Calcolabilità, Esecutori, Sintassi: Esercizi Vari

Sommario: 24/2-4/3, 2020

- Esercizi di Programmazione in C:
  Espressività e Strutturare Programmi in Sezioni Autonome
- Esercizi su Funzioni Calcolabili:
  Universale, Meaning, Rappresentazione Programmi
- Esercizi su Esecutori:
  Interprete, Compilatore, Realizzazione Mista di MA
- Esercizi su Lessico, Sintassi:
  Grammatica, Parse Tree Set, Deriva, Linguaggio

ESERCIZI - 10/3/2020

Esercizio (1)

Scrivere un programma C che legga una sequenza di valori ordinabili, calcoli l'ordinamento della sequenza,implementando l'algoritmo di QuickSort, e stampi la sequenza ordinata. Allo scopo completare il testo esponendo eventuali vincoli che si ritenga utile assumere.

Soluzione: Requisiti aggiuntivi

ESERCIZI - 10/3/2020

Esercizio (1)

Scrivere un programma C che legga una sequenza di valori ordinabili, calcoli l'ordinamento della sequenza,implementando l'algoritmo di QuickSort, e stampi la sequenza ordinata. Allo scopo completare il testo esponendo eventuali vincoli che si ritenga utile assumere.

Soluzione

(a) Requisiti aggiuntivi

```
Esercizio L1.1 del 28/2/2019.
Scrivere un programma C che calcoli la funzione "ordinamento di sequenza"
implementando l'algoritmo QuickSort. Allo scopo, si completi il programma
delle operazioni per lettura e stampa di sequenze e si aggiungano gli even-
tuali vincoli che si ritengano necessari.
**
* Marco Bellia
**
Soluzione.
Vincoli.
1) Esistenza bound, N, alla dimensione della sequenza da ordinare. Il bound è
   trattato come Parametro di Programma ed usato per l'allocazione statica della
   memoria necessaria pre contenere la sequenza.
2) Elementi della sequenza di tipo noto. Il tipo è int.
3) Lettura numero elementi K (≤N) effettivi della sequenza (da buffer di input)
4) Valore k degli elementi effettivi rilevata dopo lettura della sequenza
5) Relazione d'ordine assunta come operazione nota. Relazione < su int.
```

(b) stesura del codice

Rinviato al Laboratorio di prossimo venerdì 16/3/2017 ?

ESERCIZI - 10/3/2020

Esercizio (2)

Se e dove, nella formulazione $\mathcal{F}$, vediamo che $\mathcal{F}$ è numerabile.

- $\mathcal{F} \equiv [\mathcal{D} \to \mathcal{D}]$ *with* $\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}]$
  - *(Universale)* $\exists\, \mathcal{U} \in \mathcal{F}$,
  - *(Meaning)* $\exists\, [|\_|] : \mathcal{P} \to \mathcal{F}$, *suriettiva,*
  - $\forall \mathtt{p} \in \mathcal{P}$,
    $$\mathcal{U}(\overline{\mathtt{p}}) = \mathtt{g} \in (\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}] \equiv)\ \mathcal{F}$$
  - $\forall \mathtt{g} \in \mathcal{F}$, *sia* $\mathtt{g} = [|\mathtt{p}|]$, *per qualche* $p \in \mathcal{P}$
    $$\mathcal{U}(\overline{\mathtt{p}})(\mathtt{x}) = \mathtt{g}(\mathtt{x}) \quad \forall \mathtt{x} \in \mathcal{D}$$

*With $\mathcal{D}$ numerabile, $\mathcal{L} \in$ **LP**, $\mathcal{P}$ insieme programmi di $\mathcal{L}$, $\overline{\ } : \mathcal{P} \to \mathcal{D}$, iniettiva.*

Soluzione

ESERCIZI - 10/3/2020

Esercizio (2)

Se e dove, nella formulazione $\mathcal{F}$, vediamo che $\mathcal{F}$ è numerabile.

