Calcolabilità, Esecutori, Sintassi: Esercizi Vari

Sommario: Marzo 8-15, 2021

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

ESERCIZI L1 - 8/3/2021

Esercizio (1)

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

Soluzione

(a) Requisiti aggiuntivi

```
/*
Esercizio 1-24/2/2016.
Scrivere un programma C che calcoli la funzione "ordinamento di sequenza" implementando l'algoritmo
QuickSort. Allo scopo, si completi il testo con fgli eventuali vincoli che si ritengano necessari.
**
* Marco Bellia
**
Soluzione.
Vincoli.
1) Esistenza bound alla dimensione della sequenza da ordinare. Il bound è trattato come:
       Parametro di Programma
2) Elementi della sequenza di tipo noto. Il tipo è int.
3) Sequenza realizzata come struttura ad accesso diretto: array (ad allocazion) statica.
*/
```

(b) stesura del codice

...

Esercizi L2 - 8/3/2020

Esercizio (3)

La funzione π_1 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 allora, giustificare l'affermata calcolabilità di π_1?

Soluzione

Esercizio (3)

La funzione π_1 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 allora, giustificare l'affermata calcolabilità di π_1?

Soluzione

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

```
int main (){//program pk: k>=0
           //res = 1 sse 0<=n<=k
    int index = k; int res = 0; int n;
    scanf("%d",&n);
    if (n>0 && n<k) res = 1;
    printf("pk(%d) = %d", n, res);
    }
```

```
int main (){//program p:
            // res = 1 sse n>=0
    int res = 0; int n;
    scanf("%d",&n);
    if (n>0) res = 1;
    printf("pk(%d) = %d", n, res);
    }
```

Ognuna di queste funzioni è calcolabile ed è calcolata da uno di tali programmi. All'insieme $\{p_k \mid k \in \mathcal{N}\}$ aggiungiamo il programma p che calcola 1 su ogni n non negativo e .

(b)
Abbiamo visto in (a), che: $\pi_1 \in G \equiv \{g_k = [|p_k|] \mid k \in \mathcal{N}\} \cup \{g = [|p_k|]\}$, dove $[|\_|]$ sia la funzione meaning di C. Ognuna di queste funzioni è banalmente definita da un programma in (a) (inclusa la funzione π_1). Tuttavia, non sappiamo ad oggi, quale di tali funzioni sia π_1 ma sappiamo che 'e inclusa, quindi sappiamo che è calcolabile. Non ne conosciamo un algoritmo per calcolarla, Ma ogni linguaggio di Programmazione è in grado di:

:- Definirla e,
:- Calcolarla con la propria Macchina Astratta

Esercizi L2 - 8/3/2021

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?.

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

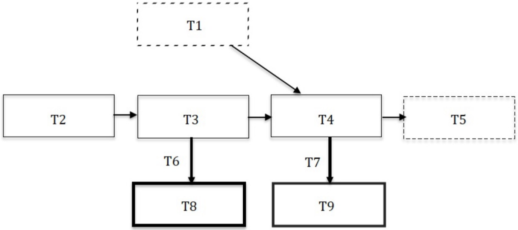
(a)

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, risulterebbe banalmente espressa da un programma e quindi calcolabile.
Al momento nessuno conosce una classe di programmi che contenga un programma per π.
Possiamo dire che: $\pi \in \mathcal{N} \rightarrow \mathcal{B}$, ovvero una qualunque funzione di decisione sui naturali. Ma $\mathcal{N} \rightarrow \mathcal{B}$ è non numerabile.

(b)

Non abbiamo argomenti per affermare che la funzione π sia calcolabile 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?

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

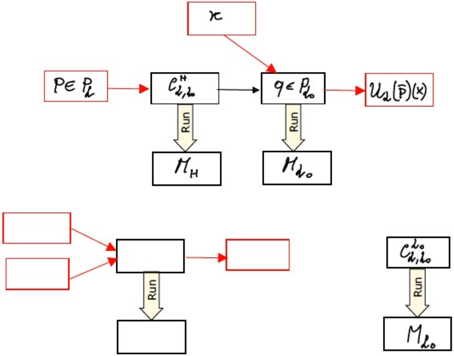


Si associ alle didascalie nella lista sotto, la corretta etichetta $\mathtt{T_i}$ del diagramma.

