T-79.179 Rinnakkaiset ja hajautetut digitaaliset järjestelmät

Paikka–siirtymä-verkot

Marko Mäkelä

22. tammikuuta 2002

Paikka–siirtymä-verkot

Yleistetään edellisellä luennolla esitetyn merkkipelin sääntöjä:

- Paikassa voi olla useita merkkejä, jotka voidaan tulkita resursseiksi.
- Paikan ja siirtymän välillä voi olla useita syöttö- ja tuloskaaria. Näiden kaarten lukumäärä esitetään yksittäisen kaaren painona.
- Siirtymä on virittynyt, jos kussakin siihen kytketyssä syöttöpaikassa on vähintään syöttökaaren painon verran merkkejä.
- Kun virittynyt siirtymä laukeaa, sen syöttöpaikoista poistetaan syöttökaarten ja tulospaikkoihin lisätään tuloskaarten painon verran merkkejä.

Verkko

Kolmikkoa $N = \langle S, T, F \rangle$ kutsutaan *verkoksi*, jos

- paikkojen joukko S ja siirtymien joukko T ovat erilliset ($S \cap T = \emptyset$) ja
- *vuorelaatio* F toteuttaa ehdon $F \subseteq (S \times T) \cup (T \times S)$.

Huomattakoon, että verkon N siirtymälle $t \in T$ pätee $\langle s, t \rangle \in F$, jos s on t:n syötepaikka, ja $\langle t, s' \rangle \in F$, jos s' on t:n tulospaikka.

Olkoon $a \in S \cup T$. Joukkoa ${}^\bullet a = \{a' \mid \langle a', a \rangle \in F\}$ kutsutaan alkion a *esijoukoksi* ja joukkoa $a^\bullet = \{a' \mid \langle a, a' \rangle \in F\}$ *jälkijoukoksi*.

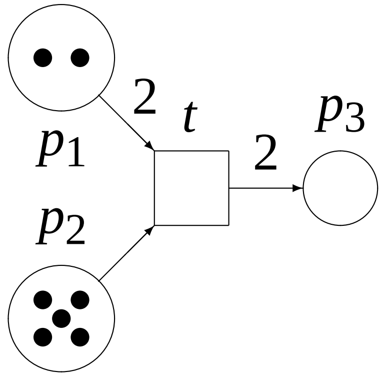
Paikka–siirtymä-verkko

Nelikko $\langle S, T, F, W \rangle$ on *paikka–siirtymä-verkko* (Stellen/Transition-Netz, *Place/Transition Net*), jos

- $\langle S, T, F \rangle$ on äärellinen verkko ja
- kuvaus $W : F \rightarrow (\mathbb{N} \setminus \{0\})$ määrää siirtymien *kaaripainot*.

Vastaava viisikko $\langle S, T, F, W, M_0 \rangle$ on *paikka–siirtymä-järjestelmä*, jossa $M_0 : S \rightarrow \mathbb{N}$ on järjestelmän alkumerkintä.

Paikka–siirtymä-verkko: esimerkki



Kuva esittää paikka–siirtymä-järjestelmää $\langle S, T, F, W, M_0 \rangle$, jolle on määritelty:

- $\langle S, T, F \rangle = \langle \{p_1, p_2, p_3\}, \{t\}, \{\langle p_1, t\rangle, \langle p_2, t\rangle, \langle t, p_3\rangle\}\rangle$,
- $W = \{\langle p_1, t\rangle \mapsto 2, \langle p_2, t\rangle \mapsto 1, \langle t, p_3\rangle \mapsto 2\}$ ja
- $M_0 = \{p_1 \mapsto 2, p_2 \mapsto 5, p_3 \mapsto 0\} = \langle 2, 5, 0\rangle$.

Paikka–siirtymä-verkon laukeamissääntö

Olkoon $\langle S,T,F,W\rangle$ paikka–siirtymä-verkko ja M verkon $\langle S,T,F\rangle$ merkintä, $M : S \to \mathbb{N}$.

