

TEORIA DELLA **COMPUTABILITÀ**

Prof. Domenico Cantone

A.A. 2013/14

PROGRAMMA

I PARTE: TEORIA DELLA CALCOLABILITA'

FUNZIONI CALCOLABILI

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

GENERAZIONE DI FUNZIONI CALCOLABILI

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

TESI DI CHURCH

- ALTRI APPROCCI ALLA CALCOLABILITA'
- 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 1° ORDINE CONSIDERATO DA DAVID HILBERT TRA I PROBLEMI PIÙ IMPORTANTI DEL SUO TEMPO

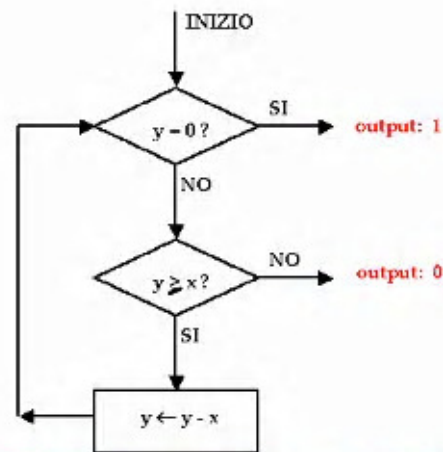
- PER RISOLVERE *IN POSITIVO* UN PROBLEMA ALGORITMICO E' SUFFICIENTE ESIBIRE UNA PROCEDURA IN UN FORMALISMO ACCETTABILE.

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

...

ESEMPIO: TEST DI DIVISIBILITA': y E' DIVISIBILE PER z ?

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



- SI PUO' DIMOSTRARE CHE:

- IL PROCESSO DI CALCOLO TERMINA SEMPRE
- SE L'OUTPUT E' 1 E L'ASSEGNAIMENTO $y \leftarrow y - x$ E' ESEGUITO z VOLTE, ALLORA VALE $y = z \cdot x$
- SE L'OUTPUT E' 0 ALLORA y NON E' DIVISIBILE PER x

QUINDI IL TEST DI DIVISIBILITA' E' COMPUTABILE

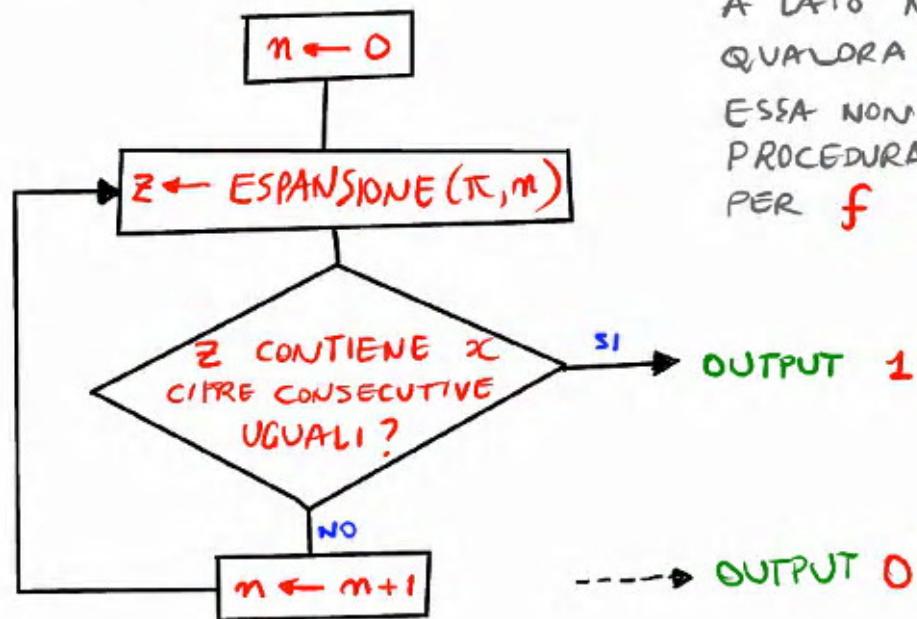
ESEMPIO

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)



