

Macchina a Stato (Minsky 1956, Wang 1957)

- **Sintassi** (ovviamente, astratta)

$\mathcal{Z} = \text{Zero } R \mid \text{Inc } R \mid \text{DJ0 } R \ N \mid \text{Halt}$

- **Termini:** \mathcal{Z} Insieme (numerabile) degli Statements, z, z_i .
 - **Azzeramento** registro R ;
 - **Incremento** registro R ;
 - **Decremento o salto** a N se R contiene 0;
 - **Arresto** esecuzione;
- **Registri:** R_i . Insieme finito di Registri a *capacità infinita*.
- **N:** interi nonnegativi
- **Programmi, \mathcal{P} :**
 - Ogni sequenza finita di \mathcal{Z} ;
 - Ogni statement accessibile attraverso la propria posizione nella sequenza

- **Esempi**

Zero R_0 ; Halt

Zero R_0 ; DJ0 R_1 4; Inc R_2 ; DJ0 R_0 1; Halt

$\mathcal{Z} = \text{Zero R} \mid \text{Inc R} \mid \text{DJ0 RN} \mid \text{Halt}$

- Semantica SOS
 - Definizione di **stato**
 - Registri (variabili con nomi predefiniti) rappresentati da una sequenza di coppie, ρ , della forma: $(R_0, n_0) \dots (R_k, n_k)$
 - Program Counter: Posizione PC del successivo statement da eseguire (inizialmente 0).
 - Uno stato è rappresentato da una coppia $\{\text{PC}, \rho\}$
 - Definizione di **Transizione** \rightarrow
 - Esecuzione di Statement z nello stato σ della macchina
 - $\langle z, \sigma \rangle \rightarrow \sigma'$, indicante ...
 - $\langle z, \sigma \rangle \rightarrow \langle z', \sigma' \rangle$, indicante ...

$\mathcal{Z} = \text{Zero R} \mid \text{Inc R} \mid \text{DJ0 R N} \mid \text{Halt}$

- Semantica SOS

- Definizione di **stato**
- Definizione di **Transizione**

- Definizione di **Computazione** di un programma di $p \in \mathcal{P}$:

- sequenza componente registro degli stati attraversati nell'esecuzione di p .

$$\langle z, \{n, \rho\} \rangle \rightarrow \langle z_1, \{n_1, \rho_1\} \rangle \rightarrow \langle z_2, \{n_2, \rho_2\} \rangle \rightarrow \dots$$

- La semantica di un programma è la sua computazione:

$$\text{Sem}(p) \equiv \rho, \rho_1, \rho_2, \dots,$$

- **Notazione.**

- $\{n, (R_1, n_1), \dots, (R_k, n_k)\}$ stato con PC $\equiv n$ e k registri
- ρ, τ (anche con pedici) sono componente registro dello stato
- $\rho(R_i) = 0$ quando $\rho = (R_1, n_1), \dots, (R_i, 0), \dots, (R_k, n_k)$,
- $\rho(R_i) \neq 0$ quando $\rho = (R_1, n_1), \dots, (R_i, n_i \neq 0), \dots, (R_k, n_k)$,
- $\rho[R_i \leftarrow 0] = (R_1, n_1), \dots, (R_i, 0), \dots, (R_k, n_k)$
- $\rho[R_i \leftarrow R_i + 1] = (R_1, n_1), \dots, (R_i, n_i + 1), \dots, (R_k, n_k)$
quando $\rho = (R_1, n_1), \dots, (R_i, n_i), \dots, (R_k, n_k)$
- $P[n]$ n -esimo statement del programma P
- $\#P$ numero di statements del programma (posizionati da 0 a $\#P - 1$)

$\mathcal{Z} = \text{Zero R} \mid \text{Inc R} \mid \text{DJO R N} \mid \text{Halt}$

$\langle \text{Halt}, \{n, \rho\} \rangle \rightarrow \{n, \rho\}$

$$\frac{0 \leq n < \#P, \quad z_n = P[n]}{\langle \text{Zero R}_i, \{n, \rho\} \rangle \rightarrow \langle z_n, \{n+1, \rho[R_i \leftarrow 0]\} \rangle}$$

$$\frac{0 \leq n < \#P, \quad z_n = P[n]}{\langle \text{Inc R}_i, \{n, \rho\} \rangle \rightarrow \langle z_n, \{n+1, \rho[R_i \leftarrow R_i + 1]\} \rangle}$$

$$\frac{0 \leq m < \#P, \quad z_m = P[m], \quad \rho(R_i) = 0}{\langle \text{DJO R}_i m, \{n, \rho\} \rangle \rightarrow \langle z_m, \{m+1, \rho\} \rangle} \quad \frac{0 \leq n < \#P, \quad z_n = P[n], \quad \rho(R_i) \neq 0}{\langle \text{DJO R}_i m, \{n, \rho\} \rangle \rightarrow \langle z_n, \{n+1, \rho[R_i \leftarrow R_i - 1]\} \rangle}$$

- 1 Si mostri la computazione di $\langle P, \{0, \rho_0\} \rangle$, dove $\rho_0 \equiv \{0, (R_0, 5), (R_1, 3), (R_2, 4)\}$
 $P \equiv \text{Zero } R_0; \text{DJ0 } R_1 \ 4; \text{Inc } R_2; \text{DJ0 } R_0 \ 1; \text{Hal.t.}$. Si assuma il primo statement di P alla posizione 0.
- 2 Si mostri la computazione del programma P sopra, nello stato $\{0, \rho\}$, dove ρ_0 sia un arbitraria configurazione dei registri R_0, R_1, R_2 .
- 3 Si scriva un programma che calcolato in uno stato iniziale arbitrario ρ_0 termini avendo copiato nel registro R_1 il contenuto del registro R_2 .
- 4 Lo stesso esercizio precedente salvo che nello stato finale ρ_k deve valere:
$$\rho_k(R_1) = \rho_0(R_1) = \rho_k(R_2).$$
- 5 Si scriva un programma che calcolato in uno stato iniziale ρ_0 termini scrivendo 0 nel registro R_0 se $\rho(R_1) \neq \rho(R_2)$. Il programma termina scrivendo 1, in caso contrario.
- 6 Si mostri la computazione del programma P sotto, nello stato $\{0, \rho_0\}$, dove ρ_0 sia un arbitraria configurazione dei registri R_0, R_1, R_2 .
$$P \equiv \text{Zero } R_0; \text{Inc } R_1; \text{DJ0 } R_0 \ 1; \text{Hal.t.}$$
- 7 Si scriva un programma che calcolato in uno stato iniziale ρ_0 termini in uno stato ρ_k tale che $\rho_k(R_2) = \rho_0(R_2) - \rho(R_1)$.
- 8 Cosa implica l'assunzione che i registri della Macchina a Stato sebbene in numero finito abbiano capacità illimitata?

Altri esercizi in: EserciziL5