Esercizi L3 (AM ed Esecutori) 6/3 - 13/3 2018

Sommario: 6 marzo, 2018

1. Si forniscano 3 esempi, in contesti diversi, di macchine astratte.
2. In cosa differisce un L. di Programmazione Low-Level da uno Hight-Level?
3. Si elenchino le trasformazioni presenti nel diagramma di una AM, basata su compilazione, di linguaggio $\mathcal{L}$ in $\mathcal{L}_o$ utilizzando $\mathcal{L}_a$.
4. Si consideri il diagramma di una AM, basata su compilazione. Si dica:
   (a) Quante AM sono presenti nel diagramma;
   (b) Per ciascuna AM si specifichi Input ed Output
5. Si consideri il diagramma di una AM, basata su interpretazione. Si dica:
   (a) Quante AM sono presenti nel diagramma;
   (b) Per ciascuna AM si specifichi Input ed Output
6. Si consideri il diagramma di una AM, basata su compilazione. Si dica:
   (a) Se e quando la macchina MA può essere rimossa;
   (b) Quando rimovibile, si mostri il diagramma ottenuto rimuovendo MA
7. Si consideri il diagramma di una AM, basata su realizzazione Mista.
   (a) Si mostri il diagramma quando il linguaggio intermedio $\mathcal{L}_i = \mathcal{L}_0$;
   (b) Sia $\mathcal{L} \neq \mathcal{L}_i$ e $\mathcal{L}_i \neq \mathcal{L}_0$. Si dica se e quando MA può essere rimossa;
   (c) Sia $\mathcal{L} \neq \mathcal{L}_i$ e $\mathcal{L}_i \neq \mathcal{L}_0$. Si dica quante AM sono presenti;
   (d) Nel caso MA sia rimovibile, si mostri il diagramma ottenuto rimuovendola

Esercizi L3 (AM ed Esecutori) 6/3 - 13/3 2018 /2

Sommario: 6 marzo, 2018

8 Nel compilatore la decodifica è fatta una volta per tutte le esecuzioni del progamma. Nell'interprete la decodifica richiede di essere ripetuta in 2 punti:
(a) Quali?
(b) Si giustifichi l'affermazione sopra, ricorrendo ad un esempio.

9 Compilatore ed Interprete sono due strumenti complementari per realizzare una AM di un L. di Programmazione ed eseguirne i programmi su dati di input diversi ovvero, stati iniziali diversi. Si dia una tabella che mostri questa complementarietà sugli aspetti trattati a lezione nel confrontare i due approcci.

10 Nella costruzione di un compilatore $\mathcal{C}_{\mathcal{L},\mathcal{L}_O}$ utilizziamo i mappings $\mathcal{U}_{\mathcal{L}}$, $\mathcal{U}_{\mathcal{L}_O}$, in modo da creare un diagramma, come quello sotto, che commuta definendo $\mathcal{C}_{\mathcal{L},\mathcal{L}_O}$. Sapreste mettere i mapping nelle giuste freccie?

11 (a) Cos'è una Macchina Virtuale e cosa rende il ricorso alle MV interessante per la Trasportabilità del codice oggetto di un L. di programmazione?
(b) Sapreste indicare qualche altro vantaggio?

12 (a) Cos'è il Front-End di un compilatore o di un interprete?
(b) Perchè Il Front-End del compilatore e dell'interprete di uno stesso L. di Programmazione potrebbe quasi coincidere?

13 (a) Cos'è il Front-End di un Compilatore?
(b) E quello di un Interprete?
(c) Sapreste indicare un meccanismo del Linguaggio C (che potrebbe essere) gestito (da librerie dello) RTS del compilatore?

Esercizi L3 (AM ed Esecutori) 6/3 - 13/3 2018 /3

Sommario: 6 marzo, 2018

14 Si elenchino i fatti che abbiamo visto a lezione e che permettono di sostenere:
"Un calcolatore ha capacità di autorappresentazione e ciò lo rende uno strumento dalle potenzialità teoriche e pratiche ancora da comprendere ed esplorare"
certamente ben diverso da un elettrodomestico, indipendentemente dai materiali costruttivi, dall'uso e dalla percezione che taluni ne hanno.