• POICHE' LA PROCEDURA A LATO NON SI FERMA QUANDO $f(x) = 0$, ESSA NON E' UNA PROCEDURA DI CALCOLO PER f

-----> OUTPUT 0

- PER RISOLVERE IN NEGATIVO UN PROBLEMA ALGORITMICO È NECESSARIO AVERE UN'IDEA CHIARA DI TUTTI I POSSIBILI ALGORITMI E QUINDI FARE VEDERE CHE CIASCUNO DI ESSI NON È 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)
(SHEPARDSON & 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 SHEPHERDSON & STURGIS [1963]
- UNA URM È DOTATA DI UN ARRAY INFINITO DI REGISTRI R_1, R_2, R_3, \dots CIASCUNO DEI QUALI CONTIENE UN NUMERO NATURALE (INTERO NON NEGATIVO) r_1, r_2, r_3, \dots

| 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 È IL CONTENUTO DEI SUOI REGISTRI: $\vec{r} \in \mathbb{N}^{\infty}$

- LO STATO DI UNA URM PUO' ESSERE MODIFICATO DALL'ESECUZIONE DI UN URM-PROGRAMMA
 - UN URM-PROGRAMMA E' UNA SEQUENZA FINITA I_1, I_2, \dots, I_s DI ISTRUZIONI DI UNO DEI SEGUENTI QUATTRO TIPI:
 - RESET
 - INCREMENTO
 - (ASSEGNAIMENTO)
 - SALTO CONDIZIONATO
- (s E' LA LUNGHEZZA DEL PROGRAMMA)

- UNA CONFIGURAZIONE ISTANTANEA È 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 È \vec{r})

ESEMPIO

$(5, (1, 0, 0, \dots))$ È UNA CONFIGURAZIONE ISTANTANEA

ISTRUZIONI DI RESET

$Z(m)$ ($m \in \mathbb{N}^+$)

$[R_m \leftarrow 0]$

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

ESEMPIO

| R_1 | R_2 | R_3 | R_4 | R_5 | R_6 | ... |
|-------|-------|-------|-------|-------|-------|-----|
| 9 | 6 | 5 | 23 | 7 | 0 | ... |

$\downarrow Z(3)$

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

ISTRUZIONI DI INCREMENTO

$S(m)$ ($m \in \mathbb{N}^+$)

$[R_m \leftarrow R_m + 1]$

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

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

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

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

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

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

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

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)

- $k_1 = 1$, $\vec{r}^{(1)} = \vec{r}$
(CIOÈ LA COMPUTAZIONE INIZIA CON LA PRIMA ISTRUZIONE DI P E CON LO STATO INIZIALE \vec{r})
- SE $(k_i, \vec{r}^{(i)})$ È 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 $P(\vec{r})$ (CIOÈ $(k_i, \vec{r}^{(i)})$ È LA CONFIGURAZIONE FINALE DI $P(\vec{r})$)