- $\mathcal{F} \equiv [\mathcal{D} \to \mathcal{D}]$ *with* $\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}]$
  - *(Universale)* $\exists\, \mathcal{U} \in \mathcal{F}$,
  - *(Meaning)* $\exists\, [|\_|] : \mathcal{P} \to \mathcal{F}$, *suriettiva,*
  - $\forall \mathtt{p} \in \mathcal{P}$,
    $\mathcal{U}(\overline{\mathtt{p}}) = \mathtt{g} \in (\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}] \equiv)\ \mathcal{F}$
  - $\forall \mathtt{g} \in \mathcal{F}$, *sia* $\mathtt{g} = [|\mathtt{p}|]$, *per qualche* $p \in \mathcal{P}$
    $\mathcal{U}(\overline{\mathtt{p}})(\mathtt{x}) = \mathtt{g}(\mathtt{x}) \quad \forall \mathtt{x} \in \mathcal{D}$

With $\mathcal{D}$ numerabile, $\mathcal{L} \in$ **LP**, *$\mathcal{P}$ insieme programmi di $\mathcal{L}$, $\overline{\ } : \mathcal{P} \to \mathcal{D}$, iniettiva.*

Soluzione
Immediato da:
-assunzione: $\mathcal{D}$ *numerabile*
-proprietà Isomorfismo: $\mathcal{D} \cong ([\mathcal{D} \to \mathcal{D}] \equiv)\mathcal{F}$
per bijettività: $\mathcal{F}$ ha cardinalità numerabile.

ESERCIZI - 10/3/2020

Esercizio (3)

(a) *Se e dove, nella formulazione $\mathcal{F}$, vediamo che la funzione $[|\_|]$ è calcolabile?* (b) *E nel caso, quale funzione calcola?*

- $\mathcal{F} \equiv [\mathcal{D} \to \mathcal{D}]$ *with* $\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}]$
  - *(Universale)* $\exists\, \mathcal{U} \in \mathcal{F}$,
  - *(Meaning)* $\exists\, [|\_|] : \mathcal{P} \to \mathcal{F}$, *suriettiva,*
  - $\forall \mathtt{p} \in \mathcal{P}$,
    $\mathcal{U}(\overline{\mathtt{p}}) = \mathtt{g} \in (\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}] \equiv)\ \mathcal{F}$
  - $\forall \mathtt{g} \in \mathcal{F}$, *sia* $\mathtt{g} = [|\mathtt{p}|]$, *per qualche* $p \in \mathcal{P}$
    $\mathcal{U}(\overline{\mathtt{p}})(\mathtt{x}) = \mathtt{g}(\mathtt{x}) \quad \forall \mathtt{x} \in \mathcal{D}$

With $\mathcal{D}$ numerabile, $\mathcal{L} \in$ **LP**, *$\mathcal{P}$ insieme programmi di $\mathcal{L}$, $\overline{\ } : \mathcal{P} \to \mathcal{D}$, iniettiva.*

Soluzione (a)

Soluzione (b)

ESERCIZI - 10/3/2020

Esercizio (3)

(a) *Se e dove, nella formulazione $\mathcal{F}$, vediamo che la funzione $[|\_|]$ è calcolabile?* (b) *E nel caso, quale funzione calcola?*

- $\mathcal{F} \equiv [\mathcal{D} \to \mathcal{D}]$ *with* $\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}]$
  - *(Universale)* $\exists\, \mathcal{U} \in \mathcal{F}$,
  - *(Meaning)* $\exists\, [|\_|] : \mathcal{P} \to \mathcal{F}$, *suriettiva,*
  - $\forall \mathtt{p} \in \mathcal{P}$,
    $\mathcal{U}(\overline{\mathtt{p}}) = \mathtt{g} \in (\mathcal{D} \cong [\mathcal{D} \to \mathcal{D}] \equiv)\ \mathcal{F}$
  - $\forall \mathtt{g} \in \mathcal{F}$, *sia* $\mathtt{g} = [|\mathtt{p}|]$, *per qualche* $p \in \mathcal{P}$
    $\mathcal{U}(\overline{\mathtt{p}})(\mathtt{x}) = \mathtt{g}(\mathtt{x}) \quad \forall \mathtt{x} \in \mathcal{D}$

With $\mathcal{D}$ numerabile, $\mathcal{L} \in$ **LP**, *$\mathcal{P}$ insieme programmi di $\mathcal{L}$, $\overline{\ } : \mathcal{P} \to \mathcal{D}$, iniettiva.*

Soluzione (a)
Non lo vediamo. Infatti, Meaning $[|\_|]$ che soddisfano la formulazione $\mathcal{F}$ non sono unici. Sia $\mathcal{H}_{\mathcal{L}}$ l'insieme di tali mapping per $\mathcal{L}$. Mostriamo che $\mathcal{H}_{\mathcal{L}} \cap \mathcal{F} \neq \{\}$. Sia $\mathtt{h}$ tale che:

$$\forall \mathtt{p} \in \mathcal{P}, \quad \mathtt{h}(\mathtt{p}) = \mathcal{U}(\overline{\mathtt{p}}).$$

