T-79.154 **Syksy 2003**
Logiikka tietotekniikassa: erityiskysymyksiä II
Laskuharjoitus 10
Ratkaisut

1. Esitetään graafi (V, E) siten, että ohjelmaan lisätään faktajoukot:

$$\{node(v) \leftarrow \mid v \in V\} \text{ ja}$$
$$\{edge(a, b) \leftarrow \mid (a, b) \in E\}$$

Muodostetaan valinta solmujen yli siten, että kukin solmu v joko otetaan peitteeseen mukaan ($in(v)$ on tosi) tai se jätetään pois ($notin(v)$ on tosi):

$$in(V) \leftarrow node(V), \text{not } notin(V)$$
$$notin(V) \leftarrow node(V), \text{not } in(V) \ .$$

Seuraavaksi todetaan, että on virhe mikäli jonkin kumpikaan pää ei kuulu peitteeseen:

$$fail \leftarrow edge(X, Y), \text{not } in(X), \text{not } in(Y) \ .$$

Lisäksi täytyy kieltää liian isot solmupeitteet säännöllä muotoa:

$$fail \leftarrow in(V_1), \ldots, in(V_{k+1}), V_1 \neq V_2, V_1 \neq V_3, \ldots, V_k \neq V_{k+1}$$

Tämä sääntö voidaan kirjoittaa vasta sitten, kun K:n arvo on jo kiinnitetty. Tavallisilla logiikkaohjelmointisäännöillä ei ole helppoa muodostaa tälle ongelmalle *yhtenäiskoodausta*, jolloin samaa ohjelmaa voisi käyttää kaikkien graafien solmupeitteiden löytämiseksi.

Lopuksi kielletään mallit, joissa *fail* on tosi:

$$f \leftarrow \text{not } f, fail \ .$$

Smodels-sääntöjen käyttäminen tekee ohjelmasta yksinkertaisemman, ja tällöin myös yhtenäiskoodaus on mahdollinen:

$$\begin{aligned} \{in(V)\} &\leftarrow node(V). \\ &\leftarrow edge(X, Y), \text{not } in(X), \text{not } in(Y) \\ &\leftarrow k + 1 \ \{in(V) : node(V)\} \ . \end{aligned}$$

2. Valitaan aluksi jokin solmujen osajoukko ytimeksi:

$$\begin{aligned} in(V) &\leftarrow node(V), \text{not } notin(V) \\ notin(V) &\leftarrow node(V), \text{not } in(V) \ . \end{aligned}$$

On virhe, jos kaksi vierekkäistä solmua ovat molemmat ytimessä:

$$\leftarrow edge(X, Y), in(X), in(Y) \ .$$

Jokaisella ytimeen kuulumattomalla solmulla täytyy olla seuraaja ytimessä:

$$\begin{aligned} ok(X) &\leftarrow edge(X, Y), \text{not } in(X), in(Y) \\ &\leftarrow node(X), \text{not } in(X), \text{not } ok(X) \ . \end{aligned}$$

Tässä otettiin käyttöön predikaatti $ok/1$, koska eksistentiaalisen määritelmän tekeminen ilman apupredikaatteja on yleensä aika hankalaa.

Smodels-sääntöjä käyttäessä apupredikaattia ei tarvita:

$$\begin{aligned} \{in(V)\} &\leftarrow node(V) \\ &\leftarrow edge(X, Y), in(X), in(Y) \\ &\leftarrow \{in(Y) : edge(X, Y)\} \ 0, \text{not } in(X), node(X) \ . \end{aligned}$$

3.
```
%% Queens.lp  -- solves the 8 queens problem
%
% usage: lparse -dn -cn=8 queens.lp | smodels 0
%
% Different board sizes may be tried with different values for n.

% use the default value 8 if nothing else is specified
const n = 8.

% define the columns and rows
row(1..n).
column(1..n).

% There has to be exactly one queen in each column ...
1 { queen_pos(X, Y) : row(Y) } 1 :-
    column(X).

% ... and exactly one queen in each row.
1 { queen_pos(X, Y) : column(X) } 1 :-
```

```
    row(Y).

% There may not be two queens in the same diagoal.
 :- queen_pos(X1,Y1),
    queen_pos(X2, Y2),
    abs(X1 - X2) - abs(Y1 - Y2) == 0,
    row(X1), row(X2),
    column(Y1), column(Y2),
    X1 != X2,
    Y1 != Y2.
```