \longmapsto ¡M⦇\nu * xvec⦈\langle N\rangle \prec Q'$
and $xvec \sharp * M$
and $yvec \sharp * Q$

shows $yvec \sharp * M$
⟨*proof*⟩

lemma *scopeExtLeft*:
fixes x :: *name*
and P :: $('a, 'b, 'c)\ psi$
and Ψ :: $'b$
and Q :: $('a, 'b, 'c)\ psi$
and *Rel* :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes $x \sharp P$
and $x \sharp \Psi$
and *eqvt Rel*
and *C1*: $\bigwedge \Psi'\ R.\ (\Psi', R, R) \in Rel$

and *C2*: $\bigwedge y\ \Psi'\ R\ S\ zvec.\ [\![y \sharp \Psi';\ y \sharp R;\ zvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (\!|\nu y|\!)((\!|\nu* zvec|\!)(R \parallel S)), (\!|\nu* zvec|\!)(R \parallel (\!|\nu y|\!)S)) \in Rel$
and *C3*: $\bigwedge \Psi'\ zvec\ R\ y.\ [\![y \sharp \Psi';\ zvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (\!|\nu y|\!)((\!|\nu* zvec|\!)R), (\!|\nu* zvec|\!)((\!|\nu y|\!)R)) \in Rel$
— Addition for Broadcast
and *C4*: $\bigwedge \Psi'\ R\ S\ zvec.\ [\![zvec \sharp* R;\ zvec \sharp* \Psi']\!] \Longrightarrow (\Psi', ((\!|\nu* zvec|\!)(R \parallel S)), (R \parallel ((\!|\nu* zvec|\!)S))) \in Rel$

shows $\Psi \rhd (\!|\nu x|\!)(P \parallel Q) \rightsquigarrow[Rel]\ P \parallel (\!|\nu x|\!)Q$
⟨*proof*⟩

lemma *scopeExtRight*:
fixes x :: *name*
and P :: $('a, 'b, 'c)$ *psi*
and Ψ :: $'b$
and Q :: $('a, 'b, 'c)$ *psi*
and *Rel* :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes $x \sharp P$
and $x \sharp \Psi$
and *eqvt Rel*
and *C1*: $\bigwedge \Psi'\ R.\ (\Psi, R, R) \in Rel$
and *C2*: $\bigwedge y\ \Psi'\ R\ S\ zvec.\ [\![y \sharp \Psi';\ y \sharp R;\ zvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (\!|\nu* zvec|\!)(R \parallel (\!|\nu y|\!)S), (\!|\nu y|\!)((\!|\nu* zvec|\!)(R \parallel S))) \in Rel$
— Addition for Broadcast
and *C3*: $\bigwedge \Psi'\ R\ S\ zvec.\ [\![zvec \sharp* R;\ zvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (R \parallel ((\!|\nu* zvec|\!)S)), ((\!|\nu* zvec|\!)(R \parallel S))) \in Rel$

shows $\Psi \rhd P \parallel (\!|\nu x|\!)Q \rightsquigarrow[Rel]\ (\!|\nu x|\!)(P \parallel Q)$
⟨*proof*⟩

lemma *simParComm*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and *Rel* :: $('b \times ('a, 'b, 'c)\ psi \times ('a, 'b, 'c)\ psi)\ set$

assumes *eqvt Rel*
and *C1*: $\bigwedge \Psi'\ R\ S.\ (\Psi', R \parallel S, S \parallel R) \in Rel$
and *C2*: $\bigwedge \Psi'\ R\ S\ xvec.\ [\![(\Psi', R, S) \in Rel;\ xvec \sharp* \Psi']\!] \Longrightarrow (\Psi', (\!|\nu* xvec|\!)R, (\!|\nu* xvec|\!)S) \in Rel$

shows $\Psi \rhd P \parallel Q \rightsquigarrow[Rel]\ Q \parallel P$
⟨*proof*⟩

lemma *bangExtLeft*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*

assumes *guarded P*
and $\bigwedge \Psi'\ Q.\ (\Psi',\ Q,\ Q) \in Rel$

shows $\Psi \rhd\ !P \rightsquigarrow [Rel]\ P \parallel\ !P$
⟨*proof*⟩

lemma *bangExtRight*:
fixes $\Psi :: 'b$
and $P :: ('a,\ 'b,\ 'c)\ psi$

assumes *C1*: $\bigwedge \Psi'\ Q.\ (\Psi',\ Q,\ Q) \in Rel$

shows $\Psi \rhd\ P \parallel\ !P \rightsquigarrow [Rel]\ !P$
⟨*proof*⟩

end

end
theory *Bisim-Pres*
imports *Bisimulation Sim-Pres*
begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Bisim_Pres* from [1].

context *env* **begin**

lemma *bisimInputPres*:
fixes Ψ :: $'b$
and P :: $('a,\ 'b,\ 'c)\ psi$
and Q :: $('a,\ 'b,\ 'c)\ psi$
and M :: $'a$
and *xvec* :: *name list*
and N :: $'a$

assumes $\bigwedge Tvec.\ length\ xvec = length\ Tvec \Longrightarrow \Psi \rhd P[xvec ::= Tvec] \sim Q[xvec ::= Tvec]$

shows $\Psi \rhd\ M(\!|\lambda * xvec\ N|\!).P \sim M(\!|\lambda * xvec\ N|\!).Q$
⟨*proof*⟩

lemma *bisimOutputPres*:
fixes $\Psi :: 'b$
and $P :: ('a,\ 'b,\ 'c)\ psi$
and $Q :: ('a,\ 'b,\ 'c)\ psi$
and $M :: 'a$
and $N :: 'a$

assumes $\Psi \rhd\ P \sim Q$

shows $\Psi \rhd\ M\langle N\rangle.P \sim M\langle N\rangle.Q$

⟨*proof*⟩

lemma *bisimCasePres*:
fixes Ψ :: *'b*
and *CsP* :: (*'c* × (*'a*, *'b*, *'c*) *psi*) *list*
and *CsQ* :: (*'c* × (*'a*, *'b*, *'c*) *psi*) *list*

