

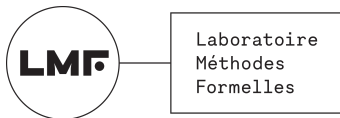
HOL-CSP : des erreurs à rattraper

Throw et Interrupt en HOL-CSP

Benoît Ballenghien¹

LMF - Paris-Saclay

January 30, 2025



Qu'est-ce que CSP ?

CSP = Communicating Sequential Processes

- **Quoi** : langage pour spécifier et vérifier des patrons de systèmes concurrents
- **Quand** : apparu dans les années 80, avec des évolutions significatives depuis
- **Qui** : pas mal de monde mais en particulier Brookes, Hoare et Roscoe
- **Domaine** : algèbres de processus (comme CCS ou LOTOS).

Syntaxe simplifiée supportée par HOL-CSP

$P ::=$	SKIP	peut seulement terminer
	STOP	interblocage (deadlock)
	$P \square P'$	choix externe : le processus suit les choix du contexte
	$P \sqcap P'$	choix interne : le contexte suit les choix du processus
	$P \llbracket A \rrbracket P'$	produit de synchronisation
	$P ; P'$	composition séquentielle
	$\square a \in A \rightarrow P \ a$	choix multi-préfixe externe

Syntaxe simplifiée supportée par HOL-CSP

$P ::=$	SKIP	peut seulement terminer
	STOP	interblocage (deadlock)
	$P \square P'$	choix externe : le processus suit les choix du contexte
	$P \sqcap P'$	choix interne : le contexte suit les choix du processus
	$P \llbracket A \rrbracket P'$	produit de synchronisation
	$P ; P'$	composition séquentielle
	$\square a \in A \rightarrow P \ a$	choix multi-préfixe externe

On signale la terminaison par un événement spécial : ✓.

Facettes dénotationnelles, algébriques et opérationnelles

Dénotationnelle :

$$\begin{aligned}\mathcal{F} (P \llbracket S \rrbracket Q) &\equiv \dots & \mathcal{D} (P \llbracket S \rrbracket Q) &\equiv \dots \\ P \sqsubseteq_{FD} Q &\equiv \mathcal{F} Q \subseteq \mathcal{F} P \wedge \mathcal{D} Q \subseteq \mathcal{D} P\end{aligned}$$

Algébrique :

$$\frac{A \subseteq S \quad B \subseteq S}{\Box_{a \in A} \rightarrow P \ a \llbracket S \rrbracket (\Box_{b \in B} \rightarrow Q \ b) = \Box_{x \in A \cap B} \rightarrow P \ x \llbracket S \rrbracket Q \ x}$$

Opérationnelle :

$$\frac{e \notin S \quad P \rightsquigarrow_e P'}{P \llbracket S \rrbracket Q \rightsquigarrow_e P' \llbracket S \rrbracket Q} \qquad \frac{P \rightsquigarrow_{\checkmark} P' \quad Q \rightsquigarrow_{\tau} Q'}{P ; Q \rightsquigarrow_{\tau} Q'}$$

HOL-CSP en quelques mots

HOL-CSP [2] a été développé par Safouan Taha, Burkhart Wolff, Lina Ye et plus récemment Benoît Ballenghien. Les théories sont disponibles sur l'AFP (Archive of Formal Proofs) <https://www.isa-afp.org/entries/HOL-CSPM.html>.

Points principaux :

- théorie de processus
- **pas de restriction de finitude**
- **pas de restriction sur les types** : 'a process
- construction dénotationnelle, les lois algébriques sont prouvées

HOL-CSP en quelques mots

HOL-CSP [2] a été développé par Safouan Taha, Burkhart Wolff, Lina Ye et plus récemment Benoît Ballenghien. Les théories sont disponibles sur l'AFP (Archive of Formal Proofs) <https://www.isa-afp.org/entries/HOL-CSPM.html>.

Points principaux :

- théorie de processus
- **pas de restriction de finitude**
- **pas de restriction sur les types** : 'a process
- construction dénotationnelle, les lois algébriques sont prouvées
- pour l'instant, les opérateurs Throw et Interrupt ne sont pas supportés.

