Epistemic Logic

Asta Halkjr From

September 30, 2020

Abstract

This work is a formalization of epistemic logic with countably many agents. It includes proofs of soundness and completeness for the axiom system K. The completeness proof is based on the textbook "Reasoning About Knowledge" by Fagin, Halpern, Moses and Vardi (MIT Press 1995) [1].

Contents

theory *Epistemic-Logic* **imports** *HOL−Library.Countable* **begin**

1 Syntax

type-synonym *id* = *string*

datatype *'i fm*
= *FF* (⊥)
| *Pro id*
| *Dis* ‹*'i fm*› ‹*'i fm*› (**infixr** ∨ *30*)
| *Con* ‹*'i fm*› ‹*'i fm*› (**infixr** ∧ *35*)
| *Imp* ‹*'i fm*› ‹*'i fm*› (**infixr** ⟶ *25*)
| *K 'i* ‹*'i fm*›

abbreviation *TT* (⊤) **where**
‹*TT* ≡ ⊥ ⟶ ⊥›

abbreviation *Neg* (¬ - [*40*] *40*) **where**
‹*Neg p* ≡ *p* ⟶ ⊥›

2 Semantics

datatype (*'i*, *'s*) *kripke* = *Kripke* (π: ‹*'s* ⇒ *id* ⇒ *bool*›) (𝒦: ‹*'i* ⇒ *'s* ⇒ *'s set*›)

primrec *semantics* :: ‹(*'i*, *'s*) *kripke* ⇒ *'s* ⇒ *'i fm* ⇒ *bool*›
(-, - ⊨ - [*50*,*50*] *50*) **where**
‹(-, - ⊨ ⊥) = *False*›
| ‹(*M*, *s* ⊨ *Pro i*) = π *M s i*›
| ‹(*M*, *s* ⊨ (*p* ∨ *q*)) = ((*M*, *s* ⊨ *p*) ∨ (*M*, *s* ⊨ *q*))›
| ‹(*M*, *s* ⊨ (*p* ∧ *q*)) = ((*M*, *s* ⊨ *p*) ∧ (*M*, *s* ⊨ *q*))›
| ‹(*M*, *s* ⊨ (*p* ⟶ *q*)) = ((*M*, *s* ⊨ *p*) ⟶ (*M*, *s* ⊨ *q*))›
| ‹(*M*, *s* ⊨ *K i p*) = (∀ *t* ∈ 𝒦 *M i s*. *M*, *t* ⊨ *p*)›

3 Utility

abbreviation *reflexive* :: ‹(*'i*, *'s*) *kripke* ⇒ *bool*› **where**
‹*reflexive M* ≡ ∀ *i s*. *s* ∈ 𝒦 *M i s*›

abbreviation *symmetric* :: ‹(*'i*, *'s*) *kripke* ⇒ *bool*› **where**
‹*symmetric M* ≡ ∀ *i s t*. *t* ∈ 𝒦 *M i s* ⟷ *s* ∈ 𝒦 *M i t*›

abbreviation *transitive* :: ‹(*'i*, *'s*) *kripke* ⇒ *bool*› **where**
‹*transitive M* ≡ ∀ *i s t u*. *t* ∈ 𝒦 *M i s* ∧ *u* ∈ 𝒦 *M i t* ⟶ *u* ∈ 𝒦 *M i s*›

lemma *Imp-intro*: ‹(*M*, *s* ⊨ *p* ⟹ *M*, *s* ⊨ *q*) ⟹ *M*, *s* ⊨ *Imp p q*›
⟨*proof*⟩

4 S5 Axioms

theorem *distribution*: ‹M, $s \models (K\ i\ p \wedge K\ i\ (p \longrightarrow q) \longrightarrow K\ i\ q)$›
⟨*proof*⟩

theorem *generalization*:
  assumes *valid*: ‹$\forall (M :: ('i, 's)\ kripke)\ s.\ M, s \models p$›
  shows ‹$(M :: ('i, 's)\ kripke), s \models K\ i\ p$›
⟨*proof*⟩

theorem *truth*:
  assumes ‹*reflexive* M›
  shows ‹$M, s \models (K\ i\ p \longrightarrow p)$›
⟨*proof*⟩

theorem *pos-introspection*:
  assumes ‹*transitive* M›
  shows ‹$M, s \models (K\ i\ p \longrightarrow K\ i\ (K\ i\ p))$›
⟨*proof*⟩

theorem *neg-introspection*:
  assumes ‹*symmetric* M› ‹*transitive* M›
  shows ‹$M, s \models (\neg\ K\ i\ p \longrightarrow K\ i\ (\neg\ K\ i\ p))$›
⟨*proof*⟩

