Abstract

We utilize and extend the method of *shallow semantic embeddings* (SSEs) in classical higher-order logic (HOL) to construct a custom theorem proving environment for *abstract objects theory* (AOT) on the basis of Isabelle/HOL.

SSEs are a means for universal logical reasoning by translating a target logic to HOL using a representation of its semantics. AOT is a foundational metaphysical theory, developed by Edward Zalta, that explains the objects presupposed by the sciences as *abstract objects* that reify property patterns. In particular, AOT aspires to provide a philosphically grounded basis for the construction and analysis of the objects of mathematics.

We can support this claim by verifying Uri Nodelman's and Edward Zalta's reconstruction of Frege's theorem: we can confirm that the Dedekind-Peano postulates for natural numbers are consistently derivable in AOT using Frege's method. Furthermore, we can suggest and discuss generalizations and variants of the construction and can thereby provide theoretical insights into, and contribute to the philosophical justification of, the construction.

In the process, we can demonstrate that our method allows for a nearly transparent exchange of results between traditional pen-and-paper-based reasoning and the computerized implementation, which in turn can retain the automation mechanisms available for Isabelle/HOL.

During our work, we could significantly contribute to the evolution of our target theory itself, while simultaneously solving the technical challenge of using an SSE to implement a theory that is based on logical foundations that significantly differ from the meta-logic HOL.

In general, our results demonstrate the fruitfulness of the practice of Computational Metaphysics, i.e. the application of computational methods to metaphysical questions and theories.

A full description of this formalization including references can be found at http://dx.doi.org/10.17169/refubium-35141.

The version of Principia Logico-Metaphysica (PLM) implemented in this formalization can be found at http://mally.stanford.edu/principia-2021-10-13.pdf, while the latest version of PLM is available at http://mally.stanford.edu/principia.pdf.

Contents

1 References

A full description of this formalization including references can be found at http://dx.doi.org/10.17169/refubium-35141.

The version of Principia Logico-Metaphysica (PLM) implemented in this formalization can be found at http://mally.stanford.edu/principia-2021-10-13.pdf, while the latest version of PLM is available at http://mally.stanford.edu/principia.pdf.

2 Model for the Logic of AOT

We introduce a primitive type for hyperintensional propositions.

typedecl o

To be able to model modal operators following Kripke semantics, we introduce a primitive type for possible worlds and assert, by axiom, that there is a surjective function mapping propositions to the boolean-valued functions acting on possible worlds. We call the result of applying this function to a proposition the Montague intension of the proposition.

typedecl *w* — The primtive type of possible worlds.
axiomatization *AOT-model-d*o :: ‹o⇒(*w*⇒*bool*)› **where**
  *d*o*-surj*: ‹*surj AOT-model-d*o›

The axioms of PLM require the existence of a non-actual world.

consts w_0 :: *w* — The designated actual world.
axiomatization where *AOT-model-nonactual-world*: ‹$\exists$ *w* . $w \neq w_0$›

Validity of a proposition in a given world can now be modelled as the result of applying that world to the Montague intension of the proposition.

definition *AOT-model-valid-in* :: ‹*w*⇒o⇒*bool*› **where**
  ‹*AOT-model-valid-in w* φ ≡ *AOT-model-d*o φ *w*›

By construction, we can choose a proposition for any given Montague intension, s.t. the proposition is valid in a possible world iff the Montague intension evaluates to true at that world.

definition *AOT-model-proposition-choice* :: ‹(*w*⇒*bool*) ⇒ o› (**binder** ‹ε_o › *8*)
  where ‹ε_o *w*. φ *w* ≡ (*inv AOT-model-d*o) φ›
lemma *AOT-model-proposition-choice-simp*: ‹*AOT-model-valid-in w* (ε_o *w*. φ *w*) = φ *w*›
  ⟨*proof*⟩

Nitpick can trivially show that there are models for the axioms above.

lemma ‹*True*› **nitpick**[*satisfy*, *user-axioms*, *expect* = *genuine*] ⟨*proof*⟩

typedecl ω — The primtive type of ordinary objects/urelements.

Validating extended relation comprehension requires a large set of special urelements. For simple models that do not validate extended relation comprehension (and consequently the predecessor axiom in the theory of natural numbers), it suffices to use a primitive type as σ, i.e. **typedecl** σ.

typedecl σ'
typedef σ = ‹*UNIV*::((ω ⇒ *w* ⇒ *bool*) *set* × (ω ⇒ *w* ⇒ *bool*) *set* × σ') *set*› ⟨*proof*⟩

typedecl *null* — Null-urelements representing non-denoting terms.

datatype υ = $\omega\upsilon$ ω | $\sigma\upsilon$ σ | *is-nullv*: *nullv null* — Type of urelements

Urrelations are proposition-valued functions on urelements. Urrelations are required to evaluate to necessarily false propositions for null-urelements (note that there may be several distinct necessarily false propositions).

typedef *urrel* = ‹{ φ . ∀ *x w* . ¬*AOT-model-valid-in w* (φ (*nullv x*)) }›
  ⟨*proof*⟩

Abstract objects will be modelled as sets of urrelations and will have to be mapped surjectively into the set of special urelements. We show that any mapping from abstract objects to special urelements has to involve at least one large set of collapsed abstract objects. We will use this fact to extend arbitrary mappings from abstract objects to special urelements to surjective mappings.

lemma *ασ-pigeonhole*:
— For any arbitrary mapping $\alpha\sigma$ from sets of urrelations to special urelements, there exists an abstract object x, s.t. the cardinal of the set of special urelements is strictly smaller than the cardinal of the set of abstract objects that are mapped to the same urelement as x under $\alpha\sigma$.
‹$\exists\, x\ .\ |UNIV{::}\sigma\ set| <o\ |\{y\ .\ \alpha\sigma\ x = \alpha\sigma\ y\}|$›
for $\alpha\sigma$ :: ‹*urrel set* ⇒ σ›
⟨*proof*⟩

We introduce a mapping from abstract objects (i.e. sets of urrelations) to special urelements $\alpha\sigma$ that is surjective and distinguishes all abstract objects that are distinguished by a (not necessarily surjective) mapping $\alpha\sigma'$. $\alpha\sigma'$ will be used to model extended relation comprehension.

consts $\alpha\sigma'$ :: ‹*urrel set* ⇒ σ›
consts $\alpha\sigma$ :: ‹*urrel set* ⇒ σ›

specification($\alpha\sigma$)
ασ-surj: ‹*surj* $\alpha\sigma$›
ασ-ασ': ‹$\alpha\sigma\ x = \alpha\sigma\ y \Longrightarrow \alpha\sigma'\ x = \alpha\sigma'\ y$›
⟨*proof*⟩

For extended models that validate extended relation comprehension (and consequently the predecessor axiom), we specify which abstract objects are distinguished by $\alpha\sigma'$.

definition *urrel-to-ωrel* :: ‹*urrel* ⇒ (ω ⇒ *w* ⇒ *bool*)› **where**
‹*urrel-to-ωrel* ≡ λ *r u w* . *AOT-model-valid-in w* (*Rep-urrel r* (ωυ *u*))›
definition *ωrel-to-urrel* :: ‹(ω ⇒ *w* ⇒ *bool*) ⇒ *urrel*› **where**
‹*ωrel-to-urrel* ≡ λ φ . *Abs-urrel*
(λ *u* . $\varepsilon_\circ$ *w* . *case u of* ωυ *x* ⇒ φ *x w* | - ⇒ *False*)›

definition *AOT-urrel-ωequiv* :: ‹*urrel* ⇒ *urrel* ⇒ *bool*› **where**
‹*AOT-urrel-ωequiv* ≡ λ *r s* . ∀ *u v* . *AOT-model-valid-in v* (*Rep-urrel r* (ωυ *u*)) =
AOT-model-valid-in v (*Rep-urrel s* (ωυ *u*))›

lemma *urrel-ωrel-quot*: ‹*Quotient3 AOT-urrel-ωequiv urrel-to-ωrel ωrel-to-urrel*›
⟨*proof*⟩

specification ($\alpha\sigma'$)
ασ-eq-ord-exts-all:
‹$\alpha\sigma'\ a = \alpha\sigma'\ b \Longrightarrow (\bigwedge s\ .\ \textit{urrel-to-}\omega\textit{rel}\ s = \textit{urrel-to-}\omega\textit{rel}\ r \Longrightarrow s \in a)$
$\Longrightarrow (\bigwedge s\ .\ \textit{urrel-to-}\omega\textit{rel}\ s = \textit{urrel-to-}\omega\textit{rel}\ r \Longrightarrow s \in b)$›
ασ-eq-ord-exts-ex:
‹$\alpha\sigma'\ a = \alpha\sigma'\ b \Longrightarrow (\exists\ s\ .\ s \in a \wedge \textit{urrel-to-}\omega\textit{rel}\ s = \textit{urrel-to-}\omega\textit{rel}\ r)$
$\Longrightarrow (\exists\, s\ .\ s \in b \wedge \textit{urrel-to-}\omega\textit{rel}\ s = \textit{urrel-to-}\omega\textit{rel}\ r)$›
⟨*proof*⟩

We enable the extended model version.

abbreviation (*input*) *AOT-ExtendedModel* **where** ‹*AOT-ExtendedModel* ≡ *True*›

Individual terms are either ordinary objects, represented by ordinary urelements, abstract objects, modelled as sets of urrelations, or null objects, used to represent non-denoting definite descriptions.

datatype κ = ωκ ω | ακ ‹*urrel set*› | *is-nullκ*: *nullκ null*

The mapping from abstract objects to urelements can be naturally lifted to a surjective mapping from individual terms to urelements.

primrec κυ :: ‹κ⇒υ› **where**
‹κυ (ωκ *x*) = ωυ *x*›
| ‹κυ (ακ *x*) = συ (ασ *x*)›
| ‹κυ (*nullκ x*) = *nullυ x*›

lemma *κυ-surj*: ‹*surj κυ*›
⟨*proof*⟩

By construction if the urelement of an individual term is exemplified by an urrelation, it cannot be a null-object.

lemma *urrel-null-false*:
assumes ‹*AOT-model-valid-in w* (*Rep-urrel f* (*κυ x*))›
shows ‹¬*is-nullκ x*›
⟨*proof*⟩

AOT requires any ordinary object to be *possibly concrete* and that there is an object that is not actually, but possibly concrete.

consts *AOT-model-concreteω* :: ‹*ω* ⇒ *w* ⇒ *bool*›
specification (*AOT-model-concreteω*)
AOT-model-ω-concrete-in-some-world:
‹∃ *w* . *AOT-model-concreteω x w*›
AOT-model-contingent-object:
‹∃ *x w* . *AOT-model-concreteω x w* ∧ ¬*AOT-model-concreteω x* w_0›
⟨*proof*⟩

We define a type class for AOT's terms specifying the conditions under which objects of that type denote and require the set of denoting terms to be non-empty.

class *AOT-Term* =
fixes *AOT-model-denotes* :: ‹′*a* ⇒ *bool*›
assumes *AOT-model-denoting-ex*: ‹∃ *x* . *AOT-model-denotes x*›

All types except the type of propositions involve non-denoting terms. We define a refined type class for those.

class *AOT-IncompleteTerm* = *AOT-Term* +
assumes *AOT-model-nondenoting-ex*: ‹∃ *x* . ¬*AOT-model-denotes x*›

Generic non-denoting term.

definition *AOT-model-nondenoting* :: ‹′*a*::*AOT-IncompleteTerm*› **where**
‹*AOT-model-nondenoting* ≡ *SOME τ* . ¬*AOT-model-denotes τ*›
lemma *AOT-model-nondenoing*: ‹¬*AOT-model-denotes* (*AOT-model-nondenoting*)›
⟨*proof*⟩

AOT-model-denotes can trivially be extended to products of types.

instantiation *prod* :: (*AOT-Term*, *AOT-Term*) *AOT-Term*
begin
definition *AOT-model-denotes-prod* :: ‹′*a*×′*b* ⇒ *bool*› **where**
‹*AOT-model-denotes-prod* ≡ λ(*x*,*y*) . *AOT-model-denotes x* ∧ *AOT-model-denotes y*›
instance ⟨*proof*⟩
end

We specify a transformation of proposition-valued functions on terms, s.t. the result is fully determined by *regular* terms. This will be required for modelling n-ary relations as functions on tuples while preserving AOT's definition of n-ary relation identity.

locale *AOT-model-irregular-spec* =
fixes *AOT-model-irregular* :: ‹(′*a* ⇒ o) ⇒ ′*a* ⇒ o›
and *AOT-model-regular* :: ‹′*a* ⇒ *bool*›
and *AOT-model-term-equiv* :: ‹′*a* ⇒ ′*a* ⇒ *bool*›
assumes *AOT-model-irregular-false*:
‹¬*AOT-model-valid-in w* (*AOT-model-irregular φ x*)›
assumes *AOT-model-irregular-equiv*:
‹*AOT-model-term-equiv x y* ⟹
AOT-model-irregular φ x = *AOT-model-irregular φ y*›
assumes *AOT-model-irregular-eqI*:
‹(⋀ *x* . *AOT-model-regular x* ⟹ *φ x* = *ψ x*) ⟹
AOT-model-irregular φ x = *AOT-model-irregular ψ x*›

We introduce a type class for individual terms that specifies being regular, being equivalent (i.e. conceptually *sharing urelements*) and the transformation on proposition-valued functions as specified above.

```
class AOT-IndividualTerm = AOT-IncompleteTerm +
  fixes AOT-model-regular :: ‹'a ⇒ bool›
  fixes AOT-model-term-equiv :: ‹'a ⇒ 'a ⇒ bool›
  fixes AOT-model-irregular :: ‹('a ⇒ o) ⇒ 'a ⇒ o›
  assumes AOT-model-irregular-nondenoting:
    ‹¬AOT-model-regular x ⟹ ¬AOT-model-denotes x›
  assumes AOT-model-term-equiv-part-equivp:
    ‹equivp AOT-model-term-equiv›
  assumes AOT-model-term-equiv-denotes:
    ‹AOT-model-term-equiv x y ⟹ (AOT-model-denotes x = AOT-model-denotes y)›
  assumes AOT-model-term-equiv-regular:
    ‹AOT-model-term-equiv x y ⟹ (AOT-model-regular x = AOT-model-regular y)›
  assumes AOT-model-irregular:
    ‹AOT-model-irregular-spec AOT-model-irregular AOT-model-regular
                              AOT-model-term-equiv›

interpretation AOT-model-irregular-spec AOT-model-irregular AOT-model-regular
                                        AOT-model-term-equiv
  ⟨proof⟩
```

Our concrete type for individual terms satisfies the type class of individual terms. Note that all unary individuals are regular. In general, an individual term may be a tuple and is regular, if at most one tuple element does not denote.

```
instantiation κ :: AOT-IndividualTerm
begin
definition AOT-model-term-equiv-κ :: ‹κ ⇒ κ ⇒ bool› where
  ‹AOT-model-term-equiv-κ ≡ λ x y . κυ x = κυ y›
definition AOT-model-denotes-κ :: ‹κ ⇒ bool› where
  ‹AOT-model-denotes-κ ≡ λ x . ¬is-nullκ x›
definition AOT-model-regular-κ :: ‹κ ⇒ bool› where
  ‹AOT-model-regular-κ ≡ λ x . True›
definition AOT-model-irregular-κ :: ‹(κ ⇒ o) ⇒ κ ⇒ o› where
  ‹AOT-model-irregular-κ ≡ SOME φ . AOT-model-irregular-spec φ
                                    AOT-model-regular AOT-model-term-equiv›
instance ⟨proof⟩
end
```

We define relations among individuals as proposition valued functions. *Denoting* unary relations (among κ) will match the urelations introduced above.

```
typedef 'a rel (‹<->›) = ‹UNIV::('a::AOT-IndividualTerm ⇒ o) set› ⟨proof⟩
setup-lifting type-definition-rel
```

We will use the transformation specified above to "fix" the behaviour of functions on irregular terms when defining λ-expressions.

```
definition fix-irregular :: ‹('a::AOT-IndividualTerm ⇒ o) ⇒ ('a ⇒ o)› where
  ‹fix-irregular ≡ λ φ x . if AOT-model-regular x
                           then φ x else AOT-model-irregular φ x›
lemma fix-irregular-denoting:
  ‹AOT-model-denotes x ⟹ fix-irregular φ x = φ x›
  ⟨proof⟩
lemma fix-irregular-regular:
  ‹AOT-model-regular x ⟹ fix-irregular φ x = φ x›
  ⟨proof⟩
lemma fix-irregular-irregular:
  ‹¬AOT-model-regular x ⟹ fix-irregular φ x = AOT-model-irregular φ x›
  ⟨proof⟩
```

Relations among individual terms are (potentially non-denoting) terms. A relation denotes, if it agrees on all equivalent terms (i.e. terms sharing urelements), is necessarily false on all non-denoting terms and is well-behaved on irregular terms.

```
instantiation rel :: (AOT-IndividualTerm) AOT-IncompleteTerm
begin

lift-definition AOT-model-denotes-rel :: ‹<'a> ⇒ bool› is
  ‹λ φ . (∀ x y . AOT-model-term-equiv x y ⟶ φ x = φ y) ∧
       (∀ w x . AOT-model-valid-in w (φ x) ⟶ AOT-model-denotes x) ∧
       (∀ x . ¬AOT-model-regular x ⟶ φ x = AOT-model-irregular φ x)› ⟨proof⟩
instance ⟨proof⟩
end
```

Auxiliary lemmata.

```
lemma AOT-model-term-equiv-eps:
  shows ‹AOT-model-term-equiv (Eps (AOT-model-term-equiv κ)) κ›
    and ‹AOT-model-term-equiv κ (Eps (AOT-model-term-equiv κ))›
    and ‹AOT-model-term-equiv κ κ' ⟹
         (Eps (AOT-model-term-equiv κ)) = (Eps (AOT-model-term-equiv κ'))›
  ⟨proof⟩

lemma AOT-model-denotes-Abs-rel-fix-irregularI:
  assumes ‹⋀ x y . AOT-model-term-equiv x y ⟹ φ x = φ y›
      and ‹⋀ w x . AOT-model-valid-in w (φ x) ⟹ AOT-model-denotes x›
    shows ‹AOT-model-denotes (Abs-rel (fix-irregular φ))›
⟨proof⟩

lemma AOT-model-term-equiv-rel-equiv:
  assumes ‹AOT-model-denotes x›
      and ‹AOT-model-denotes y›
    shows ‹AOT-model-term-equiv x y = (∀ Π w . AOT-model-denotes Π ⟶
           AOT-model-valid-in w (Rep-rel Π x) = AOT-model-valid-in w (Rep-rel Π y))›
⟨proof⟩
```

Denoting relations among terms of type κ correspond to urrelations.

```
definition rel-to-urrel :: ‹<κ> ⇒ urrel› where
  ‹rel-to-urrel ≡ λ Π . Abs-urrel (λ u . Rep-rel Π (SOME x . κυ x = u))›
definition urrel-to-rel :: ‹urrel ⇒ <κ>› where
  ‹urrel-to-rel ≡ λ φ . Abs-rel (λ x . Rep-urrel φ (κυ x))›
definition AOT-rel-equiv :: ‹<'a::AOT-IndividualTerm> ⇒ <'a> ⇒ bool› where
  ‹AOT-rel-equiv ≡ λ f g . AOT-model-denotes f ∧ AOT-model-denotes g ∧ f = g›

lemma urrel-quotient3: ‹Quotient3 AOT-rel-equiv rel-to-urrel urrel-to-rel›
⟨proof⟩

lemma urrel-quotient:
  ‹Quotient AOT-rel-equiv rel-to-urrel urrel-to-rel
           (λx y. AOT-rel-equiv x x ∧ rel-to-urrel x = y)›
  ⟨proof⟩
```

Unary individual terms are always regular and equipped with encoding and concreteness. The specification of the type class anticipates the required properties for deriving the axiom system.

```
class AOT-UnaryIndividualTerm =
  fixes AOT-model-enc :: ‹'a ⇒ <'a::AOT-IndividualTerm> ⇒ bool›
    and AOT-model-concrete :: ‹w ⇒ 'a ⇒ bool›
  assumes AOT-model-unary-regular:
      ‹AOT-model-regular x› — All unary individual terms are regular.
      and AOT-model-enc-relid:
        ‹AOT-model-denotes F ⟹
         AOT-model-denotes G ⟹
         (⋀ x . AOT-model-enc x F ⟷ AOT-model-enc x G)
         ⟹ F = G›
      and AOT-model-A-objects:
        ‹∃ x . AOT-model-denotes x ∧
             (∀ w. ¬ AOT-model-concrete w x) ∧
```

```
      (∀F. AOT-model-denotes F ⟶ AOT-model-enc x F = φ F)›
  and AOT-model-contingent:
    ‹∃ x w. AOT-model-concrete w x ∧ ¬ AOT-model-concrete w0 x›
  and AOT-model-nocoder:
    ‹AOT-model-concrete w x ⟹ ¬AOT-model-enc x F›
  and AOT-model-concrete-equiv:
    ‹AOT-model-term-equiv x y ⟹
      AOT-model-concrete w x = AOT-model-concrete w y›
  and AOT-model-concrete-denotes:
    ‹AOT-model-concrete w x ⟹ AOT-model-denotes x›
  — The following are properties that will only hold in the extended models.
  and AOT-model-enc-indistinguishable-all:
   ‹AOT-ExtendedModel ⟹
    AOT-model-denotes a ⟹ ¬(∃ w . AOT-model-concrete w a) ⟹
    AOT-model-denotes b ⟹ ¬(∃ w . AOT-model-concrete w b) ⟹
    AOT-model-denotes Π ⟹
    (⋀ Π' . AOT-model-denotes Π' ⟹
      (⋀ v . AOT-model-valid-in v (Rep-rel Π' a) =
              AOT-model-valid-in v (Rep-rel Π' b))) ⟹
    (⋀ Π' . AOT-model-denotes Π' ⟹
        (⋀ v x . ∃ w . AOT-model-concrete w x ⟹
            AOT-model-valid-in v (Rep-rel Π' x) =
            AOT-model-valid-in v (Rep-rel Π x)) ⟹
        AOT-model-enc a Π') ⟹
    (⋀ Π' . AOT-model-denotes Π' ⟹
        (⋀ v x . ∃ w . AOT-model-concrete w x ⟹
            AOT-model-valid-in v (Rep-rel Π' x) =
            AOT-model-valid-in v (Rep-rel Π x)) ⟹
        AOT-model-enc b Π')›
  and AOT-model-enc-indistinguishable-ex:
   ‹AOT-ExtendedModel ⟹
    AOT-model-denotes a ⟹ ¬(∃ w . AOT-model-concrete w a) ⟹
    AOT-model-denotes b ⟹ ¬(∃ w . AOT-model-concrete w b) ⟹
    AOT-model-denotes Π ⟹
    (⋀ Π' . AOT-model-denotes Π' ⟹
      (⋀ v . AOT-model-valid-in v (Rep-rel Π' a) =
              AOT-model-valid-in v (Rep-rel Π' b))) ⟹
    (∃ Π' . AOT-model-denotes Π' ∧ AOT-model-enc a Π' ∧
        (∀ v x . (∃ w . AOT-model-concrete w x) ⟶
            AOT-model-valid-in v (Rep-rel Π' x) =
            AOT-model-valid-in v (Rep-rel Π x))) ⟹
    (∃ Π' . AOT-model-denotes Π' ∧ AOT-model-enc b Π' ∧
        (∀ v x . (∃ w . AOT-model-concrete w x) ⟶
            AOT-model-valid-in v (Rep-rel Π' x) =
            AOT-model-valid-in v (Rep-rel Π x)))›
```

Instantiate the class of unary individual terms for our concrete type of individual terms κ.

```
instantiation κ :: AOT-UnaryIndividualTerm
begin

definition AOT-model-enc-κ :: ‹κ ⇒ <κ> ⇒ bool› where
  ‹AOT-model-enc-κ ≡ λ x F .
      case x of ακ a ⇒ AOT-model-denotes F ∧ rel-to-urrel F ∈ a
              | - ⇒ False›
primrec AOT-model-concrete-κ :: ‹w ⇒ κ ⇒ bool› where
  ‹AOT-model-concrete-κ w (ωκ x) = AOT-model-concreteω x w›
| ‹AOT-model-concrete-κ w (ακ x) = False›
| ‹AOT-model-concrete-κ w (nullκ x) = False›

lemma AOT-meta-A-objects-κ:
  ‹∃x :: κ. AOT-model-denotes x ∧
            (∀w. ¬ AOT-model-concrete w x) ∧
            (∀F. AOT-model-denotes F ⟶ AOT-model-enc x F = φ F)› for φ
```

```
  ⟨proof⟩

instance ⟨proof⟩
end
```

Products of unary individual terms and individual terms are individual terms. A tuple is regular, if at most one element does not denote. I.e. a pair is regular, if the first (unary) element denotes and the second is regular (i.e. at most one of its recursive tuple elements does not denote), or the first does not denote, but the second denotes (i.e. all its recursive tuple elements denote).

```
instantiation prod :: (AOT-UnaryIndividualTerm, AOT-IndividualTerm) AOT-IndividualTerm
begin
definition AOT-model-regular-prod :: ‹'a×'b ⇒ bool› where
  ‹AOT-model-regular-prod ≡ λ (x,y) . AOT-model-denotes x ∧ AOT-model-regular y ∨
                                      ¬AOT-model-denotes x ∧ AOT-model-denotes y›
definition AOT-model-term-equiv-prod :: ‹'a×'b ⇒ 'a×'b ⇒ bool› where
  ‹AOT-model-term-equiv-prod ≡  λ (x1,y1) (x2,y2) .
    AOT-model-term-equiv x1 x2 ∧ AOT-model-term-equiv y1 y2›
function AOT-model-irregular-prod :: ‹('a×'b ⇒ o) ⇒ 'a×'b ⇒ o› where
  AOT-model-irregular-proj2: ‹AOT-model-denotes x ⟹
    AOT-model-irregular φ (x,y) =
    AOT-model-irregular (λy. φ (SOME x' . AOT-model-term-equiv x x', y)) y›
| AOT-model-irregular-proj1: ‹¬AOT-model-denotes x ∧ AOT-model-denotes y ⟹
    AOT-model-irregular φ (x,y) =
    AOT-model-irregular (λx. φ (x, SOME y' . AOT-model-term-equiv y y')) x›
| AOT-model-irregular-prod-generic: ‹¬AOT-model-denotes x ∧ ¬AOT-model-denotes y ⟹
    AOT-model-irregular φ (x,y) =
    (SOME Φ . AOT-model-irregular-spec Φ AOT-model-regular AOT-model-term-equiv)
      φ (x,y)›
  ⟨proof⟩
termination ⟨proof⟩

instance ⟨proof⟩
end
```

Introduction rules for term equivalence on tuple terms.

```
lemma AOT-meta-prod-equivI:
  shows ⋀ (a::'a::AOT-UnaryIndividualTerm) x (y :: 'b::AOT-IndividualTerm) .
          AOT-model-term-equiv x y ⟹ AOT-model-term-equiv (a,x) (a,y)
    and ⋀ (x::'a::AOT-UnaryIndividualTerm) y (b :: 'b::AOT-IndividualTerm) .
          AOT-model-term-equiv x y ⟹ AOT-model-term-equiv (x,b) (y,b)
    ⟨proof⟩
```

The type of propositions are trivial instances of terms.

```
instantiation o :: AOT-Term
begin
definition AOT-model-denotes-o :: ‹o ⇒ bool› where
  ‹AOT-model-denotes-o ≡ λ-. True›
instance ⟨proof⟩
end
```

AOT's variables are modelled by restricting the type of terms to those terms that denote.

```
typedef 'a AOT-var = ‹{ x :: 'a::AOT-Term . AOT-model-denotes x }›
  morphisms AOT-term-of-var AOT-var-of-term
  ⟨proof⟩
```

Simplify automatically generated theorems and rules.

```
declare AOT-var-of-term-induct[induct del]
        AOT-var-of-term-cases[cases del]
        AOT-term-of-var-induct[induct del]
        AOT-term-of-var-cases[cases del]
lemmas AOT-var-of-term-inverse = AOT-var-of-term-inverse[simplified]
   and AOT-var-of-term-inject = AOT-var-of-term-inject[simplified]
```

and *AOT-var-of-term-induct* =
AOT-var-of-term-induct[*simplified*, *induct type*: *AOT-var*]
and *AOT-var-of-term-cases* =
AOT-var-of-term-cases[*simplified*, *cases type*: *AOT-var*]
and *AOT-term-of-var* = *AOT-term-of-var*[*simplified*]
and *AOT-term-of-var-cases* =
AOT-term-of-var-cases[*simplified*, *induct pred*: *AOT-term-of-var*]
and *AOT-term-of-var-induct* =
AOT-term-of-var-induct[*simplified*, *induct pred*: *AOT-term-of-var*]
and *AOT-term-of-var-inverse* = *AOT-term-of-var-inverse*[*simplified*]
and *AOT-term-of-var-inject* = *AOT-term-of-var-inject*[*simplified*]

Equivalence by definition is modelled as necessary equivalence.

consts *AOT-model-equiv-def* :: ‹o ⇒ o ⇒ *bool*›
specification(*AOT-model-equiv-def*)
AOT-model-equiv-def: ‹*AOT-model-equiv-def* φ ψ = (∀ *v* . *AOT-model-valid-in* *v* φ =
AOT-model-valid-in *v* ψ)›
⟨*proof*⟩

Identity by definition is modelled as identity for denoting terms plus co-denoting.

consts *AOT-model-id-def* :: ‹('*b* ⇒ '*a*::*AOT-Term*) ⇒ ('*b* ⇒ '*a*) ⇒ *bool*›
specification(*AOT-model-id-def*)
AOT-model-id-def: ‹(*AOT-model-id-def* τ σ) = (∀ α . *if AOT-model-denotes* (σ α)
then τ α = σ α
else ¬*AOT-model-denotes* (τ α))›
⟨*proof*⟩

To reduce definitions by identity without free variables to definitions by identity with free variables acting on the unit type, we give the unit type a trivial instantiation to *AOT-Term*.

instantiation *unit* :: *AOT-Term*
begin
definition *AOT-model-denotes-unit* :: ‹*unit* ⇒ *bool*› **where**
‹*AOT-model-denotes-unit* ≡ λ-. *True*›
instance ⟨*proof*⟩
end

Modally-strict and modally-fragile axioms are as necessary, resp. actually valid propositions.

definition *AOT-model-axiom* **where**
‹*AOT-model-axiom* ≡ λ φ . ∀ *v* . *AOT-model-valid-in* *v* φ›
definition *AOT-model-act-axiom* **where**
‹*AOT-model-act-axiom* ≡ λ φ . *AOT-model-valid-in* w_0 φ›

lemma *AOT-model-axiomI*:
assumes ‹⋀*v* . *AOT-model-valid-in* *v* φ›
shows ‹*AOT-model-axiom* φ›
⟨*proof*⟩

lemma *AOT-model-act-axiomI*:
assumes ‹*AOT-model-valid-in* w_0 φ›
shows ‹*AOT-model-act-axiom* φ›
⟨*proof*⟩

3 Outer Syntax Commands

nonterminal *AOT-prop*
nonterminal φ
nonterminal φ'
nonterminal τ
nonterminal τ'
nonterminal *AOT-axiom*
nonterminal *AOT-act-axiom*
⟨*ML*⟩

4 Approximation of the Syntax of PLM

locale *AOT-meta-syntax*
begin
notation *AOT-model-valid-in* (‹[- ⊨ -]›)
notation *AOT-model-axiom* (‹□[-]›)
notation *AOT-model-act-axiom* (‹𝓐[-]›)
end
locale *AOT-no-meta-syntax*
begin
no-notation *AOT-model-valid-in* (‹[- ⊨ -]›)
no-notation *AOT-model-axiom* (‹□[-]›)
no-notation *AOT-model-act-axiom* (‹𝓐[-]›)
end

consts *AOT-denotes* :: ‹'a::*AOT-Term* ⇒ o›
AOT-imp :: ‹[o, o] ⇒ o›
AOT-not :: ‹o ⇒ o›
AOT-box :: ‹o ⇒ o›
AOT-act :: ‹o ⇒ o›
AOT-forall :: ‹('a::*AOT-Term* ⇒ o) ⇒ o›
AOT-eq :: ‹'a::*AOT-Term* ⇒ 'a::*AOT-Term* ⇒ o›
AOT-desc :: ‹('a::*AOT-UnaryIndividualTerm* ⇒ o) ⇒ 'a›
AOT-exe :: ‹<'a::*AOT-IndividualTerm*> ⇒ 'a ⇒ o›
AOT-lambda :: ‹('a::*AOT-IndividualTerm* ⇒ o) ⇒ <'a>›
AOT-lambda0 :: ‹o ⇒ o›
AOT-concrete :: ‹<'a::*AOT-UnaryIndividualTerm*> *AOT-var*›

nonterminal κ_s **and** Π **and** Π*0* **and** α **and** *exe-arg* **and** *exe-args*
and *lambda-args* **and** *desc* **and** *free-var* **and** *free-vars*
and *AOT-props* **and** *AOT-premises* **and** *AOT-world-relative-prop*

syntax *-AOT-process-frees* :: ‹φ ⇒ φ'› (‹-›)
-AOT-verbatim :: ‹*any* ⇒ φ› (‹«-»›)
-AOT-verbatim :: ‹*any* ⇒ τ› (‹«-»›)
-AOT-quoted :: ‹φ' ⇒ *any*› (‹«-»›)
-AOT-quoted :: ‹τ' ⇒ *any*› (‹«-»›)
:: ‹φ ⇒ φ› (‹'(-')›)
-AOT-process-frees :: ‹τ ⇒ τ'› (‹-›)
:: ‹κ_s ⇒ τ› (‹-›)
:: ‹Π ⇒ τ› (‹-›)
:: ‹φ ⇒ τ› (‹'(-')›)
-AOT-term-var :: ‹*id-position* ⇒ τ› (‹-›)
-AOT-term-var :: ‹*id-position* ⇒ φ› (‹-›)
-AOT-exe-vars :: ‹*id-position* ⇒ *exe-arg*› (‹-›)
-AOT-lambda-vars :: ‹*id-position* ⇒ *lambda-args*› (‹-›)
-AOT-var :: ‹*id-position* ⇒ α› (‹-›)
-AOT-vars :: ‹*id-position* ⇒ *any*›
-AOT-verbatim :: ‹*any* ⇒ α› (‹«-»›)
-AOT-valid :: ‹*w* ⇒ φ' ⇒ *bool*› (‹[- ⊨ -]›)
-AOT-denotes :: ‹τ ⇒ φ› (‹-↓›)
-AOT-imp :: ‹[φ, φ] ⇒ φ› (**infixl** ‹→› *25*)
-AOT-not :: ‹φ ⇒ φ› (‹~-› [*50*] *50*)
-AOT-not :: ‹φ ⇒ φ› (‹¬-› [*50*] *50*)
-AOT-box :: ‹φ ⇒ φ› (‹□-› [*49*] *54*)
-AOT-act :: ‹φ ⇒ φ› (‹𝓐-› [*49*] *54*)
-AOT-all :: ‹α ⇒ φ ⇒ φ› (‹∀ - -› [*1*,*40*])
syntax (*input*)
-AOT-all-ellipse
:: ‹*id-position* ⇒ *id-position* ⇒ φ ⇒ φ› (‹∀ -...∀ - -› [*1*,*40*])
syntax (**output**)
-AOT-all-ellipse
:: ‹*id-position* ⇒ *id-position* ⇒ φ ⇒ φ› (‹∀ -...∀ -'(-')› [*1*,*40*])

syntax
-*AOT-eq* :: ‹[τ, τ] ⇒ φ› (**infixl** ‹=› *50*)
-*AOT-desc* :: ‹α ⇒ φ ⇒ *desc*› (‹ɩ-·-› [*1*,*1000*])
:: ‹*desc* ⇒ κ_s› (‹-›)
-*AOT-lambda* :: ‹*lambda-args* ⇒ φ ⇒ Π› (‹[λ- -]›)
-*explicitRelation* :: ‹τ ⇒ Π› (‹[-]›)
:: ‹κ_s ⇒ *exe-arg*› (‹-›)
:: ‹*exe-arg* ⇒ *exe-args*› (‹-›)
-*AOT-exe-args* :: ‹*exe-arg* ⇒ *exe-args* ⇒ *exe-args*› (‹--›)
-*AOT-exe-arg-ellipse* :: ‹*id-position* ⇒ *id-position* ⇒ *exe-arg*› (‹-...-›)
-*AOT-lambda-arg-ellipse*
:: ‹*id-position* ⇒ *id-position* ⇒ *lambda-args*› (‹-...-›)
-*AOT-term-ellipse* :: ‹*id-position* ⇒ *id-position* ⇒ τ› (‹-...-›)
-*AOT-exe* :: ‹Π ⇒ *exe-args* ⇒ φ› (‹--›)
-*AOT-enc* :: ‹*exe-args* ⇒ Π ⇒ φ› (‹--›)
-*AOT-lambda0* :: ‹φ ⇒ Π*0*› (‹[λ -]›)
:: ‹Π*0* ⇒ φ› (‹-›)
:: ‹Π*0* ⇒ τ› (‹-›)
-*AOT-concrete* :: ‹Π› (‹*E*!›)
:: ‹*any* ⇒ *exe-arg*› (‹«-»›)
:: ‹*desc* ⇒ *free-var*› (‹-›)
:: ‹Π ⇒ *free-var*› (‹-›)
-*AOT-appl* :: ‹*id-position* ⇒ *free-vars* ⇒ φ› (‹-′{-′}›)
-*AOT-appl* :: ‹*id-position* ⇒ *free-vars* ⇒ τ› (‹-′{-′}›)
-*AOT-appl* :: ‹*id-position* ⇒ *free-vars* ⇒ *free-vars*› (‹-′{-′}›)
-*AOT-appl* :: ‹*id-position* ⇒ *free-vars* ⇒ *free-vars*› (‹-′{-′}›)
-*AOT-term-var* :: ‹*id-position* ⇒ *free-var*› (‹-›)
:: ‹*any* ⇒ *free-var*› (‹«-»›)
:: ‹*free-var* ⇒ *free-vars*› (‹-›)
-*AOT-args* :: ‹*free-var* ⇒ *free-vars* ⇒ *free-vars*› (‹-,-›)
-*AOT-free-var-ellipse* :: ‹*id-position* ⇒ *id-position* ⇒ *free-var*› (‹-...-›)
syntax -*AOT-premises*
:: ‹*AOT-world-relative-prop* ⇒ *AOT-premises* ⇒ *AOT-premises*› (**infixr** ‹,› *3*)
-*AOT-world-relative-prop* :: φ ⇒ *AOT-world-relative-prop* (‹-›)
:: *AOT-world-relative-prop* ⇒ *AOT-premises* (‹-›)
-*AOT-prop* :: ‹*AOT-world-relative-prop* ⇒ *AOT-prop*› (‹-›)
:: ‹*AOT-prop* ⇒ *AOT-props*› (‹-›)
-*AOT-derivable* :: *AOT-premises* ⇒ φ′ ⇒ *AOT-prop* (**infixl** ‹⊢› *2*)
-*AOT-nec-derivable* :: *AOT-premises* ⇒ φ′ ⇒ *AOT-prop* (**infixl** ‹$\vdash_\Box$› *2*)
-*AOT-theorem* :: φ′ ⇒ *AOT-prop* (‹⊢ -›)
-*AOT-nec-theorem* :: φ′ ⇒ *AOT-prop* (‹$\vdash_\Box$ -›)
-*AOT-equiv-def* :: ‹φ ⇒ φ ⇒ *AOT-prop*› (**infixl** ‹$\equiv_{df}$› *3*)
-*AOT-axiom* :: φ′ ⇒ *AOT-axiom* (‹-›)
-*AOT-act-axiom* :: φ′ ⇒ *AOT-act-axiom* (‹-›)
-*AOT-axiom* :: φ′ ⇒ *AOT-prop* (‹- ∈ $\Lambda_\Box$›)
-*AOT-act-axiom* :: φ′ ⇒ *AOT-prop* (‹- ∈ Λ›)
-*AOT-id-def* :: ‹τ ⇒ τ ⇒ *AOT-prop*› (**infixl** ‹$=_{df}$› *3*)
-*AOT-for-arbitrary*
:: ‹*id-position* ⇒ *AOT-prop* ⇒ *AOT-prop*› (‹*for arbitrary* -: -› [*1000*,*1*] *1*)
syntax (**output**) -*lambda-args* :: ‹*any* ⇒ *patterns* ⇒ *patterns*› (‹--›)

translations
[*w* ⊨ φ] => *CONST AOT-model-valid-in w* φ

AOT-syntax-print-translations
[*w* ⊨ φ] <= *CONST AOT-model-valid-in w* φ

⟨*ML*⟩

AOT-register-type-constraints
Individual: ‹-::*AOT-UnaryIndividualTerm*› ‹-::*AOT-IndividualTerm*› **and**
Proposition: o **and**
Relation: ‹<-::*AOT-IndividualTerm*>› **and**

Term: ‹-::*AOT-Term*›

AOT-register-variable-names
Individual: $x\ y\ z\ \nu\ \mu\ a\ b\ c\ d$ **and**
Proposition: $p\ q\ r\ s$ **and**
Relation: $F\ G\ H\ P\ Q\ R\ S$ **and**
Term: $\alpha\ \beta\ \gamma\ \delta$

AOT-register-metavariable-names
Individual: κ **and**
Proposition: $\varphi\ \psi\ \chi\ \vartheta\ \zeta\ \xi\ \Theta$ **and**
Relation: Π **and**
Term: $\tau\ \sigma$

AOT-register-premise-set-names $\Gamma\ \Delta\ \Lambda$

⟨*ML*⟩

translations
-AOT-denotes τ => *CONST AOT-denotes* τ
-AOT-imp $\varphi\ \psi$ => *CONST AOT-imp* $\varphi\ \psi$
-AOT-not φ => *CONST AOT-not* φ
-AOT-box φ => *CONST AOT-box* φ
-AOT-act φ => *CONST AOT-act* φ
-AOT-eq $\tau\ \tau'$ => *CONST AOT-eq* $\tau\ \tau'$
-AOT-lambda0 φ => *CONST AOT-lambda0* φ
-AOT-concrete => *CONST AOT-term-of-var* (*CONST AOT-concrete*)
-AOT-lambda $\alpha\ \varphi$ => *CONST AOT-lambda* (*-abs* $\alpha\ \varphi$)
-explicitRelation Π => Π

AOT-syntax-print-translations
-AOT-lambda (*-lambda-args x y*) φ <= *CONST AOT-lambda* (*-abs* (*-pattern x y*) φ)
-AOT-lambda (*-lambda-args x y*) φ <= *CONST AOT-lambda* (*-abs* (*-patterns x y*) φ)
-AOT-lambda x φ <= *CONST AOT-lambda* (*-abs x* φ)
-lambda-args x (*-lambda-args y z*) <= *-lambda-args x* (*-patterns y z*)
-lambda-args (*x y z*) <= *-lambda-args* (*-tuple x* (*-tuple-arg* (*-tuple y z*)))

AOT-syntax-print-translations
-AOT-imp $\varphi\ \psi$ <= *CONST AOT-imp* $\varphi\ \psi$
-AOT-not φ <= *CONST AOT-not* φ
-AOT-box φ <= *CONST AOT-box* φ
-AOT-act φ <= *CONST AOT-act* φ
-AOT-all $\alpha\ \varphi$ <= *CONST AOT-forall* (*-abs* $\alpha\ \varphi$)
-AOT-all $\alpha\ \varphi$ <= *CONST AOT-forall* ($\lambda\alpha.\ \varphi$)
-AOT-eq $\tau\ \tau'$ <= *CONST AOT-eq* $\tau\ \tau'$
-AOT-desc x φ <= *CONST AOT-desc* (*-abs x* φ)
-AOT-desc x φ <= *CONST AOT-desc* ($\lambda x.\ \varphi$)
-AOT-lambda0 φ <= *CONST AOT-lambda0* φ
-AOT-concrete <= *CONST AOT-term-of-var* (*CONST AOT-concrete*)

translations
-AOT-appl φ (*-AOT-args a b*) => *-AOT-appl* (φ *a*) *b*
-AOT-appl φ *a* => φ *a*

⟨*ML*⟩

syntax (**output**)
-AOT-individual-term :: ‹$'a \Rightarrow$ *tuple-args*› (‹-›)
-AOT-individual-terms :: ‹*tuple-args* $\Rightarrow$ *tuple-args* $\Rightarrow$ *tuple-args*› (‹--›)
-AOT-relation-term :: ‹$'a \Rightarrow \Pi$›
-AOT-any-term :: ‹$'a \Rightarrow \tau$›

⟨*ML*⟩

AOT-syntax-print-translations
-AOT-individual-terms (*-AOT-individual-term x*) (*-AOT-individual-terms* (*-tuple y z*))
<= *-AOT-individual-terms* (*-tuple x* (*-tuple-args y z*))
-AOT-individual-terms (*-AOT-individual-term x*) (*-AOT-individual-term y*)
<= *-AOT-individual-terms* (*-tuple x* (*-tuple-arg y*))
-AOT-individual-terms (*-tuple x y*) <= *-AOT-individual-term* (*-tuple x y*)
-AOT-exe (*-AOT-relation-term* Π) (*-AOT-individual-term* κ) <= *CONST AOT-exe* Π κ
-AOT-denotes (*-AOT-any-term* κ) <= *CONST AOT-denotes* κ

AOT-define *AOT-conj* :: ‹[φ, φ] ⇒ φ› (**infixl** ‹&› *35*) ‹φ & ψ $\equiv_{df}$ ¬(φ → ¬ψ)›
declare *AOT-conj*[*AOT del*, *AOT-defs del*]
AOT-define *AOT-disj* :: ‹[φ, φ] ⇒ φ› (**infixl** ‹∨› *35*) ‹φ ∨ ψ $\equiv_{df}$ ¬φ → ψ›
declare *AOT-disj*[*AOT del*, *AOT-defs del*]
AOT-define *AOT-equiv* :: ‹[φ, φ] ⇒ φ› (**infix** ‹≡› *20*) ‹φ ≡ ψ $\equiv_{df}$ (φ → ψ) & (ψ → φ)›
declare *AOT-equiv*[*AOT del*, *AOT-defs del*]
AOT-define *AOT-dia* :: ‹φ ⇒ φ› (‹◇-› [*49*] *54*) ‹◇φ $\equiv_{df}$ ¬□¬φ›
declare *AOT-dia*[*AOT del*, *AOT-defs del*]

context *AOT-meta-syntax*
begin
notation *AOT-dia* (‹◇-› [*49*] *54*)
notation *AOT-conj* (**infixl** ‹**&**› *35*)
notation *AOT-disj* (**infixl** ‹∨› *35*)
notation *AOT-equiv* (**infixl** ‹≡› *20*)
end
context *AOT-no-meta-syntax*
begin
no-notation *AOT-dia* (‹◇-› [*49*] *54*)
no-notation *AOT-conj* (**infixl** ‹**&**› *35*)
no-notation *AOT-disj* (**infixl** ‹∨› *35*)
no-notation *AOT-equiv* (**infixl** ‹≡› *20*)
end

