

**Prova d'esame del 27/01/2017****Esercizio 1.** Svolgere tutti i punti.

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

```
apollo(X) :- giunone(X), not zeus(X,X).
zeus(Y,X) :- mercurio(X), giunone(Y), not apollo(X).
mercurio(Z) :- zeus(X,Y), Z=X+1.
mercurio(Y) :- zeus(Y,X).

zeus(3,4).
giunone(2).
```

b) Si aggiunga il seguente weak constraint, e si calcoli quindi il costo di ogni answer set del programma risultante. Indicare quindi quello ottimo (o quelli ottimi, se più di uno).

% DLV syntax :- ~ mercurio(X), zeus(X,Y) . [Y:X]	% ASP-Core-2 syntax :- ~ mercurio(X), zeus(X,Y) . [Y@X, X,Y]
---	---

c) Si aggiunga ora il seguente constraint (vincolo). Illustrare come cambiano gli answer set, e calcolare nuovamente quello ottimo (o quelli ottimi, se più di uno).

```
:- #sum{X,Y : zeus(X,Y), mercurio(X)} > 8.
```

**Esercizio 2.** Il nostro amico Ciccio Pasticcio è in vacanza in montagna nel paese di Nevolandia, assieme alla cara moglie Renata Limbranata. Questa inizia a pensare che Ciccio porti sfortuna: infatti, pochi giorni dopo il loro arrivo, una serie di incidenti di varia natura colpiscono abitanti e turisti del paese, e tutti insieme! Per fortuna nessun incidente è stato fatale, ma ora bisogna recuperare tutte le persone coinvolte e portarle al sicuro al quartier generale dei soccorsi. Il coordinatore gestore dell'emergenza, Tuturro Tisoccorro, conosce di fama le capacità organizzative di Ciccio, e così gli chiede aiuto nel mettere a punto le squadre di soccorso. Si aiuti il nostro amico Ciccio a scrivere un programma ASP che risolva il problema, tenendo conto delle considerazioni esposte di seguito.

- È nota una serie di posizioni geografiche, ciascuna delle quali è stata teatro di un incidente; ogni incidente ha caratteristiche specifiche: numero di persone coinvolte, lista di professionalità necessarie per completare l'intervento, stima del tempo necessario per rientrare al quartier generale.
- Tuturro ha a disposizione un insieme di soccorritori, tutti radunatisi nel quartier generale, ciascuno con specifiche abilità: alcuni soccorritori posseggono più di una abilità.
- Per ciascun luogo in cui si è verificato un incidente, è necessario predisporre una squadra di soccorritori composta in modo tale da coprire tutte le professionalità richieste dall'incidente stesso; si noti che il numero di professionalità richieste

**Prova d'esame del 27/01/2017**

potrebbe essere superiore ad uno per qualche categoria (ad esempio, per un particolare sito potrebbero essere necessari 3 paramedici, oppure 2 alpinisti, etc.).

- Per ciascun luogo, il numero dei soccorritori inviati non può essere inferiore a quello delle persone coinvolte nell'incidente.
- Se un soccorritore possiede più di una professionalità, può esercitarle tutte ricoprendo più ruoli; naturalmente, un soccorritore che ricopra più ruoli sarà presumibilmente meno efficace di uno che dovrà svolgere un compito solo. Per questo motivo, si desidera minimizzare il numero di soccorritori che svolge più di un ruolo nella squadra cui è assegnato. [Suggerimento: si potrebbe cercare di massimizzare il numero di soccorritori...]
- Ancora più importante, i medici sono preziosi al quartier generale; pertanto, per ciascun medico si vuole minimizzare il tempo in cui sarà impegnato nelle operazioni di soccorso e quindi necessario per rientrare al quartier generale.

**Modello dei dati in input:**

incidente(ID, Persone, Tempo).	← Gli incidenti per cui creare le squadre
professRichiesta(ID_Inc, Prof, Num).	← Professionalità richieste per gli incidenti. In generale, più istanze di questo predicato sono riportate per ciascun incidente. Una istanza indica che per l'incidente <i>ID_Inc</i> sono necessari <i>Num</i> soccorritori con la professionalità <i>Prof</i> .
soccorritore(ID).	← I soccorritori
professionalita(ID_Socc, Prof).	← Le professionalità possedute dai soccorritori. In generale, più istanze di questo predicato sono riportate per ciascun soccorritore. Una istanza indica che il soccorritore <i>ID_Socc</i> possiede la professionalità <i>Prof</i> .

**Esercizio 3.** Si scriva un programma DLV che consenta di risolvere il gioco descritto di seguito. Si ha una griglia quadrata di dimensione NxM, in cui ciascuna cella contiene un numero compreso tra 0 ed N-1. In verità, la griglia contiene una disposizione di tutte le tessere di un domino di dimensione N. Si ricorda che le tessere del domino sono costituite da un rettangolo diviso in due quadrati, ciascuno dei quali contiene un numero tra 0 ed N-1. Ad esempio, un insieme completo per la dimensione 4 contiene le seguenti tessere: 0-0, 0-1, 0-2, 0-3, 1-1, 1-2, 1-3, 2-2, 2-3, 3-3 – è facile quindi generalizzare. L'obiettivo del gioco è identificare il posizionamento delle tessere sulla griglia. ESEMPIO: Nella figura in basso è riportata una possibile griglia in input (sinistra) con la relativa soluzione (destra). Si noti, ad esempio, come la tessera 0-0 (evidenziata in grigio nella figura) si trovi ad occupare le celle in posizione <2,4> e <3,4>.

0	3	2	2	2
1	1	1	3	2
0	3	2	1	0
1	0	2	3	0

0	3	2	3	2
1	1	1	3	2
0	3	2	1	0
1	0	2	3	0