Le λ-calcul : sémantiques, information et points fixes

Gérard Berry

Collège de France
Chaire Informatique et sciences numériques

Cours 3, 9 décembre 2009

Questions sémantiques

- Qu'est ce qu'un modèle du λ-calcul ?
- Quelle est la bonne théorie des fonctions partielles ?
- Quelles sont les fonctions calculées par un langage comme CAML ? Quel est le rapport entre syntaxe et modèle ?
- Que signifient la stabilité et la séquentialité en sémantique ?

Les idées de base

- Interpréter un lambda-terme comme une fonction classique d'ensemble dans ensemble.
- D. Scott : rendre les fonctions totales par un indéfini explicite; définir un ordre d'information; requérir croissance et continuité, passer à l'infini
- G. Plotkin : travailler sur PCF, un « vrai » langage
- R. Milner : construire un modèle complètement adéquat de PCF, i.e. reflétant exactement la syntaxe
- G. Berry / P-L. Curien, *et. al.* : comprendre la structure sémantique de ce modèle

  causalité => stabilité, stratégie => séquentialité

Domaines sémantiques

- Types de base : b = booléens, i = entiers
- Domaine de base : $D^b = \mathbb{B} = \{tt, ff\}$, $D^i = \mathbb{N} = \{0, 1, ..., n...\}$
- Domaine fonctionnels : $D^{(\sigma\to\tau)} = (D^\sigma \to D^\tau)$

- Environnement : $\rho(x^\sigma) \in D^\sigma$, i.e. $\rho \in Env = \cup_\sigma (V^\sigma \to D^\sigma)$

  $\rho[x^\sigma \leftarrow a]$ pour $a \in D^\sigma$ est défini ainsi :

  1. $\rho[x^\sigma \leftarrow a]\ (x^\sigma) = a$
  2. $\rho[x^\sigma \leftarrow a]\ (y^\tau) = \rho(y^\tau)$ pour $y^\tau \neq x^\sigma$

λ-variables x en normal, math-variables a en italique

Sémantique des termes

$[[M^{\sigma}]]\,(\rho) \in D^{\sigma}$ i.e. $[[M^{\sigma}]] \in Env \rightarrow D^{\sigma}$

$$[[\, x^{\sigma} \,]]\,(\rho) = \rho(x^{\sigma})$$

$$[[\lambda x^{\sigma}.\, M^{\tau} \,]]\,(\rho)\,(a) = [[M\,]](\rho[x^{\sigma} \leftarrow a]) \quad \text{pour } a \in D^{\sigma}$$

$$[[\, M^{(\sigma \rightarrow \tau)}\, N^{\sigma} \,]]\,(\rho) = ([[M^{(\sigma \rightarrow \tau)}]]\,(\rho))\,([[N^{\sigma}]]\,(\rho))$$

Exemples

- $[[\lambda x^{\sigma}.\ x^{\sigma}]]\ (\rho)(a) = [[x^{\sigma}]]\ (\rho[x^{\sigma} \leftarrow a])$
  $= (\rho[x^{\sigma} \leftarrow a])\ (x^{\sigma}) = a$

  $[[\lambda x^{\sigma}.\ x^{\sigma}]]\ (\rho) =$ identité de $D^{\sigma} \rightarrow D^{\sigma}$

- $[[\lambda x^{\sigma}.\lambda y^{\tau}.\ x^{\sigma}]]\ (\rho)(a)(b) = [[x^{\sigma}]]\ (\rho[x^{\sigma} \leftarrow a][y^{\tau} \leftarrow b])$
  $= (\rho[x^{\sigma} \leftarrow a][y^{\tau} \leftarrow b])(x^{\sigma}) = a$

  $[[\lambda x^{\sigma}.\lambda y^{\tau}.\ x^{\sigma}]]\ (\rho) = \pi_1 : D^{\sigma} \rightarrow D^{\tau} \rightarrow D^{\sigma}$
  $=$ première projection

- $[[\lambda g.\lambda f.\ \lambda f.g(f(x))]]\ (\rho)(u)(v)(a)$
  $= [[g(f(x))]]\ (\rho[g \leftarrow u][f \leftarrow v]\ [x \leftarrow a])$
  $= u(v(a))$

La réduction préserve la sémantique

Théorème : si $M \rightarrow N$ alors $[[M]] = [[N]]$

preuve : par récurrence structurelle sur M

(i.e., par récurrence sur la taille de M)

Remarque : la réduction ne change pas l'information,

et même la réduit, mais elle la rend plus lisible

3+5 a valeur 8 mais est plus précis que 8

puisque 4+4 a aussi valeur 8 !

La langage PCF: vers la programmation

Partir du lambda-calcul typé, ajouter les booléens, les entiers et les fonctions de base, ajouter Y

- Types de base b pour les booléens, i pour les entiers
- Constantes booléennes tt^b et ff^b
- Constante entière $\underline{n}^i$ pour tout entier *n*
- Fonctions $\underline{+1}^{(i\to i)}$, $\underline{-1}^{(i\to i)}$ et $\underline{0?}^{(i\to b)}$
- Conditionnelles $\underline{cond}^{b\to i\to i\to i}$ et $\underline{cond}^{b\to b\to b\to b}$
- Un combinateur $Y^{(\sigma\to\sigma)\to\sigma}$ pour chaque type σ

$$\text{Fact}^{(i\to i)} = Y^{(i\to i)\to(i\to i)\to(i\to i)}$$
$$(\lambda f^{(i\to i)}.\ \lambda x^{i}.\ \underline{\text{cond}}\ (\underline{0?}\ x)\ \underline{1}\ \text{Mult}(x, f(\underline{-1}\ x)))$$

PCF est Turing-complet
Il peut être vu comme la base de ML / CAML

Interprétation de PCF

$$[\![\ \underline{\text{tt}}\ ]\!]\ (\rho) = tt$$

$$[\![\ \underline{\text{ff}}\ ]\!]\ (\rho) = ff$$

$$[\![\ \underline{\text{n}}\ ]\!]\ (\rho) = n$$

$$[\![\ \underline{+1}\ ]\!]\ (\rho)\ (n) = n+1$$

$$[\![\ \underline{0?}\ ]\!]\ (\rho)\ (n) = tt \text{ si } n=0$$

$$= ff \text{ si } n\neq 0$$

$$[\![\ \underline{\text{cond}_b}\ ]\!]\ (\rho)\ (c)\ (b_1)\ (b_2) = b_1 \text{ si } c=tt$$

$$= b_2 \text{ si } c=ff$$

$$[\![\ \underline{\text{cond}_i}\ ]\!]\ (\rho)\ (c)\ (n_1)\ (n_2) = n_1 \text{ si } c=tt$$

