TEORIA DELLA COMPUTABILITÀ

Prof. Domenico Cantone

A.A. 2010/11

PROGRAMMA

I PARTE: TEORIA DELLA CALCOLABILITÀ

FUNZIONI CALCOLABILI

- ALGORITMI, PROCEDURE EFFETTIVE
- IL MODELLO URM (UNLIMITED REGISTER MACHINE)
- FUNZIONI URM-CALCOLABILI
- PREDICATI E PROBLEMI DECIDIBILI
- CALCOLABILITÀ SU ALTRI DOMINI

GENERAZIONE DI FUNZIONI CALCOLABILI

- FUNZIONI CALCOLABILI DI BASE
- UNIONE DI PROGRAMMI
- SOSTITUZIONE
- RICORSIONE
- MINIMALIZZAZIONE

TESI DI CHURCH

- ALTRI APPROCCI ALLA CALCOLABILITÀ
- FUNZIONI PARZIALMENTE RICORSIVE
- FUNZIONI PRIMITIVE RICORSIVE
- MACCHINE DI TURING
- (SISTEMI DI POST E MARKOV)
- TESI DI CHURCH-TURING

ENUMERAZIONE DELLE FUNZIONI CALCOLABILI E I PROGRAMMI UNIVERSALI

- IL METODO DIAGONALE
- IL TEOREMA S-m-n
- FUNZIONI E PROGRAMMI UNIVERSALI
- DUE APPLICAZIONI DEL PROGRAMMA UNIVERSALE
- PROBLEMI INDECIDIBILI
- TEOREMA DI RICORSIONE

LIBRO DI TESTO:

N.J. CUTLAND: "COMPUTABILITY", CAMBRIDGE UNIVERSITY PRESS, 1986.

II PARTE

- Il modello computazionale delle Macchine di Turing
- Macchine di Turing deterministiche e nondeterministiche
- Funzioni Turing-computabili
- Equivalenza tra macchine di Turing deterministiche e nondeterministiche
- Altre varianti delle Macchine di Turing: Macchine di Turing multinastro
- Time complexity delle Macchine di Turing
- Classi di complessità di tempo
- Relazioni di complessità tra i vari modelli di Macchine di Turing
- Macchine di Turing polinomialmente limitate nel tempo
- Le classi P ed NP dei linguaggi decisi in tempo deterministico e nondeterministico polinomiale
- Riduzioni polinomiali
- Il problema P versus NP
- Il concetto di NP-completezza
- Problemi e linguaggi NP-completi e Teorema di Cook-Levin
- Catalogo di problemi NP-completi
- Space complexity delle Macchine di Turing
- Classi di complessità di spazio
- Teorema di Savitch
- Le classi PSPACE ed NPSPACE dei linguaggi decisi in spazio deterministico e nondeterministico polinomiale
- Completezza PSPACE

TEORIA DELLA CALCOLABILITÀ

CENNI STORICI:

NASCE CON I LAVORI DI ALAN TURING E ALONZO CHURCH CHE INTORNO AL 1936 HANNO DATO UNA SOLUZIONE NEGATIVA AL PROBLEMA DELLA DECISIONE PER LA LOGICA DEL I ORDINE CONSIDERATO DA DAVID HILBERT TRA I PROBLEMI PIÙ IMPORTANTI DEL SUO TEMPO

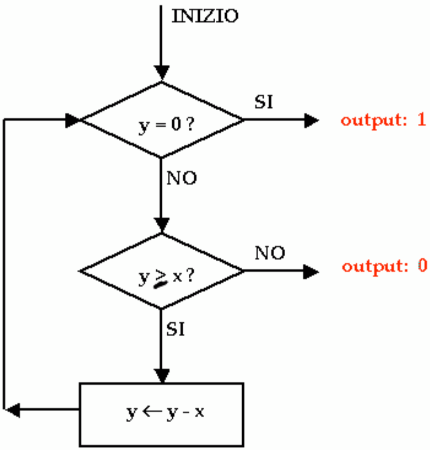
- PER RISOLVERE IN POSITIVO UN PROBLEMA ALGORITMICO E' SUFFICIENTE ESIBIRE UNA PROCEDURA IN UN FORMALISMO ACCETTABILE.

ES.
- ALGORITMO DI EUCLIDE
- ALGORITMI PER LE 4 OPERAZIONI SUI NUMERI INTERI
- ALGORITMO DI FATTORIZZAZIONE
- CRIVELLO DI ERATOSTENE
- TEST DI DIVISIBILITA'

...

ESEMPIO: TEST DI DIVISIBILITÀ: y È DIVISIBILE PER x?

FORMALISMO: DIAGRAMMI DI FLUSSO $\boxed{\exists z:\ y = xz?}$



- SI PUÒ DIMOSTRARE CHE:
  - IL PROCESSO DI CALCOLO TERMINA SEMPRE
  - SE L'OUTPUT È 1 E L'ASSEGNAMENTO $y \leftarrow y - x$ È ESEGUITO z VOLTE, ALLORA VALE $y = z \cdot x$
  - SE L'OUTPUT È 0 ALLORA y NON È DIVISIBILE PER x

QUINDI IL TEST DI DIVISIBILITÀ È COMPUTABILE

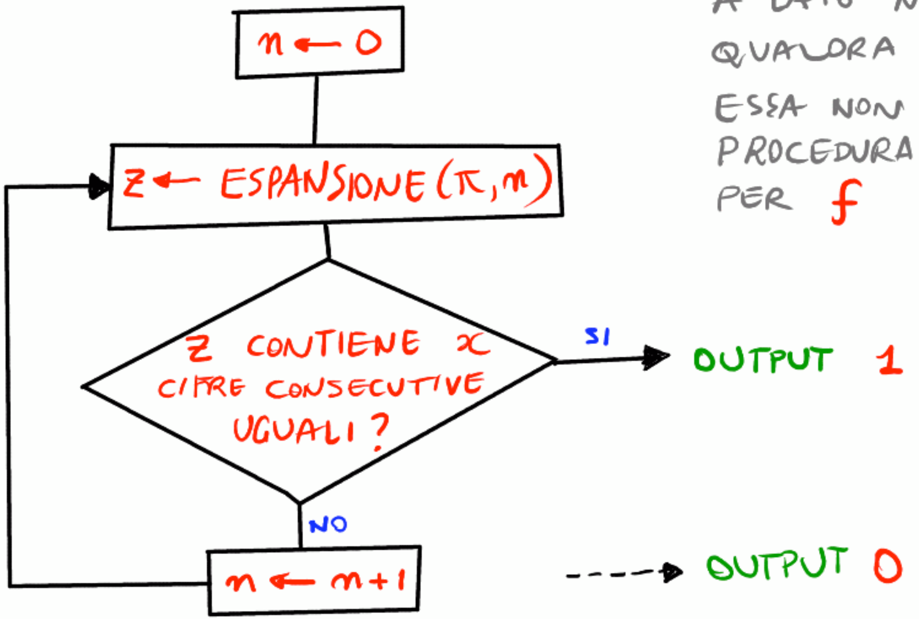
ESEMPIO

SI CONSIDERI LA FUNZIONE

$$f(x) = \begin{cases} 1 & \text{SE L'ESPANSIONE DECIMALE DI } \pi \text{ CONTIENE} \\ & \text{ESATTAMENTE } x \text{ CIFRE CONSECUTIVE EGUALI} \\ 0 & \text{ALTRIMENTI} \end{cases}$$

