Jean Goubault-Larrecq

λ -calcul

3. Pouvoir expressif

Fonctions calculables

- ❖ Plusieurs modèles:
 - ❖ Machines de Turing
 - ❖ Machines à compteurs (Marvin Minsky)
 - ❖ Fonctions récursives (Kurt Gödel)
 - ❖ λ -calcul
- ❖ Ils sont tous équivalents (« thèse de Church »)

Fonctions calculables

- ❖ On admettra que machines de Turing $\equiv$ fonctions récurrentives (voir cours de calculabilité)
- ❖ On peut implémenter la β -réduction du λ -calcul sur une machine de Turing (pour les détails, voir partie 3 du cours)
- ❖ On va montrer que toutes les **fonctions récurrentives** (générales) se codent en λ -calcul; d'abord, les fonctions **primitives récurrentives**

Fonctions primitives récuratives, et récuratives (générales)

Les fonctions primitives récursives

- ❖ L'ensemble **PR** des fonctions **primitives récursives** est le plus petit contenant:
 - ❖ les fonctions **constantes** : $\mathbf{N}^k \rightarrow \mathbf{N}$ (pour tout k)
 - ❖ la fonction **successeur** $S : m \in \mathbf{N} \mapsto m+1$
 - ❖ les **projections** $\pi_i^k : (m_1, \dots, m_k) \in \mathbf{N}^k \mapsto m_i$ ($1 \leq i \leq k$)
- ❖ et clos par les opérations de...

Les fonctions primitives récursives

❖ L'ensemble **PR** des fonctions **primitives récursives** est le plus petit contenant les **constantes**, le **successeur**, les **projections** et clos par les opérations de:

❖ **composition**: si $g_1, \dots, g_k : \mathbf{N}^\ell \rightarrow \mathbf{N}$ sont dans **PR**, et $f : \mathbf{N}^k \rightarrow \mathbf{N}$ est dans **PR**, alors

$$f \circ \langle g_1, \dots, g_k \rangle : \underline{m} \in \mathbf{N}^\ell \mapsto f(g_1(\underline{m}), \dots, g_k(\underline{m}))$$

est dans **PR**

❖ **récurrence primitive**: si $f : \mathbf{N}^k \rightarrow \mathbf{N}$ et $g : \mathbf{N}^{k+2} \rightarrow \mathbf{N}$ sont dans **PR**, alors $R_{f,g} : \mathbf{N}^{k+1} \rightarrow \mathbf{N}$ est dans **PR**, où [déf. par réc. sur le 1er arg.]

$$R_{f,g}(0, \underline{m}) \stackrel{\text{def}}{=} f(\underline{m}) \quad \text{[cas de base]}$$

$$R_{f,g}(n+1, \underline{m}) \stackrel{\text{def}}{=} g(n, R_{f,g}(n, \underline{m}), \underline{m}) \quad \text{[cas de récurrence]}$$

Les fonctions récursives générales

- ❖ Toutes les fonctions primitives récursives sont **totales**.
Mais ce n'est pas le cas de toutes les fonctions récursives (calculables).
Il nous manque un équivalent de la boucle `while` générale.
 - ❖ Traditionnellement, on utilise la **minimisation**:
pour toute fonction (partielle) $f: \mathbf{N}^{k+1} \rightarrow \mathbf{N}$,
on note $\mu f: \mathbf{N}^k \rightarrow \mathbf{N}$ la fonction (partielle) telle que:
 - $\mu f(\underline{m}) \stackrel{\text{def}}{=} l'$ unique entier n tel que $f(n, \underline{m})=0$
et pour tout entier $i < n$, $f(i, \underline{m})$ est défini et $\neq 0$
... s'il existe
 - $\mu f(\underline{m})$ indéfini sinon
- $\mu f(\underline{m}) = \ll$ le premier entier n tel que $f(n, \underline{m})=0$ $\gg$
mais pas tout à fait; presque!

Les fonctions récursives générales

- ❖ Toutes les fonctions primitives
Mais ce n'est pas le cas de toutes
(calculables).