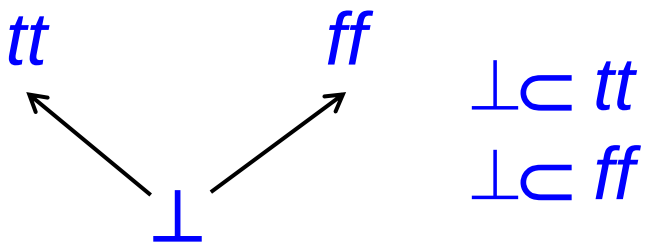
$$= n_2 \text{ si } c=ff$$

- Quid des programmes qui bouclent ?
- Quid du combinateur de point fixe Y ?

$$[\![\ Y^{(\sigma\to\sigma)\to\sigma}\ ]\!]\ (\rho) = ???$$

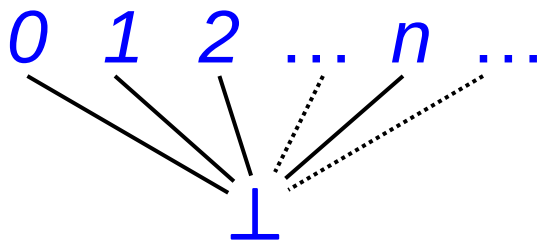
D. Scott : rendre les fonctions totales
ordre d'information
point fixe sémantique y avec $[\![\ Y\ ]\!] = y$

Les domaines de Scott



$\mathbb{O}_\perp$: espace de Sierpinski

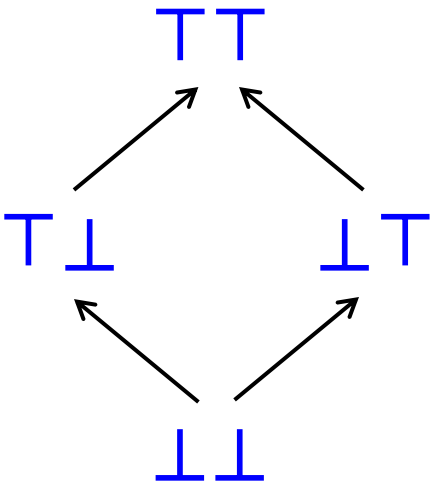
$\mathbb{B}_\perp$: booléens de Scott



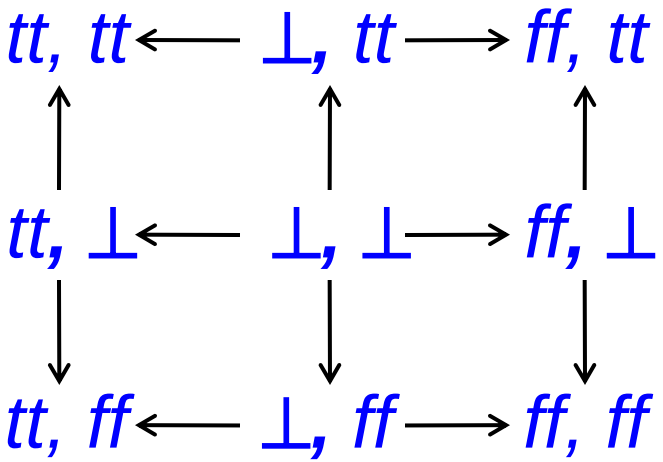
$\mathbb{N}_\perp$: entiers de Scott

Produit de domaines de Scott

$\langle D,\subset,\bot\rangle \times \langle D',\subset',\bot'\rangle$: $(x,y) \subset (x',y')$ ssi $x \subset x'$ et $y \subset' y'$



$O_\bot \times O_\bot$



$B_\bot \times B_\bot$

Fonctions croissantes

- Plus on en donne, plus on en récupère !

  f : $\langle D, \subset, \bot \rangle \rightarrow \langle D', \subset', \bot' \rangle$ croissante

  ssi $\forall x, y.\ x \subset y \Rightarrow f(x) \subset' f(y)$

- Fonction constante :

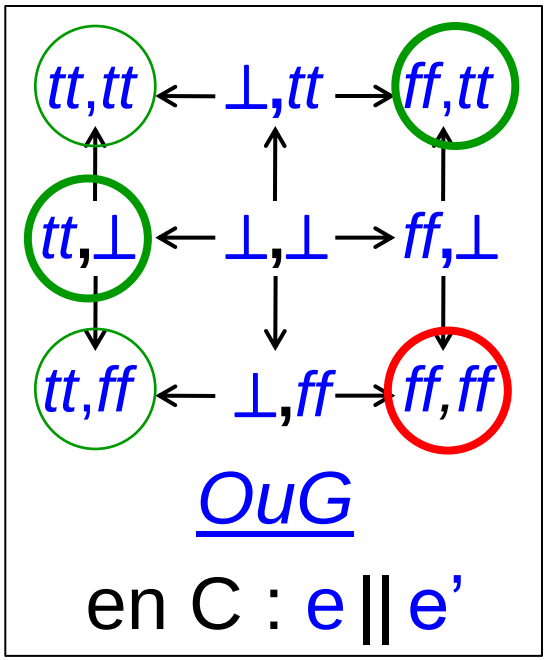
  $\forall x, y.\ f(x) = f(y)$

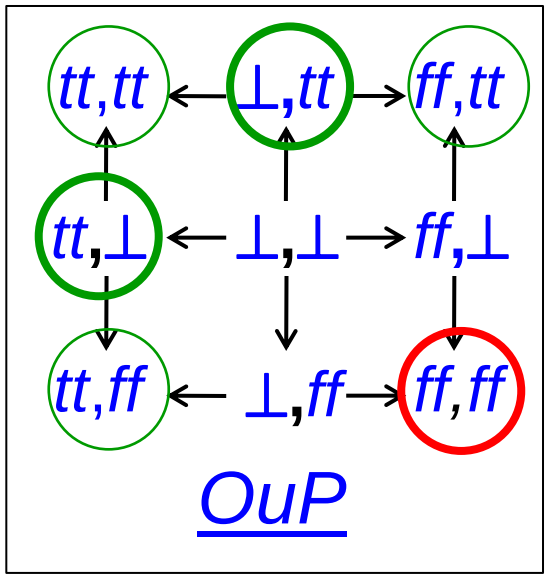
- Attention :

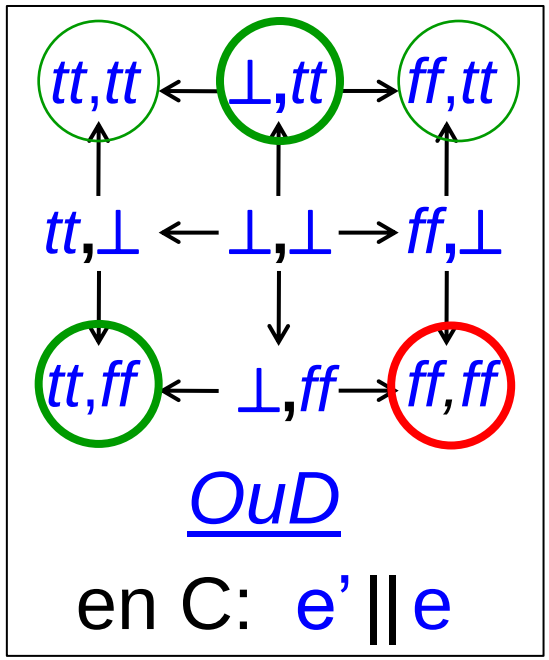
  $\{\bot \rightarrow tt,\ tt \rightarrow tt,\ ff \rightarrow tt\} : \mathbb{B}_\bot \rightarrow \mathbb{B}_\bot$ constante