PUR DISPONENDO DI UNA PROCEDURA FINITA CHE PER OGNI n CALCOLA L'ESPANSIONE DI π CON n CIFRE DECIMALI, ALLO STATO ATTUALE DELLE CONOSCENZE NON SI DISPONE DI UN PROCESSO DI CALCOLO PER f.

ESEMPIO (CONTINUA)



- POICHÈ LA PROCEDURA A LATO NON SI FERMA QUALORA $f(x) = 0$, ESSA NON È UNA PROCEDURA DI CALCOLO PER f

- PER RISOLVERE IN NEGATIVO UN PROBLEMA ALGORITMICO E' NECESSARIO AVERE UN'IDEA CHIARA DI TUTTI I POSSIBILI ALGORITMI E QUINDI FARE VEDERE CHE CIASCUNO DI ESSI NON E' IN GRADO DI RISOLVERE IL PROBLEMA IN ESAME

- OCCORRE QUINDI FORMALIZZARE IL CONCETTO INTUITIVO DI ALGORITMO:

  SEQUENZA DI ISTRUZIONI CIASCUNA DELLE QUALI PUÒ ESSERE ESEGUITA IN MANIERA NON AMBIGUA ED EFFETTIVA DA UN'OPPORTUNA MACCHINA

- IL PROBLEMA DIVENTA ALLORA QUELLO DI FORMALIZZARE UN OPPORTUNO MODELLO DI CALCOLO UNIVERSALE (MACCHINA)
- SONO STATI PROPOSTI PARECCHI MODELLI:
  - MACCHINE DI TURING (TURING, 1936)
  - λ-CALCOLO (CHURCH, 1936)
  - FUNZIONI PARZIALI RICORSIVE (GÖDEL-KLEENE, 1936)
  - SISTEMI DEDUTTIVI CANONICI (POST, 1945)
  - SISTEMI DI MARKOV (MARKOV, 1951)
  - MACCHINE A REGISTRI ILLIMITATI (URM) (SHEPERDSON & STURGIS, 1936)

- CIASCUNO DI TALI SISTEMI DA' LUOGO AD UNA DEFINIZIONE DI FUNZIONI CALCOLABILI (OVVERO DI PROBLEMI RISOLVIBILI IN MANIERA EFFETTIVA)
- E' INTERESSANTE OSSERVARE CHE TUTTI I SUDDETTI MODELLI DI CALCOLO DEFINISCONO LA MEDESIMA CLASSE DI FUNZIONI CALCOLABILI

TESI DI CHURCH-TURING

UNA FUNZIONE E' CALCOLABILE SE E SOLO SE E' CALCOLABILE IN UNO DEI SUDDETTI MODELLI DI CALCOLO.

UNLIMITED REGISTER MACHINES (URM)

(MACCHINE A REGISTRI ILLIMITATI)

- SI TRATTA DI UNA VARIANTE DELLE MACCHINE DI SHEPERDSON & STURGIS [1963]
- UNA URM E' DOTATA DI UN ARRAY INFINITO DI REGISTRI $R_1, R_2, R_3, \ldots$ CIASCUNO DEI QUALI CONTIENE UN NUMERO NATURALE (INTERO NON NEGATIVO) $r_1, r_2, r_3, \ldots$

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
r_1	r_2	r_3	r_4	r_5	r_6	r_7	...

- LO STATO DI UNA URM E' IL CONTENUTO DEI SUOI REGISTRI: $\vec{r} \in \mathbb{N}^{\infty}$

- LO STATO DI UNA URM PUÒ ESSERE MODIFICATO DALL'ESECUZIONE DI UN URM-PROGRAMMA
- UN URM-PROGRAMMA È UNA SEQUENZA FINITA $I_1, I_2, \ldots, I_s$ DI ISTRUZIONI DI UNO DEI SEGUENTI QUATTRO TIPI:
  - RESET
  - INCREMENTO
  - (ASSEGNAMENTO)
  - SALTO CONDIZIONATO

(s È LA LUNGHEZZA DEL PROGRAMMA)

- UNA CONFIGURAZIONE ISTANTANEA E' UNA COPPIA $(k, \vec{r}) \in \mathbb{N}^+ \times \mathbb{N}^\infty$ CON
  - k INTERO POSITIVO, DETTO CONTATORE DI PROGRAMMA
  - $\vec{r}$ STATO

(INFORMALMENTE, k RAPPRESENTA L'INDICE DELLA ISTRUZIONE CHE STA PER ESSERE ESEGUITA QUANDO LO STATO DEI REGISTRI E' $\vec{r}$)

ESEMPIO

$(5, (1, 0, 0, \ldots))$ E' UNA CONFIGURAZIONE ISTANTANEA

ISTRUZIONI DI RESET

$Z(n)$ $\quad (n \in \mathbb{N}^+)$ $\qquad [\, R_n \leftarrow 0 \,]$

SEMANTICA: $(k, \vec{r}) \xrightarrow{Z(n)} (k+1, \vec{r'})$ CON $r'_i = \begin{cases} r_i, & i \neq n \\ 0, & i = n \end{cases}$

ESEMPIO

R1	R2	R3	R4	R5	R6	...
9	6	5	23	7	0	...

$\downarrow Z(3)$

R1	R2	R3	R4	R5	R6	...
9	6	0	23	7	0	...

ISTRUZIONI DI INCREMENTO

$S(n)$ $\quad (n \in \mathbb{N}^+)$ $\qquad [\, R_n \leftarrow R_n + 1 \,]$

SEMANTICA: $(k, \vec{r}) \xrightarrow{S(n)} (k+1, \vec{r'})$ CON $r'_i = \begin{cases} r_i, & i \neq n \\ r_n + 1, & i = n \end{cases}$

ESEMPIO

$k=5$

R_1	R_2	R_3	R_4	R_5	R_6	...
9	6	0	23	7	0	...

