Uniform Interpolation & Bisimulation Quantifiers

Verified constructions via proof systems

Iris van der Giessen
University of Birmingham

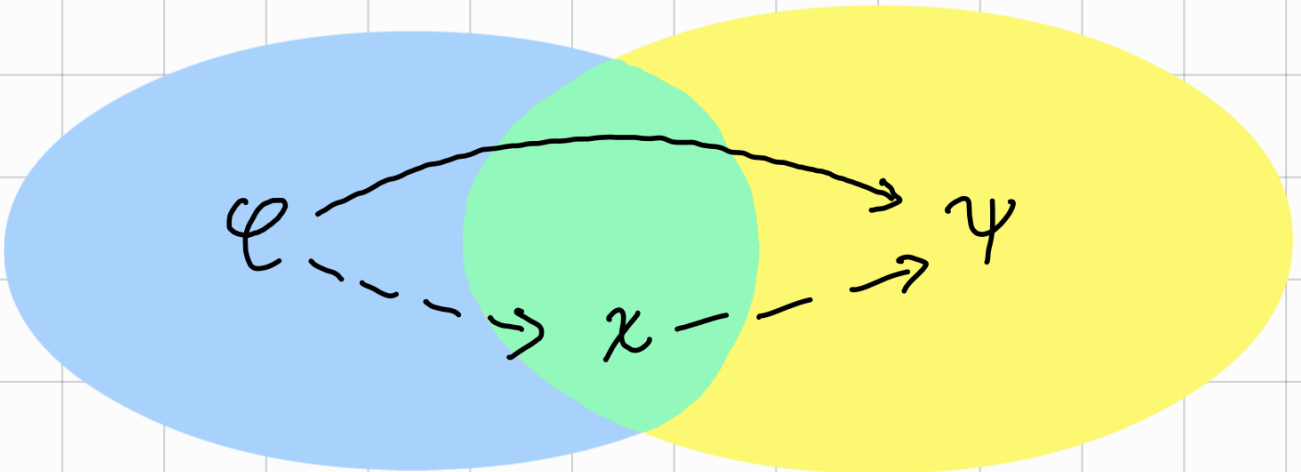
Craig interpolation workshop
Amsterdam 22-23 April 2024

Craig Interpolation

Language: $\varphi ::= p \mid \bot \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \Box \varphi$ (Var)

Def: Let L be a (modal) logic. L has the Craig interpolation property if for any $\vdash_L \varphi \to \psi$ there exists χ such that

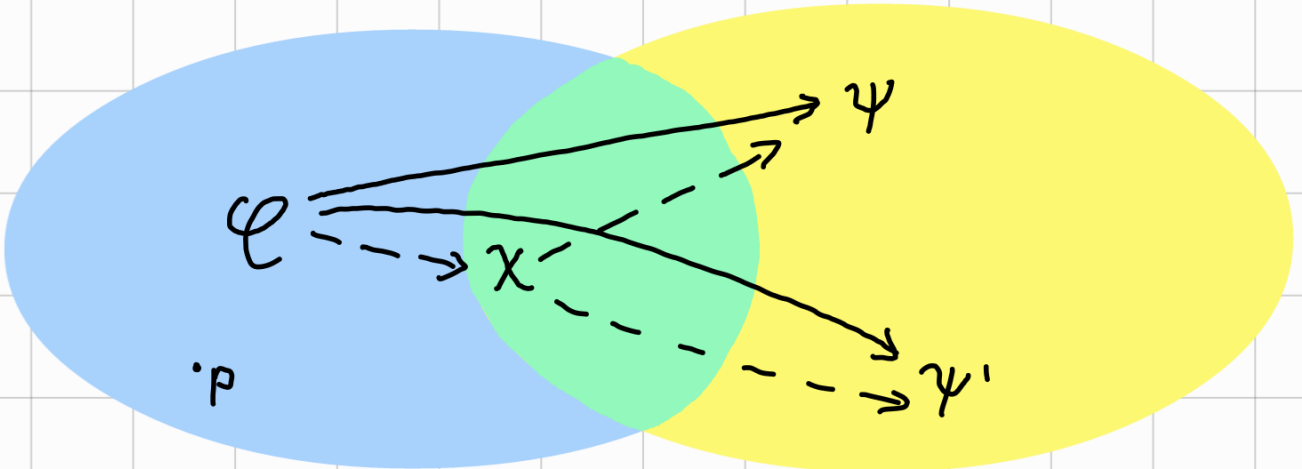
1. $Var(\chi) \subseteq Var(\varphi) \cap Var(\psi)$
2. $\vdash_L \varphi \to \chi$ and $\vdash_L \chi \to \psi$