Esercizio (1)

Si forniscano 3 esempi, in contesti diversi, di macchine astratte

Soluzione

1. bare machine (i.e. macchina cruda priva di S. Operativo)
2. macchina di S.O. (ottenuta dopo l'istallazione di S.O.)
3. macchina Server di posta elettronica (ed.es. Exchange

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (2)

In cosa differisce un Linguaggio di Programmazione Low Level da uno Hight Level? In particolare, con l'aiuto di una tabella, si dica quali priorità condizionano la definizione dell'una rispetto all'altra classe.

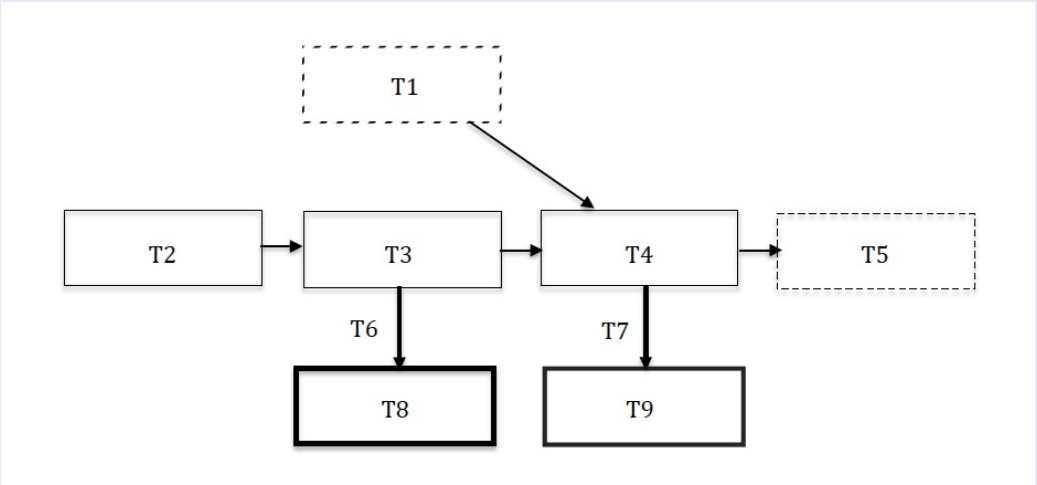
Soluzione
Differiscono nell'orientamento generale: Il Linguaggio Low-Level è pensato per fornire un linguaggio con cui equipaggiare (interagire e controllare) un esecutore per le funzioni calcolabili. Il linguaggio Hight-Level è pensato per il supporto di metodologie di sviluppo di programmi specifiche per la programmazione in una data area di applicazioni, con specifiche dinamiche di modificabilità e riuso del codice e verifica della corretta del sistema prodotto. Tutto questo è mostrato nella tabella sotto.

	Low-Level	Hight-Level
Realizzazione AM	+Hardware	+Software
Ciclo Interprete	Semplice: 1 Ciclo	Complesso: k+1 Cicli
Dati e Operazioni	Primitivi, Non Estendibili	User Defined
Metodologie di Programmazione	NO	Varie Orientate Applicazioni
Supporto per Sviluppo, Modifica e Riuso, Correttezza	NO NO NO	Si Talora Talora

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (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* $\mathcal{L}_0$
2. *Dati di Input per* $\mathcal{L}$
3. *Dati di Output per* $\mathcal{L}$
4. *Esecuzione su* `MA`
5. *Esecuzione su* `M0`
6. *Macchina Astratta* `MA`
7. *Macchina Ospite* `M0`
8. *Programma* $\mathtt{P} \in \mathcal{P}_{\mathcal{L}}$
9. *Programma* $\mathtt{P_0} \in \mathcal{P}_{\mathcal{L}_0}$