ESEMPIO

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

$I_2: S(2)$

$I_3: S(3)$

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

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

$I_6: T(3,1)$

if $R_1 = R_2$ then goto 6

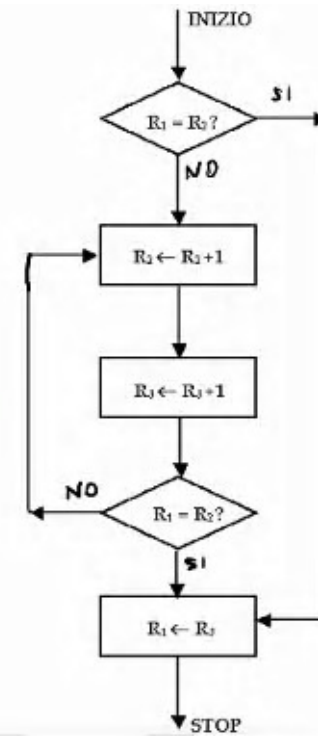
[2] $R_2 \leftarrow R_2 + 1$

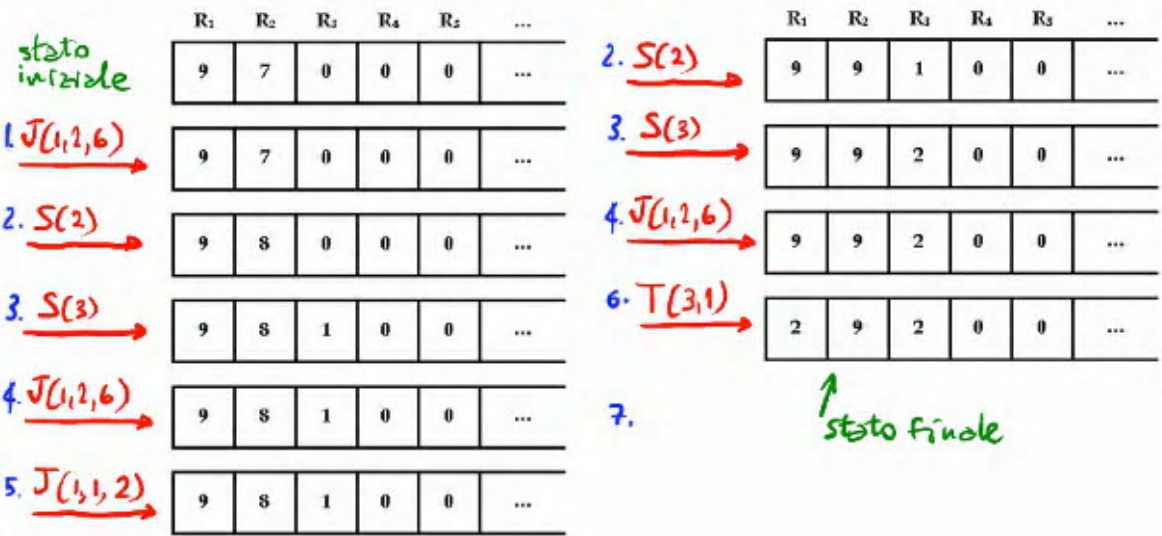
$R_3 \leftarrow R_3 + 1$

if $R_1 = R_2$ then goto 6

if $R_1 = R_3$ then goto 2

[6] $R_1 \leftarrow R_3$





-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. $J(1,2,6)$ | 6 | 7 | 0 | 0 | 0 | ... |
| 2. $S(2)$ | 6 | 8 | 0 | 0 | 0 | ... |
| 3. $S(3)$ | 6 | 8 | 1 | 0 | 0 | ... |
| 4. $J(1,2,6)$ | 6 | 8 | 1 | 0 | 0 | ... |
| 5. $J(1,1,2)$ | 6 | 8 | 1 | 0 | 0 | ... |

| | R ₁ | R ₂ | R ₃ | R ₄ | R ₅ | ... |
|---------------|----------------|----------------|----------------|----------------|----------------|-----|
| 2. $S(2)$ | 6 | 9 | 1 | 0 | 0 | ... |
| 3. $S(3)$ | 6 | 9 | 2 | 0 | 0 | ... |
| 4. $J(1,2,6)$ | 6 | 9 | 2 | 0 | 0 | ... |
| 5. $J(1,1,2)$ | 6 | 9 | 2 | 0 | 0 | ... |
| 2. $S(2)$ | 6 | 10 | 2 | 0 | 0 | ... |
| 3. $S(3)$ | 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 $P(a_1, a_2, \dots, a_n)$ 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 È IL CONTENUTO DEL REGISTRO R_1 NELLO STATO FINALE DELLA COMPUTAZIONE $P(a_1, a_2, \dots, a_n)$.
- SCRIVEREMMO $P(a_1, a_2, \dots, a_n) \downarrow b$ PER INDICARE CHE b È L'OUTPUT DELLA COMPUTAZIONE $P(a_1, \dots, a_n)$

- SIA $f: A \rightarrow B$ UNA FUNZIONE PARZIALE
TALE CIOE' CHE $\text{Dom}(f) \subseteq A$

- SE f E' DEFINITA SU a SCRIVEREMO $f(a) \downarrow$
ALTRIMENTI SCRIVEREMO $f(a) \uparrow$

- SIANO $f, g: A \rightarrow B$ 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) $f_P^{(n)}: \mathbb{N}^n \rightarrow \mathbb{N}$ E' LA
FUNZIONE n -ARIA CALCOLATA DA P

- UNA FUNZIONE PARZIALE $g: \mathbb{N}^n \rightarrow \mathbb{N}$ SI DICE
URM-CALCOLABILE SE ESISTE UN PROGRAMMA
URM P TALE CHE $g \approx f_P^{(n)}$