Allora osserviamo che:
(1) $\mathtt{h} \in \mathcal{H}_{\mathcal{L}}$. Infatti soddisfa tutte le condizioni su $[|\_|]$.
(2) $\mathtt{h}$ è in $\mathcal{F}$ quando $\overline{\ } \circ \mathcal{U} \in \mathcal{F}$,
ovvero, quando $\overline{\ } : \mathcal{P} \to \mathcal{D} \in \mathcal{F}$ [1]
(3) Anche per $\overline{\ }$ possiamo trovare $\overline{\ } \in \mathcal{F}$, sebbene non tutte le iniettive $\overline{\ }$ lo siano.

Soluzione (b)
La funzione $\mathtt{h} \in \mathcal{F}$, calcola la Funzione $\overline{\ } \circ \mathcal{U} \in \mathcal{F}$

[1] $\circ$ è la composizione in $\mathcal{F}$, $(\mathtt{g} \circ \mathtt{h})\mathtt{x} \equiv \mathtt{h}(\mathtt{g}(\mathtt{x}))$, ed è **ovviamente** in $\mathcal{F}$ (come si vede quando componiamo, a formare un unico programma, il codice, output su input, di 2 distinti programmi)

Esercizi L2 - 10/3/2020

Esercizio (3)

La funzione π_1 sotto, è calcolabile anche se non sapremo mai, forse, trovare un procedimento effettivo per calcolarla.

$$\pi_1(n) = \begin{cases} 1 & \texttt{se l'espansione decimale di } \pi \texttt{ contiene n occorrenze di 1 in sequenza} \\ 0 & \texttt{altrimenti} \end{cases}$$

(a) Sapreste fornire una classe di programmi nei quali è certamente contenuto anche **un** *programma che calcola π_1?*
(b) Sapreste giustificare l'affermata calcolabilità di π_1?

Soluzione

Esercizi L2 - 10/3/2020

Esercizio (3 - Soluzione)

La funzione π_1 sotto, è calcolabile anche se non sapremo mai, forse, trovare un procedimento effettivo per calcolarla.

$$\pi_1(n) = \begin{cases} 1 & \texttt{se l'espansione decimale di } \pi \texttt{ contiene n occorrenze di 1 in sequenza} \\ 0 & \texttt{altrimenti} \end{cases}$$

(a) Sapreste fornire una classe di programmi nei quali è certamente contenuto anche **un** *programma che calcola π_1?*
(b) Sapreste giustificare l'affermata calcolabilità di π_1?

Soluzione
(a)
Si. La classe è composta da programmi $\mathtt{p_k}$, scritti in C (o qualunque altro **LP**), calcolanti funzioni $\mathtt{g_k}$ su interi non negativi, così definite:

$$\mathtt{g_k(n)} = \begin{cases} 1 & \texttt{se n} < \texttt{k} \\ 0 & \texttt{altrimenti} \end{cases}$$

Ognuna di queste funzioni è calcolabile ed è calcolata da uno di tali programmi. All'insieme $\{\mathtt{p_k} \mid \mathtt{k} \in \mathcal{N}\}$ aggiungiamo il programma `uno` che calcola la funzione $\mathtt{g(n)} = 1$ su ogni n non negativo.

(b)
Abbiamo visto in (a), che: $\pi_1 \in \mathtt{G} \equiv \{\mathtt{g_k} \mid \mathtt{k} \in \mathcal{N}\} \cup \{\mathtt{g}\}$. Ognuna di queste funzioni è banalmente esprimibile con un programma come in (a) (ed infatti, ha un andamento che la rende continua nella topologia di Dana Scott, vedi Lezione del ...). Non sappiamo ad oggi, quale di tali funzioni in $\mathtt{G}$ sia π_1 ma sappiamo che c'è quindi sappiamo che è calcolabile.
Quindi non conosciamo un algoritmo per calcolarla, Ma ogni linguaggio di Programmazione è im grado di:

:- Definirla
:- Calcolarla con la propria Macchina Astratta

Esercizi L2 - 10/3/2020

Esercizio (5)

Consideriamo la funzione π, così definita:

$$\pi(n) = \begin{cases} 1 & \texttt{se l'espansione decimale di } \pi \texttt{ contiene la sequenza delle cifre di n} \\ 0 & \texttt{altrimenti} \end{cases}$$

(a) Sapreste fornire una classe di programmi nei quali è certamente contenuto anche **un** *programma che calcola π?*
(b) Sapreste fornire argomenti per affermare, o per negare la calcolabilità di tale funzione?.