assumes ⋀φ *P*. (φ, *P*) ∈ *set CsP* ⟹ ∃ *Q*. (φ, *Q*) ∈ *set CsQ* ∧ *guarded Q* ∧ Ψ ▷ *P* ∼ *Q*
and ⋀φ *Q*. (φ, *Q*) ∈ *set CsQ* ⟹ ∃ *P*. (φ, *P*) ∈ *set CsP* ∧ *guarded P* ∧ Ψ ▷ *P* ∼ *Q*

shows Ψ ▷ *Cases CsP* ∼ *Cases CsQ*
⟨*proof*⟩

lemma *bisimResPres*:
fixes Ψ :: *'b*
and *P* :: (*'a*, *'b*, *'c*) *psi*
and *Q* :: (*'a*, *'b*, *'c*) *psi*
and *x* :: *name*

assumes Ψ ▷ *P* ∼ *Q*
and *x* ♯ Ψ

shows Ψ ▷ (|ν*x*|)*P* ∼ (|ν*x*|)*Q*
⟨*proof*⟩

lemma *bisimResChainPres*:
fixes Ψ :: *'b*
and *P* :: (*'a*, *'b*, *'c*) *psi*
and *Q* :: (*'a*, *'b*, *'c*) *psi*
and *xvec* :: *name list*

assumes Ψ ▷ *P* ∼ *Q*
and *xvec* ♯* Ψ

shows Ψ ▷ (|ν*xvec|)*P* ∼ (|ν*xvec|)*Q*
⟨*proof*⟩

lemma *bisimParPresAux*:
fixes Ψ :: *'b*
and Ψ$_R$ :: *'b*
and *P* :: (*'a*, *'b*, *'c*) *psi*
and *Q* :: (*'a*, *'b*, *'c*) *psi*
and *R* :: (*'a*, *'b*, *'c*) *psi*
and *A*$_R$ :: *name list*

assumes Ψ ⊗ Ψ$_R$ ▷ *P* ∼ *Q*
and *FrR*: *extractFrame R* = ⟨*A*$_R$, Ψ$_R$⟩

and $A_R \sharp* \Psi$
and $A_R \sharp* P$
and $A_R \sharp* Q$

shows $\Psi \rhd P \parallel R \sim Q \parallel R$
⟨*proof*⟩

lemma *bisimParPres*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $R :: ('a, 'b, 'c)\ psi$

assumes $\Psi \rhd P \sim Q$

shows $\Psi \rhd P \parallel R \sim Q \parallel R$
⟨*proof*⟩

end

end
theory *Bisim-Struct-Cong*
imports *Bisim-Pres Sim-Struct-Cong Psi-Calculi.Structural-Congruence*
begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Bisim_Struct_Cong* from [1].

context *env* **begin**

lemma *bisimParComm*:
fixes $\Psi :: 'b$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$

shows $\Psi \rhd P \parallel Q \sim Q \parallel P$
⟨*proof*⟩

inductive-set *resCommRel* :: $('b \times ('a,'b,'c)\ psi \times ('a,'b,'c)\ psi)\ set$
where
resCommRel-refl : $(\Psi,P,P) \in resCommRel$
| *resCommRel-swap* : $[\![a \sharp \Psi;\ b \sharp \Psi]\!] \Longrightarrow (\Psi,(\!|\nu a|\!)((\!|\nu b|\!)P),(\!|\nu b|\!)((\!|\nu a|\!)P)) \in resCommRel$
| *resCommRel-res* : $[\![(\Psi,P,Q) \in resCommRel;\ a \sharp \Psi]\!] \Longrightarrow (\Psi,(\!|\nu a|\!)P,(\!|\nu a|\!)Q) \in resCommRel$

lemma *eqvtResCommRel*: *eqvt resCommRel*
⟨*proof*⟩

lemma *resCommRelStarRes*:

assumes $(\Psi,P,Q) \in resCommRel^\star$
and $a \sharp \Psi$
shows $(\Psi,(\!|\nu a|\!)P,(\!|\nu a|\!)Q) \in resCommRel^\star$
⟨*proof*⟩

lemma *resCommRelStarResChain*:
assumes $(\Psi,P,Q) \in resCommRel^\star$
and $xvec \sharp* \Psi$
shows $(\Psi,(\!|\nu* xvec|\!)P,(\!|\nu* xvec|\!)Q) \in resCommRel^\star$
⟨*proof*⟩

lemma *length-induct1*[*consumes 0*, *case-names Nil Cons*]:
assumes *b*: $P\ []$
and *s*: $\bigwedge x\ xvec.\ [\![\bigwedge yvec.\ length\ xvec = length\ yvec \Longrightarrow P\ yvec]\!] \Longrightarrow P(x\#xvec)$
shows $P\ xvec$
⟨*proof*⟩

lemma *oneStepPerm-in-rel*:
assumes $xvec \sharp* \Psi$
shows $(\Psi,(\!|\nu* xvec|\!)P,(\!|\nu*(rotate1\ xvec)|\!)P) \in resCommRel^\star$
⟨*proof*⟩

lemma *fresh-same-multiset*:
fixes *xvec*::*name list*
and *yvec*::*name list*
assumes $mset\ xvec = mset\ yvec$
and $xvec \sharp* X$
shows $yvec \sharp* X$
⟨*proof*⟩

lemma *nStepPerm-in-rel*:
assumes $xvec \sharp* \Psi$
shows $(\Psi,(\!|\nu* xvec|\!)P,(\!|\nu*(rotate\ n\ xvec)|\!)P) \in resCommRel^\star$
⟨*proof*⟩

lemma *any-perm-in-rel*:
assumes $xvec \sharp* \Psi$
and $mset\ xvec = mset\ yvec$
shows $(\Psi,(\!|\nu* xvec|\!)P,(\!|\nu* yvec|\!)P) \in resCommRel^\star$
⟨*proof*⟩

lemma *bisimResComm*:
fixes $x :: name$
and $\Psi :: 'b$
and $y :: name$
and $P :: ('a, 'b, 'c)\ psi$

shows $\Psi \rhd (\!|\nu x|\!)((\!|\nu y|\!)P) \sim (\!|\nu y|\!)((\!|\nu x|\!)P)$
⟨*proof*⟩