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

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

ESEMPLI

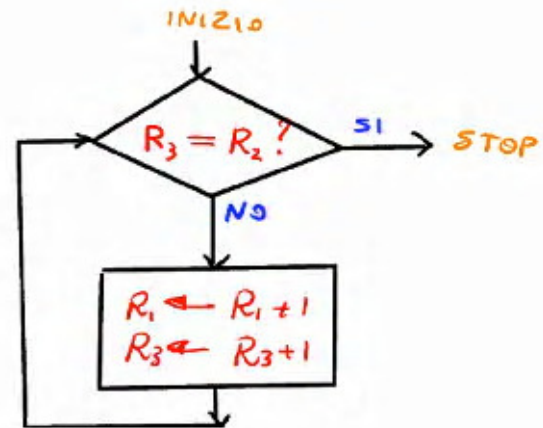
$\lambda xy. x+y$

$I_1 J(3, 2, 5)$

$I_2 S(1)$

$I_3 S(3)$

$I_4 J(1, 1, 1)$



- E' 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: $S(1)$

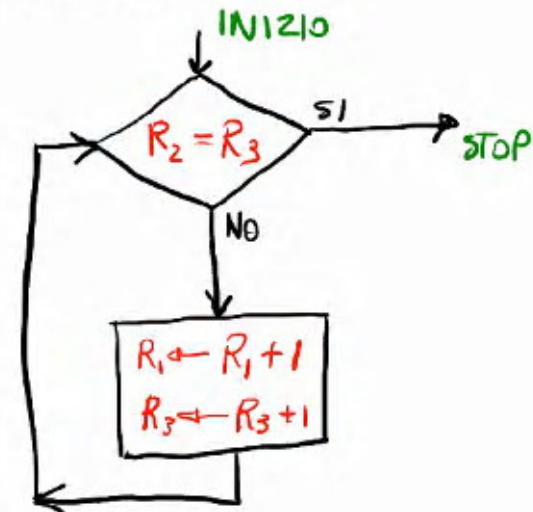
3: $S(3)$

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

$$x \dot{-} 1 = \begin{cases} x-1 & \text{SE } x \geq 1 \\ 0 & \text{ALTRIMENTI} \end{cases}$$

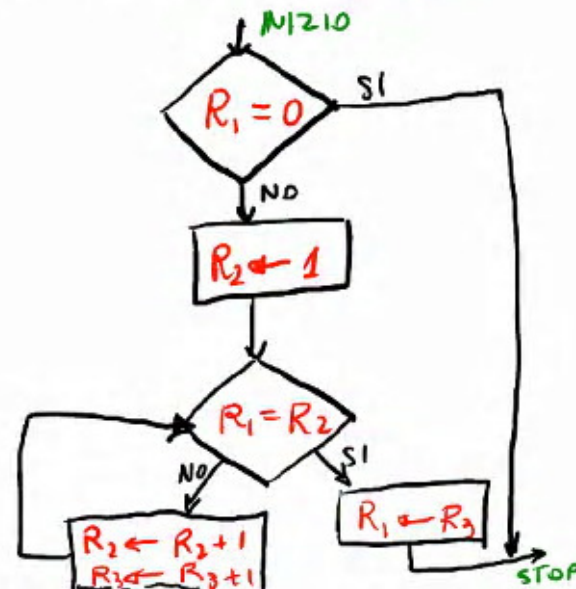
- IL SEGUENTE PROGRAMMA

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 | ... |

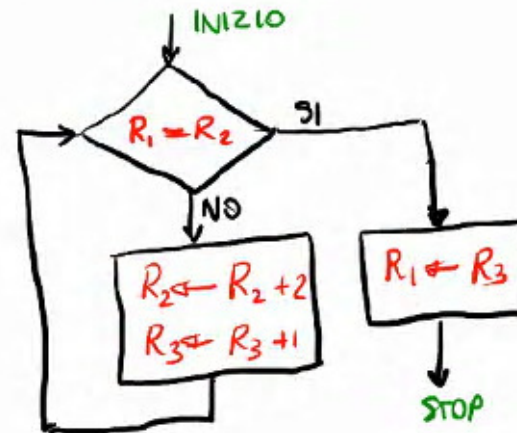


- SI CONSIDERI LA FUNZIONE $f(x) = \begin{cases} 1/2x & \text{SE } x \text{ E' PARI} \\ \uparrow & \text{ALTRIMENTI} \end{cases}$
- IL SEGUENTE PROGRAMMA CALCOLA f