5 Axiom System K

primrec *eval* :: ‹$(id \Rightarrow bool) \Rightarrow ('i\ fm \Rightarrow bool) \Rightarrow 'i\ fm \Rightarrow bool$› **where**
  ‹$eval$ - - $\bot = False$›
| ‹$eval\ g$ - $(Pro\ i) = g\ i$›
| ‹$eval\ g\ h\ (p \vee q) = (eval\ g\ h\ p \vee eval\ g\ h\ q)$›
| ‹$eval\ g\ h\ (p \wedge q) = (eval\ g\ h\ p \wedge eval\ g\ h\ q)$›
| ‹$eval\ g\ h\ (p \longrightarrow q) = (eval\ g\ h\ p \longrightarrow eval\ g\ h\ q)$›
| ‹$eval$ - $h\ (K\ i\ p) = h\ (K\ i\ p)$›

abbreviation ‹$tautology\ p \equiv \forall g\ h.\ eval\ g\ h\ p$›

inductive *SystemK* :: ‹$'i\ fm \Rightarrow bool$› ($\vdash$ - [50] 50) **where**
  A1: ‹$tautology\ p \Longrightarrow \vdash p$›
| *A2*: ‹$\vdash (K\ i\ p \wedge K\ i\ (p \longrightarrow q) \longrightarrow K\ i\ q)$›
| *R1*: ‹$\vdash p \Longrightarrow \vdash (p \longrightarrow q) \Longrightarrow \vdash q$›
| *R2*: ‹$\vdash p \Longrightarrow \vdash K\ i\ p$›

6 Soundness

lemma *eval-semantics*: ‹$eval\ (pi\ s)\ (\lambda q.\ Kripke\ pi\ r, s \models q)\ p = (Kripke\ pi\ r, s \models p)$›
  ⟨*proof*⟩

theorem *tautology*: ‹*tautology* $p \Longrightarrow M,\ s \models p$›
⟨*proof*⟩

theorem *soundness*: ‹$\vdash p \Longrightarrow M,\ s \models p$›
⟨*proof*⟩

7 Derived rules

lemma *K-FFI*: ‹$\vdash (p \longrightarrow (\neg\ p) \longrightarrow \bot)$›
⟨*proof*⟩

primrec *conjoin* :: ‹$'i$ *fm list* $\Rightarrow$ $'i$ *fm* $\Rightarrow$ $'i$ *fm*› **where**
‹*conjoin* [] $q = q$›
| ‹*conjoin* $(p\ \#\ ps)\ q = (p \wedge$ *conjoin* $ps\ q)$›

primrec *imply* :: ‹$'i$ *fm list* $\Rightarrow$ $'i$ *fm* $\Rightarrow$ $'i$ *fm*› **where**
‹*imply* [] $q = q$›
| ‹*imply* $(p\ \#\ ps)\ q = (p \longrightarrow$ *imply* $ps\ q)$›

lemma *K-imply-head*: ‹$\vdash$ *imply* $(p\ \#\ ps)\ p$›
⟨*proof*⟩

lemma *K-imply-Cons*:
assumes ‹$\vdash$ *imply* $ps\ q$›
shows ‹$\vdash$ *imply* $(p\ \#\ ps)\ q$›
⟨*proof*⟩

lemma *K-imply-member*: ‹$p \in$ *set* $ps \Longrightarrow \vdash$ *imply* $ps\ p$›
⟨*proof*⟩

lemma *K-right-mp*:
assumes ‹$\vdash$ *imply* $ps\ p$› ‹$\vdash$ *imply* $ps\ (p \longrightarrow q)$›
shows ‹$\vdash$ *imply* $ps\ q$›
⟨*proof*⟩

lemma *tautology-imply-superset*:
assumes ‹*set* $ps \subseteq$ *set* qs›
shows ‹*tautology* (*imply* $ps\ r \longrightarrow$ *imply* $qs\ r$)›
⟨*proof*⟩

lemma *tautology-imply*: ‹*tautology* $q \Longrightarrow$ *tautology* (*imply* $ps\ q$)›
⟨*proof*⟩