$P ::= P \triangle P'$	interruption d'un processus par un autre
$ P \Theta a \in A. P' \ a$	gestion des exceptions

HOL-CSP en quelques flèches



HOL-CSP en quelques flèches



axiomatisation

HOL

Higher-Order Logic

HOL-CSP en quelques flèches



axiomatisation

HOL

Higher-Order Logic

extension conservative

HOLCF

Logic for Computable Functions de Scott

HOL-CSP en quelques flèches



axiomatisation

HOL

Higher-Order Logic

extension conservative

HOLCF

Logic for Computable Functions de Scott

extension conservative

HOL-CSP

L'implémentation existante

HOL-CSP en quelques flèches



axiomatisation

HOL

Higher-Order Logic

extension conservative

HOLCF

Logic for Computable Functions de Scott

extension conservative

HOL-CSP

L'implémentation existante

extension conservative

$P \triangle Q$

Opérateurs Throw et Interrupt

$P \Theta a \in A. Q \ a$

Résumé de la construction dénotationnelle

```
datatype 'a event = ev 'a | tick (✓)
```

Résumé de la construction dénotationnelle

`datatype` 'a event = ev 'a | tick ($\langle \checkmark \rangle$)

`type_synonym` 'a trace = $\langle$ 'a event list $\rangle$

`type_synonym` 'a refusal = $\langle$ 'a event set $\rangle$

`type_synonym` 'a failure = $\langle$ 'a trace $\times$ 'a refusal $\rangle$

`type_synonym` 'a divergence = $\langle$ 'a trace $\rangle$

`type_synonym` 'a process₀ = $\langle$ 'a failure set $\times$ 'a divergence set $\rangle$

Résumé de la construction dénotationnelle

`datatype` 'a event = ev 'a | tick ($\langle \checkmark \rangle$)

`type_synonym` 'a trace = $\langle$ 'a event list $\rangle$

`type_synonym` 'a refusal = $\langle$ 'a event set $\rangle$

`type_synonym` 'a failure = $\langle$ 'a trace $\times$ 'a refusal $\rangle$

`type_synonym` 'a divergence = $\langle$ 'a trace $\rangle$

`type_synonym` 'a process₀ = $\langle$ 'a failure set $\times$ 'a divergence set $\rangle$

`definition` is_process :: $\langle$ 'a process₀ $\Rightarrow$ bool $\rangle$ `where` ...

`typedef` 'a process = $\langle \{P :: 'a \text{ process}_0 . \text{is_process } P\} \rangle$

Invariant is_process (Roscoe, 2010) [1]

$$T1 : [] \in \mathcal{T} P$$

$$T2 : \forall t X. (t, X) \in \mathcal{F} P \longrightarrow \text{ftF } t$$

$$T3 : \forall t u. t @ u \in \mathcal{T} P \longrightarrow t \in \mathcal{T} P$$

$$T4 : \forall t X Y. (t, Y) \in \mathcal{F} P \wedge X \subseteq Y \longrightarrow (t, X) \in \mathcal{F} P$$

$$T5 : \forall t X Y. (t, X) \in \mathcal{F} P \wedge (\forall e. e \in Y \longrightarrow t @ [e] \notin \mathcal{T} P) \longrightarrow (t, X \cup Y) \in \mathcal{F} P$$

$$T6 : \forall t X. t @ [\checkmark] \in \mathcal{T} P \longrightarrow (t, X - \{\checkmark\}) \in \mathcal{F} P$$

$$T7 : \forall t u. t \in \mathcal{D} P \wedge tF t \wedge \text{ftF } u \longrightarrow t @ u \in \mathcal{D} P$$

$$T8 : \forall t X. t \in \mathcal{D} P \longrightarrow (t, X) \in \mathcal{F} P$$

$$T9 : \forall t. t @ [\checkmark] \in \mathcal{D} P \longrightarrow t \in \mathcal{D} P$$

Exemple du choix interne

```
lift_definition Ndet :: ⟨'a process, 'a process⟩ ⇒ 'a process (infixl ⟨⊔⟩ 80)
  is ⟨λP Q. (ℱ P ∪ ℱ Q, ℰ P ∪ ℰ Q)⟩
proof -
  show ⟨is__process (ℱ P ∪ ℱ Q, ℰ P ∪ ℰ Q)⟩ for P Q :: ⟨'a process⟩
    by (unfold is__process_def)
      (auto intro: ...)
qed
```

Exemple du choix interne

```
lift_definition Ndet :: ⟨['a process, 'a process] ⇒ 'a process⟩ (infixl ⟨⊔⟩ 80)
  is ⟨λP Q. (ℱ P ∪ ℱ Q, ℰ P ∪ ℰ Q)⟩
```

```
proof -
```

```
  show ⟨is_process (ℱ P ∪ ℱ Q, ℰ P ∪ ℰ Q)⟩ for P Q :: ⟨'a process⟩
```

```
    by (unfold is_process_def)
```

```
      (auto intro: ...)
```

```
qed
```

```
lemma prefix_Det_Ndet : ⟨(a → P) ⊔ (a → Q) = (a → P) ⊔ (a → Q)⟩
```

Intuition

L'opérateur $P \triangle Q$ est censé :

- se comporter comme P dans un premier temps
- sauf qu'à n'importe quel moment Q peut démarrer
- et P est alors abandonné.

