

ESERCIZIO1

PREMESSA

Per risolvere problemi spesso esistono delle regole che, dai dati del problema, permettono di calcolare o *dedurre* la soluzione. Questa situazione si può descrivere col termine regola(<sigla>,<lista antecedenti>,<conseguente>)

che indica una regola di nome <sigla> che consente di dedurre <conseguente> conoscendo tutti gli elementi contenuti nella <lista antecedenti>, detta anche *premessa*. Problemi “facili” possono essere risolti con una sola regola; per problemi “difficili” una sola regola non basta a risolverli, ma occorre applicarne diverse in successione.

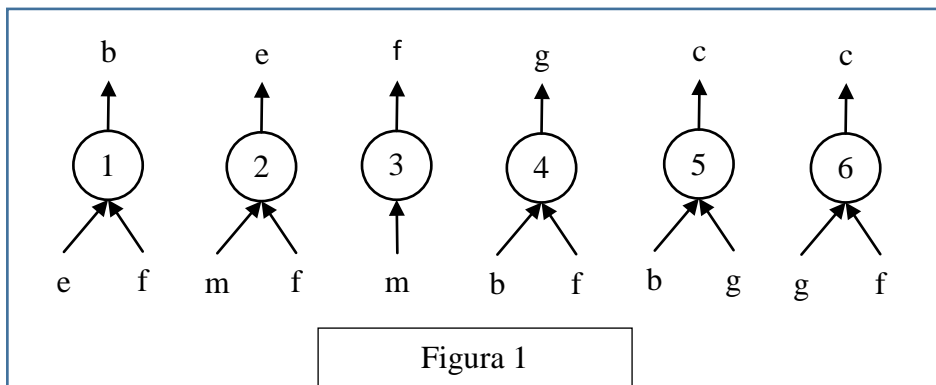
Si considerino le seguenti regole:

- regola(1,[e,f],b) regola(2,[m,f],e) regola(3,[m],f)
- regola(4,[b,f],g) regola(5,[b,g],c) regola(6,[g,f],c)

Per esempio la regola 1 dice che si può calcolare (o dedurre) **b** conoscendo **e** ed **f** (cioè gli elementi della lista [e,f]); conoscendo **b** ed **f** (cioè gli elementi della lista [b,f]) è possibile dedurre **g** con la regola 4. Quindi, a partire da **e** ed **f** è possibile dedurre prima **b** (con la regola 1) e poi **g** (con la regola 4).

Un *procedimento di deduzione* (o deduttivo, o di calcolo) è rappresentato da un *insieme di regole da applicare in sequenza opportuna* per dedurre un certo elemento (incognito) a partire da certi dati: quindi può essere descritto dalla lista delle sigle di queste regole. Il procedimento [1,4] descrive la soluzione del problema: “dedurre **g** a partire da **e** ed **f**”.

Una maniera grafica per rappresentare le regole è quella mostrata nella seguente figura 1: consiste nell’associare un albero (rovesciato) ad ogni regola: la radice (in alto) è il conseguente, le foglie (in basso) sono gli antecedenti.

