

TESI DI CHURCH-TURING → VALIDO SOLO PER I MODELLI ATTUALI

TUTTO CIO' CHE E' CALCOLABILE LO E' MEDIANTE UNA MACCHINA DI TURING

PASCAL MINIMO

MODELLO DI CALCOLO BASATO SU : → SIMULA GLI ODIERNI LINGUAGGI
DI PROGRAMMAZIONE

i) ASSEGNAZIONE:

$a \leftarrow b$

ii) SELEZIONE:

IF (COND) THEN ... ELSE

iii) ITERAZIONE:

WHILE (COND) DO

iv) STRUTTURE DATI:

ARRAY → $A[i]$

TEOREMA RELAZIONE FRA PASCAL MINIMO E MACCHINA DI TURING

$\forall P$ PROGRAMMA SCRITTO IN PASCAL MINIMO

\exists UN T TRASDUTTORE CHE:

\forall INPUT RESTITUISCE LO STESSO OUTPUT DI P

DIM

SCRIVIAMO UN PROGRAMMA P CON INPUT $n, m, A[]$

$i = 1$

WHILE ($i \leq h$) DO BEGIN:

IF ($A[i] > i$) THEN $m \leftarrow A[i]$;

$i \leftarrow i + 1$;

END;

RETURN m

ELIMINIAMO GLI ARRAY DALLE CONDIZIONI DEI CICLI

$i = 1$

WHILE ($i \leq h$) DO BEGIN:

$AUSIL \leftarrow A[i]$;

IF ($AUSIL > i$) THEN $m \leftarrow A[i]$;

$i \leftarrow i + 1$;

END;

RETURN m

LASCIAMO UN'ISTRUZIONE PER RIGA

È LO NUMERIAMO

1) $i = 1$

2) WHILE ($i \leq h$) DO BEGIN:

3) $AUSIL \leftarrow A[i]$;

4) IF ($AUSIL > i$) THEN

5) $m \leftarrow A[i]$;

6) $i \leftarrow i + 1$; END;

7) RETURN m

TRADUCIAMO ORA IL PROGRAMMA P NELLA MACCHINA T

i) NASTRI:

i.1 A VARIABILI ASSOCIAMO UN NASTRO

i.2 A ARRAY ASSOCIAMO 2 NASTRI:

UNO CON GLI ELEMENTI → **PUR SEMPLIFICARE OGNI CELLA HA UNA PAROLA**

UNO CON L'INDICE ESAMINATO

$A[3] = M$

SEPARATORE

□	M	A	M	\$	P	A	A
---	---	---	---	----	---	---	---

INDICE IN UNARIO

□	1	1	1			IND A
---	---	---	---	--	--	-------

ii) STATI:

A RIGA CORRISPONDE UNO STATO

1) $i=1 \rightarrow q_1$

2) WHILE ($i \leq h$) DO BEGIN: $\rightarrow q_2$

3) AUSIL $\leftarrow A[i]; \rightarrow q_3$

...

...

iii) SELEZIONE:

1) IF ($a > b$) THEN BEGIN $\rightarrow < q_j, (\text{sc} N_a > N_b), q_{j+1}, F >$

2) $b \leftarrow a; \rightarrow < q_{j+1}, (\text{COPIA } N_a \text{ SU } N_b), q_{j+2}, F >$

3) $a \leftarrow 0; \text{END}; \rightarrow < q_{j+2}, (\text{SCRIVI } 0 \text{ SU } N_a), q_{j+3}, F >$

4) $\rightarrow < q_j, (\text{sc} N_b > N_a), q_{j+3}, F >$



DESCRIZIONE AD ALTO LIVELLO DI UN INSIEME

DI QUINTUPLE

IV ASSIGNAZIONE:

$\overbrace{J}^{\overbrace{J+1}^{\vdots}} a \leftarrow b; \rightarrow < q_J, (\text{COPIA IL CONTENUTO DI } N_a \text{ SU } N_b), q_{J+1}, F >$

V ITERAZIONE:

$\overbrace{J}^{\overbrace{J+1}^{\vdots}}$	WHILE ($a < b$) THEN BEGIN :	$\rightarrow < q_J, (\text{SE } N_a > N_b), q_{J+1}, F >$
$\overbrace{J+1}^{\vdots}$	$b \leftarrow a;$	$\rightarrow < q_{J+1}, (\text{COPIA } N_b \text{ SU } N_a), q_{J+2}, F >$
$\overbrace{J+2}^{\vdots}$	$a \leftarrow 0; (\text{END});$	$\rightarrow < q_{J+2}, (\text{SCRIVI } 0 \text{ SU } N_a), q_J, F >$
$\overbrace{J+3}^{\vdots}$	\vdots	$\rightarrow < q_J, (\text{SE } N_a > N_b), q_{J+3}, F >$

TORNO INDietro

VI ARRAY:

$\overbrace{J}^{\overbrace{J+1}^{\vdots}} A[i] \leftarrow b;$

α	β	γ	δ	β	η	\square	N_A
1	1	1	1	\square			N_{INDA}

$< q_J, (\text{SCRIVI } N_i \text{ SU } N_{INDA} \text{ IN UNARIO}), q'_J, F >$

$< q'_J, (\text{SCORRI } N_i \in N_{INDA} \text{ VERSO Dx FINO A } \square, \text{ POI TORNO A Sx}), q''_J, F >$

$< q''_J, (\text{COPIA } N_b \text{ SU } N_a), q_{J+1}, F >$

SIAMO RIUSCITI A SCRIVERE OGNI OPERAZIONE DEL PASCAL MINIMO COME QUINTUPLE DI UNA T



TEOREMA RELAZIONE FRA MACCHINA DI TURING E PASCAL MINIMO

$\forall T_U$ MACCHINA UNIVERSALE DI TURING

\exists UN \mathcal{P}_U PROGRAMMA IN PASCAL MINIMO CHE:

\forall INPUT RESTITUISCE LO STESSO OUTPUT DI T_U

DIM

DEFINIAMO INNANZITUTTO LA MACCHINA T_U COME:

$T_U = \langle \Sigma, Q, P, q_0, \{q_A, q_R\} \rangle$ AD 1 NASTRO

DOVE P È DEL TIPO:

$$P = \left\{ \langle q_{11}, S_{11}, S_{12}, q_{12}, m_1 \rangle, \right. \\ \left\langle q_{21}, S_{21}, S_{22}, q_{22}, m_2 \rangle, \right. \\ \left\langle q_{31}, S_{31}, S_{32}, q_{32}, m_3 \rangle, \right. \\ \vdots \\ \left. \langle q_{k1}, S_{k1}, S_{k2}, q_{k2}, m_k \rangle \right\}$$

ABBIANO DIMOSTRATO CHE CON UN NASTRO
POSSIAMO FARQ QUALUNQUE COSA

AVENDO P STRUTTURATO IN QUESTO MODO POSSIAMO GESTIRE GLI ARRAY DI \mathcal{P}_U IN MODO CHE:

ARRAY:

$$Q_1[1:k] \rightarrow Q_1[3] = q_{13}$$

$$S_1[1:k]$$

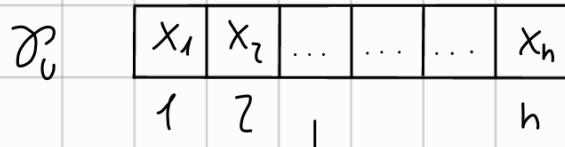
$$S_2[1:k]$$

$$Q_2[1:k]$$

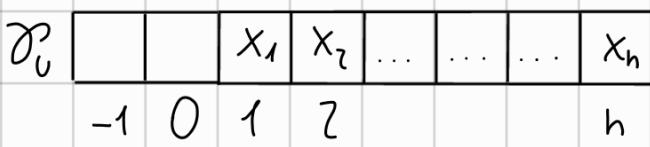
$$M[1:k] \rightarrow M[3] = -1, 0, 1 \Leftrightarrow m_3 = Sx, F, Dx$$

$$N[1:n] \rightarrow \text{PAROLA IN INPUT}$$

DOBBIAMO GESTIRE LA RAPPRESENTAZIONE DEGLI ARRAY



Poiché i nastri sono infiniti e gli array no abbiamo bisogno di renderli dinamici



Implementiamo gli indici negativi per simulare il nastro infinito

SCRIVIAMO ORA L'ALGORITMO DI SIMULAZIONE D_U

INPUT D_U : $q_0, q_A, q_R, Q[1:k], S_1[1:l_1], S_2[1:l_2], Q_1[1:l_1], M[1:h], N[1:h]$