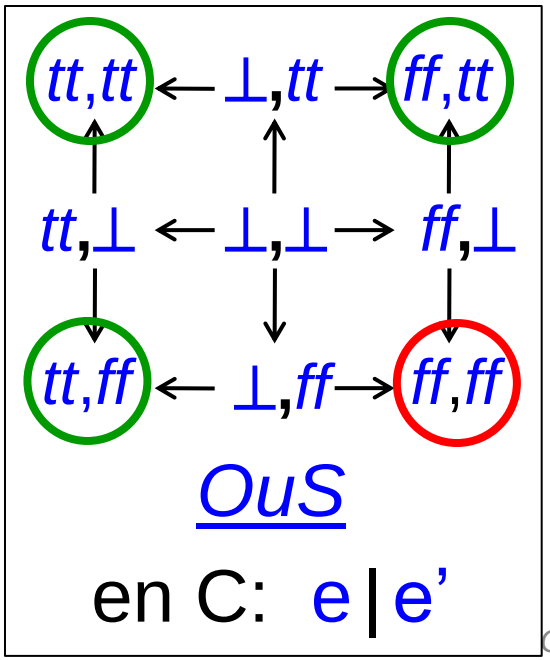
  $\{\bot \rightarrow \bot,\ tt \rightarrow tt,\ ff \rightarrow tt\} : \mathbb{B}_\bot \rightarrow \mathbb{B}_\bot$ stricte, pas constante !

Disjonction booléenne









○ = *tt*

○ = *ff*

Limites de fonctions

- Fact = $Y^{(i\to i)\to(i\to i)\to(i\to i)}$

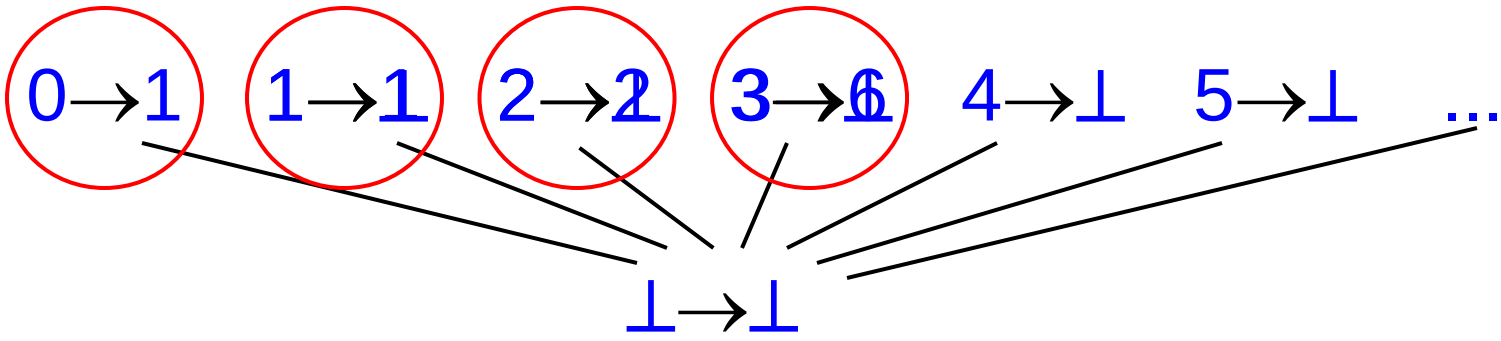
$(\lambda f^{(i\to i)}.\lambda x^{\,i}.\underline{\text{cond}}_i\ (\underline{0?}\ x)\ \underline{1}\ (\text{Mult}\ x\ (f\ (\underline{-1}\ x))))$

$FACT = [[\,\lambda f^{(i\to i)}.\ \lambda x^{\,i}.\ \underline{\text{cond}}_i\ (\underline{0?}\ x)\ \underline{1}\ (\text{Mult}\ x\ (f\ (\underline{-1}\ x)))\,]]\ (\rho)$

$FACT : (N_\perp \to N_\perp)\to(N_\perp \to N_\perp)$

$Fact_0 = FACT(\perp)$ $\qquad$ $Fact_1 = FACT(Fact_0)$

$Fact_2 = FACT(Fact_1)$ $\qquad$ $Fact_3 = FACT(Fact_2)$



CPO = Complete Partial Order

- cpo : $<D,\subset,\bot>$ tel que toute suite croissante x_n a une limite (borne supérieure) $\mathrm{U}_n x_n$

- $<(D \to D'),\subset,\bot>$: fonctions f croissantes et continues, i.e. telles que $f(\mathrm{U}_n x_n) = \mathrm{U}_n f(x_n)$, ordonnées par $f \subset g$ ssi $\forall x.\ f(x) \subset g(x)$

- Théorème : si D et D' sont des cpos, alors $<(D \to D'),\subset,\bot>$ est un cpo, avec $(\mathrm{U}_n f_n)\ (x) = \mathrm{U}_n (f_n(x))$ pour toute suite croissante de fonctions f_n

Point fixe sémantique

Théorème (Knaster-Tarski) : soit $<D,\subset,\bot>$ un cpo,

soit $f : (D \to D)$ continue. Alors $\mathcal{Y}(f) = \mathrm{U}_n f^n(\bot)$ est le plus petit point fixe de f

1. $\bot \subset f(\bot) \Rightarrow f(\bot) \subset f^2(\bot)$

   $\bot \subset f(\bot) \subset f^2(\bot) \subset f^3(\bot) \subset \ldots \subset f^n(\bot) \subset \ldots \subset \mathrm{U}_n f^n(\bot)$

   $f(\mathrm{U}_n f^n(\bot)) = \mathrm{U}_n f(f^n(\bot)) = \mathrm{U}_n f^{n+1}(\bot) = \mathrm{U}_n f^n(\bot)$

   donc $\mathcal{Y}(f) = \mathrm{U}_n f^n(\bot)$ est point fixe de f

2. Soit a point fixe de f

   $\bot \subset a \Rightarrow f(\bot) \subset f(a) = a$

   $\forall n.\ f^n(\bot) \subset f^n(a) = a \Rightarrow \mathcal{Y}(f) = \mathrm{U}_n f^n(\bot) \subset a$

