Public Announcement Logic

Asta Halkjær From

December 14, 2021

Abstract

This work is a formalization of public announcement logic with countably many agents. It includes proofs of soundness and completeness for a variant of the axiom system PA + DIST! + NEC! [1]. The completeness proof builds on the Epistemic Logic theory.

Contents

theory *PAL* **imports** *Epistemic-Logic.Epistemic-Logic* **begin**

1 Syntax

datatype *'i pfm*
  = *FF* ($\bot_!$)
  | *Pro' id* (*Pro*$_!$)
  | *Dis* ‹*'i pfm*› ‹*'i pfm*› (**infixr** $\vee_!$ *30*)
  | *Con* ‹*'i pfm*› ‹*'i pfm*› (**infixr** $\wedge_!$ *35*)
  | *Imp* ‹*'i pfm*› ‹*'i pfm*› (**infixr** $\longrightarrow_!$ *25*)
  | *K' 'i* ‹*'i pfm*› ($K_!$)
  | *Ann* ‹*'i pfm*› ‹*'i pfm*› ([-]$_!$ - [*50*, *50*] *50*)

abbreviation *PIff* :: ‹*'i pfm* ⇒ *'i pfm* ⇒ *'i pfm*› (**infixr** $\longleftrightarrow_!$ *25*) **where**
  ‹$p \longleftrightarrow_! q \equiv (p \longrightarrow_! q) \wedge_! (q \longrightarrow_! p)$›

2 Semantics

fun
  psemantics :: ‹(*'i*, *'w*) *kripke* ⇒ *'w* ⇒ *'i pfm* ⇒ *bool*› (-, - $\models_!$ - [*50*, *50*] *50*) **and**
  restrict :: ‹(*'i*, *'w*) *kripke* ⇒ *'i pfm* ⇒ (*'i*, *'w*) *kripke*› **where**
  ‹$(M, w \models_! \bot_!) = False$›
| ‹$(M, w \models_! Pro_!\ x) = \pi\ M\ w\ x$›
| ‹$(M, w \models_! (p \vee_! q)) = ((M, w \models_! p) \vee (M, w \models_! q))$›
| ‹$(M, w \models_! (p \wedge_! q)) = ((M, w \models_! p) \wedge (M, w \models_! q))$›
| ‹$(M, w \models_! (p \longrightarrow_! q)) = ((M, w \models_! p) \longrightarrow (M, w \models_! q))$›
| ‹$(M, w \models_! K_!\ i\ p) = (\forall v \in \mathcal{W}\ M \cap \mathcal{K}\ M\ i\ w.\ M, v \models_! p)$›
| ‹$(M, w \models_! [r]_!\ p) = ((M, w \models_! r) \longrightarrow (restrict\ M\ r, w \models_! p))$›
| ‹$restrict\ M\ p = Kripke\ \{w\ |w.\ w \in \mathcal{W}\ M \wedge M, w \models_! p\}\ (\pi\ M)\ (\mathcal{K}\ M)$›

primrec *static* :: ‹*'i pfm* ⇒ *bool*› **where**
  ‹$static\ \bot_! = True$›
| ‹$static\ (Pro_!\ \text{-}) = True$›
| ‹$static\ (p \vee_! q) = (static\ p \wedge static\ q)$›
| ‹$static\ (p \wedge_! q) = (static\ p \wedge static\ q)$›
| ‹$static\ (p \longrightarrow_! q) = (static\ p \wedge static\ q)$›
| ‹$static\ (K_!\ i\ p) = static\ p$›
| ‹$static\ ([r]_!\ p) = False$›

primrec *lower* :: ‹*'i pfm* ⇒ *'i fm*› **where**
  ‹$lower\ \bot_! = \bot$›
| ‹$lower\ (Pro_!\ x) = Pro\ x$›
| ‹$lower\ (p \vee_! q) = (lower\ p \vee lower\ q)$›
| ‹$lower\ (p \wedge_! q) = (lower\ p \wedge lower\ q)$›
| ‹$lower\ (p \longrightarrow_! q) = (lower\ p \longrightarrow lower\ q)$›
| ‹$lower\ (K_!\ i\ p) = K\ i\ (lower\ p)$›

