λ - Calcolo

Per noi, λ-calcolo = calcolo fondazionale per i linguaggi visti fin'ora.

- Sintassi minimale
- Esprimibilità dei comportamenti di interesse

Il λ-calcolo è un calcolo su termini (term calculus)

- insieme di termini
- sistema equazionale su termini

In realtà, tanti λ-calcoli

- $=_\beta$
- $=_\alpha$
- $=_\eta$
- $=_{\beta\eta}$
- ⋮

① λ-calcolo come teoria equazionale

② λ calcolo come calcolo fondazionale

Def Insieme Λ dei λ-termini:

$$t, s ::= x \mid \lambda x.t \mid t\,s$$

variabili appartenenti a un insieme fissato X

```
type lam_exp =
    | Var of ident
    | Abs of ident * lam_exp
    | App of lam_exp * lam_exp
```

convenzioni:

$$t_1 t_2 \dots t_n \equiv ((t_1 t_2) \cdots t_n)$$

$$\lambda x_1 \dots x_n . t \equiv \lambda x_1 . (\lambda x_2 . \dots (\lambda x_n . t) \dots)$$

Es.

$$I \triangleq \lambda x.x$$

$$K \triangleq \lambda xy.x$$

$$\Omega \triangleq (\lambda x.xx)(\lambda x.xx)$$

Def. La **relazione** $\alpha \subseteq \Lambda \times \Lambda$ è definita così:

$$\lambda x. t \;\alpha\; \lambda y. \underbrace{t[y/x]}_{\text{dobbiamo definire la sostituzione}} \quad \text{con} \quad y \notin FV(t)$$

Def. La **sostituzione** è **la** mappa $subst : \Lambda \times \Lambda \times X \to \Lambda$, notazione $t[s/x]$ per $subst(t, s, x)$, definita da:

$$y[s/x] \triangleq \begin{cases} s & \text{se } x = y \\ y & \text{altrimenti} \end{cases}$$

$$(t_1 t_2)[s/x] \triangleq t_1[s/x]\; t_2[s/x]$$

$$(\lambda y. M)[s/x] \triangleq \begin{cases} \lambda y. M & \text{se } x = y \\ \lambda y. M[s/x] & \text{se } y \notin FV(s) \\ \lambda z. M[z/y][s/x] & \text{se } y \in FV(s),\ \text{con } z \notin FV(M) \cup FV(s) \end{cases}$$

NB. La sostituzione è capture-avoiding

Def. La relazione $\alpha \subseteq \Lambda \times \Lambda$ è definita così:

$$\lambda x. t \ \alpha \ \lambda y. t[y/x] \quad \text{con} \quad y \notin FV(t)$$

Più precisamente:

$$t \ \alpha \ s \ \stackrel{\Delta}{\Leftrightarrow} \ \exists t', x, y . \begin{cases} t = \lambda x. t' \\ s = \lambda y. t'[y/x] \\ y \notin FV(t) \end{cases}$$

La relazione di α-equivalenza si ottiene estendendo l'azione di α così:

$$\frac{}{x =_\alpha x} \qquad \frac{t \ \alpha \ s}{t =_\alpha s} \qquad \frac{t =_\alpha s}{\lambda x. t =_\alpha \lambda x. s} \qquad \frac{t_1 =_\alpha s_1 \quad t_2 =_\alpha s_2}{t_1 t_2 =_\alpha s_1 s_2}$$

$$\frac{t =_\alpha s}{s =_\alpha t} \qquad \frac{t =_\alpha s \quad s =_\alpha u}{t =_\alpha u}$$

Possiamo definire in modo "rigoroso" il principio di invarianza rispetto alla scelta di variabili legate (parametri formali).

[Ogni nozione su λ-termini che definiamo deve rispettare la α-equivalenza

<u>Teorema 1</u> La sostituzione rispetta $=_\alpha$:

- $t =_\alpha s \implies t[u/x] =_\alpha s[u/x]$

- $t =_\alpha s \;\&\; u =_\alpha v \implies t[u/x] =_\alpha s[v/x]$

<u>Applicazione</u>. Se codifichiamo un linguaggio basato su substitution-model dentro Λ, $\ulcorner \cdot \urcorner : \mathcal{L} \longrightarrow \Lambda$ in modo tale che <u>tutti</u> i binder siano riducibili a λ-astrazioni, allora possiamo esprimere il principio di invarianza come invarianza rispetto a $=_\alpha$

Il teorema 1 ci dice che il modello di calcolo è coerente con questo principio.