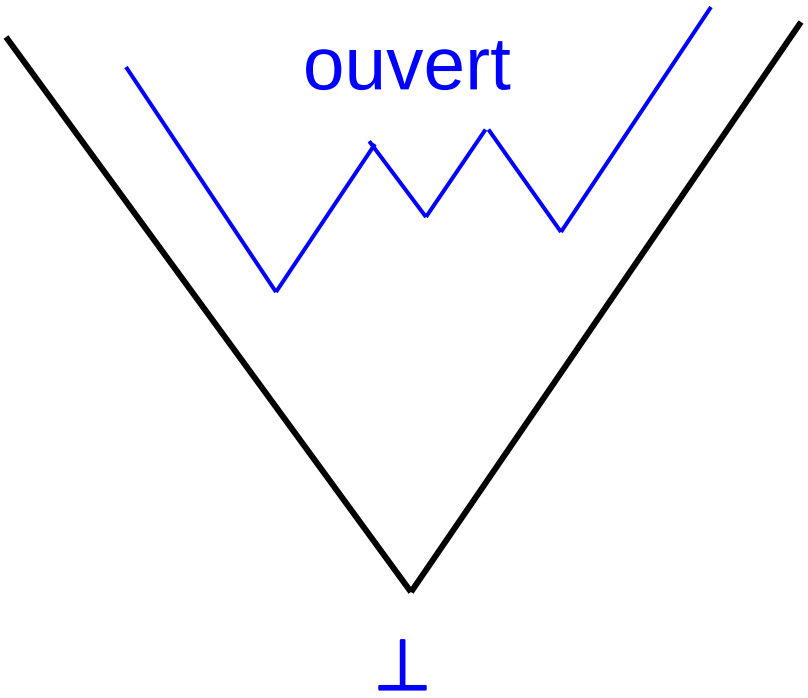
   CQFD

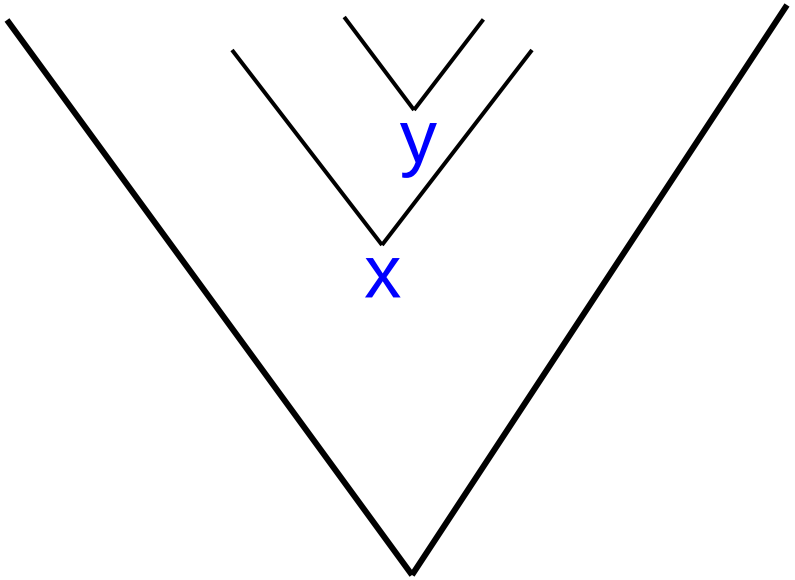
Définition : pour $<D,\subset,\bot>$ un cpo, soit $Y_D : (D \to D) \to D$ la fonctionnelle définie par $Y_D(f) = \bigcup_n f^n(\bot)$.

Théorème : Y_D est croissante et continue

Topologie semi-séparée

- Base d'ouverts : $x \rightarrow \{ y \mid y \supset x \}$





semi-séparé, ou T^0

Interprétation de PCF

$$[\![\ \underline{tt}\ ]\!]\ (\rho) = tt$$

$$[\![\ \underline{ff}\ ]\!]\ (\rho) = ff$$

$$[\![\ \underline{n}\ ]\!]\ (\rho) = n$$

$$[\![\ \underline{+1}\ ]\!]\ (\rho)\ (n) = n+1$$

$$[\![\ \underline{0?}\ ]\!]\ (\rho)\ (n) = tt \text{ si } n=0$$

$$= ff \text{ si } n\neq 0$$

$$[\![\ \underline{cond}_b\ ]\!]\ (\rho)\ (c)\ (b_1)\ (b_2) = b_1 \text{ si } b=tt$$

$$= b_2 \text{ si } b=ff$$

$$[\![\ \underline{cond}_i\ ]\!]\ (\rho)\ (b)\ (n_1)\ (n_2) = n_1 \text{ si } b=tt$$

$$= n_2 \text{ si } b=ff$$

$$[\![\ Y^{(\sigma\to\sigma)\to\sigma}\ ]\!]\ (\rho) = y_{D^\sigma}$$

Equivalence opérationnelle

- Programme : terme de type i clos (sans variable libre)
- Contexte programme $\mathbf{C}[\sigma]$: terme clos de type i avec trou de type σ
- $\mathbf{C}[M^\sigma]$: M^σ mis dans le trou de $\mathbf{C}\ [\sigma]$, doit rester clos
- Exemple : $\mathbf{C}[(i \to i)] = ([(i \to i)]\ \underline{n})$

  $\mathbf{C}[\lambda x^i.\ x^i] = ((\lambda x^i.\ x^i)\ \underline{n})$
- M et N sont opérationnellement équivalents ssi interchangeables dans tout contexte programme

$M^\sigma =_{op} N^\sigma$ ssi $\mathbf{C}[M^\sigma] \to^* \underline{n} \Leftrightarrow \mathbf{C}[N^\sigma] \to^* \underline{n}$

pour tout $\mathbf{C}[\sigma]$ t.q. $\mathbf{C}[M^\sigma]$ et $\mathbf{C}[N^\sigma]$ sont bien typés et clos

Adéquation de la sémantique

- Théorème : la sémantique dénotationnelle respecte l'équivalence opérationnelle

$$[[M]] = [[N]] \Rightarrow M =_{op} N$$

- Corollaire : toute preuve d'égalité sémantique est opérationnellement valide.

- Exemple :

AddG = Y (λm.λn. ... +1 AddG (-1 m, n)...)

AddD = Y (λm.λn. ... +1 AddD (m, -1 n)...)