| ‹$lower\ ([r]_!\ p) = undefined$›

primrec *lift* :: ‹$'i\ fm \Rightarrow 'i\ pfm$› **where**
  ‹$lift\ \bot = \bot_!$›
| ‹$lift\ (Pro\ x) = Pro_!\ x$›
| ‹$lift\ (p \lor q) = (lift\ p \lor_!\ lift\ q)$›
| ‹$lift\ (p \land q) = (lift\ p \land_!\ lift\ q)$›
| ‹$lift\ (p \longrightarrow q) = (lift\ p \longrightarrow_!\ lift\ q)$›
| ‹$lift\ (K\ i\ p) = K_!\ i\ (lift\ p)$›

lemma *lower-semantics*:
  assumes ‹$static\ p$›
  shows ‹$(M,\ w \models lower\ p) \longleftrightarrow (M,\ w \models_!\ p)$›
  ⟨*proof*⟩

lemma *lift-semantics*: ‹$(M,\ w \models p) \longleftrightarrow (M,\ w \models_!\ lift\ p)$›
  ⟨*proof*⟩

lemma *lower-lift*: ‹$lower\ (lift\ p) = p$›
  ⟨*proof*⟩

lemma *lift-lower*: ‹$static\ p \Longrightarrow lift\ (lower\ p) = p$›
  ⟨*proof*⟩

3 Soundness of Reduction

primrec *reduce'* :: ‹$'i\ pfm \Rightarrow 'i\ pfm \Rightarrow 'i\ pfm$› **where**
  ‹$reduce'\ r\ \bot_! = (r \longrightarrow_!\ \bot_!)$›
| ‹$reduce'\ r\ (Pro_!\ x) = (r \longrightarrow_!\ Pro_!\ x)$›
| ‹$reduce'\ r\ (p \lor_!\ q) = (reduce'\ r\ p \lor_!\ reduce'\ r\ q)$›
| ‹$reduce'\ r\ (p \land_!\ q) = (reduce'\ r\ p \land_!\ reduce'\ r\ q)$›
| ‹$reduce'\ r\ (p \longrightarrow_!\ q) = (reduce'\ r\ p \longrightarrow_!\ reduce'\ r\ q)$›
| ‹$reduce'\ r\ (K_!\ i\ p) = (r \longrightarrow_!\ K_!\ i\ (reduce'\ r\ p))$›
| ‹$reduce'\ r\ ([p]_!\ q) = undefined$›

primrec *reduce* :: ‹$'i\ pfm \Rightarrow 'i\ pfm$› **where**
  ‹$reduce\ \bot_! = \bot_!$›
| ‹$reduce\ (Pro_!\ x) = Pro_!\ x$›
| ‹$reduce\ (p \lor_!\ q) = (reduce\ p \lor_!\ reduce\ q)$›
| ‹$reduce\ (p \land_!\ q) = (reduce\ p \land_!\ reduce\ q)$›
| ‹$reduce\ (p \longrightarrow_!\ q) = (reduce\ p \longrightarrow_!\ reduce\ q)$›
| ‹$reduce\ (K_!\ i\ p) = K_!\ i\ (reduce\ p)$›
| ‹$reduce\ ([r]_!\ p) = reduce'\ (reduce\ r)\ (reduce\ p)$›

lemma *static-reduce'*: ‹$static\ p \Longrightarrow static\ r \Longrightarrow static\ (reduce'\ r\ p)$›
  ⟨*proof*⟩

lemma *static-reduce*: ‹$static\ (reduce\ p)$›
  ⟨*proof*⟩

lemma *reduce′-semantics*:
  assumes ‹*static q*›
  shows ‹$((M,\ w \models_! [p]_!\ (q))) = (M,\ w \models_!$ *reduce′ p q*$)$›
  ⟨*proof*⟩

lemma *reduce-semantics*: ‹$(M,\ w \models_!\ p) = (M,\ w \models_!$ *reduce p*$)$›
⟨*proof*⟩

4 Proof System

primrec *peval* :: ‹$(id \Rightarrow bool) \Rightarrow ('i\ pfm \Rightarrow bool) \Rightarrow 'i\ pfm \Rightarrow bool$› **where**
  ‹$peval$ - - $\bot_! = False$›
| ‹$peval\ g$ - $(Pro_!\ x) = g\ x$›
| ‹$peval\ g\ h\ (p \vee_!\ q) = (peval\ g\ h\ p \vee peval\ g\ h\ q)$›
| ‹$peval\ g\ h\ (p \wedge_!\ q) = (peval\ g\ h\ p \wedge peval\ g\ h\ q)$›
| ‹$peval\ g\ h\ (p \longrightarrow_!\ q) = (peval\ g\ h\ p \longrightarrow peval\ g\ h\ q)$›
| ‹$peval$ - $h\ (K_!\ i\ p) = h\ (K_!\ i\ p)$›
| ‹$peval$ - $h\ ([r]_!\ p) = h\ ([r]_!\ p)$›