1. Siirtymä $t \in T$ on *M-virittynyt*, $M[t\rangle$, jos $\forall s \in {}^{\bullet}t : M(s) \geq W(s,t)$. (Laukeamisehto)

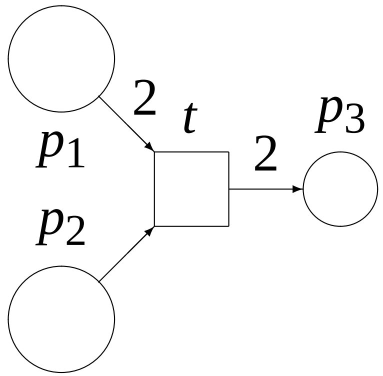
2. M-virittynyt siirtymä t voi *laueta* tuottaen *seuraajamerkinnän* $M' := [M\rangle t$, $M'(s) = M(s) - W'(s,t) + W'(t,s)$, missä $W' : (S \times T) \cup (T \times S) \to \mathbb{N}$:

   $$\begin{cases} W'(f) = W(f) & \forall f \in F \\ W'(f) = 0 & \text{muulloin.} \end{cases}$$ (Laukeamissääntö)

3. Joukon $\mathcal{M}$ *seuraajamerkinnät* ovat $\mathcal{M}[\rangle := \bigcup_{M \in \mathcal{M}} \bigcup_{t \in T} \left\{ \{[M\rangle t\} \;\middle|\; M[t\rangle \right\}$.

4. $\mathcal{M}$:stä *saavutettavat merkinnät* ovat $[\mathcal{M}\rangle := \mathcal{M}[\rangle^* = \mathcal{M} \cup \mathcal{M}[\rangle \cup \mathcal{M}[\rangle[\rangle \cup \ldots$.

Paikka–siirtymä-verkon laukeamissääntö: esimerkki



Merkintä M	$M[t\rangle$	$\{M\}[\rangle$
$\{p_1 \mapsto 2, p_2 \mapsto 5, p_3 \mapsto 0\}$	virittynyt	$\{\{p_1 \mapsto 0, p_2 \mapsto 4, p_3 \mapsto 2\}\}$
$\{p_1 \mapsto 0, p_2 \mapsto 4, p_3 \mapsto 2\}$	epävireessä	$\emptyset$
$\{p_1 \mapsto 1, p_2 \mapsto 5, p_3 \mapsto 0\}$	epävireessä	$\emptyset$

Esimerkki: monisuoritinjärjestelmä (1/4)

Mallinnetaan karkeasti järjestelmää, jossa on 5 suoritinta, 3 yhteistä muistia ja 2 väylää. Suoritinten tilat kuvataan paikoiksi, joissa on aina yhteensä 5 merkkiä:

p_1: valmis suorittamaan käskyä

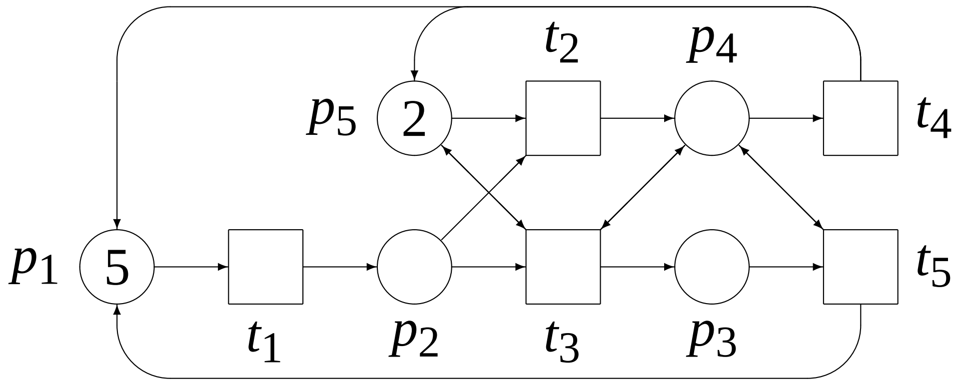
p_2: odottaa väylän vapautumista

p_3: odottaa muistin vapautumista

p_4: väylällä

Paikka p_5 pitää kirjaa muistien ja väylien tilasta.