Par récurrence sur *m* (resp. *n*), [[AddG]](ρ) = [[AddD]](ρ) = +

Donc [[AddG]] = [[AddD]], donc AddG $=_{op}$ AddD

Passage au λ-calcul pur

- Interpréter xx : domaine D tel que $D = (D \to D)$?
  - Impossible avec l'ensemble des fonctions
    diagonale de Cantor : cardinal($D \to D$) > cardinal(D)
  - mais possible en se restreignant aux fonctions continues
    ex. réels : cardinal($R \to_c R$) = cardinal(R)

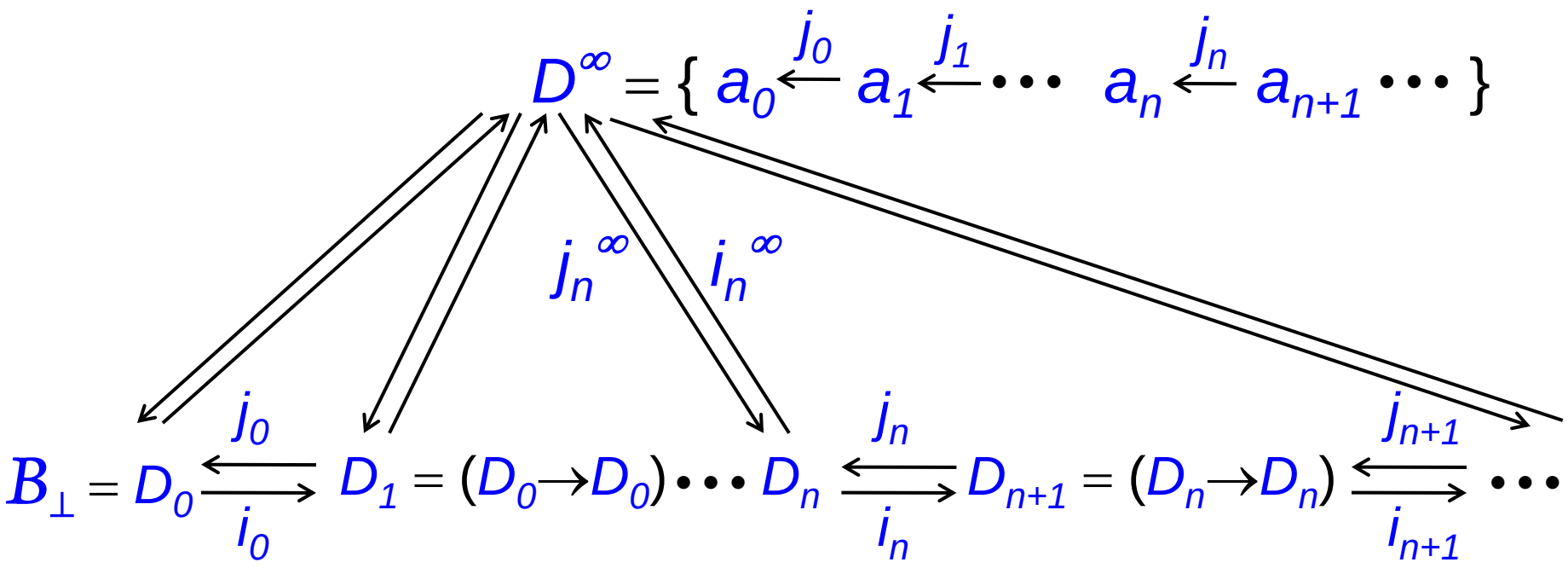
- On utilise la continuité de Scott en construisant
  $<D^\infty, \subset, \bot>$ isomorphe à $<(D^\infty \to D^\infty), \subset, \bot>$

$$D \underset{i}{\overset{j}{\leftrightarrows}} (D \to D)$$

$$i(a)(x) = a$$
$$j(f) = f(\bot)$$

$$j \ \ i(a) = a$$
$$i \ \ j(f) \subset f$$

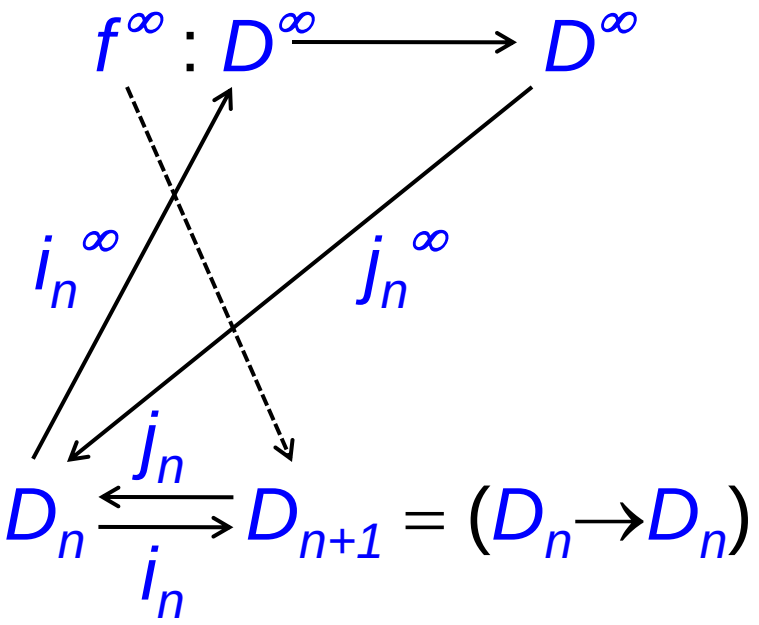
D^∞ comme limite d'une suite de cpos



$$i_n^\infty(a_n) = a_0 \xleftarrow{j_0} a_1 \xleftarrow{j_1} \cdots \; a_n \xleftarrow{j_n} a_n \xleftarrow{j_{n+1}} a_n \cdots$$

$$j_n^\infty (a_0 \xleftarrow{j_0} a_1 \xleftarrow{j_1} \cdots \; a_n \xleftarrow{j_n} a_{n+1} \cdots) = a_n$$

D^∞ comme limite d'une suite de cpos



$$f^\infty_{n+1} = j_n^\infty \quad f^\infty \quad i_n^\infty$$

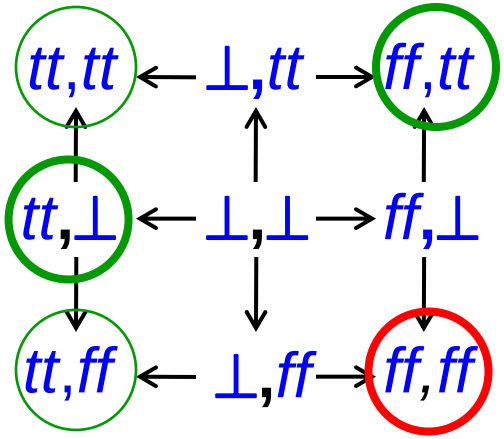
Complète Adéquation

- Question (Plotkin) : existe-t-il un modèle complètement adéquat, i.e., tel que [[M]] = [[N]] ⇔ $M =_{op} N$?