- 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

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

$$(b) \quad f(x) = 5$$

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

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

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

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

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

$$- f_p^{(1)}(x) = m \quad \forall x \in \mathbb{N}, \quad \text{OPPURE}$$

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

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

- P' NON CONTIENE ALCUNA ISTRUZIONE
DEL TIPO $T(m, m)$

- $f_P^{(m)}(\vec{x}) = f_{P'}^{(m)}(\vec{x})$, $\forall \vec{x} \in \mathbb{N}^m$

CORRETTEZZA PARZIALE

E TOTALE DI

PROGRAMMI URM

UN PROGRAMMA P SI DICE PARZIALMENTE CORRETTO
RISPETTO A DATE SPECIFICHE DI INPUT A E DI
OUTPUT B SE

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

ESEMPI

$\lambda xy. x+y$

$I_1 J(3, 2, 5)$

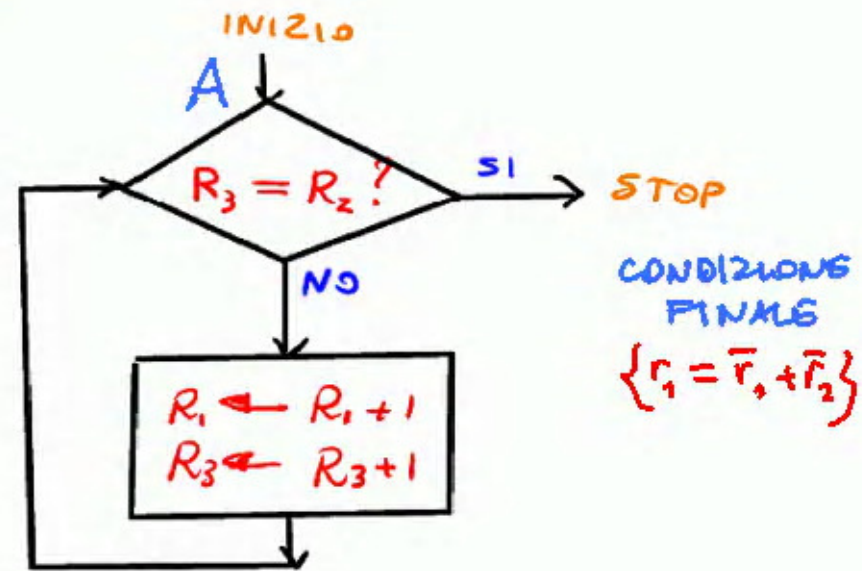
$I_2 S(1)$

$I_3 S(3)$

$I_4 J(1, 1, 1)$

CONDIZIONI
INIZIALI

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



DEBONO VALERE

CONDIZIONI
INIZIALI \Rightarrow A

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

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

$\begin{matrix} r_1 & r_2 & r_3 \\ r_1' & r_2' & r_3' \end{matrix}$

ESEMPI

$$\lambda xy. x+y$$

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

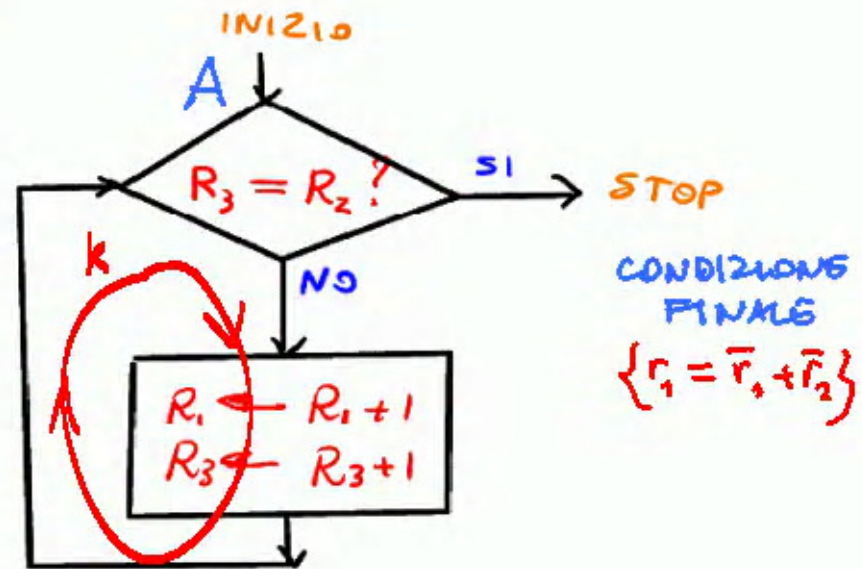
$$I_2 S(1)$$

$$I_3 S(3)$$

$$I_4 J(1, 1, 1)$$

CONDIZIONI
INIZIALI

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



DOPO k CICLI:

