

Simplification de formules et de contextes pour l'interpolation uniforme

(Formulas Rewritten And Normalized Computationally, And Intuitionistically Simplified)

Hugo Férée , Sam van Gool et Yago Iglesias Vazquez

JFLA, 28 janvier 2025

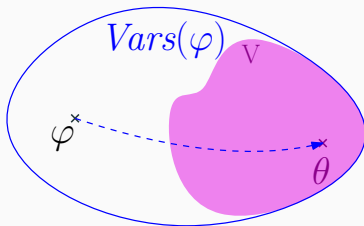
IRIF / Université Paris Cité

Interpolant uniforme (à la Pitts)

Définition (Interpolant uniforme à droite)

Une logique $\vdash_L$ a la propriété d'**interpolation uniforme** (à droite) si pour tout ϕ et $V \subseteq \text{Vars}(\phi)$, il existe θ :

- $\text{Vars}(\theta) \subseteq V$,
- $\vdash_L \phi \rightarrow \theta$, et
- pour tout ψ tel que $\text{Vars}(\phi) \cap \text{Vars}(\psi) \subseteq V$ et $\vdash_L \phi \rightarrow \psi$, alors $\vdash_L \theta \rightarrow \psi$.

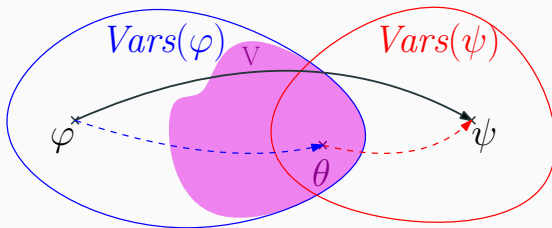


Interpolant uniforme (à la Pitts)

Définition (Interpolant uniforme à droite)

Une logique $\vdash_L$ a la propriété d'**interpolation uniforme** (à droite) si pour tout ϕ et $V \subseteq \text{Vars}(\phi)$, il existe θ :

- $\text{Vars}(\theta) \subseteq V$,
- $\vdash_L \phi \rightarrow \theta$, et
- pour tout ψ tel que $\text{Vars}(\phi) \cap \text{Vars}(\psi) \subseteq V$ et $\vdash_L \phi \rightarrow \psi$, alors $\vdash_L \theta \rightarrow \psi$.



Si $V = \text{Vars}(\phi) \setminus \{p\}$:

- l'interpolant à droite de ϕ est équivalent à $\exists p, \phi$
- l'interpolant à gauche de ϕ est équivalent à $\forall p, \phi$

Exemple (Logique classique propositionnelle)

$$\forall p, \phi \equiv \phi[\top/p] \wedge \phi[\perp/p]$$

$$\exists p, \phi \equiv \phi[\top/p] \vee \phi[\perp/p]$$

1992 Pitts démontre que la **logique intuitioniste propositionnelle (IL)** a l'interpolation uniforme (**IU**).

...

2023 van Gool, F. : formalisation en **Coq/Rocq** de la preuve de Pitts

2024 van der Giessen, van Gool, Shillito, F. : extension à **iSL** et d'autres logiques modales (toujours en **Coq/Rocq**)

1992 Pitts démontre que la **logique intuitioniste propositionnelle (IL)** a l'interpolation uniforme (**IU**).

...

2023 van Gool, F. : formalisation en **Coq/Rocq** de la preuve de Pitts

2024 van der Giessen, van Gool, Shillito, F. : extension à **iSL** et d'autres logiques modales (toujours en **Coq/Rocq**)



La fin de l'histoire ?

Besoin de calculer des interpolants de manière **effective** et **vérifiée** :

- Kocsis (arxiv 2023) :
« les connecteurs du second ordre ... ne sont pas définissables... »
- Jibladze : Implémentation Mathematica de « Pitts »

Mais la construction originale n'a que faire de l'efficacité :

- ✗ temps de calcul
- ✗ taille de la formule produite

Un calcul des séquents bien fondé

Formules : $\phi, \psi := p \mid \perp \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi$

Poids : $w(\wedge) = 2$; $w(\vee) = w(\rightarrow) = w(p) = w(\perp) = 1$.

