

**Esercizio 1.** Svolgere tutti i punti.

a-1) Si consideri il seguente programma logico e se ne calcolino tutti gli answer set, **illustrando adeguatamente il procedimento seguito.**

```
m(Y) :- s(Y,X), not v(Y). 1  
s(X,Z) :- m(X), v(Y), Z=X+Y. 3  
v(Y) | p(Y) :- not m(Y), s(Y,Z). 2  
s(2,3).
```

a-2) Si aggiunga il seguente strong constraint al programma del punto precedente.

```
:- D=#sum{Z:v(Z)}, E=#sum{H:m(H)}, D=E.
```

Come influisce sulle soluzioni del programma? Perché? Motivare adeguatamente la risposta.

a-1)

~~$m(2) :- s(2,3), \text{not } v(2).$~~   
 ~~$v(2) | p(2) :- \text{not } m(2), s(2,3).$~~   
 ~~$s(2,4) :- m(2), v(2), 4=2+2$~~

~~$m(2) :- s(2,4), \text{not } v(2).$~~   
 ~~$v(2) | p(2) :- \text{not } m(2), s(2,4).$~~

\* La seconda regola non genera più nulla perché le teste hanno generato le stesse informazioni.

=> GROUND:

$s(2,3).$

$m(2) :- \text{not } v(2).$

$v(2) | p(2) :- \text{not } m(2).$

$s(2,4) :- m(2), v(2).$   $\circ/\circ$  Non potrò mai essere verificato.  
perché  $m(2)$  e  $v(2)$  si escludono vicendemente.

AS: {

$$\left\{ \begin{array}{l} \{ s(2,3), m(2) \}, \\ \{ s(2,3), v(2) \} \end{array} \right. \quad \begin{array}{l} *1 \\ *2 \end{array}$$

a-2)

$$:- D = \# \text{sum} \{ z : v(z) \}, E = \# \text{sum} \{ H : m(H) \}, D = E.$$

Introducendo questo strong constraint NON verrebbe scelto alcun AS, in quanto se è presente uno dei due predicati presenti nelle somme, non sarà presente l'altro, con le informazioni, quindi, le somme non saranno mai uguali.

b) Si consideri ora un programma P (non è necessario sapere come è fatto) i cui answer set sono già stati calcolati e sono riportati di seguito.

```
A1: {a(1,2), a(4,5), h(1), b(4), a(1,5), k(1,1), k(1,4)}
A2: {a(1,2), a(4,5), b(1), b(4), k(1,1)}
A3: {a(1,2), a(4,5), h(1), h(4)}
A4: {a(1,2), a(4,5), b(1), h(4), k(1,1)}
```

Si supponga di aggiungere i seguenti weak constraint al programma P. Si calcoli quale sarebbe il costo di ognuno degli answer set riportati sopra: si riporti il **costo dettagliato per ciascun answer set** e si indichi quello **ottimo**, commentando adeguatamente il procedimento seguito.

```
:~ h(X), #sum{Y:a(X,Y)} > 1. [X@2,X]
:~ b(X), k(Z,X). [1@X,X,Z]
```

A1: 1@2 1@4

:~ h(1), 2+5 > 1. [1@2, 1]  
 :~ b(4), k(1,4). [1@4, 4, 1]

A2: 1@1 OPTIMUM

:~ b(1), k(1,1). [1@1]

A3:  $5@2$

$$\begin{aligned} &: \sim h(1), 2 > 1, & [1@2, 1] \\ &: \sim h(4), 5 > 1. & [4@2, 4] \end{aligned}$$

A4:  $1@1$   $4@2$

$$\begin{aligned} &: \sim h(4), 5 > 1. & [4@2, 4] \\ &: \sim b(1), K(1,1). & [1@1, 1, 1] \end{aligned}$$

l'AS ottimo è A3, in quanto paga solo  $1@1$



## Esercizio 2.

Il nostro amico Ciccio Pasticcio vuole iscriversi ad una scuola per diventare chef ma è molto indeciso sull'istituto da frequentare: le possibilità sono tante, ma nessuna lo convince appieno: una è troppo lontana, un'altra non ha una buona reputazione, un'altra ha troppi iscritti... insomma, ognuna ha qualcosa che non va e dovendo trovare un compromesso, come al solito, ci ha chiesto aiuto.

Scrivere quindi un programma che lo guidi nella scelta dell'istituto migliore, tenendo presente le considerazioni che seguono.

- 1 • L'istituto scelto non deve distare più di 20 chilometri da casa.
- 2 • Ad ogni istituto è associato un numero (da 1 a 10) che indica il livello di reputazione dell'istituto: sono da evitare quelli il cui livello è minore di 7.
- 3 • Ogni istituto fornisce un certo numero di servizi, ad esempio lezioni serali, lezioni a distanza, laboratori per la pratica, ecc. Si preferisce l'istituto il cui numero di servizi diversi offerti è maggiore.
- 4 • Cosa ancora più importante, preferisce gli istituti il cui il numero di iscritti è minore.

## Modello dei dati in input:

CAMPUS DI ARCAVACATA - Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS)  
tel. (+39) 0984 496400 fax (+39) 0984 496410 - <http://sv.mat.unical.it>

1/2

- vado(ID) | vado(ID) :- istituto(ID, -, -, -).

% Si deve andare in uno ed un solo istituto

:- #count{ID: vado(ID)} != 1.

% 1

:- vado(ID), istituto(ID, Q, -, -), quartiere(Q, D), D > 20.

% 2

:- vado(ID), istituto(ID, -, -, R), R < 7.

% 3

servizioDisponibile(S) :- servizio(-, S).

:- servizioDisponibile(S), vado(ID), not servizio(ID, S). [1@1, S]

% 4

:- vado(ID), istituto(ID, -, I, -). [I@2]