⟨*ML*⟩

AOT-define *AOT-exists* :: ‹α ⇒ φ ⇒ φ› ‹«*AOT-exists* φ» $\equiv_{df}$ ¬∀ α ¬φ{α}›
declare *AOT-exists*[*AOT del*, *AOT-defs del*]
syntax *-AOT-exists* :: ‹α ⇒ φ ⇒ φ› (‹∃ - -› [*1*,*40*])

AOT-syntax-print-translations
-AOT-exists α φ <= *CONST AOT-exists* (*-abs* α φ)
-AOT-exists α φ <= *CONST AOT-exists* (λα. φ)

⟨*ML*⟩

context *AOT-meta-syntax*
begin
notation *AOT-exists* (**binder** ‹∃ › *8*)
end
context *AOT-no-meta-syntax*
begin
no-notation *AOT-exists* (**binder** ‹∃ › *8*)
end

syntax (*input*)
-AOT-exists-ellipse :: ‹*id-position* ⇒ *id-position* ⇒ φ ⇒ φ› (‹∃ -...∃ - -› [*1*,*40*])
syntax (**output**)

-AOT-exists-ellipse :: ‹*id-position* ⇒ *id-position* ⇒ φ ⇒ φ› (‹∃ -...∃ - ′(-′)› [*1*,*40*])
⟨*ML*⟩

syntax -*AOT-DDDOT* :: φ (‹...›)
syntax -*AOT-DDDOT* :: φ (‹. . .›)
⟨*ML*⟩

context *AOT-meta-syntax*
begin
notation *AOT-denotes* (‹-↓›)
notation *AOT-imp* (**infixl** ‹→› *25*)
notation *AOT-not* (‹¬-› [*50*] *50*)
notation *AOT-box* (‹□-› [*49*] *54*)
notation *AOT-act* (‹𝓐-› [*49*] *54*)
notation *AOT-forall* (**binder** ‹∀ › *8*)
notation *AOT-eq* (**infixl** ‹=› *50*)
notation *AOT-desc* (**binder** ‹ι› *100*)
notation *AOT-lambda* (**binder** ‹λ› *100*)
notation *AOT-lambda0* (‹[λ -]›)
notation *AOT-exe* (‹⦇-,-⦈›)
notation *AOT-model-equiv-def* (**infixl** ‹$\equiv_{df}$› *10*)
notation *AOT-model-id-def* (**infixl** ‹$=_{df}$› *10*)
notation *AOT-term-of-var* (‹⟨-⟩›)
notation *AOT-concrete* (‹**E!**›)
end
context *AOT-no-meta-syntax*
begin
no-notation *AOT-denotes* (‹-↓›)
no-notation *AOT-imp* (**infixl** ‹→› *25*)
no-notation *AOT-not* (‹¬-› [*50*] *50*)
no-notation *AOT-box* (‹□-› [*49*] *54*)
no-notation *AOT-act* (‹𝓐-› [*49*] *54*)
no-notation *AOT-forall* (**binder** ‹∀ › *8*)
no-notation *AOT-eq* (**infixl** ‹=› *50*)
no-notation *AOT-desc* (**binder** ‹ι› *100*)
no-notation *AOT-lambda* (**binder** ‹λ› *100*)
no-notation *AOT-lambda0* (‹[λ -]›)
no-notation *AOT-exe* (‹⦇-,-⦈›)
no-notation *AOT-model-equiv-def* (**infixl** ‹$\equiv_{df}$› *10*)
no-notation *AOT-model-id-def* (**infixl** ‹$=_{df}$› *10*)
no-notation *AOT-term-of-var* (‹⟨-⟩›)
no-notation *AOT-concrete* (‹**E!**›)
end

bundle *AOT-syntax*
begin
declare[[*show-AOT-syntax*=*true*, *show-question-marks*=*false*, *eta-contract*=*false*]]
end

bundle *AOT-no-syntax*
begin
declare[[*show-AOT-syntax*=*false*, *show-question-marks*=*true*]]
end

⟨*ML*⟩

Special marker for printing propositions as theorems and for pretty-printing AOT terms.

definition *print-as-theorem* :: ‹o ⇒ *bool*› **where**
‹*print-as-theorem* ≡ λ φ . ∀ *v* . [*v* ⊨ φ]›
lemma *print-as-theoremI*:
assumes ‹⋀ *v* . [*v* ⊨ φ]›

shows ‹*print-as-theorem* φ›
⟨*proof*⟩
⟨*ML*⟩

definition *print-term* :: ‹$'a \Rightarrow 'a$› **where** ‹*print-term* $\equiv \lambda\ x\ .\ x$›
syntax *-AOT-print-term* :: ‹$\tau \Rightarrow 'a$› (‹*AOT'-TERM*[-]›)
translations
-AOT-print-term φ => *CONST print-term* (*-AOT-process-frees* φ)
⟨*ML*⟩

interpretation *AOT-no-meta-syntax*⟨*proof*⟩

unbundle *AOT-syntax*

5 Abstract Semantics for AOT

specification(*AOT-denotes*)
— Relate object level denoting to meta-denoting. AOT's definitions of denoting will become derivable at each type.
AOT-sem-denotes: ‹$[w \models \tau\downarrow] = $ *AOT-model-denotes* τ›
⟨*proof*⟩

lemma *AOT-sem-var-induct*[*induct type*: *AOT-var*]:
assumes *AOT-denoting-term-case*: ‹$\bigwedge\ \tau\ .\ [v \models \tau\downarrow] \Longrightarrow [v \models \varphi\{\tau\}]$›
shows ‹$[v \models \varphi\{\alpha\}]$›
⟨*proof*⟩

specification(*AOT-imp*)
AOT-sem-imp: ‹$[w \models \varphi \rightarrow \psi] = ([w \models \varphi] \longrightarrow [w \models \psi])$›
⟨*proof*⟩

specification(*AOT-not*)
AOT-sem-not: ‹$[w \models \neg\varphi] = (\neg[w \models \varphi])$›
⟨*proof*⟩

specification(*AOT-box*)
AOT-sem-box: ‹$[w \models \Box\varphi] = (\forall\ w\ .\ [w \models \varphi])$›
⟨*proof*⟩

specification(*AOT-act*)
AOT-sem-act: ‹$[w \models \boldsymbol{\mathcal{A}}\varphi] = [w_0 \models \varphi]$›
⟨*proof*⟩

Derived semantics for basic defined connectives.

lemma *AOT-sem-conj*: ‹$[w \models \varphi\ \&\ \psi] = ([w \models \varphi] \wedge [w \models \psi])$›
⟨*proof*⟩
lemma *AOT-sem-equiv*: ‹$[w \models \varphi \equiv \psi] = ([w \models \varphi] = [w \models \psi])$›
⟨*proof*⟩
lemma *AOT-sem-disj*: ‹$[w \models \varphi \vee \psi] = ([w \models \varphi] \vee [w \models \psi])$›
⟨*proof*⟩
lemma *AOT-sem-dia*: ‹$[w \models \Diamond\varphi] = (\exists\ w\ .\ [w \models \varphi])$›
⟨*proof*⟩

specification(*AOT-forall*)

AOT-sem-forall: ‹$[w \models \forall\,\alpha\ \varphi\{\alpha\}] = (\forall\ \tau\ .\ [w \models \tau\downarrow] \longrightarrow [w \models \varphi\{\tau\}])$›
⟨*proof*⟩

lemma *AOT-sem-exists*: ‹$[w \models \exists\,\alpha\ \varphi\{\alpha\}] = (\exists\ \tau\ .\ [w \models \tau\downarrow] \wedge [w \models \varphi\{\tau\}])$›
⟨*proof*⟩

specification(*AOT-eq*)
— Relate identity to denoting identity in the meta-logic. AOT's definitions of identity will become derivable at each type.
AOT-sem-eq: ‹$[w \models \tau = \tau'] = ([w \models \tau\downarrow] \wedge [w \models \tau'\downarrow] \wedge \tau = \tau')$›
⟨*proof*⟩

specification(*AOT-desc*)
— Descriptions denote, if there is a unique denoting object satisfying the matrix in the actual world.
AOT-sem-desc-denotes: ‹$[w \models \iota x(\varphi\{x\})\downarrow] = (\exists!\ \kappa\ .\ [w_0 \models \kappa\downarrow] \wedge [w_0 \models \varphi\{\kappa\}])$›
— Denoting descriptions satisfy their matrix in the actual world.
AOT-sem-desc-prop: ‹$[w \models \iota x(\varphi\{x\})\downarrow] \Longrightarrow [w_0 \models \varphi\{\iota x(\varphi\{x\})\}]$›
— Uniqueness of denoting descriptions.
AOT-sem-desc-unique: ‹$[w \models \iota x(\varphi\{x\})\downarrow] \Longrightarrow [w \models \kappa\downarrow] \Longrightarrow [w_0 \models \varphi\{\kappa\}] \Longrightarrow$
$[w \models \iota x(\varphi\{x\}) = \kappa]$›
⟨*proof*⟩

specification(*AOT-exe AOT-lambda*)
— Truth conditions of exemplification formulas.
AOT-sem-exe: ‹$[w \models [\Pi]\kappa_1...\kappa_n] = ([w \models \Pi\downarrow] \wedge [w \models \kappa_1...\kappa_n\downarrow] \wedge$
$[w \models$ «*Rep-rel* $\Pi\ \kappa_1\kappa_n$»$])$›
— η-conversion for denoting terms; equivalent to AOT's axiom
AOT-sem-lambda-eta: ‹$[w \models \Pi\downarrow] \Longrightarrow [w \models [\lambda\nu_1...\nu_n\ [\Pi]\nu_1...\nu_n] = \Pi]$›
— β-conversion; equivalent to AOT's axiom
AOT-sem-lambda-beta: ‹$[w \models [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\downarrow] \Longrightarrow [w \models \kappa_1...\kappa_n\downarrow] \Longrightarrow$
$[w \models [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\kappa_1...\kappa_n] = [w \models \varphi\{\kappa_1...\kappa_n\}]$›
— Necessary and sufficient conditions for relations to denote. Equivalent to a theorem of AOT and used to derive the base cases of denoting relations (cqt.2).
AOT-sem-lambda-denotes: ‹$[w \models [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\downarrow] =$
$(\forall\ v\ \kappa_1\kappa_n\ \kappa_1{}'\kappa_n{}'\ .\ [v \models \kappa_1...\kappa_n\downarrow] \wedge [v \models \kappa_1{}'...\kappa_n{}'\downarrow] \wedge$
$(\forall\ \Pi\ v\ .\ [v \models \Pi\downarrow] \longrightarrow [v \models [\Pi]\kappa_1...\kappa_n] = [v \models [\Pi]\kappa_1{}'...\kappa_n{}']) \longrightarrow$
$[v \models \varphi\{\kappa_1...\kappa_n\}] = [v \models \varphi\{\kappa_1{}'...\kappa_n{}'\}])$›
— Equivalent to AOT's coexistence axiom.
AOT-sem-lambda-coex: ‹$[w \models [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\downarrow] \Longrightarrow$
$(\forall\ w\ \kappa_1\kappa_n\ .\ [w \models \kappa_1...\kappa_n\downarrow] \longrightarrow [w \models \varphi\{\kappa_1...\kappa_n\}] = [w \models \psi\{\kappa_1...\kappa_n\}]) \Longrightarrow$
$[w \models [\lambda\nu_1...\nu_n\ \psi\{\nu_1...\nu_n\}]\downarrow]$›
— Only the unary case of the following should hold, but our specification has to range over all types. We might move *AOT-exe* and *AOT-lambda* to type classes in the future to solve this.
AOT-sem-lambda-eq-prop-eq: ‹«$[\lambda\nu_1...\nu_n\ \varphi]$» = «$[\lambda\nu_1...\nu_n\ \psi]$» $\Longrightarrow \varphi = \psi$›
— The following is solely required for validating n-ary relation identity and has the danger of implying artifactual theorems. Possibly avoidable by moving *AOT-exe* and *AOT-lambda* to type classes.
AOT-sem-exe-denoting: ‹$[w \models \Pi\downarrow] \Longrightarrow$ *AOT-exe* $\Pi\ \kappa s$ = *Rep-rel* $\Pi\ \kappa s$›
— The following is required for validating the base cases of denoting relations (cqt.2). A version of this meta-logical identity will become derivable in future versions of AOT, so this will ultimately not result in artifactual theorems.
AOT-sem-exe-equiv: ‹*AOT-model-term-equiv* $x\ y \Longrightarrow$ *AOT-exe* $\Pi\ x$ = *AOT-exe* $\Pi\ y$›
⟨*proof*⟩

lemma *AOT-model-lambda-denotes*:
‹*AOT-model-denotes* (*AOT-lambda* φ) = $(\forall\ v\ \kappa\ \kappa'\ .$
AOT-model-denotes $\kappa\ \wedge$ *AOT-model-denotes* $\kappa'\ \wedge$ *AOT-model-term-equiv* $\kappa\ \kappa' \longrightarrow$
$[v \models$ «$\varphi\ \kappa$»$] = [v \models$ «$\varphi\ \kappa'$»$])$›
⟨*proof*⟩

specification (*AOT-lambda0*)
AOT-sem-lambda0: *AOT-lambda0* $\varphi = \varphi$
⟨*proof*⟩

specification(*AOT-concrete*)

AOT-sem-concrete: ‹$[w \models [E!]\kappa] =$
AOT-model-concrete w κ›
⟨*proof*⟩

lemma *AOT-sem-equiv-defI*:
assumes ‹$\bigwedge v . [v \models \varphi] \Longrightarrow [v \models \psi]$›
and ‹$\bigwedge v . [v \models \psi] \Longrightarrow [v \models \varphi]$›
shows ‹*AOT-model-equiv-def* φ ψ›
⟨*proof*⟩

lemma *AOT-sem-id-defI*:
assumes ‹$\bigwedge \alpha\ v . [v \models «\sigma\ \alpha»\downarrow] \Longrightarrow [v \models «\tau\ \alpha» = «\sigma\ \alpha»]$›
assumes ‹$\bigwedge \alpha\ v . \neg[v \models «\sigma\ \alpha»\downarrow] \Longrightarrow [v \models \neg«\tau\ \alpha»\downarrow]$›
shows ‹*AOT-model-id-def* τ σ›
⟨*proof*⟩

lemma *AOT-sem-id-def2I*:
assumes ‹$\bigwedge \alpha\ \beta\ v . [v \models «\sigma\ \alpha\ \beta»\downarrow] \Longrightarrow [v \models «\tau\ \alpha\ \beta» = «\sigma\ \alpha\ \beta»]$›
assumes ‹$\bigwedge \alpha\ \beta\ v . \neg[v \models «\sigma\ \alpha\ \beta»\downarrow] \Longrightarrow [v \models \neg«\tau\ \alpha\ \beta»\downarrow]$›
shows ‹*AOT-model-id-def* (*case-prod* τ) (*case-prod* σ)›
⟨*proof*⟩

lemma *AOT-sem-id-defE1*:
assumes ‹*AOT-model-id-def* τ σ›
and ‹$[v \models «\sigma\ \alpha»\downarrow]$›
shows ‹$[v \models «\tau\ \alpha» = «\sigma\ \alpha»]$›
⟨*proof*⟩

lemma *AOT-sem-id-defE2*:
assumes ‹*AOT-model-id-def* τ σ›
and ‹$\neg[v \models «\sigma\ \alpha»\downarrow]$›
shows ‹$\neg[v \models «\tau\ \alpha»\downarrow]$›
⟨*proof*⟩

lemma *AOT-sem-id-def0I*:
assumes ‹$\bigwedge v . [v \models \sigma\downarrow] \Longrightarrow [v \models \tau = \sigma]$›
and ‹$\bigwedge v . \neg[v \models \sigma\downarrow] \Longrightarrow [v \models \neg\tau\downarrow]$›
shows ‹*AOT-model-id-def* (*case-unit* τ) (*case-unit* σ)›
⟨*proof*⟩

lemma *AOT-sem-id-def0E1*:
assumes ‹*AOT-model-id-def* (*case-unit* τ) (*case-unit* σ)›
and ‹$[v \models \sigma\downarrow]$›
shows ‹$[v \models \tau = \sigma]$›
⟨*proof*⟩

lemma *AOT-sem-id-def0E2*:
assumes ‹*AOT-model-id-def* (*case-unit* τ) (*case-unit* σ)›
and ‹$\neg[v \models \sigma\downarrow]$›
shows ‹$\neg[v \models \tau\downarrow]$›
⟨*proof*⟩

lemma *AOT-sem-id-def0E3*:
assumes ‹*AOT-model-id-def* (*case-unit* τ) (*case-unit* σ)›
and ‹$[v \models \sigma\downarrow]$›
shows ‹$[v \models \tau\downarrow]$›
⟨*proof*⟩

lemma *AOT-sem-ordinary-def-denotes*: ‹$[w \models [\lambda x\ \Diamond[E!]x]\downarrow]$›
⟨*proof*⟩
lemma *AOT-sem-abstract-def-denotes*: ‹$[w \models [\lambda x\ \neg\Diamond[E!]x]\downarrow]$›
⟨*proof*⟩

Relation identity is constructed using an auxiliary abstract projection mechanism with suitable instantiations for κ and products.

class *AOT-RelationProjection* =
  fixes *AOT-sem-proj-id* :: ‹$'a$::*AOT-IndividualTerm* ⇒ ($'a$ ⇒ o) ⇒ ($'a$ ⇒ o) ⇒ o›
  assumes *AOT-sem-proj-id-prop*:
    ‹$[v \models \Pi = \Pi']$ =
     $[v \models \Pi\downarrow$ & $\Pi'\downarrow$ & $\forall\alpha$ («*AOT-sem-proj-id* α ($\lambda\ \tau$. «$[\Pi]\tau$») ($\lambda\ \tau$. «$[\Pi']\tau$»)»)]›
      and *AOT-sem-proj-id-refl*:
    ‹$[v \models \tau\downarrow] \Longrightarrow [v \models [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}] = [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]] \Longrightarrow$
     $[v \models$ «*AOT-sem-proj-id* $\tau\ \varphi\ \varphi$»]›

class *AOT-UnaryRelationProjection* = *AOT-RelationProjection* +
  assumes *AOT-sem-unary-proj-id*:
    ‹*AOT-sem-proj-id* $\kappa\ \varphi\ \psi$ = «$[\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}] = [\lambda\nu_1...\nu_n\ \psi\{\nu_1...\nu_n\}]$»›

instantiation κ :: *AOT-UnaryRelationProjection*
begin
definition *AOT-sem-proj-id-κ* :: ‹$\kappa \Rightarrow (\kappa \Rightarrow o) \Rightarrow (\kappa \Rightarrow o) \Rightarrow o$› **where**
  ‹*AOT-sem-proj-id-κ* $\kappa\ \varphi\ \psi$ = «$[\lambda z\ \varphi\{z\}] = [\lambda z\ \psi\{z\}]$»›
instance ⟨*proof*⟩
end

instantiation *prod* ::
  ({*AOT-UnaryRelationProjection*, *AOT-UnaryIndividualTerm*}, *AOT-RelationProjection*)
  AOT-RelationProjection
begin
definition *AOT-sem-proj-id-prod* :: ‹$'a\times'b \Rightarrow ('a\times'b \Rightarrow o) \Rightarrow ('a\times'b \Rightarrow o) \Rightarrow o$› **where**
  ‹*AOT-sem-proj-id-prod* ≡ λ (x,y) φ ψ . «$[\lambda z$ «φ (z,y)»] = $[\lambda z$ «ψ (z,y)»] &
    «*AOT-sem-proj-id* y (λ a . φ (x,a)) (λ a . ψ (x,a))»»›
instance ⟨*proof*⟩
end

Sanity-check to verify that n-ary relation identity follows.

lemma ‹$[v \models \Pi = \Pi'] = [v \models \Pi\downarrow$ & $\Pi'\downarrow$ & $\forall x \forall y([\lambda z\ [\Pi]z\ y] = [\lambda z\ [\Pi']z\ y]$ &
                                   $[\lambda z\ [\Pi]x\ z] = [\lambda z\ [\Pi']x\ z])]$›
  for Π :: ‹<κ×κ>›
  ⟨*proof*⟩
lemma ‹$[v \models \Pi = \Pi'] = [v \models \Pi\downarrow$ & $\Pi'\downarrow$ & $\forall x_1\forall x_2\forall x_3$ (
  $[\lambda z\ [\Pi]z\ x_2\ x_3] = [\lambda z\ [\Pi']z\ x_2\ x_3]$ &
  $[\lambda z\ [\Pi]x_1\ z\ x_3] = [\lambda z\ [\Pi']x_1\ z\ x_3]$ &
  $[\lambda z\ [\Pi]x_1\ x_2\ z] = [\lambda z\ [\Pi']x_1\ x_2\ z])]$›
  for Π :: ‹<κ×κ×κ>›
  ⟨*proof*⟩
lemma ‹$[v \models \Pi = \Pi'] = [v \models \Pi\downarrow$ & $\Pi'\downarrow$ & $\forall x_1\forall x_2\forall x_3\forall x_4$ (
    $[\lambda z\ [\Pi]z\ x_2\ x_3\ x_4] = [\lambda z\ [\Pi']z\ x_2\ x_3\ x_4]$ &
    $[\lambda z\ [\Pi]x_1\ z\ x_3\ x_4] = [\lambda z\ [\Pi']x_1\ z\ x_3\ x_4]$ &
    $[\lambda z\ [\Pi]x_1\ x_2\ z\ x_4] = [\lambda z\ [\Pi']x_1\ x_2\ z\ x_4]$ &
    $[\lambda z\ [\Pi]x_1\ x_2\ x_3\ z] = [\lambda z\ [\Pi']x_1\ x_2\ x_3\ z])]$›
  for Π :: ‹<κ×κ×κ×κ>›
  ⟨*proof*⟩

n-ary Encoding is constructed using a similar mechanism as n-ary relation identity using an auxiliary notion of projection-encoding.

class *AOT-Enc* =
  fixes *AOT-enc* :: ‹$'a$ ⇒ <$'a$::*AOT-IndividualTerm*> ⇒ o›
    and *AOT-proj-enc* :: ‹$'a$ ⇒ ($'a$ ⇒ o) ⇒ o›
  assumes *AOT-sem-enc-denotes*:
    ‹$[v \models$ «*AOT-enc* $\kappa_1\kappa_n$ Π»] $\Longrightarrow [v \models \kappa_1...\kappa_n\downarrow] \wedge [v \models \Pi\downarrow]$›
  assumes *AOT-sem-enc-proj-enc*:
    ‹$[v \models$ «*AOT-enc* $\kappa_1\kappa_n$ Π»] =
     $[v \models \Pi\downarrow$ & «*AOT-proj-enc* $\kappa_1\kappa_n$ (λ $\kappa_1\kappa_n$. «$[\Pi]\kappa_1...\kappa_n$»)»]›
  assumes *AOT-sem-proj-enc-denotes*:

$\langle[v \models «\textit{AOT-proj-enc}\ \kappa_1\kappa_n\ \varphi»] \Longrightarrow [v \models \kappa_1...\kappa_n{\downarrow}]\rangle$
assumes *AOT-sem-enc-nec*:
$\langle[v \models «\textit{AOT-enc}\ \kappa_1\kappa_n\ \Pi»] \Longrightarrow [w \models «\textit{AOT-enc}\ \kappa_1\kappa_n\ \Pi»]\rangle$
assumes *AOT-sem-proj-enc-nec*:
$\langle[v \models «\textit{AOT-proj-enc}\ \kappa_1\kappa_n\ \varphi»] \Longrightarrow [w \models «\textit{AOT-proj-enc}\ \kappa_1\kappa_n\ \varphi»]\rangle$
assumes *AOT-sem-universal-encoder*:
$\langle\exists\ \kappa_1\kappa_n.\ [v \models \kappa_1...\kappa_n{\downarrow}] \wedge (\forall\ \Pi\ .\ [v \models \Pi{\downarrow}] \longrightarrow [v \models «\textit{AOT-enc}\ \kappa_1\kappa_n\ \Pi»]) \wedge$
$(\forall\ \varphi\ .\ [v \models [\lambda z_1...z_n\ \varphi\{z_1...z_n\}]{\downarrow}] \longrightarrow [v \models «\textit{AOT-proj-enc}\ \kappa_1\kappa_n\ \varphi»])\rangle$

AOT-syntax-print-translations
-*AOT-enc* (-*AOT-individual-term* κ) (-*AOT-relation-term* Π) <= *CONST AOT-enc* $\kappa\ \Pi$

context *AOT-meta-syntax*
begin
notation *AOT-enc* (‹⦃-,-⦄›)
end
context *AOT-no-meta-syntax*
begin
no-notation *AOT-enc* (‹⦃-,-⦄›)
end

Unary encoding additionally has to satisfy the axioms of unary encoding and the definition of property identity.

class *AOT-UnaryEnc* = *AOT-UnaryIndividualTerm* +
assumes *AOT-sem-enc-eq*: $\langle[v \models \Pi{\downarrow}\ \&\ \Pi'{\downarrow}\ \&\ \Box\forall\nu\ (\nu[\Pi] \equiv \nu[\Pi'])\rightarrow \Pi = \Pi']\rangle$
and *AOT-sem-A-objects*: $\langle[v \models \exists x\ (\neg\Diamond[E!]x\ \&\ \forall F\ (x[F] \equiv \varphi\{F\}))]\rangle$
and *AOT-sem-unary-proj-enc*: $\langle\textit{AOT-proj-enc}\ x\ \psi = \textit{AOT-enc}\ x\ «[\lambda z\ \psi\{z\}]»\rangle$
and *AOT-sem-nocoder*: $\langle[v \models [E!]\kappa] \Longrightarrow \neg[w \models «\textit{AOT-enc}\ \kappa\ \Pi»]\rangle$
and *AOT-sem-ind-eq*: $\langle([v \models \kappa{\downarrow}] \wedge [v \models \kappa'{\downarrow}] \wedge \kappa = (\kappa')) =$
$(([v \models [\lambda x\ \Diamond[E!]x]\kappa] \wedge$
$[v \models [\lambda x\ \Diamond[E!]x]\kappa'] \wedge$
$(\forall\ v\ \Pi\ .\ [v \models \Pi{\downarrow}] \longrightarrow [v \models [\Pi]\kappa] = [v \models [\Pi]\kappa']))$
$\vee\ ([v \models [\lambda x\ \neg\Diamond[E!]x]\kappa] \wedge$
$[v \models [\lambda x\ \neg\Diamond[E!]x]\kappa'] \wedge$
$(\forall\ v\ \Pi\ .\ [v \models \Pi{\downarrow}] \longrightarrow [v \models \kappa[\Pi]] = [v \models \kappa'[\Pi]])))\rangle$

and *AOT-sem-enc-indistinguishable-all*:
$\langle\textit{AOT-ExtendedModel} \Longrightarrow$
$[v \models [\lambda x\ \neg\Diamond[E!]x]\kappa] \Longrightarrow$
$[v \models [\lambda x\ \neg\Diamond[E!]x]\kappa'] \Longrightarrow$
$(\bigwedge\ \Pi'\ .\ [v \models \Pi'{\downarrow}] \Longrightarrow (\bigwedge\ w\ .\ [w \models [\Pi']\kappa] = [w \models [\Pi']\kappa'])) \Longrightarrow$
$[v \models \Pi{\downarrow}] \Longrightarrow$
$(\bigwedge\ \Pi'\ .\ [v \models \Pi'{\downarrow}] \Longrightarrow (\bigwedge\ \kappa_0\ .\ [v \models [\lambda x\ \Diamond[E!]x]\kappa_0] \Longrightarrow$
$(\bigwedge\ w\ .\ [w \models [\Pi']\kappa_0] = [w \models [\Pi]\kappa_0])) \Longrightarrow [v \models \kappa[\Pi']]) \Longrightarrow$
$(\bigwedge\ \Pi'\ .\ [v \models \Pi'{\downarrow}] \Longrightarrow (\bigwedge\ \kappa_0\ .\ [v \models [\lambda x\ \Diamond[E!]x]\kappa_0] \Longrightarrow$
$(\bigwedge\ w\ .\ [w \models [\Pi']\kappa_0] = [w \models [\Pi]\kappa_0])) \Longrightarrow [v \models \kappa'[\Pi']])\rangle$
and *AOT-sem-enc-indistinguishable-ex*:
$\langle\textit{AOT-ExtendedModel} \Longrightarrow$
$[v \models [\lambda x\ \neg\Diamond[E!]x]\kappa] \Longrightarrow$
$[v \models [\lambda x\ \neg\Diamond[E!]x]\kappa'] \Longrightarrow$
$(\bigwedge\ \Pi'\ .\ [v \models \Pi'{\downarrow}] \Longrightarrow (\bigwedge\ w\ .\ [w \models [\Pi']\kappa] = [w \models [\Pi']\kappa'])) \Longrightarrow$
$[v \models \Pi{\downarrow}] \Longrightarrow$
$\exists\ \Pi'\ .\ [v \models \Pi'{\downarrow}] \wedge [v \models \kappa[\Pi']] \wedge$
$(\forall\ \kappa_0\ .\ [v \models [\lambda x\ \Diamond[E!]x]\kappa_0] \longrightarrow$
$(\forall\ w\ .\ [w \models [\Pi']\kappa_0] = [w \models [\Pi]\kappa_0])) \Longrightarrow$
$\exists\ \Pi'\ .\ [v \models \Pi'{\downarrow}] \wedge [v \models \kappa'[\Pi']] \wedge$
$(\forall\ \kappa_0\ .\ [v \models [\lambda x\ \Diamond[E!]x]\kappa_0] \longrightarrow$
$(\forall\ w\ .\ [w \models [\Pi']\kappa_0] = [w \models [\Pi]\kappa_0]))\rangle$

We specify encoding to align with the model-construction of encoding.

consts *AOT-sem-enc-κ* :: ‹κ ⇒ <κ> ⇒ o›
specification(*AOT-sem-enc-κ*)

AOT-sem-enc-κ:
‹[$v \models$ «*AOT-sem-enc-κ* κ Π»] =
(*AOT-model-denotes* κ $\wedge$ *AOT-model-denotes* Π $\wedge$ *AOT-model-enc* κ Π)›
⟨*proof*⟩

We show that κ satisfies the generic properties of n-ary encoding.

instantiation κ :: *AOT-Enc*
begin
definition *AOT-enc-κ* :: ‹$\kappa \Rightarrow <\kappa> \Rightarrow$ o› **where**
‹*AOT-enc-κ* $\equiv$ *AOT-sem-enc-κ*›
definition *AOT-proj-enc-κ* :: ‹$\kappa \Rightarrow (\kappa \Rightarrow$ o$) \Rightarrow$ o› **where**
‹*AOT-proj-enc-κ* $\equiv \lambda\ \kappa\ \varphi$. *AOT-enc* κ «[λz «φ z»]»›
lemma *AOT-enc-κ-meta*:
‹[$v \models \kappa[\Pi]$] = (*AOT-model-denotes* κ $\wedge$ *AOT-model-denotes* Π $\wedge$ *AOT-model-enc* κ Π)›
for κ::κ
⟨*proof*⟩
instance ⟨*proof*⟩
end

We show that κ satisfies the properties of unary encoding.

instantiation κ :: *AOT-UnaryEnc*
begin
instance ⟨*proof*⟩
end

Define encoding for products using projection-encoding.

instantiation *prod* :: (*AOT-UnaryEnc*, *AOT-Enc*) *AOT-Enc*
begin
definition *AOT-proj-enc-prod* :: ‹$'a\times'b \Rightarrow ('a\times'b \Rightarrow$ o$) \Rightarrow$ o› **where**
‹*AOT-proj-enc-prod* $\equiv \lambda\ (\kappa,\kappa')\ \varphi$. «κ[$\lambda\nu$ «φ (ν,κ')»] &
«*AOT-proj-enc* κ' ($\lambda\nu$. φ (κ,ν))»»›
definition *AOT-enc-prod* :: ‹$'a\times'b \Rightarrow <'a\times'b> \Rightarrow$ o› **where**
‹*AOT-enc-prod* $\equiv \lambda\ \kappa\ \Pi$. «$\Pi\downarrow$ & «*AOT-proj-enc* κ ($\lambda\ \kappa_1{}'\kappa_n{}'$. «[$\Pi$]$\kappa_1{}'...\kappa_n{}'$»)»»›
instance ⟨*proof*⟩
end

Sanity-check to verify that n-ary encoding follows.

lemma ‹[$v \models \kappa_1\kappa_2[\Pi]$] = [$v \models \Pi\downarrow$ & $\kappa_1[\lambda\nu\ [\Pi]\nu\kappa_2]$ & $\kappa_2[\lambda\nu\ [\Pi]\kappa_1\nu]$]›
for κ_1 :: $'a$::*AOT-UnaryEnc* **and** κ_2 :: $'b$::*AOT-UnaryEnc*
⟨*proof*⟩
lemma ‹[$v \models \kappa_1\kappa_2\kappa_3[\Pi]$] =
[$v \models \Pi\downarrow$ & $\kappa_1[\lambda\nu\ [\Pi]\nu\kappa_2\kappa_3]$ & $\kappa_2[\lambda\nu\ [\Pi]\kappa_1\nu\kappa_3]$ & $\kappa_3[\lambda\nu\ [\Pi]\kappa_1\kappa_2\nu]$]›
for κ_1 κ_2 κ_3 :: $'a$::*AOT-UnaryEnc*
⟨*proof*⟩

lemma *AOT-sem-vars-denote*: ‹[$v \models \alpha_1...\alpha_n\downarrow$]›
⟨*proof*⟩

Combine the introduced type classes and register them as type constraints for individual terms.

class *AOT-κs* = *AOT-IndividualTerm* + *AOT-RelationProjection* + *AOT-Enc*
class *AOT-κ* = *AOT-κs* + *AOT-UnaryIndividualTerm* +
AOT-UnaryRelationProjection + *AOT-UnaryEnc*

instance κ :: *AOT-κ* ⟨*proof*⟩
instance *prod* :: (*AOT-κ*, *AOT-κs*) *AOT-κs* ⟨*proof*⟩

AOT-register-type-constraints
Individual: ‹-::*AOT-κ*› ‹-::*AOT-κs*› **and**
Relation: ‹<-::*AOT-κs*>›

We define semantic predicates to capture the conditions of cqt.2 (i.e. the base cases of denoting terms) on matrices of λ-expressions.

definition *AOT-instance-of-cqt-2* :: ‹('a::AOT-κs ⇒ o) ⇒ bool› **where**
‹AOT-instance-of-cqt-2 ≡ λ φ . ∀ x y . AOT-model-denotes x ∧ AOT-model-denotes y ∧
AOT-model-term-equiv x y ⟶ φ x = φ y›
definition *AOT-instance-of-cqt-2-exe-arg* :: ‹('a::AOT-κs ⇒ 'b::AOT-κs) ⇒ bool› **where**
‹AOT-instance-of-cqt-2-exe-arg ≡ λ φ . ∀ x y .
AOT-model-denotes x ∧ AOT-model-denotes y ∧ AOT-model-term-equiv x y ⟶
AOT-model-term-equiv (φ x) (φ y)›

λ-expressions with a matrix that satisfies our predicate denote.

lemma *AOT-sem-cqt-2*:
assumes ‹AOT-instance-of-cqt-2 φ›
shows ‹[v ⊨ [λν₁...νₙ φ{ν₁...νₙ}]↓]›
⟨*proof*⟩

syntax *AOT-instance-of-cqt-2* :: ‹id-position ⇒ AOT-prop›
(‹INSTANCE'-OF'-CQT'-2'(-')›)

Prove introduction rules for the predicates that match the natural language restrictions of the axiom.

named-theorems *AOT-instance-of-cqt-2-intro*
lemma *AOT-instance-of-cqt-2-intros-const*[*AOT-instance-of-cqt-2-intro*]:
‹AOT-instance-of-cqt-2 (λα. φ)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-not*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ›
shows ‹AOT-instance-of-cqt-2 (λτ. «¬φ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-imp*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ› **and** ‹AOT-instance-of-cqt-2 ψ›
shows ‹AOT-instance-of-cqt-2 (λτ. «φ{τ} → ψ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-box*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ›
shows ‹AOT-instance-of-cqt-2 (λτ. «□φ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-act*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ›
shows ‹AOT-instance-of-cqt-2 (λτ. «𝓐φ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-diamond*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ›
shows ‹AOT-instance-of-cqt-2 (λτ. «◇φ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-conj*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ› **and** ‹AOT-instance-of-cqt-2 ψ›
shows ‹AOT-instance-of-cqt-2 (λτ. «φ{τ} & ψ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-disj*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ› **and** ‹AOT-instance-of-cqt-2 ψ›
shows ‹AOT-instance-of-cqt-2 (λτ. «φ{τ} ∨ ψ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-equib*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹AOT-instance-of-cqt-2 φ› **and** ‹AOT-instance-of-cqt-2 ψ›
shows ‹AOT-instance-of-cqt-2 (λτ. «φ{τ} ≡ ψ{τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-forall*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹⋀ α . AOT-instance-of-cqt-2 (Φ α)›
shows ‹AOT-instance-of-cqt-2 (λτ. «∀ α Φ{α,τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-exists*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹⋀ α . AOT-instance-of-cqt-2 (Φ α)›
shows ‹AOT-instance-of-cqt-2 (λτ. «∃ α Φ{α,τ}»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-exe-arg-self*[*AOT-instance-of-cqt-2-intro*]:

‹*AOT-instance-of-cqt-2-exe-arg* (λx. x)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-exe-arg-const*[*AOT-instance-of-cqt-2-intro*]:
‹*AOT-instance-of-cqt-2-exe-arg* (λx. κ)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-exe-arg-fst*[*AOT-instance-of-cqt-2-intro*]:
‹*AOT-instance-of-cqt-2-exe-arg fst*›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-exe-arg-snd*[*AOT-instance-of-cqt-2-intro*]:
‹*AOT-instance-of-cqt-2-exe-arg snd*›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-exe-arg-Pair*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹*AOT-instance-of-cqt-2-exe-arg* φ› **and** ‹*AOT-instance-of-cqt-2-exe-arg* ψ›
shows ‹*AOT-instance-of-cqt-2-exe-arg* (λτ. *Pair* (φ τ) (ψ τ))›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-desc*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹⋀z :: 'a::*AOT-κ*. *AOT-instance-of-cqt-2* (Φ z)›
shows ‹*AOT-instance-of-cqt-2-exe-arg* (λ κ :: 'b::*AOT-κ* . «ιz(Φ{z,κ})»)›
⟨*proof*⟩

lemma *AOT-instance-of-cqt-2-intros-exe-const*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹*AOT-instance-of-cqt-2-exe-arg* κs›
shows ‹*AOT-instance-of-cqt-2* (λx :: 'b::*AOT-κs*. *AOT-exe* Π (κs x))›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intros-exe-lam*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹⋀ y . *AOT-instance-of-cqt-2* (λx. φ x y)›
and ‹*AOT-instance-of-cqt-2-exe-arg* κs›
shows ‹*AOT-instance-of-cqt-2* ($\lambda\kappa_1\kappa_n$ :: 'b::*AOT-κs*.
«[$\lambda\nu_1...\nu_n$ $\varphi\{\kappa_1...\kappa_n,\nu_1...\nu_n\}$]«κs $\kappa_1\kappa_n$»»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-intro-prod*[*AOT-instance-of-cqt-2-intro*]:
assumes ‹⋀ x . *AOT-instance-of-cqt-2* (φ x)›
and ‹⋀ x . *AOT-instance-of-cqt-2* (λ z . φ z x)›
shows ‹*AOT-instance-of-cqt-2* (λ(x,y) . φ x y)›
⟨*proof*⟩

The following are already derivable semantically, but not yet added to *AOT-instance-of-cqt-2-intro*. They will be added with the next planned extension of axiom cqt:2.

named-theorems *AOT-instance-of-cqt-2-intro-next*
definition *AOT-instance-of-cqt-2-enc-arg* :: ‹('a::*AOT-κs* ⇒ 'b::*AOT-κs*) ⇒ *bool*› **where**
‹*AOT-instance-of-cqt-2-enc-arg* ≡ λ φ . ∀ x y z .
AOT-model-denotes x ∧ *AOT-model-denotes* y ∧ *AOT-model-term-equiv* x y ⟶
AOT-enc (φ x) z = *AOT-enc* (φ y) z›
definition *AOT-instance-of-cqt-2-enc-rel* :: ‹('a::*AOT-κs* ⇒ <'b::*AOT-κs*>) ⇒ *bool*› **where**
‹*AOT-instance-of-cqt-2-enc-rel* ≡ λ φ . ∀ x y z .
AOT-model-denotes x ∧ *AOT-model-denotes* y ∧ *AOT-model-term-equiv* x y ⟶
AOT-enc z (φ x) = *AOT-enc* z (φ y)›
lemma *AOT-instance-of-cqt-2-intros-enc*[*AOT-instance-of-cqt-2-intro-next*]:
assumes ‹*AOT-instance-of-cqt-2-enc-rel* Π› **and** ‹*AOT-instance-of-cqt-2-enc-arg* κs›
shows ‹*AOT-instance-of-cqt-2* (λx . *AOT-enc* (κs x) «[«Π x»]»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-enc-arg-intro-const*[*AOT-instance-of-cqt-2-intro-next*]:
‹*AOT-instance-of-cqt-2-enc-arg* (λx. c)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-enc-arg-intro-desc*[*AOT-instance-of-cqt-2-intro-next*]:
assumes ‹⋀z :: 'a::*AOT-κ*. *AOT-instance-of-cqt-2* (Φ z)›
shows ‹*AOT-instance-of-cqt-2-enc-arg* (λ κ :: 'b::*AOT-κ* . «ιz(Φ{z,κ})»)›
⟨*proof*⟩
lemma *AOT-instance-of-cqt-2-enc-rel-intro*[*AOT-instance-of-cqt-2-intro-next*]:
assumes ‹⋀ κ . *AOT-instance-of-cqt-2* (λκ' :: 'b::*AOT-κs* . φ κ κ')›
assumes ‹⋀ κ' . *AOT-instance-of-cqt-2* (λκ :: 'a::*AOT-κs* . φ κ κ')›
shows ‹*AOT-instance-of-cqt-2-enc-rel* (λκ :: 'a::*AOT-κs*. *AOT-lambda* (λκ'. φ κ κ'))›
⟨*proof*⟩

Further restrict unary individual variables to type κ (rather than class *AOT-κ* only) and define being ordinary and being abstract.

