ESERCIZIO 1

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.

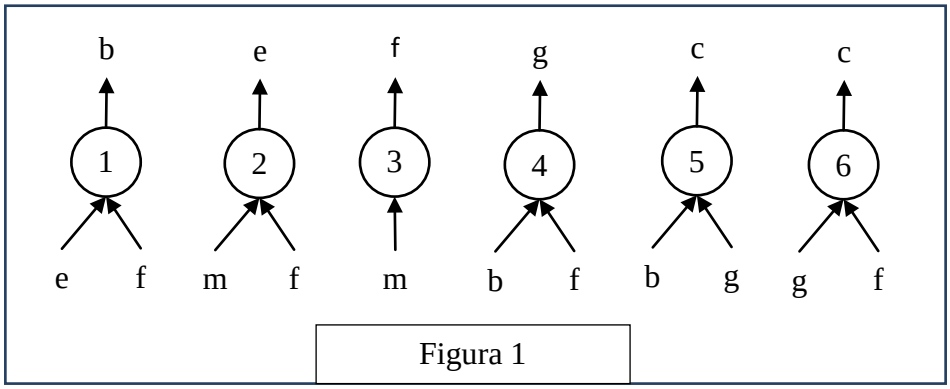

Figura 1

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 *sviluppando* 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].

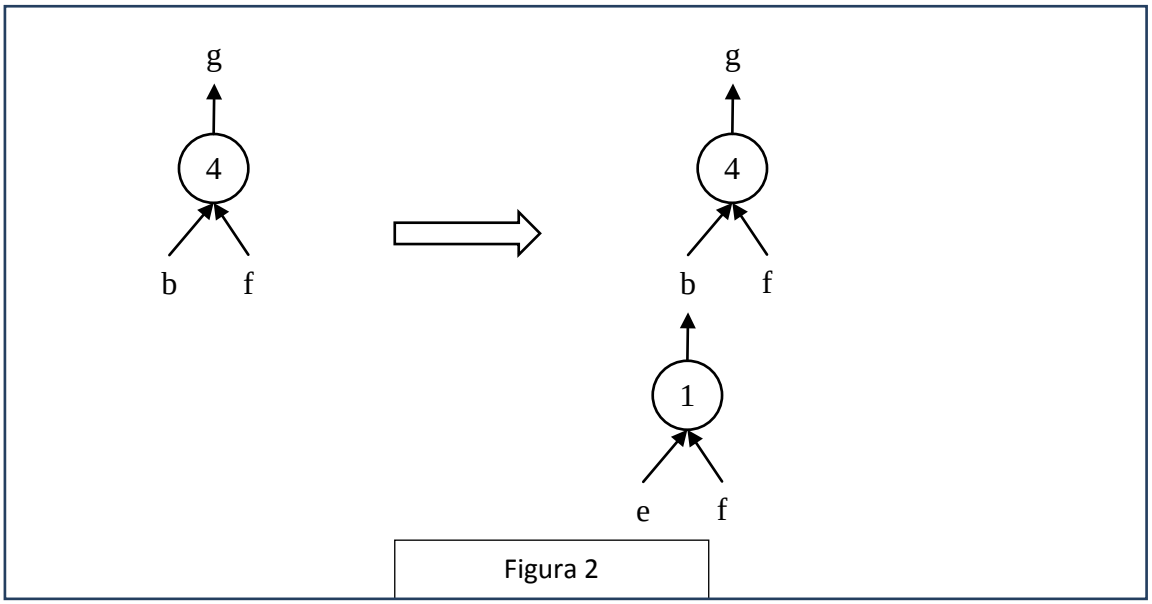

Figura 2

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 della regola che ha come conseguente l'elemento incognito da dedurre.

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.

La lista associata a un (ben preciso) procedimento si costruisce quindi 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, occorre dare la precedenza (nella lista) a quella con sigla *inferiore* (questo per rendere *unica* la lista associata al procedimento).

