

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.

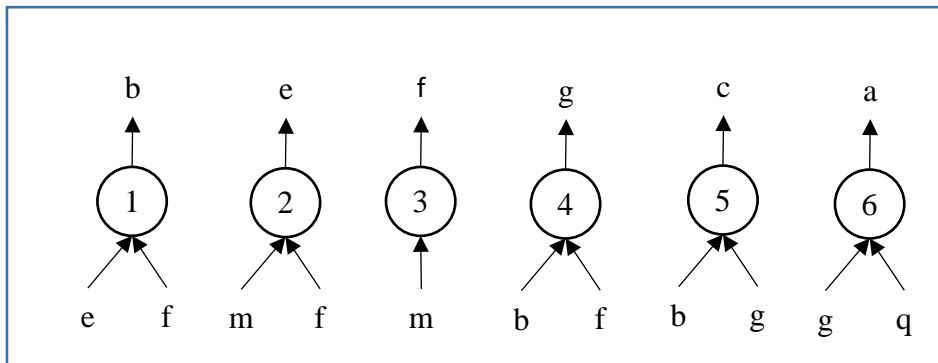
Si considerino le seguenti regole:

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,q],a)

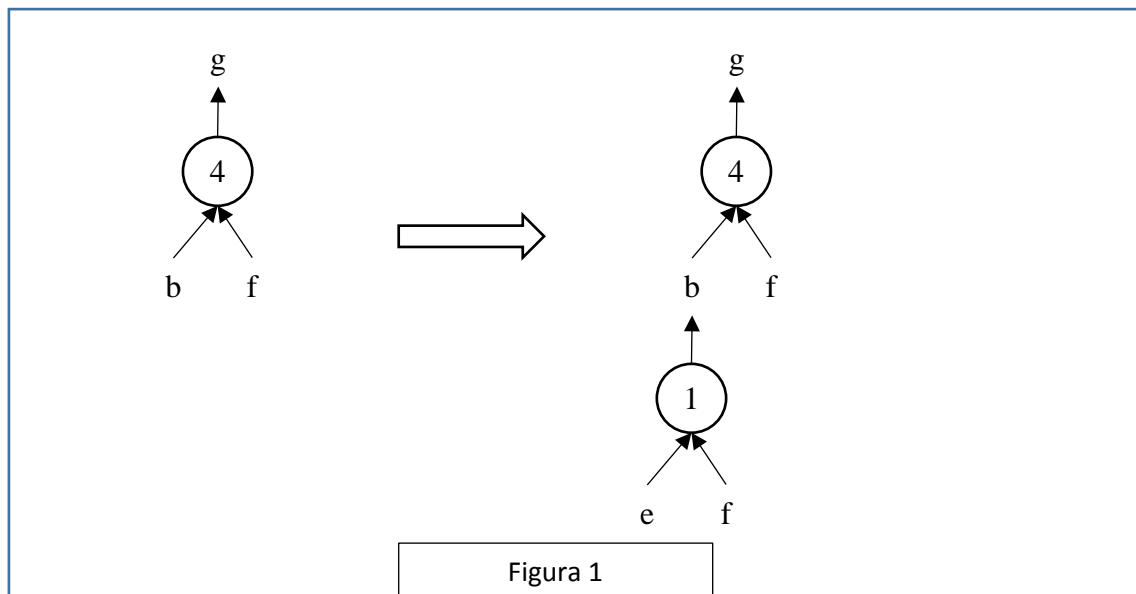
Per esempio la regola 1 dice che si può calcolare (o dedurre) **b** conoscendo **e** ed **f** (cioè gli elementi della lista [e,f]); conoscendo **b** ed **f** (cioè gli elementi della lista [b,f]) è possibile dedurre **g** con la regola 4. Quindi, a partire da **e** ed **f** è possibile dedurre prima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

Un *procedimento di deduzione* (o di calcolo) è rappresentato da un elenco di regole da applicare e 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: consiste nell’associare un albero (rovesciato) ad ogni regola: la radice (in alto) è il conseguente, le foglie (in basso) sono gli antecedenti.



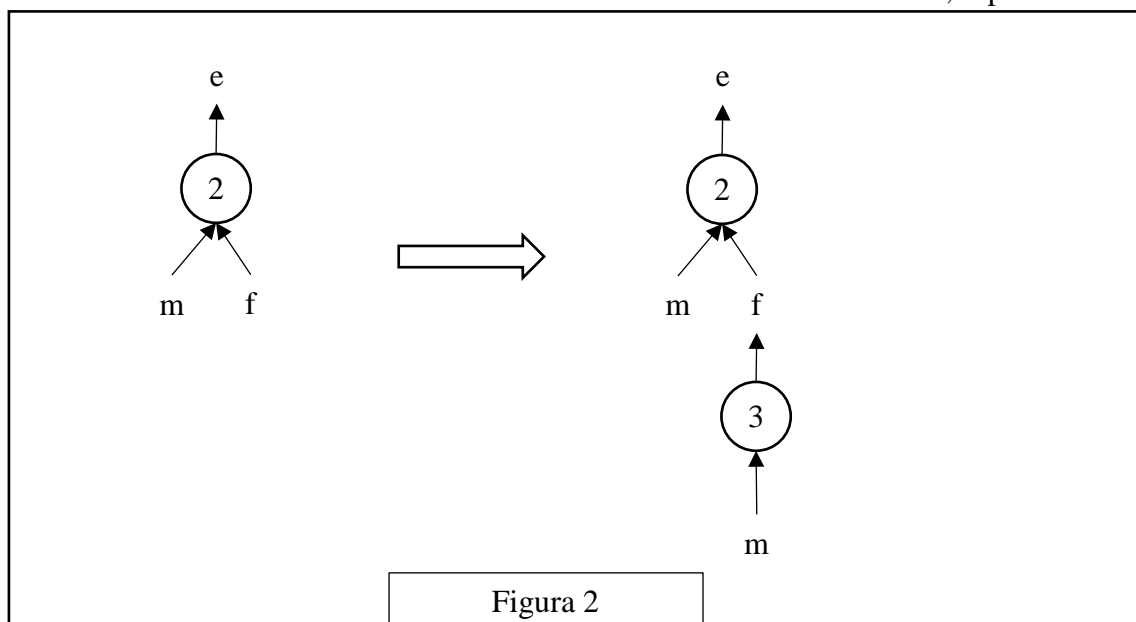
Con questa maniera grafica risolvere il problema “dedurre **g** a partire da **e** ed **f**” è particolarmente facile; si cerca un “albero” (cioè una regola) che ha come radice l’incognita (cioè **g**): in questo caso ne esiste solo uno che è la regola 4: si veda la seguente figura 1 a sinistra.