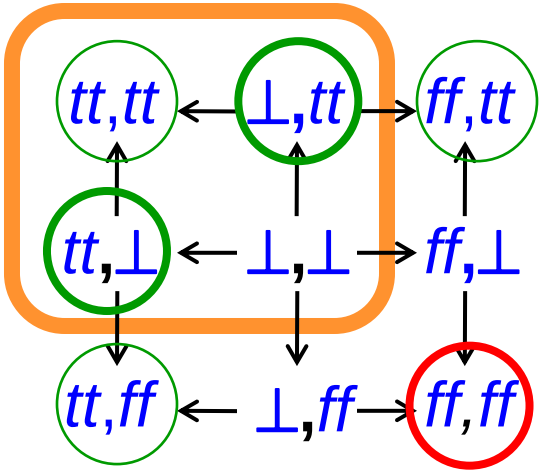
  - Théorème 1 (Milner) : il existe un et un seul modèle complètement adéquat de PCF, caractérisé par le fait que tous ses éléments (finis) sont définissables par des termes

- Théorème 2 (Plotkin / Berry) : le modèle de Scott n'est pas complètement adéquat
- preuve : le ou parallèle *OuP* est continu mais pas définissable (théorème de stabilité)

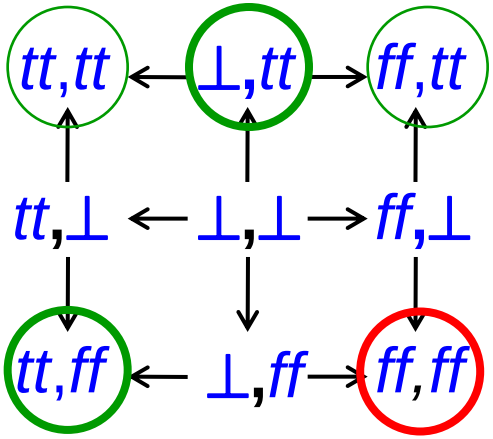
Disjonction booléenne



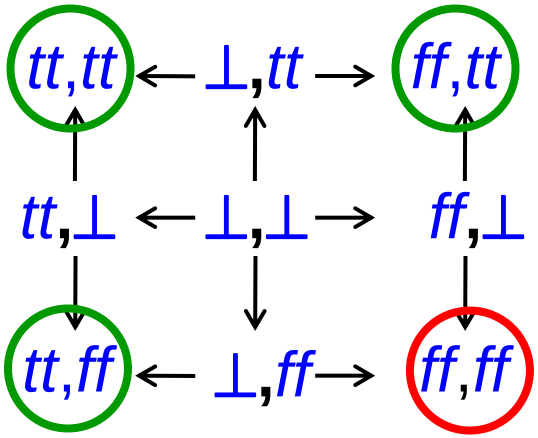
OuG



OuP

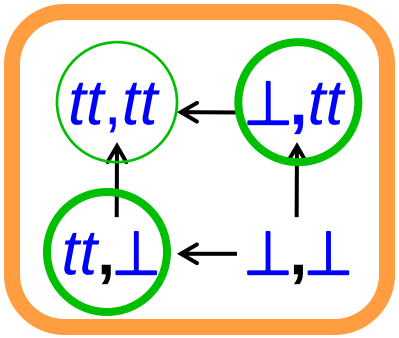


OuD



OuS

Fonctions stables et ordre stable



OuP

~~OuP(tt,Ω) →* tt~~
~~OuP(Ω,tt) →* tt~~
~~OuP(Ω,Ω) →* Ω~~

~~OuP (tt,⊥) = tt~~
~~OuP (⊥,tt) = tt~~
~~OuP (⊥, ⊥) = ⊥~~

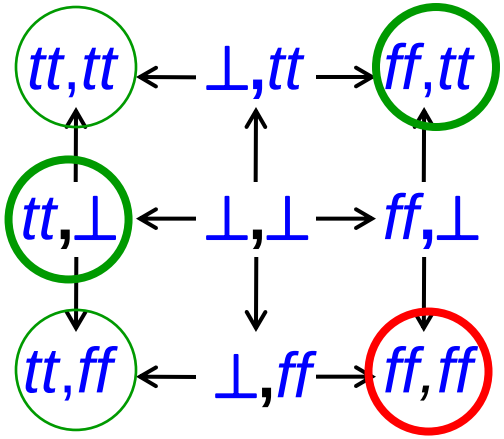
$x \uparrow y$ si $\exists z.\ x \subset z$ et $y \subset z$

Définition : f stable si $\forall x,y.\ x \uparrow y \Rightarrow f(x \wedge y) = f(x) \wedge f(y)$

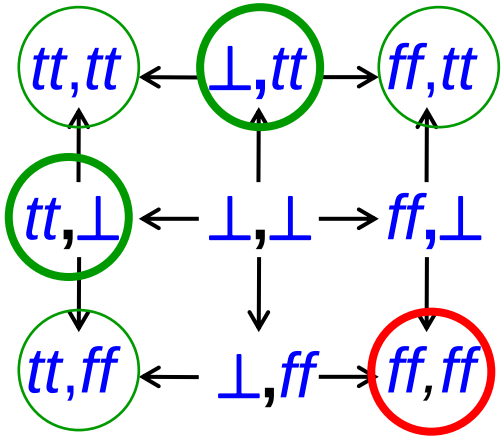
Ordre stable : $f \leq g$ si $f \subset g$
et $\forall x,y.\ x \uparrow y \Rightarrow f(x) \wedge g(y) = f(y) \wedge g(x)$

