Arbore de acoperire

Metoda arborelui de acoperire este o tehnica de analiza a Retelei Petri des utilizata in practica intrucat gradul de eficienta este ridicat in numeroase cazuri de utilizare. Astfel, aceasta tehnica se bazeaza pe construirea unui arbore in care nodurile sunt etichetate cu cate o stare Petri, iar arcele reprezinta tranzitii.

Metoda in cauza este ralativ simpla, intrucat ideea fundamentala consta in enumerarea tuturor marcajelor posibile accesibile din marcajul initial M_0. Asadar, pornind de la marcajul initial M_0, se poate construi setul de accesibilitate prin executia tuturor tranzitiilor posibile activate in toate marcajele posibile accesibile din marcajul intial. Dar procedand in aceste fel, nu cumva apare posibilitatea ca arborele sa se extinda la infinit din cauza starilor duplicate sau a caracterului nelimitat al retelei in sine? Raspunsul este nu! Aici trebuie sa fim atenti! Exista doi pasi pe care trebuie sa ii respectam astfel incat sa constrangem arborele sa fie limitat si anume:

Pas 1. Eliminam marcajele duplicate !! Daca pe drumul dintre marcajul initial M_0 si un marcaj M apare un marcaj M' identic cu marcajul M, atunci marcajul M, repetandu-se, devine nod terminal. Dar ce semnifica acest lucru? Aparitia marcajului dublat/terminal indica faptul ca toate marcajele M_i posibil accesibile din marcajul original M au fost deja adaugate arborelui. Voila! Un prim pas a fost realizat! Simplu, nu?

Pas 2. In cazul retelelor nelimitate, problema nu a mai fost atat de simpla si a trebuit sa se introduca un nou simbol care ajuta in rezolvarea acestui incovenient. Astfel, pentru a pastra arborele finit, a aparut simbolul ω, pe care il interpretam drept infinit. Presupunem ca toata lumea stie proprietatile operatorului infinit si anume ca orice intreg ai adauga/scadea din infinit tinde in continuare catre infinit. Asadar, daca pe drumul de la marcajul initial M_0 catre marcajul curent M apare un marcaj M' ale carui intrari(numar de jetoane) sunt mai mici sau egale cu intrarile corespunzatoare marcajului M, atunci intrarile lui M, care sunt strict mai mari decat intrarile lui M', se vor inlocui cu ω. Felicitari! Acum singura modalitate de a ajunge la un arbore nelimitat este daca gresesti intr-o maniera deplorabila! ☺

Inainte de a prezenta algoritmul de constructie al arborelui finit de acoperire, reamintim urmatoarele concepte si notatii:

1. **Marcajul initial** – este primul marcaj al arborelui(radacina arborelui); reprezinta starea initiala in care se gaseste Reteaua Petri;
2. **Marcajul terminal** – este orice nod din care nu se mai poate efectua nicio tranzitie;
3. **Marcajul duplicat** – este nodul identic cu un altul deja existent in arbore;
4. **Marcajul dominant** – fiind doua marcaje x si y, putem spune ca „x il domina pe y” ($x >_d y$) daca si numai daca:
   - $x(p_i) \geq y(p_i), oricare\ ar\ fi\ i = 1, \dots, n;$
   - $x(p_i) > y(p_i), pentru\ cel\ putin\ un\ i = 1, \dots, n.$

5. **Simbolul ω** – daca „x il domina pe y", atunci pentru toti i unde $x(p_i) > y(p_i)$, se inlocuieste valoarea lui $x(p_i)$ cu ω.

Algoritmul de constructie al arborelui de acoperire

Pas 1: Se initializeaza $x = x_0$ (stare initiala);

Pas 2: Pentru fiecare nod nou, x, se evalueaza tranzitia functiei $f(x, t_j)$ pentru toate $t_j \in T$:

Pas 2.1: Daca $f(x, t_j)$ este nedefinita pentru toate $t_j \in T$ (nicio tranzitie nu este valida din marcajul x), atunci se marcheaza x ca fiind nod terminal.

Pas 2.2: Daca $f(x, t_j)$ este definita pentru cateva $t_j \in T$, atunci se creeaza un nod nou $x' = f(x, t_j)$. In caz de necesitate, se ajusteaza marcajul nodului x' in felul urmator:

Pas 2.2.1: Daca $x(p_i) = \omega$ pentru cateva p_i, se seteaza $x'(p_i) = \omega$.

Pas 2.2.2: Daca exista un nod y pe calea dintre nodul initial x_0 (inclus la randul lui) si x astfel incat $x' >_d y$, se seteaza $x'(p_i) = \omega$ pentru toate p_i unde $x'(p_i) > y(p_i)$.

Pas 2.2.3: In caz contrar, $x'(p_i)$ este obtinut conform $f(x, t_j)$.

Pas 3: Daca toate nodurile noi au fost marcate ca fiind terminale sau duplicate, STOP!

Proprietati

1. **Marginire** – O pozitie p_i este marginita pentru marcajul initial M_0 daca oricare ar fi marcajul corespunzator nodului p_i, exista un numar natural m astfel incat, oricare ar fi evolutia retelei, $m(p_i) < m$; Astfel, o Retea Petri este marginita pentru un marcaj initial M_0 daca toate nodurile sale sunt marginite pentru marcajul initial.

   Cu alte cuvinte, Reteaua Petri este marginita daca numarul de jetoane din fiecare marcaj este finit (nu apare ω).