AOT-register-type-constraints
Individual: ‹κ› ‹-::*AOT-κs*›

AOT-define *AOT-ordinary* :: ‹Π› (‹*O*!›) ‹*O*! $=_{df}$ [λ*x* ◇*E*!*x*]›
declare *AOT-ordinary*[*AOT del*, *AOT-defs del*]
AOT-define *AOT-abstract* :: ‹Π› (‹*A*!›) ‹*A*! $=_{df}$ [λ*x* ¬◇*E*!*x*]›
declare *AOT-abstract*[*AOT del*, *AOT-defs del*]

context *AOT-meta-syntax*
begin
notation *AOT-ordinary* (‹**O!**›)
notation *AOT-abstract* (‹**A!**›)
end
context *AOT-no-meta-syntax*
begin
no-notation *AOT-ordinary* (‹**O!**›)
no-notation *AOT-abstract* (‹**A!**›)
end

no-translations
-*AOT-concrete* => *CONST AOT-term-of-var* (*CONST AOT-concrete*)
⟨*ML*⟩

Auxiliary lemmata.

lemma *AOT-sem-ordinary*: «*O*!» = «[λ*x* ◇*E*!*x*]»
⟨*proof*⟩
lemma *AOT-sem-abstract*: «*A*!» = «[λ*x* ¬◇*E*!*x*]»
⟨*proof*⟩
lemma *AOT-sem-ordinary-denotes*: ‹[*w* ⊨ *O*!↓]›
⟨*proof*⟩
lemma *AOT-meta-abstract-denotes*: ‹[*w* ⊨ *A*!↓]›
⟨*proof*⟩
lemma *AOT-model-abstract-ακ*: ‹∃ *a* . κ = ακ *a*› **if** ‹[*v* ⊨ *A*!κ]›
⟨*proof*⟩
lemma *AOT-model-ordinary-ωκ*: ‹∃ *a* . κ = ωκ *a*› **if** ‹[*v* ⊨ *O*!κ]›
⟨*proof*⟩
lemma *AOT-model-ωκ-ordinary*: ‹[*v* ⊨ *O*!«ωκ *x*»]›
⟨*proof*⟩
lemma *AOT-model-ακ-ordinary*: ‹[*v* ⊨ *A*!«ακ *x*»]›
⟨*proof*⟩
AOT-theorem *prod-denotesE*: **assumes** ‹«(κ_1,κ_2)»↓› **shows** ‹κ_1↓ & κ_2↓›
⟨*proof*⟩
declare *prod-denotesE*[*AOT del*]
AOT-theorem *prod-denotesI*: **assumes** ‹κ_1↓ & κ_2↓› **shows** ‹«(κ_1,κ_2)»↓›
⟨*proof*⟩
declare *prod-denotesI*[*AOT del*]

Prepare the derivation of the additional axioms that are validated by our extended models.

locale *AOT-ExtendedModel* =
assumes *AOT-ExtendedModel*: ‹*AOT-ExtendedModel*›
begin
lemma *AOT-sem-indistinguishable-ord-enc-all*:
assumes Π-*den*: ‹[*v* ⊨ Π↓]›
assumes *Ax*: ‹[*v* ⊨ *A*!*x*]›
assumes *Ay*: ‹[*v* ⊨ *A*!*y*]›
assumes *indist*: ‹[*v* ⊨ ∀ *F* □([*F*]*x* ≡ [*F*]*y*)]›
shows
‹[*v* ⊨ ∀ *G*(∀ *z*(*O*!*z* → □([*G*]*z* ≡ [Π]*z*)) → *x*[*G*])] =
[*v* ⊨ ∀ *G*(∀ *z*(*O*!*z* → □([*G*]*z* ≡ [Π]*z*)) → *y*[*G*])]›
⟨*proof*⟩

lemma *AOT-sem-indistinguishable-ord-enc-ex*:
  assumes Π-*den*: ‹[$v \models \Pi\downarrow$]›
  assumes *Ax*: ‹[$v \models A!x$]›
  assumes *Ay*: ‹[$v \models A!y$]›
  assumes *indist*: ‹[$v \models \forall F\ \Box([F]x \equiv [F]y)$]›
  shows ‹[$v \models \exists\, G(\forall\, z\ (O!z \rightarrow \Box([G]z \equiv [\Pi]z))\ \&\ x[G])$] =
      [$v \models \exists\, G(\forall\, z(O!z \rightarrow \Box([G]z \equiv [\Pi]z))\ \&\ y[G])$]›
⟨*proof*⟩
end

⟨*ML*⟩
bundle *AOT-no-atp* **begin declare** *AOT-no-atp*[*no-atp*] **end**

theory *AOT-Definitions*
  imports *AOT-semantics*
begin

6 Definitions of AOT

AOT-theorem *conventions*:*1*: ‹$\varphi\ \&\ \psi \equiv_{df} \neg(\varphi \rightarrow \neg\psi)$›
  ⟨*proof*⟩
AOT-theorem *conventions*:*2*: ‹$\varphi \vee \psi \equiv_{df} \neg\varphi \rightarrow \psi$›
  ⟨*proof*⟩
AOT-theorem *conventions*:*3*: ‹$\varphi \equiv \psi \equiv_{df} (\varphi \rightarrow \psi)\ \&\ (\psi \rightarrow \varphi)$›
  ⟨*proof*⟩
AOT-theorem *conventions*:*4*: ‹$\exists\, \alpha\ \varphi\{\alpha\} \equiv_{df} \neg\forall\, \alpha\ \neg\varphi\{\alpha\}$›
  ⟨*proof*⟩
AOT-theorem *conventions*:*5*: ‹$\Diamond\varphi \equiv_{df} \neg\Box\neg\varphi$›
  ⟨*proof*⟩

declare *conventions*:*1*[*AOT-defs*] *conventions*:*2*[*AOT-defs*]
    conventions:*3*[*AOT-defs*] *conventions*:*4*[*AOT-defs*]
    conventions:*5*[*AOT-defs*]

notepad
begin
  ⟨*proof*⟩
end

AOT-theorem *existence*:*1*: ‹$\kappa\downarrow \equiv_{df} \exists\, F\ [F]\kappa$›
  ⟨*proof*⟩
AOT-theorem *existence*:*2*: ‹$\Pi\downarrow \equiv_{df} \exists\, x_1 ... \exists\, x_n\ x_1 ... x_n[\Pi]$›
  ⟨*proof*⟩
AOT-theorem *existence*:*2*[*1*]: ‹$\Pi\downarrow \equiv_{df} \exists\, x\ x[\Pi]$›
  ⟨*proof*⟩
AOT-theorem *existence*:*2*[*2*]: ‹$\Pi\downarrow \equiv_{df} \exists\, x \exists\, y\ xy[\Pi]$›
  ⟨*proof*⟩
AOT-theorem *existence*:*2*[*3*]: ‹$\Pi\downarrow \equiv_{df} \exists\, x \exists\, y \exists\, z\ xyz[\Pi]$›
  ⟨*proof*⟩
AOT-theorem *existence*:*2*[*4*]: ‹$\Pi\downarrow \equiv_{df} \exists\, x_1 \exists\, x_2 \exists\, x_3 \exists\, x_4\ x_1 x_2 x_3 x_4[\Pi]$›
  ⟨*proof*⟩

AOT-theorem *existence*:*3*: ‹$\varphi\downarrow \equiv_{df} [\lambda x\ \varphi]\downarrow$›
  ⟨*proof*⟩

declare *existence*:*1*[*AOT-defs*] *existence*:*2*[*AOT-defs*] *existence*:*2*[*1*][*AOT-defs*]
    existence:*2*[*2*][*AOT-defs*] *existence*:*2*[*3*][*AOT-defs*]
    existence:*2*[*4*][*AOT-defs*] *existence*:*3*[*AOT-defs*]

AOT-theorem *oa:1*: ‹$O! =_{df} [\lambda x\ \Diamond E!x]$› ⟨*proof*⟩
AOT-theorem *oa:2*: ‹$A! =_{df} [\lambda x\ \neg\Diamond E!x]$› ⟨*proof*⟩

declare *oa:1[AOT-defs] oa:2[AOT-defs]*

AOT-theorem *identity:1*:
‹$x = y \equiv_{df} ([O!]x\ \&\ [O!]y\ \&\ \Box\forall F\ ([F]x \equiv [F]y)) \vee$
$([A!]x\ \&\ [A!]y\ \&\ \Box\forall F\ (x[F] \equiv y[F]))$›
⟨*proof*⟩

AOT-theorem *identity:2*:
‹$F = G \equiv_{df} F\downarrow\ \&\ G\downarrow\ \&\ \Box\forall x(x[F] \equiv x[G])$›
⟨*proof*⟩

AOT-theorem *identity:3[2]*:
‹$F = G \equiv_{df} F\downarrow\ \&\ G\downarrow\ \&\ \forall y([\lambda z\ [F]zy] = [\lambda z\ [G]zy]\ \&\ [\lambda z\ [F]yz] = [\lambda z\ [G]yz])$›
⟨*proof*⟩
AOT-theorem *identity:3[3]*:
‹$F = G \equiv_{df} F\downarrow\ \&\ G\downarrow\ \&\ \forall y_1\forall y_2([\lambda z\ [F]zy_1y_2] = [\lambda z\ [G]zy_1y_2]\ \&$
$[\lambda z\ [F]y_1zy_2] = [\lambda z\ [G]y_1zy_2]\ \&$
$[\lambda z\ [F]y_1y_2z] = [\lambda z\ [G]y_1y_2z])$›
⟨*proof*⟩
AOT-theorem *identity:3[4]*:
‹$F = G \equiv_{df} F\downarrow\ \&\ G\downarrow\ \&\ \forall y_1\forall y_2\forall y_3([\lambda z\ [F]zy_1y_2y_3] = [\lambda z\ [G]zy_1y_2y_3]\ \&$
$[\lambda z\ [F]y_1zy_2y_3] = [\lambda z\ [G]y_1zy_2y_3]\ \&$
$[\lambda z\ [F]y_1y_2zy_3] = [\lambda z\ [G]y_1y_2zy_3]\ \&$
$[\lambda z\ [F]y_1y_2y_3z] = [\lambda z\ [G]y_1y_2y_3z])$›
⟨*proof*⟩
AOT-theorem *identity:3*:
‹$F = G \equiv_{df} F\downarrow\ \&\ G\downarrow\ \&\ \forall x_1...\forall x_n$ «*AOT-sem-proj-id* x_1x_n $(\lambda\ \tau\ .\ AOT\text{-}exe\ F\ \tau)$
$(\lambda\ \tau\ .\ AOT\text{-}exe\ G\ \tau)$»›
⟨*proof*⟩

AOT-theorem *identity:4*:
‹$p = q \equiv_{df} p\downarrow\ \&\ q\downarrow\ \&\ [\lambda x\ p] = [\lambda x\ q]$›
⟨*proof*⟩

declare *identity:1[AOT-defs] identity:2[AOT-defs] identity:3[2][AOT-defs]*
identity:3[3][AOT-defs] identity:3[4][AOT-defs] identity:3[AOT-defs]
identity:4[AOT-defs]

AOT-define *AOT-nonidentical* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (**infixl** ‹≠› *50*)
=−infix: ‹$\tau \neq \sigma \equiv_{df} \neg(\tau = \sigma)$›

context *AOT-meta-syntax*
begin
notation *AOT-nonidentical* (**infixl** ‹≠› *50*)
end
context *AOT-no-meta-syntax*
begin
no-notation *AOT-nonidentical* (**infixl** ‹≠› *50*)
end

The following are purely technical pseudo-definitions required due to our internal implementation of n-ary relations and ellipses using tuples.

AOT-theorem *tuple-denotes*: ‹«(τ,τ')»$\downarrow \equiv_{df} \tau\downarrow\ \&\ \tau'\downarrow$›
⟨*proof*⟩
AOT-theorem *tuple-identity-1*: ‹«(τ,τ')» = «(σ, σ')» $\equiv_{df} (\tau = \sigma)\ \&\ (\tau' = \sigma')$›
⟨*proof*⟩
AOT-theorem *tuple-forall*: ‹$\forall \alpha_1...\forall \alpha_n\ \varphi\{\alpha_1...\alpha_n\} \equiv_{df} \forall \alpha_1(\forall \alpha_2...\forall \alpha_n\ \varphi\{$«$(\alpha_1,\ \alpha_2\alpha_n)$»$\})$›
⟨*proof*⟩

AOT-theorem *tuple-exists*: ‹$\exists\,\alpha_1 ... \exists\,\alpha_n\ \varphi\{\alpha_1 ... \alpha_n\} \equiv_{df} \exists\,\alpha_1(\exists\,\alpha_2 ... \exists\,\alpha_n\ \varphi\{«(\alpha_1,\ \alpha_2\alpha_n)»\})$›
⟨*proof*⟩
declare *tuple-denotes*[*AOT-defs*] *tuple-identity-1*[*AOT-defs*] *tuple-forall*[*AOT-defs*] *tuple-exists*[*AOT-defs*]

end

7 Axioms of PLM

AOT-axiom *pl:1*: ‹$\varphi \rightarrow (\psi \rightarrow \varphi)$›
⟨*proof*⟩
AOT-axiom *pl:2*: ‹$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$›
⟨*proof*⟩
AOT-axiom *pl:3*: ‹$(\neg\varphi \rightarrow \neg\psi) \rightarrow ((\neg\varphi \rightarrow \psi) \rightarrow \varphi)$›
⟨*proof*⟩

AOT-axiom *cqt:1*: ‹$\forall\,\alpha\ \varphi\{\alpha\} \rightarrow (\tau\downarrow \rightarrow \varphi\{\tau\})$›
⟨*proof*⟩

AOT-axiom *cqt:2*[*const-var*]: ‹$\alpha\downarrow$›
⟨*proof*⟩
AOT-axiom *cqt:2*[*lambda*]:
assumes ‹*INSTANCE-OF-CQT-2*(φ)›
shows ‹$[\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\downarrow$›
⟨*proof*⟩
AOT-axiom *cqt:2*[*lambda0*]:
shows ‹$[\lambda\ \varphi]\downarrow$›
⟨*proof*⟩

AOT-axiom *cqt:3*: ‹$\forall\,\alpha\ (\varphi\{\alpha\} \rightarrow \psi\{\alpha\}) \rightarrow (\forall\,\alpha\ \varphi\{\alpha\} \rightarrow \forall\,\alpha\ \psi\{\alpha\})$›
⟨*proof*⟩
AOT-axiom *cqt:4*: ‹$\varphi \rightarrow \forall\,\alpha\ \varphi$›
⟨*proof*⟩
AOT-axiom *cqt:5:a*: ‹$[\Pi]\kappa_1...\kappa_n \rightarrow (\Pi\downarrow\ \&\ \kappa_1...\kappa_n\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:a*[*1*]: ‹$[\Pi]\kappa \rightarrow (\Pi\downarrow\ \&\ \kappa\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:a*[*2*]: ‹$[\Pi]\kappa_1\kappa_2 \rightarrow (\Pi\downarrow\ \&\ \kappa_1\downarrow\ \&\ \kappa_2\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:a*[*3*]: ‹$[\Pi]\kappa_1\kappa_2\kappa_3 \rightarrow (\Pi\downarrow\ \&\ \kappa_1\downarrow\ \&\ \kappa_2\downarrow\ \&\ \kappa_3\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:a*[*4*]: ‹$[\Pi]\kappa_1\kappa_2\kappa_3\kappa_4 \rightarrow (\Pi\downarrow\ \&\ \kappa_1\downarrow\ \&\ \kappa_2\downarrow\ \&\ \kappa_3\downarrow\ \&\ \kappa_4\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:b*: ‹$\kappa_1...\kappa_n[\Pi] \rightarrow (\Pi\downarrow\ \&\ \kappa_1...\kappa_n\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:b*[*1*]: ‹$\kappa[\Pi] \rightarrow (\Pi\downarrow\ \&\ \kappa\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:b*[*2*]: ‹$\kappa_1\kappa_2[\Pi] \rightarrow (\Pi\downarrow\ \&\ \kappa_1\downarrow\ \&\ \kappa_2\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:b*[*3*]: ‹$\kappa_1\kappa_2\kappa_3[\Pi] \rightarrow (\Pi\downarrow\ \&\ \kappa_1\downarrow\ \&\ \kappa_2\downarrow\ \&\ \kappa_3\downarrow)$›
⟨*proof*⟩
AOT-axiom *cqt:5:b*[*4*]: ‹$\kappa_1\kappa_2\kappa_3\kappa_4[\Pi] \rightarrow (\Pi\downarrow\ \&\ \kappa_1\downarrow\ \&\ \kappa_2\downarrow\ \&\ \kappa_3\downarrow\ \&\ \kappa_4\downarrow)$›
⟨*proof*⟩

AOT-axiom *l−identity*: ‹$\alpha = \beta \rightarrow (\varphi\{\alpha\} \rightarrow \varphi\{\beta\})$›
⟨*proof*⟩

AOT-act-axiom *logic−actual*: ‹$\boldsymbol{\mathcal{A}}\varphi \rightarrow \varphi$›
⟨*proof*⟩

AOT-axiom *logic−actual−nec:1*: ‹$\boldsymbol{\mathcal{A}}\neg\varphi \equiv \neg\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩
AOT-axiom *logic−actual−nec:2*: ‹$\boldsymbol{\mathcal{A}}(\varphi \rightarrow \psi) \equiv (\boldsymbol{\mathcal{A}}\varphi \rightarrow \boldsymbol{\mathcal{A}}\psi)$›

⟨*proof*⟩

AOT-axiom *logic−actual−nec:3*: ‹$\boldsymbol{\mathcal{A}}(\forall\,\alpha\ \varphi\{\alpha\}) \equiv \forall\,\alpha\ \boldsymbol{\mathcal{A}}\varphi\{\alpha\}$›
⟨*proof*⟩
AOT-axiom *logic−actual−nec:4*: ‹$\boldsymbol{\mathcal{A}}\varphi \equiv \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩

AOT-axiom *qml:1*: ‹$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$›
⟨*proof*⟩
AOT-axiom *qml:2*: ‹$\Box\varphi \to \varphi$›
⟨*proof*⟩
AOT-axiom *qml:3*: ‹$\Diamond\varphi \to \Box\Diamond\varphi$›
⟨*proof*⟩

AOT-axiom *qml:4*: ‹$\Diamond\exists\,x\ (E!x\ \&\ \neg\boldsymbol{\mathcal{A}}E!x)$›
⟨*proof*⟩

AOT-axiom *qml−act:1*: ‹$\boldsymbol{\mathcal{A}}\varphi \to \Box\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩
AOT-axiom *qml−act:2*: ‹$\Box\varphi \equiv \boldsymbol{\mathcal{A}}\Box\varphi$›
⟨*proof*⟩

AOT-axiom *descriptions*: ‹$x = \iota x(\varphi\{x\}) \equiv \forall\,z(\boldsymbol{\mathcal{A}}\varphi\{z\} \equiv z = x)$›
⟨*proof*⟩

AOT-axiom *lambda−predicates:1*:
‹$[\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\!\downarrow\ \to [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}] = [\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]$›
⟨*proof*⟩
AOT-axiom *lambda−predicates:1[zero]*: ‹$[\lambda\ p]\!\downarrow\ \to [\lambda\ p] = [\lambda\ p]$›
⟨*proof*⟩
AOT-axiom *lambda−predicates:2*:
‹$[\lambda x_1...x_n\ \varphi\{x_1...x_n\}]\!\downarrow\ \to ([\lambda x_1...x_n\ \varphi\{x_1...x_n\}]x_1...x_n \equiv \varphi\{x_1...x_n\})$›
⟨*proof*⟩
AOT-axiom *lambda−predicates:3*: ‹$[\lambda x_1...x_n\ [F]x_1...x_n] = F$›
⟨*proof*⟩
AOT-axiom *lambda−predicates:3[zero]*: ‹$[\lambda\ p] = p$›
⟨*proof*⟩

AOT-axiom *safe−ext*:
‹$([\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\!\downarrow\ \&\ \Box\forall\,\nu_1...\forall\,\nu_n\ (\varphi\{\nu_1...\nu_n\} \equiv \psi\{\nu_1...\nu_n\})) \to$
$[\lambda\nu_1...\nu_n\ \psi\{\nu_1...\nu_n\}]\!\downarrow$›
⟨*proof*⟩
AOT-axiom *safe−ext[2]*:
‹$([\lambda\nu_1\nu_2\ \varphi\{\nu_1,\nu_2\}]\!\downarrow\ \&\ \Box\forall\,\nu_1\forall\,\nu_2\ (\varphi\{\nu_1,\ \nu_2\} \equiv \psi\{\nu_1,\ \nu_2\})) \to$
$[\lambda\nu_1\nu_2\ \psi\{\nu_1,\nu_2\}]\!\downarrow$›
⟨*proof*⟩
AOT-axiom *safe−ext[3]*:
‹$([\lambda\nu_1\nu_2\nu_3\ \varphi\{\nu_1,\nu_2,\nu_3\}]\!\downarrow\ \&\ \Box\forall\,\nu_1\forall\,\nu_2\forall\,\nu_3\ (\varphi\{\nu_1,\ \nu_2,\ \nu_3\} \equiv \psi\{\nu_1,\ \nu_2,\ \nu_3\})) \to$
$[\lambda\nu_1\nu_2\nu_3\ \psi\{\nu_1,\nu_2,\nu_3\}]\!\downarrow$›
⟨*proof*⟩
AOT-axiom *safe−ext[4]*:
‹$([\lambda\nu_1\nu_2\nu_3\nu_4\ \varphi\{\nu_1,\nu_2,\nu_3,\nu_4\}]\!\downarrow\ \&$
$\Box\forall\,\nu_1\forall\,\nu_2\forall\,\nu_3\forall\,\nu_4\ (\varphi\{\nu_1,\ \nu_2,\ \nu_3,\ \nu_4\} \equiv \psi\{\nu_1,\ \nu_2,\ \nu_3,\ \nu_4\})) \to$
$[\lambda\nu_1\nu_2\nu_3\nu_4\ \psi\{\nu_1,\nu_2,\nu_3,\nu_4\}]\!\downarrow$›
⟨*proof*⟩

AOT-axiom *nary−encoding[2]*:
‹$x_1x_2[F] \equiv x_1[\lambda y\ [F]yx_2]\ \&\ x_2[\lambda y\ [F]x_1y]$›
⟨*proof*⟩
AOT-axiom *nary−encoding[3]*:
‹$x_1x_2x_3[F] \equiv x_1[\lambda y\ [F]yx_2x_3]\ \&\ x_2[\lambda y\ [F]x_1yx_3]\ \&\ x_3[\lambda y\ [F]x_1x_2y]$›
⟨*proof*⟩
AOT-axiom *nary−encoding[4]*:

‹$x_1x_2x_3x_4[F] \equiv x_1[\lambda y\ [F]yx_2x_3x_4]$ &
$x_2[\lambda y\ [F]x_1yx_3x_4]$ &
$x_3[\lambda y\ [F]x_1x_2yx_4]$ &
$x_4[\lambda y\ [F]x_1x_2x_3y]$›
⟨*proof*⟩

AOT-axiom *encoding*: ‹$x[F] \rightarrow \Box x[F]$›
⟨*proof*⟩

AOT-axiom *nocoder*: ‹$O!x \rightarrow \neg\exists F\ x[F]$›
⟨*proof*⟩

AOT-axiom *A−objects*: ‹$\exists x\ (A!x\ \&\ \forall F(x[F] \equiv \varphi\{F\}))$›
⟨*proof*⟩

AOT-theorem *universal-closure*:
assumes ‹*for arbitrary* α: $\varphi\{\alpha\} \in \Lambda_\Box$›
shows ‹$\forall \alpha\ \varphi\{\alpha\} \in \Lambda_\Box$›
⟨*proof*⟩

AOT-theorem *act-closure*:
assumes ‹$\varphi \in \Lambda_\Box$›
shows ‹$\boldsymbol{\mathcal{A}}\varphi \in \Lambda_\Box$›
⟨*proof*⟩

AOT-theorem *nec-closure*:
assumes ‹$\varphi \in \Lambda_\Box$›
shows ‹$\Box\varphi \in \Lambda_\Box$›
⟨*proof*⟩

AOT-theorem *universal-closure-act*:
assumes ‹*for arbitrary* α: $\varphi\{\alpha\} \in \Lambda$›
shows ‹$\forall \alpha\ \varphi\{\alpha\} \in \Lambda$›
⟨*proof*⟩

The following are not part of PLM and only hold in the extended models. They are a generalization of the predecessor axiom.

context *AOT-ExtendedModel*
begin
AOT-axiom *indistinguishable-ord-enc-all*:
‹$\Pi\downarrow\ \&\ A!x\ \&\ A!y\ \&\ \forall F\ \Box([F]x \equiv [F]y) \rightarrow$
$((\forall G(\forall z(O!z \rightarrow \Box([G]z \equiv [\Pi]z)) \rightarrow x[G])) \equiv$
$\forall G(\forall z(O!z \rightarrow \Box([G]z \equiv [\Pi]z)) \rightarrow y[G]))$›
⟨*proof*⟩
AOT-axiom *indistinguishable-ord-enc-ex*:
‹$\Pi\downarrow\ \&\ A!x\ \&\ A!y\ \&\ \forall F\ \Box([F]x \equiv [F]y) \rightarrow$
$((\exists G(\forall z(O!z \rightarrow \Box([G]z \equiv [\Pi]z))\ \&\ x[G])) \equiv$
$\exists G(\forall z(O!z \rightarrow \Box([G]z \equiv [\Pi]z))\ \&\ y[G]))$›
⟨*proof*⟩
end

8 The Deductive System PLM

unbundle *AOT-no-atp*

8.1 Primitive Rule of PLM: Modus Ponens

AOT-theorem *modus−ponens*:
assumes ‹φ› **and** ‹$\varphi \rightarrow \psi$›
shows ‹ψ›

⟨*proof*⟩
lemmas *MP* = *modus−ponens*

8.2 (Modally Strict) Proofs and Derivations

AOT-theorem *non−con−thm−thm*:
assumes ‹$\vdash_\Box \varphi$›
shows ‹$\vdash \varphi$›
⟨*proof*⟩

AOT-theorem *vdash−properties*:*1*[*1*]:
assumes ‹$\varphi \in \Lambda$›
shows ‹$\vdash \varphi$›

⟨*proof*⟩

Convenience attribute for instantiating modally-fragile axioms.

⟨*ML*⟩

AOT-theorem *vdash−properties*:*1*[*2*]:
assumes ‹$\varphi \in \Lambda_\Box$›
shows ‹$\vdash_\Box \varphi$›

⟨*proof*⟩

Convenience attribute for instantiating modally-strict axioms.

⟨*ML*⟩

Convenience methods and theorem sets for applying "cqt:2".

method *cqt-2-lambda-inst-prover* =
(*fast intro*: *AOT-instance-of-cqt-2-intro*)
method *cqt*:*2*[*lambda*] =
(*rule cqt*:*2*[*lambda*][*axiom-inst*]; *cqt-2-lambda-inst-prover*)
lemmas *cqt*:*2* =
cqt:*2*[*const-var*][*axiom-inst*] *cqt*:*2*[*lambda*][*axiom-inst*]
AOT-instance-of-cqt-2-intro
method *cqt*:*2* = (*safe intro*!: *cqt*:*2*)

AOT-theorem *vdash−properties*:*3*:
assumes ‹$\vdash_\Box \varphi$›
shows ‹$\Gamma \vdash \varphi$›
⟨*proof*⟩

AOT-theorem *vdash−properties*:*5*:
assumes ‹$\Gamma_1 \vdash \varphi$› **and** ‹$\Gamma_2 \vdash \varphi \to \psi$›
shows ‹$\Gamma_1, \Gamma_2 \vdash \psi$›
⟨*proof*⟩

AOT-theorem *vdash−properties*:*6*:
assumes ‹φ› **and** ‹$\varphi \to \psi$›
shows ‹ψ›
⟨*proof*⟩

AOT-theorem *vdash−properties*:*8*:
assumes ‹$\Gamma \vdash \varphi$› **and** ‹$\varphi \vdash \psi$›
shows ‹$\Gamma \vdash \psi$›
⟨*proof*⟩

AOT-theorem *vdash−properties*:*9*:
assumes ‹φ›
shows ‹$\psi \to \varphi$›
⟨*proof*⟩

AOT-theorem *vdash−properties:10*:
assumes ‹$\varphi \to \psi$› **and** ‹φ›
shows ‹ψ›
⟨*proof*⟩
lemmas $\to E$ = *vdash−properties:10*

8.3 Two Fundamental Metarules: GEN and RN

AOT-theorem *rule−gen*:
assumes ‹*for arbitrary* α: $\varphi\{\alpha\}$›
shows ‹$\forall\,\alpha\ \varphi\{\alpha\}$›

⟨*proof*⟩
lemmas *GEN* = *rule−gen*

AOT-theorem *RN*[*prem*]:
assumes ‹$\Gamma \vdash_\Box \varphi$›
shows ‹$\Box\Gamma \vdash_\Box \Box\varphi$›
⟨*proof*⟩
AOT-theorem *RN*:
assumes ‹$\vdash_\Box \varphi$›
shows ‹$\Box\varphi$›
⟨*proof*⟩

8.4 The Inferential Role of Definitions

AOT-axiom *df−rules−formulas*[*1*]:
assumes ‹$\varphi \equiv_{df} \psi$›
shows ‹$\varphi \to \psi$›

⟨*proof*⟩
AOT-axiom *df−rules−formulas*[*2*]:
assumes ‹$\varphi \equiv_{df} \psi$›
shows ‹$\psi \to \varphi$›

⟨*proof*⟩

AOT-theorem *df−rules−formulas*[*3*]:
assumes ‹$\varphi \equiv_{df} \psi$›
shows ‹$\varphi \to \psi$›
⟨*proof*⟩
AOT-theorem *df−rules−formulas*[*4*]:
assumes ‹$\varphi \equiv_{df} \psi$›
shows ‹$\psi \to \varphi$›
⟨*proof*⟩

AOT-axiom *df−rules−terms*[*1*]:
assumes ‹$\tau\{\alpha_1...\alpha_n\} =_{df} \sigma\{\alpha_1...\alpha_n\}$›
shows ‹$(\sigma\{\tau_1...\tau_n\}\downarrow \to \tau\{\tau_1...\tau_n\} = \sigma\{\tau_1...\tau_n\})$ &
$(\neg\sigma\{\tau_1...\tau_n\}\downarrow \to \neg\tau\{\tau_1...\tau_n\}\downarrow)$›

⟨*proof*⟩
AOT-axiom *df−rules−terms*[*2*]:
assumes ‹$\tau =_{df} \sigma$›
shows ‹$(\sigma\downarrow \to \tau = \sigma)$ & $(\neg\sigma\downarrow \to \neg\tau\downarrow)$›
⟨*proof*⟩

AOT-theorem *df−rules−terms*[*3*]:
assumes ‹$\tau\{\alpha_1...\alpha_n\} =_{df} \sigma\{\alpha_1...\alpha_n\}$›
shows ‹$(\sigma\{\tau_1...\tau_n\}\downarrow \to \tau\{\tau_1...\tau_n\} = \sigma\{\tau_1...\tau_n\})$ &
$(\neg\sigma\{\tau_1...\tau_n\}\downarrow \to \neg\tau\{\tau_1...\tau_n\}\downarrow)$›

⟨*proof*⟩
AOT-theorem *df−rules−terms*[*4*]:
assumes ‹$\tau =_{df} \sigma$›
shows ‹$(\sigma\downarrow \rightarrow \tau = \sigma)$ & $(\neg\sigma\downarrow \rightarrow \neg\tau\downarrow)$›
⟨*proof*⟩

8.5 The Theory of Negations and Conditionals

AOT-theorem *if−p−then−p*: ‹$\varphi \rightarrow \varphi$›
⟨*proof*⟩

AOT-theorem *deduction−theorem*:
assumes ‹$\varphi \vdash \psi$›
shows ‹$\varphi \rightarrow \psi$›

⟨*proof*⟩
lemmas *CP* = *deduction−theorem*
lemmas →*I* = *deduction−theorem*

AOT-theorem *ded−thm−cor*:*1*:
assumes ‹$\Gamma_1 \vdash \varphi \rightarrow \psi$› **and** ‹$\Gamma_2 \vdash \psi \rightarrow \chi$›
shows ‹$\Gamma_1, \Gamma_2 \vdash \varphi \rightarrow \chi$›
⟨*proof*⟩
AOT-theorem *ded−thm−cor*:*2*:
assumes ‹$\Gamma_1 \vdash \varphi \rightarrow (\psi \rightarrow \chi)$› **and** ‹$\Gamma_2 \vdash \psi$›
shows ‹$\Gamma_1, \Gamma_2 \vdash \varphi \rightarrow \chi$›
⟨*proof*⟩

AOT-theorem *ded−thm−cor*:*3*:
assumes ‹$\varphi \rightarrow \psi$› **and** ‹$\psi \rightarrow \chi$›
shows ‹$\varphi \rightarrow \chi$›
⟨*proof*⟩
declare *ded−thm−cor*:*3*[*trans*]
AOT-theorem *ded−thm−cor*:*4*:
assumes ‹$\varphi \rightarrow (\psi \rightarrow \chi)$› **and** ‹$\psi$›
shows ‹$\varphi \rightarrow \chi$›
⟨*proof*⟩

lemmas *Hypothetical Syllogism* = *ded−thm−cor*:*3*

AOT-theorem *useful−tautologies*:*1*: ‹$\neg\neg\varphi \rightarrow \varphi$›
⟨*proof*⟩
AOT-theorem *useful−tautologies*:*2*: ‹$\varphi \rightarrow \neg\neg\varphi$›
⟨*proof*⟩
AOT-theorem *useful−tautologies*:*3*: ‹$\neg\varphi \rightarrow (\varphi \rightarrow \psi)$›
⟨*proof*⟩
AOT-theorem *useful−tautologies*:*4*: ‹$(\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$›
⟨*proof*⟩
AOT-theorem *useful−tautologies*:*5*: ‹$(\varphi \rightarrow \psi) \rightarrow (\neg\psi \rightarrow \neg\varphi)$›
⟨*proof*⟩

AOT-theorem *useful−tautologies*:*6*: ‹$(\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \neg\varphi)$›
⟨*proof*⟩

AOT-theorem *useful−tautologies*:*7*: ‹$(\neg\varphi \rightarrow \psi) \rightarrow (\neg\psi \rightarrow \varphi)$›
⟨*proof*⟩

AOT-theorem *useful−tautologies*:*8*: ‹$\varphi \rightarrow (\neg\psi \rightarrow \neg(\varphi \rightarrow \psi))$›
⟨*proof*⟩

AOT-theorem *useful−tautologies*:*9*: ‹$(\varphi \rightarrow \psi) \rightarrow ((\neg\varphi \rightarrow \psi) \rightarrow \psi)$›
⟨*proof*⟩

AOT-theorem *useful−tautologies*:*10*: ‹$(\varphi \rightarrow \neg\psi) \rightarrow ((\varphi \rightarrow \psi) \rightarrow \neg\varphi)$›
⟨*proof*⟩

AOT-theorem *dn−i−e*:*1*:
assumes ‹φ›
shows ‹$\neg\neg\varphi$›
⟨*proof*⟩
lemmas $\neg\neg I$ = *dn−i−e*:*1*
AOT-theorem *dn−i−e*:*2*:
assumes ‹$\neg\neg\varphi$›
shows ‹φ›
⟨*proof*⟩
lemmas $\neg\neg E$ = *dn−i−e*:*2*

AOT-theorem *modus−tollens*:*1*:
assumes ‹$\varphi \rightarrow \psi$› **and** ‹$\neg\psi$›
shows ‹$\neg\varphi$›
⟨*proof*⟩
AOT-theorem *modus−tollens*:*2*:
assumes ‹$\varphi \rightarrow \neg\psi$› **and** ‹$\psi$›
shows ‹$\neg\varphi$›
⟨*proof*⟩
lemmas *MT* = *modus−tollens*:*1* *modus−tollens*:*2*

AOT-theorem *contraposition*:*1*[*1*]:
assumes ‹$\varphi \rightarrow \psi$›
shows ‹$\neg\psi \rightarrow \neg\varphi$›
⟨*proof*⟩
AOT-theorem *contraposition*:*1*[*2*]:
assumes ‹$\neg\psi \rightarrow \neg\varphi$›
shows ‹$\varphi \rightarrow \psi$›
⟨*proof*⟩

AOT-theorem *contraposition*:*2*:
assumes ‹$\varphi \rightarrow \neg\psi$›
shows ‹$\psi \rightarrow \neg\varphi$›
⟨*proof*⟩

AOT-theorem *reductio−aa*:*1*:
assumes ‹$\neg\varphi \vdash \neg\psi$› **and** ‹$\neg\varphi \vdash \psi$›
shows ‹φ›
⟨*proof*⟩
AOT-theorem *reductio−aa*:*2*:
assumes ‹$\varphi \vdash \neg\psi$› **and** ‹$\varphi \vdash \psi$›
shows ‹$\neg\varphi$›
⟨*proof*⟩
lemmas *RAA* = *reductio−aa*:*1* *reductio−aa*:*2*

AOT-theorem *exc−mid*: ‹$\varphi \vee \neg\varphi$›
⟨*proof*⟩

AOT-theorem *non−contradiction*: ‹$\neg(\varphi \mathbin{\&} \neg\varphi)$›
⟨*proof*⟩

AOT-theorem *con−dis−taut*:*1*: ‹$(\varphi \mathbin{\&} \psi) \rightarrow \varphi$›
⟨*proof*⟩
AOT-theorem *con−dis−taut*:*2*: ‹$(\varphi \mathbin{\&} \psi) \rightarrow \psi$›
⟨*proof*⟩
lemmas *Conjunction Simplification* = *con−dis−taut*:*1* *con−dis−taut*:*2*

AOT-theorem *con−dis−taut*:*3*: ‹$\varphi \rightarrow (\varphi \vee \psi)$›
⟨*proof*⟩
AOT-theorem *con−dis−taut*:*4*: ‹$\psi \rightarrow (\varphi \vee \psi)$›

⟨*proof*⟩
lemmas *Disjunction Addition* = *con−dis−taut*:*3* *con−dis−taut*:*4*

AOT-theorem *con−dis−taut*:*5*: ‹$\varphi \to (\psi \to (\varphi \ \& \ \psi))$›
⟨*proof*⟩
lemmas *Adjunction* = *con−dis−taut*:*5*

AOT-theorem *con−dis−taut*:*6*: ‹$(\varphi \ \& \ \varphi) \equiv \varphi$›
⟨*proof*⟩
lemmas *Idempotence of* & = *con−dis−taut*:*6*

AOT-theorem *con−dis−taut*:*7*: ‹$(\varphi \lor \varphi) \equiv \varphi$›
⟨*proof*⟩
lemmas *Idempotence of* ∨ = *con−dis−taut*:*7*

AOT-theorem *con−dis−i−e*:*1*:
assumes ‹φ› **and** ‹ψ›
shows ‹$\varphi \ \& \ \psi$›
⟨*proof*⟩
lemmas &*I* = *con−dis−i−e*:*1*

AOT-theorem *con−dis−i−e*:*2*:*a*:
assumes ‹$\varphi \ \& \ \psi$›
shows ‹φ›
⟨*proof*⟩
AOT-theorem *con−dis−i−e*:*2*:*b*:
assumes ‹$\varphi \ \& \ \psi$›
shows ‹ψ›
⟨*proof*⟩
lemmas &*E* = *con−dis−i−e*:*2*:*a* *con−dis−i−e*:*2*:*b*

AOT-theorem *con−dis−i−e*:*3*:*a*:
assumes ‹φ›
shows ‹$\varphi \lor \psi$›
⟨*proof*⟩
AOT-theorem *con−dis−i−e*:*3*:*b*:
assumes ‹ψ›
shows ‹$\varphi \lor \psi$›
⟨*proof*⟩
AOT-theorem *con−dis−i−e*:*3*:*c*:
assumes ‹$\varphi \lor \psi$› **and** ‹$\varphi \to \chi$› **and** ‹$\psi \to \Theta$›
shows ‹$\chi \lor \Theta$›
⟨*proof*⟩
lemmas ∨*I* = *con−dis−i−e*:*3*:*a* *con−dis−i−e*:*3*:*b* *con−dis−i−e*:*3*:*c*

AOT-theorem *con−dis−i−e*:*4*:*a*:
assumes ‹$\varphi \lor \psi$› **and** ‹$\varphi \to \chi$› **and** ‹$\psi \to \chi$›
shows ‹χ›
⟨*proof*⟩
AOT-theorem *con−dis−i−e*:*4*:*b*:
assumes ‹$\varphi \lor \psi$› **and** ‹$\neg\varphi$›
shows ‹ψ›
⟨*proof*⟩
AOT-theorem *con−dis−i−e*:*4*:*c*:
assumes ‹$\varphi \lor \psi$› **and** ‹$\neg\psi$›
shows ‹φ›
⟨*proof*⟩
lemmas ∨*E* = *con−dis−i−e*:*4*:*a* *con−dis−i−e*:*4*:*b* *con−dis−i−e*:*4*:*c*

AOT-theorem *raa−cor*:*1*:
assumes ‹$\neg\varphi \vdash \psi \ \& \ \neg\psi$›
shows ‹φ›

⟨*proof*⟩
AOT-theorem *raa−cor*:*2*:
assumes ‹$\varphi \vdash \psi$ & $\neg\psi$›
shows ‹$\neg\varphi$›
⟨*proof*⟩
AOT-theorem *raa−cor*:*3*:
assumes ‹φ› **and** ‹$\neg\psi \vdash \neg\varphi$›
shows ‹ψ›
⟨*proof*⟩
AOT-theorem *raa−cor*:*4*:
assumes ‹$\neg\varphi$› **and** ‹$\neg\psi \vdash \varphi$›
shows ‹ψ›
⟨*proof*⟩
AOT-theorem *raa−cor*:*5*:
assumes ‹φ› **and** ‹$\psi \vdash \neg\varphi$›
shows ‹$\neg\psi$›
⟨*proof*⟩
AOT-theorem *raa−cor*:*6*:
assumes ‹$\neg\varphi$› **and** ‹$\psi \vdash \varphi$›
shows ‹$\neg\psi$›
⟨*proof*⟩

AOT-theorem *oth−class−taut*:*1*:*a*: ‹$(\varphi \rightarrow \psi) \equiv \neg(\varphi$ & $\neg\psi)$›
⟨*proof*⟩
AOT-theorem *oth−class−taut*:*1*:*b*: ‹$\neg(\varphi \rightarrow \psi) \equiv (\varphi$ & $\neg\psi)$›
⟨*proof*⟩
AOT-theorem *oth−class−taut*:*1*:*c*: ‹$(\varphi \rightarrow \psi) \equiv (\neg\varphi \vee \psi)$›
⟨*proof*⟩

AOT-theorem *oth−class−taut*:*2*:*a*: ‹$(\varphi$ & $\psi) \equiv (\psi$ & $\varphi)$›
⟨*proof*⟩
lemmas *Commutativity of* & = *oth−class−taut*:*2*:*a*
AOT-theorem *oth−class−taut*:*2*:*b*: ‹$(\varphi$ & $(\psi$ & $\chi)) \equiv ((\varphi$ & $\psi)$ & $\chi)$›
⟨*proof*⟩
lemmas *Associativity of* & = *oth−class−taut*:*2*:*b*
AOT-theorem *oth−class−taut*:*2*:*c*: ‹$(\varphi \vee \psi) \equiv (\psi \vee \varphi)$›
⟨*proof*⟩
lemmas *Commutativity of* $\vee$ = *oth−class−taut*:*2*:*c*
AOT-theorem *oth−class−taut*:*2*:*d*: ‹$(\varphi \vee (\psi \vee \chi)) \equiv ((\varphi \vee \psi) \vee \chi)$›
⟨*proof*⟩
lemmas *Associativity of* $\vee$ = *oth−class−taut*:*2*:*d*
AOT-theorem *oth−class−taut*:*2*:*e*: ‹$(\varphi \equiv \psi) \equiv (\psi \equiv \varphi)$›
⟨*proof*⟩
lemmas *Commutativity of* $\equiv$ = *oth−class−taut*:*2*:*e*
AOT-theorem *oth−class−taut*:*2*:*f*: ‹$(\varphi \equiv (\psi \equiv \chi)) \equiv ((\varphi \equiv \psi) \equiv \chi)$›
⟨*proof*⟩
lemmas *Associativity of* $\equiv$ = *oth−class−taut*:*2*:*f*

AOT-theorem *oth−class−taut*:*3*:*a*: ‹$\varphi \equiv \varphi$›
⟨*proof*⟩
AOT-theorem *oth−class−taut*:*3*:*b*: ‹$\varphi \equiv \neg\neg\varphi$›
⟨*proof*⟩
AOT-theorem *oth−class−taut*:*3*:*c*: ‹$\neg(\varphi \equiv \neg\varphi)$›
⟨*proof*⟩

AOT-theorem *oth−class−taut*:*4*:*a*: ‹$(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut*:*4*:*b*: ‹$(\varphi \equiv \psi) \equiv (\neg\varphi \equiv \neg\psi)$›
⟨*proof*⟩
AOT-theorem *oth−class−taut*:*4*:*c*: ‹$(\varphi \equiv \psi) \rightarrow ((\varphi \rightarrow \chi) \equiv (\psi \rightarrow \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut*:*4*:*d*: ‹$(\varphi \equiv \psi) \rightarrow ((\chi \rightarrow \varphi) \equiv (\chi \rightarrow \psi))$›
⟨*proof*⟩

AOT-theorem *oth−class−taut:4:e*: ‹$(\varphi \equiv \psi) \rightarrow ((\varphi \ \& \ \chi) \equiv (\psi \ \& \ \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:4:f*: ‹$(\varphi \equiv \psi) \rightarrow ((\chi \ \& \ \varphi) \equiv (\chi \ \& \ \psi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:4:g*: ‹$(\varphi \equiv \psi) \equiv ((\varphi \ \& \ \psi) \vee (\neg\varphi \ \& \ \neg\psi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:4:h*: ‹$\neg(\varphi \equiv \psi) \equiv ((\varphi \ \& \ \neg\psi) \vee (\neg\varphi \ \& \ \psi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:5:a*: ‹$(\varphi \ \& \ \psi) \equiv \neg(\neg\varphi \vee \neg\psi)$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:5:b*: ‹$(\varphi \vee \psi) \equiv \neg(\neg\varphi \ \& \ \neg\psi)$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:5:c*: ‹$\neg(\varphi \ \& \ \psi) \equiv (\neg\varphi \vee \neg\psi)$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:5:d*: ‹$\neg(\varphi \vee \psi) \equiv (\neg\varphi \ \& \ \neg\psi)$›
⟨*proof*⟩

lemmas *DeMorgan* = *oth−class−taut:5:c oth−class−taut:5:d*

AOT-theorem *oth−class−taut:6:a*:
‹$(\varphi \ \& \ (\psi \vee \chi)) \equiv ((\varphi \ \& \ \psi) \vee (\varphi \ \& \ \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:6:b*:
‹$(\varphi \vee (\psi \ \& \ \chi)) \equiv ((\varphi \vee \psi) \ \& \ (\varphi \vee \chi))$›
⟨*proof*⟩

AOT-theorem *oth−class−taut:7:a*: ‹$((\varphi \ \& \ \psi) \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$›
⟨*proof*⟩
lemmas *Exportation* = *oth−class−taut:7:a*
AOT-theorem *oth−class−taut:7:b*: ‹$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \ \& \ \psi) \rightarrow \chi)$›
⟨*proof*⟩
lemmas *Importation* = *oth−class−taut:7:b*

AOT-theorem *oth−class−taut:8:a*:
‹$(\varphi \rightarrow (\psi \rightarrow \chi)) \equiv (\psi \rightarrow (\varphi \rightarrow \chi))$›
⟨*proof*⟩
lemmas *Permutation* = *oth−class−taut:8:a*
AOT-theorem *oth−class−taut:8:b*:
‹$(\varphi \rightarrow \psi) \rightarrow ((\varphi \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \ \& \ \chi)))$›
⟨*proof*⟩
lemmas *Composition* = *oth−class−taut:8:b*
AOT-theorem *oth−class−taut:8:c*:
‹$(\varphi \rightarrow \chi) \rightarrow ((\psi \rightarrow \chi) \rightarrow ((\varphi \vee \psi) \rightarrow \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:8:d*:
‹$((\varphi \rightarrow \psi) \ \& \ (\chi \rightarrow \Theta)) \rightarrow ((\varphi \ \& \ \chi) \rightarrow (\psi \ \& \ \Theta))$›
⟨*proof*⟩
lemmas *Double Composition* = *oth−class−taut:8:d*
AOT-theorem *oth−class−taut:8:e*:
‹$((\varphi \ \& \ \psi) \equiv (\varphi \ \& \ \chi)) \equiv (\varphi \rightarrow (\psi \equiv \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:8:f*:
‹$((\varphi \ \& \ \psi) \equiv (\chi \ \& \ \psi)) \equiv (\psi \rightarrow (\varphi \equiv \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:8:g*:
‹$(\psi \equiv \chi) \rightarrow ((\varphi \vee \psi) \equiv (\varphi \vee \chi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:8:h*:
‹$(\psi \equiv \chi) \rightarrow ((\psi \vee \varphi) \equiv (\chi \vee \varphi))$›
⟨*proof*⟩
AOT-theorem *oth−class−taut:8:i*:
‹$(\varphi \equiv (\psi \ \& \ \chi)) \rightarrow (\psi \rightarrow (\varphi \equiv \chi))$›
⟨*proof*⟩

AOT-theorem *intro−elim:1*:
  assumes ‹$\varphi \vee \psi$› **and** ‹$\varphi \equiv \chi$› **and** ‹$\psi \equiv \Theta$›
  shows ‹$\chi \vee \Theta$›
  ⟨*proof*⟩

AOT-theorem *intro−elim:2*:
  assumes ‹$\varphi \rightarrow \psi$› **and** ‹$\psi \rightarrow \varphi$›
  shows ‹$\varphi \equiv \psi$›
  ⟨*proof*⟩
lemmas $\equiv I$ = *intro−elim:2*

AOT-theorem *intro−elim:3:a*:
  assumes ‹$\varphi \equiv \psi$› **and** ‹φ›
  shows ‹ψ›
  ⟨*proof*⟩
AOT-theorem *intro−elim:3:b*:
  assumes ‹$\varphi \equiv \psi$› **and** ‹ψ›
  shows ‹φ›
  ⟨*proof*⟩
AOT-theorem *intro−elim:3:c*:
  assumes ‹$\varphi \equiv \psi$› **and** ‹$\neg\varphi$›
  shows ‹$\neg\psi$›
  ⟨*proof*⟩
AOT-theorem *intro−elim:3:d*:
  assumes ‹$\varphi \equiv \psi$› **and** ‹$\neg\psi$›
  shows ‹$\neg\varphi$›
  ⟨*proof*⟩
AOT-theorem *intro−elim:3:e*:
  assumes ‹$\varphi \equiv \psi$› **and** ‹$\psi \equiv \chi$›
  shows ‹$\varphi \equiv \chi$›
  ⟨*proof*⟩
declare *intro−elim:3:e*[*trans*]
AOT-theorem *intro−elim:3:f*:
  assumes ‹$\varphi \equiv \psi$› **and** ‹$\varphi \equiv \chi$›
  shows ‹$\chi \equiv \psi$›
  ⟨*proof*⟩
lemmas $\equiv E$ = *intro−elim:3:a* *intro−elim:3:b* *intro−elim:3:c*
  intro−elim:3:d *intro−elim:3:e* *intro−elim:3:f*

declare *Commutativity of* $\equiv$[*THEN* $\equiv E(1)$, *sym*]

AOT-theorem *rule−eq−df:1*:
  assumes ‹$\varphi \equiv_{df} \psi$›
  shows ‹$\varphi \equiv \psi$›
  ⟨*proof*⟩
lemmas $\equiv Df$ = *rule−eq−df:1*
AOT-theorem *rule−eq−df:2*:
  assumes ‹$\varphi \equiv_{df} \psi$› **and** ‹φ›
  shows ‹ψ›
  ⟨*proof*⟩
lemmas $\equiv_{df} E$ = *rule−eq−df:2*
AOT-theorem *rule−eq−df:3*:
  assumes ‹$\varphi \equiv_{df} \psi$› **and** ‹ψ›
  shows ‹φ›
  ⟨*proof*⟩
lemmas $\equiv_{df} I$ = *rule−eq−df:3*