abbreviation ‹$ptautology\ p \equiv \forall\, g\ h.\ peval\ g\ h\ p$›

inductive *PAK* :: ‹$('i\ pfm \Rightarrow bool) \Rightarrow 'i\ pfm \Rightarrow bool$› (- $\vdash_!$ - [*50*, *50*] *50*)
  for A :: ‹$'i\ pfm \Rightarrow bool$› **where**
    PA1: ‹$ptautology\ p \Longrightarrow A \vdash_!\ p$›
  | *PA2*: ‹$A \vdash_!\ (K_!\ i\ p \wedge_!\ K_!\ i\ (p \longrightarrow_!\ q) \longrightarrow_!\ K_!\ i\ q)$›
  | *PAx*: ‹$A\ p \Longrightarrow A \vdash_!\ p$›
  | *PR1*: ‹$A \vdash_!\ p \Longrightarrow A \vdash_!\ (p \longrightarrow_!\ q) \Longrightarrow A \vdash_!\ q$›
  | *PR2*: ‹$A \vdash_!\ p \Longrightarrow A \vdash_!\ K_!\ i\ p$›
  | *PFF*: ‹$A \vdash_!\ ([r]_!\ \bot_! \longleftrightarrow_!\ (r \longrightarrow_!\ \bot_!))$›
  | *PPro*: ‹$A \vdash_!\ ([r]_!\ Pro_!\ x \longleftrightarrow_!\ (r \longrightarrow_!\ Pro_!\ x))$›
  | *PDis*: ‹$A \vdash_!\ ([r]_!\ (p \vee_!\ q) \longleftrightarrow_!\ [r]_!\ p \vee_!\ [r]_!\ q)$›
  | *PCon*: ‹$A \vdash_!\ ([r]_!\ (p \wedge_!\ q) \longleftrightarrow_!\ [r]_!\ p \wedge_!\ [r]_!\ q)$›
  | *PImp*: ‹$A \vdash_!\ (([r]_!\ (p \longrightarrow_!\ q)) \longleftrightarrow_!\ ([r]_!\ p \longrightarrow_!\ [r]_!\ q))$›
  | *PK*: ‹$A \vdash_!\ (([r]_!\ K_!\ i\ p) \longleftrightarrow_!\ (r \longrightarrow_!\ K_!\ i\ ([r]_!\ p)))$›
  | *PAnn*: ‹$A \vdash_!\ p \Longrightarrow A \vdash_!\ [r]_!\ p$›

lemma *eval-peval*: ‹$eval\ h\ (g\ o\ lift)\ p = peval\ h\ g\ (lift\ p)$›
  ⟨*proof*⟩

lemma *tautology-ptautology*: ‹$tautology\ p \Longrightarrow ptautology\ (lift\ p)$›
  ⟨*proof*⟩

lemma *peval-eval*:
  assumes ‹*static p*›
  shows ‹$eval\ h\ g\ (lower\ p) = peval\ h\ (g\ o\ lower)\ p$›
  ⟨*proof*⟩

lemma *ptautology-tautology*:

assumes ‹*static* p›
shows ‹*ptautology* $p \Longrightarrow$ *tautology* (*lower* p)›
⟨*proof*⟩

theorem *AK-PAK*: ‹A *o* *lift* $\vdash p \Longrightarrow A \vdash_{!}$ *lift* p›
⟨*proof*⟩

theorem *static-completeness*:
assumes ‹*static* p› ‹$\forall(M$ :: ($'i$ :: *countable*, $'i$ *fm set*) *kripke*) w. M, $w \models_{!} p$›
shows ‹$A \vdash_{!} p$›
⟨*proof*⟩

5 Soundness

lemma *peval-semantics*: ‹*peval* (*val* w) (λq. *Kripke* W *val* r, $w \models_{!} q$) p = (*Kripke* W *val* r, $w \models_{!} p$)›
⟨*proof*⟩

lemma *ptautology*:
assumes ‹*ptautology* p›
shows ‹M, $w \models_{!} p$›
⟨*proof*⟩

theorem *soundness*:
fixes M :: ‹($'i$, $'w$) *kripke*›
assumes
‹$\bigwedge(M$ :: ($'i$, $'w$) *kripke*) w p. A $p \Longrightarrow P$ $M \Longrightarrow M$, $w \models_{!} p$›
‹$\bigwedge(M$ :: ($'i$, $'w$) *kripke*) p. P $M \Longrightarrow P$ (*restrict* M p)›
shows ‹$A \vdash_{!} p \Longrightarrow P$ $M \Longrightarrow M$, $w \models_{!} p$›
⟨*proof*⟩

corollary ‹(λ-. *False*) $\vdash_{!} p \Longrightarrow M$, $w \models_{!} p$›
⟨*proof*⟩