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** `MO`. Etichetta: ________
6. **Macchina Astratta** `MA`. Etichetta: ________
7. **Macchina Ospite** `MO`. Etichetta: ________
8. **Programma** $\mathtt{P} \in \mathcal{P}_{\mathcal{L}}$. Etichetta: ________
9. **Programma** $\mathtt{P_0} \in \mathcal{P}_{\mathcal{L}_0}$. Etichetta: ________

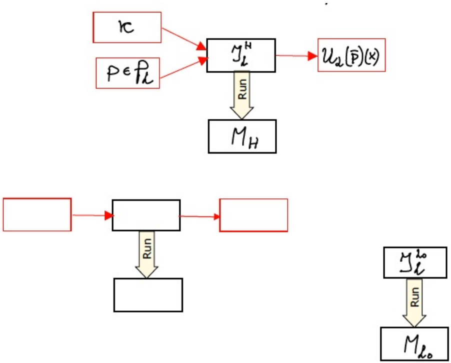
Esercizio

Costruzione di un Compilatore $\mathcal{C}_{\mathcal{L},\mathcal{L}_0}^{\mathcal{L}_0}$ quando $\mathcal{L} > \mathcal{H} \gg \mathcal{L}_0$. Sotto una tipica struttura per realizzare una AM basato su compilazione: La si completi indicando: a) Le entità da inserire nei box vuoti, e b) In quale ordine i diagrammi devono essere realizzati.



Esercizio

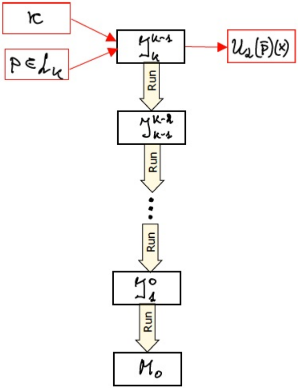
Costruzione di un Interprete $\mathcal{I}_{\mathcal{L}}^{\mathcal{L}_0}$ quando $\mathcal{L} > \mathcal{H} \gg \mathcal{L}_0$. In questo caso la costruzione di un interprete non può essere condotta utilizzando il Linguaggio oggetto perchè $\mathcal{L}_0$ è troppo a basso livello rispetto a $\mathcal{L}$.
Sotto una struttura per realizzare una AM basato su Interpretazione quando il Linguaggio oggetto é un Linguaggio Low-Level. Si chiede di completare indicando: a) Le entità da inserire nei box vuoti; b) In quale ordine i diagrammi devono essere realizzati.



Esercizi L3 - 8/3/2021

Esercizio

Una cascata di Macchine Astratte $\mathcal{M}_{\mathtt{i}}$ realizzate con Interpreti $\mathcal{I}_{\mathtt{i}}^{\mathtt{i}-1}$, per $0 < \mathtt{i} <= \mathtt{k}$, per Linguaggi $\mathcal{L}_i$, a partire dalla Macchina Concreta $\mathcal{M}_0$.
Si dica come deve operare il Sistema Operativo perchè l'esecuzione del programma $\mathtt{p}$ del Linguaggio $\mathcal{L}_{\mathtt{k}}$ con input $\mathtt{x}$ avvenga come atteso.



Esercizi L4 - 15/3/2021

Esercizio (da slide 6)

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

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

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

Soluzione

Esercizio (da slide 10)

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

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

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

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

(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

Esercizio (da slide 5)

Sia size la funzione $|\_| : (\mathtt{T} \cup \mathtt{NT})^* \rightarrow \mathcal{N}$ *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:*

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

Soluzione

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:*

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

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

Soluzione

Esercizi L4 - 15/3/2021

Esercizio (da slide 5)

Si dimostri che le grammatiche Context Free, hanno PERSISTENZA*, ovvero la relazione* $\Rightarrow$ *ha la seguente proprietà:*

$$\forall G = < T, NT, S, P > \in \mathcal{G}^{\mathcal{F}},$$
$$\forall \gamma_1, \gamma_2 \in (T \cup NT)^*,$$

Se

$$\gamma_1 \equiv \alpha_0\, u1 \ldots \alpha_{k-1}\, uk\, \alpha_k \quad (\text{per}: \ \alpha_i \in NT^*, ui \in T) \text{ And}$$
$$\gamma_1 \Rightarrow^* \gamma_2$$

Allora

$$\gamma_2 \equiv \alpha'_0\, u1 \ldots \alpha'_{k-1}\, uk\, \alpha'_k \quad (\text{con}: \ \alpha'_i \in (T \cup NT)^*)$$

Soluzione