AOT-theorem *df−simplify:1*:
  assumes ‹$\varphi \equiv (\psi \ \& \ \chi)$› **and** ‹$\psi$›
  shows ‹$\varphi \equiv \chi$›
  ⟨*proof*⟩

AOT-theorem *df−simplify:2*:
assumes ‹$\varphi \equiv (\psi\ \&\ \chi)$› **and** ‹$\chi$›
shows ‹$\varphi \equiv \psi$›
⟨*proof*⟩
lemmas $\equiv S$ = *df−simplify:1 df−simplify:2*

8.6 The Theory of Quantification

AOT-theorem *rule−ui:1*:
assumes ‹$\forall\alpha\ \varphi\{\alpha\}$› **and** ‹$\tau\downarrow$›
shows ‹$\varphi\{\tau\}$›
⟨*proof*⟩
AOT-theorem *rule−ui:2[const-var]*:
assumes ‹$\forall\alpha\ \varphi\{\alpha\}$›
shows ‹$\varphi\{\beta\}$›
⟨*proof*⟩
AOT-theorem *rule−ui:2[lambda]*:
assumes ‹$\forall F\ \varphi\{F\}$› **and** ‹*INSTANCE-OF-CQT-2*(ψ)›
shows ‹$\varphi\{[\lambda\nu_1...\nu_n\ \psi\{\nu_1...\nu_n\}]\}$›
⟨*proof*⟩
AOT-theorem *rule−ui:3*:
assumes ‹$\forall\alpha\ \varphi\{\alpha\}$›
shows ‹$\varphi\{\alpha\}$›
⟨*proof*⟩
lemmas $\forall E$ = *rule−ui:1 rule−ui:2[const-var]*
rule−ui:2[lambda] rule−ui:3

AOT-theorem *cqt−orig:1[const-var]*: ‹$\forall\alpha\ \varphi\{\alpha\} \rightarrow \varphi\{\beta\}$›
⟨*proof*⟩
AOT-theorem *cqt−orig:1[lambda]*:
assumes ‹*INSTANCE-OF-CQT-2*(ψ)›
shows ‹$\forall F\ \varphi\{F\} \rightarrow \varphi\{[\lambda\nu_1...\nu_n\ \psi\{\nu_1...\nu_n\}]\}$›
⟨*proof*⟩
AOT-theorem *cqt−orig:2*: ‹$\forall\alpha\ (\varphi \rightarrow \psi\{\alpha\}) \rightarrow (\varphi \rightarrow \forall\alpha\ \psi\{\alpha\})$›
⟨*proof*⟩
AOT-theorem *cqt−orig:3*: ‹$\forall\alpha\ \varphi\{\alpha\} \rightarrow \varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *universal*:
assumes ‹*for arbitrary* β: $\varphi\{\beta\}$›
shows ‹$\forall\alpha\ \varphi\{\alpha\}$›
⟨*proof*⟩
lemmas $\forall I$ = *universal*

⟨*ML*⟩

AOT-theorem *cqt−basic:1*: ‹$\forall\alpha\forall\beta\ \varphi\{\alpha,\beta\} \equiv \forall\beta\forall\alpha\ \varphi\{\alpha,\beta\}$›
⟨*proof*⟩

AOT-theorem *cqt−basic:2*:
‹$\forall\alpha(\varphi\{\alpha\} \equiv \psi\{\alpha\}) \equiv (\forall\alpha(\varphi\{\alpha\} \rightarrow \psi\{\alpha\})\ \&\ \forall\alpha(\psi\{\alpha\} \rightarrow \varphi\{\alpha\}))$›
⟨*proof*⟩

AOT-theorem *cqt−basic:3*: ‹$\forall\alpha(\varphi\{\alpha\} \equiv \psi\{\alpha\}) \rightarrow (\forall\alpha\ \varphi\{\alpha\} \equiv \forall\alpha\ \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *cqt−basic:4*: ‹$\forall\alpha(\varphi\{\alpha\}\ \&\ \psi\{\alpha\}) \rightarrow (\forall\alpha\ \varphi\{\alpha\}\ \&\ \forall\alpha\ \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *cqt−basic:5*: ‹$(\forall\alpha_1...\forall\alpha_n(\varphi\{\alpha_1...\alpha_n\})) \rightarrow \varphi\{\alpha_1...\alpha_n\}$›

$\langle proof \rangle$

AOT-theorem *cqt–basic:6*: ‹$\forall\,\alpha\forall\,\alpha\ \varphi\{\alpha\} \equiv \forall\,\alpha\ \varphi\{\alpha\}$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:7*: ‹$(\varphi \to \forall\,\alpha\ \psi\{\alpha\}) \equiv \forall\,\alpha(\varphi \to \psi\{\alpha\})$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:8*: ‹$(\forall\,\alpha\ \varphi\{\alpha\} \vee \forall\,\alpha\ \psi\{\alpha\}) \to \forall\,\alpha\ (\varphi\{\alpha\} \vee \psi\{\alpha\})$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:9*:
‹$(\forall\,\alpha\ (\varphi\{\alpha\} \to \psi\{\alpha\})\ \&\ \forall\,\alpha\ (\psi\{\alpha\} \to \chi\{\alpha\})) \to \forall\,\alpha(\varphi\{\alpha\} \to \chi\{\alpha\})$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:10*:
‹$(\forall\,\alpha(\varphi\{\alpha\} \equiv \psi\{\alpha\})\ \&\ \forall\,\alpha(\psi\{\alpha\} \equiv \chi\{\alpha\})) \to \forall\,\alpha\ (\varphi\{\alpha\} \equiv \chi\{\alpha\})$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:11*: ‹$\forall\,\alpha(\varphi\{\alpha\} \equiv \psi\{\alpha\}) \equiv \forall\,\alpha\ (\psi\{\alpha\} \equiv \varphi\{\alpha\})$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:12*: ‹$\forall\,\alpha\ \varphi\{\alpha\} \to \forall\,\alpha\ (\psi\{\alpha\} \to \varphi\{\alpha\})$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:13*: ‹$\forall\,\alpha\ \varphi\{\alpha\} \equiv \forall\,\beta\ \varphi\{\beta\}$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:14*:
‹$(\forall\,\alpha_1 ... \forall\,\alpha_n\ (\varphi\{\alpha_1 ... \alpha_n\} \to \psi\{\alpha_1 ... \alpha_n\})) \to$
$((\forall\,\alpha_1 ... \forall\,\alpha_n\ \varphi\{\alpha_1 ... \alpha_n\}) \to (\forall\,\alpha_1 ... \forall\,\alpha_n\ \psi\{\alpha_1 ... \alpha_n\}))$›
$\langle proof \rangle$

AOT-theorem *cqt–basic:15*:
‹$(\forall\,\alpha_1 ... \forall\,\alpha_n\ (\varphi \to \psi\{\alpha_1 ... \alpha_n\})) \to (\varphi \to (\forall\,\alpha_1 ... \forall\,\alpha_n\ \psi\{\alpha_1 ... \alpha_n\}))$›
$\langle proof \rangle$

AOT-theorem *universal–cor*:
assumes ‹*for arbitrary* β: $\varphi\{\beta\}$›
shows ‹$\forall\,\alpha\ \varphi\{\alpha\}$›
$\langle proof \rangle$

AOT-theorem *existential:1*:
assumes ‹$\varphi\{\tau\}$› **and** ‹$\tau\downarrow$›
shows ‹$\exists\,\alpha\ \varphi\{\alpha\}$›
$\langle proof \rangle$

AOT-theorem *existential:2[const-var]*:
assumes ‹$\varphi\{\beta\}$›
shows ‹$\exists\,\alpha\ \varphi\{\alpha\}$›
$\langle proof \rangle$

AOT-theorem *existential:2[lambda]*:
assumes ‹$\varphi\{[\lambda\nu_1 ... \nu_n\ \psi\{\nu_1 ... \nu_n\}]\}$› **and** ‹*INSTANCE-OF-CQT-2*(ψ)›
shows ‹$\exists\,\alpha\ \varphi\{\alpha\}$›
$\langle proof \rangle$
lemmas $\exists\,I$ = *existential:1 existential:2[const-var]*
existential:2[lambda]

AOT-theorem *instantiation*:
assumes ‹*for arbitrary* β: $\varphi\{\beta\} \vdash \psi$› **and** ‹$\exists\,\alpha\ \varphi\{\alpha\}$›
shows ‹ψ›
$\langle proof \rangle$

lemmas $\exists E = instantiation$

AOT-theorem *cqt−further:1*: ‹$\forall\alpha\ \varphi\{\alpha\} \rightarrow \exists\alpha\ \varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *cqt−further:2*: ‹$\neg\forall\alpha\ \varphi\{\alpha\} \rightarrow \exists\alpha\ \neg\varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *cqt−further:3*: ‹$\forall\alpha\ \varphi\{\alpha\} \equiv \neg\exists\alpha\ \neg\varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *cqt−further:4*: ‹$\neg\exists\alpha\ \varphi\{\alpha\} \rightarrow \forall\alpha\ \neg\varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *cqt−further:5*: ‹$\exists\alpha\ (\varphi\{\alpha\}\ \&\ \psi\{\alpha\}) \rightarrow (\exists\alpha\ \varphi\{\alpha\}\ \&\ \exists\alpha\ \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *cqt−further:6*: ‹$\exists\alpha\ (\varphi\{\alpha\} \vee \psi\{\alpha\}) \rightarrow (\exists\alpha\ \varphi\{\alpha\} \vee \exists\alpha\ \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *cqt−further:7*: ‹$\exists\alpha\ \varphi\{\alpha\} \equiv \exists\beta\ \varphi\{\beta\}$›
⟨*proof*⟩

AOT-theorem *cqt−further:8*:
‹$(\forall\alpha\ \varphi\{\alpha\}\ \&\ \forall\alpha\ \psi\{\alpha\}) \rightarrow \forall\alpha\ (\varphi\{\alpha\} \equiv \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *cqt−further:9*:
‹$(\neg\exists\alpha\ \varphi\{\alpha\}\ \&\ \neg\exists\alpha\ \psi\{\alpha\}) \rightarrow \forall\alpha\ (\varphi\{\alpha\} \equiv \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *cqt−further:10*:
‹$(\exists\alpha\ \varphi\{\alpha\}\ \&\ \neg\exists\alpha\ \psi\{\alpha\}) \rightarrow \neg\forall\alpha\ (\varphi\{\alpha\} \equiv \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *cqt−further:11*: ‹$\exists\alpha\exists\beta\ \varphi\{\alpha,\beta\} \equiv \exists\beta\exists\alpha\ \varphi\{\alpha,\beta\}$›
⟨*proof*⟩

8.7 Logical Existence, Identity, and Truth

AOT-theorem *log−prop−prop:1*: ‹$[\lambda\ \varphi]\downarrow$›
⟨*proof*⟩

AOT-theorem *log−prop−prop:2*: ‹$\varphi\downarrow$›
⟨*proof*⟩

AOT-theorem *exist−nec*: ‹$\tau\downarrow \rightarrow \Box\tau\downarrow$›
⟨*proof*⟩

class *AOT-Term-id* = *AOT-Term* +
assumes *t=t−proper:1[AOT]*: ‹$[v \models \tau = \tau' \rightarrow \tau\downarrow]$›
and *t=t−proper:2[AOT]*: ‹$[v \models \tau = \tau' \rightarrow \tau'\downarrow]$›

instance κ :: *AOT-Term-id*
⟨*proof*⟩

instance *rel* :: (*AOT-κs*) *AOT-Term-id*
⟨*proof*⟩

instance o :: *AOT-Term-id*
⟨*proof*⟩

instance *prod* :: (*AOT-Term-id*, *AOT-Term-id*) *AOT-Term-id*
⟨*proof*⟩

AOT-register-type-constraints
Term: ‹-::*AOT-Term-id*› ‹-::*AOT-Term-id*›
AOT-register-type-constraints
Individual: ‹κ› ‹-::{*AOT-κs*, *AOT-Term-id*}›
AOT-register-type-constraints
Relation: ‹<-::{*AOT-κs*, *AOT-Term-id*}>›

AOT-theorem *id−rel−nec−equiv*:*1*:
‹$\Pi = \Pi' \rightarrow \Box\forall x_1...\forall x_n\ ([\Pi]x_1...x_n \equiv [\Pi']x_1...x_n)$›
⟨*proof*⟩

AOT-theorem *id−rel−nec−equiv*:*2*: ‹$\varphi = \psi \rightarrow \Box(\varphi \equiv \psi)$›
⟨*proof*⟩

AOT-theorem *rule=E*:
assumes ‹$\varphi\{\tau\}$› **and** ‹$\tau = \sigma$›
shows ‹$\varphi\{\sigma\}$›
⟨*proof*⟩

AOT-theorem *propositions−lemma*:*1*: ‹$[\lambda\ \varphi] = \varphi$›
⟨*proof*⟩

AOT-theorem *propositions−lemma*:*2*: ‹$[\lambda\ \varphi] \equiv \varphi$›
⟨*proof*⟩

propositions-lemma:3 through propositions-lemma:5 hold implicitly

AOT-theorem *propositions−lemma*:*6*: ‹$(\varphi \equiv \psi) \equiv ([\lambda\ \varphi] \equiv [\lambda\ \psi])$›
⟨*proof*⟩

dr-alphabetic-rules holds implicitly

AOT-theorem *oa−exist*:*1*: ‹$O!\downarrow$›
⟨*proof*⟩

AOT-theorem *oa−exist*:*2*: ‹$A!\downarrow$›
⟨*proof*⟩

AOT-theorem *oa−exist*:*3*: ‹$O!x \vee A!x$›
⟨*proof*⟩

AOT-theorem *p−identity−thm2*:*1*: ‹$F = G \equiv \Box\forall x(x[F] \equiv x[G])$›
⟨*proof*⟩

AOT-theorem *p−identity−thm2*:*2*[*2*]:
‹$F = G \equiv \forall y_1([\lambda x\ [F]xy_1] = [\lambda x\ [G]xy_1]\ \&\ [\lambda x\ [F]y_1x] = [\lambda x\ [G]y_1x])$›
⟨*proof*⟩

AOT-theorem *p−identity−thm2*:*2*[*3*]:
‹$F = G \equiv \forall y_1\forall y_2([\lambda x\ [F]xy_1y_2] = [\lambda x\ [G]xy_1y_2]\ \&$
$[\lambda x\ [F]y_1xy_2] = [\lambda x\ [G]y_1xy_2]\ \&$
$[\lambda x\ [F]y_1y_2x] = [\lambda x\ [G]y_1y_2x])$›
⟨*proof*⟩

AOT-theorem *p−identity−thm2*:*2*[*4*]:
‹$F = G \equiv \forall y_1\forall y_2\forall y_3([\lambda x\ [F]xy_1y_2y_3] = [\lambda x\ [G]xy_1y_2y_3]\ \&$
$[\lambda x\ [F]y_1xy_2y_3] = [\lambda x\ [G]y_1xy_2y_3]\ \&$
$[\lambda x\ [F]y_1y_2xy_3] = [\lambda x\ [G]y_1y_2xy_3]\ \&$
$[\lambda x\ [F]y_1y_2y_3x] = [\lambda x\ [G]y_1y_2y_3x])$›
⟨*proof*⟩

AOT-theorem *p−identity−thm2:2*:
‹$F = G \equiv \forall x_1...\forall x_n$ «*AOT-sem-proj-id* x_1x_n ($\lambda\ \tau$. «$[F]\tau$») ($\lambda\ \tau$. «$[G]\tau$»)»›
⟨*proof*⟩

AOT-theorem *p−identity−thm2:3*:
‹$p = q \equiv [\lambda x\ p] = [\lambda x\ q]$›
⟨*proof*⟩

class *AOT-Term-id-2* = *AOT-Term-id* + **assumes** *id−eq:1*: ‹$[v \models \alpha = \alpha]$›

instance κ :: *AOT-Term-id-2*
⟨*proof*⟩

instance *rel* :: ({*AOT-κs*,*AOT-Term-id-2*}) *AOT-Term-id-2*
⟨*proof*⟩

instance o :: *AOT-Term-id-2*
⟨*proof*⟩

instance *prod* :: (*AOT-Term-id-2*, *AOT-Term-id-2*) *AOT-Term-id-2*
⟨*proof*⟩

AOT-register-type-constraints
Term: ‹-::*AOT-Term-id-2*› ‹-::*AOT-Term-id-2*›
AOT-register-type-constraints
Individual: ‹κ› ‹-::{*AOT-κs*, *AOT-Term-id-2*}›
AOT-register-type-constraints
Relation: ‹<-::{*AOT-κs*, *AOT-Term-id-2*}>›

AOT-theorem *id−eq:2*: ‹$\alpha = \beta \to \beta = \alpha$›
⟨*proof*⟩

AOT-theorem *id−eq:3*: ‹$\alpha = \beta\ \&\ \beta = \gamma \to \alpha = \gamma$›
⟨*proof*⟩

AOT-theorem *id−eq:4*: ‹$\alpha = \beta \equiv \forall\gamma\ (\alpha = \gamma \equiv \beta = \gamma)$›
⟨*proof*⟩

AOT-theorem *rule=I:1*:
assumes ‹$\tau\downarrow$›
shows ‹$\tau = \tau$›
⟨*proof*⟩

AOT-theorem *rule=I:2[const-var]*: $\alpha = \alpha$
⟨*proof*⟩

AOT-theorem *rule=I:2[lambda]*:
assumes ‹*INSTANCE-OF-CQT-2*(φ)›
shows $[\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}] = [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]$
⟨*proof*⟩

lemmas *=I* = *rule=I:1* *rule=I:2[const-var]* *rule=I:2[lambda]*

AOT-theorem *rule−id−df:1*:
assumes ‹$\tau\{\alpha_1...\alpha_n\} =_{df} \sigma\{\alpha_1...\alpha_n\}$› **and** ‹$\sigma\{\tau_1...\tau_n\}\downarrow$›
shows ‹$\tau\{\tau_1...\tau_n\} = \sigma\{\tau_1...\tau_n\}$›
⟨*proof*⟩

AOT-theorem *rule−id−df:1[zero]*:
assumes ‹$\tau =_{df} \sigma$› **and** ‹$\sigma\downarrow$›
shows ‹$\tau = \sigma$›
⟨*proof*⟩

AOT-theorem *rule−id−df:2:a*:
assumes ‹$\tau\{\alpha_1...\alpha_n\} =_{df} \sigma\{\alpha_1...\alpha_n\}$› **and** ‹$\sigma\{\tau_1...\tau_n\}\downarrow$› **and** ‹$\varphi\{\tau\{\tau_1...\tau_n\}\}$›
shows ‹$\varphi\{\sigma\{\tau_1...\tau_n\}\}$›
⟨*proof*⟩

AOT-theorem *rule−id−df:2:a*[*2*]:
assumes ‹$\tau\{«(\alpha_1,\alpha_2)»\} =_{df} \sigma\{«(\alpha_1,\alpha_2)»\}$›
and ‹$\sigma\{«(\tau_1,\tau_2)»\}\downarrow$›
and ‹$\varphi\{\tau\{«(\tau_1,\tau_2)»\}\}$›
shows ‹$\varphi\{\sigma\{«(\tau_1::'a::AOT\text{-}Term\text{-}id\text{-}2,\tau_2::'b::AOT\text{-}Term\text{-}id\text{-}2)»\}\}$›
⟨*proof*⟩

AOT-theorem *rule−id−df:2:a*[*zero*]:
assumes ‹$\tau =_{df} \sigma$› **and** ‹$\sigma\downarrow$› **and** ‹$\varphi\{\tau\}$›
shows ‹$\varphi\{\sigma\}$›
⟨*proof*⟩

lemmas $=_{df}E$ = *rule−id−df:2:a rule−id−df:2:a*[*zero*]

AOT-theorem *rule−id−df:2:b*:
assumes ‹$\tau\{\alpha_1...\alpha_n\} =_{df} \sigma\{\alpha_1...\alpha_n\}$› **and** ‹$\sigma\{\tau_1...\tau_n\}\downarrow$› **and** ‹$\varphi\{\sigma\{\tau_1...\tau_n\}\}$›
shows ‹$\varphi\{\tau\{\tau_1...\tau_n\}\}$›
⟨*proof*⟩

AOT-theorem *rule−id−df:2:b*[*2*]:
assumes ‹$\tau\{«(\alpha_1,\alpha_2)»\} =_{df} \sigma\{«(\alpha_1,\alpha_2)»\}$›
and ‹$\sigma\{«(\tau_1,\tau_2)»\}\downarrow$›
and ‹$\varphi\{\sigma\{«(\tau_1,\tau_2)»\}\}$›
shows ‹$\varphi\{\tau\{«(\tau_1::'a::AOT\text{-}Term\text{-}id\text{-}2,\tau_2::'b::AOT\text{-}Term\text{-}id\text{-}2)»\}\}$›
⟨*proof*⟩

AOT-theorem *rule−id−df:2:b*[*zero*]:
assumes ‹$\tau =_{df} \sigma$› **and** ‹$\sigma\downarrow$› **and** ‹$\varphi\{\sigma\}$›
shows ‹$\varphi\{\tau\}$›
⟨*proof*⟩

lemmas $=_{df}I$ = *rule−id−df:2:b rule−id−df:2:b*[*zero*]

AOT-theorem *free−thms:1*: ‹$\tau\downarrow \equiv \exists\beta\ (\beta = \tau)$›
⟨*proof*⟩

AOT-theorem *free−thms:2*: ‹$\forall\alpha\ \varphi\{\alpha\} \rightarrow (\exists\beta\ (\beta = \tau) \rightarrow \varphi\{\tau\})$›
⟨*proof*⟩

AOT-theorem *free−thms:3*[*const-var*]: ‹$\exists\beta\ (\beta = \alpha)$›
⟨*proof*⟩

AOT-theorem *free−thms:3*[*lambda*]:
assumes ‹*INSTANCE-OF-CQT-2*(φ)›
shows ‹$\exists\beta\ (\beta = [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}])$›
⟨*proof*⟩

AOT-theorem *free−thms:4*[*rel*]:
‹$([\Pi]\kappa_1...\kappa_n \vee \kappa_1...\kappa_n[\Pi]) \rightarrow \exists\beta\ (\beta = \Pi)$›
⟨*proof*⟩

AOT-theorem *free−thms:4*[*vars*]:
‹$([\Pi]\kappa_1...\kappa_n \vee \kappa_1...\kappa_n[\Pi]) \rightarrow \exists\beta_1...\exists\beta_n\ (\beta_1...\beta_n = \kappa_1...\kappa_n)$›
⟨*proof*⟩

AOT-theorem *free−thms:4*[*1,rel*]:
‹$([\Pi]\kappa \vee \kappa[\Pi]) \rightarrow \exists\beta\ (\beta = \Pi)$›

⟨proof⟩

AOT-theorem *free−thms:4[1,1]*:
‹$([\Pi]\kappa \lor \kappa[\Pi]) \to \exists\beta\ (\beta = \kappa)$›
⟨proof⟩

AOT-theorem *free−thms:4[2,rel]*:
‹$([\Pi]\kappa_1\kappa_2 \lor \kappa_1\kappa_2[\Pi]) \to \exists\beta\ (\beta = \Pi)$›
⟨proof⟩
AOT-theorem *free−thms:4[2,1]*:
‹$([\Pi]\kappa_1\kappa_2 \lor \kappa_1\kappa_2[\Pi]) \to \exists\beta\ (\beta = \kappa_1)$›
⟨proof⟩
AOT-theorem *free−thms:4[2,2]*:
‹$([\Pi]\kappa_1\kappa_2 \lor \kappa_1\kappa_2[\Pi]) \to \exists\beta\ (\beta = \kappa_2)$›
⟨proof⟩
AOT-theorem *free−thms:4[3,rel]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3 \lor \kappa_1\kappa_2\kappa_3[\Pi]) \to \exists\beta\ (\beta = \Pi)$›
⟨proof⟩
AOT-theorem *free−thms:4[3,1]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3 \lor \kappa_1\kappa_2\kappa_3[\Pi]) \to \exists\beta\ (\beta = \kappa_1)$›
⟨proof⟩
AOT-theorem *free−thms:4[3,2]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3 \lor \kappa_1\kappa_2\kappa_3[\Pi]) \to \exists\beta\ (\beta = \kappa_2)$›
⟨proof⟩
AOT-theorem *free−thms:4[3,3]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3 \lor \kappa_1\kappa_2\kappa_3[\Pi]) \to \exists\beta\ (\beta = \kappa_3)$›
⟨proof⟩
AOT-theorem *free−thms:4[4,rel]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3\kappa_4 \lor \kappa_1\kappa_2\kappa_3\kappa_4[\Pi]) \to \exists\beta\ (\beta = \Pi)$›
⟨proof⟩
AOT-theorem *free−thms:4[4,1]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3\kappa_4 \lor \kappa_1\kappa_2\kappa_3\kappa_4[\Pi]) \to \exists\beta\ (\beta = \kappa_1)$›
⟨proof⟩
AOT-theorem *free−thms:4[4,2]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3\kappa_4 \lor \kappa_1\kappa_2\kappa_3\kappa_4[\Pi]) \to \exists\beta\ (\beta = \kappa_2)$›
⟨proof⟩
AOT-theorem *free−thms:4[4,3]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3\kappa_4 \lor \kappa_1\kappa_2\kappa_3\kappa_4[\Pi]) \to \exists\beta\ (\beta = \kappa_3)$›
⟨proof⟩
AOT-theorem *free−thms:4[4,4]*:
‹$([\Pi]\kappa_1\kappa_2\kappa_3\kappa_4 \lor \kappa_1\kappa_2\kappa_3\kappa_4[\Pi]) \to \exists\beta\ (\beta = \kappa_4)$›
⟨proof⟩

AOT-theorem *ex:1:a*: ‹$\forall\alpha\ \alpha\downarrow$›
⟨proof⟩
AOT-theorem *ex:1:b*: ‹$\forall\alpha\exists\beta(\beta = \alpha)$›
⟨proof⟩

AOT-theorem *ex:2:a*: ‹$\Box\alpha\downarrow$›
⟨proof⟩
AOT-theorem *ex:2:b*: ‹$\Box\exists\beta(\beta = \alpha)$›
⟨proof⟩

AOT-theorem *ex:3:a*: ‹$\Box\forall\alpha\ \alpha\downarrow$›
⟨proof⟩
AOT-theorem *ex:3:b*: ‹$\Box\forall\alpha\exists\beta(\beta = \alpha)$›
⟨proof⟩

AOT-theorem *ex:4:a*: ‹$\forall\alpha\ \Box\alpha\downarrow$›
⟨proof⟩
AOT-theorem *ex:4:b*: ‹$\forall\alpha\Box\exists\beta(\beta = \alpha)$›
⟨proof⟩

AOT-theorem *ex:5:a*: ‹$\Box\forall\alpha\ \Box\alpha\downarrow$›

⟨*proof*⟩
AOT-theorem *ex:5:b*: ‹$\Box\forall\alpha\Box\exists\beta(\beta = \alpha)$›
⟨*proof*⟩

AOT-theorem *all−self=:1*: ‹$\Box\forall\alpha(\alpha = \alpha)$›
⟨*proof*⟩
AOT-theorem *all−self=:2*: ‹$\forall\alpha\Box(\alpha = \alpha)$›
⟨*proof*⟩

AOT-theorem *id−nec:1*: ‹$\alpha = \beta \rightarrow \Box(\alpha = \beta)$›
⟨*proof*⟩

AOT-theorem *id−nec:2*: ‹$\tau = \sigma \rightarrow \Box(\tau = \sigma)$›
⟨*proof*⟩

AOT-theorem *term−out:1*: ‹$\varphi\{\alpha\} \equiv \exists\beta\ (\beta = \alpha\ \&\ \varphi\{\beta\})$›
⟨*proof*⟩

AOT-theorem *term−out:2*: ‹$\tau\downarrow \rightarrow (\varphi\{\tau\} \equiv \exists\alpha(\alpha = \tau\ \&\ \varphi\{\alpha\}))$›
⟨*proof*⟩

AOT-theorem *term−out:3*:
‹$(\varphi\{\alpha\}\ \&\ \forall\beta(\varphi\{\beta\} \rightarrow \beta = \alpha)) \equiv \forall\beta(\varphi\{\beta\} \equiv \beta = \alpha)$›
⟨*proof*⟩

AOT-theorem *term−out:4*:
‹$(\varphi\{\beta\}\ \&\ \forall\alpha(\varphi\{\alpha\} \rightarrow \alpha = \beta)) \equiv \forall\alpha(\varphi\{\alpha\} \equiv \alpha = \beta)$›
⟨*proof*⟩

AOT-define *AOT-exists-unique* :: ‹$\alpha \Rightarrow \varphi \Rightarrow \varphi$› *uniqueness:1*:
‹«*AOT-exists-unique* φ» $\equiv_{df} \exists\alpha\ (\varphi\{\alpha\}\ \&\ \forall\beta\ (\varphi\{\beta\} \rightarrow \beta = \alpha))$›
syntax (*input*) *-AOT-exists-unique* :: ‹$\alpha \Rightarrow \varphi \Rightarrow \varphi$› (‹∃!- -› [*1*,*40*])
syntax (**output**) *-AOT-exists-unique* :: ‹$\alpha \Rightarrow \varphi \Rightarrow \varphi$› (‹∃!-′(-′)› [*1*,*40*])
AOT-syntax-print-translations
-AOT-exists-unique τ φ <= *CONST AOT-exists-unique* (*-abs* τ φ)
syntax
-AOT-exists-unique-ellipse :: ‹*id-position* ⇒ *id-position* ⇒ $\varphi \Rightarrow \varphi$›
(‹∃!-...∃!- -› [*1*,*40*])
⟨*ML*⟩

context *AOT-meta-syntax*
begin
notation *AOT-exists-unique* (**binder** ‹∃ **!**› *20*)
end
context *AOT-no-meta-syntax*
begin
no-notation *AOT-exists-unique* (**binder** ‹∃ **!**› *20*)
end

AOT-theorem *uniqueness:2*: ‹$\exists!\alpha\ \varphi\{\alpha\} \equiv \exists\alpha\forall\beta(\varphi\{\beta\} \equiv \beta = \alpha)$›
⟨*proof*⟩

AOT-theorem *uni−most*: ‹$\exists!\alpha\ \varphi\{\alpha\} \rightarrow \forall\beta\forall\gamma((\varphi\{\beta\}\ \&\ \varphi\{\gamma\}) \rightarrow \beta = \gamma)$›
⟨*proof*⟩

AOT-theorem *nec−exist−!*: ‹$\forall\alpha(\varphi\{\alpha\} \rightarrow \Box\varphi\{\alpha\}) \rightarrow (\exists!\alpha\ \varphi\{\alpha\} \rightarrow \exists!\alpha\ \Box\varphi\{\alpha\})$›
⟨*proof*⟩

8.8 The Theory of Actuality and Descriptions

AOT-theorem *act−cond*: ‹$\mathcal{A}(\varphi \rightarrow \psi) \rightarrow (\mathcal{A}\varphi \rightarrow \mathcal{A}\psi)$›
⟨*proof*⟩

AOT-theorem *nec−imp−act*: ‹$\Box\varphi \rightarrow \mathcal{A}\varphi$›
⟨*proof*⟩

AOT-theorem *act−conj−act*:*1*: ‹$\mathcal{A}(\mathcal{A}\varphi \rightarrow \varphi)$›
⟨*proof*⟩

AOT-theorem *act−conj−act*:*2*: ‹$\mathcal{A}(\varphi \rightarrow \mathcal{A}\varphi)$›
⟨*proof*⟩

AOT-theorem *act−conj−act*:*3*: ‹$(\mathcal{A}\varphi \;\&\; \mathcal{A}\psi) \rightarrow \mathcal{A}(\varphi \;\&\; \psi)$›
⟨*proof*⟩

AOT-theorem *act−conj−act*:*4*: ‹$\mathcal{A}(\mathcal{A}\varphi \equiv \varphi)$›
⟨*proof*⟩

inductive *arbitrary-actualization* **for** φ **where**
‹*arbitrary-actualization* φ «$\mathcal{A}\varphi$»›
| ‹*arbitrary-actualization* φ «$\mathcal{A}\psi$»› **if** ‹*arbitrary-actualization* φ ψ›
declare *arbitrary-actualization.cases*[*AOT*]
arbitrary-actualization.induct[*AOT*]
arbitrary-actualization.simps[*AOT*]
arbitrary-actualization.intros[*AOT*]
syntax *arbitrary-actualization* :: ‹$\varphi' \Rightarrow \varphi' \Rightarrow$ *AOT-prop*›
(‹*ARBITRARY′-ACTUALIZATION′*(-,-′)›)

notepad
begin
⟨*proof*⟩
end

AOT-theorem *closure−act*:*1*:
assumes ‹*ARBITRARY-ACTUALIZATION*($\mathcal{A}\varphi \equiv \varphi$, ψ)›
shows ‹ψ›
⟨*proof*⟩

AOT-theorem *closure−act*:*2*: ‹$\forall\alpha\; \mathcal{A}(\mathcal{A}\varphi\{\alpha\} \equiv \varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *closure−act*:*3*: ‹$\mathcal{A}\forall\alpha\; \mathcal{A}(\mathcal{A}\varphi\{\alpha\} \equiv \varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *closure−act*:*4*: ‹$\mathcal{A}\forall\alpha_1 ... \forall\alpha_n\; \mathcal{A}(\mathcal{A}\varphi\{\alpha_1 ... \alpha_n\} \equiv \varphi\{\alpha_1 ... \alpha_n\})$›
⟨*proof*⟩

AOT-act-theorem *RA*[*1*]:
assumes ‹$\vdash \varphi$›
shows ‹$\vdash \mathcal{A}\varphi$›
— While this proof is rejected in PLM, we merely state it as modally-fragile rule, which addresses the concern in PLM.
⟨*proof*⟩
AOT-theorem *RA*[*2*]:
assumes ‹$\vdash_\Box \varphi$›
shows ‹$\vdash_\Box \mathcal{A}\varphi$›
— This rule is in fact a consequence of RN and does not require an appeal to the semantics itself.
⟨*proof*⟩
AOT-theorem *RA*[*3*]:

assumes ‹$\Gamma \vdash_\Box \varphi$›
shows ‹$\boldsymbol{\mathcal{A}}\Gamma \vdash_\Box \boldsymbol{\mathcal{A}}\varphi$›

This rule is only derivable from the semantics, but apparently no proof actually relies on it. If this turns out to be required, it is valid to derive it from the semantics just like RN, but we refrain from doing so, unless necessary.

⟨*proof*⟩

AOT-act-theorem *ANeg:1*: ‹$\neg\boldsymbol{\mathcal{A}}\varphi \equiv \neg\varphi$›
⟨*proof*⟩

AOT-act-theorem *ANeg:2*: ‹$\neg\boldsymbol{\mathcal{A}}\neg\varphi \equiv \varphi$›
⟨*proof*⟩

AOT-theorem *Act−Basic:1*: ‹$\boldsymbol{\mathcal{A}}\varphi \vee \boldsymbol{\mathcal{A}}\neg\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Basic:2*: ‹$\boldsymbol{\mathcal{A}}(\varphi \ \& \ \psi) \equiv (\boldsymbol{\mathcal{A}}\varphi \ \& \ \boldsymbol{\mathcal{A}}\psi)$›
⟨*proof*⟩

AOT-theorem *Act−Basic:3*: ‹$\boldsymbol{\mathcal{A}}(\varphi \equiv \psi) \equiv (\boldsymbol{\mathcal{A}}(\varphi \rightarrow \psi) \ \& \ \boldsymbol{\mathcal{A}}(\psi \rightarrow \varphi))$›
⟨*proof*⟩

AOT-theorem *Act−Basic:4*: ‹$(\boldsymbol{\mathcal{A}}(\varphi \rightarrow \psi) \ \& \ \boldsymbol{\mathcal{A}}(\psi \rightarrow \varphi)) \equiv (\boldsymbol{\mathcal{A}}\varphi \equiv \boldsymbol{\mathcal{A}}\psi)$›
⟨*proof*⟩

AOT-theorem *Act−Basic:5*: ‹$\boldsymbol{\mathcal{A}}(\varphi \equiv \psi) \equiv (\boldsymbol{\mathcal{A}}\varphi \equiv \boldsymbol{\mathcal{A}}\psi)$›
⟨*proof*⟩

AOT-theorem *Act−Basic:6*: ‹$\boldsymbol{\mathcal{A}}\varphi \equiv \Box\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Basic:7*: ‹$\boldsymbol{\mathcal{A}}\Box\varphi \rightarrow \Box\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Basic:8*: ‹$\Box\varphi \rightarrow \Box\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Basic:9*: ‹$\boldsymbol{\mathcal{A}}(\varphi \vee \psi) \equiv (\boldsymbol{\mathcal{A}}\varphi \vee \boldsymbol{\mathcal{A}}\psi)$›
⟨*proof*⟩

AOT-theorem *Act−Basic:10*: ‹$\boldsymbol{\mathcal{A}}\exists\,\alpha\ \varphi\{\alpha\} \equiv \exists\,\alpha\ \boldsymbol{\mathcal{A}}\varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *Act−Basic:11*:
‹$\boldsymbol{\mathcal{A}}\forall\,\alpha(\varphi\{\alpha\} \equiv \psi\{\alpha\}) \equiv \forall\,\alpha(\boldsymbol{\mathcal{A}}\varphi\{\alpha\} \equiv \boldsymbol{\mathcal{A}}\psi\{\alpha\})$›
⟨*proof*⟩

AOT-act-theorem *act−quant−uniq*:
‹$\forall\,\beta(\boldsymbol{\mathcal{A}}\varphi\{\beta\} \equiv \beta = \alpha) \equiv \forall\,\beta(\varphi\{\beta\} \equiv \beta = \alpha)$›
⟨*proof*⟩

AOT-act-theorem *fund−cont−desc*: ‹$x = \boldsymbol{\iota}x(\varphi\{x\}) \equiv \forall\,z(\varphi\{z\} \equiv z = x)$›
⟨*proof*⟩

AOT-act-theorem *hintikka*: ‹$x = \boldsymbol{\iota}x(\varphi\{x\}) \equiv (\varphi\{x\} \ \& \ \forall\,z\ (\varphi\{z\} \rightarrow z = x))$›
⟨*proof*⟩

locale *russell-axiom* =
fixes ψ
assumes *ψ-denotes-asm*: $[v \models \psi\{\kappa\}] \Longrightarrow [v \models \kappa\downarrow]$

begin
AOT-act-theorem *russell−axiom*:
‹$\psi\{\iota x\ \varphi\{x\}\} \equiv \exists\, x(\varphi\{x\}\ \&\ \forall\, z(\varphi\{z\} \rightarrow z = x)\ \&\ \psi\{x\})$›
⟨*proof*⟩
end

interpretation *russell−axiom[exe,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,2,1,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa\kappa'$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,2,1,2]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa'\kappa$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,2,2]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa\kappa$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,3,1,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa\kappa'\kappa''$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,3,1,2]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa'\kappa\kappa''$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,3,1,3]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa'\kappa''\kappa$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,3,2,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa\kappa\kappa'$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,3,2,2]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa\kappa'\kappa$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,3,2,3]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa'\kappa\kappa$»›
⟨*proof*⟩
interpretation *russell−axiom[exe,3,3]*: *russell-axiom* ‹$\lambda\ \kappa$. «$[\Pi]\kappa\kappa\kappa$»›
⟨*proof*⟩

interpretation *russell−axiom[enc,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,2,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa\kappa'[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,2,2]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa'\kappa[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,2,3]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa\kappa[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,3,1,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa\kappa'\kappa''[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,3,1,2]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa'\kappa\kappa''[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,3,1,3]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa'\kappa''\kappa[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,3,2,1]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa\kappa\kappa'[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,3,2,2]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa\kappa'\kappa[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,3,2,3]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa'\kappa\kappa[\Pi]$»›
⟨*proof*⟩
interpretation *russell−axiom[enc,3,3]*: *russell-axiom* ‹$\lambda\ \kappa$. «$\kappa\kappa\kappa[\Pi]$»›
⟨*proof*⟩

AOT-act-theorem !−*exists:1*: ‹$\iota x\ \varphi\{x\}\downarrow\ \equiv \exists !x\ \varphi\{x\}$›
⟨*proof*⟩

AOT-act-theorem !−*exists:2*: ‹$\exists\, y(y{=}\iota x\ \varphi\{x\}) \equiv \exists !x\ \varphi\{x\}$›
⟨*proof*⟩

AOT-act-theorem *y−in:1*: ‹$x = \iota x\ \varphi\{x\} \rightarrow \varphi\{x\}$›
⟨*proof*⟩

AOT-act-theorem *y−in:2*: ‹$z = \iota x\ \varphi\{x\} \rightarrow \varphi\{z\}$› ⟨*proof*⟩

AOT-act-theorem *y−in:3*: ‹$\iota x\ \varphi\{x\}\downarrow \rightarrow \varphi\{\iota x\ \varphi\{x\}\}$›
⟨*proof*⟩

AOT-act-theorem *y−in:4*: ‹$\exists\, y\ (y = \iota x\ \varphi\{x\}) \rightarrow \varphi\{\iota x\ \varphi\{x\}\}$›
⟨*proof*⟩

AOT-theorem *act−quant−nec*:
‹$\forall\, \beta\ (\mathcal{A}\varphi\{\beta\} \equiv \beta = \alpha) \equiv \forall\, \beta(\mathcal{A}\mathcal{A}\varphi\{\beta\} \equiv \beta = \alpha)$›
⟨*proof*⟩

AOT-theorem *equi−desc−descA:1*: ‹$x = \iota x\ \varphi\{x\} \equiv x = \iota x(\mathcal{A}\varphi\{x\})$›
⟨*proof*⟩

AOT-theorem *equi−desc−descA:2*: ‹$\iota x\ \varphi\{x\}\downarrow \rightarrow \iota x\ \varphi\{x\} = \iota x(\mathcal{A}\varphi\{x\})$›
⟨*proof*⟩

AOT-theorem *nec−hintikka−scheme*:
‹$x = \iota x\ \varphi\{x\} \equiv \mathcal{A}\varphi\{x\}\ \&\ \forall\, z(\mathcal{A}\varphi\{z\} \rightarrow z = x)$›
⟨*proof*⟩

AOT-theorem *equiv−desc−eq:1*:
‹$\mathcal{A}\forall\, x(\varphi\{x\} \equiv \psi\{x\}) \rightarrow \forall\, x\ (x = \iota x\ \varphi\{x\} \equiv x = \iota x\ \psi\{x\})$›
⟨*proof*⟩

AOT-theorem *equiv−desc−eq:2*:
‹$\iota x\ \varphi\{x\}\downarrow\ \&\ \mathcal{A}\forall\, x(\varphi\{x\} \equiv \psi\{x\}) \rightarrow \iota x\ \varphi\{x\} = \iota x\ \psi\{x\}$›
⟨*proof*⟩

AOT-theorem *equiv−desc−eq:3*:
‹$\iota x\ \varphi\{x\}\downarrow\ \&\ \Box\forall\, x(\varphi\{x\} \equiv \psi\{x\}) \rightarrow \iota x\ \varphi\{x\} = \iota x\ \psi\{x\}$›
⟨*proof*⟩

AOT-theorem *equiv−desc−eq:4*: ‹$\iota x\ \varphi\{x\}\downarrow \rightarrow \Box\iota x\ \varphi\{x\}\downarrow$›
⟨*proof*⟩

AOT-theorem *equiv−desc−eq:5*: ‹$\iota x\ \varphi\{x\}\downarrow \rightarrow \exists\, y\ \Box(y = \iota x\ \varphi\{x\})$›
⟨*proof*⟩

AOT-act-theorem *equiv−desc−eq2:1*:
‹$\forall\, x\ (\varphi\{x\} \equiv \psi\{x\}) \rightarrow \forall\, x\ (x = \iota x\ \varphi\{x\} \equiv x = \iota x\ \psi\{x\})$›
⟨*proof*⟩

AOT-act-theorem *equiv−desc−eq2:2*:
‹$\iota x\ \varphi\{x\}\downarrow\ \&\ \forall\, x\ (\varphi\{x\} \equiv \psi\{x\}) \rightarrow \iota x\ \varphi\{x\} = \iota x\ \psi\{x\}$›
⟨*proof*⟩

context *russell-axiom*
begin
AOT-theorem *nec−russell−axiom*:
‹$\psi\{\iota x\ \varphi\{x\}\} \equiv \exists\, x(\mathcal{A}\varphi\{x\}\ \&\ \forall\, z(\mathcal{A}\varphi\{z\} \rightarrow z = x)\ \&\ \psi\{x\})$›
⟨*proof*⟩
end

AOT-theorem *actual−desc:1*: ‹$\iota x\ \varphi\{x\}\downarrow \equiv \exists!x\ \mathcal{A}\varphi\{x\}$›
⟨*proof*⟩

AOT-theorem *actual−desc:2*: ‹$x = \iota x\ \varphi\{x\} \rightarrow \mathcal{A}\varphi\{x\}$›
⟨*proof*⟩

AOT-theorem *actual−desc:3*: ‹$z = \iota x\ \varphi\{x\} \rightarrow \mathcal{A}\varphi\{z\}$›
⟨*proof*⟩

AOT-theorem *actual−desc:4*: ‹$\iota x\ \varphi\{x\}\downarrow \rightarrow \mathcal{A}\varphi\{\iota x\ \varphi\{x\}\}$›
⟨*proof*⟩

AOT-theorem *actual−desc:5*: ‹$\iota x\ \varphi\{x\} = \iota x\ \psi\{x\} \rightarrow \mathcal{A}\forall x(\varphi\{x\} \equiv \psi\{x\})$›
⟨*proof*⟩

AOT-theorem *!box−desc:1*: ‹$\exists !x\ \Box\varphi\{x\} \rightarrow \forall y\ (y = \iota x\ \varphi\{x\} \rightarrow \varphi\{y\})$›
⟨*proof*⟩

AOT-theorem *!box−desc:2*:
‹$\forall x\ (\varphi\{x\} \rightarrow \Box\varphi\{x\}) \rightarrow (\exists !x\ \varphi\{x\} \rightarrow \forall y\ (y = \iota x\ \varphi\{x\} \rightarrow \varphi\{y\}))$›
⟨*proof*⟩

AOT-theorem *dr−alphabetic−thm*: ‹$\iota\nu\ \varphi\{\nu\}\downarrow \rightarrow \iota\nu\ \varphi\{\nu\} = \iota\mu\ \varphi\{\mu\}$›
⟨*proof*⟩

8.9 The Theory of Necessity

AOT-theorem *RM:1[prem]*:
assumes ‹$\Gamma \vdash_\Box \varphi \rightarrow \psi$›
shows ‹$\Box\Gamma \vdash_\Box \Box\varphi \rightarrow \Box\psi$›
⟨*proof*⟩