Contexte : $\Gamma := \phi_1, \dots, \phi_n$ (multiensemble)

Séquents : $S := \Gamma \Rightarrow \psi$

Ordre : $S \prec S'$: ordre multiensembliste dérivé de $w(\cdot)$.

Calcul (G4IP) :
$$\frac{S_1 \quad \dots \quad S_n}{S} \quad \text{avec} \quad \forall i, S_i \prec S$$

Construction de Pitts (aperçu)

$\exists p, \phi := E_p(\{\phi\})$ et $\forall p, \phi := A_p(\emptyset \Rightarrow \phi)$,

avec E_p et A_p définie par induction bien fondée ($\prec$) :

$E_p(\Gamma) := \bigwedge \mathcal{E}_p(\Gamma)$ et $A_p(\Gamma \Rightarrow \phi) := \bigvee \mathcal{A}_p(\Gamma \Rightarrow \phi)$

	Γ est de la forme :	$\mathcal{E}_p(\Gamma)$ contient :
(E_p0)	$\Gamma', \perp$	$\perp$
(E_p1)	Γ', q	$E_p(\Gamma') \wedge q$
(E_p8)	$\Gamma', (\delta_1 \rightarrow \delta_2) \rightarrow \delta_3$	$[E_p(\Gamma', (\delta_2 \rightarrow \delta_3)) \rightarrow A_p(\Gamma', (\delta_2 \rightarrow \delta_3) \Rightarrow \delta_1 \rightarrow \delta_2)] \rightarrow E_p(\Gamma', \delta_3)$
...		
	s est de la forme :	$\mathcal{A}_p(s)$ contient :
(A_p3)	$\Gamma, \psi_1 \vee \psi_2 \Rightarrow \phi$	$[E_p(\Gamma, \psi_1) \rightarrow A_p(\Gamma, \psi_1 \Rightarrow \phi)] \wedge [E_p(\Gamma, \psi_2) \rightarrow A_p(\Gamma, \psi_2 \Rightarrow \phi)]$
...		

2×13 règles ; complexité doublement exponentielle

Exemple (simple)

$$\phi_n := (a_0 \wedge p) \wedge \cdots \wedge (a_{n-1} \wedge p)$$

$$E_p(\phi_n) = \bigwedge_{0 \leq i < n} E_p([a_0 \wedge p, \dots, a_{i-1} \wedge p, a_i, p, a_{i+1} \wedge p, \dots, a_{n-1} \wedge p])$$

Soit $n!$ appels récurifs; et une formule tout aussi grande.

Alors que : $E_p(\phi_n) \equiv a_0 \wedge \cdots \wedge a_{n-1}$

Le plan

Cette construction repose sur la terminaison de G4IP ; mais n'utilise aucune des propriétés de la logique ou du calcul des séquents.

Ingrédients :

- Inversibilité de certaines règles du calcul
- Admissibilité de l'affaiblissement
- Admissibilité de la coupure

Programmes :

- Procédure de décision pour IPC
- Simplification d'une formule dans un contexte
- Simplification d'un contexte

Procédure de décision

En théorie

Décroissance d'un ordre bien fondé sur chaque règle, donc $\vdash$ est décidable.

En pratique

Utiliser les propriétés des règles pour s'éviter une recherche exponentielle inutile :

- vérifier l'applicabilité d'un axiome ;
- puis appliquer en priorité les règles inversibles linéaires ;
- puis les autres règles inversibles ;
- puis les règles non-inversibles (nombreux branchements).

Remarque

Hudelmaier 1993 : algorithme en espace $\mathcal{O}(n \log(n))$; à implémenter en Coq.

Simplification d'une formule dans un contexte

Exemple

$$(p \wedge (q \rightarrow s)) \wedge ((q \rightarrow r) \wedge (p \rightarrow q)) \equiv p \wedge s \wedge r \wedge q$$

1. Simplifier $p \wedge (q \rightarrow s)$ sachant $(q \rightarrow r) \wedge (p \rightarrow q) : p \wedge s$
2. Simplifier $(q \rightarrow r) \wedge (p \rightarrow q)$ sachant $p \wedge s$:
 - a. Simplifier $q \rightarrow r$ sachant $p \wedge s$ et $p \rightarrow q : r$ (car $p \wedge s, p \rightarrow q \vdash q$)
 - b. Simplifier $p \rightarrow q$ sachant $p \wedge s$ et $r : q$ (car $p \wedge s, r \vdash p$)