theorem *K-imply-weaken*:
assumes ‹$\vdash$ *imply* $ps\ q$› ‹*set* $ps \subseteq$ *set* ps'›
shows ‹$\vdash$ *imply* $ps'\ q$›
⟨*proof*⟩

lemma *imply-append*: ‹$imply\ (ps\ @\ ps')\ q = imply\ ps\ (imply\ ps'\ q)$›
⟨*proof*⟩

lemma *K-ImpI*:
assumes ‹$\vdash imply\ (p\ \#\ G)\ q$›
shows ‹$\vdash imply\ G\ (p \longrightarrow q)$›
⟨*proof*⟩

lemma *cut*: ‹$\vdash imply\ G\ p \Longrightarrow \vdash imply\ (p\ \#\ G)\ q \Longrightarrow \vdash imply\ G\ q$›
⟨*proof*⟩

lemma *K-Boole*: ‹$\vdash imply\ ((\neg\ p)\ \#\ G)\ \bot \Longrightarrow \vdash imply\ G\ p$›
⟨*proof*⟩

lemma *K-DisE*:
assumes ‹$\vdash imply\ (A\ \#\ G)\ C$› ‹$\vdash imply\ (B\ \#\ G)\ C$› ‹$\vdash imply\ G\ (A \lor B)$›
shows ‹$\vdash imply\ G\ C$›
⟨*proof*⟩

lemma *K-conjoin-imply*:
assumes ‹$\vdash (\neg\ conjoin\ G\ (\neg\ p))$›
shows ‹$\vdash imply\ G\ p$›
⟨*proof*⟩

lemma *K-distrib-K-imp*:
assumes ‹$\vdash K\ i\ (imply\ G\ q)$›
shows ‹$\vdash imply\ (map\ (K\ i)\ G)\ (K\ i\ q)$›
⟨*proof*⟩

8 Consistency

definition *consistency* :: ‹$'i\ fm\ set\ set \Rightarrow bool$› **where**
‹$consistency\ C \equiv \forall S \in C.$
$(\forall p.\ \neg\ (Pro\ p \in S \land (\neg\ Pro\ p) \in S)) \land$
$\bot \notin S \land$
$(\forall Z.\ (\neg\ (\neg\ Z)) \in S \longrightarrow S \cup \{Z\} \in C) \land$
$(\forall A\ B.\ (A \land B) \in S \longrightarrow S \cup \{A, B\} \in C) \land$
$(\forall A\ B.\ (\neg\ (A \lor B)) \in S \longrightarrow S \cup \{\neg\ A, \neg\ B\} \in C) \land$
$(\forall A\ B.\ (A \lor B) \in S \longrightarrow S \cup \{A\} \in C \lor S \cup \{B\} \in C) \land$
$(\forall A\ B.\ (\neg\ (A \land B)) \in S \longrightarrow S \cup \{\neg\ A\} \in C \lor S \cup \{\neg\ B\} \in C) \land$
$(\forall A\ B.\ (A \longrightarrow B) \in S \longrightarrow S \cup \{\neg\ A\} \in C \lor S \cup \{B\} \in C) \land$
$(\forall A\ B.\ (\neg\ (A \longrightarrow B)) \in S \longrightarrow S \cup \{A, \neg\ B\} \in C) \land$
$(\forall A.\ tautology\ A \longrightarrow S \cup \{A\} \in C) \land$
$(\forall A\ i.\ \neg\ (K\ i\ A \in S \land (\neg\ K\ i\ A) \in S))$›

8.1 Closure under subsets

definition *close* :: ‹$'i\ fm\ set\ set \Rightarrow 'i\ fm\ set\ set$› **where**
‹$close\ C \equiv \{S.\ \exists S' \in C.\ S \subseteq S'\}$›

definition *subset-closed* :: ‹$'a$ *set set* ⇒ *bool*› **where**
‹*subset-closed* $C \equiv (\forall S' \in C.\ \forall S.\ S \subseteq S' \longrightarrow S \in C)$›

lemma *subset-in-close*:
assumes ‹$S' \subseteq S$› ‹$S \cup x \in C$›
shows ‹$S' \cup x \in$ *close* C›
⟨*proof*⟩

theorem *close-consistency*:
fixes C :: ‹$'i$ *fm set set*›
assumes ‹*consistency* C›
shows ‹*consistency* (*close* C)›
⟨*proof*⟩