2. **Viabilitate** – O tranzitie T_j este viabila pentru marcajul initial M_0 daca, pentru orice marcaj aferent retelei (M), exista o secventa de tranzitii S care porneste din M si contine T_j. De asemenea, o tranzitie este cvasiviabila pentru M_0 daca exista cel putin o secventa de tranzitii S care porneste din M_0 si contine T_j; Asadar, o Retea Petri este viabila pentru marcajul initial M_0 daca toate tranzitiile sale sunt viabile pentru M_0. Analog, o Retea Petri este cvasiviabila daca toate tranzitiile sale sunt cel putin cvasiviabile pentru M_0.

   Cu alte cuvinte, Reteaua Petri este viabila daca putem executa toate tranzitiile plecand din orice marcaj (nu neaparat direct, trebuie sa existe un drum). De asemenea, putem verifica in mod facil cvasiviabilitatea prin intermediul arborelui de acoperire. Astfel, daca in arborele de acoperire se regasesc toate tranzitiile din retea, atunci reteaua este cel putin cvasiviabila.

3. **Blocaje** – Un blocaj este un marcaj pentru care nicio tranzitie nu mai este valida. Asadar, ne uitam in arbore si daca observam ca dintr-un marcaj nu pleaca nicio tranzitie (sageata) inseamna ca acel marcaj reprezinta un blocaj. In cazul contrar in care nu se gaseste niciun blocaj, se afirma ca Reteaua Petri este lipsita de blocaje.

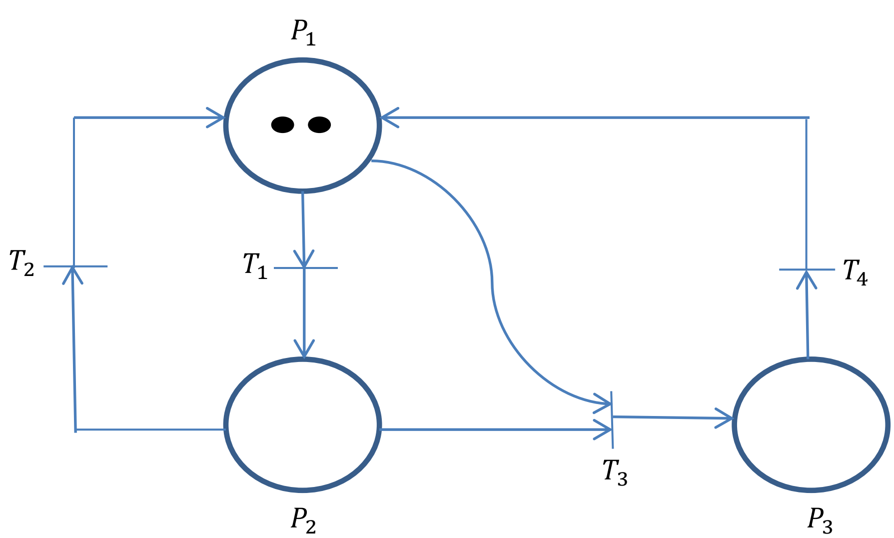
4. **Reinitializabilitate** – O Retea Petri are o stare de primire M_a pentru un marcaj initial M_0 daca pentru toate marcajele accesibile M_i exista o secventa de tranzitii S astfel incat $M_i(S) \to M_a$. Asadar, o Retea Petri este reinitializabila pentru un marcaj initial M_0 daca M_0 este stare de primire.

   Cu alte cuvinte, reteaua este reinitializabila daca din fiecare marcaj exista drum catre M_0.

5. **Conflicte structurale si efective** – a) O Retea Petri prezinta un conflict structural daca o pozitie p_i este intrare pentru cel putin doua tranzitii, T_j si T_k. Cu alte cuvinte, daca dintr-o pozitie pleaca cel putin doua sageti, atunci devine conflict structural cu alegere libera(free choice). Daca orice pozitie are cel mult o tranzitie de iesire, atunci reteaua petri este fara conflict.

   b) Un conflict efectiv este reprezentat de un conflict structural care intr-un marcaj are valide cel putin doua tranzitii ale conflictului dar pentru care numarul de jetoane din pozitia de conflict este insuficient pentru a executa toate tranzitiile valide ale conflictului.

Exemplu:



Arbore de acoperire:

$$\begin{tikzcd}
M_0 = \langle 2|0|0\rangle^T \arrow[r, "T_1"] & M_1 = \langle 1|1|0\rangle^T \arrow[r, "T_3"] \arrow[d, "T_2"] \arrow[dr, "T_1"] & M_2 = \langle 0|0|1\rangle^T \arrow[r, "T_4"] & M_4 = \langle 1|0|0\rangle^T \arrow[r, "T_1"] & M_5 = \langle 0|1|0\rangle^T \arrow[r, "T_2"] & M_4 \\
 & M_0 & M_3 = \langle 0|2|0\rangle^T \arrow[r, "T_2"] & M_1 & &
\end{tikzcd}$$

1. *RP este marginita deoarece numarul de jetoane este finit. (nu apare ω si nr max = 2.*
2. *Nu exista blocaje. Fiecare marcaj are cel putin o tranzitie valida.*
3. *RP e cvasiviabila, apar toate tranzitile in arbore, dar plecand din M_2 nu avem cum sa trecem prin T_3.*
4. *RP nu e reinitializabila deoarece din M_2 nu se poate ajunge in M_0.*
5. *Conflicte structurale :*
   - $< p_1, \{T_1, T_3\} >$
   - $< p_2, \{T_2, T_3\} >$

   Ambele devin efective in M_1.