Es. Definiamo $\mathcal{L}$ come segue:

$$e ::= n \mid b \mid e + e \mid \text{if } e \text{ then } e \text{ else } e$$
$$\mid \text{let } x = e \text{ in } e$$
$$\mid e\, e \mid \text{fun } x \to e$$

Possiamo iniziare a definire una codifica di $\mathcal{L}$ in Λ:

$$\ulcorner n \urcorner \triangleq ?$$
$$\vdots$$
$$\ulcorner \text{let } x = e \text{ in } e' \urcorner = (\lambda x. \ulcorner e' \urcorner) \ulcorner e \urcorner$$
$$\ulcorner e\, e' \urcorner = \ulcorner e \urcorner \ulcorner e' \urcorner$$
$$\ulcorner \text{fun } x \to e \urcorner = \lambda x. \ulcorner e \urcorner$$

Abbiamo che $\ulcorner \text{let } x = e \text{ in } e' \urcorner =_\alpha \ulcorner \text{let } y = e \text{ in } e'[y/x] \urcorner$

$$\ulcorner \text{subst}(e, e', x) \urcorner = \ulcorner e \urcorner [\ulcorner e' \urcorner / x]$$

Def. La <u>relazione</u> $\beta \subseteq \Lambda \times \Lambda$ è definita così:

$$(\lambda x.\, t)\, s \;\;\beta\;\; t[s/x]$$

La relazioni di β-equivalenza e β-riduzione sono definite così:

- $$\frac{t \;\beta\; s}{t \to_\beta s} \qquad \frac{t \to_\beta s}{\lambda x.t \to_\beta \lambda x.s} \qquad \frac{t \to_\beta t'}{t\,s \to_\beta t'\,s} \qquad \frac{s \to_\beta s'}{t\,s \to_\beta t\,s'}$$

- $$\frac{}{t \to_\beta^* t} \qquad \frac{t \to_\beta s \quad s \to_\beta^* u}{t \to_\beta^* u}$$

- $$\frac{t \to_\beta^* s}{t =_\beta s} \qquad \frac{t =_\beta s}{s =_\beta t}$$

Teorema $=_\beta \; = (\to_\beta \cup \leftarrow_\beta)^*$

↳ uguaglianza per "computazione"

Teorema $\to_\beta$ (e quindi $\to_\beta^*$ e $=_\beta$) rispetta $=_\alpha$

$$t \to_\beta s \;\&\; t =_\alpha t' \implies \exists s'.\; s =_\alpha s' \;\&\; t' \to_\beta s'$$

Def. Un termine $t \in \Lambda$ è in forma normale se

$$\neg \exists t'.\ t \to_\beta t' \qquad (t \not\to_\beta)$$

Es.
- I è in forma normale
- $\lambda y. I$ è una forma normale di $K I$
- Ω non ha forma normale

Domande
- Perchè le forme normali sono interessanti?
- Quante forme normali ha un termine?
- Due termini sono β-equivalenti esattamente quando hanno una stessa forma normale?

① Le forme normali sono interessanti perché rappresentano i risultati del calcolo.

Ci si chiede quindi se:

- Un calcolo ha sempre un risultato? No! $\Omega \to_\beta \Omega \to_\beta \cdots$
- Il risultato di un calcolo è necessariamente unico?
- Qual è il rapporto tra $=_\beta$ e "avere lo stesso risultato"

$$\underbrace{t =_{KE} s \overset{\Delta}{\iff} \exists u . \begin{cases} u \not\to_\beta \\ t \to^*_\beta u \\ s \to^*_\beta u \end{cases}}_{\text{Uguaglianza di Kleene}}$$

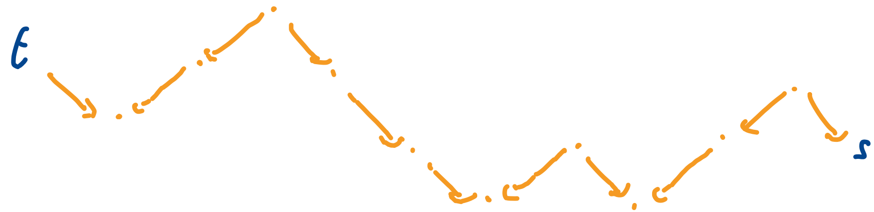
<u>Teorema</u> (Church-Rosser). La relazione $\to_\beta$ è confluente

$$\begin{tikzcd} & t \arrow[dl, "*"'] \arrow[dr, "*"] & \\ t_1 \arrow[dr, dashed, "*"'] & & t_2 \arrow[dl, dashed, "*"] \\ & s & \end{tikzcd}$$