Esimerkki: monisuoritinjärjestelmä (2/4)



t_1	saantipyyntö (uuden käskyn suoritus)
t_2	haluttu muisti on vapaana
t_3	haluttu muisti on varattuna; jonotus
t_4	vapaana olevan muistin käsittely ja paluu
t_5	varattuna olleen muistin käsittely ja paluu

Esimerkki: monisuoritinjärjestelmä (3/4)

Mallista nähdään muun muassa seuraavat asiat:

- Vaikka yksi suoritin on varannut muistin, toista väylää pitkin voidaan viedä saantipyyntöjä jonottamaan.

- Jos kaksi suoritinta valitsee vapaan muistin, muut suorittimet jäävät odottamaan.

- Kaikille verkon saavutettaville merkinnöille $M \in \{M_0\}[\rangle$ pätevät seuraavat *invariantit*:

$$
\begin{aligned}
M(p_1)+M(p_2)+M(p_3)+M(p_4) &= M_0(p_1)+M_0(p_2)+M_0(p_3)+M_0(p_4) \\
M(p_4)+M(p_5) &= M_0(p_4)+M_0(p_5)
\end{aligned}
$$

Esimerkki: monisuoritinjärjestelmä (4/4): huomioita

- Suorittimia, väyliä ja muisteja ei ole yksilöity (niillä ei ole *identiteettiä*).
- Voidaan tutkia mielivaltaista suoritinmäärää pelkkää alkumerkintää muuttamalla.
- Voidaanko väyliä lisätä yhtä helposti?
- Muistien määrään malli ei ota kantaa.
- Määrittelemällä siirtymille laukeamistodennäköisyydet tai -ajat voidaan tarkastella järjestelmän suorituskykyä, kuten sitä, montako suoritinta keskimäärin odottaa väylän vapautumista tai kauanko yhden pyynnön palveleminen kuluttaa aikaa.

Saavutettavuusgraafi

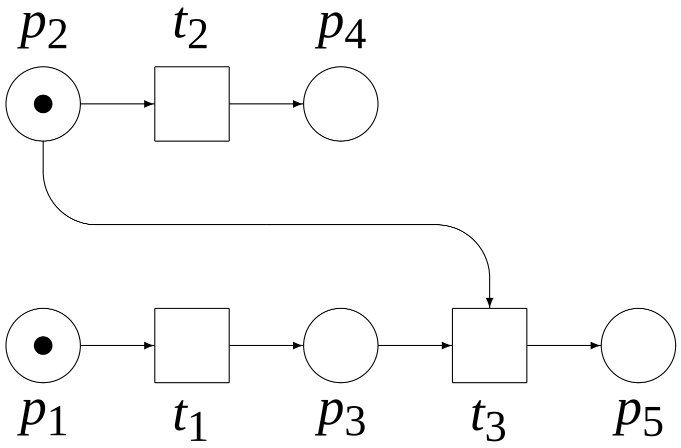
Paikka–siirtymäjärjestelmän $\langle S, T, F, W, M_0 \rangle$ *saavutettavuusgraafi (reachability graph)* on juurellinen, suunnattu ja kaarimerkitty graafi $G = \langle V, E \subseteq V \times T \times V, v_0 \in V \rangle$, jossa

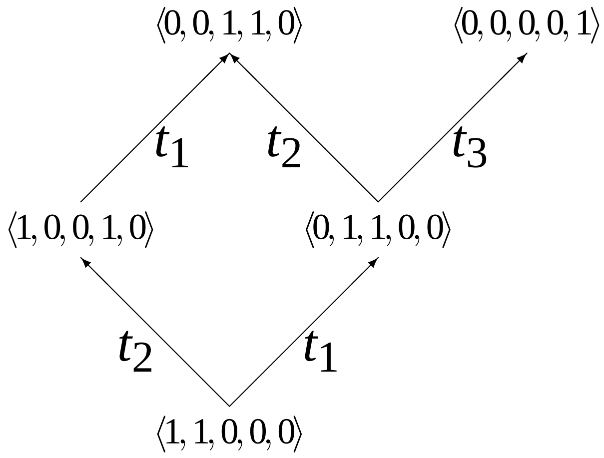
- $V = [\{M_0\}\rangle$ on graafin solmujen joukko,
- $v_0 = M_0$ on graafin juurisolmu ja
- $E = \left\{ \langle M, t, M' \rangle \;\middle|\; M \in V \wedge M' = [M\rangle t \right\}$ on graafin kaarten joukko.