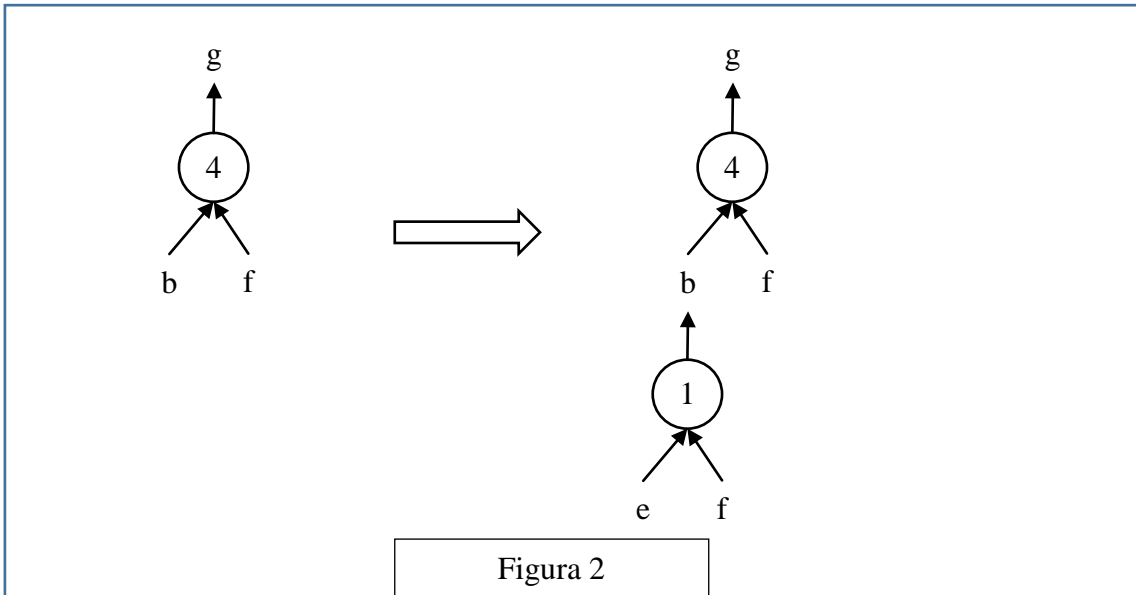


Con questa rappresentazione grafica, risolvere il problema “dedurre **g** a partire da **e** ed **f**” è particolarmente facile; si cerca un “albero” (cioè una regola) che ha come radice l’incognita (cioè **g**): in questo caso ne esiste solo uno che è la regola 4: si veda la figura 2 a sinistra.

Le foglie di questo albero (**b** ed **f**) *non* sono tutte note: quelle note (**f** in questo caso) sono vere e proprie foglie, quelle incognite (**b** in questo caso) vanno considerati come “anelli” a cui “appendere” un altro albero; quindi bisogna continuare *sviluppendo* la foglia incognita **b**, cioè “appendendo” a **b** l’albero rappresentato dalla regola 1, come illustrato nella figura 2 a destra.

Adesso tutte le foglie dell’albero così ottenuto (**e** ed **f**) sono note e il problema è risolto.

Si può anche dire che un albero le cui foglie sono tutte note rappresenta un procedimento per dedurre la “radice” a partire dalle “foglie”. Per costruire la lista corrispondente occorre *partire dal basso*: prima si applica la regola 1, che utilizza solo i dati; poi si può applicare la regola 4. Il procedimento è quindi (individuato dalla lista) [1,4].

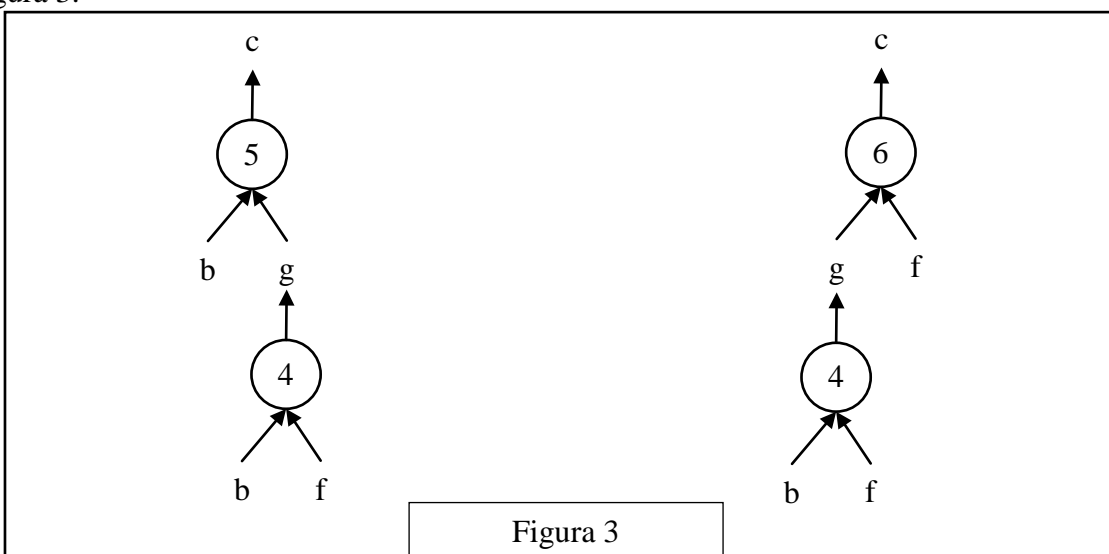


N.B. Nella lista non ci sono regole *ripetute*: infatti un procedimento di deduzione è un *insieme* di regole da applicare in opportuna sequenza. L'applicazione di una regola rende disponibile il conseguente da utilizzare (come antecedente) nell'applicazione di regole successive.