$f \leq g$: *f et g* ont même causalité

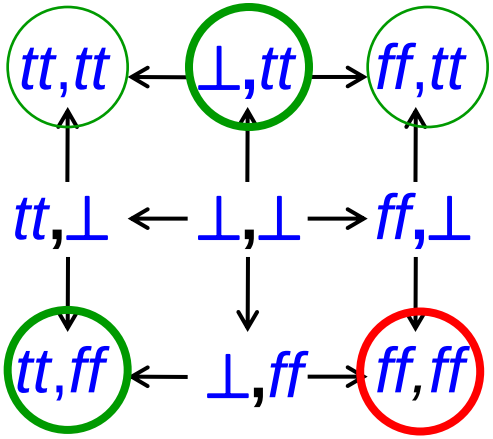
Disjonction booléenne



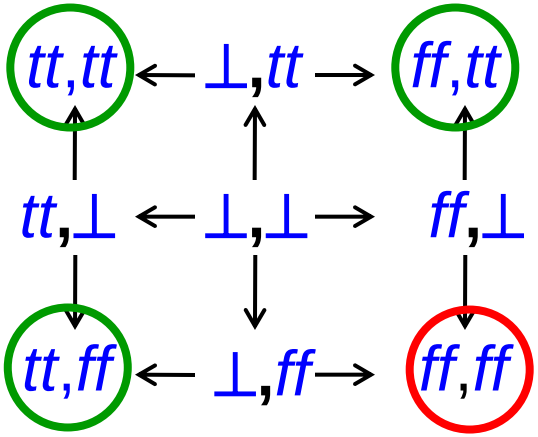
OuG



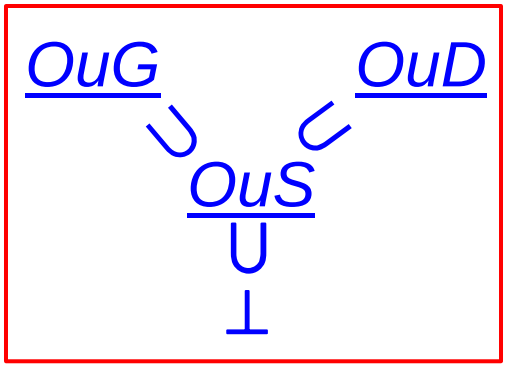
OuP

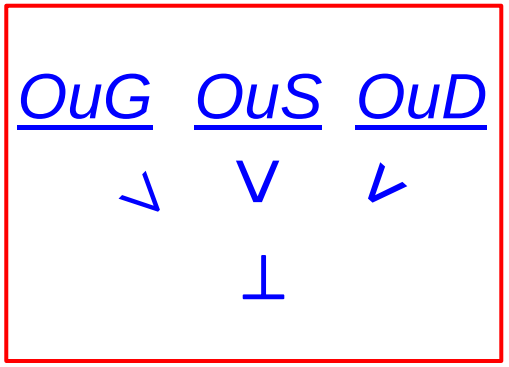


OuD



OuS





Le modèle stable

- Bicpos <D,$\subset$,$\leq$,$\perp$> avec les bons axiomes

  <($D \rightarrow D'$),$\subset$, $\leq$,$\perp$> fonctions stables

  $\subset$ point par point, $\leq$ stable

- J-Y Girard : le modèle naturel de Système F

- Rejette *OuP* mais pas complètement adéquat, car *Gus* est stable mais pas définissable (théorème de séquentialité)

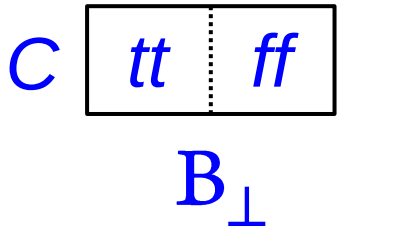
  Gus $\perp$ *tt ff* = *tt*

  Gus ff $\perp$ *tt* = *tt*

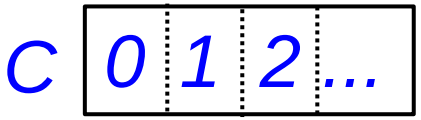
  Gus tt ff $\perp$ = *tt*

Structures de données concrètes

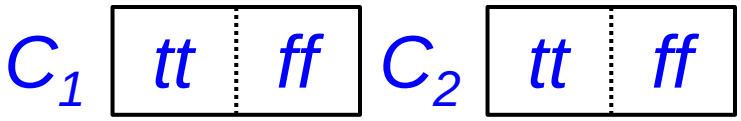
- Kahn-Plotkin : dualités concret / KP-CPOs
- (C, V, E,⊢) : Cellules, Valeurs, Evénements, E⊂C×V



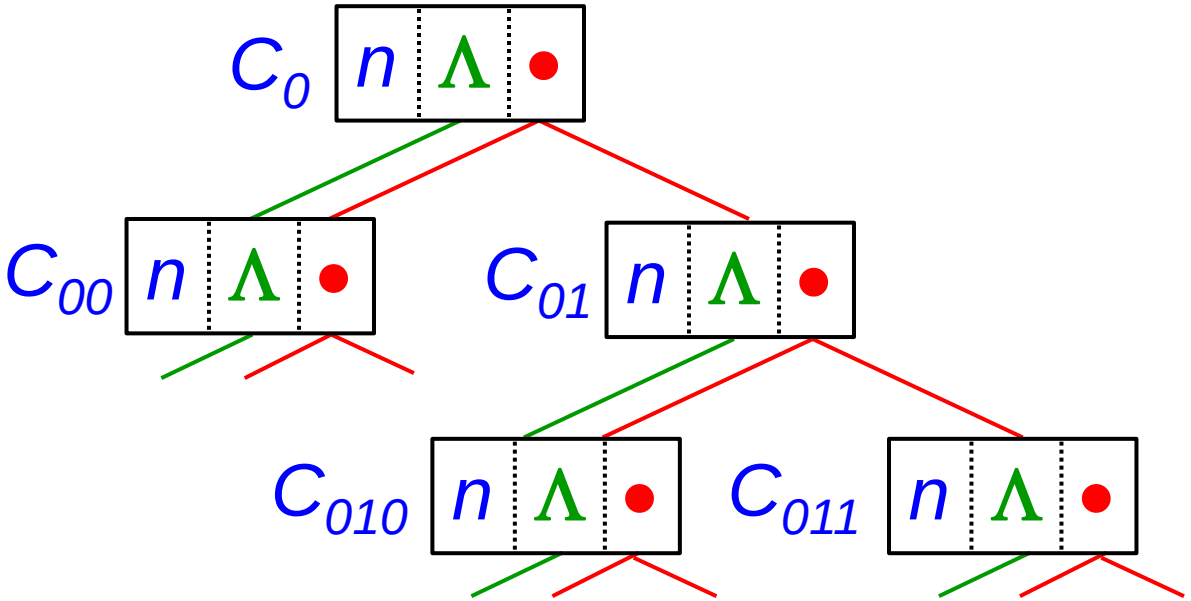
$\mathbb{B}_\perp$



$\mathbb{N}_\perp$



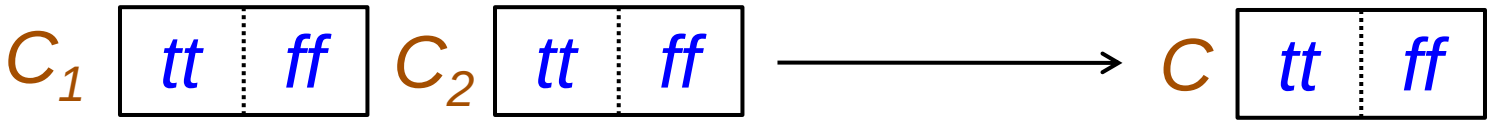
$\mathbb{B}_\perp^{\,2}$



$\Lambda(1 \bullet \Lambda(1 \bullet 2)) = \lambda x.\, x(\lambda y.\, yx)$

$\Lambda(- \bullet \Lambda(- \bullet 2)) = \lambda x.\, \Omega(\lambda y.\, \Omega\ x)$