Merkinnästä toiseen johtavat kaaret on merkitty kulloinkin laukeavan siirtymän nimellä.

Käytännössä kaikki analysointityökalut ja -menetelmät edellyttävät saavutettavuusgraafin laskentaa. Graafi esittää rinnakkaisen ja hajautetun järjestelmän tavallisena tilakoneena. Siitä ei voi suoraan erottaa toisistaan riippumattomia ja rinnakkaisia toimintoja.

Saavutettavuusgraafi: esimerkki





- Kunkin kaaren paino on 1.
- Usein kirjoitetaan $\{p_1, p_2\}$ tarkoitettaessa merkintää $\langle 1,1,0,0,0\rangle$.
- Toisistaan riippumattomien siirtymien t_1 ja t_2 kaikki *lomitukset* esiintyvät graafissa. Jos tarkasteltaisiin myös samanaikaisia siirtymiä, graafiin merkittäisiin myös kaari $\{p_1, p_2\} \xrightarrow{t_1,t_2} \{p_3, p_4\}$.

Saavutettavuusgraafin laskeminen

SAAVUTETTAVUUSGRAAFI$(\langle S, T, F, W, M_0 \rangle)$

$G : \langle V, E, v_0 \rangle \leftarrow \langle \{M_0\}, \emptyset, M_0 \rangle$;
$Markings$: stack $\leftarrow \langle M_0 \rangle$;
while $Markings \neq \langle\rangle$
do $M \leftarrow Markings.\mathrm{pop}()$;
  for $t \in \mathrm{enabled}(M)$
  do $M' \leftarrow \mathrm{fire}(M, t)$;
    if $M' \notin V$
      then $V \leftarrow V \cup \{M'\}$
        $Markings.\mathrm{push}(M')$;
    $E \leftarrow E \cup \{\langle M, t, M' \rangle\}$;
return G;

Algoritmi käyttää kahta funktiota:

- $\mathrm{enabled}(M) := \{t \mid M[t\rangle\}$
- $\mathrm{fire}(M, t) := [M\rangle t$

Vaihtamalla hakupino hakujonoksi syvyyshaku muuttuu leveyshauksi. Leveyshaku löytää lyhimmän siirtymäpolun alkumerkinnästä virheelliseen merkintään. Joissakin sovelluksissa tarvitaan syvyyshakua.

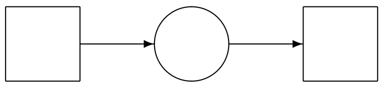
Saavutettavuusgraafin koko

Montako solmua paikka–siirtymä-järjestelmän saavutettavuusgraafissa voi olla? Ehdoton yläraja on järjestelmän mahdollisten merkintöjen määrä.

Jos järjestelmän tiedetään olevan sellainen, että paikassa p_i voi olla enintään m_i merkkiä, erilaisia merkintöjä (ja graafin solmuja) voi olla enintään $\prod_{i=1}^{n} m_i = m_1 \cdot m_2 \cdot \ldots \cdot m_n$.

Saavutettavuusgraafissa voi olla enintään $|E| \leq |V| \cdot |T| \cdot |V|$ kaarta, yksi kutakin tilaparin ja siirtymän yhdistelmää kohden.

Yleisessä tapauksessa järjestelmän paikoilla ei välttämättä ole luontaista enimmäismerkkimäärää, ja saavutettavia merkintöjä voi olla äärettömän monta, kuten alla:



Saavutettavuusgraafin äärellisyys

Jos paikka–siirtymä-järjestelmän saavutettavissa merkinnöissä paikoissa voi esiintyä enintään k merkkiä, järjestelmän sanotaan olevan k-*turvallinen*.

k-turvallisella järjestelmällä on enintään $(k+1)^n$ merkintää; 1-turvallisella 2^n.