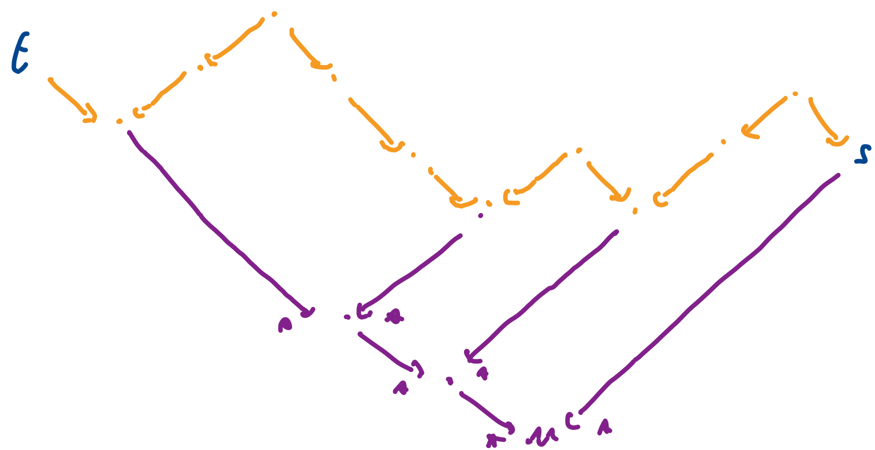
$$t \to^*_\beta t_1 \ \& \ t \to^*_\beta t_2 \implies \exists s .\ t_2 \to^*_\beta s \ \& \ t_1 \to^*_\beta s$$

Corollario. $t =_\beta s$ sse $\exists u.\ t \to_\beta^* u$ & $s \to_\beta^* u$

Dim. Se $t =_\beta s$, allora $t \ ({}_\beta\!\leftarrow \cup \to_\beta)^* \ s$. quindi:



Applichiamo la confluenza:



Corollario. Se un termine ha una forma normale, questa è necessariamente unica

Osservazioni:

1. Non tutti i termini sono β-equivalenti:

$$K \neq_\beta I$$

2. $=_{\mathcal{M}} \;\triangleq\; =_\beta$

3. $t =_\beta s$ sse t ed s riducono ad un termine comune

$$\Omega \triangleq (\lambda x. xx)(\lambda x\, xx)$$

$$\Omega \rightarrow_\beta \Omega$$

$$\omega_3 \triangleq (\lambda x. xxx)$$

$$\omega_3 \omega_3 \rightarrow_\beta \omega_3 \omega_3 \omega_3 \rightarrow_\beta \cdots$$

Quindi: $\Omega \neq_\beta \omega_3 \omega_3$

② Forme normali dal punto di vista di: $\mathcal{L}$.

$\mathcal{L}$:
- sintassi: $\longleftrightarrow$ λ-termini
- programmi: $\longleftrightarrow$ λ-termini chiusi
- valori: $\longleftrightarrow$ forme normali chiuse
- dinamica $\longleftrightarrow$ $\to_\beta^*$

(teoria equazionale su programmi data da $=_\beta$)

"<u>Teorema</u>". $\forall e, v \in \mathcal{L}$.

$$e \Rightarrow v \quad \text{implica} \quad \ulcorner e \urcorner \to_\beta^* \ulcorner v \urcorner$$

<u>Osservazioni</u>. Abbiamo $\ulcorner v \urcorner \not\to_\beta$?

Abbiamo il viceversa?

Codifica di $\mathcal{L}$ in Λ

Completiamo la codifica di Λ in Λ.

① Come rappresentiamo i numeri? $\leadsto$ Numerali di Church

$$\ulcorner 0 \urcorner \triangleq \lambda f. \lambda x. \, x$$
$$\ulcorner 1 \urcorner \triangleq \lambda f. \lambda x. \, f x$$
$$\ulcorner 2 \urcorner \triangleq \lambda f. \lambda x. \, f (f x)$$
$$\vdots$$
$$\ulcorner n \urcorner \triangleq \lambda f. \lambda x. \, f^{(n)} x$$

$$\begin{cases} f^{(0)} x = x \\ f^{(n+1)} x = f \left(f^{(n)} x \right) \end{cases}$$

Es. Se definiamo

$$succ \triangleq \lambda n. \lambda f. \lambda x. \, f (n \, x)$$

allora

$$succ \ulcorner n \urcorner =_\beta \ulcorner n+1 \urcorner$$