Informatiquement:

```
let i = ref 0 in (  
    while f(!i, m) != 0 do i := !i + 1;  
    !i)
```

Il nous manque un équivalent de la minimisation

- ❖ Traditionnellement, on utilise la **minimisation**:
pour toute fonction (partielle) $f: \mathbf{N}^{k+1} \rightarrow \mathbf{N}$,
on note $\mu f: \mathbf{N}^k \rightarrow \mathbf{N}$ la fonction (partielle) telle que:

— $\mu f(\underline{m}) \stackrel{\text{def}}{=} l'$ unique entier n tel que $f(n, \underline{m}) = 0$

et pour tout entier $i < n$, $f(i, \underline{m})$ est défini et $\neq 0$

... s'il existe

— $\mu f(\underline{m})$ indéfini sinon

$\mu f(\underline{m}) = \ll \text{le premier entier } n \text{ tel que } f(n, \underline{m}) = 0 \gg$

à condition que tous les $f(i, \underline{m})$, $i < n$, soient **définis**

Les fonctions récursives générales

- ❖ L'ensemble **Rec** des fonctions **récursives** est le plus petit contenant les **constantes**, le **successeur**, les **projections** et clos par les opérations de:

$(f \circ \langle g_1, \dots, g_k \rangle)(\underline{m})$ défini ssi
côté droit $f(g_1(\underline{m}), \dots, g_k(\underline{m}))$ **défini**

- ❖ **composition**: si $g_1, \dots, g_k : \mathbf{N}^\ell \rightarrow \mathbf{N} \in \mathbf{Rec}$, et $f : \mathbf{N}^k \rightarrow \mathbf{N} \in \mathbf{Rec}$, alors

$$f \circ \langle g_1, \dots, g_k \rangle : \underline{m} \in \mathbf{N}^\ell \rightarrow f(g_1(\underline{m}), \dots, g_k(\underline{m}))$$

est dans **Rec**

- ❖ **récurrence primitive**: si $f : \mathbf{N}^k \rightarrow \mathbf{N}$ et $g : \mathbf{N}^{k+2} \rightarrow \mathbf{N} \in \mathbf{Rec}$, alors

$$R_{f,g} : \mathbf{N}^{k+1} \rightarrow \mathbf{N} \in \mathbf{Rec}, \text{ où}$$

$$R_{f,g}(0, \underline{m}) \stackrel{\text{def}}{=} f(\underline{m})$$

$$R_{f,g}(n+1, \underline{m}) \stackrel{\text{def}}{=} g(n, R_{f,g}(n, \underline{m}), \underline{m})$$

ceci voulant maintenant dire que
le côté gauche est défini
ssi le côté droit est **défini**

Nouveau!

- ❖ **minimisation**: si $f : \mathbf{N}^{k+1} \rightarrow \mathbf{N} \in \mathbf{Rec}$, alors $\mu f : \mathbf{N}^k \rightarrow \mathbf{N} \in \mathbf{Rec}$

Les entiers de Church

- ❖ Avant de commencer, il va falloir trouver une représentation des **entiers**
- ❖ Il y en a plusieurs, mais l'une des plus naturelles est:
$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . \underbrace{f(f(\dots(fx)))}_{n \text{ fois } f} \quad [\text{entier de Church}]$$
- ❖ i.e., $\ulcorner n \urcorner$ est « la fonctionnelle $f \mapsto f^n$ » [itérateur]
- ❖ Je noterai donc, abusivement:
$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But du codage

sans variable libre

- ❖ On cherche, pour toute fonction récursive $f: \mathbf{N}^k \rightarrow \mathbf{N}$, un λ -terme clos $\ulcorner f \urcorner$ tel que, pour tous $n_1, \dots, n_k \in \mathbf{N}$, si $f(n_1, \dots, n_k)$ est définie (et vaut, disons, n), alors

$$\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner =_{\beta} \ulcorner n \urcorner$$

- ❖ **Note:** ceci est équivalent à:

$$\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner \rightarrow^* \ulcorner n \urcorner \quad (\text{pourquoi?})$$

Codage des fonctions de base

❖ Fonctions constantes:

si $f(n_1, \dots, n_k) = n$ la même constante n
pour tous $n_1, \dots, n_k$,

$$\ulcorner f \urcorner \stackrel{\text{def}}{=} \lambda z_1 \dots \lambda z_k . \ulcorner n \urcorner$$

❖ Projections:

$$\ulcorner \pi_i^k \urcorner \stackrel{\text{def}}{=} \lambda z_1 \dots \lambda z_k . z_i$$