lemma *bisimResComm'*:
fixes x :: *name*
and Ψ :: $'b$
and *xvec* :: *name list*
and P :: $('a, 'b, 'c)$ *psi*

assumes $x \sharp \Psi$
and *xvec* $\sharp* \Psi$

shows $\Psi \rhd (\!|\nu x|\!)((\!|\nu * xvec|\!)P) \sim (\!|\nu * xvec|\!)((\!|\nu x|\!)P)$
⟨*proof*⟩

lemma *bisimParPresSym*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and R :: $('a, 'b, 'c)$ *psi*

assumes $\Psi \rhd P \sim Q$

shows $\Psi \rhd R \parallel P \sim R \parallel Q$
⟨*proof*⟩

lemma *bisimScopeExt*:
fixes x :: *name*
and Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*

assumes $x \sharp P$

shows $\Psi \rhd (\!|\nu x|\!)(P \parallel Q) \sim P \parallel (\!|\nu x|\!)Q$
⟨*proof*⟩

lemma *bisimScopeExtChain*:
fixes *xvec* :: *name list*
and Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*

assumes *xvec* $\sharp* \Psi$
and *xvec* $\sharp* P$

shows $\Psi \rhd (\!|\nu * xvec|\!)(P \parallel Q) \sim P \parallel ((\!|\nu * xvec|\!)Q)$
⟨*proof*⟩

lemma *bisimParAssoc*:
fixes Ψ :: $'b$

and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$
and $R :: ('a, 'b, 'c)\ psi$

shows $\Psi \rhd (P \parallel Q) \parallel R \sim P \parallel (Q \parallel R)$
⟨*proof*⟩

lemma *bisimParNil*:
fixes $P :: ('a, 'b, 'c)\ psi$

shows $\Psi \rhd P \parallel \mathbf{0} \sim P$
⟨*proof*⟩

lemma *bisimResNil*:
fixes $x :: name$
and $\Psi :: 'b$

shows $\Psi \rhd (\!|\nu x|\!)\mathbf{0} \sim \mathbf{0}$
⟨*proof*⟩

lemma *bisimOutputPushRes*:
fixes $x :: name$
and $\Psi :: 'b$
and $M :: 'a$
and $N :: 'a$
and $P :: ('a, 'b, 'c)\ psi$

assumes $x \sharp M$
and $x \sharp N$

shows $\Psi \rhd (\!|\nu x|\!)(M\langle N\rangle.P) \sim M\langle N\rangle.(\!|\nu x|\!)P$
⟨*proof*⟩

lemma *bisimInputPushRes*:
fixes $x :: name$
and $\Psi :: 'b$
and $M :: 'a$
and $xvec :: name\ list$
and $N :: 'a$
and $P :: ('a, 'b, 'c)\ psi$

assumes $x \sharp M$
and $x \sharp xvec$
and $x \sharp N$

shows $\Psi \rhd (\!|\nu x|\!)(M(\!|\lambda * xvec\ N|\!).P) \sim M(\!|\lambda * xvec\ N|\!).(\!|\nu x|\!)P$
⟨*proof*⟩

lemma *bisimCasePushRes*:

fixes x :: *name*
and Ψ :: $'b$
and Cs :: $('c \times ('a, 'b, 'c)\ psi)\ list$

assumes $x \sharp (map\ fst\ Cs)$

shows $\Psi \rhd (\!|\nu x|\!)(Cases\ Cs) \sim Cases(map\ (\lambda(\varphi, P).\ (\varphi, (\!|\nu x|\!)P))\ Cs)$
⟨*proof*⟩

lemma *bangExt*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$

assumes *guarded* P

shows $\Psi \rhd !P \sim P \parallel !P$
⟨*proof*⟩

lemma *bisimScopeExtSym*:
fixes x :: *name*
and Q :: $('a, 'b, 'c)\ psi$
and P :: $('a, 'b, 'c)\ psi$

assumes $x \sharp \Psi$
and $x \sharp Q$

shows $\Psi \rhd (\!|\nu x|\!)(P \parallel Q) \sim ((\!|\nu x|\!)P) \parallel Q$
⟨*proof*⟩

lemma *bisimScopeExtChainSym*:
fixes *xvec* :: *name list*
and Q :: $('a, 'b, 'c)\ psi$
and P :: $('a, 'b, 'c)\ psi$

assumes *xvec* $\sharp* \Psi$
and *xvec* $\sharp* Q$

shows $\Psi \rhd (\!|\nu* xvec|\!)(P \parallel Q) \sim ((\!|\nu* xvec|\!)P) \parallel Q$
⟨*proof*⟩

lemma *bisimParPresAuxSym*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Q :: $('a, 'b, 'c)\ psi$
and R :: $('a, 'b, 'c)\ psi$

assumes $\Psi \otimes \Psi_R \rhd P \sim Q$
and $extractFrame\ R = \langle A_R, \Psi_R \rangle$
and $A_R \sharp* \Psi$

and $A_R \sharp* P$
and $A_R \sharp* Q$

shows $\Psi \rhd R \parallel P \sim R \parallel Q$
⟨*proof*⟩

lemma *bangDerivative*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and α :: $'a$ *action*
and P' :: $('a, 'b, 'c)$ *psi*

assumes $\Psi \rhd !P \longmapsto \alpha \prec P'$
and $\Psi \rhd P \sim Q$
and *bn* $\alpha \sharp* \Psi$
and *bn* $\alpha \sharp* P$
and *bn* $\alpha \sharp* Q$
and *bn* $\alpha \sharp* subject\ \alpha$
and *guarded* Q

obtains Q' R T **where** $\Psi \rhd !Q \longmapsto \alpha \prec Q'$ **and** $\Psi \rhd P' \sim R \parallel !P$ **and** $\Psi \rhd Q' \sim T \parallel !Q$ **and** $\Psi \rhd R \sim T$
and ((*supp* R)::*name set*) $\subseteq$ *supp* P' **and** ((*supp* T)::*name set*) $\subseteq$ *supp* Q'
⟨*proof*⟩

lemma *structCongBisim*:
fixes P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*

assumes $P \equiv_s Q$

shows $P \sim Q$
⟨*proof*⟩

lemma *bisimBangPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*

assumes $\Psi \rhd P \sim Q$
and *guarded* P
and *guarded* Q

shows $\Psi \rhd !P \sim !Q$
⟨*proof*⟩

end

end

theory *Bisim-Subst*
imports *Bisim-Struct-Cong Psi-Calculi.Close-Subst*
begin

This file is a (heavily modified) variant of the theory *Psi_Calculi.Bisim_Subst* from [1].