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

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

CONDIZIONI
INIZIALI

\Rightarrow A

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

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

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

INFATTI:

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

(CONDIZIONE FINALE)

$$(A \wedge r_3 \neq r_2 \wedge r'_1 = r_1 + 1 \wedge r'_3 = r_3 + 1 \wedge r'_2 = r_2) \rightarrow A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$$

INFATTI:

$$\left. \begin{matrix} r_1 = \bar{r}_1 + r_3 \\ r_2 = \bar{r}_2 \end{matrix} \right\} \wedge r_3 \neq r_2 \wedge r'_1 = r_1 + 1 \wedge r'_3 = r_3 + 1 \wedge r'_2 = r_2$$

$$\rightarrow \left. \begin{matrix} r'_1 = r_1 + 1 = \bar{r}_1 + r_3 + 1 = \bar{r}_1 + r'_3 \\ r'_2 = r_2 = \bar{r}_2 \end{matrix} \right\} \equiv \left. \begin{matrix} r'_1 = \bar{r}_1 + r'_3 \\ r'_2 = \bar{r}_2 \end{matrix} \right\}$$

$$\equiv A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$$

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 FONDATA CHE DECRESCA STRETTAMENTE
AD OGNI ITERAZIONE.

SPESSE COME INSIEME BEN FONDATA SI PUO' SCEGLIERE \mathbb{N}

NEL NOSTRO CASO, C'E' UN'ESPRESSIONE NATURALE DEI NOSTRI
REGISTRI CHE DECRESCA 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 :

$$B \equiv A \wedge r_2 - r_3 \geq 0 \equiv \left\{ \begin{array}{l} r_1 = \bar{r}_1 + r_3 \\ r_2 = \bar{r}_2 \\ r_2 - r_3 \geq 0 \end{array} \right\}$$

E QUINDI VERIFICARE CHE:

CONDIZIONI INIZIALI $\Rightarrow B$

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

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

$\begin{matrix} r_1 & r_2 & r_3 \\ r_1' & r_2' & r_3' \end{matrix}$

AD ESEMPIO, VERIFICHIAMO LA CONDIZIONE (b)

$$(B \wedge r_3 \neq r_2 \wedge r'_3 = r_3 + 1 \wedge r'_1 = r_3 + 1 \wedge r'_2 = r_2) \\ \rightarrow (r'_2 - r'_3 < r_2 - r_3)$$

BANALMENTE SI HA:

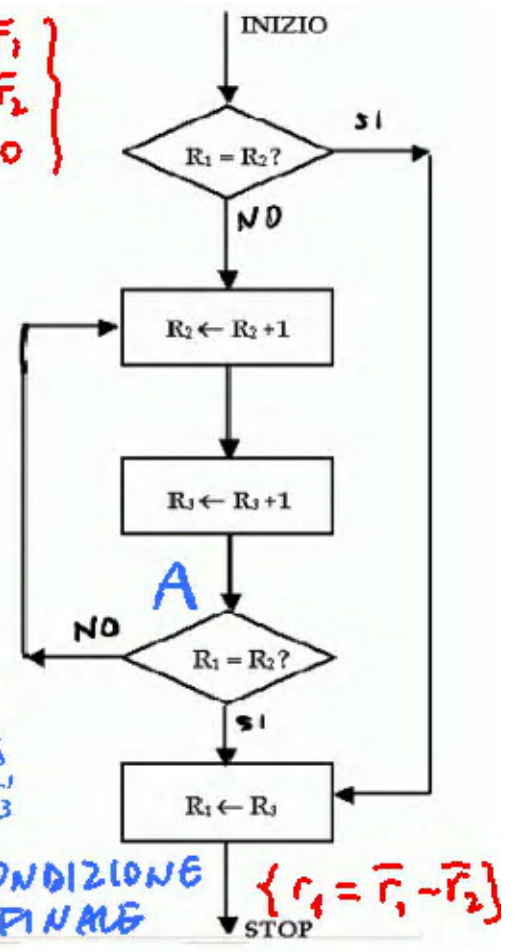
$$r'_2 - r'_3 = r_2 - r_3 - 1 < r_2 - r_3$$