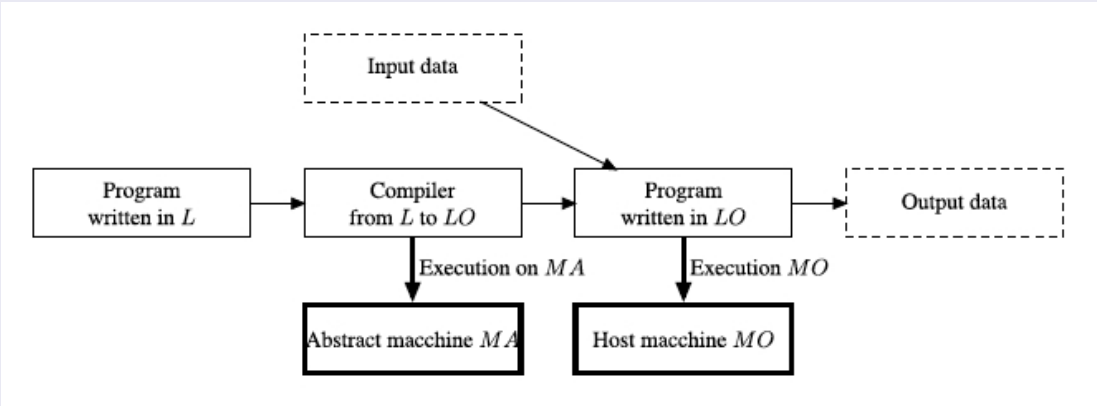
Soluzione

1 = ; 2 = ; 3 = ; 4 = ; 5 = ;
6 = ; 7 = ; 8 = ; 9 = ;

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (4)

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



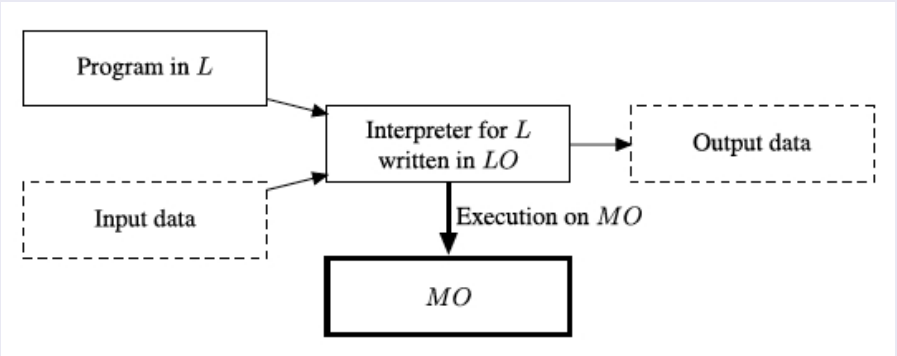
Si dica:
(a) Quante AM sono presenti nel diagramma;
(b) Si indichino Input e Output di ogni AM presente;

Soluzione
(a)
(b)

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (5)

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



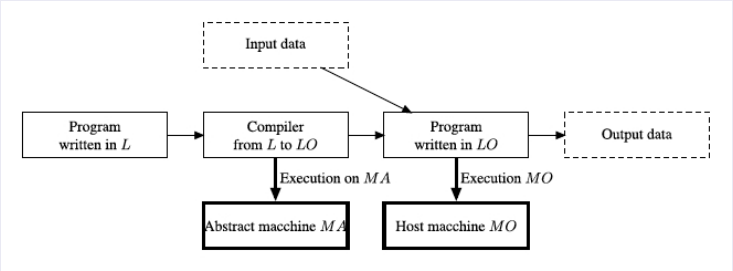
Si dica:
(a) Quante AM sono presenti nel diagramma;
(b) Si indichino Input e Output di ogni AM presente;

Soluzione
(a)
(b)

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (6)