6 Completeness

lemma *ConE*:
assumes ‹$A \vdash_{!} (p \wedge_{!} q)$›
shows ‹$A \vdash_{!} p$› ‹$A \vdash_{!} q$›
⟨*proof*⟩

lemma *Iff-Dis*:
assumes ‹$A \vdash_{!} (p \longleftrightarrow_{!} p')$› ‹$A \vdash_{!} (q \longleftrightarrow_{!} q')$›
shows ‹$A \vdash_{!} ((p \vee_{!} q) \longleftrightarrow_{!} (p' \vee_{!} q'))$›
⟨*proof*⟩

lemma *Iff-Con*:
assumes ‹$A \vdash_{!} (p \longleftrightarrow_{!} p')$› ‹$A \vdash_{!} (q \longleftrightarrow_{!} q')$›

shows ‹$A \vdash_! ((p \wedge_! q) \longleftrightarrow_! (p' \wedge_! q'))$›
⟨*proof*⟩

lemma *Iff-Imp*:
assumes ‹$A \vdash_! (p \longleftrightarrow_! p')$› ‹$A \vdash_! (q \longleftrightarrow_! q')$›
shows ‹$A \vdash_! ((p \longrightarrow_! q) \longleftrightarrow_! (p' \longrightarrow_! q'))$›
⟨*proof*⟩

lemma *Iff-sym*: ‹$(A \vdash_! (p \longleftrightarrow_! q)) = (A \vdash_! (q \longleftrightarrow_! p))$›
⟨*proof*⟩

lemma *Iff-Iff*:
assumes ‹$A \vdash_! (p \longleftrightarrow_! p')$› ‹$A \vdash_! (p \longleftrightarrow_! q)$›
shows ‹$A \vdash_! (p' \longleftrightarrow_! q)$›
⟨*proof*⟩

lemma *K'-A2'*: ‹$A \vdash_! (K_!\ i\ (p \longrightarrow_! q) \longrightarrow_! K_!\ i\ p \longrightarrow_! K_!\ i\ q)$›
⟨*proof*⟩

lemma *K'-map*:
assumes ‹$A \vdash_! (p \longrightarrow_! q)$›
shows ‹$A \vdash_! (K_!\ i\ p \longrightarrow_! K_!\ i\ q)$›
⟨*proof*⟩

lemma *ConI*:
assumes ‹$A \vdash_! p$› ‹$A \vdash_! q$›
shows ‹$A \vdash_! (p \wedge_! q)$›
⟨*proof*⟩

lemma *Iff-wk*:
assumes ‹$A \vdash_! (p \longleftrightarrow_! q)$›
shows ‹$A \vdash_! ((r \longrightarrow_! p) \longleftrightarrow_! (r \longrightarrow_! q))$›
⟨*proof*⟩

lemma *Iff-reduce'*:
assumes ‹*static* p›
shows ‹$A \vdash_! ([r]_!\ p \longleftrightarrow_! reduce'\ r\ p)$›
⟨*proof*⟩

lemma *Iff-Ann1*:
assumes r: ‹$A \vdash_! (r \longleftrightarrow_! r')$› **and** ‹*static* p›
shows ‹$A \vdash_! ([r]_!\ p \longleftrightarrow_! [r']_!\ p)$›
⟨*proof*⟩

lemma *Iff-Ann2*:
assumes ‹$A \vdash_! (p \longleftrightarrow_! p')$›
shows ‹$A \vdash_! ([r]_!\ p \longleftrightarrow_! [r]_!\ p')$›
⟨*proof*⟩

lemma *Iff-reduce*: ‹$A \vdash_! (p \longleftrightarrow_! \textit{reduce}\ p)$›
⟨*proof*⟩

theorem *completeness*:
  assumes ‹$\forall (M :: ('i :: \textit{countable}, 'i\ \textit{fm set})\ \textit{kripke})\ w.\ M, w \models_! p$›
  shows ‹$A \vdash_! p$›
⟨*proof*⟩

end

References

[1] Y. Wang and Q. Cao. On axiomatizations of public announcement logic. *Synthese*, 190(Supplement-1):103–134, 2013.