AOT-theorem *RM:1*:
assumes ‹$\vdash_\Box \varphi \rightarrow \psi$›
shows ‹$\vdash_\Box \Box\varphi \rightarrow \Box\psi$›
⟨*proof*⟩

lemmas *RM* = *RM:1*

AOT-theorem *RM:2[prem]*:
assumes ‹$\Gamma \vdash_\Box \varphi \rightarrow \psi$›
shows ‹$\Box\Gamma \vdash_\Box \Diamond\varphi \rightarrow \Diamond\psi$›
⟨*proof*⟩

AOT-theorem *RM:2*:
assumes ‹$\vdash_\Box \varphi \rightarrow \psi$›
shows ‹$\vdash_\Box \Diamond\varphi \rightarrow \Diamond\psi$›
⟨*proof*⟩

lemmas *RM◇* = *RM:2*

AOT-theorem *RM:3[prem]*:
assumes ‹$\Gamma \vdash_\Box \varphi \equiv \psi$›
shows ‹$\Box\Gamma \vdash_\Box \Box\varphi \equiv \Box\psi$›
⟨*proof*⟩

AOT-theorem *RM:3*:
assumes ‹$\vdash_\Box \varphi \equiv \psi$›
shows ‹$\vdash_\Box \Box\varphi \equiv \Box\psi$›
⟨*proof*⟩

lemmas *RE* = *RM:3*

AOT-theorem *RM:4[prem]*:
assumes ‹$\Gamma \vdash_\Box \varphi \equiv \psi$›
shows ‹$\Box\Gamma \vdash_\Box \Diamond\varphi \equiv \Diamond\psi$›
⟨*proof*⟩

AOT-theorem *RM:4*:
assumes ‹$\vdash_\Box \varphi \equiv \psi$›
shows ‹$\vdash_\Box \Diamond\varphi \equiv \Diamond\psi$›
⟨*proof*⟩

lemmas *RE*$\Diamond$ = *RM:4*

AOT-theorem *KBasic:1*: ‹$\Box\varphi \rightarrow \Box(\psi \rightarrow \varphi)$›
⟨*proof*⟩

AOT-theorem *KBasic:2*: ‹$\Box\neg\varphi \rightarrow \Box(\varphi \rightarrow \psi)$›
⟨*proof*⟩

AOT-theorem *KBasic:3*: ‹$\Box(\varphi \mathbin{\&} \psi) \equiv (\Box\varphi \mathbin{\&} \Box\psi)$›
⟨*proof*⟩

AOT-theorem *KBasic:4*: ‹$\Box(\varphi \equiv \psi) \equiv (\Box(\varphi \rightarrow \psi) \mathbin{\&} \Box(\psi \rightarrow \varphi))$›
⟨*proof*⟩

AOT-theorem *KBasic:5*: ‹$(\Box(\varphi \rightarrow \psi) \mathbin{\&} \Box(\psi \rightarrow \varphi)) \rightarrow (\Box\varphi \equiv \Box\psi)$›
⟨*proof*⟩

AOT-theorem *KBasic:6*: ‹$\Box(\varphi \equiv \psi) \rightarrow (\Box\varphi \equiv \Box\psi)$›
⟨*proof*⟩
AOT-theorem *KBasic:7*: ‹$((\Box\varphi \mathbin{\&} \Box\psi) \vee (\Box\neg\varphi \mathbin{\&} \Box\neg\psi)) \rightarrow \Box(\varphi \equiv \psi)$›
⟨*proof*⟩

AOT-theorem *KBasic:8*: ‹$\Box(\varphi \mathbin{\&} \psi) \rightarrow \Box(\varphi \equiv \psi)$›
⟨*proof*⟩
AOT-theorem *KBasic:9*: ‹$\Box(\neg\varphi \mathbin{\&} \neg\psi) \rightarrow \Box(\varphi \equiv \psi)$›
⟨*proof*⟩
AOT-theorem *KBasic:10*: ‹$\Box\varphi \equiv \Box\neg\neg\varphi$›
⟨*proof*⟩
AOT-theorem *KBasic:11*: ‹$\neg\Box\varphi \equiv \Diamond\neg\varphi$›
⟨*proof*⟩
AOT-theorem *KBasic:12*: ‹$\Box\varphi \equiv \neg\Diamond\neg\varphi$›
⟨*proof*⟩
AOT-theorem *KBasic:13*: ‹$\Box(\varphi \rightarrow \psi) \rightarrow (\Diamond\varphi \rightarrow \Diamond\psi)$›
⟨*proof*⟩
lemmas *K*$\Diamond$ = *KBasic:13*
AOT-theorem *KBasic:14*: ‹$\Diamond\Box\varphi \equiv \neg\Box\Diamond\neg\varphi$›
⟨*proof*⟩
AOT-theorem *KBasic:15*: ‹$(\Box\varphi \vee \Box\psi) \rightarrow \Box(\varphi \vee \psi)$›
⟨*proof*⟩

AOT-theorem *KBasic:16*: ‹$(\Box\varphi \mathbin{\&} \Diamond\psi) \rightarrow \Diamond(\varphi \mathbin{\&} \psi)$›
⟨*proof*⟩

AOT-theorem *rule−sub−lem:1:a*:
assumes ‹$\vdash_\Box \Box(\psi \equiv \chi)$›
shows ‹$\vdash_\Box \neg\psi \equiv \neg\chi$›
⟨*proof*⟩

AOT-theorem *rule−sub−lem:1:b*:
assumes ‹$\vdash_\Box \Box(\psi \equiv \chi)$›
shows ‹$\vdash_\Box (\psi \rightarrow \Theta) \equiv (\chi \rightarrow \Theta)$›
⟨*proof*⟩

AOT-theorem *rule−sub−lem:1:c*:
assumes ‹$\vdash_\Box \Box(\psi \equiv \chi)$›
shows ‹$\vdash_\Box (\Theta \rightarrow \psi) \equiv (\Theta \rightarrow \chi)$›
⟨*proof*⟩

AOT-theorem *rule−sub−lem:1:d*:
assumes ‹*for arbitrary* α: $\vdash_\Box \Box(\psi\{\alpha\} \equiv \chi\{\alpha\})$›
shows ‹$\vdash_\Box \forall\alpha\ \psi\{\alpha\} \equiv \forall\alpha\ \chi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *rule−sub−lem:1:e*:
assumes ‹$\vdash_\Box \Box(\psi \equiv \chi)$›
shows ‹$\vdash_\Box [\lambda\ \psi] \equiv [\lambda\ \chi]$›
⟨*proof*⟩

AOT-theorem *rule−sub−lem:1:f*:
assumes ‹$\vdash_\Box \Box(\psi \equiv \chi)$›
shows ‹$\vdash_\Box \boldsymbol{\mathcal{A}}\psi \equiv \boldsymbol{\mathcal{A}}\chi$›
⟨*proof*⟩

AOT-theorem *rule−sub−lem:1:g*:
assumes ‹$\vdash_\Box \Box(\psi \equiv \chi)$›
shows ‹$\vdash_\Box \Box\psi \equiv \Box\chi$›
⟨*proof*⟩

Note that instead of deriving *rule−sub−lem:2*, *rule−sub−lem:3*, *rule−sub−lem:4*, and *rule−sub−nec*, we construct substitution methods instead.

class *AOT-subst* =
fixes *AOT-subst* :: ($'a \Rightarrow$ o) $\Rightarrow$ *bool*
and *AOT-subst-cond* :: $'a \Rightarrow {}'a \Rightarrow$ *bool*
assumes *AOT-subst*:
AOT-subst $\varphi \Longrightarrow$ *AOT-subst-cond* $\psi\ \chi \Longrightarrow [v \models$ «$\varphi\ \psi$» $\equiv$ «$\varphi\ \chi$»$]$

named-theorems *AOT-substI*

instantiation o :: *AOT-subst*
begin

inductive *AOT-subst*-o **where**
AOT-subst-o-*id*[*AOT-substI*]:
‹*AOT-subst*-o ($\lambda\varphi.\ \varphi$)›
| *AOT-subst*-o-*const*[*AOT-substI*]:
‹*AOT-subst*-o ($\lambda\varphi.\ \psi$)›
| *AOT-subst*-o-*not*[*AOT-substI*]:
‹*AOT-subst*-o $\Theta \Longrightarrow$ *AOT-subst*-o ($\lambda\ \varphi.$ «$\neg\Theta\{\varphi\}$»)›
| *AOT-subst*-o-*imp*[*AOT-substI*]:
‹*AOT-subst*-o $\Theta \Longrightarrow$ *AOT-subst*-o $\Xi \Longrightarrow$ *AOT-subst*-o ($\lambda\ \varphi.$ «$\Theta\{\varphi\} \rightarrow \Xi\{\varphi\}$»)›
| *AOT-subst*-o-*lambda0*[*AOT-substI*]:
‹*AOT-subst*-o $\Theta \Longrightarrow$ *AOT-subst*-o ($\lambda\ \varphi.$ (*AOT-lambda0* ($\Theta\ \varphi$)))›
| *AOT-subst*-o-*act*[*AOT-substI*]:
‹*AOT-subst*-o $\Theta \Longrightarrow$ *AOT-subst*-o ($\lambda\ \varphi.$ «$\boldsymbol{\mathcal{A}}\Theta\{\varphi\}$»)›
| *AOT-subst*-o-*box*[*AOT-substI*]:
‹*AOT-subst*-o $\Theta \Longrightarrow$ *AOT-subst*-o ($\lambda\ \varphi.$ «$\Box\Theta\{\varphi\}$»)›
| *AOT-subst*-o-*by-def*[*AOT-substI*]:
‹($\bigwedge\ \psi$. *AOT-model-equiv-def* ($\Theta\ \psi$) ($\Xi\ \psi$)) $\Longrightarrow$
AOT-subst-o $\Xi \Longrightarrow$ *AOT-subst*-o Θ›

definition *AOT-subst-cond*-o **where**
‹*AOT-subst-cond*-o $\equiv \lambda\ \psi\ \chi\ .\ \forall\ v\ .\ [v \models \psi \equiv \chi]$›

instance
⟨*proof*⟩
end

instantiation *fun* :: (*AOT-Term-id-2*, *AOT-subst*) *AOT-subst*
begin

definition *AOT-subst-cond-fun* :: ‹('a ⇒ 'b) ⇒ ('a ⇒ 'b) ⇒ *bool*› **where**
‹*AOT-subst-cond-fun* ≡ λ φ ψ . ∀ α . *AOT-subst-cond* (φ (*AOT-term-of-var* α))
(ψ (*AOT-term-of-var* α))›

inductive *AOT-subst-fun* :: ‹(('a ⇒ 'b) ⇒ o) ⇒ *bool*› **where**
AOT-subst-fun-const[*AOT-substI*]:
‹*AOT-subst-fun* (λφ. ψ)›
| *AOT-subst-fun-id*[*AOT-substI*]:
‹*AOT-subst* Ψ ⟹ *AOT-subst-fun* (λφ. Ψ (φ (*AOT-term-of-var* α)))›
| *AOT-subst-fun-all*[*AOT-substI*]:
‹*AOT-subst* Ψ ⟹ (⋀ α . *AOT-subst-fun* (Θ (*AOT-term-of-var* α))) ⟹
AOT-subst-fun (λφ :: 'a ⇒ 'b. Ψ «∀ α «Θ (α::'a) φ»»)›
| *AOT-subst-fun-not*[*AOT-substI*]:
‹*AOT-subst* Ψ ⟹ *AOT-subst-fun* (λφ. «¬«Ψ φ»»)›
| *AOT-subst-fun-imp*[*AOT-substI*]:
‹*AOT-subst* Ψ ⟹ *AOT-subst* Θ ⟹ *AOT-subst-fun* (λφ. ««Ψ φ» → «Θ φ»»)›
| *AOT-subst-fun-lambda0*[*AOT-substI*]:
‹*AOT-subst* Θ ⟹ *AOT-subst-fun* (λ φ. (*AOT-lambda0* (Θ φ)))›
| *AOT-subst-fun-act*[*AOT-substI*]:
‹*AOT-subst* Θ ⟹ *AOT-subst-fun* (λ φ. «$\boldsymbol{\mathcal{A}}$«Θ φ»»)›
| *AOT-subst-fun-box*[*AOT-substI*]:
‹*AOT-subst* Θ ⟹ *AOT-subst-fun* (λ φ. «□«Θ φ»»)›
| *AOT-subst-fun-def*[*AOT-substI*]:
‹(⋀ φ . *AOT-model-equiv-def* (Θ φ) (Ψ φ)) ⟹
AOT-subst-fun Ψ ⟹ *AOT-subst-fun* Θ›

instance ⟨*proof*⟩
end

⟨*ML*⟩

AOT-theorem *rule−sub−remark*:*1*[*1*]:
assumes ‹$\vdash_\Box$ *A*!*x* ≡ ¬◇*E*!*x*› **and** ‹¬*A*!*x*›
shows ‹¬¬◇*E*!*x*›
⟨*proof*⟩

AOT-theorem *rule−sub−remark*:*1*[*2*]:
assumes ‹$\vdash_\Box$ *A*!*x* ≡ ¬◇*E*!*x*› **and** ‹¬¬◇*E*!*x*›
shows ‹¬*A*!*x*›
⟨*proof*⟩

AOT-theorem *rule−sub−remark*:*2*[*1*]:
assumes ‹$\vdash_\Box$ [*R*]*xy* ≡ ([*R*]*xy* & ([*Q*]*a* ∨ ¬[*Q*]*a*))›
and ‹*p* → [*R*]*xy*›
shows ‹*p* → [*R*]*xy* & ([*Q*]*a* ∨ ¬[*Q*]*a*)›
⟨*proof*⟩

AOT-theorem *rule−sub−remark*:*2*[*2*]:
assumes ‹$\vdash_\Box$ [*R*]*xy* ≡ ([*R*]*xy* & ([*Q*]*a* ∨ ¬[*Q*]*a*))›
and ‹*p* → [*R*]*xy* & ([*Q*]*a* ∨ ¬[*Q*]*a*)›
shows ‹*p* → [*R*]*xy*›
⟨*proof*⟩

AOT-theorem *rule−sub−remark*:*3*[*1*]:
assumes ‹*for arbitrary x*: $\vdash_\Box$ *A*!*x* ≡ ¬◇*E*!*x*›
and ‹∃ *x* *A*!*x*›
shows ‹∃ *x* ¬◇*E*!*x*›
⟨*proof*⟩

AOT-theorem *rule−sub−remark*:*3*[*2*]:
assumes ‹*for arbitrary x*: $\vdash_\Box$ *A*!*x* ≡ ¬◇*E*!*x*›
and ‹∃ *x* ¬◇*E*!*x*›
shows ‹∃ *x* *A*!*x*›

⟨*proof*⟩

AOT-theorem *rule−sub−remark:4[1]*:
assumes ‹$\vdash_\Box$ ¬¬[P]x ≡ [P]x› **and** ‹$\mathcal{A}$¬¬[P]x›
shows ‹$\mathcal{A}$[P]x›
⟨*proof*⟩

AOT-theorem *rule−sub−remark:4[2]*:
assumes ‹$\vdash_\Box$ ¬¬[P]x ≡ [P]x› **and** ‹$\mathcal{A}$[P]x›
shows ‹$\mathcal{A}$¬¬[P]x›
⟨*proof*⟩

AOT-theorem *rule−sub−remark:5[1]*:
assumes ‹$\vdash_\Box$ (φ → ψ) ≡ (¬ψ → ¬φ)› **and** ‹□(φ → ψ)›
shows ‹□(¬ψ → ¬φ)›
⟨*proof*⟩

AOT-theorem *rule−sub−remark:5[2]*:
assumes ‹$\vdash_\Box$ (φ → ψ) ≡ (¬ψ → ¬φ)› **and** ‹□(¬ψ → ¬φ)›
shows ‹□(φ → ψ)›
⟨*proof*⟩

AOT-theorem *rule−sub−remark:6[1]*:
assumes ‹$\vdash_\Box$ ψ ≡ χ› **and** ‹□(φ → ψ)›
shows ‹□(φ → χ)›
⟨*proof*⟩

AOT-theorem *rule−sub−remark:6[2]*:
assumes ‹$\vdash_\Box$ ψ ≡ χ› **and** ‹□(φ → χ)›
shows ‹□(φ → ψ)›
⟨*proof*⟩

AOT-theorem *rule−sub−remark:7[1]*:
assumes ‹$\vdash_\Box$ φ ≡ ¬¬φ› **and** ‹□(φ → φ)›
shows ‹□(¬¬φ → φ)›
⟨*proof*⟩

AOT-theorem *rule−sub−remark:7[2]*:
assumes ‹$\vdash_\Box$ φ ≡ ¬¬φ› **and** ‹□(¬¬φ → φ)›
shows ‹□(φ → φ)›
⟨*proof*⟩

AOT-theorem *KBasic2:1*: ‹□¬φ ≡ ¬◇φ›
⟨*proof*⟩

AOT-theorem *KBasic2:2*: ‹◇(φ ∨ ψ) ≡ (◇φ ∨ ◇ψ)›
⟨*proof*⟩

AOT-theorem *KBasic2:3*: ‹◇(φ & ψ) → (◇φ & ◇ψ)›
⟨*proof*⟩

AOT-theorem *KBasic2:4*: ‹◇(φ → ψ) ≡ (□φ → ◇ψ)›
⟨*proof*⟩

AOT-theorem *KBasic2:5*: ‹◇◇φ ≡ ¬□□¬φ›
⟨*proof*⟩

AOT-theorem *KBasic2:6*: ‹□(φ ∨ ψ) → (□φ ∨ ◇ψ)›
⟨*proof*⟩

AOT-theorem *KBasic2:7*: ‹(□(φ ∨ ψ) & ◇¬φ) → ◇ψ›
⟨*proof*⟩

AOT-theorem *T−S5−fund:1*: ‹$\varphi \rightarrow \Diamond\varphi$›
⟨*proof*⟩
lemmas *T*$\Diamond$ = *T−S5−fund:1*

AOT-theorem *T−S5−fund:2*: ‹$\Diamond\Box\varphi \rightarrow \Box\varphi$›
⟨*proof*⟩
lemmas *5*$\Diamond$ = *T−S5−fund:2*

AOT-theorem *Act−Sub:1*: ‹$\boldsymbol{\mathcal{A}}\varphi \equiv \neg\boldsymbol{\mathcal{A}}\neg\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Sub:2*: ‹$\Diamond\varphi \equiv \boldsymbol{\mathcal{A}}\Diamond\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Sub:3*: ‹$\boldsymbol{\mathcal{A}}\varphi \rightarrow \Diamond\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Sub:4*: ‹$\boldsymbol{\mathcal{A}}\varphi \equiv \Diamond\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩

AOT-theorem *Act−Sub:5*: ‹$\Diamond\boldsymbol{\mathcal{A}}\varphi \rightarrow \boldsymbol{\mathcal{A}}\Diamond\varphi$›
⟨*proof*⟩

AOT-theorem *S5Basic:1*: ‹$\Diamond\varphi \equiv \Box\Diamond\varphi$›
⟨*proof*⟩

AOT-theorem *S5Basic:2*: ‹$\Box\varphi \equiv \Diamond\Box\varphi$›
⟨*proof*⟩

AOT-theorem *S5Basic:3*: ‹$\varphi \rightarrow \Box\Diamond\varphi$›
⟨*proof*⟩
lemmas *B* = *S5Basic:3*

AOT-theorem *S5Basic:4*: ‹$\Diamond\Box\varphi \rightarrow \varphi$›
⟨*proof*⟩
lemmas *B*$\Diamond$ = *S5Basic:4*

AOT-theorem *S5Basic:5*: ‹$\Box\varphi \rightarrow \Box\Box\varphi$›
⟨*proof*⟩
lemmas *4* = *S5Basic:5*

AOT-theorem *S5Basic:6*: ‹$\Box\varphi \equiv \Box\Box\varphi$›
⟨*proof*⟩

AOT-theorem *S5Basic:7*: ‹$\Diamond\Diamond\varphi \rightarrow \Diamond\varphi$›
⟨*proof*⟩

lemmas *4*$\Diamond$ = *S5Basic:7*

AOT-theorem *S5Basic:8*: ‹$\Diamond\Diamond\varphi \equiv \Diamond\varphi$›
⟨*proof*⟩

AOT-theorem *S5Basic:9*: ‹$\Box(\varphi \vee \Box\psi) \equiv (\Box\varphi \vee \Box\psi)$›
⟨*proof*⟩

AOT-theorem *S5Basic:10*: ‹$\Box(\varphi \vee \Diamond\psi) \equiv (\Box\varphi \vee \Diamond\psi)$›
⟨*proof*⟩

AOT-theorem *S5Basic:11*: ‹$\Diamond(\varphi \ \& \ \Diamond\psi) \equiv (\Diamond\varphi \ \& \ \Diamond\psi)$›
⟨*proof*⟩

AOT-theorem *S5Basic:12*: ‹$\Diamond(\varphi \ \& \ \Box\psi) \equiv (\Diamond\varphi \ \& \ \Box\psi)$›

⟨*proof*⟩

AOT-theorem *S5Basic*:*13*: ‹$\Box(\varphi \rightarrow \Box\psi) \equiv \Box(\Diamond\varphi \rightarrow \psi)$›
⟨*proof*⟩

AOT-theorem *derived−S5−rules*:*1*:
assumes ‹$\Gamma \vdash_{\Box} \Diamond\varphi \rightarrow \psi$›
shows ‹$\Box\Gamma \vdash_{\Box} \varphi \rightarrow \Box\psi$›
⟨*proof*⟩

AOT-theorem *derived−S5−rules*:*2*:
assumes ‹$\Gamma \vdash_{\Box} \varphi \rightarrow \Box\psi$›
shows ‹$\Box\Gamma \vdash_{\Box} \Diamond\varphi \rightarrow \psi$›
⟨*proof*⟩

AOT-theorem *BFs*:*1*: ‹$\forall \alpha\ \Box\varphi\{\alpha\} \rightarrow \Box\forall \alpha\ \varphi\{\alpha\}$›
⟨*proof*⟩
lemmas *BF* = *BFs*:*1*

AOT-theorem *BFs*:*2*: ‹$\Box\forall \alpha\ \varphi\{\alpha\} \rightarrow \forall \alpha\ \Box\varphi\{\alpha\}$›
⟨*proof*⟩
lemmas *CBF* = *BFs*:*2*

AOT-theorem *BFs*:*3*: ‹$\Diamond\exists \alpha\ \varphi\{\alpha\} \rightarrow \exists \alpha\ \Diamond\varphi\{\alpha\}$›
⟨*proof*⟩
lemmas *BF*◇ = *BFs*:*3*

AOT-theorem *BFs*:*4*: ‹$\exists \alpha\ \Diamond\varphi\{\alpha\} \rightarrow \Diamond\exists \alpha\ \varphi\{\alpha\}$›
⟨*proof*⟩
lemmas *CBF*◇ = *BFs*:*4*

AOT-theorem *sign−S5−thm*:*1*: ‹$\exists \alpha\ \Box\varphi\{\alpha\} \rightarrow \Box\exists \alpha\ \varphi\{\alpha\}$›
⟨*proof*⟩
lemmas *Buridan* = *sign−S5−thm*:*1*

AOT-theorem *sign−S5−thm*:*2*: ‹$\Diamond\forall \alpha\ \varphi\{\alpha\} \rightarrow \forall \alpha\ \Diamond\varphi\{\alpha\}$›
⟨*proof*⟩
lemmas *Buridan*◇ = *sign−S5−thm*:*2*

AOT-theorem *sign−S5−thm*:*3*:
‹$\Diamond\exists \alpha\ (\varphi\{\alpha\}\ \&\ \psi\{\alpha\}) \rightarrow \Diamond(\exists \alpha\ \varphi\{\alpha\}\ \&\ \exists \alpha\ \psi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *sign−S5−thm*:*4*: ‹$\Diamond\exists \alpha\ (\varphi\{\alpha\}\ \&\ \psi\{\alpha\}) \rightarrow \Diamond\exists \alpha\ \varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *sign−S5−thm*:*5*:
‹$(\Box\forall \alpha\ (\varphi\{\alpha\} \rightarrow \psi\{\alpha\})\ \&\ \Box\forall \alpha\ (\psi\{\alpha\} \rightarrow \chi\{\alpha\})) \rightarrow \Box\forall \alpha\ (\varphi\{\alpha\} \rightarrow \chi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *sign−S5−thm*:*6*:
‹$(\Box\forall \alpha\ (\varphi\{\alpha\} \equiv \psi\{\alpha\})\ \&\ \Box\forall \alpha(\psi\{\alpha\} \equiv \chi\{\alpha\})) \rightarrow \Box\forall \alpha(\varphi\{\alpha\} \equiv \chi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *exist−nec2*:*1*: ‹$\Diamond\tau\downarrow \rightarrow \tau\downarrow$›
⟨*proof*⟩

AOT-theorem *exists−nec2*:*2*: ‹$\Diamond\tau\downarrow \equiv \Box\tau\downarrow$›
⟨*proof*⟩

AOT-theorem *exists−nec2*:*3*: ‹$\neg\tau\downarrow \rightarrow \Box\neg\tau\downarrow$›
⟨*proof*⟩

AOT-theorem *exists−nec2:4*: ‹$\Diamond\neg\tau\downarrow \equiv \Box\neg\tau\downarrow$›
⟨*proof*⟩

AOT-theorem *id−nec2:1*: ‹$\Diamond\alpha = \beta \rightarrow \alpha = \beta$›
⟨*proof*⟩

AOT-theorem *id−nec2:2*: ‹$\alpha \neq \beta \rightarrow \Box\alpha \neq \beta$›
⟨*proof*⟩

AOT-theorem *id−nec2:3*: ‹$\Diamond\alpha \neq \beta \rightarrow \alpha \neq \beta$›
⟨*proof*⟩

AOT-theorem *id−nec2:4*: ‹$\Diamond\alpha = \beta \rightarrow \Box\alpha = \beta$›
⟨*proof*⟩

AOT-theorem *id−nec2:5*: ‹$\Diamond\alpha \neq \beta \rightarrow \Box\alpha \neq \beta$›
⟨*proof*⟩

AOT-theorem *sc−eq−box−box:1*: ‹$\Box(\varphi \rightarrow \Box\varphi) \equiv (\Diamond\varphi \rightarrow \Box\varphi)$›
⟨*proof*⟩

AOT-theorem *sc−eq−box−box:2*: ‹$(\Box(\varphi \rightarrow \Box\varphi) \vee (\Diamond\varphi \rightarrow \Box\varphi)) \rightarrow (\Diamond\varphi \equiv \Box\varphi)$›
⟨*proof*⟩

AOT-theorem *sc−eq−box−box:3*: ‹$\Box(\varphi \rightarrow \Box\varphi) \rightarrow (\neg\Box\varphi \equiv \Box\neg\varphi)$›
⟨*proof*⟩

AOT-theorem *sc−eq−box−box:4*:
‹$(\Box(\varphi \rightarrow \Box\varphi)\ \&\ \Box(\psi \rightarrow \Box\psi)) \rightarrow ((\Box\varphi \equiv \Box\psi) \rightarrow \Box(\varphi \equiv \psi))$›
⟨*proof*⟩

AOT-theorem *sc−eq−box−box:5*:
‹$(\Box(\varphi \rightarrow \Box\varphi)\ \&\ \Box(\psi \rightarrow \Box\psi)) \rightarrow \Box((\varphi \equiv \psi) \rightarrow \Box(\varphi \equiv \psi))$›
⟨*proof*⟩

AOT-theorem *sc−eq−box−box:6*: ‹$\Box(\varphi \rightarrow \Box\varphi) \rightarrow ((\varphi \rightarrow \Box\psi) \rightarrow \Box(\varphi \rightarrow \psi))$›
⟨*proof*⟩

AOT-theorem *sc−eq−box−box:7*: ‹$\Box(\varphi \rightarrow \Box\varphi) \rightarrow ((\varphi \rightarrow \boldsymbol{\mathcal{A}}\psi) \rightarrow \boldsymbol{\mathcal{A}}(\varphi \rightarrow \psi))$›
⟨*proof*⟩

AOT-theorem *sc−eq−fur:1*: ‹$\Diamond\boldsymbol{\mathcal{A}}\varphi \equiv \Box\boldsymbol{\mathcal{A}}\varphi$›
⟨*proof*⟩

AOT-theorem *sc−eq−fur:2*: ‹$\Box(\varphi \rightarrow \Box\varphi) \rightarrow (\boldsymbol{\mathcal{A}}\varphi \equiv \varphi)$›
⟨*proof*⟩

AOT-theorem *sc−eq−fur:3*:
‹$\Box\forall x\ (\varphi\{x\} \rightarrow \Box\varphi\{x\}) \rightarrow (\exists!x\ \varphi\{x\} \rightarrow \iota x\ \varphi\{x\}\downarrow)$›
⟨*proof*⟩

AOT-theorem *sc−eq−fur:4*:
‹$\Box\forall x\ (\varphi\{x\} \rightarrow \Box\varphi\{x\}) \rightarrow (x = \iota x\ \varphi\{x\} \equiv (\varphi\{x\}\ \&\ \forall z\ (\varphi\{z\} \rightarrow z = x)))$›
⟨*proof*⟩

AOT-theorem *id−act:1*: ‹$\alpha = \beta \equiv \boldsymbol{\mathcal{A}}\alpha = \beta$›
⟨*proof*⟩

AOT-theorem *id−act:2*: ‹$\alpha \neq \beta \equiv \boldsymbol{\mathcal{A}}\alpha \neq \beta$›
⟨*proof*⟩

AOT-theorem *A−Exists:1*: ‹$\boldsymbol{\mathcal{A}}\exists!\alpha\ \varphi\{\alpha\} \equiv \exists!\alpha\ \boldsymbol{\mathcal{A}}\varphi\{\alpha\}$›
⟨*proof*⟩

AOT-theorem *A−Exists:2*: ‹$\iota x\ \varphi\{x\}\downarrow \equiv \mathcal{A}\exists!x\ \varphi\{x\}$›
⟨*proof*⟩

AOT-theorem *id−act−desc:1*: ‹$\iota x\ (x = y)\downarrow$›
⟨*proof*⟩

AOT-theorem *id−act−desc:2*: ‹$y = \iota x\ (x = y)$›
⟨*proof*⟩

AOT-theorem *pre−en−eq:1[1]*: ‹$x_1[F] \to \Box x_1[F]$›
⟨*proof*⟩

AOT-theorem *pre−en−eq:1[2]*: ‹$x_1x_2[F] \to \Box x_1x_2[F]$›
⟨*proof*⟩

AOT-theorem *pre−en−eq:1[3]*: ‹$x_1x_2x_3[F] \to \Box x_1x_2x_3[F]$›
⟨*proof*⟩

AOT-theorem *pre−en−eq:1[4]*: ‹$x_1x_2x_3x_4[F] \to \Box x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *pre−en−eq:2[1]*: ‹$\neg x_1[F] \to \Box\neg x_1[F]$›
⟨*proof*⟩
AOT-theorem *pre−en−eq:2[2]*: ‹$\neg x_1x_2[F] \to \Box\neg x_1x_2[F]$›
⟨*proof*⟩

AOT-theorem *pre−en−eq:2[3]*: ‹$\neg x_1x_2x_3[F] \to \Box\neg x_1x_2x_3[F]$›
⟨*proof*⟩

AOT-theorem *pre−en−eq:2[4]*: ‹$\neg x_1x_2x_3x_4[F] \to \Box\neg x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *en−eq:1[1]*: ‹$\Diamond x_1[F] \equiv \Box x_1[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:1[2]*: ‹$\Diamond x_1x_2[F] \equiv \Box x_1x_2[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:1[3]*: ‹$\Diamond x_1x_2x_3[F] \equiv \Box x_1x_2x_3[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:1[4]*: ‹$\Diamond x_1x_2x_3x_4[F] \equiv \Box x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *en−eq:2[1]*: ‹$x_1[F] \equiv \Box x_1[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:2[2]*: ‹$x_1x_2[F] \equiv \Box x_1x_2[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:2[3]*: ‹$x_1x_2x_3[F] \equiv \Box x_1x_2x_3[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:2[4]*: ‹$x_1x_2x_3x_4[F] \equiv \Box x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *en−eq:3[1]*: ‹$\Diamond x_1[F] \equiv x_1[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:3[2]*: ‹$\Diamond x_1x_2[F] \equiv x_1x_2[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:3[3]*: ‹$\Diamond x_1x_2x_3[F] \equiv x_1x_2x_3[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:3[4]*: ‹$\Diamond x_1x_2x_3x_4[F] \equiv x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *en−eq:4[1]*:
‹$(x_1[F] \equiv y_1[G]) \equiv (\Box x_1[F] \equiv \Box y_1[G])$›
⟨*proof*⟩

AOT-theorem *en−eq:4[2]*:
‹$(x_1x_2[F] \equiv y_1y_2[G]) \equiv (\Box x_1x_2[F] \equiv \Box y_1y_2[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:4[3]*:
‹$(x_1x_2x_3[F] \equiv y_1y_2y_3[G]) \equiv (\Box x_1x_2x_3[F] \equiv \Box y_1y_2y_3[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:4[4]*:
‹$(x_1x_2x_3x_4[F] \equiv y_1y_2y_3y_4[G]) \equiv (\Box x_1x_2x_3x_4[F] \equiv \Box y_1y_2y_3y_4[G])$›
⟨*proof*⟩

AOT-theorem *en−eq:5[1]*:
‹$\Box(x_1[F] \equiv y_1[G]) \equiv (\Box x_1[F] \equiv \Box y_1[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:5[2]*:
‹$\Box(x_1x_2[F] \equiv y_1y_2[G]) \equiv (\Box x_1x_2[F] \equiv \Box y_1y_2[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:5[3]*:
‹$\Box(x_1x_2x_3[F] \equiv y_1y_2y_3[G]) \equiv (\Box x_1x_2x_3[F] \equiv \Box y_1y_2y_3[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:5[4]*:
‹$\Box(x_1x_2x_3x_4[F] \equiv y_1y_2y_3y_4[G]) \equiv (\Box x_1x_2x_3x_4[F] \equiv \Box y_1y_2y_3y_4[G])$›
⟨*proof*⟩

AOT-theorem *en−eq:6[1]*:
‹$(x_1[F] \equiv y_1[G]) \equiv \Box(x_1[F] \equiv y_1[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:6[2]*:
‹$(x_1x_2[F] \equiv y_1y_2[G]) \equiv \Box(x_1x_2[F] \equiv y_1y_2[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:6[3]*:
‹$(x_1x_2x_3[F] \equiv y_1y_2y_3[G]) \equiv \Box(x_1x_2x_3[F] \equiv y_1y_2y_3[G])$›
⟨*proof*⟩
AOT-theorem *en−eq:6[4]*:
‹$(x_1x_2x_3x_4[F] \equiv y_1y_2y_3y_4[G]) \equiv \Box(x_1x_2x_3x_4[F] \equiv y_1y_2y_3y_4[G])$›
⟨*proof*⟩

AOT-theorem *en−eq:7[1]*: ‹$\neg x_1[F] \equiv \Box\neg x_1[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:7[2]*: ‹$\neg x_1x_2[F] \equiv \Box\neg x_1x_2[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:7[3]*: ‹$\neg x_1x_2x_3[F] \equiv \Box\neg x_1x_2x_3[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:7[4]*: ‹$\neg x_1x_2x_3x_4[F] \equiv \Box\neg x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *en−eq:8[1]*: ‹$\Diamond\neg x_1[F] \equiv \neg x_1[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:8[2]*: ‹$\Diamond\neg x_1x_2[F] \equiv \neg x_1x_2[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:8[3]*: ‹$\Diamond\neg x_1x_2x_3[F] \equiv \neg x_1x_2x_3[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:8[4]*: ‹$\Diamond\neg x_1x_2x_3x_4[F] \equiv \neg x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *en−eq:9[1]*: ‹$\Diamond\neg x_1[F] \equiv \Box\neg x_1[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:9[2]*: ‹$\Diamond\neg x_1x_2[F] \equiv \Box\neg x_1x_2[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:9[3]*: ‹$\Diamond\neg x_1x_2x_3[F] \equiv \Box\neg x_1x_2x_3[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:9[4]*: ‹$\Diamond\neg x_1x_2x_3x_4[F] \equiv \Box\neg x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *en−eq:10[1]*: ‹$\boldsymbol{\mathcal{A}}x_1[F] \equiv x_1[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:10[2]*: ‹$\boldsymbol{\mathcal{A}}x_1x_2[F] \equiv x_1x_2[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:10[3]*: ‹$\boldsymbol{\mathcal{A}}x_1x_2x_3[F] \equiv x_1x_2x_3[F]$›
⟨*proof*⟩
AOT-theorem *en−eq:10[4]*: ‹$\boldsymbol{\mathcal{A}}x_1x_2x_3x_4[F] \equiv x_1x_2x_3x_4[F]$›
⟨*proof*⟩

AOT-theorem *oa−facts:1*: ‹$O!x \rightarrow \Box O!x$›
⟨*proof*⟩

AOT-theorem *oa−facts:2*: ‹$A!x \rightarrow \Box A!x$›
⟨*proof*⟩

AOT-theorem *oa−facts:3*: ‹$\Diamond O!x \rightarrow O!x$›
⟨*proof*⟩
AOT-theorem *oa−facts:4*: ‹$\Diamond A!x \rightarrow A!x$›
⟨*proof*⟩

AOT-theorem *oa−facts:5*: ‹$\Diamond O!x \equiv \Box O!x$›
⟨*proof*⟩

AOT-theorem *oa−facts:6*: ‹$\Diamond A!x \equiv \Box A!x$›
⟨*proof*⟩

AOT-theorem *oa−facts:7*: ‹$O!x \equiv \boldsymbol{\mathcal{A}}O!x$›
⟨*proof*⟩

AOT-theorem *oa−facts:8*: ‹$A!x \equiv \boldsymbol{\mathcal{A}}A!x$›
⟨*proof*⟩

8.10 The Theory of Relations

AOT-theorem *beta−C−meta*:
‹$[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n,\ \nu_1...\nu_n\}]\downarrow \rightarrow$
$([\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n,\ \nu_1...\nu_n\}]\nu_1...\nu_n \equiv \varphi\{\nu_1...\nu_n,\ \nu_1...\nu_n\})$›
⟨*proof*⟩

AOT-theorem *beta−C−cor:1*:
‹$(\forall\nu_1...\forall\nu_n([\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n,\ \nu_1...\nu_n\}]\downarrow)) \rightarrow$
$\forall\nu_1...\forall\nu_n\ ([\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n,\ \nu_1...\nu_n\}]\nu_1...\nu_n \equiv \varphi\{\nu_1...\nu_n,\ \nu_1...\nu_n\})$›
⟨*proof*⟩

AOT-theorem *beta−C−cor:2*:
‹$[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\downarrow \rightarrow$
$\forall\nu_1...\forall\nu_n\ ([\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\nu_1...\nu_n \equiv \varphi\{\nu_1...\nu_n\})$›
⟨*proof*⟩

theorem *beta−C−cor:3*:
assumes ‹$\bigwedge\nu_1\nu_n.$ *AOT-instance-of-cqt-2* $(\varphi\ ($*AOT-term-of-var* $\nu_1\nu_n))$›
shows ‹$[v \models \forall\nu_1...\forall\nu_n\ ([\lambda\mu_1...\mu_n\ \varphi\{\nu_1...\nu_n,\ \mu_1...\mu_n\}]\nu_1...\nu_n \equiv$
$\varphi\{\nu_1...\nu_n,\ \nu_1...\nu_n\})]$›
⟨*proof*⟩

AOT-theorem *betaC:1:a*: ‹$[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\kappa_1...\kappa_n \vdash_\Box \varphi\{\kappa_1...\kappa_n\}$›
⟨*proof*⟩

AOT-theorem *betaC:1:b*: ‹$\neg\varphi\{\kappa_1...\kappa_n\} \vdash_\Box \neg[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\kappa_1...\kappa_n$›
⟨*proof*⟩

lemmas $\beta{\rightarrow}C$ = *betaC:1:a betaC:1:b*

AOT-theorem *betaC*:*2*:*a*:
‹$[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\downarrow$, $\kappa_1...\kappa_n\downarrow$, $\varphi\{\kappa_1...\kappa_n\} \vdash_\Box$
$[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\kappa_1...\kappa_n$›
⟨*proof*⟩

AOT-theorem *betaC*:*2*:*b*:
‹$[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\downarrow$, $\kappa_1...\kappa_n\downarrow$, $\neg[\lambda\mu_1...\mu_n\ \varphi\{\mu_1...\mu_n\}]\kappa_1...\kappa_n \vdash_\Box$
$\neg\varphi\{\kappa_1...\kappa_n\}$›
⟨*proof*⟩

lemmas $\beta{\leftarrow}C$ = *betaC*:*2*:*a* *betaC*:*2*:*b*

AOT-theorem *eta−conversion−lemma1*:*1*: ‹$\Pi\downarrow \rightarrow [\lambda x_1...x_n\ [\Pi]x_1...x_n] = \Pi$›
⟨*proof*⟩

AOT-theorem *eta−conversion−lemma1*:*2*: ‹$\Pi\downarrow \rightarrow [\lambda\nu_1...\nu_n\ [\Pi]\nu_1...\nu_n] = \Pi$›
⟨*proof*⟩

Note: not explicitly part of PLM.

AOT-theorem *id-sym*:
assumes ‹$\tau = \tau'$›
shows ‹$\tau' = \tau$›
⟨*proof*⟩
declare *id-sym*[*sym*]

Note: not explicitly part of PLM.

AOT-theorem *id-trans*:
assumes ‹$\tau = \tau'$› **and** ‹$\tau' = \tau''$›
shows ‹$\tau = \tau''$›
⟨*proof*⟩
declare *id-trans*[*trans*]

method ηC **for** Π :: ‹<'*a*::{*AOT-Term-id-2*,*AOT-κs*}>› =
(*match* **conclusion in** $[v \models \tau\{\Pi\} = \tau'\{\Pi\}]$ **for** $v\ \tau\ \tau'$ ⇒ ‹
rule rule=*E*[*rotated 1*, *OF eta−conversion−lemma1*:*2*
[*THEN* →*E*, *of v* «[Π]», *symmetric*]]›)

AOT-theorem *sub−des−lam*:*1*:
‹$[\lambda z_1...z_n\ \ \chi\{z_1...z_n, \iota x\ \varphi\{x\}\}]\downarrow\ \&\ \iota x\ \varphi\{x\} = \iota x\ \psi\{x\} \rightarrow$
$[\lambda z_1...z_n\ \chi\{z_1...z_n, \iota x\ \varphi\{x\}\}] = [\lambda z_1...z_n\ \chi\{z_1...z_n, \iota x\ \psi\{x\}\}]$›
⟨*proof*⟩

AOT-theorem *sub−des−lam*:*2*:
‹$\iota x\ \varphi\{x\} = \iota x\ \psi\{x\} \rightarrow \chi\{\iota x\ \varphi\{x\}\} = \chi\{\iota x\ \psi\{x\}\}$› **for** χ :: ‹κ ⇒ o›
⟨*proof*⟩

AOT-theorem *prop−equiv*: ‹$F = G \equiv \forall x\ (x[F] \equiv x[G])$›
⟨*proof*⟩

AOT-theorem *relations*:*1*:
assumes ‹*INSTANCE-OF-CQT-2*(φ)›
shows ‹$\exists F\ \Box\forall x_1...\forall x_n\ ([F]x_1...x_n \equiv \varphi\{x_1...x_n\})$›
⟨*proof*⟩

AOT-theorem *relations*:*2*:
assumes ‹*INSTANCE-OF-CQT-2*(φ)›
shows ‹$\exists F\ \Box\forall x\ ([F]x \equiv \varphi\{x\})$›
⟨*proof*⟩

AOT-theorem *block−paradox*:*1*: ‹$\neg[\lambda x\ \exists G\ (x[G]\ \&\ \neg[G]x)]\downarrow$›
⟨*proof*⟩

AOT-theorem *block−paradox:2*: $\langle\neg\exists F\ \forall x([F]x \equiv \exists G(x[G]\ \&\ \neg[G]x))\rangle$
⟨*proof*⟩

AOT-theorem *block−paradox:3*: $\langle\neg\forall y\ [\lambda z\ z = y]\downarrow\rangle$
⟨*proof*⟩

AOT-theorem *block−paradox:4*: $\langle\neg\forall y\ \exists F\ \forall x([F]x \equiv x = y)\rangle$
⟨*proof*⟩

AOT-theorem *block−paradox:5*: $\langle\neg\exists F\forall x\forall y([F]xy \equiv y = x)\rangle$
⟨*proof*⟩

AOT-act-theorem *block−paradox2:1*:
$\langle\forall x\ [G]x \rightarrow \neg[\lambda x\ [G]\iota y\ (y = x\ \&\ \exists H\ (x[H]\ \&\ \neg[H]x))]\downarrow\rangle$
⟨*proof*⟩

Note: Strengthens the above to a modally-strict theorem. Not explicitly part of PLM.