Uniform Interpolation

Def: Let L be a (modal) logic. L has the (right) uniform interpolation property if for any formula φ and atom p there exists formula χ such that:

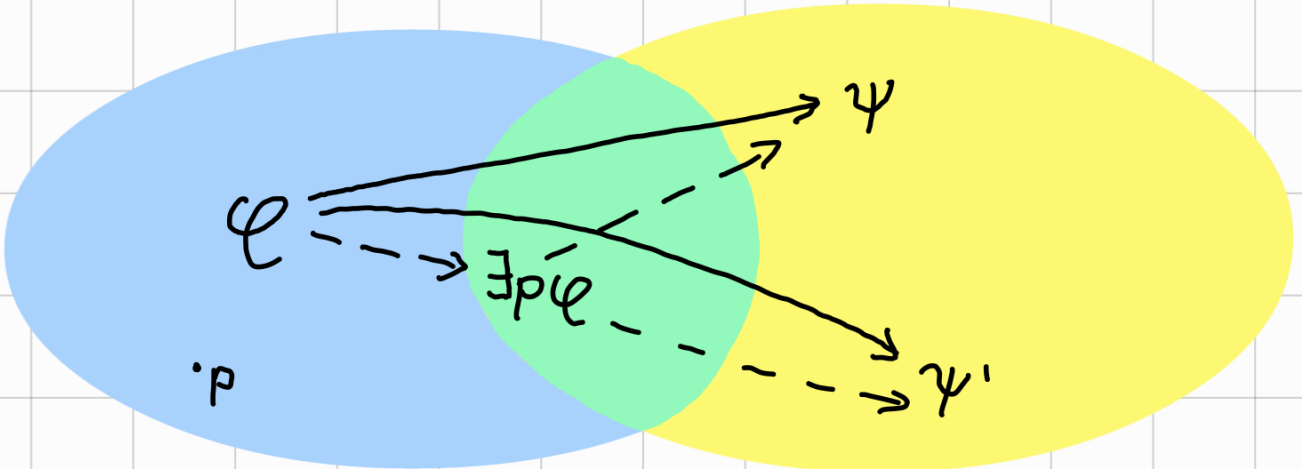
1. $Var(\chi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all ψ such that $p \notin Var(\psi)$
   $\vdash_L \varphi \to \psi$ iff $\vdash_L \chi \to \psi$



Uniform Interpolation

Def: Let L be a (modal) logic. L has the (right) uniform interpolation property if for any formula φ and atom p there exists formula $\exists p \varphi$ such that:

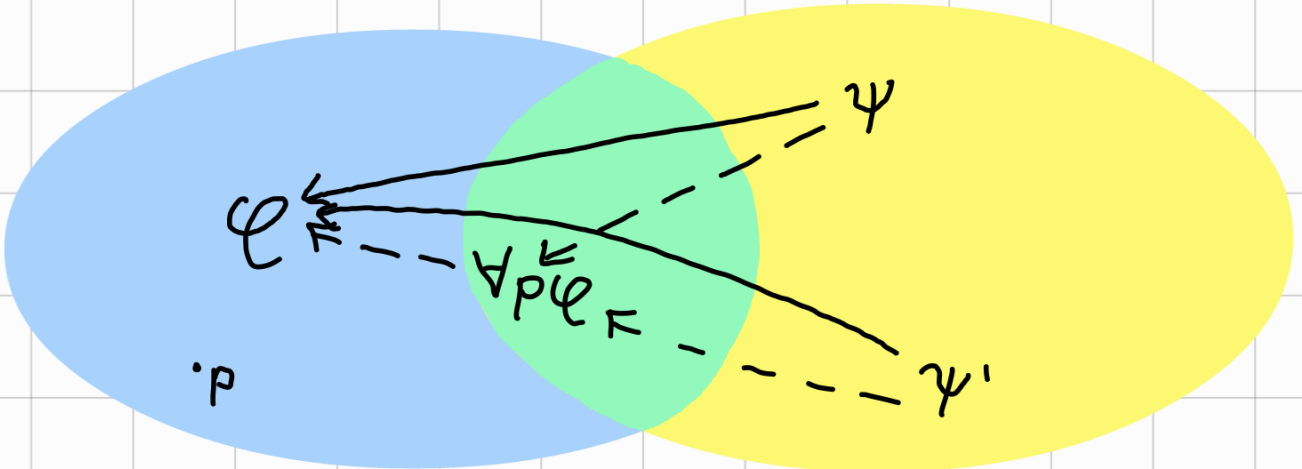
1. $Var(\exists p \varphi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all ψ such that $p \notin Var(\psi)$
   $\vdash_L \varphi \to \psi$ iff $\vdash_L \exists p \varphi \to \psi$



