Generic Epistemic and Public Announcement Logic Completeness Results

Asta Halkjær From,
DTU Compute

Logic & AI @ AlgoLoG seminar,
DTU Compute, September 2022

Agenda

Tour de force of a published paper and an extension currently under submission:

- A. H. From. “*Formalized Soundness and Completeness of Epistemic Logic*”. In: Logic, Language, Information, and Computation - 27th International Workshop, **WoLLIC** 2021, Virtual Event, October 5-8, 2021, Proceedings. Ed. by A. Silva, R. Wassermann, and R. J. G. B. de Queiroz. Vol. 13038. Lecture Notes in Computer Science. Springer, 2021, pp. 1–15. doi: 10.1007/978-3-030-88853-4_1.
- A. H. From. “*Formalized Soundness and Completeness of Epistemic and Public Announcement Logic*”. In: **Journal of Logic and Computation — Special Issue** from the 27th Workshop on Logic, Language, Information and Computation (WoLLIC 2021) (2022). **Under submission.**

https://www.isa-afp.org/entries/Epistemic_Logic.html

https://www.isa-afp.org/entries/Public_Announcement_Logic.html

Modal Logic

Normal Modal Logics

Proof system for normal modal logics *parameterized* by extra axioms A.

```
inductive AK :: ‹('i fm ⇒ bool) ⇒ 'i fm ⇒ bool› (‹_ ⊢ _› [50, 50] 50)
  for A :: ‹'i fm ⇒ bool› where
    A1: ‹tautology p ⟹ A ⊢ p›
  | A2: ‹A ⊢ K i p ∧ K i (p ⟶ q) ⟶ K i q›
  | Ax: ‹A p ⟹ A ⊢ p›
  | R1: ‹A ⊢ p ⟹ A ⊢ p ⟶ q ⟹ A ⊢ q›
  | R2: ‹A ⊢ p ⟹ A ⊢ K i p›
```

System K: `A` always false

System T: `A` true for `(K i p ⟶ p)`, false otherwise,

System …

Generic Soundness

If all axioms admitted by A are valid on P-models,
then any formula derived under A is valid on P-models:

```
theorem soundness:
  assumes ‹⋀M w p. A p ⟹ P M ⟹ w ∈ 𝒲 M ⟹ M, w ⊨ p›
  shows ‹A ⊢ p ⟹ P M ⟹ w ∈ 𝒲 M ⟹ M, w ⊨ p›
```

Same thing under assumptions G:

```
theorem strong_soundness:
  assumes ‹⋀M w p. A p ⟹ P M ⟹ w ∈ 𝒲 M ⟹ M, w ⊨ p›
  shows ‹A; G ⊢ p ⟹ P; G ⊫ p›
```

Generic Strong Completeness

If the set of formulas G imply p on P-models, and
the canonical model for axioms A is a P-model,
then under axioms A we can derive p from G.

```
theorem strong_completeness:
  assumes ‹P; G ⊫ p› and ‹P (canonical A)›
  shows ‹A; G ⊢ p›
```

The canonical model for axioms A is the usual construction with maximal consistent sets (wrt. A) as worlds.

(A-consistency, A-maximality, Lindenbaum extension wrt. A, etc.)

Example: System K

We already have the results for K (no axioms, all models):

```
abbreviation SystemK (‹_ ⊢K _› [50] 50) where
  ‹G ⊢K p ≡ (λ_. False); G ⊢ p›

abbreviation validK (‹_ ⊫K _› [50, 50] 50) where
  ‹G ⊫K p ≡ (λ_. True); G ⊫ p›
```

Soundness and completeness:

```
theorem main_K: ‹G ⊫K p ⟷ G ⊢K p›
```

Example: System T

Fix a different A:

```
inductive AxT :: ‹'i fm ⇒ bool› where
  ‹AxT (K i p ⟶ p)›
```

It is sound on reflexive models:

```
lemma soundness_AxT: ‹AxT p ⟹ reflexive M ⟹ w ∈ 𝒲 M ⟹ M, w ⊨ p›
```

And forces the canonical model to be reflexive:

```
lemma reflexiveT:
  assumes ‹AxT ≤ A›
  shows ‹reflexive (canonical A)›
```

Example: System S4

Combine axioms T and 4 using:

```
‹(A ⊕ A') p ≡ A p ∨ A' p›
```

T forces reflexivity, 4 forces transitivity.
We combine those separate results with the generic theorem:

```
lemma strong_completeness⇩S⇩4: ‹G ⊫⇩S⇩4 p ⟹ G ⊢⇩S⇩4 p›
  using strong_completeness[of refltrans]
        reflexive⇩T[of ‹AxT ⊕ Ax4›]
        transitive⇩K⇩4[of ‹AxT ⊕ Ax4›]
```

We can reuse the previous result because `AxT ≤ AxT ⊕ Ax4`

Public Announcement Logic

Semantics Reminder

Semantics of p under public announcement of r:

```
| ‹M, w ⊨! [r]! p      ⟷      M, w ⊨! r ⟶ M[r!], w ⊨! p›

| ‹M[r!]               =      M ⦇𝒲 := {w. w ∈ 𝒲 M ∧ M, w ⊨! r}⦈›
```

In the restricted model, we only keep worlds that satisfy r.