theorem *close-closed*: ‹*subset-closed* (*close* C)›
⟨*proof*⟩

theorem *close-subset*: ‹$C \subseteq$ *close* C›
⟨*proof*⟩

8.2 Finite character

definition *finite-char* :: ‹$'a$ *set set* ⇒ *bool*› **where**
‹*finite-char* $C \equiv (\forall S.\ S \in C = (\forall S'.\ \mathit{finite}\ S' \longrightarrow S' \subseteq S \longrightarrow S' \in C))$›

definition *mk-finite-char* :: ‹$'a$ *set set* ⇒ $'a$ *set set*› **where**
‹*mk-finite-char* $C \equiv \{S.\ \forall S'.\ \mathit{finite}\ S' \longrightarrow S' \subseteq S \longrightarrow S' \in C\}$›

theorem *finite-char*: ‹*finite-char* (*mk-finite-char* C)›
⟨*proof*⟩

theorem *finite-char-closed*: ‹*finite-char* $C \Longrightarrow$ *subset-closed* C›
⟨*proof*⟩

theorem *finite-char-subset*: ‹*subset-closed* $C \Longrightarrow C \subseteq$ *mk-finite-char* C›
⟨*proof*⟩

theorem *finite-consistency*:
fixes C :: ‹$'i$ *fm set set*›
assumes ‹*consistency* C› ‹*subset-closed* C›
shows ‹*consistency* (*mk-finite-char* C)›
⟨*proof*⟩

8.3 Maximal extension

instantiation *fm* :: (*countable*) *countable* **begin**
instance ⟨*proof*⟩
end

definition *is-chain* :: ‹(*nat* ⇒ *'a set*) ⇒ *bool*› **where**
‹*is-chain f* ≡ ∀ *n*. *f n* ⊆ *f* (*Suc n*)›

lemma *is-chainD*: ‹*is-chain f* ⟹ *x* ∈ *f m* ⟹ *x* ∈ *f* (*m* + *n*)›
⟨*proof*⟩

lemma *is-chainD'*:
assumes ‹*is-chain f*› ‹*x* ∈ *f m*› ‹*m* ≤ *k*›
shows ‹*x* ∈ *f k*›
⟨*proof*⟩

lemma *chain-index*:
assumes ‹*is-chain f*› ‹*finite F*›
shows ‹*F* ⊆ (⋃ *n*. *f n*) ⟹ ∃ *n*. *F* ⊆ *f n*›
⟨*proof*⟩

lemma *chain-union-closed'*:
assumes ‹*is-chain f*› ‹∀ *n*. *f n* ∈ *C*› ‹∀ *S'* ∈ *C*. ∀ *S* ⊆ *S'*. *S* ∈ *C*› ‹*finite S'*› ‹*S'* ⊆ (⋃ *n*. *f n*)›
shows ‹*S'* ∈ *C*›
⟨*proof*⟩

lemma *chain-union-closed*:
assumes ‹*finite-char C*› ‹*is-chain f*› ‹∀ *n*. *f n* ∈ *C*›
shows ‹(⋃ *n*. *f n*) ∈ *C*›
⟨*proof*⟩

primrec *extend* :: ‹*'i fm set* ⇒ *'i fm set set* ⇒ (*nat* ⇒ *'i fm*) ⇒ *nat* ⇒ *'i fm set*›
where
‹*extend S C f 0* = *S*› |
‹*extend S C f* (*Suc n*) =
(*if extend S C f n* ∪ {*f n*} ∈ *C*
then extend S C f n ∪ {*f n*}
else extend S C f n)›

definition *Extend* :: ‹*'i fm set* ⇒ *'i fm set set* ⇒ (*nat* ⇒ *'i fm*) ⇒ *'i fm set*›
where
‹*Extend S C f* ≡ ⋃ *n*. *extend S C f n*›

lemma *is-chain-extend*: ‹*is-chain* (*extend S C f*)›
⟨*proof*⟩

lemma *extend-in-C*: ‹*consistency C* ⟹ *S* ∈ *C* ⟹ *extend S C f n* ∈ *C*›
⟨*proof*⟩

theorem *Extend-in-C*: ‹*consistency C* ⟹ *finite-char C* ⟹ *S* ∈ *C* ⟹ *Extend S C f* ∈ *C*›
⟨*proof*⟩

theorem *Extend-subset*: $\langle S \subseteq \mathit{Extend}\ S\ C\ f \rangle$
  $\langle \mathit{proof} \rangle$

definition *maximal* :: $\langle 'a\ set \Rightarrow 'a\ set\ set \Rightarrow bool \rangle$ **where**
  $\langle \mathit{maximal}\ S\ C \equiv \forall S' \in C.\ S \subseteq S' \longrightarrow S = S' \rangle$

theorem *Extend-maximal*:
  assumes $\langle \forall y :: 'i\ fm.\ \exists n.\ y = f\ n \rangle$ $\langle \mathit{finite\text{-}char}\ C \rangle$
  shows $\langle \mathit{maximal}\ (\mathit{Extend}\ S\ C\ f)\ C \rangle$
  $\langle \mathit{proof} \rangle$