$\downarrow S(5)$

$k=6$

R_1	R_2	R_3	R_4	R_5	R_6	...
9	6	0	23	8	0	...

ISTRUZIONI DI ASSEGNAMENTO

$T(m,n)$ $(m,n \in \mathbb{N}^+)$ $[\, R_n \leftarrow R_m \,]$

SEMANTICA: $(k, \vec{r}) \xrightarrow{T(m,n)} (k+1, \vec{r}\,')$ CON $r_i' = \begin{cases} r_i, & i \neq n \\ r_m, & i = n \end{cases}$

ESEMPIO

$k=5$

R_1	R_2	R_3	R_4	R_5	R_6	...
9	6	0	23	8	0	...

$\downarrow T(5,1)$

$k=6$

R_1	R_2	R_3	R_4	R_5	R_6	...
8	6	0	23	8	0	...

ISTRUZIONI DI SALTO CONDIZIONATO

$J(m,n,q)$ $(m,n,q \in \mathbb{N}^+)$ [if $R_m = R_n$ then goto q]

SEMANTICA: $(k, \vec{r}) \xrightarrow{J(m,n,q)} (k', \vec{r})$ CON $k' = \begin{cases} q, & r_m = r_n \\ k+1, & r_m \neq r_n \end{cases}$

ESEMPIO

$k=5$

R_1	R_2	R_3	R_4	R_5	R_6	...
8	6	0	23	8	0	...

$$\downarrow J(5,1,9)$$

$k'=9$

R_1	R_2	R_3	R_4	R_5	R_6	...
8	6	0	23	8	0	...

COMPUTAZIONI

- DATI
  - $P = I_1, I_2, \dots, I_s$ (PROGRAMMA DI LUNGHEZZA s)
  - $\vec{r}$ (STATO DEI REGISTRI)

LA COMPUTAZIONE $P(\vec{r})$ DI P A PARTIRE DALLO STATO INIZIALE $\vec{r}$ È LA SEQUENZA (FINITA O INFINITA) DI CONFIGURAZIONI ISTANTANEE

$$(k_1, \vec{r^{(1)}}), (k_2, \vec{r^{(2)}}), (k_3, \vec{r^{(3)}}), \dots$$

TALE CHE

COMPUTAZIONI (CONTINUA)

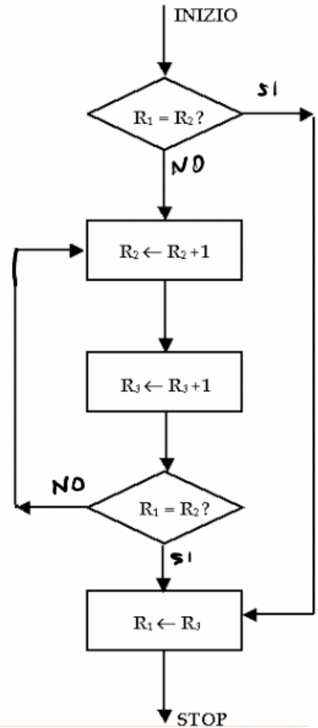
- $k_1 = 1$, $\vec{r^{(1)}} = \vec{r}$

  (CIOE' LA COMPUTAZIONE INIZIA CON LA PRIMA ISTRUZIONE DI P E CON LO STATO INIZIALE $\vec{r}$)

- SE $(k_i, \vec{r^{(i)}})$ E' L' i-ESIMA CONFIGURAZIONE ISTANTANEA DELLA COMPUTAZIONE $P(\vec{r})$ ALLORA
  - SE $k_i \leq s$, $(k_i, \vec{r^{(i)}})$ HA SUCCESSORE IN $P(\vec{r})$ E SI HA $(k_i, \vec{r^{(i)}}) \xrightarrow{I_{k_i}} (k_{i+1}, \vec{r^{(i+1)}})$
  - SE $k_i > s$, $(k_i, \vec{r^{(i)}})$ NON HA SUCCESSORE IN $P(\vec{r})$ (CIOE' $(k_i, \vec{r^{(i)}})$ E' LA CONFIGURAZIONE FINALE DI $P(\vec{r})$)

ESEMPIO

$I_1 : J(1,2,6)$		if $R_1 = R_2$ then goto 6
$I_2 : S(2)$	[2]	$R_2 \leftarrow R_2 + 1$
$I_3 : S(3)$		$R_3 \leftarrow R_3 + 1$
$I_4 : J(1,2,6)$		if $R_1 = R_2$ then goto 6
$I_5 : J(1,1,2)$		if $R_1 = R_1$ then goto 2
$I_6 : T(3,1)$	[6]	$R_1 \leftarrow R_3$



	R_1	R_2	R_3	R_4	R_5	...
stato iniziale	9	7	0	0	0	...
1. $J(1,2,6)$	9	7	0	0	0	...
2. $S(2)$	9	8	0	0	0	...
3. $S(3)$	9	8	1	0	0	...
4. $J(1,2,6)$	9	8	1	0	0	...
5. $J(1,1,2)$	9	8	1	0	0	...
2. $S(2)$	9	9	1	0	0	...
3. $S(3)$	9	9	2	0	0	...
4. $J(1,2,6)$	9	9	2	0	0	...
6. $T(3,1)$	2	9	2	0	0	...

7.

stato finale

- SI TRATTA DI UNA COMPUTAZIONE FINITA O TERMINANTE

	R_1	R_2	R_3	R_4	R_5	...
stato iniziale	6	7	0	0	0	...
1. $J(1,2,6) \rightarrow$	6	7	0	0	0	...
2. $S(2) \rightarrow$	6	8	0	0	0	...
3. $S(3) \rightarrow$	6	8	1	0	0	...
4. $J(1,2,6) \rightarrow$	6	8	1	0	0	...
5. $J(1,1,2) \rightarrow$	6	8	1	0	0	...
2. $S(2) \rightarrow$	6	9	1	0	0	...
3. $S(3) \rightarrow$	6	9	2	0	0	...
4. $J(1,2,6) \rightarrow$	6	9	2	0	0	...
5. $J(1,1,2) \rightarrow$	6	9	2	0	0	...
2. $S(2) \rightarrow$	6	10	2	0	0	...
3. $S(3) \rightarrow$	6	10	3	0	0	...
...		...				