❖ Successeur: plusieurs possibilités:

$$\ulcorner S \urcorner \stackrel{\text{def}}{=} \lambda z . \lambda f . \lambda x . f(zfx) \quad \dots \text{ par exemple}$$

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

Le successeur

- ❖ **Successeur:** plusieurs possibilités:

$$\ulcorner S \urcorner \stackrel{\text{def}}{=} \lambda z . \lambda f . \lambda x . f(zfx)$$

- ❖ Vérifions:

$$\begin{aligned} \ulcorner S \urcorner \ulcorner n \urcorner &\rightarrow \lambda f . \lambda x . f(\ulcorner n \urcorner f x) \\ &\rightarrow^2 \lambda f . \lambda x . f(f^n(x)) \\ &= \ulcorner n+1 \urcorner \end{aligned}$$

deux étapes ici,
d'accord?

- ❖ Voyez-vous d'autres définitions possibles de $\ulcorner S \urcorner$?

- ❖ $\lambda z . \lambda f . \lambda x . zf(fx)$, par exemple

au fait, pourquoi ce terme-ci n'est-il pas β -équivalent à $\ulcorner S \urcorner$ ci-dessus?

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

Une parenthèse: opérations arithmétiques

❖ Toutes les opérations arithmétiques sont primitives récursives, mais on peut aussi les définir directement

❖ $\ulcorner + \urcorner \stackrel{\text{def}}{=} \lambda z_1 . \lambda z_2 . \lambda f . \lambda x . z_1 f(z_2 f x)$

... ou $\lambda z_1 . \lambda z_2 . \lambda f . \lambda x . z_2 f(z_1 f x)$

... ou $\lambda z_1 . \lambda z_2 . z_1 \ulcorner S \urcorner z_2$

... ou $\lambda z_1 . z_1 \ulcorner S \urcorner$

[tous β -inéquivalents!]

❖ $\ulcorner \times \urcorner \stackrel{\text{def}}{=} \lambda z_1 . \lambda z_2 . \lambda f . z_1(z_2 f)$

[... par exemple]

❖ $\ulcorner \text{pow} \urcorner \stackrel{\text{def}}{=} \lambda z_1 . \lambda z_2 . z_1 z_2$

[... ou juste $\lambda z . z!$]

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

Composition

composition: si $g_1, \dots, g_k : \mathbf{N}^l \rightarrow \mathbf{N} \in \mathbf{Rec}$,
et $f : \mathbf{N}^k \rightarrow \mathbf{N} \in \mathbf{Rec}$, alors
 $f \circ \langle g_1, \dots, g_k \rangle : \underline{m} \in \mathbf{N}^l \mapsto f(g_1(\underline{m}), \dots, g_k(\underline{m}))$
est dans **Rec**

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

- ❖ $\ulcorner f \circ \langle g_1, \dots, g_k \rangle \urcorner \stackrel{\text{def}}{=} \lambda z_1 . \dots . \lambda z_l .$
 $\ulcorner f \urcorner (\ulcorner g_1 \urcorner z_1 \dots z_l) \dots (\ulcorner g_k \urcorner z_1 \dots z_l)$
- ❖ La récurrence primitive et la minimisation sont beaucoup plus intéressantes... et nécessitent de petits détours

Détour 1: booléens

❖ Booléens de Church:

$\ulcorner \text{vrai} \urcorner \stackrel{\text{def}}{=} \lambda \text{then} . \lambda \text{else} . \text{then}$

$\ulcorner \text{faux} \urcorner \stackrel{\text{def}}{=} \lambda \text{then} . \lambda \text{else} . \text{else}$

❖ $\ulcorner \text{if} \urcorner \stackrel{\text{def}}{=} \lambda \text{test} . \lambda \text{then} . \lambda \text{else} . \text{test then else}$
(ou bien $\lambda \text{test} . \text{test}!$)

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

Détour 2: test à zéro

- ❖ Test d'égalité d'un entier à 0:

$$\ulcorner =0 \urcorner \stackrel{\text{def}}{=} \lambda z . z (\lambda _ . \ulcorner \text{faux} \urcorner) \ulcorner \text{vrai} \urcorner$$