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



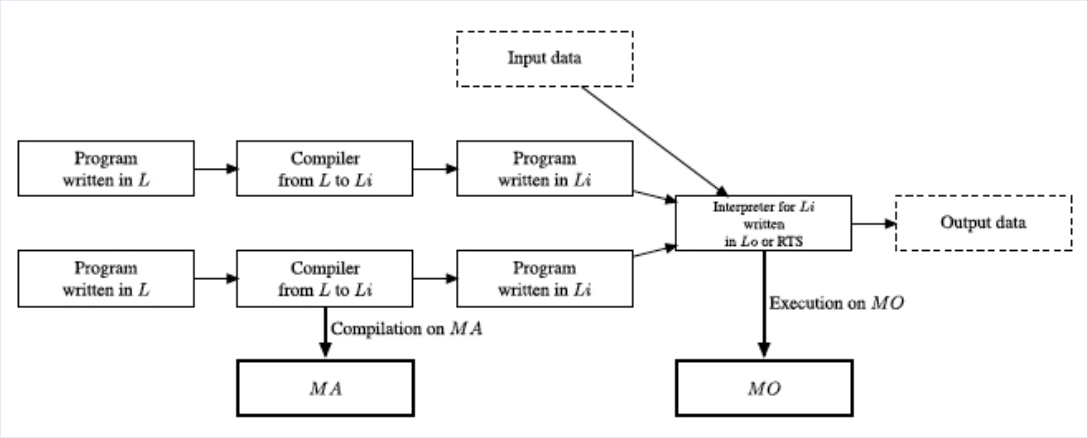
Si dica:
(a) Se e quando la macchina MA può essere rimossa;
(b) Nel caso sia rimovibile, si mostri il diagramma ottenuto rimuovendola

Soluzione
(a)
(b)

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (7)

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



(a) Si mostri il diagramma quando il linguaggio intermedio $\mathcal{L}_{\mathtt{i}} = \mathcal{L}_0$;
(b) Sia $\mathcal{L} \neq \mathcal{L}_{\mathtt{i}}$ e $\mathcal{L}_{\mathtt{i}} \neq \mathcal{L}_0$. Si dica se e quando la macchina MA può essere rimossa;
(c) Sia $\mathcal{L} \neq \mathcal{L}_{\mathtt{i}}$ e $\mathcal{L}_{\mathtt{i}} \neq \mathcal{L}_0$. Si dica quante AM sono presenti;
(d) Nel caso MA sia rimovibile, si mostri il diagramma ottenuto rimuovendola

Soluzione

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (8)

Nel compilatore la decodifica è fatta una volta per tutte le esecuzioni. Nell'interprete la decodifica richiede di essere ripetuta in 2 punti:
(a) Quali?
(b) Si giustifichi l'affermazione sopra, ricorrendo ad un esempio.

Soluzione

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (9)

Compilatore ed Interprete sono due strumenti complementari per realizzare una AM di un L. di Programmazione ed eseguirne i programmi su dati di input diversi ovvero, stati iniziali diversi. Si dia una tabella che mostri questa complementarità sugli aspetti trattati a lezione nel confrontare i due approcci.

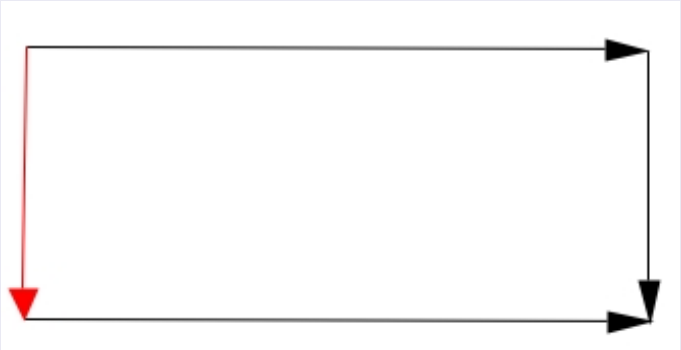
Soluzione
Dobbiamo completare la tabella sotto.

	Interprete	Compilatore
Efficienza	$\downarrow$: bassa (decodifica a run time)	$\uparrow$: alta (decodifica una volta sola e staticamente)
Occ. Memoria	$\uparrow$: Sorgente	$\downarrow$: contiene anche la decodifica
Costruzione	$\updownarrow$: Rapid Prototyping	$\updownarrow$: Condividono Front-End, RTS
Flessibilità Programmazione	NO	Varie Orientate Applicazioni
Supporto per Sviluppo, Modifica e Riuso, Correttezza	NO NO NO	Si Talora Talora

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (10)

Nella costruzione di un compilatore $\mathcal{C}_{\mathcal{L},\mathcal{L}_O}$ utilizziamo i mappings $\mathcal{U}_{\mathcal{L}}$, $\mathcal{U}_{\mathcal{L}_O}$, in modo da creare un diagramma, come quello sotto, che commuta definendo $\mathcal{C}_{\mathcal{L},\mathcal{L}_O}$. Sapreste mettere i mapping nelle giuste freccie?