Esercizi L2 - 10/3/2020

Esercizio (5 - Soluzione)

Consideriamo la funzione π, così definita:

$$\pi(n) = \begin{cases} 1 & \texttt{se l'espansione decimale di } \pi \texttt{ contiene la sequenza delle cifre di n} \\ 0 & \texttt{altrimenti} \end{cases}$$

(a) Sapreste fornire una classe di programmi nei quali è certamente contenuto anche **un** *programma che calcola π?*
(b) Sapreste fornire argomenti per affermare, o per negare la calcolabilità di tale funzione?.

Soluzione

(a)

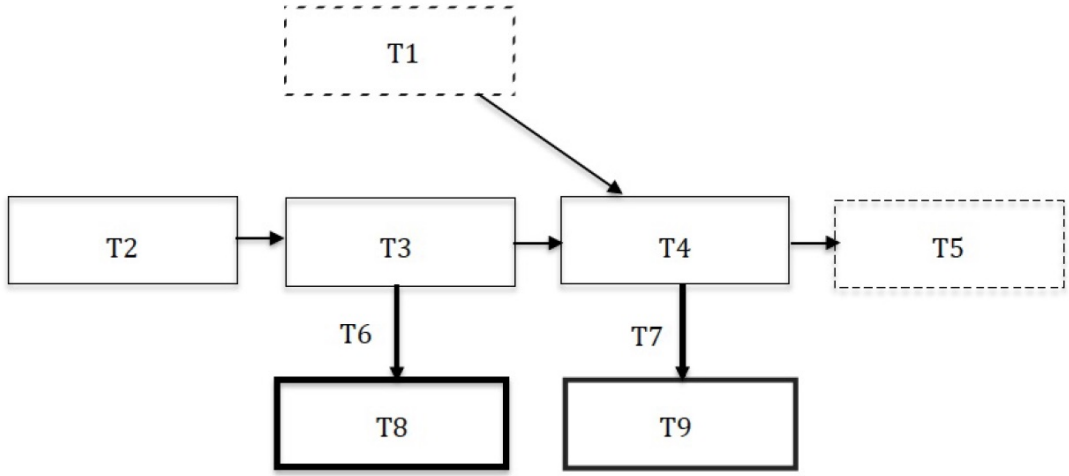
Ad oggi di questa funzione si sa ben poco, in particolare sul suo andamento. Taluni ipotizzano che l'espansione contenga ogni sequenza finita di cifre decimali: In questo caso la funzione calcola 1 su tutti gli interi non negativi. Al momento nessuno conosce una classe di programmi come quella richiesta (peraltro data invece, per la funzione π_1 in esercizio L2.3).
Possiamo al momento dire che: $\pi \in \mathcal{N} \to \mathcal{B}$, ovvero una qualunque funzione di decisione sui naturali (da naturali in booleani). Ma $\mathcal{N} \to \mathcal{B}$ non è numerabile.

(b)

Non abbiamo argomenti per affermare che la funzione π sia calcola nè il contrario. Le approssimazioni date per l'espansione di π ci offrono una funzione π con qualche milione di punti (più o meno isolati) che hanno immagine 1 e un'infinito di punti su cui ancora nessuno ha provato a calcolare l'occorrenza della relativa sequenza: Ad esempio: 25252525228 non è stato trovato nelle prime $2 * 10^9$ cifre decimali (vedi: //www.subidiom.com/pi/pi.asp). Ma oltre?

Esercizi L3 - 10/3/2020

3. Si consideri il diagramma di una AM, basata su compilazione, per il linguaggio $\mathcal{L}$.



Si associ alle didascalie nella lista sotto, la corretta etichetta $\mathtt{T_i}$ del diagramma.

1. **Compilatore da $\mathcal{L}$ a $\mathcal{L}_0$.** Etichetta: _____
2. **Dati di Input per $\mathcal{L}$.** Etichetta: _____
3. **Dati di Output per $\mathcal{L}$.** Etichetta: _____
4. **Esecuzione su `MA`.** Etichetta: _____
5. **Esecuzione su `M0`.** Etichetta: _____
6. **Macchina Astratta `MA`.** Etichetta: _____
7. **Macchina Ospite `M0`.** Etichetta: _____
8. **Programma $\mathtt{P} \in \mathcal{P}_{\mathcal{L}}$.** Etichetta: _____
9. **Programma $\mathtt{P}_0 \in \mathcal{P}_{\mathcal{L}_0}$.** Etichetta: _____