- ❖ $\ulcorner =0 \urcorner \ulcorner 0 \urcorner \rightarrow \ulcorner 0 \urcorner (\lambda _ . \ulcorner \text{faux} \urcorner) \ulcorner \text{vrai} \urcorner$
 $\rightarrow (\lambda x . x) \ulcorner \text{vrai} \urcorner \rightarrow \ulcorner \text{vrai} \urcorner$

- ❖ $\ulcorner =0 \urcorner \ulcorner n+1 \urcorner \rightarrow \ulcorner n+1 \urcorner (\lambda _ . \ulcorner \text{faux} \urcorner) \ulcorner \text{vrai} \urcorner$
 $\rightarrow^2 (\lambda _ . \ulcorner \text{faux} \urcorner)^{n+1} \ulcorner \text{vrai} \urcorner$
 $= (\lambda _ . \ulcorner \text{faux} \urcorner) ((\lambda _ . \ulcorner \text{faux} \urcorner)^n \ulcorner \text{vrai} \urcorner)$
 $\rightarrow \ulcorner \text{faux} \urcorner$

- ❖ Le terme réellement difficile à trouver, c'est le **prédécesseur**

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

Détour 3: couples

❖ Formation de couples:

$$\ulcorner \text{pair} \urcorner \stackrel{\text{def}}{=} \lambda z_1 . \lambda z_2 . \lambda p . p z_1 z_2$$

$$\ulcorner \text{pair} \urcorner uv \rightarrow^2 \langle u, v \rangle \stackrel{\text{def}}{=} \lambda p . p uv$$

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

❖ Projections:

$$\ulcorner p_1 \urcorner \stackrel{\text{def}}{=} \lambda c . c(\lambda z_1 . \lambda z_2 . z_1) \quad (\text{oui, } = \lambda c . c \ulcorner \text{vrai} \urcorner)$$

$$\ulcorner p_2 \urcorner \stackrel{\text{def}}{=} \lambda c . c(\lambda z_1 . \lambda z_2 . z_2) \quad (\text{oui, } = \lambda c . c \ulcorner \text{faux} \urcorner)$$

$$\ulcorner p_1 \urcorner \langle u, v \rangle \rightarrow \langle u, v \rangle (\lambda z_1 . \lambda z_2 . z_1) \rightarrow (\lambda z_1 . \lambda z_2 . z_1) uv \rightarrow^2 u$$

$$\ulcorner p_2 \urcorner \langle u, v \rangle \rightarrow \langle u, v \rangle (\lambda z_1 . \lambda z_2 . z_2) \rightarrow (\lambda z_1 . \lambda z_2 . z_2) uv \rightarrow^2 v$$

Détour 4: prédécesseur

❖ Soit $P(n) \stackrel{\text{def}}{=} n-1$ si $n \geq 1$,

0 si $n=0$ (prédécesseur)

❖ Idée de codage:

$c := \langle 0,0 \rangle$; répéter n fois: remplacer $\langle m,m' \rangle$ par $\langle m+1,m \rangle$

Ainsi c passe par $\langle 0,0 \rangle, \langle 1,0 \rangle, \langle 2,1 \rangle, \dots, \langle n,n-1 \rangle$ (si $n \geq 1$)

Finalelement, prendre la 2ème projection.

❖ $\ulcorner P \urcorner \stackrel{\text{def}}{=} \lambda z . \ulcorner p_2 \urcorner (z(\lambda c . \langle \ulcorner S \urcorner (\ulcorner p_1 \urcorner c), \ulcorner p_1 \urcorner c \rangle) \langle \ulcorner 0 \urcorner, \ulcorner 0 \urcorner \rangle))$

Vérification: exercice!

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

letrec et points fixes

- ❖ On va coder la récurrence primitive $R_{f,g}$ par:

`letrec $R_{f,g}(z, \underline{m}) =$ (informellement)
 if $z=0$ then $f(\underline{m})$
 else $g(P(z), R_{f,g}(P(z), \underline{m}), \underline{m})$`

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

- ❖ On a tout (if, =0, P, composition)... reste `letrec`
- ❖ `letrec  $f(\underline{z}) = e(f, \underline{z})$` , c'est trouver f tel que $f =_{\beta} \lambda \underline{z} . e(f, \underline{z})$
- ❖ autrement dit un **point fixe** de $\lambda f . \lambda \underline{z} . e(f, \underline{z})$ (mod $=_{\beta}$)

Points fixes

un combinateur
(de point fixe, ici)