Uniform Interpolation

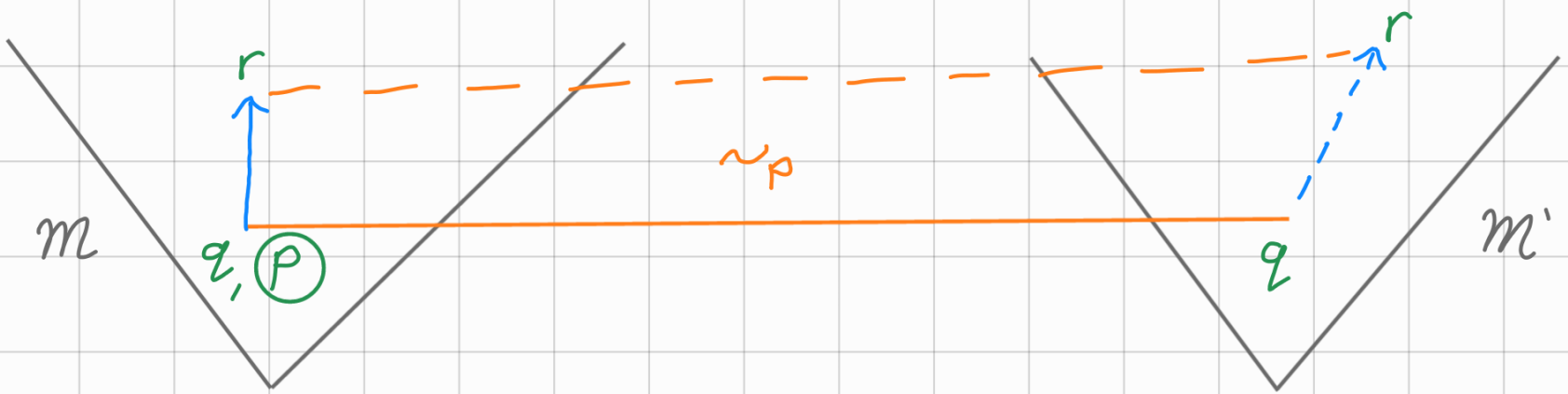
Def: Let L be a (modal) logic. L has the (left) uniform interpolation property if for any formula φ and atom p there exists formula $\forall p \varphi$ such that:

1. $Var(\forall p \varphi) \subseteq Var(\varphi) \setminus \{p\}$
2. For all ψ such that $p \notin Var(\psi)$
   $\vdash_L \varphi \to \psi$ iff $\vdash_L \forall p \varphi \to \psi$



Bisimulation quantifiers

- Bisimilar models modulo p, written $\mathcal{M} \sim_p \mathcal{M}'$:



- Modal equivalence: If $\mathcal{M} \sim_p \mathcal{M}'$, then for each ψ s.t. $p \notin Var(\psi)$:
  $\mathcal{M} \models \psi$ iff $\mathcal{M}' \models \psi$

Def: $\mathcal{M}, w \models \tilde{\forall} p \varphi$ iff for all $\mathcal{M}', w' \sim_p \mathcal{M}, w$: $\mathcal{M}', w' \models \varphi$