context *env* **begin**

abbreviation
bisimSubstJudge (‹- ⊳ - ∼$_s$ -› [70, 70, 70] 65) **where** $\Psi \rhd P \sim_s Q \equiv (\Psi, P, Q) \in closeSubst\ bisim$
abbreviation
bisimSubstNilJudge (‹- ∼$_s$ -› [70, 70] 65) **where** $P \sim_s Q \equiv SBottom' \rhd P \sim_s Q$

lemmas *bisimSubstClosed*[*eqvt*] = *closeSubstClosed*[*OF bisimEqvt*]
lemmas *bisimSubstEqvt*[*simp*] = *closeSubstEqvt*[*OF bisimEqvt*]

lemma *bisimSubstOutputPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and N :: $'a$

assumes $\Psi \rhd P \sim_s Q$

shows $\Psi \rhd M\langle N\rangle.P \sim_s M\langle N\rangle.Q$
⟨*proof*⟩

lemma *seqSubstInputChain*[*simp*]:
fixes *xvec* :: *name list*
and N :: $'a$
and P :: $('a, 'b, 'c)$ *psi*
and σ :: (*name list* × $'a$ *list*) *list*

assumes *xvec* ♯* σ

shows *seqSubs'* (*inputChain xvec N P*) σ = *inputChain xvec* (*substTerm.seqSubst N* σ) (*seqSubs P* σ)
⟨*proof*⟩

lemma *bisimSubstInputPres*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*
and M :: $'a$
and *xvec* :: *name list*

and N :: $'a$

assumes $\Psi \rhd P \sim_s Q$
and $xvec \sharp* \Psi$
and *distinct xvec*

shows $\Psi \rhd M(\!|\lambda* xvec\ N|\!).P \sim_s M(\!|\lambda* xvec\ N|\!).Q$
⟨*proof*⟩

lemma *bisimSubstCasePresAux*:
fixes Ψ :: $'b$
and *CsP* :: $('c \times ('a, 'b, 'c)\ psi)\ list$
and *CsQ* :: $('c \times ('a, 'b, 'c)\ psi)\ list$

assumes *C1*: $\bigwedge\varphi\ P.\ (\varphi, P) \in set\ CsP \Longrightarrow \exists\, Q.\ (\varphi, Q) \in set\ CsQ \wedge guarded\ Q \wedge \Psi \rhd P \sim_s Q$
and *C2*: $\bigwedge\varphi\ Q.\ (\varphi, Q) \in set\ CsQ \Longrightarrow \exists\, P.\ (\varphi, P) \in set\ CsP \wedge guarded\ P \wedge \Psi \rhd P \sim_s Q$

shows $\Psi \rhd Cases\ CsP \sim_s Cases\ CsQ$
⟨*proof*⟩

lemma *bisimSubstReflexive*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$

shows $\Psi \rhd P \sim_s P$
⟨*proof*⟩

lemma *bisimSubstTransitive*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Q :: $('a, 'b, 'c)\ psi$
and R :: $('a, 'b, 'c)\ psi$

assumes $\Psi \rhd P \sim_s Q$
and $\Psi \rhd Q \sim_s R$

shows $\Psi \rhd P \sim_s R$
⟨*proof*⟩

lemma *bisimSubstSymmetric*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)\ psi$
and Q :: $('a, 'b, 'c)\ psi$

assumes $\Psi \rhd P \sim_s Q$

shows $\Psi \rhd Q \sim_s P$

⟨*proof*⟩

lemma *bisimSubstCasePres*:
fixes Ψ :: $'b$
and *CsP* :: $('c \times ('a, 'b, 'c)\ psi)\ list$
and *CsQ* :: $('c \times ('a, 'b, 'c)\ psi)\ list$

assumes *length CsP* = *length CsQ*
and *C*: $\bigwedge(i::nat)\ \varphi\ P\ \varphi'\ Q.\ [\![i <= length\ CsP; (\varphi, P) = nth\ CsP\ i; (\varphi', Q) = nth\ CsQ\ i]\!] \Longrightarrow \varphi = \varphi' \wedge \Psi \rhd P \sim_s Q \wedge guarded\ P \wedge guarded\ Q$

shows $\Psi \rhd Cases\ CsP \sim_s Cases\ CsQ$
⟨*proof*⟩

lemma *bisimSubstParPres*:
fixes Ψ :: $'b$
and *P* :: $('a, 'b, 'c)\ psi$
and *Q* :: $('a, 'b, 'c)\ psi$
and *R* :: $('a, 'b, 'c)\ psi$

assumes $\Psi \rhd P \sim_s Q$

shows $\Psi \rhd P \parallel R \sim_s Q \parallel R$
⟨*proof*⟩

lemma *bisimSubstResPres*:
fixes Ψ :: $'b$
and *P* :: $('a, 'b, 'c)\ psi$
and *Q* :: $('a, 'b, 'c)\ psi$
and *x* :: *name*

assumes $\Psi \rhd P \sim_s Q$
and $x \sharp \Psi$

shows $\Psi \rhd (\!|\nu x|\!) P \sim_s (\!|\nu x|\!) Q$
⟨*proof*⟩

lemma *bisimSubstBangPres*:
fixes Ψ :: $'b$
and *P* :: $('a, 'b, 'c)\ psi$
and *Q* :: $('a, 'b, 'c)\ psi$

assumes $\Psi \rhd P \sim_s Q$
and *guarded P*
and *guarded Q*

shows $\Psi \rhd\ !P \sim_s\ !Q$
⟨*proof*⟩

lemma *substNil*[*simp*]:
  fixes *xvec* :: *name list*
    and *Tvec* :: *'a list*