AOT-theorem *block−paradox2:1[strict]*:
$\langle\forall x\ \boldsymbol{\mathcal{A}}[G]x \rightarrow \neg[\lambda x\ [G]\iota y\ (y = x\ \&\ \exists H\ (x[H]\ \&\ \neg[H]x))]\downarrow\rangle$
⟨*proof*⟩

AOT-act-theorem *block−paradox2:2*:
$\langle\exists G\ \neg[\lambda x\ [G]\iota y\ (y = x\ \&\ \exists H\ (x[H]\ \&\ \neg[H]x))]\downarrow\rangle$
⟨*proof*⟩

AOT-theorem *propositions*: $\langle\exists p\ \Box(p \equiv \varphi)\rangle$
⟨*proof*⟩

AOT-theorem *pos−not−equiv−ne:1*:
$\langle(\Diamond\neg\forall x_1...\forall x_n\ ([F]x_1...x_n \equiv [G]x_1...x_n)) \rightarrow F \neq G\rangle$
⟨*proof*⟩

AOT-theorem *pos−not−equiv−ne:2*: $\langle(\Diamond\neg(\varphi\{F\} \equiv \varphi\{G\})) \rightarrow F \neq G\rangle$
⟨*proof*⟩

AOT-theorem *pos−not−equiv−ne:2[zero]*: $\langle(\Diamond\neg(\varphi\{p\} \equiv \varphi\{q\})) \rightarrow p \neq q\rangle$
⟨*proof*⟩

AOT-theorem *pos−not−equiv−ne:3*:
$\langle(\neg\forall x_1...\forall x_n\ ([F]x_1...x_n \equiv [G]x_1...x_n)) \rightarrow F \neq G\rangle$
⟨*proof*⟩

AOT-theorem *pos−not−equiv−ne:4*: $\langle(\neg(\varphi\{F\} \equiv \varphi\{G\})) \rightarrow F \neq G\rangle$
⟨*proof*⟩

AOT-theorem *pos−not−equiv−ne:4[zero]*: $\langle(\neg(\varphi\{p\} \equiv \varphi\{q\})) \rightarrow p \neq q\rangle$
⟨*proof*⟩

AOT-define *relation-negation* :: $\Pi \Rightarrow \Pi$ (‹-$^-$›)
df−relation−negation: $[F]^- =_{df} [\lambda x_1...x_n\ \neg[F]x_1...x_n]$

nonterminal φneg
syntax :: $\varphi neg \Rightarrow \tau$ (‹-›)
syntax :: $\varphi neg \Rightarrow \varphi$ (‹'(-')›)

AOT-define *relation-negation-0* :: ‹$\varphi \Rightarrow \varphi neg$› (‹'(-')$^-$›)
df−relation−negation[zero]: $(p)^- =_{df} [\lambda\ \neg p]$

AOT-theorem *rel−neg−T:1*: $\langle[\lambda x_1...x_n\ \neg[\Pi]x_1...x_n]\downarrow\rangle$
⟨*proof*⟩

AOT-theorem *rel−neg−T:1[zero]*: $\langle[\lambda\ \neg\varphi]\downarrow\rangle$

⟨*proof*⟩

AOT-theorem *rel−neg−T:2*: ‹$[\Pi]^- = [\lambda x_1...x_n\ \neg[\Pi]x_1...x_n]$›
⟨*proof*⟩

AOT-theorem *rel−neg−T:2[zero]*: ‹$(\varphi)^- = [\lambda\ \neg\varphi]$›
⟨*proof*⟩

AOT-theorem *rel−neg−T:3*: ‹$[\Pi]^-\downarrow$›
⟨*proof*⟩

AOT-theorem *rel−neg−T:3[zero]*: ‹$(\varphi)^-\downarrow$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:1*: ‹$[F]^- x_1...x_n \equiv \neg[F]x_1...x_n$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:2*: ‹$\neg[F]^- x_1...x_n \equiv [F]x_1...x_n$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:3*: ‹$((p)^-) \equiv \neg p$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:4*: ‹$(\neg((p)^-)) \equiv p$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:5*: ‹$[F] \neq [F]^-$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:6*: ‹$p \neq (p)^-$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:7*: ‹$(p)^- = (\neg p)$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:8*: ‹$p = q \rightarrow (\neg p) = (\neg q)$›
⟨*proof*⟩

AOT-theorem *thm−relation−negation:9*: ‹$p = q \rightarrow (p)^- = (q)^-$›
⟨*proof*⟩

AOT-define *Necessary* :: ‹$\Pi \Rightarrow \varphi$› (‹*Necessary*′(-′)›)
contingent−properties:1:
‹$Necessary([F]) \equiv_{df} \Box\forall x_1...\forall x_n\ [F]x_1...x_n$›

AOT-define *Necessary0* :: ‹$\varphi \Rightarrow \varphi$› (‹*Necessary0*′(-′)›)
contingent−properties:1[zero]:
‹$Necessary0(p) \equiv_{df} \Box p$›

AOT-define *Impossible* :: ‹$\Pi \Rightarrow \varphi$› (‹*Impossible*′(-′)›)
contingent−properties:2:
‹$Impossible([F]) \equiv_{df} F\downarrow\ \&\ \Box\forall x_1...\forall x_n\ \neg[F]x_1...x_n$›

AOT-define *Impossible0* :: ‹$\varphi \Rightarrow \varphi$› (‹*Impossible0*′(-′)›)
contingent−properties:2[zero]:
‹$Impossible0(p) \equiv_{df} \Box\neg p$›

AOT-define *NonContingent* :: ‹$\Pi \Rightarrow \varphi$› (‹*NonContingent*′(-′)›)
contingent−properties:3:
‹$NonContingent([F]) \equiv_{df} Necessary([F]) \vee Impossible([F])$›

AOT-define *NonContingent0* :: ‹$\varphi \Rightarrow \varphi$› (‹*NonContingent0*′(-′)›)
contingent−properties:3[zero]:

‹*NonContingent0*(p) $\equiv_{df}$ *Necessary0*(p) $\vee$ *Impossible0*(p)›

AOT-define *Contingent* :: ‹$\Pi \Rightarrow \varphi$› (‹*Contingent*′(-′)›)
contingent−properties:*4*:
‹*Contingent*($[F]$) $\equiv_{df}$ $F\downarrow$ & $\neg$(*Necessary*($[F]$) $\vee$ *Impossible*($[F]$))›

AOT-define *Contingent0* :: ‹$\varphi \Rightarrow \varphi$› (‹*Contingent0*′(-′)›)
contingent−properties:*4*[*zero*]:
‹*Contingent0*(p) $\equiv_{df}$ $\neg$(*Necessary0*(p) $\vee$ *Impossible0*(p))›

AOT-theorem *thm−cont−prop*:*1*: ‹*NonContingent*($[F]$) $\equiv$ *NonContingent*($[F]^-$)›
⟨*proof*⟩

AOT-theorem *thm−cont−prop*:*2*: ‹*Contingent*($[F]$) $\equiv \Diamond\exists\, x\ [F]x$ & $\Diamond\exists\, x\ \neg[F]x$›
⟨*proof*⟩

AOT-theorem *thm−cont−prop*:*3*:
‹*Contingent*($[F]$) $\equiv$ *Contingent*($[F]^-$)› **for** *F*::‹<κ> *AOT-var*›
⟨*proof*⟩

AOT-define *concrete-if-concrete* :: ‹Π› (‹*L*›)
L-def: ‹$L =_{df} [\lambda x\ E!x \rightarrow E!x]$›

AOT-theorem *thm−noncont−e−e*:*1*: ‹*Necessary*(L)›
⟨*proof*⟩

AOT-theorem *thm−noncont−e−e*:*2*: ‹*Impossible*($[L]^-$)›
⟨*proof*⟩

AOT-theorem *thm−noncont−e−e*:*3*: ‹*NonContingent*(L)›
⟨*proof*⟩

AOT-theorem *thm−noncont−e−e*:*4*: ‹*NonContingent*($[L]^-$)›
⟨*proof*⟩

AOT-theorem *thm−noncont−e−e*:*5*:
‹$\exists\, F\ \exists\, G$ ($F \neq$ «*G*::<κ>» & *NonContingent*($[F]$) & *NonContingent*($[G]$))›
⟨*proof*⟩

AOT-theorem *lem−cont−e*:*1*: ‹$\Diamond\exists\, x$ ($[F]x$ & $\Diamond\neg[F]x$) $\equiv \Diamond\exists\, x$ ($\neg[F]x$ & $\Diamond[F]x$)›
⟨*proof*⟩

AOT-theorem *lem−cont−e*:*2*:
‹$\Diamond\exists\, x$ ($[F]x$ & $\Diamond\neg[F]x$) $\equiv \Diamond\exists\, x$ ($[F]^-x$ & $\Diamond\neg[F]^-x$)›
⟨*proof*⟩

AOT-theorem *thm−cont−e*:*1*: ‹$\Diamond\exists\, x$ ($E!x$ & $\Diamond\neg E!x$)›
⟨*proof*⟩

AOT-theorem *thm−cont−e*:*2*: ‹$\Diamond\exists\, x$ ($\neg E!x$ & $\Diamond E!x$)›
⟨*proof*⟩

AOT-theorem *thm−cont−e*:*3*: ‹$\Diamond\exists\, x\ E!x$›
⟨*proof*⟩

AOT-theorem *thm−cont−e*:*4*: ‹$\Diamond\exists\, x\ \neg E!x$›
⟨*proof*⟩

AOT-theorem *thm−cont−e*:*5*: ‹*Contingent*($[E!]$)›
⟨*proof*⟩

AOT-theorem *thm−cont−e*:*6*: ‹*Contingent*($[E!]^-$)›

$\langle proof \rangle$

AOT-theorem *thm−cont−e:7*:
$‹\exists F \exists G\ (Contingent([«F::<\kappa>»])\ \&\ Contingent([G])\ \&\ F \neq G)›$
$\langle proof \rangle$

AOT-theorem *property−facts:1*:
$‹NonContingent([F]) \rightarrow \neg\exists G\ (Contingent([G])\ \&\ G = F)›$
$\langle proof \rangle$

AOT-theorem *property−facts:2*:
$‹Contingent([F]) \rightarrow \neg\exists G\ (NonContingent([G])\ \&\ G = F)›$
$\langle proof \rangle$

AOT-theorem *property−facts:3*:
$‹L \neq [L]^-\ \&\ L \neq E!\ \&\ L \neq E!^-\ \&\ [L]^- \neq [E!]^-\ \&\ E! \neq [E!]^-›$
$\langle proof \rangle$

AOT-theorem *thm−cont−propos:1*:
$‹NonContingent0(p) \equiv NonContingent0(((p)^-))›$
$\langle proof \rangle$

AOT-theorem *thm−cont−propos:2*: $‹Contingent0(\varphi) \equiv \Diamond\varphi\ \&\ \Diamond\neg\varphi›$
$\langle proof \rangle$

AOT-theorem *thm−cont−propos:3*: $‹Contingent0(p) \equiv Contingent0(((p)^-))›$
$\langle proof \rangle$

AOT-define *noncontingent-prop* :: $‹\varphi›\ (‹p_0›)$
p_0*-def*: $(p_0) =_{df} (\forall x\ (E!x \rightarrow E!x))$

AOT-theorem *thm−noncont−propos:1*: $‹Necessary0((p_0))›$
$\langle proof \rangle$

AOT-theorem *thm−noncont−propos:2*: $‹Impossible0(((p_0)^-))›$
$\langle proof \rangle$

AOT-theorem *thm−noncont−propos:3*: $‹NonContingent0((p_0))›$
$\langle proof \rangle$

AOT-theorem *thm−noncont−propos:4*: $‹NonContingent0(((p_0)^-))›$
$\langle proof \rangle$

AOT-theorem *thm−noncont−propos:5*:
$‹\exists p \exists q\ (NonContingent0((p))\ \&\ NonContingent0((q))\ \&\ p \neq q)›$
$\langle proof \rangle$

AOT-act-theorem *no−cnac*: $‹\neg\exists x(E!x\ \&\ \neg\mathcal{A}E!x)›$
$\langle proof \rangle$

AOT-theorem *pos−not−pna:1*: $‹\neg\mathcal{A}\exists x\ (E!x\ \&\ \neg\mathcal{A}E!x)›$
$\langle proof \rangle$

AOT-theorem *pos−not−pna:2*: $‹\Diamond\neg\exists x(E!x\ \&\ \neg\mathcal{A}E!x)›$
$\langle proof \rangle$

AOT-theorem *pos−not−pna:3*: $‹\exists x\ (\Diamond E!x\ \&\ \neg\mathcal{A}E!x)›$
$\langle proof \rangle$

AOT-define *contingent-prop* :: $\varphi\ (‹q_0›)$
q_0*-def*: $‹(q_0) =_{df} (\exists x\ (E!x\ \&\ \neg\mathcal{A}E!x))›$

AOT-theorem q_0*-prop*: $‹\Diamond q_0\ \&\ \Diamond\neg q_0›$

⟨*proof*⟩

AOT-theorem *basic−prop:1*: $\langle Contingent0((q_0))\rangle$
⟨*proof*⟩

AOT-theorem *basic−prop:2*: $\langle \exists\, p\ Contingent0((p))\rangle$
⟨*proof*⟩

AOT-theorem *basic−prop:3*: $\langle Contingent0(((q_0)^-))\rangle$
⟨*proof*⟩

AOT-theorem *basic−prop:4*:
$\langle \exists\, p \exists\, q\ (p \neq q\ \&\ Contingent0(p)\ \&\ Contingent0(q))\rangle$
⟨*proof*⟩

AOT-theorem *proposition−facts:1*:
$\langle NonContingent0(p) \rightarrow \neg\exists\, q\ (Contingent0(q)\ \&\ q = p)\rangle$
⟨*proof*⟩

AOT-theorem *proposition−facts:2*:
$\langle Contingent0(p) \rightarrow \neg\exists\, q\ (NonContingent0(q)\ \&\ q = p)\rangle$
⟨*proof*⟩

AOT-theorem *proposition−facts:3*:
$\langle (p_0) \neq (p_0)^-\ \&\ (p_0) \neq (q_0)\ \&\ (p_0) \neq (q_0)^-\ \&\ (p_0)^- \neq (q_0)^-\ \&\ (q_0) \neq (q_0)^-\rangle$
⟨*proof*⟩

AOT-define *ContingentlyTrue* :: $\langle\varphi \Rightarrow \varphi\rangle$ ($\langle ContingentlyTrue'(\text{-}')\rangle$)
cont−tf:1: $\langle ContingentlyTrue(p) \equiv_{df}\ p\ \&\ \Diamond\neg p\rangle$

AOT-define *ContingentlyFalse* :: $\langle\varphi \Rightarrow \varphi\rangle$ ($\langle ContingentlyFalse'(\text{-}')\rangle$)
cont−tf:2: $\langle ContingentlyFalse(p) \equiv_{df}\ \neg p\ \&\ \Diamond p\rangle$

AOT-theorem *cont−true−cont:1*:
$\langle ContingentlyTrue((p)) \rightarrow Contingent0((p))\rangle$
⟨*proof*⟩

AOT-theorem *cont−true−cont:2*:
$\langle ContingentlyFalse((p)) \rightarrow Contingent0((p))\rangle$
⟨*proof*⟩

AOT-theorem *cont−true−cont:3*:
$\langle ContingentlyTrue((p)) \equiv ContingentlyFalse(((p)^-))\rangle$
⟨*proof*⟩

AOT-theorem *cont−true−cont:4*:
$\langle ContingentlyFalse((p)) \equiv ContingentlyTrue(((p)^-))\rangle$
⟨*proof*⟩

AOT-theorem *cont−true−cont:5*:
$\langle (ContingentlyTrue((p))\ \&\ Necessary0((q))) \rightarrow p \neq q\rangle$
⟨*proof*⟩

AOT-theorem *cont−true−cont:6*:
$\langle (ContingentlyFalse((p))\ \&\ Impossible0((q))) \rightarrow p \neq q\rangle$
⟨*proof*⟩

AOT-act-theorem *q0cf:1*: $\langle ContingentlyFalse(q_0)\rangle$
⟨*proof*⟩

AOT-act-theorem *q0cf:2*: $\langle ContingentlyTrue(((q_0)^-))\rangle$
⟨*proof*⟩

AOT-theorem *cont−tf−thm*:*1*: ‹$\exists\, p$ *ContingentlyTrue*((*p*))›
⟨*proof*⟩

AOT-theorem *cont−tf−thm*:*2*: ‹$\exists\, p$ *ContingentlyFalse*((*p*))›
⟨*proof*⟩

AOT-theorem *property−facts1*:*1*: ‹$\exists\, F \exists\, x$ ($[F]x$ & $\Diamond\neg[F]x$)›
⟨*proof*⟩

AOT-theorem *property−facts1*:*2*: ‹$\exists\, F \exists\, x$ ($\neg[F]x$ & $\Diamond[F]x$)›
⟨*proof*⟩

context
begin

private AOT-lemma *eqnotnec-123-Aux-ζ*: ‹$[L]x \equiv (E!x \rightarrow E!x)$›
⟨*proof*⟩ **AOT-lemma** *eqnotnec-123-Aux-ω*: ‹$[\lambda z\ \varphi]x \equiv \varphi$›
⟨*proof*⟩ **AOT-lemma** *eqnotnec-123-Aux-ϑ*: ‹$\varphi \equiv \forall\, x([L]x \equiv [\lambda z\ \varphi]x)$›
⟨*proof*⟩ **lemmas** *eqnotnec-123-Aux-ξ* =
eqnotnec-123-Aux-ϑ[*THEN oth−class−taut*:*4*:*b*[*THEN* ≡*E*(*1*)],
THEN conventions:*3*[*THEN* ≡*Df*, *THEN* ≡*E*(*1*), *THEN* &*E*(*1*)],
THEN RM◊]
private lemmas *eqnotnec-123-Aux-ξ′* =
eqnotnec-123-Aux-ϑ[
THEN conventions:*3*[*THEN* ≡*Df*, *THEN* ≡*E*(*1*), *THEN* &*E*(*1*)],
THEN RM◊]

AOT-theorem *eqnotnec*:*1*: ‹$\exists\, F \exists\, G(\forall\, x([F]x \equiv [G]x)$ & $\Diamond\neg\forall\, x([F]x \equiv [G]x))$›
⟨*proof*⟩

AOT-theorem *eqnotnec*:*2*: ‹$\exists\, F \exists\, G(\neg\forall\, x([F]x \equiv [G]x)$ & $\Diamond\forall\, x([F]x \equiv [G]x))$›
⟨*proof*⟩

AOT-theorem *eqnotnec*:*3*: ‹$\exists\, F \exists\, G(\boldsymbol{\mathcal{A}}\neg\forall\, x([F]x \equiv [G]x)$ & $\Diamond\forall\, x([F]x \equiv [G]x))$›
⟨*proof*⟩

end

AOT-theorem *eqnotnec*:*4*: ‹$\forall\, F \exists\, G(\forall\, x([F]x \equiv [G]x)$ & $\Diamond\neg\forall\, x([F]x \equiv [G]x))$›
⟨*proof*⟩

AOT-theorem *eqnotnec*:*5*: ‹$\forall\, F \exists\, G(\neg\forall\, x([F]x \equiv [G]x)$ & $\Diamond\forall\, x([F]x \equiv [G]x))$›
⟨*proof*⟩

AOT-theorem *eqnotnec*:*6*: ‹$\forall\, F \exists\, G(\boldsymbol{\mathcal{A}}\neg\forall\, x([F]x \equiv [G]x)$ & $\Diamond\forall\, x([F]x \equiv [G]x))$›
⟨*proof*⟩

AOT-theorem *oa−contingent*:*1*: ‹$O! \neq A!$›
⟨*proof*⟩

AOT-theorem *oa−contingent*:*2*: ‹$O!x \equiv \neg A!x$›
⟨*proof*⟩

AOT-theorem *oa−contingent*:*3*: ‹$A!x \equiv \neg O!x$›
⟨*proof*⟩

AOT-theorem *oa−contingent*:*4*: ‹*Contingent*($O!$)›
⟨*proof*⟩

AOT-theorem *oa−contingent*:*5*: ‹*Contingent*($A!$)›
⟨*proof*⟩

AOT-theorem *oa−contingent*:*7*: $\langle O!^{-}x \equiv \neg A!^{-}x\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *oa−contingent*:*6*: $\langle O!^{-} \neq A!^{-}\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *oa−contingent*:*8*: $\langle \mathit{Contingent}(O!^{-})\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *oa−contingent*:*9*: $\langle \mathit{Contingent}(A!^{-})\rangle$
$\langle$*proof*$\rangle$

AOT-define *WeaklyContingent* :: $\langle \Pi \Rightarrow \varphi\rangle$ $(\langle \mathit{WeaklyContingent}'(\text{-}')\rangle)$
df−cont−nec:
$\langle \mathit{WeaklyContingent}([F]) \equiv_{df} \mathit{Contingent}([F])\ \&\ \forall x\ (\Diamond[F]x \rightarrow \Box[F]x)\rangle$

AOT-theorem *cont−nec−fact1*:*1*:
$\langle \mathit{WeaklyContingent}([F]) \equiv \mathit{WeaklyContingent}([F]^{-})\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *cont−nec−fact1*:*2*:
$\langle(\mathit{WeaklyContingent}([F])\ \&\ \neg \mathit{WeaklyContingent}([G])) \rightarrow F \neq G\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *cont−nec−fact2*:*1*: $\langle \mathit{WeaklyContingent}(O!)\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *cont−nec−fact2*:*2*: $\langle \mathit{WeaklyContingent}(A!)\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *cont−nec−fact2*:*3*: $\langle\neg \mathit{WeaklyContingent}(E!)\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *cont−nec−fact2*:*4*: $\langle\neg \mathit{WeaklyContingent}(L)\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *cont−nec−fact2*:*5*: $\langle O! \neq E!\ \&\ O! \neq E!^{-}\ \&\ O! \neq L\ \&\ O! \neq L^{-}\rangle$
$\langle$*proof*$\rangle$

AOT-theorem *cont−nec−fact2*:*6*: $\langle A! \neq E!\ \&\ A! \neq E!^{-}\ \&\ A! \neq L\ \&\ A! \neq L^{-}\rangle$
$\langle$*proof*$\rangle$

AOT-define *necessary-or-contingently-false* :: $\langle\varphi \Rightarrow \varphi\rangle$ $(\langle\boldsymbol{\Delta}\text{-}\rangle\ [49]\ 54)$
$\langle\boldsymbol{\Delta}p \equiv_{df} \Box p \vee (\neg\boldsymbol{\mathcal{A}}p\ \&\ \Diamond p)\rangle$

AOT-theorem *sixteen*:
shows $\langle\exists F_1\exists F_2\exists F_3\exists F_4\exists F_5\exists F_6\exists F_7\exists F_8\exists F_9\exists F_{10}\exists F_{11}\exists F_{12}\exists F_{13}\exists F_{14}\exists F_{15}\exists F_{16}\ ($
$«F_1::<\kappa>» \neq F_2\ \&\ F_1 \neq F_3\ \&\ F_1 \neq F_4\ \&\ F_1 \neq F_5\ \&\ F_1 \neq F_6\ \&\ F_1 \neq F_7\ \&$
$F_1 \neq F_8\ \&\ F_1 \neq F_9\ \&\ F_1 \neq F_{10}\ \&\ F_1 \neq F_{11}\ \&\ F_1 \neq F_{12}\ \&\ F_1 \neq F_{13}\ \&$
$F_1 \neq F_{14}\ \&\ F_1 \neq F_{15}\ \&\ F_1 \neq F_{16}\ \&$
$F_2 \neq F_3\ \&\ F_2 \neq F_4\ \&\ F_2 \neq F_5\ \&\ F_2 \neq F_6\ \&\ F_2 \neq F_7\ \&\ F_2 \neq F_8\ \&$
$F_2 \neq F_9\ \&\ F_2 \neq F_{10}\ \&\ F_2 \neq F_{11}\ \&\ F_2 \neq F_{12}\ \&\ F_2 \neq F_{13}\ \&\ F_2 \neq F_{14}\ \&$
$F_2 \neq F_{15}\ \&\ F_2 \neq F_{16}\ \&$
$F_3 \neq F_4\ \&\ F_3 \neq F_5\ \&\ F_3 \neq F_6\ \&\ F_3 \neq F_7\ \&\ F_3 \neq F_8\ \&\ F_3 \neq F_9\ \&\ F_3 \neq F_{10}\ \&$
$F_3 \neq F_{11}\ \&\ F_3 \neq F_{12}\ \&\ F_3 \neq F_{13}\ \&\ F_3 \neq F_{14}\ \&\ F_3 \neq F_{15}\ \&\ F_3 \neq F_{16}\ \&$
$F_4 \neq F_5\ \&\ F_4 \neq F_6\ \&\ F_4 \neq F_7\ \&\ F_4 \neq F_8\ \&\ F_4 \neq F_9\ \&\ F_4 \neq F_{10}\ \&\ F_4 \neq F_{11}\ \&$
$F_4 \neq F_{12}\ \&\ F_4 \neq F_{13}\ \&\ F_4 \neq F_{14}\ \&\ F_4 \neq F_{15}\ \&\ F_4 \neq F_{16}\ \&$
$F_5 \neq F_6\ \&\ F_5 \neq F_7\ \&\ F_5 \neq F_8\ \&\ F_5 \neq F_9\ \&\ F_5 \neq F_{10}\ \&\ F_5 \neq F_{11}\ \&\ F_5 \neq F_{12}\ \&$
$F_5 \neq F_{13}\ \&\ F_5 \neq F_{14}\ \&\ F_5 \neq F_{15}\ \&\ F_5 \neq F_{16}\ \&$
$F_6 \neq F_7\ \&\ F_6 \neq F_8\ \&\ F_6 \neq F_9\ \&\ F_6 \neq F_{10}\ \&\ F_6 \neq F_{11}\ \&\ F_6 \neq F_{12}\ \&\ F_6 \neq F_{13}\ \&$
$F_6 \neq F_{14}\ \&\ F_6 \neq F_{15}\ \&\ F_6 \neq F_{16}\ \&$
$F_7 \neq F_8\ \&\ F_7 \neq F_9\ \&\ F_7 \neq F_{10}\ \&\ F_7 \neq F_{11}\ \&\ F_7 \neq F_{12}\ \&\ F_7 \neq F_{13}\ \&\ F_7 \neq F_{14}\ \&$

$F_7 \neq F_{15}$ & $F_7 \neq F_{16}$ &
$F_8 \neq F_9$ & $F_8 \neq F_{10}$ & $F_8 \neq F_{11}$ & $F_8 \neq F_{12}$ & $F_8 \neq F_{13}$ & $F_8 \neq F_{14}$ & $F_8 \neq F_{15}$ &
$F_8 \neq F_{16}$ &
$F_9 \neq F_{10}$ & $F_9 \neq F_{11}$ & $F_9 \neq F_{12}$ & $F_9 \neq F_{13}$ & $F_9 \neq F_{14}$ & $F_9 \neq F_{15}$ & $F_9 \neq F_{16}$ &
$F_{10} \neq F_{11}$ & $F_{10} \neq F_{12}$ & $F_{10} \neq F_{13}$ & $F_{10} \neq F_{14}$ & $F_{10} \neq F_{15}$ & $F_{10} \neq F_{16}$ &
$F_{11} \neq F_{12}$ & $F_{11} \neq F_{13}$ & $F_{11} \neq F_{14}$ & $F_{11} \neq F_{15}$ & $F_{11} \neq F_{16}$ &
$F_{12} \neq F_{13}$ & $F_{12} \neq F_{14}$ & $F_{12} \neq F_{15}$ & $F_{12} \neq F_{16}$ &
$F_{13} \neq F_{14}$ & $F_{13} \neq F_{15}$ & $F_{13} \neq F_{16}$ &
$F_{14} \neq F_{15}$ & $F_{14} \neq F_{16}$ &
$F_{15} \neq F_{16}$)›
⟨*proof*⟩

8.11 The Theory of Objects

AOT-theorem *o−objects−exist:1*: ‹$\Box\exists x\ O!x$›
⟨*proof*⟩

AOT-theorem *o−objects−exist:2*: ‹$\Box\exists x\ A!x$›
⟨*proof*⟩

AOT-theorem *o−objects−exist:3*: ‹$\Box\neg\forall x\ O!x$›
⟨*proof*⟩

AOT-theorem *o−objects−exist:4*: ‹$\Box\neg\forall x\ A!x$›
⟨*proof*⟩

AOT-theorem *o−objects−exist:5*: ‹$\Box\neg\forall x\ E!x$›
⟨*proof*⟩

AOT-theorem *partition*: ‹$\neg\exists x\ (O!x\ \&\ A!x)$›
⟨*proof*⟩

AOT-define *eq-E* :: ‹Π› (‹'($=_E$')›)
=*E*: ‹$(=_E) =_{df} [\lambda xy\ O!x\ \&\ O!y\ \&\ \Box\forall F\ ([F]x \equiv [F]y)]$›

syntax *-AOT-eq-E-infix* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (**infixl** ‹$=_E$› *50*)
translations
-AOT-eq-E-infix $\kappa\ \kappa'$ == *CONST AOT-exe* (*CONST eq-E*) (*CONST Pair* $\kappa\ \kappa'$)
⟨*ML*⟩

Note: Not explicitly mentioned as theorem in PLM.

AOT-theorem =*E*[*denotes*]: ‹$[(=_E)]\downarrow$›
⟨*proof*⟩

AOT-theorem =*E−simple:1*: ‹$x =_E y \equiv (O!x\ \&\ O!y\ \&\ \Box\forall F\ ([F]x \equiv [F]y))$›
⟨*proof*⟩

AOT-theorem =*E−simple:2*: ‹$x =_E y \rightarrow x = y$›
⟨*proof*⟩

AOT-theorem *id−nec3:1*: ‹$x =_E y \equiv \Box(x =_E y)$›
⟨*proof*⟩

AOT-theorem *id−nec3:2*: ‹$\Diamond(x =_E y) \equiv x =_E y$›
⟨*proof*⟩

AOT-theorem *id−nec3:3*: ‹$\Diamond(x =_E y) \equiv \Box(x =_E y)$›
⟨*proof*⟩

syntax *-AOT-non-eq-E* :: ‹Π› (‹'($\neq_E$')›)
translations
(Π) ($\neq_E$) == (Π) $(=_E)^-$
syntax *-AOT-non-eq-E-infix* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (**infixl** ‹$\neq_E$› *50*)

translations
-*AOT-non-eq-E-infix* $\kappa\ \kappa'$ ==
CONST AOT-exe (*CONST relation-negation* (*CONST eq-E*)) (*CONST Pair* $\kappa\ \kappa'$)
⟨*ML*⟩
AOT-theorem *thm−neg=E*: ‹$x \neq_E y \equiv \neg(x =_E y)$›
⟨*proof*⟩

AOT-theorem *id−nec4:1*: ‹$x \neq_E y \equiv \Box(x \neq_E y)$›
⟨*proof*⟩

AOT-theorem *id−nec4:2*: ‹$\Diamond(x \neq_E y) \equiv (x \neq_E y)$›
⟨*proof*⟩

AOT-theorem *id−nec4:3*: ‹$\Diamond(x \neq_E y) \equiv \Box(x \neq_E y)$›
⟨*proof*⟩

AOT-theorem *id−act2:1*: ‹$x =_E y \equiv \boldsymbol{\mathcal{A}}x =_E y$›
⟨*proof*⟩
AOT-theorem *id−act2:2*: ‹$x \neq_E y \equiv \boldsymbol{\mathcal{A}}x \neq_E y$›
⟨*proof*⟩

AOT-theorem *ord=Eequiv:1*: ‹$O!x \to x =_E x$›
⟨*proof*⟩

AOT-theorem *ord=Eequiv:2*: ‹$x =_E y \to y =_E x$›
⟨*proof*⟩

AOT-theorem *ord=Eequiv:3*: ‹$(x =_E y \ \&\ y =_E z) \to x =_E z$›
⟨*proof*⟩

AOT-theorem *ord−=E=:1*: ‹$(O!x \lor O!y) \to \Box(x = y \equiv x =_E y)$›
⟨*proof*⟩

AOT-theorem *ord−=E=:2*: ‹$O!y \to [\lambda x\ x = y]\downarrow$›
⟨*proof*⟩

AOT-theorem *ord−=E=:3*: ‹$[\lambda xy\ O!x \ \&\ O!y \ \&\ x = y]\downarrow$›
⟨*proof*⟩

AOT-theorem *ind−nec*: ‹$\forall F\ ([F]x \equiv [F]y) \to \Box\forall F\ ([F]x \equiv [F]y)$›
⟨*proof*⟩

AOT-theorem *ord=E:1*: ‹$(O!x \ \&\ O!y) \to (\forall F\ ([F]x \equiv [F]y) \to x =_E y)$›
⟨*proof*⟩

AOT-theorem *ord=E:2*: ‹$(O!x \ \&\ O!y) \to (\forall F\ ([F]x \equiv [F]y) \to x = y)$›
⟨*proof*⟩

AOT-theorem *ord=E2:1*:
‹$(O!x \ \&\ O!y) \to (x \neq y \equiv [\lambda z\ z =_E x] \neq [\lambda z\ z =_E y])$›
⟨*proof*⟩

AOT-theorem *ord=E2:2*:
‹$(O!x \ \&\ O!y) \to (x \neq y \equiv [\lambda z\ z = x] \neq [\lambda z\ z = y])$›
⟨*proof*⟩

AOT-theorem *ordnecfail*: ‹$O!x \to \Box\neg\exists F\ x[F]$›
⟨*proof*⟩

AOT-theorem *ab−obey:1*: ‹$(A!x \ \&\ A!y) \to (\forall F\ (x[F] \equiv y[F]) \to x = y)$›
⟨*proof*⟩

AOT-theorem *ab−obey:2*:
‹$(\exists F\ (x[F]\ \&\ \neg y[F]) \vee \exists F\ (y[F]\ \&\ \neg x[F])) \rightarrow x \neq y$›
⟨*proof*⟩

AOT-theorem *encoders−are−abstract*: ‹$\exists F\ x[F] \rightarrow A!x$›
⟨*proof*⟩

AOT-theorem *denote=:1*: ‹$\forall H \exists x\ x[H]$›
⟨*proof*⟩

AOT-theorem *denote=:2*: ‹$\forall G \exists x_1 ... \exists x_n\ x_1 ... x_n[H]$›
⟨*proof*⟩

AOT-theorem *denote=:2[2]*: ‹$\forall G \exists x_1 \exists x_2\ x_1 x_2[H]$›
⟨*proof*⟩

AOT-theorem *denote=:2[3]*: ‹$\forall G \exists x_1 \exists x_2 \exists x_3\ x_1 x_2 x_3[H]$›
⟨*proof*⟩

AOT-theorem *denote=:2[4]*: ‹$\forall G \exists x_1 \exists x_2 \exists x_3 \exists x_4\ x_1 x_2 x_3 x_4[H]$›
⟨*proof*⟩

AOT-theorem *denote=:3*: ‹$\exists x\ x[\Pi] \equiv \exists H\ (H = \Pi)$›
⟨*proof*⟩

AOT-theorem *denote=:4*: ‹$(\exists x_1 ... \exists x_n\ x_1 ... x_n[\Pi]) \equiv \exists H\ (H = \Pi)$›
⟨*proof*⟩

AOT-theorem *denote=:4[2]*: ‹$(\exists x_1 \exists x_2\ x_1 x_2[\Pi]) \equiv \exists H\ (H = \Pi)$›
⟨*proof*⟩

AOT-theorem *denote=:4[3]*: ‹$(\exists x_1 \exists x_2 \exists x_3\ x_1 x_2 x_3[\Pi]) \equiv \exists H\ (H = \Pi)$›
⟨*proof*⟩

AOT-theorem *denote=:4[4]*: ‹$(\exists x_1 \exists x_2 \exists x_3 \exists x_4\ x_1 x_2 x_3 x_4[\Pi]) \equiv \exists H\ (H = \Pi)$›
⟨*proof*⟩

AOT-theorem *A−objects!*: ‹$\exists !x\ (A!x\ \&\ \forall F\ (x[F] \equiv \varphi\{F\}))$›
⟨*proof*⟩

AOT-theorem *obj−oth:1*: ‹$\exists !x\ (A!x\ \&\ \forall F\ (x[F] \equiv [F]y))$›
⟨*proof*⟩

AOT-theorem *obj−oth:2*: ‹$\exists !x\ (A!x\ \&\ \forall F\ (x[F] \equiv [F]y\ \&\ [F]z))$›
⟨*proof*⟩

AOT-theorem *obj−oth:3*: ‹$\exists !x\ (A!x\ \&\ \forall F\ (x[F] \equiv [F]y \vee [F]z))$›
⟨*proof*⟩

AOT-theorem *obj−oth:4*: ‹$\exists !x\ (A!x\ \&\ \forall F\ (x[F] \equiv \Box [F]y))$›
⟨*proof*⟩

AOT-theorem *obj−oth:5*: ‹$\exists !x\ (A!x\ \&\ \forall F\ (x[F] \equiv F = G))$›
⟨*proof*⟩

AOT-theorem *obj−oth:6*: ‹$\exists !x\ (A!x\ \&\ \forall F\ (x[F] \equiv \Box \forall y([G]y \rightarrow [F]y)))$›
⟨*proof*⟩

AOT-theorem *A−descriptions*: ‹$\iota x\ (A!x\ \&\ \forall F\ (x[F] \equiv \varphi\{F\}))\downarrow$›
⟨*proof*⟩

AOT-act-theorem *thm−can−terms2*:
‹$y = \iota x(A!x\ \&\ \forall F\ (x[F] \equiv \varphi\{F\})) \rightarrow (A!y\ \&\ \forall F\ (y[F] \equiv \varphi\{F\}))$›

⟨proof⟩

AOT-theorem *can−ab2*: ‹$y = \iota x(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\})) \rightarrow \ A!y$›
⟨proof⟩

AOT-act-theorem *desc−encode:1*: ‹$\iota x(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\}))[F] \equiv \varphi\{F\}$›
⟨proof⟩

AOT-act-theorem *desc−encode:2*: ‹$\iota x(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\}))[G] \equiv \varphi\{G\}$›
⟨proof⟩

AOT-theorem *desc−nec−encode:1*:
‹$\iota x \ (A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\}))[F] \equiv \boldsymbol{\mathcal{A}}\varphi\{F\}$›
⟨proof⟩

AOT-theorem *desc−nec−encode:2*:
‹$\iota x \ (A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\}))[G] \equiv \boldsymbol{\mathcal{A}}\varphi\{G\}$›
⟨proof⟩

AOT-theorem *Box−desc−encode:1*: ‹$\Box\varphi\{G\} \rightarrow \iota x(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{G\}))[G]$›
⟨proof⟩

AOT-theorem *Box−desc−encode:2*:
‹$\Box\varphi\{G\} \rightarrow \Box(\iota x(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{G\}))[G] \equiv \varphi\{G\})$›
⟨proof⟩

definition *rigid-condition* **where**
‹*rigid-condition* $\varphi \equiv \forall v \ . \ [v \models \forall \alpha \ (\varphi\{\alpha\} \rightarrow \Box\varphi\{\alpha\})]$›
syntax *rigid-condition* :: ‹*id-position* ⇒ *AOT-prop*› (‹*RIGID′-CONDITION′*(-′)›)

AOT-theorem *strict−can:1[E]*:
assumes ‹*RIGID-CONDITION*(φ)›
shows ‹$\forall \alpha \ (\varphi\{\alpha\} \rightarrow \Box\varphi\{\alpha\})$›
⟨proof⟩

AOT-theorem *strict−can:1[I]*:
assumes ‹$\vdash_\Box \forall \alpha \ (\varphi\{\alpha\} \rightarrow \Box\varphi\{\alpha\})$›
shows ‹*RIGID-CONDITION*(φ)›
⟨proof⟩

AOT-theorem *box−phi−a:1*:
assumes ‹*RIGID-CONDITION*(φ)›
shows ‹$(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\})) \rightarrow \Box(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\}))$›
⟨proof⟩

AOT-theorem *box−phi−a:2*:
assumes ‹*RIGID-CONDITION*(φ)›
shows ‹$y = \iota x(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\})) \rightarrow (A!y \ \& \ \forall F \ (y[F] \equiv \varphi\{F\}))$›
⟨proof⟩

AOT-theorem *box−phi−a:3*:
assumes ‹*RIGID-CONDITION*(φ)›
shows ‹$\iota x(A!x \ \& \ \forall F \ (x[F] \equiv \varphi\{F\}))[F] \equiv \varphi\{F\}$›
⟨proof⟩

AOT-define *Null* :: ‹$\tau \Rightarrow \varphi$› (‹*Null′*(-′)›)
df−null−uni:1: ‹$Null(x) \equiv_{df} A!x \ \& \ \neg\exists F \ x[F]$›

AOT-define *Universal* :: ‹$\tau \Rightarrow \varphi$› (‹*Universal′*(-′)›)
df−null−uni:2: ‹$Universal(x) \equiv_{df} A!x \ \& \ \forall F \ x[F]$›

AOT-theorem *null−uni−uniq:1*: ‹$\exists !x \ Null(x)$›
⟨proof⟩

AOT-theorem *null−uni−uniq*:*2*: ‹$\exists !x\ Universal(x)$›
⟨*proof*⟩

AOT-theorem *null−uni−uniq*:*3*: ‹$\iota x\ Null(x)\downarrow$›
⟨*proof*⟩

AOT-theorem *null−uni−uniq*:*4*: ‹$\iota x\ Universal(x)\downarrow$›
⟨*proof*⟩

AOT-define *Null-object* :: ‹κ_s› (‹$a_\emptyset$›)
df−null−uni−terms:*1*: ‹$a_\emptyset =_{df} \iota x\ Null(x)$›

AOT-define *Universal-object* :: ‹κ_s› (‹a_V›)
df−null−uni−terms:*2*: ‹$a_V =_{df} \iota x\ Universal(x)$›

AOT-theorem *null−uni−facts*:*1*: ‹$Null(x) \rightarrow \Box Null(x)$›
⟨*proof*⟩

AOT-theorem *null−uni−facts*:*2*: ‹$Universal(x) \rightarrow \Box Universal(x)$›
⟨*proof*⟩

AOT-theorem *null−uni−facts*:*3*: ‹$Null(a_\emptyset)$›
⟨*proof*⟩

AOT-theorem *null−uni−facts*:*4*: ‹$Universal(a_V)$›
⟨*proof*⟩

AOT-theorem *null−uni−facts*:*5*: ‹$a_\emptyset \neq a_V$›
⟨*proof*⟩

AOT-theorem *null−uni−facts*:*6*: ‹$a_\emptyset = \iota x(A!x\ \&\ \forall F\ (x[F] \equiv F \neq F))$›
⟨*proof*⟩

AOT-theorem *null−uni−facts*:*7*: ‹$a_V = \iota x(A!x\ \&\ \forall F\ (x[F] \equiv F = F))$›
⟨*proof*⟩

AOT-theorem *aclassical*:*1*:
‹$\forall R \exists x \exists y(A!x\ \&\ A!y\ \&\ x \neq y\ \&\ [\lambda z\ [R]zx] = [\lambda z\ [R]zy])$›
⟨*proof*⟩

AOT-theorem *aclassical*:*2*:
‹$\forall R \exists x \exists y(A!x\ \&\ A!y\ \&\ x \neq y\ \&\ [\lambda z\ [R]xz] = [\lambda z\ [R]yz])$›
⟨*proof*⟩

AOT-theorem *aclassical*:*3*:
‹$\forall F \exists x \exists y(A!x\ \&\ A!y\ \&\ x \neq y\ \&\ [\lambda\ [F]x] = [\lambda\ [F]y])$›
⟨*proof*⟩

AOT-theorem *aclassical2*: ‹$\exists x \exists y\ (A!x\ \&\ A!y\ \&\ x \neq y\ \&\ \forall F\ ([F]x \equiv [F]y))$›
⟨*proof*⟩

AOT-theorem *kirchner−thm*:*1*:
‹$[\lambda x\ \varphi\{x\}]\downarrow \equiv \Box \forall x \forall y(\forall F([F]x \equiv [F]y) \rightarrow (\varphi\{x\} \equiv \varphi\{y\}))$›
⟨*proof*⟩

AOT-theorem *kirchner−thm*:*2*:
‹$[\lambda x_1 ... x_n\ \varphi\{x_1 ... x_n\}]\downarrow \equiv \Box \forall x_1 ... \forall x_n \forall y_1 ... \forall y_n$
$(\forall F([F]x_1 ... x_n \equiv [F]y_1 ... y_n) \rightarrow (\varphi\{x_1 ... x_n\} \equiv \varphi\{y_1 ... y_n\}))$›
⟨*proof*⟩

AOT-theorem *kirchner−thm−cor*:*1*:
‹$[\lambda x\ \varphi\{x\}]\downarrow \rightarrow \forall x \forall y(\forall F([F]x \equiv [F]y) \rightarrow \Box(\varphi\{x\} \equiv \varphi\{y\}))$›

⟨*proof*⟩

AOT-theorem *kirchner−thm−cor:2*:
$‹[\lambda x_1...x_n\ \varphi\{x_1...x_n\}]\downarrow \rightarrow \forall x_1...\forall x_n \forall y_1...\forall y_n$
$(\forall F([F]x_1...x_n \equiv [F]y_1...y_n) \rightarrow \Box(\varphi\{x_1...x_n\} \equiv \varphi\{y_1...y_n\}))›$
⟨*proof*⟩

8.12 Propositional Properties

AOT-define *propositional* :: $‹\Pi \Rightarrow \varphi›$ (‹*Propositional*′(-′)›)
prop−prop1: $‹Propositional([F]) \equiv_{df} \exists p(F = [\lambda y\ p])›$

AOT-theorem *prop−prop2:1*: $‹\forall p\ [\lambda y\ p]\downarrow›$
⟨*proof*⟩

AOT-theorem *prop−prop2:2*: $‹[\lambda \nu\ \varphi]\downarrow›$
⟨*proof*⟩

AOT-theorem *prop−prop2:3*: $‹F = [\lambda y\ p] \rightarrow \Box\forall x([F]x \equiv p)›$
⟨*proof*⟩

AOT-theorem *prop−prop2:4*: $‹Propositional([F]) \rightarrow \Box Propositional([F])›$
⟨*proof*⟩

AOT-define *indicriminate* :: $‹\Pi \Rightarrow \varphi›$ (‹*Indiscriminate*′(-′)›)
prop−indis: $‹Indiscriminate([F]) \equiv_{df} F\downarrow\ \&\ \Box(\exists x\ [F]x \rightarrow \forall x\ [F]x)›$

AOT-theorem *prop−in−thm*: $‹Propositional([\Pi]) \rightarrow Indiscriminate([\Pi])›$
⟨*proof*⟩

AOT-theorem *prop−in−f:1*: $‹Necessary([F]) \rightarrow Indiscriminate([F])›$
⟨*proof*⟩

AOT-theorem *prop−in−f:2*: $‹Impossible([F]) \rightarrow Indiscriminate([F])›$
⟨*proof*⟩

AOT-theorem *prop−in−f:3:a*: $‹\neg Indiscriminate([E!])›$
⟨*proof*⟩

AOT-theorem *prop−in−f:3:b*: $‹\neg Indiscriminate([E!]^-)›$
⟨*proof*⟩

AOT-theorem *prop−in−f:3:c*: $‹\neg Indiscriminate(O!)›$
⟨*proof*⟩

AOT-theorem *prop−in−f:3:d*: $‹\neg Indiscriminate(A!)›$
⟨*proof*⟩

AOT-theorem *prop−in−f:4:a*: $‹\neg Propositional(E!)›$
⟨*proof*⟩

AOT-theorem *prop−in−f:4:b*: $‹\neg Propositional(E!^-)›$
⟨*proof*⟩

AOT-theorem *prop−in−f:4:c*: $‹\neg Propositional(O!)›$
⟨*proof*⟩

AOT-theorem *prop−in−f:4:d*: $‹\neg Propositional(A!)›$
⟨*proof*⟩

AOT-theorem *prop−prop−nec:1*: $‹\Diamond\exists p\ (F = [\lambda y\ p]) \rightarrow \exists p(F = [\lambda y\ p])›$
⟨*proof*⟩

AOT-theorem *prop−prop−nec:2*: ‹$\forall p\ (F \neq [\lambda y\ p]) \rightarrow \Box\forall p(F \neq [\lambda y\ p])$›
⟨*proof*⟩

AOT-theorem *prop−prop−nec:3*: ‹$\exists p\ (F = [\lambda y\ p]) \rightarrow \Box\exists p(F = [\lambda y\ p])$›
⟨*proof*⟩

AOT-theorem *prop−prop−nec:4*: ‹$\Diamond\forall p\ (F \neq [\lambda y\ p]) \rightarrow \forall p(F \neq [\lambda y\ p])$›
⟨*proof*⟩

AOT-theorem *enc−prop−nec:1*:
‹$\Diamond\forall F\ (x[F] \rightarrow \exists p(F = [\lambda y\ p])) \rightarrow \forall F(x[F] \rightarrow \exists p\ (F = [\lambda y\ p]))$›
⟨*proof*⟩

AOT-theorem *enc−prop−nec:2*:
‹$\forall F\ (x[F] \rightarrow \exists p(F = [\lambda y\ p])) \rightarrow \Box\forall F(x[F] \rightarrow \exists p\ (F = [\lambda y\ p]))$›
⟨*proof*⟩

9 Basic Logical Objects

AOT-define *TruthValueOf* :: ‹$\tau \Rightarrow \varphi \Rightarrow \varphi$› (‹*TruthValueOf*′(-,-′)›)
tv−p: ‹$TruthValueOf(x,p) \equiv_{df} A!x\ \&\ \forall F\ (x[F] \equiv \exists q((q \equiv p)\ \&\ F = [\lambda y\ q]))$›