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 *sviluppando* la foglia incognita **b**, cioè “appendendo” a **b** l’albero rappresentato dalla regola 1, come illustrato nella figura 1 a destra.

Adesso tutte le foglie dell’albero così ottenuto (**e** ed **f**) sono note e il problema è risolto. Per costruire la lista 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 [1,4].

Come altro esempio, in figura 2 è illustrata la soluzione del problema: “dedurre **e** a partire da **m**”. Tale soluzione si ottiene costruendo successivamente i due alberi mostrati; il procedimento è [3,2].



N.B. Nelle liste richieste occorre elencare le sigle delle regole 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 che ha come conseguente l’elemento incognito da dedurre richiesto dal problema.



In ogni procedimento di deduzione, l'applicazione di una regola rende disponibile il conseguente da utilizzare (come antecedente) per poter applicare regole successive: la prima regola è sempre applicabile a partire *solo* dai dati e non ci sono regole *ripetute*.

Inoltre, ad ogni passo del procedimento, se ci fossero più regole applicabili contemporaneamente, nella lista occorre dare la precedenza a quella con sigla inferiore.

PROBLEMA

Sono date le seguenti regole:

regola(1,[r,u],x)	regola(2,[f,q],g)	regola(3,[p,r],u)
regola(4,[a,f],c)	regola(5,[d],f)	regola(6,[u,x],a)
regola(7,[n,g],p)	regola(8,[p,g],d)	regola(9,[y,w],d)
regola(10,[f,d],a)	regola(11,[f],q)	regola(12,[f,g],n)

Trovare:

1. la lista L1 che descrive il procedimento per dedurre **a** conoscendo **p** e **r**;
2. la lista L2 che descrive il procedimento per dedurre **c** conoscendo **y** e **w**;
3. la lista L3 che descrive il procedimento per dedurre **d** conoscendo **f**.

L1	[]
L2	[]
L3	[]



ESERCIZIO 3

PROBLEMA

You are given 16 numbers; the first 7 numbers have an average of 6.00; 6 numbers (distinct from the before mentioned ones) have an average of 7.00; the last 3 numbers have an average of 20.00.

What is the average of all 16 numbers?

Put your answer in the box below as a decimal number with two decimal places; use a dot as decimal mark.

ESERCIZIO 4

PROBLEMA

Si consideri la seguente procedura PROVA1.

```
procedure PROVA1;  
variables A, B, K, J integer;  
A ← 0;  
B ← 0;  
for J from 1 to 4 step 1 do  
  for K from 1 to 5 step 1 do  
    A ← A + J;  
    B ← B + K;  
  endfor;  
endfor;  
output A,B;  
endprocedure;
```

Determinare il valore di output di A e B.

A	
B	

ESERCIZIO 5

PREMESSA

La ripetizione di un gruppo di azioni può essere comandata non solo con la struttura “for” già vista, ma anche con la struttura “while”, illustrata dal seguente esempio.

```

B ← 10;
A ← 0;
K ← 0;
while A < B do
    K ← K + 1;
    A ← K × K + A;
endwhile;
output A;
    
```

Se il predicato $A < B$ è vero, il ciclo viene ripetuto; quando diventa falso si passa alla esecuzione della istruzione successiva a “endwhile”. In questo caso il valore di B rimane fisso a 10, mentre quello di A cambia dopo ogni iterazione assumendo i seguenti valori: 1, 5, 14. Dopo la terza iterazione il valore di A non è più minore di quello di B e il ciclo si arresta: in output si ha quindi 14.

PROBLEMA

Si consideri la seguente procedura PROVA2.

```

procedure PROVA2;
variables A, B, K, Z integer;
A ← 1;
B ← 0;
K ← 0;
input Z;
while A > B do
    K ← K + 1;
    A ← A + Z × K;
    B ← K × K + B;
endwhile;
output A,B;
endprocedure;
    
```

Determinare il valore di output di A e B , ordinatamente per tre valori di input per Z : 3, 6, 9.

Valore in input per Z	Valore in output per A	Valore in output per B
3		
6		
9		

ESERCIZIO 6

PROBLEMA

Si consideri la seguente procedura PROVA3.

```

procedure PROVA3;
variables A, B, K, Z integer;
Z ← 0;
for K from 1 to 3 step 1 do
    A ← 40;
    B ← 0;
    Z ← Z + 2;
    while A > B do
        A ← A + Z × K;
        B ← (Z + K) × (Z + K) + B;
    endwhile;
    output A, B;
endfor;
endprocedure;
    
```

Determinare le tre coppie di valori di output per A e B.

	A	B
prima coppia di valori		
seconda coppia di valori		
terza coppia di valori		