

# Construction d'une sémantique concurrente par instrumentation d'une sémantique opérationnelle structurelle L'exemple du langage Orc

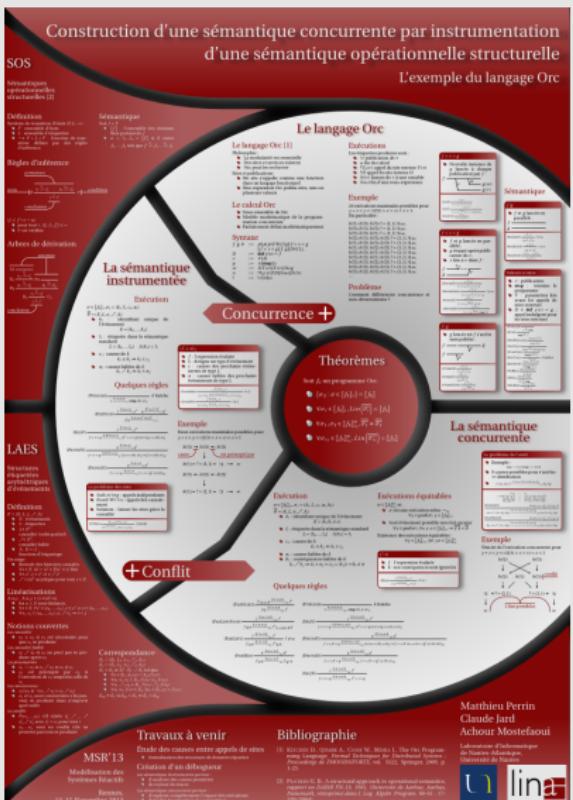
Matthieu Perrin

Claude Jard

Achour Mostefaoui

Université de Nantes, LINA

MSR'13 — 13-15/09/2013



# Le langage Orc

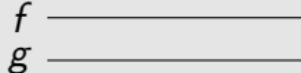
## Domaines d'application :

- ▶ orchestration du web
- ▶ véritable modèle de la programmation concurrente

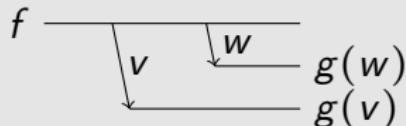
## Particularités :

- ▶ utilisation de sites
- ▶ publications multiples
- ▶ connecteurs originaux

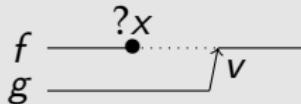
▶  $f|g$



▶  $f >x> g$



▶  $f <x< g$



▶  $f;g$



# La sémantique opérationnelle structurelle

## Sémantique :

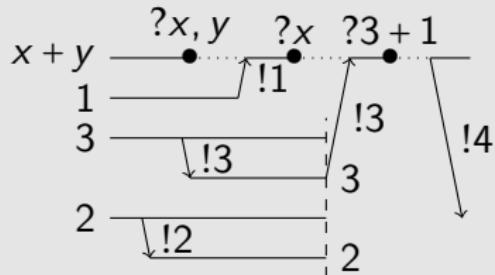
- ▶ système de transition d'états
- ▶  $\llbracket f \rrbracket$  : l'ensemble des chemins partant de  $f$

## Quelques règles :

$$\frac{f \xrightarrow{!} f'}{f|g \xrightarrow{!} f'|g}$$
$$\frac{g \xrightarrow{!v} g'}{f < x < g \xrightarrow{h(!v)} [v/x]f}$$
$$\frac{f \xrightarrow{!v} f'}{f > x > g \xrightarrow{h(!v)} f' > x > g|[v/x]g}$$

## Exemple :

$$y + z < y < ((2|3) > x > x) < z < 1$$

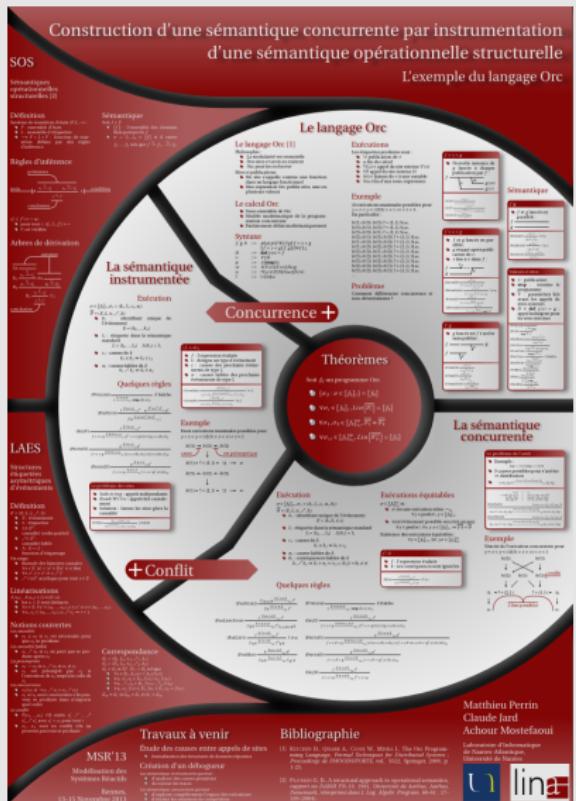


## $\llbracket f \rrbracket$ :

- ▶  $h(!1).h(!3).h(!3).? + (3, 1).!4.\omega$
- ▶  $h(!3).h(!1).h(!3).? + (3, 1).!4.\omega$
- ▶  $h(!3).h(!3).h(!1).? + (3, 1).!4.\omega$
- ▶  $h(!2).h(!1).h(!3).h(!3).? + (3, 1).!4.\omega$
- ▶  $h(!1).h(!2).h(!3).h(!3).? + (3, 1).!4.\omega$
- ▶  $h(!1).h(!3).h(!2).h(!3).? + (3, 1).!4.\omega$
- ▶ ...

# Deux nouvelles sémantiques !

- ▶ une sémantique instrumentée
- ▶ une sémantique concurrente



Merci de votre attention

MSR'13  
Modélisation des Systèmes Réactifs  
13-15 Novembre 2013

## Travaux à venir

- Etude des outils mis en oeuvre pour la construction de la sémantique
- Création d'un débuggeur
- Étude et mise en œuvre de l'interprétation
- Étude et mise en œuvre de la compilation
- Étude et mise en œuvre de la génération de code

## Bibliographie

- 10. Mousavi, M., Gehr, A., and W. Weis. J. The Orc Programming Language: Formal Definition by Operational Semantics. In: Formal Methods for Programming Languages, ed. G. B. Simeoni. Springer, 2013, p. 2-25.
- 11. Pfenning, F., and D. Walker. Totality: A Type System for Program Verification. In: Formal Methods for Programming Languages, ed. G. B. Simeoni. Springer, 2013, p. 26-42.

Matthieu Perrin  
Claude Jard  
Achour Mostéfaoui  
Laboratoire d'Informatique de Paris  
Université de Paris