- ❖ **Proposition.** Tout λ -terme u a un point fixe $Y u \text{ mod } =_{\beta}$ (i.e., $u(Y u) =_{\beta} Y u$).
- ❖ En fait, il existe un λ -terme clos Y tel que, pour tout λ -terme u , $u(Y u) =_{\beta} Y u$.
- ❖ Combinateur de point fixe... **de Church** [surprise!]
$$Y \stackrel{\text{def}}{=} \lambda f . (\lambda x . f(x x)) (\lambda x . f(x x))$$

- ❖ $Y u \rightarrow (\lambda x . u(x x)) (\lambda x . u(x x))$ [x fraîche, $\notin \text{fv}(u)$]
 $\rightarrow u (A_u A_u)$
 $\leftarrow u(Y u)$

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

Points fixes

- ❖ **Proposition.** Tout λ -terme u a un point fixe $Yu \text{ mod } =_{\beta}$ (i.e., $u(Yu) =_{\beta} Yu$).
- ❖ En fait, il existe un λ -terme clos Y tel que, pour tout λ -terme u , $u(Yu) =_{\beta} Yu$.
- ❖ Il y en a d'autres: le combinateur de point fixe **de Turing**
 $\Theta \stackrel{\text{def}}{=} AA$, où $A \stackrel{\text{def}}{=} \lambda g . \lambda h . h(ggh)$
- ❖ $\Theta u = (\lambda g . \lambda h . h(ggh)) A u$
 $\rightarrow^2 u(AAu) = u(\Theta u)$ [le sens de réduction est le bon!]

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

$$\begin{aligned} \text{But: } \ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner \\ =_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner \\ \text{si } f(n_1, \dots, n_k) \text{ défini} \end{aligned}$$

Récurrance primitive

récurrance primitive: si $f : \mathbf{N}^k \rightarrow \mathbf{N}$
et $g : \mathbf{N}^{k+2} \rightarrow \mathbf{N} \in \mathbf{Rec}$,

alors $R_{f,g} : \mathbf{N}^{k+1} \rightarrow \mathbf{N} \in \mathbf{Rec}$, où

$$R_{f,g}(0, \underline{m}) \stackrel{\text{def}}{=} f(\underline{m})$$

$$R_{f,g}(n+1, \underline{m}) \stackrel{\text{def}}{=} g(n, R_{f,g}(n, \underline{m}), \underline{m})$$

$$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

$$\diamond \ulcorner R_{f,g} \urcorner \stackrel{\text{def}}{=} Y(\lambda R . \lambda z . \lambda z_1 \dots \lambda z_k .$$

$$\ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner z)$$

$$(\ulcorner f \urcorner z_1 \dots z_k)$$

$$(\ulcorner g \urcorner (\ulcorner P \urcorner z) (R (\ulcorner P \urcorner z) z_1 \dots z_k) z_1 \dots z_k))$$

$$\diamond \ulcorner R_{f,g} \urcorner \ulcorner 0 \urcorner u_1 \dots u_k =_{\beta} \ulcorner f \urcorner u_1 \dots u_k$$

$$\ulcorner R_{f,g} \urcorner \ulcorner n+1 \urcorner u_1 \dots u_k =_{\beta} \ulcorner g \urcorner \ulcorner n \urcorner (\ulcorner R_{f,g} \urcorner \ulcorner n \urcorner u_1 \dots u_k) u_1 \dots u_k$$

(exercice... ou voir transparents suivants)

Le récursueur

$$\begin{aligned} \ulcorner R_{f,g} \urcorner &\stackrel{\text{def}}{=} Y(\lambda R . \lambda z . \lambda z_1 . \dots . \lambda z_k . \\ &\quad \ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner z) \\ &\quad (\ulcorner f \urcorner z_1 \dots z_k) \\ &\quad (\ulcorner g \urcorner (\ulcorner P \urcorner z) (R (\ulcorner P \urcorner z) z_1 \dots z_k) z_1 \dots z_k)) \end{aligned}$$

❖ **Fait.** $\ulcorner R_{f,g} \urcorner \ulcorner 0 \urcorner u_1 \dots u_k =_{\beta} \ulcorner f \urcorner u_1 \dots u_k$