For *static* formulas without announcements, the semantics coincide:

```
lemma lower_semantics:
  assumes ‹static p›
  shows ‹(M, w ⊨ lower p) ⟷ (M, w ⊨! p)›
```

Reduction to Static Formulas

We can *reduce* formulas to static equivalents:

```
lemma static_reduce: ‹static (reduce p)›
lemma reduce_semantics: ‹M, w ⊨! p ⟷ M, w ⊨! reduce p›
```

Our proof system is as before + reduction axioms (+ B next slide):

```
| PFF: ‹A; B ⊢! [r]! ⊥! ↔! (r ⟶! ⊥!)›
| PPro: ‹A; B ⊢! [r]! Pro! x ↔! (r ⟶! Pro! x)›
| PDis: ‹A; B ⊢! [r]! (p ∨! q) ↔! [r]! p ∨! [r]! q›
| PCon: ‹A; B ⊢! [r]! (p ∧! q) ↔! [r]! p ∧! [r]! q›
| PImp: ‹A; B ⊢! [r]! (p ⟶! q) ↔! ([r]! p ⟶! [r]! q)›
| PK: ‹A; B ⊢! [r]! K! i p ↔! (r ⟶! K! i ([r]! p))›
```

Allowed Announcements

We use B to restrict the *announcable* formulas:

```
| PAnn: ‹A; B ⊢! p ⟹ B r ⟹ A; B ⊢! [r]! p›
```

We can guarantee soundness over P-models when B-formulas preserve P:

```
theorem strong_soundnessP:
  assumes
    ‹⋀M w p. A p ⟹ P M ⟹ w ∈ 𝒲 M ⟹ M, w ⊨! p›
    ‹⋀M r. P M ⟹ B r ⟹ P (M[r!])›
  shows ‹A; B; G ⊢! p ⟹ P; G ⊫! p›
```

Completeness

We can lift "P-completeness" for static formulas:

```
theorem strong_static_completeness:
  assumes ‹static p› and ‹∀q ∈ G. static q› and ‹P; G ⊫! p›
    and ‹⋀G p. P; G ⊫ p ⟹ A o lift; G ⊢ p›
  shows ‹A; B; G ⊢! p›
```

And extend this to announcements using the reductions:

```
theorem strong_completenessₚ:
  assumes ‹P; G ⊫! p›
    and ‹∀r ∈ anns p. B r› ‹∀q ∈ G. ∀r ∈ anns q. B r›
    and ‹⋀G p. P; G ⊫ p ⟹ A o lift; G ⊢ p›
  shows ‹A; B; G ⊢! p›
```

Example: PAL + K4

We add axiom 4 and allow all announcements:

```
inductive AxP4 :: ‹'i pfm ⇒ bool› where
  ‹AxP4 (K! i p ⟶! K! i (K! i p))›
```

```
abbreviation SystemPK4 (‹_ ⊢!K4 _› [50, 50] 50) where
  ‹G ⊢!K4 p ≡ AxP4; (λ_. True); G ⊢! p›
```

Announcements preserve transitivity:

```
lemma transitive_restrict: ‹transitive M ⟹ transitive (M[r!])›
```

And we get soundness and completeness from the generic results:

```
theorem mainPK4: ‹G ⊫!K4 p ⟷ G ⊢!K4 p›
```

Summary

Covered Systems +/- Announcements

System	Axioms	Class
K		All frames
T	T	Reflexive frames
KB	B	Symmetric frames
K4	4	Transitive frames
K5	5	Euclidean frames
S4	T, 4	Reflexive and transitive frames
S5	T, B, 4	Frames with equivalence relations
S5'	T, 5	Frames with equivalence relations

Takeaways

- Prove results for a class of proof systems (`A`, `B`) from the start
- Isabelle/HOL encourages composable results
- Possible extension: *serial* Public Announcement Logic (actually use `B`)

Laura P. Gamboa Guzman recently built on this work:

> This work is a formalization of Stalnaker's epistemic logic with countably many agents and its soundness and completeness theorems, *as well as the equivalence between the axiomatization of S4 available in the Epistemic Logic theory and the topological one. It builds on the Epistemic Logic theory.*
>
> https://www.isa-afp.org/entries/Stalnaker_Logic.html