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 alberi diversi e quindi con insiemi diversi di regole). Per esempio il problema "dedurre **c** a partire da **b** ed **f**" (dalle regole viste sopra) ha due di-

stinti procedimenti risolutivi; gli alberi relativi ai due procedimenti sono mostrati nella seguente figura 3.

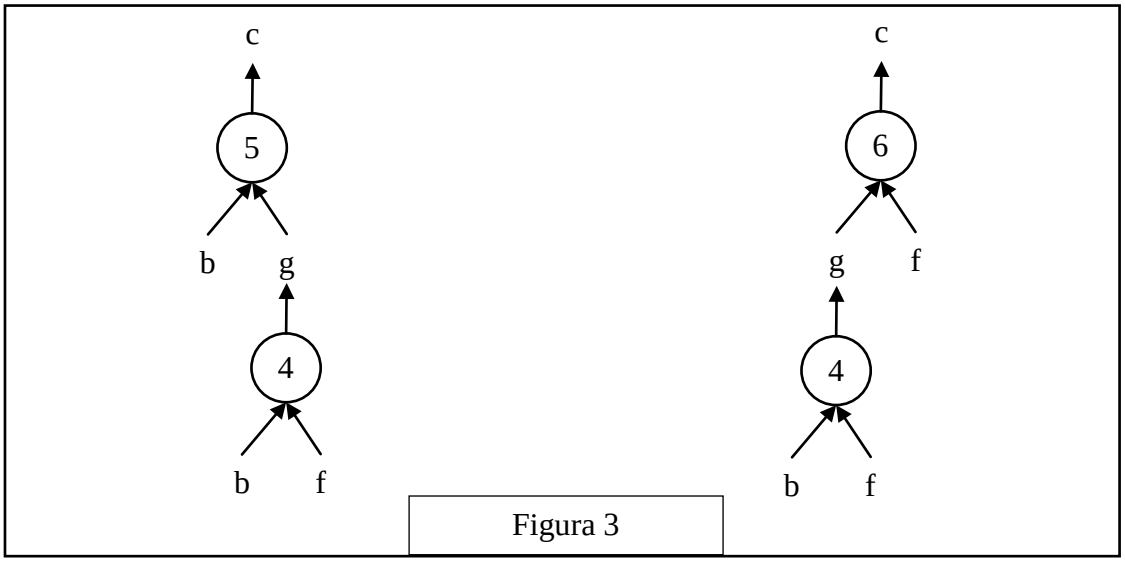

Figura 3

Le liste associate sono, rispettivamente, [4,5] e [4,6].

In un procedimento deduttivo, il numero di regole *differenti* coinvolte (e, quindi, anche il numero di elementi della lista corrispondente al procedimento) si dice *lunghezza* del procedimento.

PROBLEMA
Sono date le seguenti regole:

regola(1,[m,n,u],v)	regola(2,[u,v],z)	regola(3,[v,w],z)
regola(4,[t],u)	regola(5,[p,q],u)	regola(6,[m],u)
regola(7,[u,w],z)	regola(8,[t,u],v)	regola(9,[r],v)
regola(10,[p,t],r)	regola(11,[p,q,u],w)	regola(12,[n,o],m)

Trovare:

1. la lista L1 che descrive il procedimento per dedurre **z** conoscendo **p** e **q**;
2. la lista L2 che descrive il procedimento per dedurre **z** conoscendo **n** e **o**;
3. il numero N di modi diversi per dedurre **z** conoscendo **t** e **w**.

L1	
L2	
N	

SOLUZIONE

L1	[5,11,7]
L2	[12,6,1,2]
N	3

COMMENTI ALLA SOLUZIONE

La incognita comune alle tre domande, **z**, è deducibile con tre regole: 2, 3 e 7, che hanno come antecedenti, rispettivamente, [u,v], [v,w], [u,w].