8.4 K consistency

theorem *K-consistency*: $\langle \mathit{consistency}\ \{\mathit{set}\ G \mid G.\ \neg \vdash \mathit{imply}\ G\ \bot\} \rangle$
  $\langle \mathit{proof} \rangle$

theorem *K-finite-consistency*: $\langle \mathit{consistency}\ (\mathit{mk\text{-}finite\text{-}char}\ (\mathit{close}\ \{\mathit{set}\ G \mid G.\ \neg \vdash \mathit{imply}\ G\ \bot\})) \rangle$
  $\langle \mathit{proof} \rangle$

theorem *K-concrete-finite-consistency*:
  defines $\langle C \equiv \mathit{mk\text{-}finite\text{-}char}\ (\mathit{close}\ \{\mathit{set}\ G \mid G.\ \neg \vdash \mathit{imply}\ G\ \bot\}) \rangle$
  assumes $\langle \mathit{set}\ G \in C \rangle$
  shows $\langle \neg \vdash \mathit{imply}\ G\ \bot \rangle$
  $\langle \mathit{proof} \rangle$

9 Model existence

lemma *at-least-one-in-maximal*:
  assumes $\langle \mathit{consistency}\ C \rangle$ $\langle V \in C \rangle$ $\langle \mathit{maximal}\ V\ C \rangle$
  shows $\langle p \in V \lor (\neg\ p) \in V \rangle$
$\langle \mathit{proof} \rangle$

lemma *at-most-one-in-maximal*:
  assumes $\langle \mathit{consistency}\ C \rangle$ $\langle V \in C \rangle$ $\langle \mathit{maximal}\ V\ C \rangle$
  shows $\langle \neg\ (p \in V \land (\neg\ p) \in V) \rangle$
$\langle \mathit{proof} \rangle$

theorem *exactly-one-in-maximal*:
  assumes $\langle \mathit{consistency}\ C \rangle$ $\langle V \in C \rangle$ $\langle \mathit{maximal}\ V\ C \rangle$
  shows $\langle (p \in V) \neq ((\neg\ p) \in V) \rangle$
  $\langle \mathit{proof} \rangle$

theorem *conjuncts-in-maximal*:
  assumes $\langle \mathit{consistency}\ C \rangle$ $\langle V \in C \rangle$ $\langle \mathit{maximal}\ V\ C \rangle$
  shows $\langle (p \land q) \in V \longleftrightarrow p \in V \land q \in V \rangle$
$\langle \mathit{proof} \rangle$

theorem *consequent-in-maximal*:

assumes ‹$consistency\ C$› ‹$V \in C$› ‹$maximal\ V\ C$› ‹$p \in V$› ‹$(p \longrightarrow q) \in V$›
shows ‹$q \in V$›
⟨*proof*⟩

lemma *K-not-neg-in-consistency*: ‹$\neg \vdash (\neg\ p) \Longrightarrow \{p\} \in \{set\ G \mid G.\ \neg \vdash imply\ G\ \bot\}$›
⟨*proof*⟩

lemma *K-inconsistent-neg*:
defines ‹$C \equiv mk\text{-}finite\text{-}char\ (close\ \{set\ G \mid G.\ \neg \vdash imply\ G\ \bot\})$›
assumes ‹$\{p\} \notin C$›
shows ‹$\vdash (\neg\ p)$›
⟨*proof*⟩

lemma *conjuncts-in-consistency*:
assumes ‹$consistency\ C$› ‹$subset\text{-}closed\ C$› ‹$S \cup \{p \wedge q\} \in C$›
shows ‹$S \cup \{p, q\} \in C$›
⟨*proof*⟩

lemma *conjoined-in-consistency*:
assumes ‹$consistency\ C$› ‹$subset\text{-}closed\ C$› ‹$S \cup \{conjoin\ ps\ q\} \in C$›
shows ‹$S \cup set\ ps \cup \{q\} \in C$›
⟨*proof*⟩

