

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.**

```
a(X,X) :- b(X), b(Y), not c(X). 3  
b(Y) :- a(Y,X), c(X), X=1+Y. 2  
a(X,Y) | b(X) | b(Y) :- c(X), Y=X+1. 1  
  
a(1,2).  
c(2).  
c(3).
```

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

```
: - #sum{X:a(X,Y)} <= 5.
```

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

a-1)

~~a(2,3) | b(2) | b(3) :- c(2), 2 = 1 + 1.~~
~~a(3,4) | b(3) | b(4) :- c(3), 3 = 2 + 1.~~
~~b(1) :- a(1,1), c(2), 2 = 1 + 1.~~
~~b(2) :- a(2,3), c(3), 3 = 2 + 1.~~

~~a(1,1) :- b(1), b(*), not c(1).~~
~~a(2,2) :- b(2), b(*), not c(2).~~
~~a(3,3) :- b(3), b(*), not c(3).~~
~~a(4,4) :- b(4), b(*), not c(4).~~

* si può inserire un valore qualunque che marche, non è importante.

Non potendosi più generare $c/1$, ed essendo le regole precedenti dipendenti da quest'ultimo, il grounding termine.

AS: {

~~{ a(1,2), c(2), c(3), b(1), b(3), a(1,1) }~~

~~{ a(1,2), c(2), c(3), b(1), b(2), a(3,4), a(1,1) }~~

~~$\{a(1,2), c(2), c(3), b(1), b(2), b(4), a(4,4), a(1,1)\}$~~

a-2)

$\vdash \# \text{sum} \{ X : a(X, Y) \} \leq 5.$

il programma diventerebbe incrementale, in quanto nel corso del primo AS la sum restituisce 1, nel secondo 4, nel terzo 5

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: {h(3), m(3,3), c(3,2), m(3,2), c(2,2)}

A2: {h(2), m(3,3), m(2,3), m(3,2), m(2,2)}

A3: {h(3), h(2), m(3,3), c(3,2), m(3,2), m(2,2)}

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.**

:~ m(X, Y), h(X). [Y@X, X, Y]
 :~ #min{X, Y : m(X, Y)} > 2. [1@2]

b)

A1: 1@2 5@3

:~ m(3,3), h(3). [3@3, 3, 3]
 :~ m(3,2), h(3). [2@3, 3, 2]
~~#min~~
 :~ 3 > 2 [1@2]

A2: 5@2 OPTIMUM

:~ m(2,3), h(2). [3@2, 2, 3]
 :~ m(2,2), h(2). [2@2, 2, 2]

X :~ 2 > 2

A3: $2@2 \ 5@3$

$\vdash m(3,3), h(3)$
 $\vdash m(3,2), h(3)$
 $\vdash m(2,2), h(2)$

$[3@3, 3, 3]$
 $[2@3, 3, 2]$
 $[2@2, 2, 2]$

X $\vdash m(2,2)$

A2 è l'AS ottimo

Esercizio 2.

Il figlio dei nostri cari amici Ciccio Pasticcio e Renata Limbranata sta per compiere 18 anni e per festeggiare l'evento i suoi genitori hanno deciso di organizzare per lui una bella festa. Pippo avrebbe voluto festeggiare in discoteca con tantissima gente per lasciare entrare tutti i suoi amici, ma anche conoscenti, amici di amici, amici sui social, ecc. Tuttavia, date le restrizioni vigenti e la paura per il virus che circola, i suoi genitori lo hanno convinto per una soluzione che consenta di mantenere l'ordine e le distanze: una cena in un ristorante molto capiente, così da poter invitare più gente possibile, ma in cui gli ospiti siano costretti a rimanere seduti ai propri posti. Nonostante la grande capienza, Pippo non può invitare tutti quelli che vorrebbe ma deve selezionare con cura gli invitati così da garantire che la cena sia piacevole e non ci siano disgradi. Per facilitare la scelta e non rischiare di dimenticare di invitare qualcuno a cui tiene, ha deciso di identificare i possibili invitati tra la sua rete di amicizia in un social network molto famoso. In tale social network, le persone iscritte possono aggiungere amici e contrassegnarne alcuni come amici stretti; inoltre, il social network effettua alcune statistiche sul grado di antipatia tra due

persone nella rete. La rete di amicizia di Pippo però è molto vasta e per non commettere errori ci ha chiesto di utilizzare DLV per automatizzare la selezione.

Si scriva un programma DLV che aiuti Pippo ad esplorare le relazioni di amicizia del social network e fornisca opportuni suggerimenti sulle persone che potrebbero essere invitate, tenendo presenti le considerazioni che seguono.

- Le persone da invitare devono essere scelte da una lista formata da tutti gli amici stretti di Pippo e dagli amici stretti delle persone già in lista.
- 1 - Pippo non vuole ragazzini alla sua festa, perciò la lista deve contenere solo maggiorenni.
- 2 - Vuole una festa giovane e perciò l'età media degli invitati deve essere minore di 25 anni.
- 3 - Il numero di invitati non deve superare la capienza della sala.
- 4 - Pippo vorrebbe invitare più persone possibili.
- 5 - Ma è ancora più importante garantire un clima piacevole e rilassato, perciò sono preferite le soluzioni in cui il grado totale di antipatia tra gli invitati è il più piccolo possibile.
- 6 - E soprattutto, non vuole far torto ai suoi amici stretti lasciandone a casa qualcuno. Perciò sono preferite le soluzioni in cui il numero di amici stretti di Pippo che sono stati selezionati per essere invitati è il più grande possibile.

Modello dei dati in input:

amicostretto(X,Y). \leftarrow La relazione di amicizia tra le persone
eta(X,E). \leftarrow Le età degli amici
capienza(C). \leftarrow La capienza della sala
antipatia(X,Y,G). \leftarrow Il grado di antipatia tra le persone
etaMedia(25). \leftarrow L'età media degli invitati

* Si presuppone dello relaz. "amicostretto" che non ci sono due pers. con stesso nome ed età diverse

persona(X, E) :- amicostreto($X, -$), era(X, E).
persona(Y, E) :- amicostreto($-, Y$), era(Y, E).

invito(pippo). % sicuramente il festeggiato è invitato
- invito(X) | invito(X) :- persona($X, -$).

% 1

:- invito(X), persona(X, E), $E < 18$.

% 2

:- # avg{ E, X : invito(X), persona(X, E)} ≥ 25 .

% 3

:- capienza(C), # count{ X : invito(X)} $> C$.

% 4

:- \~ - invito(X). [1@1, X]

% 5

:- \~ invito(X), invito(Y), antipatia(X, Y, G), $X > Y$. [G@2, X, Y]

% 6

:- \~ amicostreto(pippo, Z), - invito(Z). [1@3, Z]