assumes *wellFormedSubst* σ
  and *distinct xvec*

shows (**0**[<σ>]) = **0**
  ⟨*proof*⟩

lemma *bisimSubstParNil*:
  fixes Ψ :: *'b*
    and P :: (*'a*, *'b*, *'c*) *psi*

shows $\Psi \rhd P \parallel \mathbf{0} \sim_s P$
  ⟨*proof*⟩

lemma *bisimSubstParComm*:
  fixes Ψ :: *'b*
    and P :: (*'a*, *'b*, *'c*) *psi*
    and Q :: (*'a*, *'b*, *'c*) *psi*

shows $\Psi \rhd P \parallel Q \sim_s Q \parallel P$
  ⟨*proof*⟩

lemma *bisimSubstParAssoc*:
  fixes Ψ :: *'b*
    and P :: (*'a*, *'b*, *'c*) *psi*
    and Q :: (*'a*, *'b*, *'c*) *psi*
    and R :: (*'a*, *'b*, *'c*) *psi*

shows $\Psi \rhd (P \parallel Q) \parallel R \sim_s P \parallel (Q \parallel R)$
  ⟨*proof*⟩

lemma *bisimSubstResNil*:
  fixes Ψ :: *'b*
    and x :: *name*

shows $\Psi \rhd (\!|\nu x|\!)\mathbf{0} \sim_s \mathbf{0}$
⟨*proof*⟩

lemma *seqSubst2*:
  fixes x :: *name*
    and P :: (*'a*, *'b*, *'c*) *psi*

assumes *wellFormedSubst* σ
  and $x \sharp \sigma$
  and $x \sharp P$

shows $x \sharp P[<\sigma>]$
⟨*proof*⟩

notation *substTerm.seqSubst* (‹-[<->]› [*100*, *100*] *100*)

lemma *bisimSubstScopeExt*:
fixes $\Psi :: 'b$
and $x :: name$
and $P :: ('a, 'b, 'c)\ psi$
and $Q :: ('a, 'b, 'c)\ psi$

assumes $x \sharp P$

shows $\Psi \rhd (\!|\nu x|\!)(P \parallel Q) \sim_s P \parallel (\!|\nu x|\!)Q$
⟨*proof*⟩

lemma *bisimSubstCasePushRes*:
fixes $x :: name$
and $\Psi :: 'b$
and $Cs :: ('c \times ('a, 'b, 'c)\ psi)\ list$

assumes $x \sharp map\ fst\ Cs$

shows $\Psi \rhd (\!|\nu x|\!)(Cases\ Cs) \sim_s Cases\ map\ (\lambda(\varphi, P).\ (\varphi, (\!|\nu x|\!)P))\ Cs$
⟨*proof*⟩

lemma *bisimSubstOutputPushRes*:
fixes $x :: name$
and $\Psi :: 'b$
and $M :: 'a$
and $N :: 'a$
and $P :: ('a, 'b, 'c)\ psi$

assumes $x \sharp M$
and $x \sharp N$

shows $\Psi \rhd (\!|\nu x|\!)(M\langle N\rangle.P) \sim_s M\langle N\rangle.(\!|\nu x|\!)P$
⟨*proof*⟩

lemma *bisimSubstInputPushRes*:
fixes $x :: name$
and $\Psi :: 'b$
and $M :: 'a$
and $xvec :: name\ list$
and $N :: 'a$

assumes $x \sharp M$
and $x \sharp xvec$
and $x \sharp N$

shows $\Psi \rhd (\!|\nu x|\!)(M(\!|\lambda{*}xvec\ N|\!).P) \sim_s M(\!|\lambda{*}xvec\ N|\!).(\!|\nu x|\!)P$
⟨*proof*⟩

lemma *bisimSubstResComm*:
fixes x :: *name*
and y :: *name*

shows $\Psi \rhd (\!|\nu x|\!)((\!|\nu y|\!)P) \sim_s (\!|\nu y|\!)((\!|\nu x|\!)P)$
⟨*proof*⟩

lemma *bisimSubstExtBang*:
fixes Ψ :: $'b$
and P :: $('a, 'b, 'c)$ *psi*

assumes *guarded P*

shows $\Psi \rhd !P \sim_s P \parallel !P$
⟨*proof*⟩

lemma *structCongBisimSubst*:
fixes P :: $('a, 'b, 'c)$ *psi*
and Q :: $('a, 'b, 'c)$ *psi*

assumes $P \equiv_s Q$

shows $P \sim_s Q$
⟨*proof*⟩

end

end
theory *Broadcast-Thms*
imports *Broadcast-Chain Broadcast-Frame Semantics Simulation Bisimulation Sim-Pres*
Sim-Struct-Cong Bisim-Pres Bisim-Struct-Cong Bisim-Subst
begin

context *env*
begin

2.1 Syntax

2.1.1 Psi calculus agents – Definition 2

M, N range over message terms. P, Q range over processes. C ranges over cases.

- Output: $M\langle N\rangle.P$

- Input: $M(\!|\lambda{*}xvec\ N|\!).P$
- Case: $Case\ C$
- Par: $P \parallel Q$
- Res: $(\!|\nu x|\!)P$
- Assert: $\{\!|\Psi|\!\}$
- Bang: $!P$

- Cases: $\Box\ \varphi \Rightarrow P\ C$

2.1.2 Parameters – Definition 1

- Channel equivalence: $M \leftrightarrow N$
- Composition: $\Psi_P \otimes \Psi_Q$
- Unit: $\mathbf{1}$
- Entailment: $\Psi \vdash \varphi$

2.1.3 Extra predicates for broadcast – Definition 5

- Output connectivity: $M \preceq N$
- Input connectivity: $M \succeq N$

2.1.4 Transitions – Definition 3

- $\Psi \rhd P \longmapsto \alpha\ P'$

2.1.5 Actions (α) – Definition 7

- Input: $M(\!|N|\!)$
- Output: $M(\!|\nu{*}xvec|\!)\langle N\rangle$
- Broadcast input: ¿$M(\!|N|\!)$
- Broadcast output: ¡$M(\!|\nu{*}xvec|\!)\langle N\rangle$
- Silent action: τ

2.2 Semantics

2.2.1 Basic Psi semantics – Table 1

- Theorem *Input*:

  $[\![\Psi \vdash M \leftrightarrow K;\ \mathit{distinct\ xvec};\ \mathit{set\ xvec} \subseteq \mathit{supp}\ N;\ \mathit{length\ xvec} = \mathit{length\ Tvec}]\!]$
  $\Longrightarrow \Psi \rhd M(\!|\lambda{*}\mathit{xvec}\ N|\!).P \longmapsto K(\!|N[\mathit{xvec}::=\mathit{Tvec}]|\!) \prec P[\mathit{xvec}::=\mathit{Tvec}]$

- Theorem *Output*:

  $\Psi \vdash M \leftrightarrow K \Longrightarrow \Psi \rhd M\langle N\rangle.P \longmapsto K\langle N\rangle \prec P$

- Theorem *Case*:

  $[\![\Psi \rhd P \longmapsto \mathit{Rs};\ (\varphi,\ P)\ \mathit{mem\ Cs};\ \Psi \vdash \varphi;\ \mathit{guarded}\ P]\!] \Longrightarrow \Psi \rhd \mathit{Cases\ Cs} \longmapsto \mathit{Rs}$

- Theorems *Par1* and *Par2*:

  $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto \alpha \prec P';\ \mathit{extractFrame}\ Q = \langle A_Q,\ \Psi_Q\rangle;\ \mathit{bn}\ \alpha\ \sharp{*}\ Q;\ A_Q\ \sharp{*}\ \Psi;$
  $A_Q\ \sharp{*}\ P;\ A_Q\ \sharp{*}\ \alpha]\!]$
  $\Longrightarrow \Psi \rhd P \parallel Q \longmapsto \alpha \prec P' \parallel Q$

  $[\![\Psi \otimes \Psi_P \rhd Q \longmapsto \alpha \prec Q';\ \mathit{extractFrame}\ P = \langle A_P,\ \Psi_P\rangle;\ \mathit{bn}\ \alpha\ \sharp{*}\ P;\ A_P\ \sharp{*}\ \Psi;$
  $A_P\ \sharp{*}\ Q;\ A_P\ \sharp{*}\ \alpha]\!]$
  $\Longrightarrow \Psi \rhd P \parallel Q \longmapsto \alpha \prec P \parallel Q'$

- Theorems *Comm1* and *Comm2*:

  $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|N|\!) \prec P';\ \mathit{extractFrame}\ P = \langle A_P,\ \Psi_P\rangle;$
  $\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|\nu{*}\mathit{xvec}|\!)\langle N\rangle \prec Q';\ \mathit{extractFrame}\ Q = \langle A_Q,\ \Psi_Q\rangle;$
  $\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;\ A_P\ \sharp{*}\ \Psi;\ A_P\ \sharp{*}\ P;\ A_P\ \sharp{*}\ Q;\ A_P\ \sharp{*}\ M;\ A_P\ \sharp{*}\ A_Q;\ A_Q\ \sharp{*}\ \Psi;$
  $A_Q\ \sharp{*}\ P;\ A_Q\ \sharp{*}\ Q;\ A_Q\ \sharp{*}\ K;\ \mathit{xvec}\ \sharp{*}\ P]\!]$
  $\Longrightarrow \Psi \rhd P \parallel Q \longmapsto \tau \prec (\!|\nu{*}\mathit{xvec}|\!)P' \parallel Q'$

  $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto M(\!|\nu{*}\mathit{xvec}|\!)\langle N\rangle \prec P';\ \mathit{extractFrame}\ P = \langle A_P,\ \Psi_P\rangle;$
  $\Psi \otimes \Psi_P \rhd Q \longmapsto K(\!|N|\!) \prec Q';\ \mathit{extractFrame}\ Q = \langle A_Q,\ \Psi_Q\rangle;\ \Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \leftrightarrow K;$
  $A_P\ \sharp{*}\ \Psi;\ A_P\ \sharp{*}\ P;\ A_P\ \sharp{*}\ Q;\ A_P\ \sharp{*}\ M;\ A_P\ \sharp{*}\ A_Q;\ A_Q\ \sharp{*}\ \Psi;\ A_Q\ \sharp{*}\ P;\ A_Q\ \sharp{*}\ Q;$
  $A_Q\ \sharp{*}\ K;\ \mathit{xvec}\ \sharp{*}\ Q]\!]$
  $\Longrightarrow \Psi \rhd P \parallel Q \longmapsto \tau \prec (\!|\nu{*}\mathit{xvec}|\!)P' \parallel Q'$

- Theorem *Open*:

$[\![\Psi \rhd P \longmapsto M(\!|\nu*(xvec \mathbin{@} yvec)|\!)\langle N\rangle \prec P';\ x \in supp\ N;\ x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;$
$x \sharp yvec]\!]$
$\Longrightarrow \Psi \rhd (\!|\nu x|\!)P \longmapsto M(\!|\nu*(xvec \mathbin{@} x \mathbin{\#} yvec)|\!)\langle N\rangle \prec P'$

- Theorem *Scope*:

  $[\![\Psi \rhd P \longmapsto \alpha \prec P';\ x \sharp \Psi;\ x \sharp \alpha]\!] \Longrightarrow \Psi \rhd (\!|\nu x|\!)P \longmapsto \alpha \prec (\!|\nu x|\!)P'$

- Theorem *Bang*:

  $[\![\Psi \rhd P \parallel !P \longmapsto Rs;\ guarded\ P]\!] \Longrightarrow \Psi \rhd !P \longmapsto Rs$

2.2.2 Broadcast rules – Table 2

- Theorem *BrInput*:

  $[\![\Psi \vdash K \succeq M;\ distinct\ xvec;\ set\ xvec \subseteq supp\ N;\ length\ xvec = length\ Tvec]\!]$
  $\Longrightarrow \Psi \rhd M(\!|\lambda*xvec\ N|\!).P \longmapsto ¿K(\!|N[xvec::=Tvec]|\!) \prec P[xvec::=Tvec]$

- Theorem *BrOutput*:

  $\Psi \vdash M \preceq K \Longrightarrow \Psi \rhd M\langle N\rangle.P \longmapsto ¡K\langle N\rangle \prec P$