Intuition

L'opérateur $P \triangle Q$ est censé :

- se comporter comme P dans un premier temps
- sauf qu'à n'importe quel moment Q peut démarrer
- et P est alors abandonné.

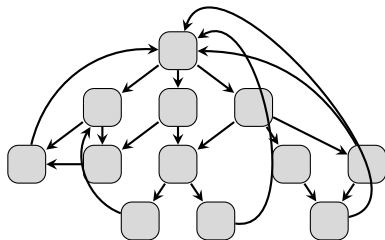
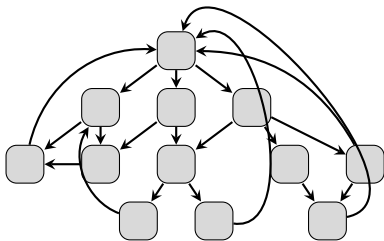


Figure: Fermer une interface graphique avec Interrupt

Intuition

L'opérateur $P \triangle Q$ est censé :

- se comporter comme P dans un premier temps
- sauf qu'à n'importe quel moment Q peut démarrer
- et P est alors abandonné.



$\triangle \quad \times \rightarrow \text{CLOSE}$

Figure: Fermer une interface graphique avec Interrupt

Définitions de la littérature (Roscoe, 2010) [1]

- $\mathcal{D} (P \triangle Q) = \mathcal{D} P \cup \{t @ u \mid t \in \mathcal{T} P \wedge t \text{F} t \wedge u \in \mathcal{D} Q\}$
- $\mathcal{F} (P \triangle Q) =$
 $\{(t, X) \mid (t, X) \in \mathcal{F} P \wedge ([], X) \in \mathcal{F} Q\} \cup$
 $\{(t @ u, X) \mid t \in \mathcal{T} P \wedge t \text{F} t \wedge (u, X) \in \mathcal{F} Q \wedge u \neq []\} \cup$
 $\{(t, X) \mid t \in \mathcal{D} P\} \cup$
 $\{(t @ u, X) \mid t \in \mathcal{T} P \wedge t \text{F} t \wedge u \in \mathcal{D} Q\}$

Définitions de la littérature (Roscoe, 2010) [1]

- $\mathcal{D} (P \triangle Q) = \mathcal{D} P \cup \{t @ u \mid t \in \mathcal{T} P \wedge tF t \wedge u \in \mathcal{D} Q\}$
- $\mathcal{F} (P \triangle Q) =$
 $\{(t, X) \mid (t, X) \in \mathcal{F} P \wedge ([], X) \in \mathcal{F} Q\} \cup$
 $\{(t @ u, X) \mid t \in \mathcal{T} P \wedge tF t \wedge (u, X) \in \mathcal{F} Q \wedge u \neq []\} \cup$
 $\{(t, X) \mid t \in \mathcal{D} (P \triangle Q)\}$

Conditions transgressées par cette définition

T5 : $\forall t X Y. (t, X) \in \mathcal{F} P \wedge (\forall e. e \in Y \longrightarrow t @ [e] \notin \mathcal{T} P) \longrightarrow (t, X \cup Y) \in \mathcal{F} P$
avec $\text{SKIP} \triangle e \rightarrow P$ comme contre-exemple.

T6 : $\forall t X. t @ [\checkmark] \in \mathcal{T} P \longrightarrow (t, X - \{\checkmark\}) \in \mathcal{F} P$
avec $\text{SKIP} \triangle e \rightarrow P$ et $e \rightarrow P \triangle \text{SKIP}$ comme contre-exemples.

Définitions corrigées

- $\mathcal{F}(P \triangle Q) =$
 $\{(t @ [\checkmark], X) \mid t @ [\checkmark] \in \mathcal{T} P\} \cup \{(t, X - \{\checkmark\}) \mid t @ [\checkmark] \in \mathcal{T} P\} \cup$
 $\{(t, X) \mid (t, X) \in \mathcal{F} P \wedge tF t \wedge ([], X) \in \mathcal{F} Q\} \cup$
 $\{(t @ u, X) \mid t \in \mathcal{T} P \wedge tF t \wedge (u, X) \in \mathcal{F} Q \wedge u \neq []\} \cup$
 $\{(t, X - \{\checkmark\}) \mid t \in \mathcal{T} P \wedge tF t \wedge [\checkmark] \in \mathcal{T} Q\} \cup$
 $\{(t, X) \mid t \in \mathcal{D} P\} \cup$
 $\{(t @ u, X) \mid t \in \mathcal{T} P \wedge tF t \wedge u \in \mathcal{D} Q\}$