- SI DIMOSTRA CHE TALE COMPUTAZIONE E' INFINITA CIOE' NON TERMINANTE

ALCUNE NOTAZIONI UTILI

- SIA P UN PROGRAMMA ED $\vec{r}$ UNO STATO (INIZIALE) DEI REGISTRI.
- SE LA COMPUTAZIONE $P(\vec{r})$ E' TERMINANTE SCRIVEREMO $P(\vec{r})\downarrow$
- SE LA COMPUTAZIONE $P(\vec{r})$ E' NON TERMINANTE SCRIVEREMO $P(\vec{r})\uparrow$
- CON LA NOTAZIONE $P(a_1, a_2, \ldots, a_n)$ INDICHEREMO LA COMPUTAZIONE $P(\vec{r})$ DOVE

$$r_i = \begin{cases} a_i & \text{SE } i \leq n \\ 0 & \text{ALTRIMENTI} \end{cases}$$

CONVENZIONI DI INPUT E OUTPUT

- SIA $P(a_1, a_2, \dots, a_n)\downarrow$.
  L'OUTPUT DI P SU INPUT $a_1, a_2, \dots, a_n$ E' IL CONTENUTO DEL REGISTRO R_1 NELLO STATO FINALE DELLA COMPUTAZIONE $P(a_1, a_2, \dots, a_n)$.
- SCRIVEREMO $P(a_1, a_2, \dots, a_n)\downarrow b$ PER INDICARE CHE b E' L'OUTPUT DELLA COMPUTAZIONE $P(a_1, \dots, a_n)$

- SIA $f : A \to B$ UNA FUNZIONE PARZIALE (TALE CIOE' CHE $Dom(f) \subseteq A$)
- SE f E' DEFINITA SU a SCRIVEREMO $f(a)\downarrow$
ALTRIMENTI SCRIVEREMO $f(a)\uparrow$

- SIANO $f, g : A \to B$ FUNZIONI PARZIALI,
SCRIVEREMO $f \simeq g$ PER INDICARE CHE

1) PER OGNI $a \in A$: $f(a)\downarrow$ SE E SOLO SE $g(a)\downarrow$
2) PER OGNI $a \in A$: SE $f(a)\downarrow$ ALLORA $f(a) = g(a)$

- SIA P UN PROGRAMMA.
- PER OGNI $n \geq 1$ PONIAMO

$$f_P^{(n)}(a_1, a_2, \dots, a_n) = \begin{cases} b & \text{SE } P(a_1, a_2, \dots, a_n) \downarrow b \\ \uparrow & \text{ALTRIMENTI} \end{cases}$$

- LA FUNZIONE (PARZIALE) $f_P^{(n)} : \mathbb{N}^n \to \mathbb{N}$ E' LA FUNZIONE n-ARIA CALCOLATA DA P
- UNA FUNZIONE PARZIALE $g : \mathbb{N}^n \to \mathbb{N}$ SI DICE URM-CALCOLABILE SE ESISTE UN PROGRAMMA URM P TALE CHE $g \simeq f_P^{(n)}$

- INDICHEREMO CON $\mathcal{C}$ LA COLLEZIONE DI TUTTE LE FUNZIONI URM-CALCOLABILI (DI QUALUNQUE ARIETÀ)
- INDICHEREMO CON $\mathcal{C}_n$ LA COLLEZIONE DI TUTTE LE FUNZIONI n-ARIE URM-CALCOLABILI
- PERTANTO VALE

$$\mathcal{C} = \bigcup_{n=1}^{\infty} \mathcal{C}_n$$

ESEMPI

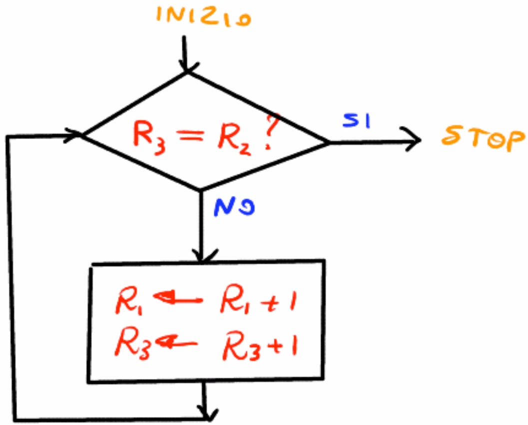
$\lambda xy.\, x+y$

I_1 $J(3,2,5)$

I_2 $S(1)$

I_3 $S(3)$

I_4 $J(1,1,1)$



- È FACILE VERIFICARE CHE IL PROGRAMMA

$I_1 : J(1,2,6)$

$I_2 : S(2)$

$I_3 : S(3)$

$I_4 : J(1,2,6)$

$I_5 : J(1,1,2)$

$I_6 : T(3,1)$

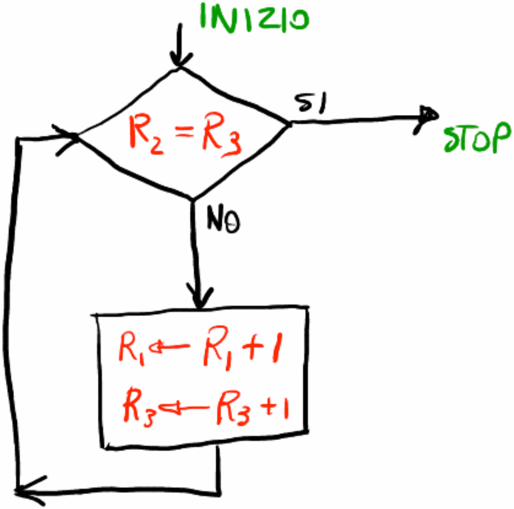
CALCOLA LA FUNZIONE BINARIA

$$f(x,y) = \begin{cases} x - y & \text{SE } x \geq y \\ \uparrow & \text{ALTRIMENTI} \end{cases}$$

ALCUNI ESEMPI

- SI CONSIDERI LA FUNZIONE $x+y$
- IL SEGUENTE PROGRAMMA CALCOLA $x+y$

1: $J(2,3,5)$
2: $S(1)$
3: $S(3)$
4: $J(1,1,1)$

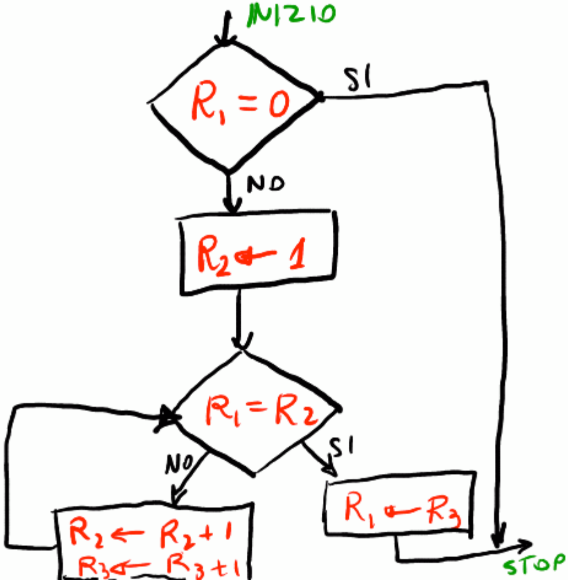


STATO TIPICO

R_1	R_2	R_3	R_4	R_5	
$x+k$	y	k	0	0	...