ESEMPIO

$I_1 : J(1,2,6)$ 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] $R_1 \leftarrow R_3$

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

- (CONDIZIONI INIZIALI $\wedge r_1 = r_2 \wedge r'_1 = r_3 \wedge r'_2 = r_2 \wedge r'_3 = r_3$) $\rightarrow r'_1 = \bar{r}_1 - \bar{r}_2$
- (CONDIZIONI INIZIALI $\wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1$) $\rightarrow A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$
- ($A \wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1$) $\rightarrow A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$
- ($A \wedge r_1 = r_2 \wedge r'_1 = r_3 \wedge r'_2 = r_2 \wedge r'_3 = r_3$) $\rightarrow r'_1 = \bar{r}_1 - \bar{r}_2$



• (CONDIZIONI INIZIALI $\wedge r_1 = r_2 \wedge r'_1 = r_3 \wedge r'_2 = r_2 \wedge r'_3 = r_3$) $\rightarrow r'_1 = \bar{r}_1 - \bar{r}_2$

$$\left(\begin{cases} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{cases} \wedge r_1 = r_2 \wedge r'_1 = r_3 \wedge r'_2 = r_2 \wedge r'_3 = r_3 \right) \rightarrow$$

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



ESEMPIO

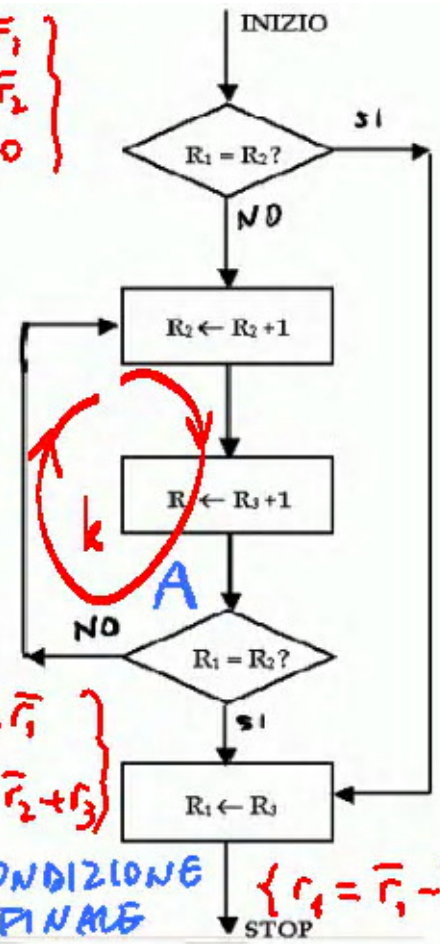
- $I_1 : J(1,2,6)$
- $I_2 : S(2)$
- $I_3 : S(3)$
- $I_4 : J(1,2,6)$
- $I_5 : J(1,1,2)$
- $I_6 : T(3,1)$

- if $R_1 = R_2$ then goto 6
- [2] $R_2 \leftarrow R_2 + 1$
- $R_3 \leftarrow R_3 + 1$
- if $R_1 = R_2$ then goto 6
- if $R_1 = R_3$ then goto 2
- [6] $R_1 \leftarrow R_3$

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

DDPO k CICLI:

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



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

(CONDIZIONI INIZIALI $\wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1$) $\rightarrow A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$

$$\left(\begin{matrix} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{matrix} \wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1 \right)$$

$$\Rightarrow \left\{ \begin{matrix} r'_1 = r_1 = \bar{r}_1 \\ r'_2 = r_2 + 1 = \bar{r}_2 + 1 = \bar{r}_2 + (r_3 + 1) = \bar{r}_2 + r'_3 \end{matrix} \right\} \Rightarrow \left\{ \begin{matrix} r'_1 = \bar{r}_1 \\ r'_2 = \bar{r}_2 + r'_3 \end{matrix} \right\}$$

$$\Rightarrow A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$$

$$\bullet (A \wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1) \rightarrow A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$$

$$\left(\begin{matrix} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 + r_3 \end{matrix} \right) \wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1$$