AOT-theorem *p−has−!tv:1*: ‹$\exists x\ TruthValueOf(x,p)$›
⟨*proof*⟩

AOT-theorem *p−has−!tv:2*: ‹$\exists!x\ TruthValueOf(x,p)$›
⟨*proof*⟩

AOT-theorem *uni−tv*: ‹$\iota x\ TruthValueOf(x,p)\downarrow$›
⟨*proof*⟩

AOT-define *TheTruthValueOf* :: ‹$\varphi \Rightarrow \kappa_s$› (‹∘-› [*100*] *100*)
the−tv−p: ‹$\circ p =_{df} \iota x\ TruthValueOf(x,p)$›

AOT-define *PropEnc* :: ‹$\tau \Rightarrow \varphi \Rightarrow \varphi$› (**infixl** ‹**Σ**› *40*)
prop−enc: ‹$x\boldsymbol{\Sigma}p \equiv_{df} x\downarrow\ \&\ x[\lambda y\ p]$›

AOT-theorem *tv−id:1*: ‹$\circ p = \iota x\ (A!x\ \&\ \forall F\ (x[F] \equiv \exists q((q \equiv p)\ \&\ F = [\lambda y\ q])))$›
⟨*proof*⟩

AOT-theorem *tv−id:2*: ‹$\circ p\boldsymbol{\Sigma}p$›
⟨*proof*⟩

AOT-theorem *TV−lem1:1*:
‹$p \equiv \forall F(\exists q\ (q\ \&\ F = [\lambda y\ q]) \equiv \exists q((q \equiv p)\ \&\ F = [\lambda y\ q]))$›
⟨*proof*⟩

AOT-theorem *TV−lem1:2*:
‹$\neg p \equiv \forall F(\exists q\ (\neg q\ \&\ F = [\lambda y\ q]) \equiv \exists q((q \equiv p)\ \&\ F = [\lambda y\ q]))$›
⟨*proof*⟩

AOT-define *TruthValue* :: ‹$\tau \Rightarrow \varphi$› (‹*TruthValue*′(-′)›)
T−value: ‹$TruthValue(x) \equiv_{df} \exists p\ (TruthValueOf(x,p))$›

AOT-act-theorem *T−lem:1*: ‹$TruthValueOf(\circ p,\ p)$›
⟨*proof*⟩

AOT-act-theorem *T−lem:2*: ‹$\forall F\ (\circ p[F] \equiv \exists q((q \equiv p)\ \&\ F = [\lambda y\ q]))$›
⟨*proof*⟩

AOT-act-theorem *T−lem:3*: ‹$\circ p \boldsymbol{\Sigma} r \equiv (r \equiv p)$›
⟨*proof*⟩

AOT-act-theorem *T−lem:4*: ‹$\mathit{TruthValueOf}(x, p) \equiv x = \circ p$›
⟨*proof*⟩

AOT-theorem *TV−lem2:1*:
‹$(A!x\ \&\ \forall F\ (x[F] \equiv \exists q\ (q\ \&\ F = [\lambda y\ q]))) \rightarrow \mathit{TruthValue}(x)$›
⟨*proof*⟩

AOT-theorem *TV−lem2:2*:
‹$(A!x\ \&\ \forall F\ (x[F] \equiv \exists q\ (\neg q\ \&\ F = [\lambda y\ q]))) \rightarrow \mathit{TruthValue}(x)$›
⟨*proof*⟩

AOT-define *TheTrue* :: κ_s (‹$\top$›)
the−true:1: ‹$\top =_{df} \iota x\ (A!x\ \&\ \forall F\ (x[F] \equiv \exists p(p\ \&\ F = [\lambda y\ p])))$›
AOT-define *TheFalse* :: κ_s (‹$\bot$›)
the−true:2: ‹$\bot =_{df} \iota x\ (A!x\ \&\ \forall F\ (x[F] \equiv \exists p(\neg p\ \&\ F = [\lambda y\ p])))$›

AOT-theorem *the−true:3*: ‹$\top \neq \bot$›
⟨*proof*⟩

AOT-act-theorem *T−T−value:1*: ‹$\mathit{TruthValue}(\top)$›
⟨*proof*⟩

AOT-act-theorem *T−T−value:2*: ‹$\mathit{TruthValue}(\bot)$›
⟨*proof*⟩

AOT-theorem *two−T*: ‹$\exists x \exists y(\mathit{TruthValue}(x)\ \&\ \mathit{TruthValue}(y)\ \&\ x \neq y\ \&$
$\forall z\ (\mathit{TruthValue}(z) \rightarrow z = x \vee z = y))$›
⟨*proof*⟩

AOT-act-theorem *valueof−facts:1*: ‹$\mathit{TruthValueOf}(x, p) \rightarrow (p \equiv x = \top)$›
⟨*proof*⟩

AOT-act-theorem *valueof−facts:2*: ‹$\mathit{TruthValueOf}(x, p) \rightarrow (\neg p \equiv x = \bot)$›
⟨*proof*⟩

AOT-act-theorem *q−True:1*: ‹$p \equiv (\circ p = \top)$›
⟨*proof*⟩

AOT-act-theorem *q−True:2*: ‹$\neg p \equiv (\circ p = \bot)$›
⟨*proof*⟩

AOT-act-theorem *q−True:3*: ‹$p \equiv \top \boldsymbol{\Sigma} p$›
⟨*proof*⟩

AOT-act-theorem *q−True:5*: ‹$\neg p \equiv \bot \boldsymbol{\Sigma} p$›
⟨*proof*⟩

AOT-act-theorem *q−True:4*: ‹$p \equiv \neg(\bot \boldsymbol{\Sigma} p)$›
⟨*proof*⟩

AOT-act-theorem *q−True:6*: ‹$\neg p \equiv \neg(\top \boldsymbol{\Sigma} p)$›

⟨*proof*⟩

AOT-define *ExtensionOf* :: ‹$\tau \Rightarrow \varphi \Rightarrow \varphi$› (‹*ExtensionOf*'(-,-')›)
exten−p: ‹*ExtensionOf*$(x,p) \equiv_{df}$ $A!x$ &
$\forall F\ (x[F] \rightarrow$ *Propositional*$([F]))$ &
$\forall q\ ((x\boldsymbol{\Sigma}q) \equiv (q \equiv p))$›

AOT-theorem *extof−e*: ‹*ExtensionOf*$(x,\ p) \equiv$ *TruthValueOf*$(x,\ p)$›
⟨*proof*⟩

AOT-theorem *ext−p−tv*:*1*: ‹$\exists!x$ *ExtensionOf*$(x,\ p)$›
⟨*proof*⟩

AOT-theorem *ext−p−tv*:*2*: ‹$\iota x($*ExtensionOf*$(x,\ p))\downarrow$›
⟨*proof*⟩

AOT-theorem *ext−p−tv*:*3*: ‹$\iota x($*ExtensionOf*$(x,\ p)) = \circ p$›
⟨*proof*⟩

10 Restricted Variables

locale *AOT-restriction-condition* =
fixes ψ :: ‹'a::*AOT-Term-id-2* ⇒ o›
assumes *res−var*:*2*[*AOT*]: ‹$[v \models \exists\alpha\ \psi\{\alpha\}]$›
assumes *res−var*:*3*[*AOT*]: ‹$[v \models \psi\{\tau\} \rightarrow \tau\downarrow]$›

⟨*ML*⟩

locale *AOT-rigid-restriction-condition* = *AOT-restriction-condition* +
assumes *rigid*[*AOT*]: ‹$[v \models \forall\alpha(\psi\{\alpha\} \rightarrow \Box\psi\{\alpha\})]$›
begin
lemma *rigid-condition*[*AOT*]: ‹$[v \models \Box(\psi\{\alpha\} \rightarrow \Box\psi\{\alpha\})]$›
⟨*proof*⟩
lemma *type-set-nonempty*[*AOT-no-atp*, *no-atp*]: ‹$\exists x\ .\ x \in \{\ \alpha\ .\ [w_0 \models \psi\{\alpha\}]\}$›
⟨*proof*⟩
end

locale *AOT-restricted-type* = *AOT-rigid-restriction-condition* +
fixes *Rep* **and** *Abs*
assumes *AOT-restricted-type-definition*[*AOT-no-atp*]:
‹*type-definition Rep Abs* $\{\ \alpha\ .\ [w_0 \models \psi\{\alpha\}]\}$›
begin

AOT-theorem *restricted-var-condition*: ‹$\psi\{$«*AOT-term-of-var* (*Rep* α)»$\}$›
⟨*proof*⟩
lemmas ψ = *restricted-var-condition*

AOT-theorem *GEN*: **assumes** ‹*for arbitrary* α: $\varphi\{$«*AOT-term-of-var* (*Rep* α)»$\}$›
shows ‹$\forall\alpha\ (\psi\{\alpha\} \rightarrow \varphi\{\alpha\})$›
⟨*proof*⟩
lemmas $\forall I$ = *GEN*

end

lemma *AOT-restricted-type-intro*[*AOT-no-atp*, *no-atp*]:
assumes ‹*type-definition Rep Abs* $\{\ \alpha\ .\ [w_0 \models \psi\{\alpha\}]\}$›
and ‹*AOT-rigid-restriction-condition* ψ›
shows ‹*AOT-restricted-type* ψ *Rep Abs*›
⟨*proof*⟩

⟨*ML*⟩

context *AOT-restricted-type*
begin

AOT-theorem *rule−ui*:
  assumes ‹$\forall\,\alpha(\psi\{\alpha\} \to \varphi\{\alpha\})$›
  shows ‹$\varphi\{$«*AOT-term-of-var* (*Rep* α)»$\}$›
⟨*proof*⟩
lemmas $\forall E$ = *rule−ui*

AOT-theorem *instantiation*:
  assumes ‹*for arbitrary* β: $\varphi\{$«*AOT-term-of-var* (*Rep* β)»$\} \vdash \chi$› **and** ‹$\exists\,\alpha\ (\psi\{\alpha\}\ \&\ \varphi\{\alpha\})$›
  shows ‹χ›
⟨*proof*⟩
lemmas $\exists E$ = *instantiation*

AOT-theorem *existential*: **assumes** ‹$\varphi\{$«*AOT-term-of-var* (*Rep* β)»$\}$›
  shows ‹$\exists\ \alpha\ (\psi\{\alpha\}\ \&\ \varphi\{\alpha\})$›
  ⟨*proof*⟩
lemmas $\exists I$ = *existential*
end

context *AOT-rigid-restriction-condition*
begin

AOT-theorem *res−var−bound−reas*[*1*]:
  ‹$\forall\,\alpha(\psi\{\alpha\} \to \forall\,\beta\ \varphi\{\alpha, \beta\}) \equiv \forall\,\beta\forall\,\alpha\ (\psi\{\alpha\} \to \varphi\{\alpha, \beta\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*BF*]:
  ‹$\forall\,\alpha(\psi\{\alpha\} \to \Box\varphi\{\alpha\}) \to \Box\forall\,\alpha(\psi\{\alpha\} \to \varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*CBF*]:
‹$\Box\forall\,\alpha(\psi\{\alpha\} \to \varphi\{\alpha\}) \to \forall\,\alpha(\psi\{\alpha\} \to \Box\varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*2*]:
‹$\forall\,\alpha\ (\psi\{\alpha\} \to \boldsymbol{\mathcal{A}}\varphi\{\alpha\}) \to \boldsymbol{\mathcal{A}}\forall\,\alpha\ (\psi\{\alpha\} \to \varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*3*]:
  ‹$\boldsymbol{\mathcal{A}}\forall\,\alpha\ (\psi\{\alpha\} \to \varphi\{\alpha\}) \to \forall\,\alpha\ (\psi\{\alpha\} \to \boldsymbol{\mathcal{A}}\varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*Buridan*]:
  ‹$\exists\,\alpha\ (\psi\{\alpha\}\ \&\ \Box\varphi\{\alpha\}) \to \Box\exists\,\alpha\ (\psi\{\alpha\}\ \&\ \varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*BF*$\Diamond$]:
  ‹$\Diamond\exists\,\alpha\ (\psi\{\alpha\}\ \&\ \varphi\{\alpha\}) \to \exists\,\alpha\ (\psi\{\alpha\}\ \&\ \Diamond\varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*CBF*$\Diamond$]:
  ‹$\exists\,\alpha\ (\psi\{\alpha\}\ \&\ \Diamond\varphi\{\alpha\}) \to \Diamond\exists\,\alpha\ (\psi\{\alpha\}\ \&\ \varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *res−var−bound−reas*[*A−Exists*:*1*]:

‹$\boldsymbol{\mathcal{A}}\exists!\alpha\ (\psi\{\alpha\}\ \&\ \varphi\{\alpha\}) \equiv \exists!\alpha\ (\psi\{\alpha\}\ \&\ \boldsymbol{\mathcal{A}}\varphi\{\alpha\})$›
⟨*proof*⟩

end

theory *AOT-ExtendedRelationComprehension*
imports *AOT-RestrictedVariables*
begin

11 Extended Relation Comprehension

This theory depends on choosing extended models.

interpretation *AOT-ExtendedModel* ⟨*proof*⟩

Auxiliary lemma: negations of denoting relations denote.

AOT-theorem *negation-denotes*: ‹$[\lambda x\ \varphi\{x\}]\downarrow \rightarrow [\lambda x\ \neg\varphi\{x\}]\downarrow$›
⟨*proof*⟩

Auxiliary lemma: conjunctions of denoting relations denote.

AOT-theorem *conjunction-denotes*: ‹$[\lambda x\ \varphi\{x\}]\downarrow\ \&\ [\lambda x\ \psi\{x\}]\downarrow \rightarrow [\lambda x\ \varphi\{x\}\ \&\ \psi\{x\}]\downarrow$›
⟨*proof*⟩

AOT-register-rigid-restricted-type
Ordinary: ‹$O!\kappa$›
⟨*proof*⟩

AOT-register-variable-names
Ordinary: *u v r t s*

In PLM this is defined in the Natural Numbers chapter, but since it is helpful for stating the comprehension principles, we already define it here.

AOT-define *eqE* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (**infixl** ‹$\equiv_E$› *50*)
eqE: ‹$F \equiv_E G \equiv_{df} F\downarrow\ \&\ G\downarrow\ \&\ \forall u\ ([F]u \equiv [G]u)$›

Derive existence claims about relations from the axioms.

AOT-theorem *denotes-all*: ‹$[\lambda x\ \forall G\ (\Box G \equiv_E F \rightarrow x[G])]\downarrow$›
and *denotes-all-neg*: ‹$[\lambda x\ \forall G\ (\Box G \equiv_E F \rightarrow \neg x[G])]\downarrow$›
⟨*proof*⟩

Reformulate the existence claims in terms of their negations.

AOT-theorem *denotes-ex*: ‹$[\lambda x\ \exists G\ (\Box G \equiv_E F\ \&\ x[G])]\downarrow$›
⟨*proof*⟩

AOT-theorem *denotes-ex-neg*: ‹$[\lambda x\ \exists G\ (\Box G \equiv_E F\ \&\ \neg x[G])]\downarrow$›
⟨*proof*⟩

Derive comprehension principles.

AOT-theorem *Comprehension-1*:
shows ‹$\Box\forall F\forall G(\Box G \equiv_E F \rightarrow (\varphi\{F\} \equiv \varphi\{G\})) \rightarrow [\lambda x\ \exists F\ (\varphi\{F\}\ \&\ x[F])]\downarrow$›
⟨*proof*⟩

AOT-theorem *Comprehension-2*:
shows ‹$\Box\forall F\forall G(\Box G \equiv_E F \rightarrow (\varphi\{F\} \equiv \varphi\{G\})) \rightarrow [\lambda x\ \exists F\ (\varphi\{F\}\ \&\ \neg x[F])]\downarrow$›
⟨*proof*⟩

Derived variants of the comprehension principles above.

AOT-theorem *Comprehension-1′*:
shows ‹$\Box\forall F\forall G(\Box G \equiv_E F \rightarrow (\varphi\{F\} \equiv \varphi\{G\})) \rightarrow [\lambda x\ \forall F\ (x[F] \rightarrow \varphi\{F\})]\downarrow$›
⟨*proof*⟩

AOT-theorem *Comprehension-2′*:
shows ‹$\Box\forall F\forall G(\Box G \equiv_E F \to (\varphi\{F\} \equiv \varphi\{G\})) \to [\lambda x\, \forall F\, (\varphi\{F\} \to x[F])]\downarrow$›
⟨*proof*⟩

Derive a combined comprehension principles.

AOT-theorem *Comprehension-3*:
‹$\Box\forall F\forall G(\Box G \equiv_E F \to (\varphi\{F\} \equiv \varphi\{G\})) \to [\lambda x\, \forall F\, (x[F] \equiv \varphi\{F\})]\downarrow$›
⟨*proof*⟩

notepad
begin

Verify that the original axioms are equivalent to $\vdash_\Box [\lambda x\, \exists G\, (\Box G \equiv_E F\ \&\ x[G])]\downarrow$ and $\vdash_\Box [\lambda x\, \exists G\, (\Box G \equiv_E F\ \&\ \neg x[G])]\downarrow$.

⟨*proof*⟩
end
end

12 Possible Worlds

AOT-define *Situation* :: ‹$\tau \Rightarrow \varphi$› (‹*Situation*′(-′)›)
situations: ‹$Situation(x) \equiv_{df} A!x\ \&\ \forall F\, (x[F] \to Propositional([F]))$›

AOT-theorem *T−sit*: ‹$TruthValue(x) \to Situation(x)$›
⟨*proof*⟩

AOT-theorem *possit−sit:1*: ‹$Situation(x) \equiv \Box Situation(x)$›
⟨*proof*⟩

AOT-theorem *possit−sit:2*: ‹$\Diamond Situation(x) \equiv Situation(x)$›
⟨*proof*⟩

AOT-theorem *possit−sit:3*: ‹$\Diamond Situation(x) \equiv \Box Situation(x)$›
⟨*proof*⟩

AOT-theorem *possit−sit:4*: ‹$\mathcal{A} Situation(x) \equiv Situation(x)$›
⟨*proof*⟩

AOT-theorem *possit−sit:5*: ‹$Situation(\circ p)$›
⟨*proof*⟩

AOT-theorem *possit−sit:6*: ‹$Situation(\top)$›
⟨*proof*⟩

AOT-theorem *possit−sit:7*: ‹$Situation(\bot)$›
⟨*proof*⟩

AOT-register-rigid-restricted-type
Situation: ‹$Situation(\kappa)$›
⟨*proof*⟩

AOT-register-variable-names
Situation: *s*

AOT-define *TruthInSituation* :: ‹$\tau \Rightarrow \varphi \Rightarrow \varphi$› (‹(- ⊨/ -)› [*100*, *40*] *100*)
true−in−s: ‹$s \models p \equiv_{df} s\mathbf{\Sigma}p$›

notepad
begin

⟨*proof*⟩
end

AOT-theorem *lem1*: ‹$Situation(x) \rightarrow (x \models p \equiv x[\lambda y\ p])$›
⟨*proof*⟩

AOT-theorem *lem2:1*: ‹$s \models p \equiv \Box s \models p$›
⟨*proof*⟩

AOT-theorem *lem2:2*: ‹$\Diamond s \models p \equiv s \models p$›
⟨*proof*⟩

AOT-theorem *lem2:3*: ‹$\Diamond s \models p \equiv \Box s \models p$›
⟨*proof*⟩

AOT-theorem *lem2:4*: ‹$\boldsymbol{\mathcal{A}}(s \models p) \equiv s \models p$›
⟨*proof*⟩

AOT-theorem *lem2:5*: ‹$\neg s \models p \equiv \Box\neg s \models p$›
⟨*proof*⟩

AOT-theorem *sit−identity*: ‹$s = s' \equiv \forall p(s \models p \equiv s' \models p)$›
⟨*proof*⟩

AOT-define *PartOfSituation* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (**infixl** ‹$\unlhd$› *80*)
sit−part−whole: ‹$s \unlhd s' \equiv_{df} \forall p\ (s \models p \rightarrow s' \models p)$›

AOT-theorem *part:1*: ‹$s \unlhd s$›
⟨*proof*⟩

AOT-theorem *part:2*: ‹$s \unlhd s'\ \&\ s \neq s' \rightarrow \neg(s' \unlhd s)$›
⟨*proof*⟩

AOT-theorem *part:3*: ‹$s \unlhd s'\ \&\ s' \unlhd s'' \rightarrow s \unlhd s''$›
⟨*proof*⟩

AOT-theorem *sit−identity2:1*: ‹$s = s' \equiv s \unlhd s'\ \&\ s' \unlhd s$›
⟨*proof*⟩

AOT-theorem *sit−identity2:2*: ‹$s = s' \equiv \forall s''\ (s'' \unlhd s \equiv s'' \unlhd s')$›
⟨*proof*⟩

AOT-define *Persistent* :: ‹$\varphi \Rightarrow \varphi$› (‹*Persistent*′(-)′›)
persistent: ‹$Persistent(p) \equiv_{df} \forall s\ (s \models p \rightarrow \forall s'\ (s \unlhd s' \rightarrow s' \models p))$›

AOT-theorem *pers−prop*: ‹$\forall p\ Persistent(p)$›
⟨*proof*⟩

AOT-define *NullSituation* :: ‹$\tau \Rightarrow \varphi$› (‹*NullSituation*′(-)′›)
df−null−trivial:1: ‹$NullSituation(s) \equiv_{df} \neg\exists p\ s \models p$›

AOT-define *TrivialSituation* :: ‹$\tau \Rightarrow \varphi$› (‹*TrivialSituation*′(-)′›)
df−null−trivial:2: ‹$TrivialSituation(s) \equiv_{df} \forall p\ s \models p$›

AOT-theorem *thm−null−trivial:1*: ‹$\exists!x\ NullSituation(x)$›
⟨*proof*⟩

AOT-theorem *thm−null−trivial:2*: ‹$\exists!x\ TrivialSituation(x)$›
⟨*proof*⟩

AOT-theorem *thm−null−trivial:3*: ‹$\iota x\ NullSituation(x)\downarrow$›
⟨*proof*⟩

AOT-theorem *thm−null−trivial:4*: ‹ιx *TrivialSituation*(x)↓›
⟨*proof*⟩

AOT-define *TheNullSituation* :: ‹κ_s› (‹$\mathbf{s}_\emptyset$›)
df−the−null−sit:1: ‹$\mathbf{s}_\emptyset =_{df} \iota x$ *NullSituation*(x)›

AOT-define *TheTrivialSituation* :: ‹κ_s› (‹$\mathbf{s}_V$›)
df−the−null−sit:2: ‹$\mathbf{s}_V =_{df} \iota x$ *TrivialSituation*(x)›

AOT-theorem *null−triv−sc:1*: ‹*NullSituation*(x) → □*NullSituation*(x)›
⟨*proof*⟩

AOT-theorem *null−triv−sc:2*: ‹*TrivialSituation*(x) → □*TrivialSituation*(x)›
⟨*proof*⟩

AOT-theorem *null−triv−sc:3*: ‹*NullSituation*($\mathbf{s}_\emptyset$)›
⟨*proof*⟩

AOT-theorem *null−triv−sc:4*: ‹*TrivialSituation*($\mathbf{s}_V$)›
⟨*proof*⟩

AOT-theorem *null−triv−facts:1*: ‹*NullSituation*(x) ≡ *Null*(x)›
⟨*proof*⟩

AOT-theorem *null−triv−facts:2*: ‹$\mathbf{s}_\emptyset = a_\emptyset$›
⟨*proof*⟩

AOT-theorem *null−triv−facts:3*: ‹$\mathbf{s}_V \neq a_V$›
⟨*proof*⟩

definition *ConditionOnPropositionalProperties* :: ‹(<κ> ⇒ o) ⇒ *bool*› **where**
cond−prop: ‹*ConditionOnPropositionalProperties* ≡ λ φ . ∀ *v* .
[*v* ⊨ ∀ *F* (φ{*F*} → *Propositional*([*F*]))]›

syntax *ConditionOnPropositionalProperties* :: ‹*id-position* ⇒ *AOT-prop*›
(‹*CONDITION′-ON′-PROPOSITIONAL′-PROPERTIES′*(-′)›)

AOT-theorem *cond−prop*[*E*]:
assumes ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
shows ‹∀ *F* (φ{*F*} → *Propositional*([*F*]))›
⟨*proof*⟩

AOT-theorem *cond−prop*[*I*]:
assumes ‹$\vdash_\Box$ ∀ *F* (φ{*F*} → *Propositional*([*F*]))›
shows ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
⟨*proof*⟩

AOT-theorem *pre−comp−sit*:
assumes ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
shows ‹(*Situation*(x) & ∀ *F* (x[*F*] ≡ φ{*F*})) ≡ (*A*!x & ∀ *F* (x[*F*] ≡ φ{*F*}))›
⟨*proof*⟩

AOT-theorem *comp−sit:1*:
assumes ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
shows ‹∃ *s* ∀ *F*(*s*[*F*] ≡ φ{*F*})›
⟨*proof*⟩

AOT-theorem *comp−sit:2*:
assumes ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
shows ‹∃ !*s* ∀ *F*(*s*[*F*] ≡ φ{*F*})›
⟨*proof*⟩

AOT-theorem *can−sit−desc*:*1*:
assumes ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
shows ‹$\iota s(\forall F\ (s[F] \equiv \varphi\{F\}))\downarrow$›
⟨*proof*⟩

AOT-theorem *can−sit−desc*:*2*:
assumes ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
shows ‹$\iota s(\forall F\ (s[F] \equiv \varphi\{F\})) = \iota x(A!x\ \&\ \forall F\ (x[F] \equiv \varphi\{F\}))$›
⟨*proof*⟩

AOT-theorem *strict−sit*:
assumes ‹*RIGID-CONDITION*(φ)›
and ‹*CONDITION-ON-PROPOSITIONAL-PROPERTIES*(φ)›
shows ‹$y = \iota s(\forall F\ (s[F] \equiv \varphi\{F\})) \rightarrow \forall F\ (y[F] \equiv \varphi\{F\})$›
⟨*proof*⟩

AOT-define *actual* :: ‹$\tau \Rightarrow \varphi$› (‹*Actual*′(-′)›)
‹$Actual(s) \equiv_{df} \forall p\ (s \models p \rightarrow p)$›

AOT-theorem *act−and−not−pos*: ‹$\exists s\ (Actual(s)\ \&\ \Diamond\neg Actual(s))$›
⟨*proof*⟩

AOT-theorem *actual−s*:*1*: ‹$\exists s\ Actual(s)$›
⟨*proof*⟩

AOT-theorem *actual−s*:*2*: ‹$\exists s\ \neg Actual(s)$›
⟨*proof*⟩

AOT-theorem *actual−s*:*3*: ‹$\exists p \forall s(Actual(s) \rightarrow \neg s \models p)$›
⟨*proof*⟩

AOT-theorem *comp*:
‹$\exists s\ (s' \trianglelefteq s\ \&\ s'' \trianglelefteq s\ \&\ \forall s'''\ (s' \trianglelefteq s'''\ \&\ s'' \trianglelefteq s''' \rightarrow s \trianglelefteq s'''))$›
⟨*proof*⟩

AOT-theorem *act−sit*:*1*: ‹$Actual(s) \rightarrow (s \models p \rightarrow [\lambda y\ p]s)$›
⟨*proof*⟩

AOT-theorem *act−sit*:*2*:
‹$(Actual(s')\ \&\ Actual(s'')) \rightarrow \exists x\ (Actual(x)\ \&\ s' \trianglelefteq x\ \&\ s'' \trianglelefteq x)$›
⟨*proof*⟩

AOT-define *Consistent* :: ‹$\tau \Rightarrow \varphi$› (‹*Consistent*′(-′)›)
cons: ‹$Consistent(s) \equiv_{df} \neg\exists p\ (s \models p\ \&\ s \models \neg p)$›

AOT-theorem *sit−cons*: ‹$Actual(s) \rightarrow Consistent(s)$›
⟨*proof*⟩

AOT-theorem *cons−rigid*:*1*: ‹$\neg Consistent(s) \equiv \Box\neg Consistent(s)$›
⟨*proof*⟩

AOT-theorem *cons−rigid*:*2*: ‹$\Diamond Consistent(x) \equiv Consistent(x)$›
⟨*proof*⟩

AOT-define *possible* :: ‹$\tau \Rightarrow \varphi$› (‹*Possible*′(-′)›)
pos: ‹$Possible(s) \equiv_{df} \Diamond Actual(s)$›

AOT-theorem *sit−pos*:*1*: ‹$Actual(s) \rightarrow Possible(s)$›
⟨*proof*⟩

AOT-theorem *sit−pos*:*2*: ‹$\exists p\ ((s \models p)\ \&\ \neg\Diamond p) \rightarrow \neg Possible(s)$›

⟨*proof*⟩

AOT-theorem *pos−cons−sit:1*: ‹$Possible(s) \rightarrow Consistent(s)$›
⟨*proof*⟩

AOT-theorem *pos−cons−sit:2*: ‹$\exists\, s\ (Consistent(s)\ \&\ \neg Possible(s))$›
⟨*proof*⟩

AOT-theorem *sit−classical:1*: ‹$\forall\, p\ (s \models p \equiv p) \rightarrow \forall\, q(s \models \neg q \equiv \neg s \models q)$›
⟨*proof*⟩

AOT-theorem *sit−classical:2*:
‹$\forall\, p\ (s \models p \equiv p) \rightarrow \forall\, q \forall\, r((s \models (q \rightarrow r)) \equiv (s \models q \rightarrow s \models r))$›
⟨*proof*⟩

AOT-theorem *sit−classical:3*:
‹$\forall\, p\ (s \models p \equiv p) \rightarrow ((s \models \forall\, \alpha\ \varphi\{\alpha\}) \equiv \forall\, \alpha\ s \models \varphi\{\alpha\})$›
⟨*proof*⟩

AOT-theorem *sit−classical:4*: ‹$\forall\, p\ (s \models p \equiv p) \rightarrow \forall\, q\ (s \models \Box q \rightarrow \Box s \models q)$›
⟨*proof*⟩

AOT-theorem *sit−classical:5*:
‹$\forall\, p\ (s \models p \equiv p) \rightarrow \exists\, q(\Box(s \models q)\ \&\ \neg(s \models \Box\ q))$›
⟨*proof*⟩

AOT-theorem *sit−classical:6*:
‹$\exists\, s\ \forall\, p\ (s \models p \equiv p)$›
⟨*proof*⟩

AOT-define *PossibleWorld* :: ‹$\tau \Rightarrow \varphi$› (‹*PossibleWorld*′(-′)›)
world:1: ‹$PossibleWorld(x) \equiv_{df} Situation(x)\ \&\ \Diamond\forall\, p(x \models p \equiv p)$›

AOT-theorem *world:2*: ‹$\exists\, x\ PossibleWorld(x)$›
⟨*proof*⟩

AOT-theorem *world:3*: ‹$PossibleWorld(\kappa) \rightarrow \kappa{\downarrow}$›
⟨*proof*⟩

AOT-theorem *rigid−pw:1*: ‹$PossibleWorld(x) \equiv \Box PossibleWorld(x)$›
⟨*proof*⟩

AOT-theorem *rigid−pw:2*: ‹$\Diamond PossibleWorld(x) \equiv PossibleWorld(x)$›
⟨*proof*⟩

AOT-theorem *rigid−pw:3*: ‹$\Diamond PossibleWorld(x) \equiv \Box PossibleWorld(x)$›
⟨*proof*⟩

AOT-theorem *rigid−pw:4*: ‹$\boldsymbol{\mathcal{A}} PossibleWorld(x) \equiv PossibleWorld(x)$›
⟨*proof*⟩

AOT-register-rigid-restricted-type
PossibleWorld: ‹$PossibleWorld(\kappa)$›
⟨*proof*⟩
AOT-register-variable-names
PossibleWorld: *w*

AOT-theorem *world−pos*: ‹$Possible(w)$›
⟨*proof*⟩

AOT-theorem *world−cons:1*: ‹$Consistent(w)$›
⟨*proof*⟩

AOT-theorem *world−cons:2*: ‹$\neg$ *TrivialSituation*(w)›
⟨*proof*⟩

AOT-theorem *rigid−truth−at:1*: ‹$w \models p \equiv \Box w \models p$›
⟨*proof*⟩

AOT-theorem *rigid−truth−at:2*: ‹$\Diamond w \models p \equiv w \models p$›
⟨*proof*⟩

AOT-theorem *rigid−truth−at:3*: ‹$\Diamond w \models p \equiv \Box w \models p$›
⟨*proof*⟩

AOT-theorem *rigid−truth−at:4*: ‹$\boldsymbol{\mathcal{A}} w \models p \equiv w \models p$›
⟨*proof*⟩

AOT-theorem *rigid−truth−at:5*: ‹$\neg w \models p \equiv \Box\neg w \models p$›
⟨*proof*⟩

AOT-define *Maximal* :: ‹$\tau \Rightarrow \varphi$› (‹*Maximal*′(-′)›)
max: ‹*Maximal*(s) $\equiv_{df}$ $\forall p$ ($s \models p \vee s \models \neg p$)›

AOT-theorem *world−max*: ‹*Maximal*(w)›
⟨*proof*⟩

AOT-theorem *world=maxpos:1*: ‹*Maximal*(x) $\rightarrow$ $\Box$*Maximal*(x)›
⟨*proof*⟩

AOT-theorem *world=maxpos:2*: ‹*PossibleWorld*(x) $\equiv$ *Maximal*(x) & *Possible*(x)›
⟨*proof*⟩

AOT-define *NecImpl* :: ‹$\varphi \Rightarrow \varphi \Rightarrow \varphi$› (**infixl** ‹⇒› *26*)
nec−impl−p:1: ‹$p \Rightarrow q \equiv_{df} \Box(p \rightarrow q)$›
AOT-define *NecEquiv* :: ‹$\varphi \Rightarrow \varphi \Rightarrow \varphi$› (**infixl** ‹⇔› *21*)
nec−impl−p:2: ‹$p \Leftrightarrow q \equiv_{df} (p \Rightarrow q)$ & $(q \Rightarrow p)$›

AOT-theorem *nec−equiv−nec−im*: ‹$p \Leftrightarrow q \equiv \Box(p \equiv q)$›
⟨*proof*⟩

AOT-theorem *world-closed-lem-1-a*:
‹($s \models (\varphi$ & $\psi)) \rightarrow (\forall p\ (s \models p \equiv p) \rightarrow (s \models \varphi$ & $s \models \psi$))›
⟨*proof*⟩

AOT-theorem *world-closed-lem-1-b*:
‹($s \models \varphi$ & $(\varphi \rightarrow q)) \rightarrow (\forall p\ (s \models p \equiv p) \rightarrow s \models q)$›
⟨*proof*⟩

AOT-theorem *world-closed-lem-1-c*:
‹($s \models \varphi$ & $s \models (\varphi \rightarrow \psi)) \rightarrow (\forall p\ (s \models p \equiv p) \rightarrow s \models \psi)$›
⟨*proof*⟩

AOT-theorem *world−closed−lem:1*[*0*]:
‹$q \rightarrow (\forall p\ (s \models p \equiv p) \rightarrow s \models q)$›
⟨*proof*⟩

AOT-theorem *world−closed−lem:1*[*1*]:
‹$s \models p_1$ & $(p_1 \rightarrow q) \rightarrow (\forall p\ (s \models p \equiv p) \rightarrow s \models q)$›
⟨*proof*⟩

AOT-theorem *world−closed−lem:1*[*2*]:
‹$s \models p_1$ & $s \models p_2$ & $((p_1$ & $p_2) \rightarrow q) \rightarrow (\forall p\ (s \models p \equiv p) \rightarrow s \models q)$›
⟨*proof*⟩

AOT-theorem *world−closed−lem:1[3]*:
‹$s \models p_1$ & $s \models p_2$ & $s \models p_3$ & $((p_1$ & p_2 & $p_3) \rightarrow q) \rightarrow (\forall p\ (s \models p \equiv p) \rightarrow s \models q)$›
⟨*proof*⟩

AOT-theorem *world−closed−lem:1[4]*:
‹$s \models p_1$ & $s \models p_2$ & $s \models p_3$ & $s \models p_4$ & $((p_1$ & p_2 & p_3 & $p_4) \rightarrow q) \rightarrow$
$(\forall p\ (s \models p \equiv p) \rightarrow s \models q)$›
⟨*proof*⟩

AOT-theorem *coherent:1*: ‹$w \models \neg p \equiv \neg w \models p$›
⟨*proof*⟩

AOT-theorem *coherent:2*: ‹$w \models p \equiv \neg w \models \neg p$›
⟨*proof*⟩

AOT-theorem *act−world:1*: ‹$\exists w\ \forall p\ (w \models p \equiv p)$›
⟨*proof*⟩

AOT-theorem *act−world:2*: ‹$\exists! w\ Actual(w)$›
⟨*proof*⟩

AOT-theorem *pre−walpha*: ‹$\iota w\ Actual(w)\downarrow$›
⟨*proof*⟩

AOT-define *TheActualWorld* :: ‹κ_s› (‹$\mathbf{w}_\alpha$›)
w−alpha: ‹$\mathbf{w}_\alpha =_{df} \iota w\ Actual(w)$›

AOT-theorem *true-in-truth-act-true*: ‹$\top \models p \equiv \mathcal{A}p$›
⟨*proof*⟩

AOT-theorem *T−world*: ‹$\top = \mathbf{w}_\alpha$›
⟨*proof*⟩

AOT-act-theorem *truth−at−alpha:1*: ‹$p \equiv \mathbf{w}_\alpha = \iota x\ (ExtensionOf(x, p))$›
⟨*proof*⟩

AOT-act-theorem *truth−at−alpha:2*: ‹$p \equiv \mathbf{w}_\alpha \models p$›
⟨*proof*⟩

AOT-theorem *alpha−world:1*: ‹$PossibleWorld(\mathbf{w}_\alpha)$›
⟨*proof*⟩

AOT-theorem *alpha−world:2*: ‹$Maximal(\mathbf{w}_\alpha)$›
⟨*proof*⟩

AOT-theorem *t−at−alpha−strict*: ‹$\mathbf{w}_\alpha \models p \equiv \mathcal{A}p$›
⟨*proof*⟩

AOT-act-theorem *not−act*: ‹$w \neq \mathbf{w}_\alpha \rightarrow \neg Actual(w)$›
⟨*proof*⟩

AOT-act-theorem *w−alpha−part*: ‹$Actual(s) \equiv s \trianglelefteq \mathbf{w}_\alpha$›
⟨*proof*⟩

AOT-act-theorem *act−world2:1*: ‹$\mathbf{w}_\alpha \models p \equiv [\lambda y\ p]\mathbf{w}_\alpha$›
⟨*proof*⟩

AOT-act-theorem *act−world2:2*: ‹$p \equiv \mathbf{w}_\alpha \models [\lambda y\ p]\mathbf{w}_\alpha$›
⟨*proof*⟩

AOT-theorem *fund−lem:1*: ‹$\Diamond p \rightarrow \Diamond\exists w\ (w \models p)$›
⟨*proof*⟩

AOT-theorem *fund−lem:2*: ‹$\Diamond\exists\, w\ (w \models p) \rightarrow \exists\, w\ (w \models p)$›
⟨*proof*⟩

AOT-theorem *fund−lem:3*: ‹$p \rightarrow \forall\, s(\forall\, q\ (s \models q \equiv q) \rightarrow s \models p)$›
⟨*proof*⟩

AOT-theorem *fund−lem:4*: ‹$\Box p \rightarrow \Box\forall\, s(\forall\, q\ (s \models q \equiv q) \rightarrow s \models p)$›
⟨*proof*⟩

AOT-theorem *fund−lem:5*: ‹$\Box\forall\, s\ \varphi\{s\} \rightarrow \forall\, s\ \Box\varphi\{s\}$›
⟨*proof*⟩

Note: not explicit in PLM.

AOT-theorem *fund−lem:5[world]*: ‹$\Box\forall\, w\ \varphi\{w\} \rightarrow \forall\, w\ \Box\varphi\{w\}$›
⟨*proof*⟩

AOT-theorem *fund−lem:6*: ‹$\forall\, w\ w \models p \rightarrow \Box\forall\, w\ w \models p$›
⟨*proof*⟩

AOT-theorem *fund−lem:7*: ‹$\Box\forall\, w(w \models p) \rightarrow \Box p$›
⟨*proof*⟩

AOT-theorem *fund:1*: ‹$\Diamond p \equiv \exists\, w\ w \models p$›
⟨*proof*⟩

AOT-theorem *fund:2*: ‹$\Box p \equiv \forall\, w\ (w \models p)$›
⟨*proof*⟩

AOT-theorem *fund:3*: ‹$\neg\Diamond p \equiv \neg\exists\, w\ w \models p$›
⟨*proof*⟩

AOT-theorem *fund:4*: ‹$\neg\Box p \equiv \exists\, w\ \neg w \models p$›
⟨*proof*⟩

AOT-theorem *nec−dia−w:1*: ‹$\Box p \equiv \exists\, w\ w \models \Box p$›
⟨*proof*⟩

AOT-theorem *nec−dia−w:2*: ‹$\Box p \equiv \forall\, w\ w \models \Box p$›
⟨*proof*⟩

AOT-theorem *nec−dia−w:3*: ‹$\Diamond p \equiv \exists\, w\ w \models \Diamond p$›
⟨*proof*⟩

AOT-theorem *nec−dia−w:4*: ‹$\Diamond p \equiv \forall\, w\ w \models \Diamond p$›
⟨*proof*⟩

AOT-theorem *conj−dist−w:1*: ‹$w \models (p\ \&\ q) \equiv ((w \models p)\ \&\ (w \models q))$›
⟨*proof*⟩

AOT-theorem *conj−dist−w:2*: ‹$w \models (p \rightarrow q) \equiv ((w \models p) \rightarrow (w \models q))$›
⟨*proof*⟩

AOT-theorem *conj−dist−w:3*: ‹$w \models (p \lor q) \equiv ((w \models p) \lor (w \models q))$›
⟨*proof*⟩

AOT-theorem *conj−dist−w:4*: ‹$w \models (p \equiv q) \equiv ((w \models p) \equiv (w \models q))$›
⟨*proof*⟩

AOT-theorem *conj−dist−w:5*: ‹$w \models (\forall\, \alpha\ \varphi\{\alpha\}) \equiv (\forall\ \alpha\ (w \models \varphi\{\alpha\}))$›
⟨*proof*⟩

AOT-theorem *conj−dist−w:6*: ‹$w \models (\exists\, \alpha\ \varphi\{\alpha\}) \equiv (\exists\ \alpha\ (w \models \varphi\{\alpha\}))$›

⟨*proof*⟩

AOT-theorem *conj−dist−w*:*7*: ‹$(w \models \Box p) \rightarrow \Box w \models p$›
⟨*proof*⟩

AOT-theorem *conj−dist−w*:*8*: ‹$\exists\, w \exists\, p((\Box w \models p)\ \&\ \neg w \models \Box p)$›
⟨*proof*⟩

AOT-theorem *conj−dist−w*:*9*: ‹$(\Diamond w \models p) \rightarrow w \models \Diamond p$›
⟨*proof*⟩

AOT-theorem *conj−dist−w*:*10*: ‹$\exists\, w \exists\, p((w \models \Diamond p)\ \&\ \neg \Diamond w \models p)$›
⟨*proof*⟩

AOT-theorem *two−worlds−exist*:*1*: ‹$\exists\, p(\mathit{ContingentlyTrue}(p)) \rightarrow \exists\, w\ (\neg \mathit{Actual}(w))$›
⟨*proof*⟩

AOT-theorem *two−worlds−exist*:*2*: ‹$\exists\, p(\mathit{ContingentlyFalse}(p)) \rightarrow \exists\, w\ (\neg \mathit{Actual}(w))$›
⟨*proof*⟩

AOT-theorem *two−worlds−exist*:*3*: ‹$\exists\, w\ \neg \mathit{Actual}(w)$›
⟨*proof*⟩

AOT-theorem *two−worlds−exist*:*4*: ‹$\exists\, w \exists\, w'(w \neq w')$›
⟨*proof*⟩

AOT-theorem *w−rel*:*1*: ‹$[\lambda x\ \varphi\{x\}]\downarrow \rightarrow [\lambda x\ w \models \varphi\{x\}]\downarrow$›
⟨*proof*⟩

AOT-theorem *w−rel*:*2*: ‹$[\lambda x_1 ... x_n\ \varphi\{x_1 ... x_n\}]\downarrow \rightarrow [\lambda x_1 ... x_n\ w \models \varphi\{x_1 ... x_n\}]\downarrow$›
⟨*proof*⟩

AOT-theorem *w−rel*:*3*: ‹$[\lambda x_1 ... x_n\ w \models [F]x_1 ... x_n]\downarrow$›
⟨*proof*⟩

AOT-define *WorldIndexedRelation* :: ‹$\Pi \Rightarrow \tau \Rightarrow \Pi$› (‹-_-›)
w−index: ‹$[F]_w =_{df} [\lambda x_1 ... x_n\ w \models [F]x_1 ... x_n]$›

AOT-define *Rigid* :: ‹$\tau \Rightarrow \varphi$› (‹*Rigid*′(-′)›)
df−rigid−rel:*1*:
‹$\mathit{Rigid}(F) \equiv_{df} F\downarrow\ \&\ \Box \forall\, x_1 ... \forall\, x_n([F]x_1 ... x_n \rightarrow \Box [F]x_1 ... x_n)$›

AOT-define *Rigidifies* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (‹*Rigidifies*′(-,-′)›)
df−rigid−rel:*2*:
‹$\mathit{Rigidifies}(F,\ G) \equiv_{df} \mathit{Rigid}(F)\ \&\ \forall\, x_1 ... \forall\, x_n([F]x_1 ... x_n \equiv [G]x_1 ... x_n)$›

AOT-theorem *rigid−der*:*1*: ‹$[[F]_w]x_1 ... x_n \equiv w \models [F]x_1 ... x_n$›
⟨*proof*⟩

AOT-theorem *rigid−der*:*2*: ‹$\mathit{Rigid}([G]_w)$›
⟨*proof*⟩

AOT-theorem *rigid−der*:*3*: ‹$\exists\, F\ \mathit{Rigidifies}(F,\ G)$›
⟨*proof*⟩

AOT-theorem *rigid−rel−thms*:*1*:
‹$\Box(\forall\, x_1 ... \forall\, x_n\ ([F]x_1 ... x_n \rightarrow \Box [F]x_1 ... x_n)) \equiv \forall\, x_1 ... \forall\, x_n(\Diamond [F]x_1 ... x_n \rightarrow \Box [F]x_1 ... x_n)$›
⟨*proof*⟩

AOT-theorem *rigid−rel−thms*:*2*:

$‹\Box(\forall x_1...\forall x_n\ ([F]x_1...x_n \rightarrow \Box[F]x_1...x_n)) \equiv \forall x_1...\forall x_n(\Box[F]x_1...x_n \vee \Box\neg[F]x_1...x_n)›$
⟨*proof*⟩

AOT-theorem *rigid−rel−thms:3*: $‹Rigid(F) \equiv \forall x_1...\forall x_n\ (\Box[F]x_1...x_n \vee \Box\neg[F]x_1...x_n)›$
⟨*proof*⟩

13 Natural Numbers

AOT-define *CorrelatesOneToOne* :: $‹\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi›$ (‹- |: - $_{1-1}\longleftrightarrow$ -›)
1−1−cor: $‹R\ |:\ F\ {}_{1-1}\!\longleftrightarrow\ G \equiv_{df} R\downarrow\ \&\ F\downarrow\ \&\ G\downarrow\ \&$
$\forall x\ ([F]x \rightarrow \exists!y([G]y\ \&\ [R]xy))\ \&$
$\forall y\ ([G]y \rightarrow \exists!x([F]x\ \&\ [R]xy))›$

AOT-define *MapsTo* :: $‹\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi›$ (‹- |: - $\longrightarrow$ -›)
fFG:1: $‹R\ |:\ F \longrightarrow G \equiv_{df} R\downarrow\ \&\ F\downarrow\ \&\ G\downarrow\ \&\ \forall x\ ([F]x \rightarrow \exists!y([G]y\ \&\ [R]xy))›$

AOT-define *MapsToOneToOne* :: $‹\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi›$ (‹- |: - $_{1-1}\longrightarrow$ -›)
fFG:2: $‹R\ |:\ F\ {}_{1-1}\!\longrightarrow\ G \equiv_{df}$
$R\ |:\ F \longrightarrow G\ \&\ \forall x \forall y \forall z\ (([F]x\ \&\ [F]y\ \&\ [G]z) \rightarrow ([R]xz\ \&\ [R]yz \rightarrow x = y))›$

AOT-define *MapsOnto* :: $‹\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi›$ (‹- |: - $\longrightarrow_{onto}$ -›)
fFG:3: $‹R\ |:\ F \longrightarrow_{onto} G \equiv_{df} R\ |:\ F \longrightarrow G\ \&\ \forall y\ ([G]y \rightarrow \exists x([F]x\ \&\ [R]xy))›$

AOT-define *MapsOneToOneOnto* :: $‹\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi›$ (‹- |: - $_{1-1}\longrightarrow_{onto}$ -›)
fFG:4: $‹R\ |:\ F\ {}_{1-1}\!\longrightarrow_{onto}\ G \equiv_{df} R\ |:\ F\ {}_{1-1}\!\longrightarrow\ G\ \&\ R\ |:\ F \longrightarrow_{onto} G›$

AOT-theorem *eq−1−1*: $‹R\ |:\ F\ {}_{1-1}\!\longleftrightarrow\ G \equiv R\ |:\ F\ {}_{1-1}\!\longrightarrow_{onto}\ G›$
⟨*proof*⟩

We have already introduced the restricted type of Ordinary objects in the Extended Relation Comprehension theory. However, make sure all variable names are defined as expected (avoiding conflicts with situations of possible world theory).