- SI CONSIDERI LA FUNZIONE $x \dot{-} 1 = \begin{cases} x-1 & \text{SE } x \geq 1 \\ 0 & \text{ALTRIMENTI} \end{cases}$
- IL SEGUENTE PROGRAMMA CALCOLA $x \dot{-} 1$

1: J(1,2,8)
2: S(2)
3: J(1,2,7)
4: S(2)
5: S(3)
6: J(1,1,3)
7: T(3,1)



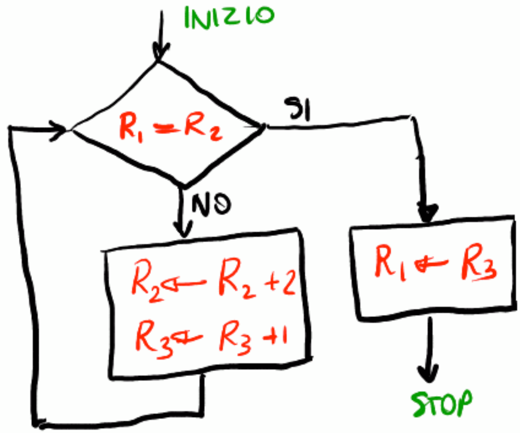
STATO TIPICO

R_1	R_2	R_3	R_4	R_5	
x	$k+1$	k	0	0	...

- SI CONSIDERI LA FUNZIONE $f(x) = \begin{cases} \frac{1}{2}x & \text{SE } x \text{ E' PARI} \\ \uparrow & \text{ALTRIMENTI} \end{cases}$

- IL SEGUENTE PROGRAMMA CALCOLA f

1: $J(1,2,6)$
2: $S(2)$
3: $S(2)$
4: $S(3)$
5: $J(1,1,1)$
6: $T(3,1)$



STATO TIPICO

R_1	R_2	R_3	R_4	R_5	...
x	$2k$	k	0	0	...

ESERCIZI

1. SI DIMOSTRI CHE LE SEGUENTI FUNZIONI SONO CALCOLABILI ESIBENDO OPPORTUNI PROGRAMMI URM

(a) $f(x) = \begin{cases} 0 & \text{SE } x = 0 \\ 1 & \text{SE } x \neq 0 \end{cases}$

(b) $f(x) = 5$

(c) $f(x,y) = \begin{cases} 0 & \text{SE } x = y \\ 1 & \text{SE } x \neq y \end{cases}$

(d) $f(x,y) = \begin{cases} 0 & \text{SE } x \leq y \\ 1 & \text{SE } x > y \end{cases}$

(e) $f(x) = \begin{cases} \frac{1}{3}x & \text{SE } x \text{ E' UN MULTIPLO DI } 3 \\ \uparrow & \text{ALTRIMENTI} \end{cases}$

(f) $f(x) = \left\lfloor \frac{2x}{3} \right\rfloor$

2. SIA P UN PROGRAMMA URM CHE NON CONTIENE ALCUNA ISTRUZIONE DEL TIPO $J(m,n,p)$. SI DIMOSTRI CHE ESISTE $m \in \mathbb{N}$ TALE CHE:

- $f_P^{(1)}(x) = m \quad \forall x \in \mathbb{N}$, OPPURE
- $f_P^{(1)}(x) = x + m \quad \forall x \in \mathbb{N}$.

3. DATO UN PROGRAMMA URM P SI DIMOSTRI CHE ESISTE UN ALTRO PROGRAMMA URM P' TALE CHE

- P' NON CONTIENE ALCUNA ISTRUZIONE DEL TIPO $T(m,n)$
- $f_P^{(n)}(\vec{x}) = f_{P'}^{(n)}(\vec{x})$, $\forall \vec{x} \in \mathbb{N}^n$

CORRETTEZZA PARZIALE E TOTALE DI PROGRAMMI URM

UN PROGRAMMA P SI DICE PARZIALMENTE CORRETTO RISPETTO A DATE SPECIFICHE DI INPUT $\mathcal{A}$ E DI OUTPUT $\mathcal{B}$ SE

- PER OGNI INPUT $\vec{x}$ SODDISFACENTE LE SPECIFICHE $\mathcal{A}$, SE L'ESECUZIONE DI P SU $\vec{x}$ E' TERMINANTE, ALLORA L'OUTPUT DI P SODDISFA LE SPECIFICHE $\mathcal{B}$

ESEMPI

$\lambda xy.\, x+y$

$I_1 \quad J(3,2,5)$

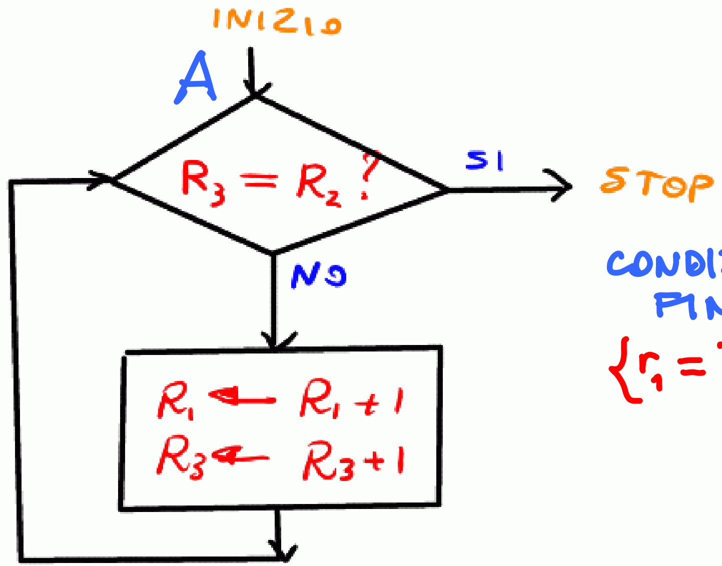
$I_2 \quad S(1)$

$I_3 \quad S(3)$

$I_4 \quad J(1,1,1)$

CONDIZIONI INIZIALI

$$\left\{ \begin{array}{l} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{array} \right\}$$