Esercizi L4 - 10/3/2020

Esercizio (da slide 6)

La derivazione ci fornisce una prova di appartenenza di una stringa al Linguaggio generato da una grammatica.
Infatti da $\mathcal{L}(\mathtt{G}) = \{\alpha \in \mathtt{T}^\star \mid \mathtt{S} \Rightarrow^\star \alpha\}$ *segue che* $\forall \alpha \in \mathtt{T}^\star, \alpha \in \mathcal{L}(\mathtt{G})$ `iff` $\mathtt{S} \Rightarrow^\star \alpha$.
Si dimostri che $\mathtt{I} * (\mathtt{I} + \mathtt{I}) \in \mathcal{L}(\mathtt{G})$.

$G = (\{\mathtt{E}\}, \{\mathtt{I}, +, *\}, \mathtt{E}, \mathtt{R})$

$R = \{\mathtt{E} \to \mathtt{E} + \mathtt{E}, \mathtt{E} \to \mathtt{E} * \mathtt{E}, \mathtt{E} \to \mathtt{I}, \mathtt{E} \to (\mathtt{E})\}$

Soluzione

Esercizi L4 - 10/3/2020

Esercizio (da slide 10)

Si considerino le tre derivazioni su stringhe data la grammatica G e derivanti la stessa stringa `I*I+I`.

$G = (\{\mathtt{E}\}, \{\mathtt{I}, +, *, (,)\}, \mathtt{E}, \mathtt{R})$

$R = \{\mathtt{E} \to \mathtt{E} + \mathtt{E}, \mathtt{E} \to \mathtt{E} * \mathtt{E}, \mathtt{E} \to \mathtt{I}, \mathtt{E} \to (\mathtt{E})\}$

$$\mathtt{E} \Rightarrow_2 \mathtt{E} * \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{E} \Rightarrow_1 \mathtt{I} * \mathtt{E} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{I} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{I} + \mathtt{I}$$

$$\mathtt{E} \Rightarrow_2 \mathtt{E} * \mathtt{E} \Rightarrow_1 \mathtt{E} * \mathtt{E} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{E} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{I} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{I} + \mathtt{I}$$

$$\mathtt{E} \Rightarrow_1 \mathtt{E} + \mathtt{E} \Rightarrow_2 \mathtt{E} * \mathtt{E} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{E} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{I} + \mathtt{E} \Rightarrow_3 \mathtt{I} * \mathtt{I} + \mathtt{I}$$

(a) Si calcolino le corrispondenti (uso di stessa produzione) derivazioni su parse tree e si mostri quali di queste derivazioni sono le stesse (generano lo stesso albero) e quali no.

(b) Si dica cosa significa e cosa comporta questa differenza.

Soluzione

Esercizi L4 - 10/3/2020

Esercizio (da slide 5)

Sia size la funzione $|\_| : (\mathtt{T} \cup \mathtt{NT})^* \to \mathcal{N}$ *che applicata ad una stringa di terminali e non terminali di una grammatica calcola il numero di terminali presenti in essa.*
Si dimostri che le grammatiche Context Free, hanno relazione `Rightarrow` *monotona rispetto a size, ovvero:*

$$\forall \mathtt{G} =< \mathtt{T}, \mathtt{NT}, \mathtt{S}, \mathtt{P} >,$$
$$\forall \gamma_1, \gamma_2 \in (\mathtt{T} \cup \mathtt{NT})^*, \quad \mathtt{Se}\ \gamma_1 \Rightarrow \gamma_2\ \mathtt{Allora}\ |\gamma_1| \leq |\gamma_2|$$

Soluzione

Esercizi L4 - 10/3/2020

Esercizio (da slide 5)

Si utilizzi la proprietà di monotonia, discussa sopra, ottenere la decidibilità dell'appartenenza ad $\mathcal{L}(\mathtt{G})$*, data una Context Free* $\mathtt{G}$*. In particolare, si dimostri che* $\mathtt{II} \notin \mathcal{L}(\mathtt{G})$ *allochè:*

$\mathtt{G} = (\{\mathtt{E}\}, \{\mathtt{I}, +, *\}, \mathtt{E}, \mathtt{R})$

$\mathtt{R} = \{\mathtt{E} \rightarrow \mathtt{E} + \mathtt{E}, \mathtt{E} \rightarrow \mathtt{E} * \mathtt{E}, \mathtt{E} \rightarrow \mathtt{I}, \mathtt{E} \rightarrow (\mathtt{E})\}$

Soluzione