

Esercizio 1. Svolgere tutti i punti.

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

```
ape(X,Y) | vespa(Y) :- punge(X,Y), not zanzara(X,Y). 1  
zanzara(X,Y) :- fastidio(X,Y,Z), not ape(X,Y), not punge(X,Z). 2  
ape(X,Y) :- zanzara(X,Z), Y=Z+2. 3  
  
fastidio(1,1,1). fastidio(1,2,3).  
punge(1,1).  
punge(1,2).
```

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

```
:- #sum{Z,T:zanzara(X,Z),ape(Y,T)}>K, vespa(K).
```

a-1)

~~ape(1,1) | vespa(1) :- punge(1,1), not zanzara(1,1).~~

~~ape(1,2) | vespa(2) :- punge(1,2), not zanzara(1,2).~~ punge(1,1) è un fatto

~~zanzara(1,1) :- fastidio(1,1,1), not ape(1,1), not punge(1,1).~~

~~zanzara(1,2) :- fastidio(1,2,3), not ape(1,2), not punge(1,3).~~

~~ape(1,3) :- zanzara(1,1), 3 = 1 + 2.~~

nessuna regola ha in testa punge(1,3)

~~ape(1,4) :- zanzara(1,2), 4 = 2 + 2.~~

Il grounding termine

AS:

A1: {[... EDB ...], ape(1,1), zanzara(1,2), ape(1,4)}

X A2: {[... EDB ...], vespa(1), zanzara(1,2), ape(1,4)}

A3: {[... EDB ...], ape(1,1), ape(1,2)}

A4: {[... EDB ...], vespa(1), ape(1,2)}

a-2)

$\vdash \# \text{sum} \{ z, T : \text{zanzara}(X, z), \text{ape}(Y, T) \} > k, \text{respa}(k).$

Introducendo questo strong constraint, viene scelto A2, in quanto, oltre ad essere l'unica a contenere tutti i prelievi necessari, gli stessi verificano il corpo del constraint.

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: { b(2,2), b(2,5), c(2,2), c(2,3), d(2,3) }

A2: { a(2), b(2,3), c(2,2), c(2,3) }

A3: { b(2,3), b(2,2), c(2,2), c(2,3) }

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 il procedimento seguito.

:~ b(K, T), c(K, X). [K@T, K, T, X]
:~ d(X, Y), not a(X), Z=X-1. [X@Z, X, Y, Z]

b)

A1: 2@1 4@2 4@5

:~ b(2,2), c(2,2). [2@2, 2, 2, 2]

:~ b(2,2), c(2,3). [2@2, 2, 2, 3]

:~ b(2,5), c(2,2). [2@5, 2, 5, 2]

:~ b(2,5), c(2,3). [2@5, 2, 5, 3]

:~ d(2,3), not a(2) 1=2-1. [2@1, 2, 3, 1]

A2: 4@3 ← OPTIMUM

:~ b(2,3), c(2,2) [2@3, 2, 3, 2]

$\sim b(1,3), c(1,3)$ $[2@3, 2, 3, 3]$

A3: 4@2 4@3

$\sim b(2,2), c(2,2),$	$[1@2, 2, 2, 2]$
$\sim b(2,2), c(2,3),$	$[1@2, 2, 2, 3]$
$\sim b(2,3), c(2,2),$	$[1@3, 2, 3, 2]$
$\sim b(2,3), c(2,3),$	$[1@3, 2, 3, 3]$

L'AS ottimo è A2



Esercizio 2. L'Accademia Letteraria di Pasticciopoli sta organizzando un concorso letterario: alcuni famosi scrittori italiani saranno i giudici del concorso e dovranno votare il romanzo più bello tra quelli in gara. I cittadini di Pasticciopoli, emozionati per la novità e per la possibilità di conoscere i giudici, si riscoprono tutti scrittori in erba e decidono di partecipare al concorso, inviando i loro manoscritti. L'organizzazione, che non credeva che un concorso letterario potesse riscontrare un tale successo, si ritrova con centinaia di romanzi da votare e pochissimo tempo a disposizione: un'impresa impossibile. Decide perciò di effettuare una prima scrematura, scartando a priori romanzi che sembrano non essere promettenti, così da inviare alla fase finale del concorso, la votazione dei giudici, solo un'accurata selezione. I nostri amici Ciccio Pasticcio e Renata Limbranata, che fanno parte dell'organizzazione, sono i responsabili della selezione, e come al solito ci chiedono aiuto per salvare la situazione....

Si scriva un programma logico ASP che selezioni i romanzi da promuovere alla fase finale del concorso, tenendo conto delle specifiche indicate di seguito.

- 1 • I romanzi possono essere di vario genere (giallo, avventura, romantico, ecc.) e ogni genere deve essere rappresentato al concorso, quindi per ogni genere, almeno un romanzo deve essere selezionato.
- 2 • Per avere più possibilità di vincere, alcuni partecipanti hanno deciso di inviare più manoscritti. Non è possibile selezionare più di due romanzi il cui autore è lo stesso.
- 3 • E' stato fissato un numero massimo di pagine che in totale i giudici possono leggere: il numero totale di pagine dei lavori selezionati non deve superare tale limite.

Ci sono poi dei criteri fondamentali per l'ottimizzazione della selezione; sono elencati di seguito, IN ORDINE DI IMPORTANZA CRESCENTE:

- 4 • Al concorso hanno partecipato anche scrittori provenienti dalle cittadine limitrofe. Per evitare che i cittadini di Pasticciopoli siano privilegiati, si vuole massimizzare il numero di romanzi selezionati con autori che provengono da città diverse da Pasticciopoli.
- 5 • L'idea del concorso è nata per dare spazio ai giovani, perciò si preferisce che l'età media degli autori dei romanzi selezionati sia la più piccola possibile.
- 6 • Uomini e donne devono avere più o meno la stessa visibilità. Perciò si preferiscono le selezioni in cui il numero di autori (di sesso maschile) e il numero di autrici siano i più vicini possibili.

MODELLO DEI DATI IN INPUT

romanzo(ID, IDAutore, Genere, Pagine)
autore(ID, Eta, Sesso, CittaDiProvenienza)
limite(P)

\leftarrow i romanzi
 \leftarrow gli autori
 \leftarrow il limite di pagine in totale

- partecipa(IDR, IDA, G, P) | partecipa(IDR, IDA, G, P) :- romanzo(IDR, IDA, G, P).

% 1

:- #count{G : partecipa(-, -, G, -)} != #count{G : romanzo(-, -, G, -)}.

% 2

:- #count{IDR : partecipa(IDR, IDA, -, -)} > 2, autore(IDA, -, -, -).

% 3

:- #sum{P, IDR : partecipa(IDR, -, -, P)} > PT, limite(PT).

% 4

:~ partecipa(-, IDA, -, -), autore(IDA, -, -, pasticeiopoli). [1@1, IDA]

% 5

:~ #avg{E, IDA : partecipa(-, IDA, -, -), autore(IDA, E, -, -)} = Media.
[Media@2]

% 6 m - uomini / f - donne

:~ #count{IDA : partecipa(-, IDA, -, -), autore(IDA, -, m, -)} = U,
#count{IDA : partecipa(-, IDA, -, -), autore(IDA, -, f, -)} = D, Diff = U - D,
& abs(Diff, Z). [Z@3]