CONDIZIONE FINALE

$$\{ r_1 = \bar{r}_1 + \bar{r}_2 \}$$

DEBBONO VALERE

CONDIZIONI INIZIALI $\Longrightarrow$ A

$A \wedge r_3 = r_2 \Longrightarrow$ CONDIZIONE FINALE

$$(A \wedge r_3 \neq r_2 \wedge r_1' = r_1 + 1 \wedge r_3' = r_3 + 1 \wedge r_2' = r_2) \longrightarrow A^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$$

ESEMPI

$\lambda xy.\,x+y$

$I_1 \quad J(3,2,5)$

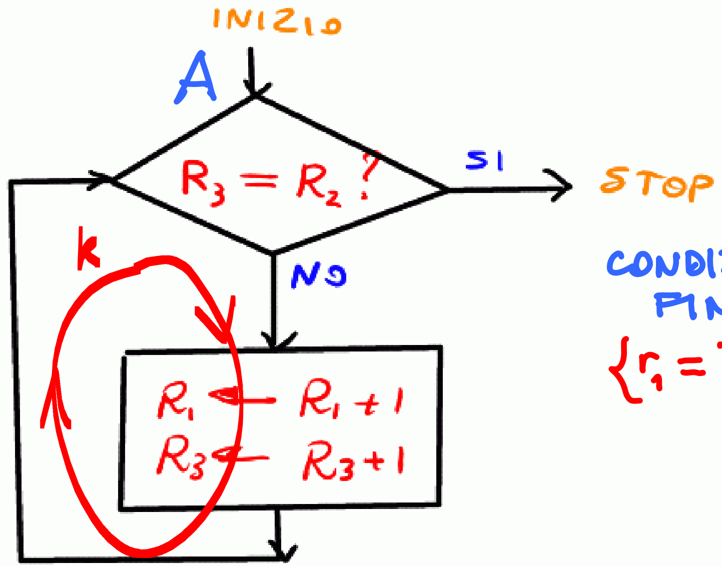
$I_2 \quad S(1)$

$I_3 \quad S(3)$

$I_4 \quad J(1,1,1)$

CONDIZIONI INIZIALI

$$\begin{Bmatrix} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{Bmatrix}$$



CONDIZIONE FINALE

$$\{ r_1 = \bar{r}_1 + \bar{r}_2 \}$$

DOPO k CICLI:

$$\begin{Bmatrix} r_1^{(k)} = \bar{r}_1 + k \\ r_2^{(k)} = \bar{r}_2 \\ r_3^{(k)} = k \end{Bmatrix} \Rightarrow \begin{matrix} r_1^{(k)} = \bar{r}_1 + r_3^{(k)} \\ r_2^{(k)} = \bar{r}_2 \end{matrix} \Rightarrow A = \begin{Bmatrix} r_1 = \bar{r}_1 + r_3 \\ r_2 = \bar{r}_2 \end{Bmatrix}$$

CONDIZIONI INIZIALI $\Longrightarrow$ A

$$\begin{cases} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{cases} \Longrightarrow \begin{cases} r_1 = \bar{r}_1 + r_3 \\ r_2 = \bar{r}_2 \end{cases}$$

INFATTI: $r_1 = \bar{r}_1 = \bar{r}_1 + 0 = \bar{r}_1 + r_3$

$A \wedge r_3 = r_2 \Longrightarrow$ CONDIZIONE FINALE

INFATTI:

$$\begin{cases} r_1 = \bar{r}_1 + r_3 \\ r_2 = \bar{r}_2 \end{cases} \wedge \quad r_3 = r_2 \Longrightarrow r_1 = \bar{r}_1 + r_3 = \bar{r}_1 + r_2 = \bar{r}_1 + \bar{r}_2$$

(CONDIZIONE FINALE)

$$\left(A \wedge r_3 \neq r_2 \wedge r_1' = r_1 + 1 \wedge r_3' = r_3 + 1 \wedge r_2' = r_2\right) \rightarrow A^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$$

INFATTI:

$$\left\{\begin{array}{l} r_1 = \bar{r}_1 + r_3 \\ r_2 = \bar{r}_2 \end{array}\right\} \wedge r_3 \neq r_2 \wedge r_1' = r_1 + 1 \wedge r_3' = r_3 + 1 \wedge r_2' = r_2$$

$$\rightarrow \left\{\begin{array}{l} r_1' = r_1 + 1 = \bar{r}_1 + r_3 + 1 = \bar{r}_1 + r_3' \\ r_2' = r_2 = \bar{r}_2 \end{array}\right\} \equiv \left\{\begin{array}{l} r_1' = \bar{r}_1 + r_3' \\ r_2' = \bar{r}_2 \end{array}\right\}$$

$$\equiv A^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$$

PERTANTO IL NOSTRO PROGRAMMA E' PARZIALMENTE CORRETTO, RISPETTO ALLE SPECIFICHE DATE SULLE CONDIZIONI INIZIALI E QUELLE FINALI

PER DIMOSTRARE LA TOTALE CORRETTEZZA, OCCORRE VERIFICARE CHE IL NOSTRO PROGRAMMA SI FERMA SEMPRE (TERMINAZIONE)

E' SUFFICIENTE TROVARE UNA MISURA A VALORI IN UN INSIEME BEN FONDATO CHE DECRESCE STRETTAMENTE AD OGNI ITERAZIONE.

SPESSO COME INSIEME BEN FONDATO SI PUÒ SCEGLIERE $\mathbb{N}$

NEL NOSTRO CASO, C'E' UN'ESPRESSIONE NATURALE DEI NOSTRI REGISTRI CHE DECRESCE STRETTAMENTE AD OGNI ITERAZIONE?