AOT-register-variable-names
Ordinary: $u\ v\ r\ t\ s$

AOT-theorem *equi:1*: $‹\exists!u\ \varphi\{u\} \equiv \exists u\ (\varphi\{u\}\ \&\ \forall v\ (\varphi\{v\} \rightarrow v =_E u))›$
⟨*proof*⟩

AOT-define *CorrelatesEOneToOne* :: $‹\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi›$ (‹- |: - $_{1-1}\longleftrightarrow_E$ -›)
equi:2: $‹R\ |:\ F\ {}_{1-1}\!\longleftrightarrow_E\ G \equiv_{df} R\downarrow\ \&\ F\downarrow\ \&\ G\downarrow\ \&$
$\forall u\ ([F]u \rightarrow \exists!v([G]v\ \&\ [R]uv))\ \&$
$\forall v\ ([G]v \rightarrow \exists!u([F]u\ \&\ [R]uv))›$

AOT-define *EquinumerousE* :: $‹\tau \Rightarrow \tau \Rightarrow \varphi›$ (**infixl** $‹\approx_E›$ *50*)
equi:3: $‹F \approx_E G \equiv_{df} \exists R\ (R\ |:\ F\ {}_{1-1}\!\longleftrightarrow_E\ G)›$

Note: not explicitly in PLM.

AOT-theorem *eq-den-1*: $‹\Pi\downarrow›$ **if** $‹\Pi \approx_E \Pi'›$
⟨*proof*⟩

Note: not explicitly in PLM.

AOT-theorem *eq-den-2*: $‹\Pi'\downarrow›$ **if** $‹\Pi \approx_E \Pi'›$
⟨*proof*⟩

AOT-theorem *eq−part:1*: $‹F \approx_E F›$
⟨*proof*⟩

AOT-theorem *eq−part:2*: $‹F \approx_E G \rightarrow G \approx_E F›$
⟨*proof*⟩

Note: not explicitly in PLM.

AOT-theorem *eq−part:2[terms]*: ‹$\Pi \approx_E \Pi' \rightarrow \Pi' \approx_E \Pi$›
⟨*proof*⟩
declare *eq−part:2[terms][THEN* $\rightarrow$*E, sym]*

AOT-theorem *eq−part:3*: ‹$(F \approx_E G \ \& \ G \approx_E H) \rightarrow F \approx_E H$›
⟨*proof*⟩

Note: not explicitly in PLM.

AOT-theorem *eq−part:3[terms]*: ‹$\Pi \approx_E \Pi''$› **if** ‹$\Pi \approx_E \Pi'$› **and** ‹$\Pi' \approx_E \Pi''$›
⟨*proof*⟩
declare *eq−part:3[terms][trans]*

AOT-theorem *eq−part:4*: ‹$F \approx_E G \equiv \forall H \ (H \approx_E F \equiv H \approx_E G)$›
⟨*proof*⟩

AOT-define *MapsE* :: ‹$\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi$› (‹- |: - ⟶*E* -›)
equi−rem:1:
‹R |: F ⟶*E* $G \equiv_{df} R\downarrow \ \& \ F\downarrow \ \& \ G\downarrow \ \& \ \forall u \ ([F]u \rightarrow \exists! v \ ([G]v \ \& \ [R]uv))$›

AOT-define *MapsEOneToOne* :: ‹$\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi$› (‹- |: - $_{1-1}$⟶*E* -›)
equi−rem:2:
‹R |: F $_{1-1}$⟶*E* $G \equiv_{df}$
R |: F ⟶*E* $G \ \& \ \forall t \forall u \forall v \ (([F]t \ \& \ [F]u \ \& \ [G]v) \rightarrow ([R]tv \ \& \ [R]uv \rightarrow t =_E u))$›

AOT-define *MapsEOnto* :: ‹$\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi$› (‹- |: - ⟶$_{onto}$*E* -›)
equi−rem:3:
‹R |: F ⟶$_{onto}$*E* $G \equiv_{df}$ R |: F ⟶*E* $G \ \& \ \forall v \ ([G]v \rightarrow \exists u \ ([F]u \ \& \ [R]uv))$›

AOT-define *MapsEOneToOneOnto* :: ‹$\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi$› (‹- |: - $_{1-1}$⟶$_{onto}$*E* -›)
equi−rem:4:
‹R |: F $_{1-1}$⟶$_{onto}$*E* $G \equiv_{df}$ R |: F $_{1-1}$⟶*E* G & R |: F ⟶$_{onto}$*E* G›

AOT-theorem *equi−rem−thm*:
‹R |: F $_{1-1}$⟷$_E$ $G \equiv$ R |: F $_{1-1}$⟶$_{onto}$*E* G›
⟨*proof*⟩

AOT-theorem *empty−approx:1*: ‹$(\neg\exists u \ [F]u \ \& \ \neg\exists v \ [H]v) \rightarrow F \approx_E H$›
⟨*proof*⟩

AOT-theorem *empty−approx:2*: ‹$(\exists u \ [F]u \ \& \ \neg\exists v \ [H]v) \rightarrow \neg(F \approx_E H)$›
⟨*proof*⟩

AOT-define *FminusU* :: ‹$\Pi \Rightarrow \tau \Rightarrow \Pi$› (‹-$^{-}$-›)
F−u: ‹$[F]^{-x} =_{df} [\lambda z \ [F]z \ \& \ z \neq_E x]$›

Note: not explicitly in PLM.

AOT-theorem *F−u[den]*: ‹$[F]^{-x}\downarrow$›
⟨*proof*⟩
AOT-theorem *F−u[equiv]*: ‹$[[F]^{-x}]y \equiv ([F]y \ \& \ y \neq_E x)$›
⟨*proof*⟩

AOT-theorem *eqP′*: ‹$F \approx_E G \ \& \ [F]u \ \& \ [G]v \rightarrow [F]^{-u} \approx_E [G]^{-v}$›
⟨*proof*⟩

AOT-theorem *P′−eq*: ‹$[F]^{-u} \approx_E [G]^{-v} \ \& \ [F]u \ \& \ [G]v \rightarrow F \approx_E G$›
⟨*proof*⟩

AOT-theorem *approx−cont:1*: ‹$\exists F \exists G \ \Diamond(F \approx_E G \ \& \ \Diamond\neg F \approx_E G)$›
⟨*proof*⟩

AOT-theorem *approx−cont:2*:
‹$\exists F \exists G\ \Diamond([\lambda z\ \mathcal{A}[F]z] \approx_E G\ \&\ \Diamond\neg[\lambda z\ \mathcal{A}[F]z] \approx_E G)$›
⟨*proof*⟩

notepad
begin

We already have defined being equivalent on the ordinary objects in the Extended Relation Comprehension theory.

⟨*proof*⟩
end

AOT-theorem *apE−eqE:1*: ‹$F \equiv_E G \rightarrow F \approx_E G$›
⟨*proof*⟩

AOT-theorem *apE−eqE:2*: ‹$(F \approx_E G\ \&\ G \equiv_E H) \rightarrow F \approx_E H$›
⟨*proof*⟩

AOT-act-theorem *eq−part−act:1*: ‹$[\lambda z\ \mathcal{A}[F]z] \equiv_E F$›
⟨*proof*⟩

AOT-act-theorem *eq−part−act:2*: ‹$[\lambda z\ \mathcal{A}[F]z] \approx_E F$›
⟨*proof*⟩

AOT-theorem *actuallyF:1*: ‹$\mathcal{A}(F \approx_E [\lambda z\ \mathcal{A}[F]z])$›
⟨*proof*⟩

AOT-theorem *actuallyF:2*: ‹$Rigid([\lambda z\ \mathcal{A}[F]z])$›
⟨*proof*⟩

AOT-theorem *approx−nec:1*: ‹$Rigid(F) \rightarrow F \approx_E [\lambda z\ \mathcal{A}[F]z]$›
⟨*proof*⟩

AOT-theorem *approx−nec:2*:
‹$F \approx_E G \equiv \forall H\ ([\lambda z\ \mathcal{A}[H]z] \approx_E F \equiv [\lambda z\ \mathcal{A}[H]z] \approx_E G)$›
⟨*proof*⟩

AOT-theorem *approx−nec:3*:
‹$(Rigid(F)\ \&\ Rigid(G)) \rightarrow \Box(F \approx_E G \rightarrow \Box F \approx_E G)$›
⟨*proof*⟩

AOT-define *numbers* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (‹*Numbers*′(-,-′)›)
‹$Numbers(x,G) \equiv_{df} A!x\ \&\ G\downarrow\ \&\ \forall F(x[F] \equiv [\lambda z\ \mathcal{A}[F]z] \approx_E G)$›

AOT-theorem *numbers*[*den*]:
‹$\Pi\downarrow \rightarrow (Numbers(\kappa, \Pi) \equiv A!\kappa\ \&\ \forall F(\kappa[F] \equiv [\lambda z\ \mathcal{A}[F]z] \approx_E \Pi))$›
⟨*proof*⟩

AOT-theorem *num−tran:1*:
‹$G \approx_E H \rightarrow (Numbers(x, G) \equiv Numbers(x, H))$›
⟨*proof*⟩

AOT-theorem *num−tran:2*:
‹$(Numbers(x, G)\ \&\ Numbers(x,H)) \rightarrow G \approx_E H$›
⟨*proof*⟩

AOT-theorem *num−tran:3*:
‹$G \equiv_E H \rightarrow (Numbers(x, G) \equiv Numbers(x, H))$›

⟨*proof*⟩

AOT-theorem *pre−Hume*:
‹$(Numbers(x,G) \ \& \ Numbers(y,H)) \rightarrow (x = y \equiv G \approx_E H)$›
⟨*proof*⟩

AOT-theorem *two−num−not*:
‹$\exists u \exists v(u \neq v) \rightarrow \exists x \exists G \exists H(Numbers(x,G) \ \& \ Numbers(x, H) \ \& \ \neg G \equiv_E H)$›
⟨*proof*⟩

AOT-theorem *num:1*: ‹$\exists x \ Numbers(x,G)$›
⟨*proof*⟩

AOT-theorem *num:2*: ‹$\exists !x \ Numbers(x,G)$›
⟨*proof*⟩

AOT-theorem *num−cont:1*:
‹$\exists x \exists G(Numbers(x, G) \ \& \ \neg\Box Numbers(x, G))$›
⟨*proof*⟩

AOT-theorem *num−cont:2*:
‹$Rigid(G) \rightarrow \Box\forall x(Numbers(x,G) \rightarrow \Box Numbers(x,G))$›
⟨*proof*⟩

AOT-theorem *num−cont:3*:
‹$\Box\forall x(Numbers(x, [\lambda z \ \boldsymbol{\mathcal{A}}[G]z]) \rightarrow \Box Numbers(x, [\lambda z \ \boldsymbol{\mathcal{A}}[G]z]))$›
⟨*proof*⟩

AOT-theorem *num−uniq*: ‹$\iota x \ Numbers(x, G)\downarrow$›
⟨*proof*⟩

AOT-define *num* :: ‹$\tau \Rightarrow \kappa_s$› (‹#-› [*100*] *100*)
num−def:1: ‹$\#G =_{df} \iota x \ Numbers(x, G)$›

AOT-theorem *num−def:2*: ‹$\#G\downarrow$›
⟨*proof*⟩

AOT-theorem *num−can:1*:
‹$\#G = \iota x(A!x \ \& \ \forall F \ (x[F] \equiv [\lambda z \ \boldsymbol{\mathcal{A}}[F]z] \approx_E G))$›
⟨*proof*⟩

AOT-theorem *num−can:2*: ‹$\#G = \iota x(A!x \ \& \ \forall F \ (x[F] \equiv F \approx_E G))$›
⟨*proof*⟩

AOT-define *NaturalCardinal* :: ‹$\tau \Rightarrow \varphi$› (‹*NaturalCardinal*′(-′)›)
card: ‹$NaturalCardinal(x) \equiv_{df} \exists G(x = \#G)$›

AOT-theorem *natcard−nec*: ‹$NaturalCardinal(x) \rightarrow \Box NaturalCardinal(x)$›
⟨*proof*⟩

AOT-act-theorem *hume:1*: ‹$Numbers(\#G, G)$›
⟨*proof*⟩

AOT-act-theorem *hume:2*: ‹$\#F = \#G \equiv F \approx_E G$›
⟨*proof*⟩

AOT-act-theorem *hume:3*: ‹$\#F = \#G \equiv \exists R \ (R \ |: F \ {}_{1-1}\!\longrightarrow_{onto} E \ G)$›
⟨*proof*⟩

AOT-act-theorem *hume:4*: ‹$F \equiv_E G \rightarrow \#F = \#G$›
⟨*proof*⟩

AOT-theorem *hume−strict:1*:

‹$\exists x$ ($Numbers(x, F)$ & $Numbers(x, G)$) $\equiv F \approx_E G$›
⟨*proof*⟩

AOT-theorem *hume−strict:2*:
‹$\exists x \exists y$ ($Numbers(x, F)$ &
$\forall z(Numbers(z,F) \rightarrow z = x)$ &
$Numbers(y, G)$ &
$\forall z$ ($Numbers(z, G) \rightarrow z = y$) &
$x = y$) $\equiv$
$F \approx_E G$›
⟨*proof*⟩

AOT-theorem *unotEu*: ‹$\neg\exists y[\lambda x\ O!x\ \&\ x \neq_E x]y$›
⟨*proof*⟩

AOT-define *zero* :: ‹κ_s› (‹0›)
zero:1: ‹$0 =_{df} \#[\lambda x\ O!x\ \&\ x \neq_E x]$›

AOT-theorem *zero:2*: ‹$0\downarrow$›
⟨*proof*⟩

AOT-theorem *zero−card*: ‹$NaturalCardinal(0)$›
⟨*proof*⟩

AOT-theorem *eq−num:1*:
‹$\mathcal{A}Numbers(x, G) \equiv Numbers(x,[\lambda z\ \mathcal{A}[G]z])$›
⟨*proof*⟩

AOT-theorem *eq−num:2*: ‹$Numbers(x,[\lambda z\ \mathcal{A}[G]z]) \equiv x = \#G$›
⟨*proof*⟩

AOT-theorem *eq−num:3*: ‹$Numbers(\#G, [\lambda y\ \mathcal{A}[G]y])$›
⟨*proof*⟩

AOT-theorem *eq−num:4*:
‹$A!\#G\ \&\ \forall F\ (\#G[F] \equiv [\lambda z\ \mathcal{A}[F]z] \approx_E [\lambda z\ \mathcal{A}[G]z])$›
⟨*proof*⟩

AOT-theorem *eq−num:5*: ‹$\#G[G]$›
⟨*proof*⟩

AOT-theorem *eq−num:6*: ‹$Numbers(x, G) \rightarrow NaturalCardinal(x)$›
⟨*proof*⟩

AOT-theorem *eq−df−num*: ‹$\exists G\ (x = \#G) \equiv \exists G\ (Numbers(x,G))$›
⟨*proof*⟩

AOT-theorem *card−en*: ‹$NaturalCardinal(x) \rightarrow \forall F(x[F] \equiv x = \#F)$›
⟨*proof*⟩

AOT-theorem *0F:1*: ‹$\neg\exists u\ [F]u \equiv Numbers(0, F)$›
⟨*proof*⟩

AOT-theorem *0F:2*: ‹$\neg\exists u\ \mathcal{A}[F]u \equiv \#F = 0$›
⟨*proof*⟩

AOT-theorem *0F:3*: ‹$\Box\neg\exists u\ [F]u \rightarrow \#F = 0$›
⟨*proof*⟩

AOT-theorem *0F:4*: ‹$w \models \neg\exists u\ [F]u \equiv \#[F]_w = 0$›
⟨*proof*⟩

AOT-act-theorem *zero=:1*:

‹$NaturalCardinal(x) \rightarrow \forall F\ (x[F] \equiv Numbers(x, F))$›
⟨*proof*⟩

AOT-act-theorem *zero=:2*: ‹$0[F] \equiv \neg\exists u[F]u$›
⟨*proof*⟩

AOT-act-theorem *zero=:3*: ‹$\neg\exists u[F]u \equiv \#F = 0$›
⟨*proof*⟩

AOT-define *Hereditary* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (‹$Hereditary'$(-,-')›)
hered:1:
‹$Hereditary(F, R) \equiv_{df} R\downarrow\ \&\ F\downarrow\ \&\ \forall x \forall y([R]xy \rightarrow ([F]x \rightarrow [F]y))$›

AOT-theorem *hered:2*:
‹$[\lambda xy\ \forall F((\forall z([R]xz \rightarrow [F]z)\ \&\ Hereditary(F,R)) \rightarrow [F]y)]\downarrow$›
⟨*proof*⟩

AOT-define *StrongAncestral* :: ‹$\tau \Rightarrow \Pi$› (‹-*›)
ances−df:
‹$R^* =_{df} [\lambda xy\ \forall F((\forall z([R]xz \rightarrow [F]z)\ \&\ Hereditary(F,R)) \rightarrow [F]y)]$›

AOT-theorem *ances*:
‹$[R^*]xy \equiv \forall F((\forall z([R]xz \rightarrow [F]z)\ \&\ Hereditary(F,R)) \rightarrow [F]y)$›
⟨*proof*⟩

AOT-theorem *anc−her:1*:
‹$[R]xy \rightarrow [R^*]xy$›
⟨*proof*⟩

AOT-theorem *anc−her:2*:
‹$([R^*]xy\ \&\ \forall z([R]xz \rightarrow [F]z)\ \&\ Hereditary(F,R)) \rightarrow [F]y$›
⟨*proof*⟩

AOT-theorem *anc−her:3*:
‹$([F]x\ \&\ [R^*]xy\ \&\ Hereditary(F, R)) \rightarrow [F]y$›
⟨*proof*⟩

AOT-theorem *anc−her:4*: ‹$([R]xy\ \&\ [R^*]yz) \rightarrow [R^*]xz$›
⟨*proof*⟩

AOT-theorem *anc−her:5*: ‹$[R^*]xy \rightarrow \exists z\ [R]zy$›
⟨*proof*⟩

AOT-theorem *anc−her:6*: ‹$([R^*]xy\ \&\ [R^*]yz) \rightarrow [R^*]xz$›
⟨*proof*⟩

AOT-define *OneToOne* :: ‹$\tau \Rightarrow \varphi$› (‹$1\text{−}1'$(-')›)
df−1−1:1: ‹$1\text{−}1(R) \equiv_{df} R\downarrow\ \&\ \forall x \forall y \forall z([R]xz\ \&\ [R]yz \rightarrow x = y)$›

AOT-define *RigidOneToOne* :: ‹$\tau \Rightarrow \varphi$› (‹$Rigid_{1-1}'$(-')›)
df−1−1:2: ‹$Rigid_{1-1}(R) \equiv_{df} 1\text{−}1(R)\ \&\ Rigid(R)$›

AOT-theorem *df−1−1:3*: ‹$Rigid_{1-1}(R) \rightarrow \Box 1\text{−}1(R)$›
⟨*proof*⟩

AOT-theorem *df−1−1:4*: ‹$\forall R(Rigid_{1-1}(R) \rightarrow \Box Rigid_{1-1}(R))$›
⟨*proof*⟩

AOT-define *InDomainOf* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (‹$InDomainOf'$(-,-')›)
df−1−1:5: ‹$InDomainOf(x, R) \equiv_{df} \exists y\ [R]xy$›

AOT-register-rigid-restricted-type
RigidOneToOneRelation: ‹$Rigid_{1-1}(\Pi)$›

⟨*proof*⟩
AOT-register-variable-names
RigidOneToOneRelation: $\mathcal{R}$ $\mathcal{S}$

AOT-define *IdentityRestrictedToDomain* :: ‹$\tau \Rightarrow \Pi$› (‹'(=-')›)
id−d−R: ‹$(=_{\mathcal{R}}) =_{df} [\lambda xy\ \exists z\ ([\mathcal{R}]xz\ \&\ [\mathcal{R}]yz)]$›

syntax *-AOT-id-d-R-infix* :: ‹$\tau \Rightarrow \tau \Rightarrow \tau \Rightarrow \varphi$› (‹(- =-/ -)› *[50, 51, 51] 50*)
translations
-AOT-id-d-R-infix $\kappa\ \Pi\ \kappa'$ ==
CONST AOT-exe (*CONST IdentityRestrictedToDomain* Π) (κ,κ')

AOT-theorem *id−R−thm:1*: ‹$x =_{\mathcal{R}} y \equiv \exists z\ ([\mathcal{R}]xz\ \&\ [\mathcal{R}]yz)$›
⟨*proof*⟩

AOT-theorem *id−R−thm:2*:
‹$x =_{\mathcal{R}} y \rightarrow (InDomainOf(x, \mathcal{R})\ \&\ InDomainOf(y, \mathcal{R}))$›
⟨*proof*⟩

AOT-theorem *id−R−thm:3*: ‹$x =_{\mathcal{R}} y \rightarrow x = y$›
⟨*proof*⟩

AOT-theorem *id−R−thm:4*:
‹$(InDomainOf(x, \mathcal{R}) \vee InDomainOf(y, \mathcal{R})) \rightarrow (x =_{\mathcal{R}} y \equiv x = y)$›
⟨*proof*⟩

AOT-theorem *id−R−thm:5*: ‹$InDomainOf(x, \mathcal{R}) \rightarrow x =_{\mathcal{R}} x$›
⟨*proof*⟩

AOT-theorem *id−R−thm:6*: ‹$x =_{\mathcal{R}} y \rightarrow y =_{\mathcal{R}} x$›
⟨*proof*⟩

AOT-theorem *id−R−thm:7*: ‹$x =_{\mathcal{R}} y\ \&\ y =_{\mathcal{R}} z \rightarrow x =_{\mathcal{R}} z$›
⟨*proof*⟩

AOT-define *WeakAncestral* :: ‹$\Pi \Rightarrow \Pi$› (‹$-^+$›)
w−ances−df: ‹$[\mathcal{R}]^+ =_{df} [\lambda xy\ [\mathcal{R}]^*xy \vee x =_{\mathcal{R}} y]$›

AOT-theorem *w−ances−df[den1]*: ‹$[\lambda xy\ [\Pi]^*xy \vee x =_{\Pi} y]\downarrow$›
⟨*proof*⟩
AOT-theorem *w−ances−df[den2]*: ‹$[\Pi]^+\downarrow$›
⟨*proof*⟩

AOT-theorem *w−ances*: ‹$[\mathcal{R}]^+xy \equiv ([\mathcal{R}]^*xy \vee x =_{\mathcal{R}} y)$›
⟨*proof*⟩

AOT-theorem *w−ances−her:1*: ‹$[\mathcal{R}]xy \rightarrow [\mathcal{R}]^+xy$›
⟨*proof*⟩

AOT-theorem *w−ances−her:2*:
‹$[F]x\ \&\ [\mathcal{R}]^+xy\ \&\ Hereditary(F, \mathcal{R}) \rightarrow [F]y$›
⟨*proof*⟩

AOT-theorem *w−ances−her:3*: ‹$([\mathcal{R}]^+xy\ \&\ [\mathcal{R}]yz) \rightarrow [\mathcal{R}]^*xz$›
⟨*proof*⟩

AOT-theorem *w−ances−her:4*: ‹$([\mathcal{R}]^*xy\ \&\ [\mathcal{R}]yz) \rightarrow [\mathcal{R}]^+xz$›
⟨*proof*⟩

AOT-theorem *w−ances−her:5*: ‹$([\mathcal{R}]xy\ \&\ [\mathcal{R}]^+yz) \rightarrow [\mathcal{R}]^*xz$›
⟨*proof*⟩

AOT-theorem *w−ances−her:6*: ‹$([\mathcal{R}]^+xy\ \&\ [\mathcal{R}]^+yz) \rightarrow [\mathcal{R}]^+xz$›

⟨*proof*⟩

AOT-theorem *w−ances−her:7*: ‹$[\mathcal{R}]^* xy \rightarrow \exists z([\mathcal{R}]^+ xz \ \& \ [\mathcal{R}]zy)$›
⟨*proof*⟩

AOT-theorem *1−1−R:1*: ‹$([\mathcal{R}]xy \ \& \ [\mathcal{R}]^* zy) \rightarrow [\mathcal{R}]^+ zx$›
⟨*proof*⟩

AOT-theorem *1−1−R:2*: ‹$[\mathcal{R}]xy \rightarrow (\neg[\mathcal{R}]^* xx \rightarrow \neg[\mathcal{R}]^* yy)$›
⟨*proof*⟩

AOT-theorem *1−1−R:3*: ‹$\neg[\mathcal{R}]^* xx \rightarrow ([\mathcal{R}]^+ xy \rightarrow \neg[\mathcal{R}]^* yy)$›
⟨*proof*⟩

AOT-theorem *1−1−R:4*: ‹$[\mathcal{R}]^* xy \rightarrow \mathit{InDomainOf}(x,\mathcal{R})$›
⟨*proof*⟩

AOT-theorem *1−1−R:5*: ‹$[\mathcal{R}]^+ xy \rightarrow \mathit{InDomainOf}(x,\mathcal{R})$›
⟨*proof*⟩

AOT-theorem *pre−ind*:
‹$([F]z \ \& \ \forall x \forall y(([\mathcal{R}]^+ zx \ \& \ [\mathcal{R}]^+ zy) \rightarrow ([\mathcal{R}]xy \rightarrow ([F]x \rightarrow [F]y)))) \rightarrow$
$\forall x \ ([\mathcal{R}]^+ zx \rightarrow [F]x)$›
⟨*proof*⟩

The following is not part of PLM, but a theorem of AOT. It states that the predecessor relation coexists with numbering a property. We will use this fact to derive the predecessor axiom, which asserts that the predecessor relation denotes, from the fact that our models validate that numbering a property denotes.

AOT-theorem *pred-coex*:
‹$[\lambda xy \ \exists F \exists u \ ([F]u \ \& \ \mathit{Numbers}(y,F) \ \& \ \mathit{Numbers}(x,[F]^{-u}))]\downarrow \equiv \forall F \ ([\lambda x \ \mathit{Numbers}(x,F)]\downarrow)$›
⟨*proof*⟩

The following is not part of PLM, but a consequence of extended relation comprehension and can be used to *derive* the predecessor axiom.

AOT-theorem *numbers-prop-den*: ‹$[\lambda x \ \mathit{Numbers}(x,G)]\downarrow$›
⟨*proof*⟩

The two theorems above allow us to derive the predecessor axiom of PLM as theorem.

AOT-theorem *pred*: ‹$[\lambda xy \ \exists F \exists u \ ([F]u \ \& \ \mathit{Numbers}(y,F) \ \& \ \mathit{Numbers}(x,[F]^{-u}))]\downarrow$›
⟨*proof*⟩

AOT-define *Predecessor* :: ‹Π› (‹$\mathbb{P}$›)
pred−thm:1:
‹$\mathbb{P} =_{df} [\lambda xy \ \exists F \exists u \ ([F]u \ \& \ \mathit{Numbers}(y,F) \ \& \ \mathit{Numbers}(x,[F]^{-u}))]$›

AOT-theorem *pred−thm:2*: ‹$\mathbb{P}\downarrow$›
⟨*proof*⟩

AOT-theorem *pred−thm:3*:
‹$[\mathbb{P}]xy \equiv \exists F \exists u \ ([F]u \ \& \ \mathit{Numbers}(y,F) \ \& \ \mathit{Numbers}(x,[F]^{-u}))$›
⟨*proof*⟩

AOT-theorem *pred−1−1:1*: ‹$[\mathbb{P}]xy \rightarrow \Box[\mathbb{P}]xy$›
⟨*proof*⟩

AOT-theorem *pred−1−1:2*: ‹$\mathit{Rigid}(\mathbb{P})$›
⟨*proof*⟩

AOT-theorem *pred−1−1:3*: ‹$\mathit{1\text{−}1}(\mathbb{P})$›
⟨*proof*⟩

AOT-theorem *pred−1−1:4*: ‹$\mathit{Rigid}_{1-1}(\mathbb{P})$›

⟨*proof*⟩

AOT-theorem *assume−anc:1*:
‹$[\mathbb{P}]^* = [\lambda xy\ \forall F((\forall z([\mathbb{P}]xz \rightarrow [F]z)\ \&\ \mathit{Hereditary}(F,\mathbb{P})) \rightarrow [F]y)]$›
⟨*proof*⟩

AOT-theorem *assume−anc:2*: ‹$\mathbb{P}^*\downarrow$›
⟨*proof*⟩

AOT-theorem *assume−anc:3*:
‹$[\mathbb{P}^*]xy \equiv \forall F((\forall z([\mathbb{P}]xz \rightarrow [F]z)\ \&\ \forall x'\forall y'([\mathbb{P}]x'y' \rightarrow ([F]x' \rightarrow [F]y'))) \rightarrow [F]y)$›
⟨*proof*⟩

AOT-theorem *no−pred−0:1*: ‹$\neg\exists x\ [\mathbb{P}]x\ 0$›
⟨*proof*⟩

AOT-theorem *no−pred−0:2*: ‹$\neg\exists x\ [\mathbb{P}^*]x\ 0$›
⟨*proof*⟩

AOT-theorem *no−pred−0:3*: ‹$\neg[\mathbb{P}^*]0\ 0$›
⟨*proof*⟩

AOT-theorem *assume1:1*: ‹$(=_{\mathbb{P}}) = [\lambda xy\ \exists z\ ([\mathbb{P}]xz\ \&\ [\mathbb{P}]yz)]$›
⟨*proof*⟩

AOT-theorem *assume1:2*: ‹$x =_{\mathbb{P}} y \equiv \exists z\ ([\mathbb{P}]xz\ \&\ [\mathbb{P}]yz)$›
⟨*proof*⟩

AOT-theorem *assume1:3*: ‹$[\mathbb{P}]^+ = [\lambda xy\ [\mathbb{P}]^*xy \vee x =_{\mathbb{P}} y]$›
⟨*proof*⟩

AOT-theorem *assume1:4*: ‹$[\mathbb{P}]^+\downarrow$›
⟨*proof*⟩

AOT-theorem *assume1:5*: ‹$[\mathbb{P}]^+xy \equiv [\mathbb{P}]^*xy \vee x =_{\mathbb{P}} y$›
⟨*proof*⟩

AOT-define *NaturalNumber* :: ‹τ› (‹$\mathbb{N}$›)
nnumber:1: ‹$\mathbb{N} =_{df} [\lambda x\ [\mathbb{P}]^+ 0x]$›

AOT-theorem *nnumber:2*: ‹$\mathbb{N}\downarrow$›
⟨*proof*⟩

AOT-theorem *nnumber:3*: ‹$[\mathbb{N}]x \equiv [\mathbb{P}]^+ 0x$›
⟨*proof*⟩

AOT-theorem *0−n*: ‹$[\mathbb{N}]0$›
⟨*proof*⟩

AOT-theorem *mod−col−num:1*: ‹$[\mathbb{N}]x \rightarrow \Box[\mathbb{N}]x$›
⟨*proof*⟩

AOT-theorem *mod−col−num:2*: ‹*Rigid*($\mathbb{N}$)›
⟨*proof*⟩

AOT-register-rigid-restricted-type
Number: ‹$[\mathbb{N}]\kappa$›
⟨*proof*⟩
AOT-register-variable-names
Number: *m n k i j*

AOT-theorem *0−pred*: ‹$\neg\exists n\ [\mathbb{P}]n\ 0$›
⟨*proof*⟩

AOT-theorem *no−same−succ*:
‹$\forall n \forall m \forall k([\mathbb{P}]nk \; \& \; [\mathbb{P}]mk \rightarrow n = m)$›
⟨*proof*⟩

AOT-theorem *induction*:
‹$\forall F([F]0 \; \& \; \forall n \forall m([\mathbb{P}]nm \rightarrow ([F]n \rightarrow [F]m)) \rightarrow \forall n[F]n)$›
⟨*proof*⟩

AOT-theorem *suc−num:1*: ‹$[\mathbb{P}]nx \rightarrow [\mathbb{N}]x$›
⟨*proof*⟩

AOT-theorem *suc−num:2*: ‹$[[\mathbb{P}]^*]nx \rightarrow [\mathbb{N}]x$›
⟨*proof*⟩

AOT-theorem *suc−num:3*: ‹$[\mathbb{P}]^+nx \rightarrow [\mathbb{N}]x$›
⟨*proof*⟩

AOT-theorem *pred−num*: ‹$[\mathbb{P}]xn \rightarrow [\mathbb{N}]x$›
⟨*proof*⟩

AOT-theorem *nat−card*: ‹$[\mathbb{N}]x \rightarrow NaturalCardinal(x)$›
⟨*proof*⟩

AOT-theorem *pred−func:1*: ‹$[\mathbb{P}]xy \; \& \; [\mathbb{P}]xz \rightarrow y = z$›
⟨*proof*⟩

AOT-theorem *pred−func:2*: ‹$[\mathbb{P}]nm \; \& \; [\mathbb{P}]nk \rightarrow m = k$›
⟨*proof*⟩

AOT-theorem *being-number-of-den*: ‹$[\lambda x \; x = \#G]\downarrow$›
⟨*proof*⟩

axiomatization *ω-nat* :: ‹$\omega \Rightarrow nat$› **where** *ω-nat*: ‹*surj ω-nat*›

Unfortunately, since the axiom requires the type ω to have an infinite domain, **nitpick** can only find a potential model and no genuine model. However, since we could trivially choose ω as a copy of *nat*, we can still be assured that above axiom is consistent.

lemma ‹*True*› **nitpick**[*satisfy*, *user-axioms*, *card nat=1*, *expect = potential*] ⟨*proof*⟩

AOT-axiom *modal−axiom*:
‹$\exists x([\mathbb{N}]x \; \& \; x = \#G) \rightarrow \Diamond\exists y([E!]y \; \& \; \forall u \; (\boldsymbol{\mathcal{A}}[G]u \rightarrow u \neq_E y))$›
⟨*proof*⟩

AOT-theorem *modal−lemma*:
‹$\Diamond\forall u(\boldsymbol{\mathcal{A}}[G]u \rightarrow u \neq_E v) \rightarrow \forall u(\boldsymbol{\mathcal{A}}[G]u \rightarrow u \neq_E v)$›
⟨*proof*⟩

AOT-theorem *th−succ*: ‹$\forall n \exists! m \; [\mathbb{P}]nm$›
⟨*proof*⟩

AOT-define *Successor* :: ‹$\tau \Rightarrow \kappa_s$› (‹-′′› [*100*] *100*)
def−suc: ‹$n' =_{df} \iota m([\mathbb{P}]nm)$›

Note: not explicitly in PLM

AOT-theorem *def−suc*[*den1*]: ‹$\iota m([\mathbb{P}]nm)\downarrow$›
⟨*proof*⟩

Note: not explicitly in PLM

AOT-theorem *def−suc*[*den2*]: **shows** ‹$n'\downarrow$›
⟨*proof*⟩

AOT-theorem *suc-eq-desc*: ‹$n' = \iota m([\mathbb{P}]nm)$›
⟨*proof*⟩

AOT-theorem *suc−fact*: ‹$n = m \rightarrow n' = m'$›
⟨*proof*⟩

AOT-theorem *ind−gnd*: ‹$m = 0 \vee \exists n(m = n')$›
⟨*proof*⟩

AOT-theorem *suc−thm*: ‹$[\mathbb{P}]n\ n'$›
⟨*proof*⟩

AOT-define *Numeral1* :: ‹κ_s› (‹*1*›)
numerals:*1*: ‹$1 =_{df} 0'$›

AOT-theorem *prec−facts*:*1*: ‹$[\mathbb{P}]0\ 1$›
⟨*proof*⟩

AOT-define *Finite* :: ‹$\tau \Rightarrow \varphi$› (‹*Finite*′(-′)›)
inf−card:*1*: ‹$Finite(x) \equiv_{df} NaturalCardinal(x)\ \&\ [\mathbb{N}]x$›
AOT-define *Infinite* :: ‹$\tau \Rightarrow \varphi$› (‹*Infinite*′(-′)›)
inf−card:*2*: ‹$Infinite(x) \equiv_{df} NaturalCardinal(x)\ \&\ \neg Finite(x)$›

AOT-theorem *inf−card−exist*:*1*: ‹$NaturalCardinal(\#O!)$›
⟨*proof*⟩

AOT-theorem *inf−card−exist*:*2*: ‹$Infinite(\#O!)$›
⟨*proof*⟩
theory *AOT-misc*
imports *AOT-NaturalNumbers*
begin

14 Miscellaneous Theorems

AOT-theorem *PossiblyNumbersEmptyPropertyImpliesZero*:
‹$\Diamond Numbers(x,[\lambda z\ O!z\ \&\ z \neq_E z]) \rightarrow x = 0$›
⟨*proof*⟩

AOT-define *Numbers′* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (‹*Numbers*′′′(-,-′)›)
‹$Numbers'(x,\ G) \equiv_{df} A!x\ \&\ G\downarrow\ \&\ \forall F\ (x[F] \equiv F \approx_E G)$›
AOT-theorem *Numbers′equiv*: ‹$Numbers'(x,G) \equiv A!x\ \&\ \forall F\ (x[F] \equiv F \approx_E G)$›
⟨*proof*⟩

AOT-theorem *Numbers′DistinctZeroes*:
‹$\exists x \exists y\ (\Diamond Numbers'(x,[\lambda z\ O!z\ \&\ z \neq_E z])\ \&\ \Diamond Numbers'(y,[\lambda z\ O!z\ \&\ z \neq_E z])\ \&\ x \neq y)$›
⟨*proof*⟩

AOT-theorem *restricted-identity*:
‹$x =_{\mathcal{R}} y \equiv (InDomainOf(x,\mathcal{R})\ \&\ InDomainOf(y,\mathcal{R})\ \&\ x = y)$›
⟨*proof*⟩

AOT-theorem *induction′*: ‹$\forall F\ ([F]0\ \&\ \forall n([F]n \rightarrow [F]n') \rightarrow \forall n\ [F]n)$›
⟨*proof*⟩

AOT-define *ExtensionOf* :: ‹$\tau \Rightarrow \Pi \Rightarrow \varphi$› (‹*ExtensionOf*′(-,-′)›)
exten−property:*1*: ‹$ExtensionOf(x,[G]) \equiv_{df} A!x\ \&\ G\downarrow\ \&\ \forall F(x[F] \equiv \forall z([F]z \equiv [G]z))$›

AOT-define *OrdinaryExtensionOf* :: ‹$\tau \Rightarrow \Pi \Rightarrow \varphi$› (‹*OrdinaryExtensionOf*′(-,-′)›)

‹$OrdinaryExtensionOf(x,[G]) \equiv_{df}\ A!x\ \&\ G\downarrow\ \&\ \forall F(x[F] \equiv \forall z(O!z \rightarrow ([F]z \equiv [G]z)))$›

AOT-theorem *BeingOrdinaryExtensionOfDenotes*:
‹$[\lambda x\ OrdinaryExtensionOf(x,[G])]\downarrow$›
⟨*proof*⟩

Fragments of PLM's theory of Concepts.

AOT-define *FimpG* :: ‹$\Pi \Rightarrow \Pi \Rightarrow \varphi$› (**infixl** ‹$\Rightarrow$› *50*)
F−imp−G: ‹$[G] \Rightarrow [F] \equiv_{df}\ F\downarrow\ \&\ G\downarrow\ \&\ \Box\forall x\ ([G]x \rightarrow [F]x)$›

AOT-define *concept* :: ‹Π› (‹$C!$›)
concepts: ‹$C! =_{df}\ A!$›

AOT-register-rigid-restricted-type
Concept: ‹$C!\kappa$›
⟨*proof*⟩

AOT-register-variable-names
Concept: $c\ d\ e$

AOT-theorem *concept−comp:1*: ‹$\exists x(C!x\ \&\ \forall F(x[F] \equiv \varphi\{F\}))$›
⟨*proof*⟩

AOT-theorem *concept−comp:2*: ‹$\exists !x(C!x\ \&\ \forall F(x[F] \equiv \varphi\{F\}))$›
⟨*proof*⟩

AOT-theorem *concept−comp:3*: ‹$\iota x(C!x\ \&\ \forall F(x[F] \equiv \varphi\{F\}))\downarrow$›
⟨*proof*⟩

AOT-theorem *concept−comp:4*:
‹$\iota x(C!x\ \&\ \forall F(x[F] \equiv \varphi\{F\})) = \iota x(A!x\ \&\ \forall F(x[F] \equiv \varphi\{F\}))$›
⟨*proof*⟩

AOT-define *conceptInclusion* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (**infixl** ‹$\preceq$› *100*)
con:1: ‹$c \preceq d \equiv_{df} \forall F(c[F] \rightarrow d[F])$›

AOT-define *conceptOf* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (‹$ConceptOf'$(-,-')›)
concept−of−G: ‹$ConceptOf(c,G) \equiv_{df}\ G\downarrow\ \&\ \forall F\ (c[F] \equiv [G] \Rightarrow [F])$›

AOT-theorem *ConceptOfOrdinaryProperty*: ‹$([H] \Rightarrow O!) \rightarrow [\lambda x\ ConceptOf(x,H)]\downarrow$›
⟨*proof*⟩

AOT-theorem *con−exists:1*: ‹$\exists c\ ConceptOf(c,G)$›
⟨*proof*⟩

AOT-theorem *con−exists:2*: ‹$\exists !c\ ConceptOf(c,G)$›
⟨*proof*⟩

AOT-theorem *con−exists:3*: ‹$\iota c\ ConceptOf(c,G)\downarrow$›
⟨*proof*⟩

AOT-define *theConceptOfG* :: ‹$\tau \Rightarrow \kappa_s$› (‹**c**-›)
concept−G: ‹$\mathbf{c}_G =_{df} \iota c\ ConceptOf(c,\ G)$›

AOT-theorem *concept−G[den]*: ‹$\mathbf{c}_G\downarrow$›
⟨*proof*⟩

AOT-theorem *concept−G[concept]*: ‹$C!\mathbf{c}_G$›
⟨*proof*⟩

AOT-theorem *conG−strict*: ‹$\mathbf{c}_G = \iota c\ \forall F(c[F] \equiv [G] \Rightarrow [F])$›
⟨*proof*⟩

AOT-theorem *conG−lemma:1*: ‹$\forall F(\mathbf{c}_G[F] \equiv [G] \Rightarrow [F])$›
⟨*proof*⟩

AOT-theorem *conH-enc-ord*:
‹$([H] \Rightarrow O!) \rightarrow \Box\forall F\ \forall G\ (\Box G \equiv_E F \rightarrow (\mathbf{c}_H[F] \equiv \mathbf{c}_H[G]))$›
⟨*proof*⟩

AOT-theorem *concept-inclusion-denotes-1*:
‹$([H] \Rightarrow O!) \rightarrow [\lambda x\ \mathbf{c}_H \preceq x]\downarrow$›
⟨*proof*⟩

AOT-theorem *concept-inclusion-denotes-2*:
‹$([H] \Rightarrow O!) \rightarrow [\lambda x\ x \preceq \mathbf{c}_H]\downarrow$›
⟨*proof*⟩

AOT-define *ThickForm* :: ‹$\tau \Rightarrow \tau \Rightarrow \varphi$› (‹$FormOf'$(-,-')›)
tform−of: ‹$FormOf(x,G) \equiv_{df} A!x\ \&\ G\downarrow\ \&\ \forall F(x[F] \equiv [G] \Rightarrow [F])$›

AOT-theorem *FormOfOrdinaryProperty*: ‹$([H] \Rightarrow O!) \rightarrow [\lambda x\ FormOf(x,H)]\downarrow$›
⟨*proof*⟩

AOT-theorem *equal-E-rigid-one-to-one*: ‹$Rigid_{1-1}((=_E))$›
⟨*proof*⟩

AOT-theorem *equal-E-domain*: ‹$InDomainOf(x,(=_E)) \equiv O!x$›
⟨*proof*⟩

AOT-theorem *shared-urelement-projection-identity*:
assumes ‹$\forall y\ [\lambda x\ (y[\lambda z\ [R]zx])]\downarrow$›
shows ‹$\forall F([F]a \equiv [F]b) \rightarrow [\lambda z\ [R]za] = [\lambda z\ [R]zb]$›
⟨*proof*⟩

AOT-theorem *shared-urelement-exemplification-identity*:
assumes ‹$\forall y\ [\lambda x\ (y[\lambda z\ [G]x])]\downarrow$›
shows ‹$\forall F([F]a \equiv [F]b) \rightarrow ([G]a) = ([G]b)$›
⟨*proof*⟩

The assumptions of the theorems above are derivable, if the additional introduction rules for the upcoming extension of *AOT-instance-of-cqt-2* $\varphi \Longrightarrow [\lambda\nu_1...\nu_n\ \varphi\{\nu_1...\nu_n\}]\downarrow \in \Lambda_\Box$ are explicitly allowed (while they are currently not part of the abstraction layer).

notepad
begin
⟨*proof*⟩
end

end