$q \leftarrow q_0$; → STATO INIZIALE

$p_c \leftarrow 1; u_c \leftarrow h$; → p_c = PRIMA CASILLA, u_c = ULTIMA CASILLA

$t \leftarrow 1;$

$i \leftarrow 1;$

WHILE ($q \neq q_A \wedge q \neq q_R$) DO BEGIN

$\bar{j} \leftarrow 1;$

TROVATA ← FALSE;

WHILE ($\bar{j} \leq k \wedge \text{TROVATA} = \text{FALSE}$) DO BEGIN

IF ($q = Q_1[\bar{j}] \wedge N[\bar{j}] = S_1[\bar{j}]$) THEN

TROVATA ← TRUE; END;

$\bar{j} \leftarrow \bar{j} + 1;$

END;

IF ($\text{TROVATA} = \text{FALSE}$) THEN

IF ($q \neq q_A$) THEN $q \leftarrow q_R$; END;

END;

ELSE (% TROVATA = TRUE) BEGIN

$N[t] \leftarrow S[j-1];$

$q \leftarrow Q_2[j-1];$

$t \leftarrow t + M[j-1];$

IF ($t > U_c$) THEN

$U_c \leftarrow U_c + 1;$

$N[t] = \square; END; \rightarrow$ INIZIALIZZO IL CARATTERE \square

ELIF ($t < P_c$) THEN

$P_c \leftarrow P_c - 1;$

$N[t] = \square; END; \rightarrow$ INIZIALIZZO IL CARATTERE \square

END;

END;

RETURN q;

PROGRAMMI PER MACCHINE NON DETERMINISTICHE \rightarrow NON RICHIESTO PER L'CSAM

LUNGHEZZA DELLA COMPUTAZIONE:

Q SIMULA RICORSIVO $(Q, INT, \Sigma, N, INIT)$, \Rightarrow TUTTI I CASI VILL TRAVERSO N

INPUT: $q_0, q_A, q_R, Q_1[1..k], S_1[1..k], S_2[1..k], Q_2[1..k], M[1..k], N[1..n]$

IF $(i=0)$ THEN

$p \leftarrow q_i$

ELSE BEGIN

$j \leftarrow 1;$

RIGUITTO \leftarrow VERO

WHILE $(j \leq k \wedge q \neq q_A)$ DO BEGIN

WHILE $(j \leq k \wedge [q, Q_1[j]] \wedge N[G] \neq S_1[j])$ DO

$j \leftarrow j+1$; END;

IF $(j = k+1)$ THEN $p \leftarrow q_R$;

ELSE $(q = Q_1[j] \wedge N[G] = S_1[j])$ BEGIN

$N[G] \leftarrow S_2[j];$

IF $(G + M[j] > V_C)$ THEN BEGIN $V_C \leftarrow V_C + 1$; $N[G] \leftarrow \square$; END;

CLIF $(G + M[j] < P_C)$ THEN BEGIN $P_C \leftarrow P_C - 1$; $N[G] = \square$; END.

$p \leftarrow$ SIMULA RICORSIVO $(Q_2[j], G + M[j], N, i - 1)$;

\hookrightarrow POSSONO MODIFICARE SOLO QUI ALTRIMENTI

NON VERIFICO TUTTI LG QUINTUPLO

IF $(p = q_A)$ THEN $q = q_A$;

ELSE IF $(p \neq q_R)$ THEN RIGUITTO \leftarrow FALSO;

$N[G] \leftarrow S_1[j]$

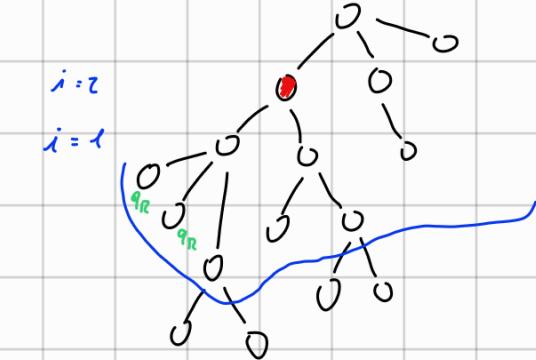
$j \leftarrow j + 1$

END;

END;

IF $(p \neq q_A \wedge RIGUITTO = \text{FALSO})$ THEN $p \leftarrow q_0$

RETURN p ;



VA BENE VAGLI, BASTA CHE p SIA DIVERSO DA q_A E q_R