Järjestelmän saavutettavuusgraafi voidaan todeta äärettömäksi ottamalla käyttöön *peittävyyden* käsite ($M \leq M'$ jos $\forall s \in S : M(s) \leq M'(s)$) ja muuttamalla esitettyä laskenta-algoritmia siten, että jos laskettu seuraajamerkintä peittää jonkin siihen johtavalla siirtymäpolulla olevan merkinnän, graafi todetaan äärettömäksi.

Laskennallinen vaativuus

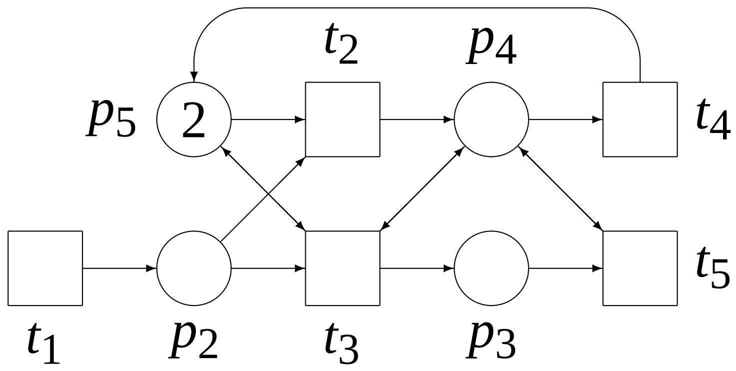
Vaikka paikka–siirtymä-järjestelmällä voi olla äärettömän monta saavutettavaa merkintää, saavutettavuusongelma (se, onko annetun ehdon mukainen merkintä saavutettavissa alkumerkinnästä) on ratkeava (1-turvallisille järjestelmille PSPACE-täydellinen ja muille $\mathit{EXPSPACE}$-kova).

Eräs paikka–siirtymä-verkkojen laajennus sisältää *estokaaret*. Paikasta siirtymään johtava estokaari estää siirtymän virittymisen, jos kyseisessä paikassa on vähintään estokaaren painon verran merkkejä. Tällä tavalla laajennettu järjestelmä kykenee simuloimaan Turingin konetta, joten sen saavutettavuusongelma on ratkeamaton.

Saavutettavuusongelman yleinen ratkeavuudella on enemmän teoreettista kuin käytännön merkitystä. Käytännössä tulee ongelmia, jos saavutettavien merkintöjen tai tilojen tarkistus kestää päiviä tai jos muisti tai levytila loppuvat kesken. Joitakin ongelmia voidaan ratkaista tehokkaammin, jos järjestelmä on kuvattu teorian kannalta "helpommin".

Komplementtipaikat

Joskus on tarvetta rajoittaa paikka–siirtymä-järjestelmän paikkoihin tulevien merkkien määrää. Voidaan esimerkiksi rajoittaa tarkastelu sellaisiin tapauksiin, joissa järjestelmä voi ottaa palveltavakseen enintään k asiakasta kerrallaan. Luontevinta on määritellä merkeille "suljettu kierto" erityisen *komplementtipaikan* avulla.

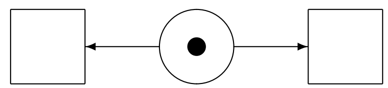


Siirtymä t_1 kuvaa palveltavien asiakkaiden saapumista ja siirtymät t_4 ja t_5 niiden poistumista. t_1:n laukeamista voidaan rajoittaa lisäämällä sille esipaikka, joka on t_4:n ja t_5:n jälkipaikka. Esipaikan alkumerkintä määrää kierrossa olevien asiakkaiden määrän.

Kilpatilanne ja sekaannus

Siirtymät ovat *riippumattomia* jos ja vain jos niillä ei ole yhteisiä esi- eikä jälkipaikkoja. Tällöin ne voivat laueta mielivaltaisessa järjestyksessä tai rinnakkain. Siirtymät ovat *samanaikaiset* (concurrent) .

