T-79.154 **Syksy 2001**

Logiikka tietotekniikassa: erityiskysymyksiä II

Laskuharjoitus 5

Ratkaisut

1. Ratkaisussa käytetään predikaatteja:

   - $at(V, L, I)$ — imuri V on huoneessa L hetkellä I.
   - $clean(L, I)$ — huone L on siisti hetkellä I.
   - $move(V, F, T, I)$ — imuri V siirretään huoneesta F huoneeseen T hetkellä I.
   - $suction(V, L, I)$ — imuri V siivoaa huoneen L hetkellä I.
   - $next\_to(L_1, L_2)$ — huone L_2 on huoneen L_1 vieressä.

   Suunniteluoperaattorien mallintamisessa käytetään periaatetta, jonka mukaan tapahtuma implikoi sekä esi- että jälkiehtonsa.

   Operaattorien mallintamiseen tarvitaan seuraavat lauseskeemat:

   (a) **Suction**

   - Jos imuri v siivoaa huoneen l, se muuttuu siistiksi:

   $$\forall v \forall l \forall i (at(v, l, i) \wedge suction(v, l, i) \rightarrow clean(l, i + 1))$$

   - Imuri voi siivota vain huoneen, jossa se on:

   $$\forall v \forall l \forall i (suction(v, l, i) \rightarrow at(v, l, i))$$

   - Imuri ei voi liikkua siivotessaan:

   $$\forall v \forall l \forall i (suction(v, l, i) \rightarrow at(v, l, i + 1))$$

   - Puhtaita huoneita ei siivota:

   $$\forall v \forall l \forall i (suction(v, l, i) \rightarrow \neg clean(l, i))$$

   (b) **Move**

   - Imuri voi siirtyä pois ainoastaan huoneesta, jossa se on:

   $$\forall v \forall f \forall t \forall i (move(v, f, t, i) \rightarrow at(v, f, i))$$

- Imuri siirtyy kohteeseensa:

$$\forall v \forall f \forall t \forall i (move(v, f, t, i) \rightarrow at(v, t, i + 1))$$

- Imuri voi siirtyä ainoastaan viereiseen huoneeseen:

$$\forall v \forall f \forall t \forall i (move(v, f, t, i) \rightarrow next\_to(f, t))$$

Lisäksi tarvitaan koko joukko *kehysaksioomia*, jotka pitävät maailmankuvan konsistenttina ja yllätyksettömänä:

- Imuri ei voi olla yhtä aikaa kahdessa paikassa:

$$\forall v \forall l_1 \forall l_2 \forall i (l_1 \neq l_2 \rightarrow \neg at(v, l_1, i) \vee \neg at(v, l_2, i))$$

- Jos huone muuttuu siistiksi, on sen joku siivonnut:

$$\forall l \forall i (\neg clean(l, i) \wedge clean(l, i + 1) \rightarrow \exists v (suction(v, l, i)))$$

- Jos imuri ilmestyy jonnekin, se on siirtynyt normaalilla tavalla:

$$\forall v \forall t \forall i (\neg at(v, t, i) \wedge at(v, t, i + 1) \rightarrow \exists f (move(v, f, t, i)))$$

Suunnittelun alkutilassa kaikki huoneet ovat likaisia, imureita on vain yksi ja se on olohuoneessa:

$$\forall l (\neg clean(l, 0)) \wedge at(c, oh, 0)$$

Relaatio $next\_to$ täytyy määritellä kokonaan, eli lausejoukkoon täytyy lisätä $next\_to(x, y)$ aina, kun x:n ja y:n välillä on ovi, ja $\neg next\_to(x, y)$, jos ovea ei ole.

Maalitila on yksinkertaisesti:

$$\forall l (clean(l, g)),$$

missä g on askelten määrä suunnitelmassa.

2. **SATPLAN**

Ylläoleva lausejoukko kelpaa melkein sellaisenaan SATPLAN-järjestelmälle. Ainoastaan $move/4$ operaattori täytyy jakaa kahteen osaan, $move\_from/3$ ja $move\_to/3$ operaattoreiksi, sillä SATPLAN hyväksyy vain kolmeargumenttisia predikaatteja.

Esimerkiksi lause $\forall v \forall l \forall i (suction(v, l, i) \rightarrow at(v, l, i))$ syötetään SATPLANille seuraavasti:

```
  for (i = 0; i < goal; i++) {
    All(v, cleaner, 1,
        All( l, room, 1,
             Disj(
                  Not( L3( "suction", v, l, i));
                  L3( "at", v, l, i))));
  }
```

Siirtymistä koskevat lauseet muuttuvat muotoon:

- Imuri voi siirtyä pois ainoastaan huoneesta, jossa se jo on:

$$\forall v \forall f \forall i (move\_from(v, f, i) \rightarrow at(v, f, i))$$

- Imuri siirtyy kohdehuoneeseen:

$$\forall v \forall t \forall i (move\_to(v, t, i) \rightarrow at(v, t, i + 1))$$

- Jokaiseen siirtymiseen liittyy sekä alku, että loppu:

$$\begin{array}{l} \forall v \forall f \forall i (move\_from(v, f, i) \rightarrow \exists t (move\_to(v, t, i))) \\ \forall v \forall t \forall i (move\_to(v, f, i) \rightarrow \exists f (move\_from(v, t, i))) \end{array}$$

Ylläolevat lauseet sallivat imurin siirtymisen mihin huoneeseen tahansa. Yksinkertaisin keino pakottaa imuri käyttämään ovia on käyttää SATPLANin tukea kiinteiden relaatioiden määrittelemiseen. Tällöin esimerkiksi kolmas lauseista kirjoitetaan seuraavasti:

```
All(v, cleaner, 1,
    All(f, room, 1,
        Disj(
             Not( L3( "move-from", v, f, i));
             All(t, room, reln2(neighbours, f, t),
                 L3( "move-to", v, t, i)))));
```

SATPLANin dokumentaatio kertoo asiasta lisää.