È più "facile" partire dalla terza domanda, in cui **w** è noto; **w** da solo oppure [t,w] non compaiono nella premessa di alcuna regola: **t**, però, è l'unico antecedente della regola 4 che permette di dedurre **u**. A questo punto è già individuato un procedimento: [4,7]. D'altra parte dopo aver applicato la regola 4, si può applicare la regola 8 per dedurre **v** da **t** e **u**. A questo punto sono noti **t**, **w**, **u** e **v**: si può dedurre l'incognita (**z**) sia con la regola 2 (da [u,v]), sia con la regola 3 (da [v,w]). Quindi, ricapitolando: dati **t** e **w**, ci sono (solo) 3 procedimenti che permettono di ricavare **z**:

[4,7]
[4,8,2]
[4,8,3]

Per la prima domanda, in cui sono dati **p** e **q**, si vede immediatamente che la regola 5 permette di ricavare **u**: per dimostrare **z** occorre **v** oppure **w**. A questo punto è abbastanza evidente che bisogna applicare la regola 11 per ottenere **w** (**v** non è comunque deducibile). Il procedimento risolutivo è [5,11,7].

Per la seconda domanda, in cui sono dati **n** ed **o**, si vede immediatamente che l'unica regola applicabile è la 12 che permette di dedurre **m**. Adesso è applicabile la regola 6 che da **m** permette di dedurre **u**. Occorre ora dedurre **v** o **w** per poter ottenere **z**: solo la prima possibilità è realizzabile con la regola 1 che da **m**, **n** e **u** permette di dedurre **v**. Il procedimento completo è [12,6,1,2].

ESERCIZIO 2

PREMESSA
In un deposito di minerali esistono esemplari di vario peso e valore individuati da sigle di riconoscimento. Ciascun minerale è descritto da un termine che contiene le seguenti informazioni:

tab(<sigla del minerale>, <valore in euro>, <peso in Kg>).

Il deposito contiene i seguenti minerali:

tab(m1,55,84)	tab(m2,53,86)	tab(m3,58,87)
tab(m4,56,83)	tab(m5,52,82)	tab(m6,54,88)
tab(m7,57,81)	tab(m8,51,88)	tab(m9,59,89)

PROBLEMA
Disponendo di un autocarro con portata massima di 165 Kg, trovare la lista L1 delle sigle di 2 minerali diversi trasportabili con questo autocarro che consente di trasportare il massimo valore possibile.
Disponendo di un autocarro con portata massima di 170 Kg, trovare la lista L2 delle sigle di 2 minerali diversi trasportabili con questo autocarro che consente di trasportare il massimo valore possibile.
Disponendo di un autocarro con portata massima di 255 Kg, trovare la lista L3 delle sigle di 3 minerali diversi trasportabili con questo autocarro che consente di trasportare il massimo valore possibile.

N.B. Nelle liste, elencare le sigle in ordine crescente; per le sigle si ha il seguente ordine:
m1<m2<… <m9.

L1	
L2	
L3	

SOLUZIONE

L1	[m4,m7]
L2	[m7,m9]
L3	[m4,m7,m9]

COMMENTI ALLA SOLUZIONE
In generale, un metodo per risolvere il problema (detto della *forza bruta*) è quello di generare tutte le combinazioni di 2 e di 3 minerali presi tra i nove del deposito, calcolarne peso e valore e scegliere, tra quelle trasportabili, quella che ha valore maggiore; poiché tali combinazioni sono rispettivamente (9×8)/(2×1) = 36, (9×8×7)/(3×2×1) = 84, tale metodo è pesante (cioè richiede molti “calcoli” e molto “spazio”).

Per singoli problemi esistono comunque modi “più veloci”, detti *euristici* che consentono di (costruire ed) esaminare un minor numero di combinazioni.
In questo particolare problema conviene mettere i 9 minerali in ordine *decrescente* rispetto al valore.