Algorithmes séquentiels



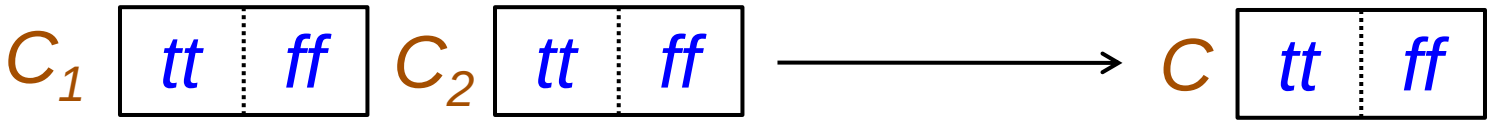
OuS :

```
C ← C1 ?
      tt → C2 ?
            tt → ! tt
            ff → ! tt
      ff → C2 ?
            tt → ! tt
            ff → ! ff
```

OuG :

```
C ← C1 ?
      tt → ! tt
      ff → C2 ?
            tt → ! tt
            ff → ! ff
```

Plusieurs algorithmes par fonction



<u>OuS-1-2</u> :

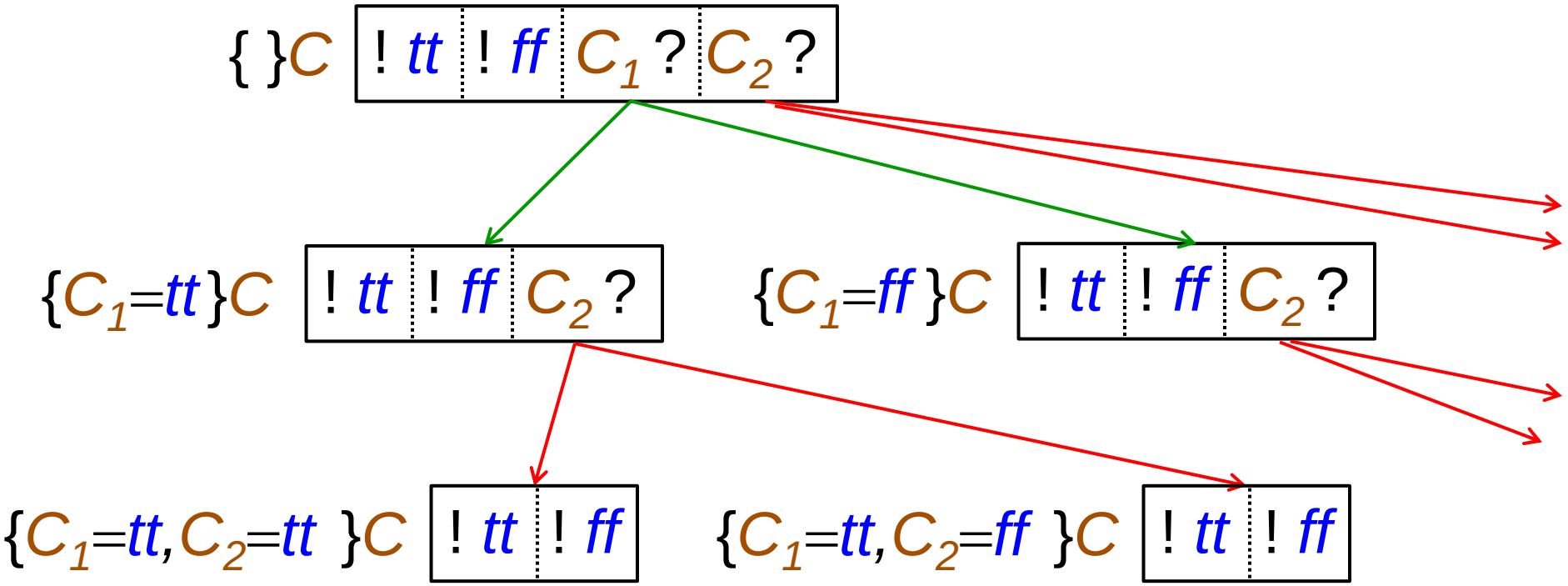
$C \leftarrow C_1$?
- $tt \rightarrow C_2$?
  - $tt \rightarrow$! tt
  - $ff \rightarrow$! tt
- $ff \rightarrow C_2$?
  - $tt \rightarrow$! tt
  - $ff \rightarrow$! ff

$\neq$

<u>OuS-2-1</u> :

$C \leftarrow C_2$?
- $tt \rightarrow C_1$?
  - $tt \rightarrow$! tt
  - $ff \rightarrow$! tt
- $ff \rightarrow C_1$?
  - $tt \rightarrow$! tt
  - $ff \rightarrow$! ff

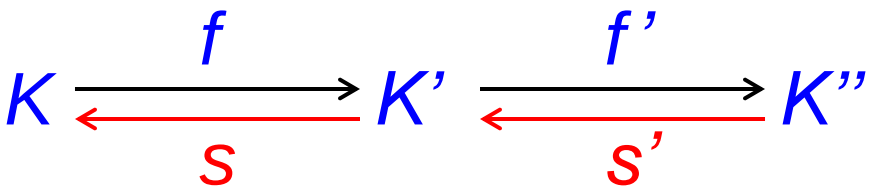
Algorithmes → CDS



- Théorème : les algorithmes entre deux CDS forment une CDS

Catégories et dualité

- Concret : objets = CDS, morphismes = algorithmes
- Abstrait : objets = KP-CPOs , morphismes = stratégies



- Quotient par l'égalité des fonctions => modèle séquentiel de PCF

$<K, \sqsubset, \leq, \subset>$

algorithmes stable fonctions

Complète adéquation pour PCF+

- Tristesse (Curien) : le modèle CDS + algorithmes (quotientés) n'est pas complètement adéquat pour PCF d'exceptions
- Théorème (Curien) : le modèle CDS + algorithmes est complètement adéquat pour PCF augmenté par des opérateurs de levée et de traitement d'exceptions

exception E_1, E_2

$C[\ ] = \text{try}\ [\ b \rightarrow b \rightarrow b\ ]\ (\text{raise}\ E_1)\ (\text{raise}\ E_2)$

with $E_1 \rightarrow <\ \text{gauche}\ >$

| $E_2 \rightarrow <\ \text{droit}\ >$

Conclusion

- La sémantique de Scott a introduit les notions fondamentales d'ordre d'information, de continuité et de point fixe
- Le résultat d'existence et d'unicité de Milner a lancé la chasse aux modèles
- Le modèle stable intègre une notion de causalité
- Le modèle des algorithmes séquentiels traite bien la séquentialité, mais ne résout pas le problème
- Mais peut-être le problème était-il mal posé :

  vive les exceptions !

Références

- *The next 700 Programming Languages*.

  Peter .J. Landin

  Communications of the ACM, vol. 9, n^{o} 3, pp. 157-166 (1966)

- *Domains and Lambda-Calculi*

  Roberto Amadio et Pierre-Louis Curien

  Cambridge University Press (1998)

- *Theory and Practice of Sequential Algorithms: the Kernel of the Programming Language CDS*

  G. Berry et P-L. Curien

  Dans Algebraic Methods in Semantics,
  Cambridge University Press (1985) pp. 35-88.