To prove uniform interpolation

Semantic proofs

- Bisimulation quantifiers in Kripke semantics
- Examples: K (Ghilardi, Zawadowski, '95), (Visser, '96)
  GL (Shavrukov, '94), (Visser, '96)

Proof theoretic proofs

- construction via proof systems
- Examples: IPC (Pitts '92)
  GL (Bílková, '06)
  iK and iKD (Iemhoff, '19)
  iSL (Férée, vd G., v Gool, Shillito, '24)

Constructions & correctness

Construction

--- proof theory ---

Syntactic proof of correctness

- uniform interpolant

--- syntax ---

Semantic proof of correctness

- bisimulation quantifier

--- semantics ---

Constructions via proof systems

Craig interpolation (Maehara)	uniform interpolation (Pitts '92)
Sequent style definition of Craig interpolation	Sequent style definition of uniform interpolation
interpolant is a formula	interpolant is a formula
induction along a cut-free proof	induction along finite proof search

NB: Iemhoff ('19): Uniform interpolation and "well-behaved" sequent calculi

Constructions & correctness

Construction

- construct uniform interpolant via sequents (Pitts, '92)

Syntactic proof of correctness

- uniform interpolant

--- syntax ---

Semantic proof of correctness

- bisimulation quantifier

--- semantics ---

Constructions & correctness

Construction

- construct uniform interpolant via sequents (Pitts, '92)

Syntactic proof of correctness

- uniform interpolant
- induction on proof search

Semantic proof of correctness

- bisimulation quantifier

--- semantics ---

Constructions & correctness

Construction

- construct uniform interpolant via sequents (Pitts, '92)
- construct bisimulation quantifier via nested sequents (vdG., Jalali, Kuznets, '23)

Syntactic proof of correctness

- uniform interpolant
- induction on proof search

Semantic proof of correctness

- bisimulation quantifier

--- semantics ---

Constructions & correctness

Construction

- construct uniform interpolant via sequents (Pitts, '92)
- construct bisimulation quantifier via nested sequents (vdG., Jalali, Kuznets, '23)

Syntactic proof of correctness

- uniform interpolant
- induction on proof search

Semantic proof of correctness

- bisimulation quantifier
- model modifications

Constructions & correctness

Construction

- construct uniform interpolant via sequents (Pitts, '92)
- construct bisimulation quantifier via nested sequents (vdG., Jalali, Kuznets, '23)

Syntactic proof of correctness

- uniform interpolant
- induction on proof search

- in Coq
- (Férée, vGool, '22): IPC

Semantic proof of correctness

- bisimulation quantifier
- model modifications

Why formalising a 30 years old proof?

1. Better understand the proof
2. Obtain a provably correct & executable implementation

   $\exists p\varphi$ and $\forall p\varphi$ are often not feasibly computable by hand
3. Use the implementation to derive theoretical results
   - ↳ Zoltan Kocsis: non-definability of some connectives (Taranovsky's realizability disjunction)
4. "Easily" derive uniform interpolant proofs for other logics
   - ↳ iSL
5. Check for bugs in the paper proof
   - ↳ GL

Constructions & correctness

Construction

- construct uniform interpolant via sequents (Pitts, '92)
- construct bisimulation quantifier via nested sequents (vdG., Jalali, Kuznets, '23)

Syntactic proof of correctness

- uniform interpolant
- induction on proof search

- in Coq
- (Férée, vGool, '22): IPC
- (Férée, vdG, vGool, Shillito, '24):
  - K
  - GL fix bug in (Bílková, '06)
  - iSL new

Semantic proof of correctness

- bisimulation quantifier
- model modifications

DEMO

The end of this talk ...

Construction

- construct uniform interpolant via sequents (Pitts, '92)
- construct bisimulation quantifier via nested sequents (vdG., Jalali, Kuznets, '23)

Syntactic proof of correctness

- uniform interpolant
- induction on proof search

- in Coq
- (Férée, vGool, '22): IPC
- (Férée, vdG, vGool, Shillito, '24):
  - K
  - GL fix bug in (Bílková, '06)
  - iSL new

Semantic proof of correctness

- bisimulation quantifier
- model modifications