UNA POSSIBILE FUNZIONE MISURA NEL NOSTRO CASO E': $r_2 - r_3$

MA OCCORRERA' VERIFICARE CHE VALGONO SEMPRE

(a) $r_2 - r_3 \in \mathbb{N}$, CIOE' $r_2 - r_3 \geq 0$

(b) $r_2' - r_3' < r_2 - r_3$

A TAL FINE E' UTILE RAFFORZARE L'INVARIANTE:

$$B \equiv A \wedge r_2 - r_3 \geq 0 \equiv \begin{cases} r_1 = \overline{r}_1 + r_3 \\ r_2 = \overline{r}_2 \\ r_2 - r_3 \geq 0 \end{cases}$$

E QUINDI VERIFICARE CHE:

CONDIZIONI INIZIALI $\Longrightarrow B$

$B \wedge r_3 = r_2 \Longrightarrow$ CONDIZIONE FINALE

$$(B \wedge r_3 \neq r_2 \wedge r_1' = r_1 + 1 \wedge r_3' = r_3 + 1 \wedge r_2' = r_2) \longrightarrow B^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$$

AD ESEMPIO, VERIFICHIAMO LA CONDIZIONE (b)

$$(B \wedge r_3 \neq r_2 \wedge r_3' = r_3 + 1 \wedge r_1' = r_1 + 1 \wedge r_2' = r_2) \rightarrow (r_2' - r_3' < r_2 - r_3)$$

BANALMENTE SI HA:

$$r_2' - r_3' = r_2 - r_3 - 1 < r_2 - r_3$$

ESEMPIO

CONDIZIONI INIZIALI $\begin{cases} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{cases}$

$I_1 : J(1,2,6)$ — if $R_1 = R_2$ then goto 6

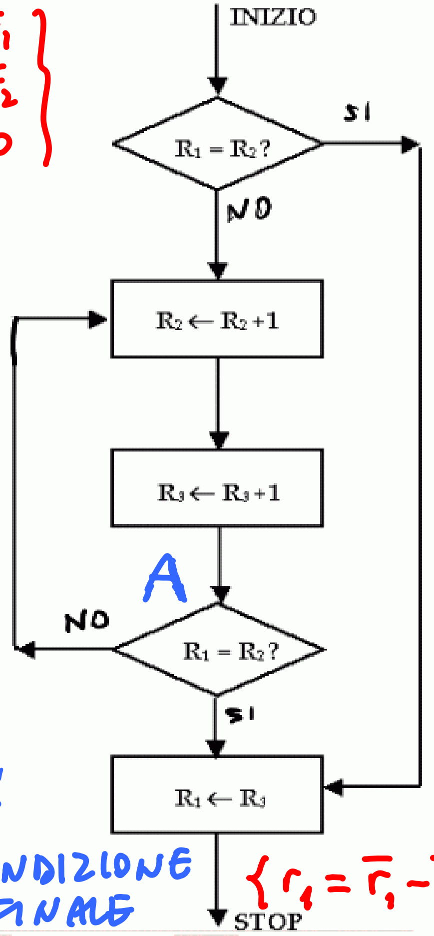
$I_2 : S(2)$ — [2] $R_2 \leftarrow R_2 + 1$

$I_3 : S(3)$ — $R_3 \leftarrow R_3 + 1$

$I_4 : J(1,2,6)$ — if $R_1 = R_2$ then goto 6

$I_5 : J(1,1,2)$ — if $R_1 = R_1$ then goto 2

$I_6 : T(3,1)$ — [6] $R_1 \leftarrow R_3$



CONDIZIONE FINALE $\{ r_1 = \bar{r}_1 - \bar{r}_2 \}$

- $(\text{CONDIZIONI INIZIALI} \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3) \rightarrow r_1' = \bar{r}_1 - \bar{r}_2$
- $(\text{CONDIZIONI INIZIALI} \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1) \rightarrow A^{r_1 r_2 r_3}_{r_1' r_2' r_3'}$
- $(A \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1) \rightarrow A^{r_1 r_2 r_3}_{r_1' r_2' r_3'}$
- $(A \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3) \rightarrow r_1' = \bar{r}_1 - \bar{r}_2$

- $\left(\begin{array}{l}\text{CONDIZIONI}\\ \text{INIZIALI}\end{array} \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3\right) \rightarrow r_1' = \bar{r}_1 - \bar{r}_2$

$$\left(\begin{cases} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{cases} \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3\right) \rightarrow$$

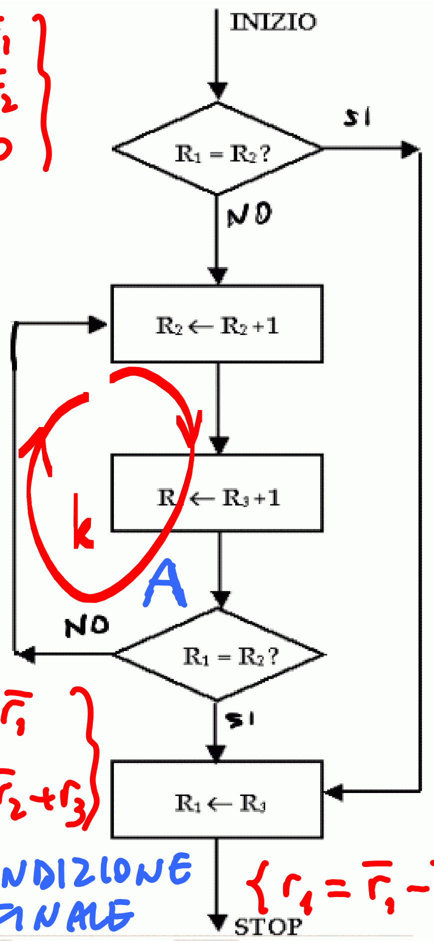
$$r_1' = r_3 = 0 = r_1 - r_2 = \bar{r}_1 - \bar{r}_2$$

$\blacksquare$

ESEMPIO

CONDIZIONI INIZIALI $\begin{cases} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{cases}$

I_1 : $J(1,2,6)$		if $R_1 = R_2$ then goto 6
I_2 : $S(2)$	[2]	$R_2 \leftarrow R_2 + 1$
I_3 : $S(3)$		$R_3 \leftarrow R_3 + 1$
I_4 : $J(1,2,6)$		if $R_1 = R_2$ then goto 6
I_5 : $J(1,1,2)$		if $R_1 = R_1$ then goto 2
I_6 : $T(3,1)$	[6]	$R_1 \leftarrow R_3$



DOPO k CICLI:

$$\left.\begin{aligned} r_1^{(k)} &= \bar{r}_1 \\ r_2^{(k)} &= \bar{r}_2 + k + 1 \\ r_3^{(k)} &= k + 1 \end{aligned}\right\} \Rightarrow \begin{aligned} r_1^{(k)} &= \bar{r}_1 \\ r_2^{(k)} &= \bar{r}_2 + r_3^{(k)} \end{aligned}$$

$$A = \begin{cases} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 + r_3 \end{cases}$$