❖ $\ulcorner R_{f,g} \urcorner \ulcorner 0 \urcorner u_1 \dots u_k =_{\beta} (\lambda R . \dots) \ulcorner R_{f,g} \urcorner \ulcorner 0 \urcorner u_1 \dots u_k$

(déf. de Y)

❖ $\rightarrow^* \ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner \ulcorner 0 \urcorner)$
 $(\ulcorner f \urcorner u_1 \dots u_k)$

Or $\ulcorner =0 \urcorner \ulcorner 0 \urcorner \rightarrow^* \ulcorner \text{vrai} \urcorner$
 et $\ulcorner \text{if} \urcorner \ulcorner \text{vrai} \urcorner T E \rightarrow^* T$

$(\ulcorner g \urcorner (\ulcorner P \urcorner \ulcorner 0 \urcorner) (\ulcorner R_{f,g} \urcorner (\ulcorner P \urcorner \ulcorner 0 \urcorner) u_1 \dots u_k) u_1 \dots u_k))$

❖ $\rightarrow^* (\ulcorner f \urcorner u_1 \dots u_k)$

Le récursueur

$$\begin{aligned} \ulcorner R_{f,g} \urcorner &\stackrel{\text{def}}{=} Y(\lambda R . \lambda z . \lambda z_1 . \dots . \lambda z_k . \\ &\quad \ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner z) \\ &\quad (\ulcorner f \urcorner z_1 \dots z_k) \\ &\quad (\ulcorner g \urcorner (\ulcorner P \urcorner z) (R (\ulcorner P \urcorner z) z_1 \dots z_k) z_1 \dots z_k)) \end{aligned}$$

❖ **Fait.** $\ulcorner R_{f,g} \urcorner \ulcorner n+1 \urcorner u_1 \dots u_k =_{\beta} \ulcorner g \urcorner \ulcorner n \urcorner (\ulcorner R_{f,g} \urcorner \ulcorner n \urcorner u_1 \dots u_k) u_1 \dots u_k$

❖ $\ulcorner R_{f,g} \urcorner \ulcorner n+1 \urcorner u_1 \dots u_k =_{\beta} (\lambda R . \dots) \ulcorner R_{f,g} \urcorner \ulcorner n+1 \urcorner u_1 \dots u_k$ (déf. de Y)

❖ $\rightarrow^* \ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner \ulcorner n+1 \urcorner) (\ulcorner f \urcorner u_1 \dots u_k) (\ulcorner g \urcorner (\ulcorner P \urcorner \ulcorner n+1 \urcorner) (\ulcorner R_{f,g} \urcorner (\ulcorner P \urcorner \ulcorner n+1 \urcorner) u_1 \dots u_k) u_1 \dots u_k))$

Or $\ulcorner =0 \urcorner \ulcorner n+1 \urcorner \rightarrow^* \ulcorner \text{faux} \urcorner$
et $\ulcorner \text{if} \urcorner \ulcorner \text{faux} \urcorner T E \rightarrow^* E$

Et $\ulcorner P \urcorner \ulcorner n+1 \urcorner \rightarrow^* \ulcorner n \urcorner$

❖ $\rightarrow^* (\ulcorner g \urcorner \ulcorner n \urcorner (\ulcorner R_{f,g} \urcorner \ulcorner n \urcorner u_1 \dots u_k) u_1 \dots u_k))$

Minimisation

minimisation: si $f: \mathbf{N}^{k+1} \rightarrow \mathbf{N} \in \mathbf{Rec}$,
alors $\mu f: \mathbf{N}^k \rightarrow \mathbf{N} \in \mathbf{Rec}$

❖ $\ulcorner \mu f \urcorner \stackrel{\text{def}}{=} \lambda z_1 \dots \lambda z_k .$
 $Y(\lambda search . \lambda z .$

$\ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner (\ulcorner f \urcorner z z_1 \dots z_k))$

z

$search (\ulcorner S \urcorner z))$

$\ulcorner 0 \urcorner$

❖ Si $\ulcorner f \urcorner \ulcorner 0 \urcorner u_1 \dots u_k =_{\beta} \ulcorner m_0 \urcorner, \dots, \ulcorner f \urcorner \ulcorner n \urcorner u_1 \dots u_k =_{\beta} \ulcorner m_n \urcorner,$
avec $m_0, \dots, m_{n-1}$ non nuls et $m_n=0$, alors