MINERALE	VALORE	PESO
m9	59	89
m3	58	87
m7	57	81
m4	56	83
m1	55	84
m6	54	88
m2	53	86
m5	52	82
m8	51	88

Per la prima domanda (autocarro di portata 165 kg), si vede che la coppia di maggior valore *trasportabile* è [m7,m4], quindi risolve il problema (ordinando gli elementi come richiesto).
Per la seconda domanda (autocarro di portata 170 kg), si vede che la coppia di maggior valore *trasportabile* è [m9,m7], quindi risolve il problema (ordinando gli elementi come richiesto).
Per la terza domanda (autocarro di portata 225 kg), si vede che la terna di maggior valore *trasportabile* è [m9,m7,m4], quindi risolve il problema (ordinando gli elementi come richiesto).

ESERCIZIO 3

PROBLEMA

John has a lock with three dials; each dials has the symbols A, B, C, D, E, F, G and H. John is a very smart boy, but he is constantly forgetting the combination of his lock. If he can try one lock combination every two seconds, how long will it take to him to try every possible lock combination?

Put your answer in the fields below, keeping in mind that each field is a two digits field: for example 2 hours, 0 minutes and 12 seconds should be written as 02:00:12.

	:		:	

SOLUZIONE

00	:	17	:	04

COMMENTI ALLA SOLUZIONE

Il numero di posizioni per ogni ruota è 8; il numero totale di posizioni delle tre ruote è 8×8×8 = 512. Occorrono a John quindi 1024 secondi; 17 minuti sono 1020 secondi, quindi 1024 secondi sono 0 ore, 17 minuti, 4 secondi.

ESERCIZIO 4

PROBLEMA
Si consideri la seguente procedura PROVA1.

```
procedure PROVA1;
variables A, K integer;
A ← -1;
K ← 0;
while A < 0 do
        K ← K + 1;
        A ← 2 × K × K × K – 40 × K × K + K + 1;
endwhile;
output K, A;
endprocedure;
```

Determinare il valore di output di K e A.

K	
A	

SOLUZIONE

K	20
A	21

COMMENTI ALLA SOLUZIONE
La seguente tabella riporta i valore delle variabili A e K alla *fine* del corpo del ciclo “while”.

K	A
1	-36
2	-141
3	-302
4	-507
5	-744
6	-1001
7	-1266
8	-1527
9	-1772
10	-1989
11	-2166
12	-2291
13	-2352
14	-2337
15	-2234
16	-2031
17	-1716
18	-1277
19	-702
20	21

N.B. Naturalmente, in un caso così laborioso, si può scrivere ed eseguire un programma che riproduca il comportamento della procedura appena vista.

ESERCIZIO 5

PROBLEMA
Si consideri la seguente procedura PROVA2.

```
procedure PROVA2;
variables A, B, K integer;
A ← 0;
B ← 100;
K ← 0;
while A<B do
        K ← K +1;
        A ← A + K^2;
        B ← B – K + 1;
endwhile;
output A, B, K;
endprocedure;
```

Determinare i valori di output.

N.B. Il simbolo ^ denota l'elevamento a potenza.

A	
B	
K	

SOLUZIONE

A	91
B	85
K	6

COMMENTI ALLA SOLUZIONE
Alla fine del corpo del ciclo "while" le variabili K, A e B assumono i valori riportati nella tabella seguente.

K	A	B
1	1	100
2	5	99
3	14	97
4	30	94
5	55	90
6	91	85

ESERCIZIO 6

PROBLEMA
Si consideri la seguente procedura PROVA3.

```
procedure PROVA3;
variables A, P, Q, R, K integer;
input A;
P ← A;
Q ← A;
R ← A;
for K from 1 to 9 step 1 do
        input A;
        if A > P then Q ← P; P ← A;
                else if A > Q then Q ← A; endif;
        endif;
        if A<R  then R ← A;  endif;
endfor;
output P, Q, R;
endprocedure;
```

Se i valori di input per A sono 12, 9, 15, 3, 9, 3, 1, 8, 12, 13 calcolare i valori di output.

P	
Q	
R	

SOLUZIONE

P	15
Q	13
R	1

COMMENTI ALLA SOLUZIONE
Vengono acquisiti 10 valori interi positivi per A; in R rimane il più piccolo di tali valori. In P rimane il massimo dei valori per A e in Q rimane il valore immediatamente inferiore al massimo.