- Theorem *BrMerge*:

  $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;$
  $\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ A_P \sharp* \Psi;$
  $A_P \sharp* P;$
  $A_P \sharp* Q;\ A_P \sharp* M;\ A_P \sharp* A_Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M]\!]$
  $\Longrightarrow \Psi \rhd P \parallel Q \longmapsto ¿M(\!|N|\!) \prec P' \parallel Q'$

- Theorems *BrComm1* and *BrComm2*:

  $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¿M(\!|N|\!) \prec P';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;$
  $\Psi \otimes \Psi_P \rhd Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec Q';\ extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ A_P \sharp* \Psi;$
  $A_P \sharp* P;\ A_P \sharp* Q;\ A_P \sharp* M;\ A_P \sharp* A_Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;$
  $xvec \sharp* P]\!]$
  $\Longrightarrow \Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P' \parallel Q'$

  $[\![\Psi \otimes \Psi_Q \rhd P \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P';\ extractFrame\ P = \langle A_P,\ \Psi_P\rangle;$
  $\Psi \otimes \Psi_P \rhd Q \longmapsto ¿M(\!|N|\!) \prec Q';\ extractFrame\ Q = \langle A_Q,\ \Psi_Q\rangle;\ A_P \sharp* \Psi;$
  $A_P \sharp* P;$
  $A_P \sharp* Q;\ A_P \sharp* M;\ A_P \sharp* A_Q;\ A_Q \sharp* \Psi;\ A_Q \sharp* P;\ A_Q \sharp* Q;\ A_Q \sharp* M;$
  $xvec \sharp* Q]\!]$
  $\Longrightarrow \Psi \rhd P \parallel Q \longmapsto ¡M(\!|\nu*xvec|\!)\langle N\rangle \prec P' \parallel Q'$

- Theorem *BrClose*:

$$[\![\Psi \rhd P \longmapsto ¡M(\!|\nu{*}xvec|\!)\langle N\rangle \prec P';\ x \in supp\ M;\ x \sharp \Psi]\!]$$
$$\Longrightarrow \Psi \rhd (\!|\nu x|\!)P \longmapsto \tau \prec (\!|\nu x|\!)((\!|\nu{*}xvec|\!)P')$$

- Theorem *BrOpen*:

$$[\![\Psi \rhd P \longmapsto ¡M(\!|\nu{*}(xvec\ @\ yvec)|\!)\langle N\rangle \prec P';\ x \in supp\ N;\ x \sharp \Psi;\ x \sharp M;\ x \sharp xvec;$$
$$x \sharp yvec]\!]$$
$$\Longrightarrow \Psi \rhd (\!|\nu x|\!)P \longmapsto ¡M(\!|\nu{*}(xvec\ @\ x\ \#\ yvec)|\!)\langle N\rangle \prec P'$$

2.2.3 Requirements for broadcast – Definition 6

- Theorem *chanOutConSupp*:

$$\Psi \vdash M \preceq N \Longrightarrow supp\ N \subseteq supp\ M$$

- Theorem *chanInConSupp*:

$$\Psi \vdash N \succeq M \Longrightarrow supp\ N \subseteq supp\ M$$

2.2.4 Strong bisimulation – Definition 4

- Theorem *bisim.step*:

$$[\![insertAssertion\ (extractFrame\ P)\ \Psi \simeq_F insertAssertion\ (extractFrame\ Q)\ \Psi;$$
$$\Psi \rhd P \rightsquigarrow[bisim]\ Q;\ \forall \Psi'.\ \Psi \otimes \Psi' \rhd P \sim Q;\ \Psi \rhd Q \sim P]\!]$$
$$\Longrightarrow \Psi \rhd P \sim Q$$

2.3 Theorems

2.3.1 Congruence properties of strong bisimulation – Theorem 8

- Theorem *bisimOutputPres*:

$$\Psi \rhd P \sim Q \Longrightarrow \Psi \rhd M\langle N\rangle.P \sim M\langle N\rangle.Q$$

- Theorem *bisimInputPres*:

$$(\bigwedge Tvec.\ length\ xvec = length\ Tvec \Longrightarrow \Psi \rhd P[xvec::=Tvec] \sim Q[xvec::=Tvec])$$
$$\Longrightarrow$$
$$\Psi \rhd M(\!|\lambda{*}xvec\ N|\!).P \sim M(\!|\lambda{*}xvec\ N|\!).Q$$

- Theorem *bisimCasePres*:

  $[\![\bigwedge\varphi\ P.\ (\varphi,\ P)\ mem\ CsP \Longrightarrow \exists\, Q.\ (\varphi,\ Q)\ mem\ CsQ \wedge guarded\ Q \wedge \Psi \rhd P \sim Q;$
  $\bigwedge\varphi\ Q.\ (\varphi,\ Q)\ mem\ CsQ \Longrightarrow \exists\, P.\ (\varphi,\ P)\ mem\ CsP \wedge guarded\ P \wedge \Psi \rhd P \sim Q]\!]$
  $\Longrightarrow \Psi \rhd Cases\ CsP \sim Cases\ CsQ$

- Theorems *bisimParPres* and *bisimParPresSym*:

  $\Psi \rhd P \sim Q \Longrightarrow \Psi \rhd P \parallel R \sim Q \parallel R$

  $\Psi \rhd P \sim Q \Longrightarrow \Psi \rhd R \parallel P \sim R \parallel Q$

- Theorem *bisimResPres*:

  $[\![\Psi \rhd P \sim Q;\ x \sharp \Psi]\!] \Longrightarrow \Psi \rhd (\!|\nu x|\!)P \sim (\!|\nu x|\!)Q$

- Theorem *bisimBangPres*:

  $[\![\Psi \rhd P \sim Q;\ guarded\ P;\ guarded\ Q]\!] \Longrightarrow \Psi \rhd\ !P \sim\ !Q$

2.3.2 Strong congruence, bisimulation closed under substitution – Definition 9

- Theorem *closeSubst_def*:

  $closeSubst\ Rel \equiv$
  $\{(\Psi,\ P,\ Q)\ |\Psi\ P\ Q.\ \forall\sigma.\ wellFormedSubst\ \sigma \longrightarrow (\Psi,\ P[<\sigma>],\ Q[<\sigma>]) \in Rel\}$

