1. Cos'è un algoritmo ?
2. MdT
3. Programmi FOR (terminano), WHILE (non sempre)
4. Funzioni ricorsive primitive (convergono)

Esempio: $f_1 = \lambda x.\ x$

$f_2 = \lambda x.\ x+1$

$f_3 = \lambda x_1, x_2, x_3.\ x_2$

$f_4 = f_2(f_3(x_1, x_2, x_3))$

$$\begin{cases} f_5(0, x_2) = f_1(x_2) \\ f_5(n+1, x_2) = f_4(n, f_5(n, x_2), x_2) \end{cases}$$

Calcolo $f_5(2,3)$:

$f_5(2,3)$ Questa si chiama REDEX, viene ridotta applicando le regole

$= f_4(1, f_5(1,3), 3) =$

$= f_4(1, f_4(0, f_5(0,3), 3), 3) =$

$= f_4(1, f_4(0, f_2(3), 3), 3) = \dots$

La scelta del redex prende il nome di regola di valutazione.

$f_7(x_1, x_2) = x_2$

$f_8(x_1, x_2) = x_1$

$$\begin{cases} pred(0) = 0 \\ pred(x+1) = f_8(x, pred(x)) \end{cases}$$

$$f_9(x,y,z) = pred(f_3(x,y,z))$$

$$\begin{cases} f_{10}(0,y) = y \\ f_{10}(x+1,y) = f_9(x, f_{10}(x,y), y) \end{cases}$$

MENO LIMITATO

$$x \dot{-} y = f_{10}(f_7(x,y), f_8(x,y))$$

Somma
$$\begin{cases} 0+y = y \\ (x+1)+y = (x+y)+1 \end{cases}$$

↓ generalizzo

Prodotto
$$\begin{cases} 0 \cdot y = 0 \\ (x+1) \cdot y = (x \cdot y) + y \end{cases}$$

↓

Potenza
$$\begin{cases} x^0 = 1 \\ x^{y+1} = x^y \cdot x \end{cases}$$

L'operazione di somma itera l'operatore successore x volte fino a ricondurmi al caso base

Definisco un predicato P ricorsivo primitivo se la sua funzione caratteristica è ricorsiva primitiva.

Funzione caratteristica $\chi_P(\vec{x}) = \begin{cases} 1 & \text{se } \vec{x} \in P \\ 0 & \text{altr} \end{cases}$

Definisco Primo(x) il predicato che dice se x è primo;

Primo(x) è ricorsivo primitivo.

TEOREMA dell'UNICA FATTORIZZAZIONE Siano $p_0 < \dots < p_n$ primi,

$\forall n \in \mathbb{N}.\ \exists!$ numero finito di esponenti

$x_i \neq 0$ t.c. $n = p_0^{x_0} \cdot \dots \cdot p_k^{x_k}$

$\exists! f: n \xrightarrow{c} \{ x_i \mid x_i \text{ fattori di } n \}$ e sono ricorsive primitive

Si può quindi codificare una MdT: si usa un metodo detto di Gödelizzazione:

$M = (Q, \Sigma, \delta, q_0)$ $\quad \{L, R, -\}$

$Q = \{q_0, \dots, q_n\}$ $\quad \delta(q_i, \sigma_j) = (q'_k, \sigma'_l, \vec{D})$

$\Sigma = \{\sigma_0, \dots, \sigma_m\}$

Codifica: $q_2 = \underline{p_0^{i+1} \cdot p_1^{j+1} \cdot p_2^{h+1} \cdot p_3^{l+1} \cdot p_4^{m_D}}$

Definisco: $\begin{cases} h \to n+1 \\ L \to m_L = m+2 \\ R \to m_R = m+3 \\ - \to m_- = m+4 \end{cases}$

- Si ordinano le quintuple lessicograficamente
- Si calcola la produttoria

$p_0^{g_0} \cdot \dots \cdot p_r^{g_r} = N_M$

Posso codificare stringhe $\forall w \in \Sigma^*.\ w \rightleftarrows n_w$

configurazioni $\forall \gamma = (q, \omega).\ \gamma \rightleftarrows n_\gamma$

sequente $(\gamma_0 \to \dots \to \gamma_n) \rightleftarrows n_{comp}$ (numero della computazione)

FUNZIONE di ACKERMANN

$A(0,0,y) = y$ caso base della somma

$A(0,x+1,y) = A(0, x, y) + y$

$A(1,0,y) = 0$ caso base del prodotto

$A(z+2,0,y) = 1$ potenza

$A(z+1, x+1, y) = A(z, A(z+1, x, y), y)$

Ricorre più volte della ricorsione primitiva

Termina sempre, è totale,

NON è ricorsiva primitiva

Esempio: $A(0,x,y) = x+y$

$A(1,x,y) = x \cdot y$

$A(2,x,y) = y^x$

$A(3,x,y) = \underbrace{y^{y^{y}}}_{x \text{ volte}}$