$\ulcorner \mu f \urcorner u_1 \dots u_k =_{\beta} \ulcorner n \urcorner$ (exercice... ou transparents suivants)

« let $\mu f(\underline{m}) =$
letrec search $z =$
if $f(z, \underline{m})=0$ then z
else search $(z+1)$
in search 0 »

$\ulcorner n \urcorner \stackrel{\text{def}}{=} \lambda f . \lambda x . f^n(x)$

But: $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$
 $=_{\beta} \ulcorner f(n_1, \dots, n_k) \urcorner$
si $f(n_1, \dots, n_k)$ défini

La minimisation

$$\begin{aligned} \ulcorner \mu f \urcorner &\stackrel{\text{def}}{=} \lambda z_1 \dots \lambda z_k . \\ &Y(\lambda \text{search} . \lambda z . \\ &\quad \ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner (\ulcorner f \urcorner z z_1 \dots z_k)) \\ &\quad z \\ &\quad \text{search} (\ulcorner S \urcorner z)) \\ &\ulcorner 0 \urcorner \end{aligned}$$

- ❖ **Prop.** Si $\ulcorner f \urcorner \ulcorner 0 \urcorner u_1 \dots u_k =_{\beta} \ulcorner m_0 \urcorner, \dots, \ulcorner f \urcorner \ulcorner n \urcorner u_1 \dots u_k =_{\beta} \ulcorner m_n \urcorner,$
avec $m_0, \dots, m_{n-1}$ non nuls et $m_n=0$, alors

$$\ulcorner \mu f \urcorner u_1 \dots u_k =_{\beta} \ulcorner n \urcorner.$$

- ❖ Posons $F \stackrel{\text{def}}{=} Y(\lambda \text{search} . \lambda z .$
 $\ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner (\ulcorner f \urcorner z u_1 \dots u_k))$
 z
 $\text{search} (\ulcorner S \urcorner z))$

$$\begin{aligned} F \ulcorner i \urcorner &=_{\beta} (\lambda \text{search} . \lambda z \dots) F \ulcorner i \urcorner \\ &\rightarrow^* \ulcorner \text{if} \urcorner (\ulcorner =0 \urcorner (\ulcorner f \urcorner \ulcorner i \urcorner u_1 \dots u_k)) \\ &\quad \ulcorner i \urcorner \\ &\quad F (\ulcorner S \urcorner \ulcorner i \urcorner)) \\ &=_{\beta} \ulcorner i \urcorner \text{ si } \ulcorner f \urcorner \ulcorner i \urcorner u_1 \dots u_k =_{\beta} \ulcorner 0 \urcorner, \\ &\quad F \ulcorner i+1 \urcorner \text{ sinon} \end{aligned}$$

- ❖ Donc $\ulcorner \mu f \urcorner u_1 \dots u_k \rightarrow^* F \ulcorner 0 \urcorner$
 $=_{\beta} F \ulcorner 1 \urcorner$
 $=_{\beta} \dots$
 $=_{\beta} F \ulcorner n \urcorner =_{\beta} \ulcorner n \urcorner.$

Pouvoir expressif

- ❖ Au total, nous avons:
- ❖ **Théorème.** Pour toute fonction récursive $f: \mathbf{N}^k \rightarrow \mathbf{N}$,
il existe un λ -terme clos $\ulcorner f \urcorner$ tel que,
pour tous $n_1, \dots, n_k \in \mathbf{N}$,
si $f(n_1, \dots, n_k)$ est définie (et vaut, disons, n), alors
$$\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner =_{\beta} \ulcorner n \urcorner$$
- ❖ Kleene a montré encore plus fort:
on peut demander que si $f(n_1, \dots, n_k)$ n'est pas **définie**,
alors $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$ n'est **pas normalisable**
- ❖ C'est beaucoup plus dur, et utilise le **théorème de standardisation**
(prochaine fois), ou bien de démontrer qu'en fait $\ulcorner f \urcorner \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$ n'a
même pas de **forme normale de tête** (prochaine fois)
... sans parler du fait que notre composition est en CBN, mais celle de Kleene est en CBV!

La prochaine fois

La prochaine fois

- ❖ Stratégies de réduction
- ❖ Rédexes de tête, formes normales de tête
- ❖ Réductions de tête, gauche, standard
- ❖ Théorème de standardisation