Yhteisiä esipaikkoja jakavien siirtymien välillä voi syntyä *kilpatilanne* (conflict):

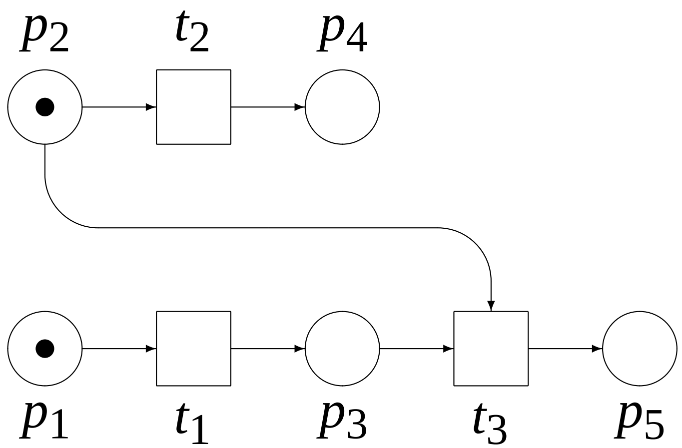


Vain toinen siirtymistä voi laueta. Kilpatilanne kuvaa järjestelmän epädeterministisyyttä.

Sellaista tilannetta, jossa samanaikaisten siirtymien laukeaminen voi vaikuttaa ilmenevien kilpatilanteiden määrään, kutsutaan *sekaannukseksi* (confusion).

Sekaannus: esimerkki

Oletetaan, että järjestelmää tarkkailee kaksi havainnoitsijaa (H_1 ja H_2), jotka havaitsevat yhden siirtymän laukeamisen kerrallaan.

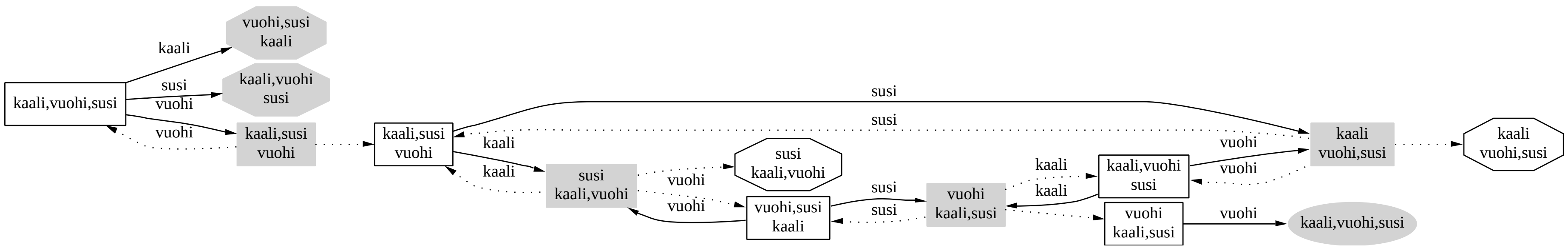


- H_1 havaitsee t_1:n ja t_2:n laukeavan tässä järjestyksessä.
- H_2 havaitsee t_2:n ja t_1:n laukeavan tässä järjestyksessä.
- H_1 havaitsee t_2:n kilpailevan t_3:n kanssa, mutta H_2 ei havaitse kilpatilannetta.

Loppukevennys: pähkinöiden ratkaiseminen

Paikka–siirtymä-järjestelmillä voidaan kuvata muitakin kuin rinnakkaisia järjestelmiä. Eräs vanha ongelma on kaalinpään, vuohen ja suden siirtäminen vesistön yli. Vene kantaa soutajan ja yhden matkustajan. Ollessaan vartijatta vuohi syö kaalinpään ja susi vuohen.

Tehdään malli, joka kuvaa mahdolliset toimenpiteet (vesistön ylittäminen kuorman kanssa tai ilman) ja kielletyt tilat (joissa syöminen on mahdollista). Mallin saavutettavuusgraafi sisältää ratkaisupolun, jos ratkaisu on olemassa.



Marko Mäkelä