



Prova d'esame del 21/11/2014

Esercizio 1. Svolgere tutti i punti.

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

```
P.
nessuno(1,4).    nessuno(2,3).
tutti(1,5).     tutti(2,2).

alcuni(X) :- not nessuno(X,Y), tanti(Y), pochi(X).
tanti(Y) | alcuni(X) | pochi(X) :- tutti(X,Y), not alcuni(Y).

:- #max{X:tanti(X)}>1, #min{Y:alcuni(Y)}<2.
```

b) Si aggiungano i seguenti weak constraint:

```
:~ tanti(X). [X:1]
:~ pochi(X). [2:X]
```

Calcolare quindi gli answer set riportando per ciascuno il costo. Indicare quindi quello ottimo (o quelli ottimi, se più di uno).

Esercizio 2. Il nostro amico Ciccio Pasticcio deve organizzare un incontro d'affari tra due compagnie che storicamente sono state l'una per l'altra concorrenti feroci e, a volte, persino sleali: la CompraQuelloCheDicoIo SPA e la ScelgoIoPAghiTu LTD. Stavolta devono fare fronte comune contro una nuova compagnia, giovane, innovativa e dinamica, che sta togliendo loro tutti i clienti: la TantoNoiSiamoRicchi SRL. L'incontro tra le due compagnie sarà un vero e proprio evento sociale, dato che saranno coinvolti, tra dirigenti, avvocati e accompagnatori vari un sacco di strani personaggi. Il problema è che è necessario scegliere come metterli a sedere al tavolo per l'evento serale, e non è semplice, soprattutto dato che alcuni di loro, protagonisti di passaggi da una parte all'altra e di voltagabbana clamorosi, non vanno affatto d'accordo!

Aiutate il nostro amico scrivendo un programma DLV che risolva il suo problema, tenendo conto delle considerazioni illustrate di seguito.

- Nella struttura presa in affitto per l'occasione, sono disponibili un certo numero di tavoli, ciascuno dei quali ha una certa capienza ed un certo costo. Il costo non è legato alla capienza, bensì alla qualità del tavolo e delle sedie ed alla posizione in sala.
- Tutti gli invitati devono essere seduti esattamente presso uno ed un solo tavolo; naturalmente, ciascun tavolo non può ospitare più persone di quante previste dalla propria capienza.
- Non tutti i tavoli devono essere impegnati: un tavolo può essere anche lasciato vuoto.
- Ciccio ha individuato le seguenti categorie di persone: dirigenti, avvocati, accompagnatori, spie. Ciascun tavolo deve avere rappresentanti di almeno 3 di queste 4 categorie. Da questo si evince che se un tavolo



Prova d'esame del 21/11/2014

non è lasciato vuoto, allora deve avere almeno altri 3 persone (in pratica un tavolo può avere 0 [zero] persone assegnate, oppure un qualunque numero $N \geq 3$, ma NON 1 o 2).

- Alcune persone sono poco compatibili tra loro, e non possono assolutamente stare sedute allo stesso tavolo.
- Si vuole massimizzare il numero di tavoli che ospitano almeno un esponente per ciascuna 4 categorie di persone identificate da Ciccio.
- Cosa più importante, si vuole minimizzare il costo totale.

Modello dei dati in input

persona(Nome,Compagnia,Tipologia)

← i planner disponibili

stanza(ID)

← le stanze disponibili

tavolo(ID, ID-Stanza, Capienza, Costo)

← i tavoli disponibili

incompatibili(Persona1,Persona2)

← l'elenco delle coppie di persone poco compatibili

Esercizio 3. Si scriva un programma DLV che consenta di risolvere qualunque istanza del gioco descritto di seguito. Si ha una “scacchiera” quadrata di dimensione 9×9 , suddivisa in 9 settori, ciascuno di dimensione 3×3 , similmente al sudoku. Sulla scacchiera alcune caselle sono occupate da un certo numero di pedine, mentre altre sono libere. Ciascuna pedina può essere spostata dalla posizione iniziale ad una casella adiacente, al massimo una volta sola. Lo scopo è passare dalla configurazione iniziale ad una in cui le pedine formano un percorso unico e non spezzato (una specie di “serpentone”), tenendo conto delle regole indicate di seguito.

- Per ogni pedina, si può scegliere se lasciarla nella casella dove si trova inizialmente, OPPURE se spostarla su una delle caselle adiacenti. Ciascuna pedina può subire uno ed un solo spostamento.
- Data una casella, si considerano adiacenti ad essa quelle a nord, sud, est ed ovest (in “croce”).
- Per ciascuno dei settori in cui è divisa la scacchiera, il numero massimo di pedine sottoposte a spostamento è UNO.
- Il numero totale di pedine spostate non può essere superiore ad un valore prefissato, che si intende dato in input assieme alla configurazione iniziale della scacchiera.
- Gli spostamenti devono essere tali che tutte le pedine facciano parte di un unico “serpentone”, il quale deve essere continuo e non spezzato.

ESEMPIO: Se in input si avesse la scacchiera con le pedine posizionate come nella figura qui sotto a sinistra, e il numero di spostamenti consentiti fosse 4, allora basterebbe spostare le pedine dalle caselle evidenziate in grigio in come indicato in figura per ottenere un percorso completo (visibile a destra in figura).