Nelle liste richieste le sigle delle regole sono elencate nell'ordine che corrisponde alla sequenza di applicazione: la prima (a sinistra) della lista deve essere la sigla che corrisponde alla prima regola da applicare (che ha come antecedenti solo dati); l'ultima (a destra) deve essere la sigla della regola che ha come conseguente l'elemento incognito da dedurre.

Per rendere unica la lista associata a un ben preciso procedimento (cioè a un ben preciso insieme di regole), si costruisce tale lista per passi successivi a partire dal primo elemento che è la sigla della prima regola da applicare; ad ogni passo, se ci fossero più regole applicabili, *per quel procedimento*, occorre dare la precedenza (nella lista) alla regola con sigla *inferiore*.

N.B. In alcuni casi esistono più procedimenti deduttivi possibili che permettono di ricavare un certo elemento dagli stessi dati in maniere diverse (cioè con *insiemi diversi di regole* e quindi con alberi diversi). Per esempio il problema “dedurre **c** a partire da **b** ed **f**” (dalle regole viste sopra) ha due distinti procedimenti risolutivi; gli alberi relativi ai due procedimenti sono mostrati nella seguente figura 3.



ESERCIZIO 4

PROBLEMA

Si consideri la seguente procedura PROVA1.

```

procedure PROVA1;
variables A, B, C, K integer;
input A, B;
for K = 1 to 8 step 1 do
    if A < B then C ← A; output C, K; input A; input B; endif;
    if A > B then C ← B; output C, K; input A; input B; endif;
endfor;
endprocedure;
    
```

I valori di input per A sono: 4, 5, 4, 3, 6, 1, 1, ... e per B sono: 5, 6, 2, 7, 5, 1, 1, ...

N.B. I valori in input per entrambe le variabili sono uguale a 1 dopo i primi 5.

Determinare la lista LC che contiene nell'ordine i valori di output di C e la lista LK che contiene nell'ordine i valori in output per K.

LC	[]
LK	[]

ESERCIZIO 5

PROBLEMA

Si consideri la seguente procedura PROVA2.

N.B. Le variabili di tipo float hanno valore razionale; le costanti di tipo float si scrivono col punto e almeno una cifra decimale. Il calcolo delle espressioni float deve essere fatto con (almeno) 6 cifre decimali (dopo la virgola) esatte; in una espressione float possono comparire variabili intere.

```

procedure PROVA2;
variables K integer; A, B, C float;
A ← 0.0;
B ← 1.0;
K ← 0;
while A<B do
    K ← K +1;
    C ← A;
    A ← A + B × K / 20.0;
    B ← B - C × K / 20.0;
endwhile;
output K;
endprocedure;
    
```

Determinare il valore di output per K..

K	
---	--

ESERCIZIO 6

PROBLEMA

Si consideri la seguente procedura PROVA3.

N.B. Le variabili di tipo float hanno valore razionale; le costanti di tipo float si scrivono col punto e almeno una cifra decimale. Il calcolo delle espressioni float deve essere fatto con (almeno) 6 cifre decimali (dopo la virgola) esatte; in una espressione float possono comparire variabili intere.

```

procedure PROVA3;
variables K integer; A, B float;
input A;
input B;
K ← 0;
while A>B do
    K ← K +1;
    A ← A - 1.0/K;
endwhile;
output K;
endprocedure;
    
```

Se il valore di input per A è 3.0 e quello per B è 0.1, calcolare il valore di output per K.

K	
---	--

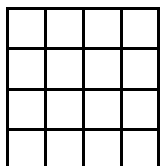
ESERCIZIO 7

PROBLEMA

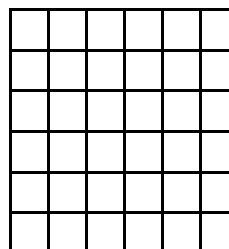
Considerare con attenzione le seguenti tre figure.



A1



A2



A3

Calcolare quanti quadrati, Q1, Q2, Q3 compaiono (in tutto) rispettivamente in A1, A2 e A3.

Q1	
Q2	
Q3	

ESERCIZIO 8

PROBLEMA

In the first tournament of a problem solving (individual) competition, the average score of a group of students (some coming from school A and some from school B) was 80 (over a maximum of 100). If the students from school A averaged 75 and the students from school B averaged 90, what is the ratio RAB of the number of students from A to the number of students from B?

In the next tournament, students from school C joined; the average total score was 85, and students from C scored an average of 95, while students from A and B maintained their overall score. What is the ratio RCB of the number of newcomers from the school C to the number of students from B?

Put your answers as number with two decimal digit (and “.” as decimal mark) in the boxes below.

RAB	
RCB	