- $\Psi \rhd P \sim_s Q$

2.3.3 Strong congruence is a process congruence for all Ψ – Theorem 10

- Theorem *bisimSubstOutputPres*:

  $\Psi \rhd P \sim_s Q \Longrightarrow \Psi \rhd M\langle N\rangle.P \sim_s M\langle N\rangle.Q$

- Theorem *bisimSubstInputPres*:

  $[\![\Psi \rhd P \sim_s Q;\ xvec \sharp* \Psi;\ distinct\ xvec]\!] \Longrightarrow \Psi \rhd M(\!|\lambda* xvec\ N|\!).P \sim_s M(\!|\lambda* xvec\ N|\!).Q$

- Theorem *bisimSubstCasePres*:

  $[\![length\ CsP = length\ CsQ;$
  $\bigwedge i\ \varphi\ P\ \varphi'\ Q.$
  $\quad [\![i \leq length\ CsP;\ (\varphi,\ P) = CsP\ !\ i;\ (\varphi',\ Q) = CsQ\ !\ i]\!]$
  $\quad \Longrightarrow \varphi = \varphi' \wedge \Psi \rhd P \sim_s Q \wedge guarded\ P \wedge guarded\ Q]\!]$
  $\Longrightarrow \Psi \rhd Cases\ CsP \sim_s Cases\ CsQ$

- Theorem *bisimSubstParPres*:

  $\Psi \rhd P \sim_s Q \Longrightarrow \Psi \rhd P \parallel R \sim_s Q \parallel R$

- Theorem *bisimSubstResPres*:

  $[\![\Psi \rhd P \sim_s Q;\ x \sharp \Psi]\!] \Longrightarrow \Psi \rhd (\!|\nu x|\!) P \sim_s (\!|\nu x|\!) Q$

- Theorem *bisimSubstBangPres*:

  $[\![\Psi \rhd P \sim_s Q;\ guarded\ P;\ guarded\ Q]\!] \Longrightarrow \Psi \rhd\ !P \sim_s\ !Q$

2.3.4 Structural equivalence – Theorem 11

- Theorem *bisimCasePushRes*:

  $x \sharp map\ fst\ Cs \Longrightarrow \Psi \rhd (\!|\nu x|\!)(Cases\ Cs) \sim Cases\ map\ (\lambda(\varphi,\ P).\ (\varphi,\ (\!|\nu x|\!) P))\ Cs$

- Theorem *bisimInputPushRes*:

  $[\![x \sharp M;\ x \sharp xvec;\ x \sharp N]\!] \Longrightarrow \Psi \rhd (\!|\nu x|\!)(M(\!|\lambda * xvec\ N|\!).P) \sim M(\!|\lambda * xvec\ N|\!).(\!|\nu x|\!) P$

- Theorem *bisimOutputPushRes*:

  $[\![x \sharp M;\ x \sharp N]\!] \Longrightarrow \Psi \rhd (\!|\nu x|\!)(M\langle N\rangle.P) \sim M\langle N\rangle.(\!|\nu x|\!) P$

- Theorem *bisimParAssoc*:

  $\Psi \rhd P \parallel Q \parallel R \sim P \parallel (Q \parallel R)$

- Theorem *bisimParComm*:

  $\Psi \rhd P \parallel Q \sim Q \parallel P$

- Theorem *bisimResNil*:

$\Psi \rhd (\!|\nu x|\!)\mathbf{0} \sim \mathbf{0}$

- Theorem *bisimScopeExt*:

  $x \sharp P \Longrightarrow \Psi \rhd (\!|\nu x|\!)(P \parallel Q) \sim P \parallel (\!|\nu x|\!)Q$

- Theorem *bisimResComm*:

  $\Psi \rhd (\!|\nu x|\!)((\!|\nu y|\!)P) \sim (\!|\nu y|\!)((\!|\nu x|\!)P)$

- Theorem *bangExt*:

  $guarded\ P \Longrightarrow \Psi \rhd !P \sim P \parallel !P$

- Theorem *bisimParNil*:

  $\Psi \rhd P \parallel \mathbf{0} \sim P$

2.3.5 Support of processes in broadcast transitions – Lemma 14

- Theorem *brInputTermSupp*:

  $\Psi \rhd P \longmapsto ¿K(\!|N|\!) \prec P' \Longrightarrow supp\ K \subseteq supp\ P$

- Theorem *brOutputTermSupp*:

  $\Psi \rhd P \longmapsto RBrOut\ K\ ((\!|\nu{*}xvec|\!)N \prec' P') \Longrightarrow supp\ K \subseteq supp\ P$

end

end

References

[1] Jesper Bengtson. Psi-calculi in Isabelle. *Arch. Formal Proofs*, 2012. https://www.isa-afp.org/entries/Psi_Calculi.html, Formal proof development.

[2] Jesper Bengtson, Magnus Johansson, Joachim Parrow, and Björn Victor. Psi-calculi: a framework for mobile processes with nominal data and logic. *Log. Methods Comput. Sci.*, 7(1), 2011.

[3] Jesper Bengtson, Joachim Parrow, and Tjark Weber. Psi-calculi in Isabelle. *J. Autom. Reason.*, 56(1):1–47, 2016.

[4] Johannes Borgström, Shuqin Huang, Magnus Johansson, Palle Raabjerg, Björn Victor, Johannes Åman Pohjola, and Joachim Parrow. Broadcast Psi-calculi with an application to wireless protocols. In Gilles Barthe, Alberto Pardo, and Gerardo Schneider, editors, *Software Engineering and Formal Methods - 9th International Conference, SEFM 2011, Montevideo, Uruguay, November 14-18, 2011. Proceedings*, volume 7041 of *Lecture Notes in Computer Science*, pages 74–89. Springer, 2011.

[5] Johannes Borgström, Shuqin Huang, Magnus Johansson, Palle Raabjerg, Björn Victor, Johannes Åman Pohjola, and Joachim Parrow. Broadcast psi-calculi with an application to wireless protocols. *Softw. Syst. Model.*, 14(1):201–216, 2015.