CONDIZIONE FINALE $\{ r_1 = \bar{r}_1 - \bar{r}_2 \}$

$$\left(\text{CONDIZIONI INIZIALI} \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1\right) \rightarrow A^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$$

$$\left(\begin{Bmatrix} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{Bmatrix} \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1 \right)$$

$$\Rightarrow \begin{Bmatrix} r_1' = r_1 = \bar{r}_1 \\ r_2' = r_2 + 1 = \bar{r}_2 + 1 = \bar{r}_2 + (r_3 + 1) = \bar{r}_2 + r_3' \end{Bmatrix} \Rightarrow \begin{Bmatrix} r_1' = r_1 \\ r_2' = \bar{r}_2 + r_3' \end{Bmatrix}$$

$$\rightarrow A^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$$

- $\left(A \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1\right) \to A^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$

$$\left(\left\{\begin{array}{l} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 + r_3 \end{array}\right\} \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1\right)$$

$$\Rightarrow \left\{\begin{array}{l} r_1' = r_1 = \bar{r}_1 \\ r_2' = r_2 + 1 = \bar{r}_2 + r_3 + 1 = \bar{r}_2 + r_3' \end{array}\right\}$$

$$\Rightarrow \left\{\begin{array}{l} r_1' = \bar{r}_1 \\ r_2' = \bar{r}_2 + r_3' \end{array}\right\} \equiv A^{r_1\, r_2\, r_3}_{r_1'\, r_2'\, r_3'}$$

- $(A \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3) \to r_1' = \bar{r}_1 - \bar{r}_2$

$$\left(\left\{ \begin{array}{l} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 + r_3 \end{array} \right\} \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3 \right)$$

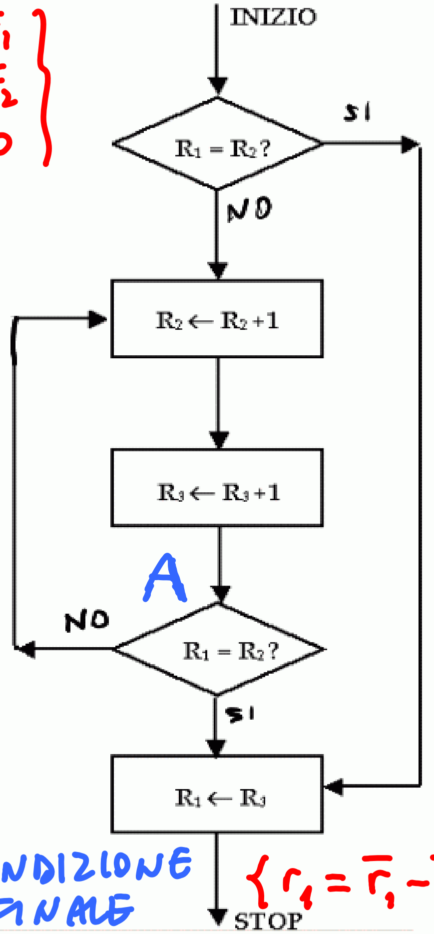
$$\Rightarrow \quad r_1' = r_3 = r_2 - \bar{r}_2 = r_1 - \bar{r}_2 = \bar{r}_1 - \bar{r}_2$$

$$\Rightarrow \quad r_1' = \bar{r}_1 - \bar{r}_2$$

ESEMPIO

CONDIZIONI INIZIALI $\left\{ \begin{array}{l} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{array} \right\}$

I_1 : $J(1,2,6)$		if $R_1 = R_2$ then goto 6
I_2 : $S(2)$	[2]	$R_2 \leftarrow R_2 + 1$
I_3 : $S(3)$		$R_3 \leftarrow R_3 + 1$
I_4 : $J(1,2,6)$		if $R_1 = R_2$ then goto 6
I_5 : $J(1,1,2)$		if $R_1 = R_1$ then goto 2
I_6 : $T(3,1)$	[6]	$R_1 \leftarrow R_3$



CONDIZIONE FINALE $\{ r_1 = \bar{r}_1 - \bar{r}_2 \}$

TERMINAZIONE

CONDIZ. INIZIALI* $\equiv$ CONDIZ. INIZIALI $\wedge$ $\bar{r}_2 \leq \bar{r}_1$

$$A^* \equiv A \wedge r_2 \leq r_1$$

FUNZIONE MISURA $\equiv$ $r_1 - r_2$

CONDIZIONI DI VERIFICA

- $(\text{CONDIZIONI INIZIALI}^* \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3) \rightarrow r_1' = \bar{r}_1 - \bar{r}_2$
- $(\text{CONDIZIONI INIZIALI}^* \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1) \rightarrow A^{*\,r_1\,r_2\,r_3}_{\;\;r_1'\,r_2'\,r_3'}$
- $(A^* \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1) \rightarrow A^{*\,r_1\,r_2\,r_3}_{\;\;r_1'\,r_2'\,r_3'}$
- $(A^* \wedge r_1 = r_2 \wedge r_1' = r_3 \wedge r_2' = r_2 \wedge r_3' = r_3) \rightarrow r_1' = \bar{r}_1 - \bar{r}_2$

CONDIZIONE DI TERMINAZIONE

$$(A^* \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1) \rightarrow (r_1' - r_2' < r_1 - r_2)$$

$$(A^* \wedge r_1 \neq r_2 \wedge r_1' = r_1 \wedge r_2' = r_2 + 1 \wedge r_3' = r_3 + 1)$$

$$\Rightarrow \quad r_1' - r_2' = r_1 - r_2 - 1 < r_1 - r_2$$

PERTANTO IL NOSTRO PROGRAMMA CALCOLA LA FUNZIONE

$$f(x,y) = \begin{cases} x - y & \text{SE } x \geq y \\ ? & \text{ALTRIMENTI} \end{cases}$$

- STUDIAMO LA CONDIZIONE INIZIALE $\bar{r}_1 < \bar{r}_2$

IN TAL CASO POSSIAMO DIMOSTRARE CHE VALE L'INVARIANTE

$$\bar{A} \equiv A \wedge r_1 < r_2$$

E QUINDI IL TEST DELL'ISTRUZIONE $I_4 : J(1,2,6)$ DARÀ IN QUESTO CASO ESITO SEMPRE NEGATIVO E IL SUCCESSIVO CONTROLLO $I_5 : J(1,1,2)$ RIPORTERÀ A 2 IL VALORE DEL CONTATORE DI PROGRAMMA.

NE SEGUE CHE CON CONDIZIONE INIZIALE $\bar{r}_1 < \bar{r}_2$, IL NOSTRO PROGRAMMA NON SI FERMA MAI E QUINDI LA FUNZIONE CALCOLATA È:

$$f(x,y) = \begin{cases} x - y & \text{SE } x \geq y \\ \uparrow & \text{ALTRIMENTI} \end{cases}$$

$\blacksquare$