- Si $\Delta \vdash \varphi$, alors $\text{simp}_{\Delta}(\varphi) = \top$ (procédure de décision sur les atomes)
- $\Delta \vdash (\text{simp}_{\Delta}(\varphi) \leftrightarrow \varphi)$
- $(\Delta, \varphi \vdash \theta) \iff (\Delta, \text{simp}_{\Delta}(\varphi) \vdash \theta)$ (admissibilité de la coupure)
- $w(\text{simp}_{\Delta}(\varphi)) \leq w(\varphi)$

Simplification des contextes

Algorithme ($\text{simp}(\Delta)$)

Réécrire itérativement Δ en Δ' si :

1. $\frac{\Delta' \vdash \cdot}{\Delta \vdash \cdot}$ est une instance d'une règle gauche linéaire inversible
2. $\Delta' = \Delta \setminus \{\varphi\}$ si $\Delta' \vdash \varphi$ (affaiblissement)
3. $\Delta' = \Delta \setminus \{\varphi\} \cup \{\text{simp}_{\Delta \setminus \{\varphi\}}(\varphi)\}$ si w décroît (coupure)

- Termine par $\prec$
- $(\Delta \vdash \theta) \leftrightarrow \text{simp}(\Delta) \vdash \theta$
- $\text{simp}(\Delta) \preceq \Delta$
- $\text{simp}(\Delta) \vdash \cdot$ n'est pas une instance d'une règle inversible (à gauche)

Algorithme de Pitts (mis à jour)

$$A_p(\Delta; \phi) := \bigvee \mathcal{A}_p(\text{simp}(\Delta); \phi)$$

$$E_p(\Delta) := \bigwedge \mathcal{E}_p(\text{simp}(\Delta))$$

✓ terminaison

✓ correction

(E _p 0)	$\Gamma', \perp$	$\perp$
(E _p 1)	Γ', q	$E_p(\Gamma') \wedge q$
(E _p 2)	$\Gamma', \psi_1 \wedge \psi_2$	$E_p(\Gamma', \psi_1, \psi_2)$
(E _p 3)	$\Gamma', \psi_1 \vee \psi_2$	$E_p(\Gamma', \psi_1) \vee E_p(\Gamma', \psi_2)$
(E _p 4)	$\Gamma', (q \rightarrow \psi)$	$q \rightarrow E_p(\Gamma', \psi)$
(E _p 5)	$\Gamma', r, (r \rightarrow \psi)$	$E_p(\Gamma', r, \psi)$
(E _p 6)	$\Gamma', (\delta_1 \wedge \delta_2) \rightarrow \delta_3$	$E_p(\Gamma', (\delta_1 \rightarrow (\delta_2 \rightarrow \delta_3)))$
(E _p 7)	$\Gamma', (\delta_1 \vee \delta_2) \rightarrow \delta_3$	$E_p(\Gamma', (\delta_1 \rightarrow \delta_3), (\delta_2 \rightarrow \delta_3)))$
(E _p 8)	$\Gamma', (\delta_1 \rightarrow \delta_2) \rightarrow \delta_3$	$[E_p(\Gamma', (\delta_2 \rightarrow \delta_3)) \rightarrow A_p(\Gamma', (\delta_2 \rightarrow \delta_3) \Rightarrow \delta_1 \rightarrow \delta_2)] \rightarrow E_p(\Gamma', \delta_3)$
(A _p 2)	$\Gamma', \psi_1 \wedge \psi_2 \Rightarrow \phi$	$A_p(\Gamma', \psi_1, \psi_2 \Rightarrow \phi)$
(A _p 3)	$\Gamma', \psi_1 \vee \psi_2 \Rightarrow \phi$	$[E_p(\Gamma', \psi_1) \rightarrow A_p(\Gamma', \psi_1 \Rightarrow \phi)] \wedge [E_p(\Gamma', \psi_2) \rightarrow A_p(\Gamma', \psi_2 \Rightarrow \phi)]$
(A _p 4)	$\Gamma', (q \rightarrow \psi) \Rightarrow \phi$	$q \wedge A_p(\Gamma', \psi \Rightarrow \phi)$
(A _p 5)	$\Gamma', r, (r \rightarrow \psi) \Rightarrow \phi$	$A_p(\Gamma', r, \psi \Rightarrow \phi)$
(A _p 6)	$\Gamma', (\delta_1 \wedge \delta_2) \rightarrow \delta_3 \Rightarrow \phi$	$A_p(\Gamma', (\delta_1 \rightarrow (\delta_2 \rightarrow \delta_3)) \Rightarrow \phi)$
(A _p 7)	$\Gamma', (\delta_1 \vee \delta_2) \rightarrow \delta_3 \Rightarrow \phi$	$A_p(\Gamma', (\delta_1 \rightarrow \delta_3), (\delta_2 \rightarrow \delta_3)) \Rightarrow \phi$
(A _p 10)	$\Gamma, p \Rightarrow p$	$\top$
(A _p 11)	$\Gamma \Rightarrow \phi_1 \wedge \phi_2$	$A_p(\Gamma \Rightarrow \phi_1) \wedge A_p(\Gamma \Rightarrow \phi_2)$
(A _p 13)	$\Gamma \Rightarrow \phi_1 \rightarrow \phi_2$	$E_p(\Gamma, \phi_1) \rightarrow A_p(\Gamma, \phi_1 \Rightarrow \phi_2)$
(A _p 8)	$\Gamma', (\delta_1 \rightarrow \delta_2) \rightarrow \delta_3 \Rightarrow \phi$	$[E_p(\Gamma', (\delta_2 \rightarrow \delta_3)) \rightarrow A_p(\Gamma', (\delta_2 \rightarrow \delta_3) \Rightarrow \delta_1 \rightarrow \delta_2)] \wedge A_p(\Gamma', \delta_3 \Rightarrow \phi)$
(A _p 9)	$\Gamma \Rightarrow q$	q
(A _p 12)	$\Gamma \Rightarrow \phi_1 \vee \phi_2$	$A_p(\Gamma \Rightarrow \phi_1) \vee A_p(\Gamma \Rightarrow \phi_2)$