Définitions corrigées

- $\mathcal{F} (P \triangle Q) =$
 $\{(t @ [\checkmark], X) \mid t @ [\checkmark] \in \mathcal{T} P\} \cup \{(t, X - \{\checkmark\}) \mid t @ [\checkmark] \in \mathcal{T} P\} \cup$
 $\{(t, X) \mid (t, X) \in \mathcal{F} P \wedge tF t \wedge ([], X) \in \mathcal{F} Q\} \cup$
 $\{(t @ u, X) \mid t \in \mathcal{T} P \wedge tF t \wedge (u, X) \in \mathcal{F} Q \wedge u \neq []\} \cup$
 $\{(t, X - \{\checkmark\}) \mid t \in \mathcal{T} P \wedge tF t \wedge [\checkmark] \in \mathcal{T} Q\} \cup$
 $\{(t, X) \mid t \in \mathcal{D} P\} \cup$
 $\{(t @ u, X) \mid t \in \mathcal{T} P \wedge tF t \wedge u \in \mathcal{D} Q\}$
- $\mathcal{T} (P \triangle Q) = \mathcal{T} P \cup \{t @ u \mid t \in \mathcal{T} P \wedge tF t \wedge u \in \mathcal{T} Q\}$

Lois prouvées

- **Associativité** : $P \triangle (Q \triangle R) = P \triangle Q \triangle R$
- **Neutre** : $P \triangle \text{STOP} = P$ $\text{STOP} \triangle P = P$
- **Distributivité du non-déterminisme** :
 $P \triangle Q1 \sqcap Q2 = (P \triangle Q1) \sqcap (P \triangle Q2)$
 $P1 \sqcap P2 \triangle Q = (P1 \triangle Q) \sqcap (P2 \triangle Q)$
- **Loi de pas** : $(\Box_{a \in A} \rightarrow P \ a) \triangle Q = Q \sqcap (\Box_{a \in A} \rightarrow P \ a \triangle Q)$

Quelques mots sur Throw

L'opérateur $P \Theta a \in A. Q$ a est censé :

- se comporter comme P dans un premier temps
- sauf que si P performe un élément $a \in A$, alors Q a démarre
- et P est alors abandonné.

Quelques mots sur Throw

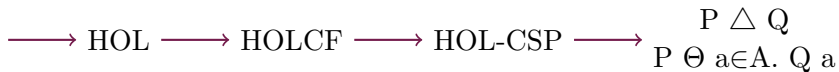
L'opérateur $P \Theta a \in A. Q$ a est censé :

- se comporter comme P dans un premier temps
- sauf que si P performe un élément $a \in A$, alors Q a démarre
- et P est alors abandonné.

En résumé :

- preuve que les définitions de la littérature sont correctes,
- preuve des lois algébriques attendues
- et nouvelle technique de preuve pour la continuité (voir le papier).

Conclusion



- Addition de deux nouveaux opérateurs à HOL-CSP
- Correction d'une définition de la littérature et preuve que l'invariant est désormais conservé
- Preuve des lois algébriques attendues

Annexe : lois de Throw et lemme pour la continuité

- **Alphabet disjoint** : $A \cap \alpha(P) = \emptyset \implies P \Theta a \in A. Q a = P$

- **Distributivité du non-déterminisme** :

$$P1 \sqcap P2 \Theta a \in A. Q a = (P1 \Theta a \in A. Q a) \sqcap (P2 \Theta a \in A. Q a)$$

$$P \Theta a \in A. Q1 a \sqcap Q2 a = (P \Theta a \in A. Q1 a) \sqcap (P \Theta a \in A. Q2 a)$$

- **Loi de pas** :

$$(\Box a \in A \rightarrow P a) \Theta b \in B. Q b = \Box a \in A \rightarrow (\text{if } a \in B \text{ then } Q a \text{ else } P a \Theta b \in B. Q b)$$

- **Lemme pour la continuité** :

$$\llbracket \forall i. S i \neq \emptyset; \text{finite } (S 0); \forall i. S (\text{Suc } i) \subseteq S i \rrbracket \implies (\bigcap_i S i) \neq \emptyset$$

- [1] A. Roscoe. *Understanding Concurrent Systems*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010. ISBN 184882257X, 9781848822573.
- [2] S. Taha, L. Ye, and B. Wolff. HOL-CSP Version 2.0. *Archive of Formal Proofs*, Apr. 2019. ISSN 2150-914x. URL <http://isa-afp.org/entries/HOL-CSP.html>.