lemma *inconsistent-conjoin*:
defines ‹$C \equiv mk\text{-}finite\text{-}char\ (close\ \{set\ G \mid G.\ \neg \vdash imply\ G\ \bot\})$›
assumes ‹$set\ G \cup \{p\} \notin C$›
shows ‹$\vdash (\neg\ conjoin\ G\ p)$›
⟨*proof*⟩

theorem *K-in-maximal*:
defines ‹$C \equiv mk\text{-}finite\text{-}char\ (close\ \{set\ G \mid G.\ \neg \vdash imply\ G\ \bot\})$›
assumes ‹$\vdash p$› ‹$V \in C$› ‹$maximal\ V\ C$›
shows ‹$p \in V$›
⟨*proof*⟩

lemma *exists-finite-inconsistent*:
fixes C :: ‹$'i\ fm\ set\ set$›
assumes ‹$finite\text{-}char\ C$› ‹$V \cup \{\neg\ p\} \notin C$›
shows ‹$\exists W.\ W \cup \{\neg\ p\} \subseteq V \cup \{\neg\ p\} \wedge (\neg\ p) \notin W \wedge finite\ W \wedge W \cup \{\neg\ p\} \notin C$›
⟨*proof*⟩

theorem *exists-maximal-superset*:
fixes C :: ‹$('i :: countable)\ fm\ set\ set$›
assumes ‹$consistency\ C$› ‹$finite\text{-}char\ C$› ‹$V \in C$›
obtains W **where** ‹$V \subseteq W$› ‹$W \in C$› ‹$maximal\ W\ C$›
⟨*proof*⟩

type-synonym $'i$ *s-max* = ‹$'i$ *fm set*›

abbreviation *pi* :: ‹$'i$ *s-max* ⇒ *id* ⇒ *bool*› **where**
‹*pi s i* ≡ *Pro i* ∈ *s*›

abbreviation *partition* :: ‹$'i$ *fm set* ⇒ $'i$ ⇒ $'i$ *fm set*› **where**
‹*partition V i* ≡ {*p*. *K i p* ∈ *V*}›

abbreviation *reach* :: ‹$'i$ *fm set set* ⇒ $'i$ ⇒ $'i$ *s-max* ⇒ $'i$ *s-max set*› **where**
‹*reach C i V* ≡ {*W*. *partition V i* ⊆ *W* ∧ *W* ∈ *C* ∧ *maximal W C*}›

theorem *model-existence*:
fixes *p* :: ‹($'i$ :: *countable*) *fm*›
defines ‹*C* ≡ *mk-finite-char* (*close* {*set G* | *G*. ¬ ⊢ *imply G* ⊥})›
assumes ‹*V* ∈ *C*› ‹*maximal V C*›
shows ‹(*p* ∈ *V* ⟷ *Kripke pi* (*reach C*), *V* ⊨ *p*) ∧
((¬ *p*) ∈ *V* ⟷ *Kripke pi* (*reach C*), *V* ⊨ ¬ *p*)›
⟨*proof*⟩

9.1 Completeness

lemma *imply-completeness*:
assumes *valid*: ‹∀ (*M* :: ($'i$, ($'i$ :: *countable*) *fm set*) *kripke*) *s*.
list-all (λ*q*. *M*, *s* ⊨ *q*) *G* ⟶ *M*, *s* ⊨ *p*›
shows ‹⊢ *imply G p*›
⟨*proof*⟩

theorem *completeness*:
assumes ‹∀ (*M* :: ($'i$ :: *countable*, $'i$ *fm set*) *kripke*) *s*. *M*, *s* ⊨ *p*›
shows ‹⊢ *p*›
⟨*proof*⟩

10 Main Result

— System K is sound and complete

abbreviation ‹*valid p* ≡ ∀ (*M* :: (*nat*, *nat s-max*) *kripke*) *s*. *M*, *s* ⊨ *p*›

theorem *main*: ‹*valid p* ⟷ ⊢ *p*›
⟨*proof*⟩

corollary ‹*valid p* ⟶ *M*, *s* ⊨ *p*›
⟨*proof*⟩

11 Acknowledgements

The definition of a consistency property, subset closure, finite character and maximally consistent sets is based on work by Berghofer, but has been

adapted from first-order logic to epistemic logic.

- Stefan Berghofer: First-Order Logic According to Fitting. https://www.isa-afp.org/entries/FOL-Fitting.shtml

end

References

[1] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.