$$\Rightarrow \left\{ \begin{matrix} r'_1 = r_1 = \bar{r}_1 \\ r'_2 = r_2 + 1 = \bar{r}_2 + r_3 + 1 = \bar{r}_2 + r'_3 \end{matrix} \right\}$$

$$\Rightarrow \left\{ \begin{matrix} r'_1 = \bar{r}_1 \\ r'_2 = \bar{r}_2 + r'_3 \end{matrix} \right\} \equiv A \begin{matrix} r_1 & r_2 & r_3 \\ r'_1 & r'_2 & r'_3 \end{matrix}$$

$$\bullet (A \wedge r_1 = r_2 \wedge r'_1 = r_3 \wedge r'_2 = r_2 \wedge r'_3 = r_3) \rightarrow r'_1 = \bar{r}_1 - \bar{r}_2$$

$$\left(\begin{array}{l} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 + r_3 \end{array} \right) \wedge r_1 = r_2 \wedge r'_1 = r_3 \wedge r'_2 = r_2 \wedge r'_3 = r_3$$

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

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

ESEMPIO

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

$I_2: S(2)$

$I_3: S(3)$

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

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

$I_6: T(3,1)$

if $R_1 = R_2$ then goto 6

[2] $R_2 \leftarrow R_2 + 1$

$R_3 \leftarrow R_3 + 1$

if $R_1 = R_2$ then goto 6

if $R_1 = R_3$ then goto 2

[6] $R_1 \leftarrow R_3$

CONDIZIONI
INIZIALI

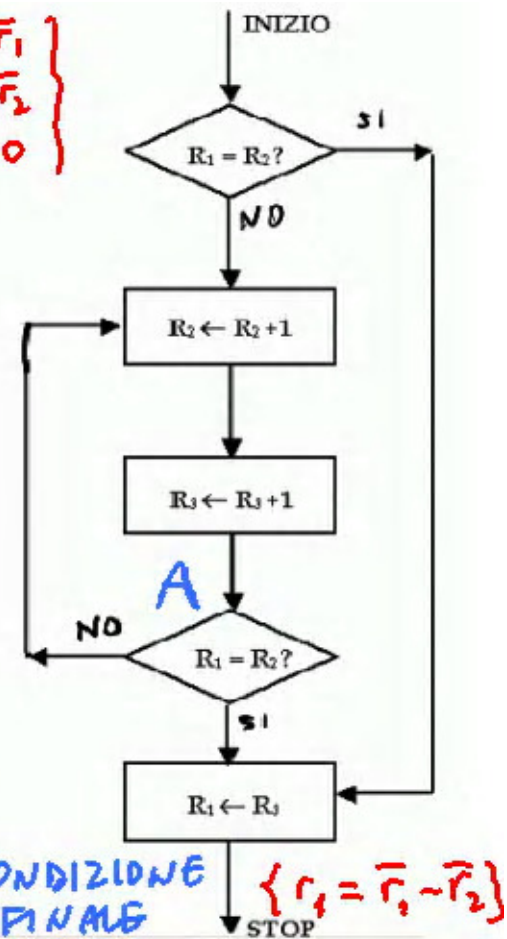
$$\begin{cases} r_1 = \bar{r}_1 \\ r_2 = \bar{r}_2 \\ r_3 = 0 \end{cases}$$

TERMINAZIONE

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

$A^* \equiv A \wedge r_3 \leq r_1$

FUNZIONE MISURA $\equiv r_1 - r_2$



CONDIZIONI DI VERIFICA

- (CONDIZIONI INIZIALI* $\wedge r_1 = r_2 \wedge r'_1 = r_3 \wedge r'_2 = r_2 \wedge r'_3 = r_3$) $\rightarrow r'_1 = \bar{r}_1 - \bar{r}_2$
- (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

$$(A^* \wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1) \rightarrow (r'_1 - r'_2 < r_1 - r_2)$$

$$(A^* \wedge r_1 \neq r_2 \wedge r'_1 = r_1 \wedge r'_2 = r_2 + 1 \wedge r'_3 = r_3 + 1)$$

$$\Rightarrow r'_1 - r'_2 = r_1 - r_2 - 1 < r_1 - r_2$$

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 $I_4: J(1, 2, 6)$
DARA' IN QUESTO CASO ESITO SEMPRE NEGATIVO E IL SUCCESSIVO
CONTROLLO $I_5: J(1, 1, 2)$ RIPORTERA' 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 E':

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