Soluzione

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (11)

(a) Cos'è una Macchina Virtuale e cosa rende il ricorso alle MV interessante per la Trasportabilità del codice oggetto di un L. di programmazione?
(b) Sapreste indicare qualche altro vantaggio?

Soluzione

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (12)

(a) Cos'è il Front-End di un compilatore o di un interprete?
(b) Perchè Il Front-End del compilatore e dell'interprete di uno stesso L. di Programmazione potrebbe quasi coincidere?

Soluzione

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (13)

(a) Cos'è il Front-End di un Compilatore?
(b) e quello di un Interprete?
(c) Sapreste indicare un meccanismo del Linguaggio C (che potrebbe essere) gestito (da librerie del) RTS del compilatore?

Soluzione

Esercizi L3 del 6/3 - 13/3 2018

Esercizio (14)

Si elenchino i fatti che abbiamo visto a lezione e che permettono di sostenere:
"Un calcolatore ha capacità di autorappresentazione e ciò lo rende uno strumento dalle potenzialità teoriche e pratiche ancora da comprendere ed esplorare"
certamente ben diverso da un elettrodomestico, indipendentemente dai materiali costruttivi, dall'uso e dalla percezione che taluni ne hanno.

Soluzione
Vediamo i fatti da elencare. Primo e fondamentale fatto è la natura di un calcolatore.
1) Un calcolatore è un L. di Programmazione $\mathcal{L}$ insieme al relativo esecutore.
Che $\mathcal{L}$ sia poco espressivo (tedioso da usare) e che il calcolatore abbia più o meno numerose devices per l'interazione uomo-macchina e/o che l'interazione sia meglio gestita da un Sistema Operativo, sono tutte ragioni che hanno portato al diffuso impiego e alla percezione che si ha oggi, del calcolatore senza però cambiarne la natura. La capacità di autorappresentazione di un calcolatore allora deriva dall'esistenza dell' universale $\mathcal{U} \in \mathcal{F} \equiv [\mathcal{D} \to \mathcal{D}]$. A livello astratto, $\mathcal{U}$ è una funzione calcolabile che applicata ad una descrizione $\overline{p} \in \mathcal{D}$ di una qualunque funzione calcolabile g $\in [\mathcal{D} \to \mathcal{D}]$, i.e. g $\equiv [|\mathrm{p}|]$, si comporta come la funzione g, ovvero $\mathcal{U}(\overline{\mathrm{p}})$ è g. A livello concreto, questa funzione si presenta come un interprete di $\mathcal{L}$ scritto in $\mathcal{L}$ stesso, ovvero un programma di $\mathcal{L}$, equipaggiato con strutture dati e codice di $\mathcal{L}$ per descrivere l'esecutore di $\mathcal{L}$ stesso, ovvero struttura della memoria, rappresentazione dei programmi ed dei dati manipolati, decodifica, interpretazione e supporto al controllo dell'esecuzione dei costrutti di $\mathcal{L}$.

L'esistenza della funzione $\mathcal{U}$ si deve A. Turing che ne diede una descrizione nel caso del suo formalismo (Macchine di Turing) utilizzando il suo formalismo stesso. Un'altra descrizione di $\mathcal{U}$ è stata data da John McCarthy in Lisp per il Lisp. L'interprete di McCarthy però aggiunge nuovi aspetti. In aggiunta all'autorappresentazione, vediamo la reflection che permette l'introspezione della computazione e la modifica del codice esplorando aspetti molto più complessi di calcolabilità da un lato, e conducendo oggi a tecniche di programmazione quali quelle impiegate nei sistemi evolutivi e ad autoapprendimento.