Banc d'essai

θ	$w(\theta)$ original	$w(\theta)$ simp
$\exists p, \phi \rightarrow (6)$	131699*	25
$\forall p, \phi \rightarrow (7)$	263402*	27
$\exists p, \phi \rightarrow (7)$	—	25
$\exists p, \phi \rightarrow (8)$	—	25
$\exists p, \phi \rightarrow (9)$	—	—
$\exists p, \phi \wedge (3)$	42142*	10
$\exists p, \phi \wedge (4)$	—	13
$\forall p, r \leftrightarrow ((p \rightarrow q) \vee ((p \rightarrow q) \rightarrow q))$	84	4
$\exists p, r \leftrightarrow ((p \rightarrow q) \vee ((p \rightarrow q) \rightarrow q))$	—	5
$\exists p, (x \leftrightarrow (p \vee \neg p)) \wedge (y \leftrightarrow (q \vee \neg q)) \wedge \neg(p \wedge q)$	—	44
$\exists q, \exists p, (x \leftrightarrow (p \vee \neg p)) \wedge (y \leftrightarrow (q \vee \neg q)) \wedge \neg(p \wedge q)$	—	8
$\forall p, (q \rightarrow (p \vee r)) \rightarrow \neg(t \vee p)$	429	1
$\exists p, (q \rightarrow (p \vee r)) \rightarrow \neg(t \vee p)$	668	13
$\forall p, (\neg p \rightarrow q) \wedge (\neg\neg p \rightarrow q)$	33	1
$\exists p, (\neg p \rightarrow q) \wedge (\neg\neg p \rightarrow q)$	160	5

* : ≥ 2 minutes

— : ≥ 5 minutes

- **Procédures de simplification** de formules/contextes pour IPC
- **Généralisable** à d'autres logiques (iSL déjà supporté), étant donné :
 - un calcul des séquents bien fondé
 - des règles admissibles (affaiblissement, coupure)
 - des propriétés sur les règles (linéarité, inversibilité)
- **Intégration** transparente à l'algorithme de Pitts.
- Preuve papier $\rightsquigarrow$ preuve mécanisée, exécutable, **et effective**

Demo & code source : <https://hferee.github.io/UIML/demo.html>