Modell ellenőrzés - Model checking

Fülöp Zoltán

Szegedi Tudományegyetem
Természettudományi Kar
Számítástudomány Alapjai Tanszék

2010. január

Modell ellenőrzés (vizsgálat) - Model checking

- Egyidejű (időben párhuzamosan működő), újraéledő hardver-szoftver rendszerek működésének helyességét vizsgáljuk. (Concurrent, reactive systems)
- Teszteléssel nem vizsgálható, mivel a rendszer működése nem reprodukálható.
- Felállítjuk a rendszer egy M *modelljét*, melynek véges sok (általában nagyon sok) állapota van és amely az állapotait diszkrét időpillanatonként változtatja.
- A *temporális logika* egy f formulájával megfogalmazzuk a rendszer valamely (kívánatos vagy nemkívánatos) tulajdonságát.
- Azt ellenőrizzük, hogy az M modell mely s állapotai elégítik ki f-et, vagyis mely s-ekre igaz $M, s \models f$.

Tankönyvek

- Michael R. A. Huth, Mark D. Ryan: Logic in computer science, Cambridge University Press, 2000.
- Edmund M. Clarke, O. Grumber, D. A. Peled: Model Checking, MIT Press, 1999.
- Vardi: An Automata-Theoric Approach to Linear Temporal Logic, LNCS 1043, Springer-Verlag, 1996.
- Stirling: Modal and Temporal Properties of Processes, Springer-Verlag, 2001.
- C. Baier, J.-P. Katoen, Principles of Model Checking, The MIT Press, 2008
- Formális módszerek az informatikában (Szerk. Pataricza András), TYPOTEX, Budapest 2004.

Kripke struktúrák: az egyidejű rendszerek modelljei

Legyen AP *atomi propozíció*k (állítások) véges halmaza.

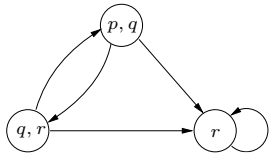
Definíció. AP *feletti Kripke struktúrá*nak nevezünk egy $M = (S, S_0, R, L)$ rendszert, ahol

- S az állapotok véges halmaza
- $S_0 \subseteq S$ a *kezdőállapotok* halmaza
- $R \subseteq S \times S$ egy totális reláció, az *átmeneti reláció* $((\forall s \in S)(\exists s' \in S) : sRs')$.
- $L : S \to \mathcal{P}(AP)$ egy *címkézőfüggvény*, mely minden $s \in S$ állapothoz hozzárendeli azon $L(s)$ atomi állításokat, melyek igazak s-ben.

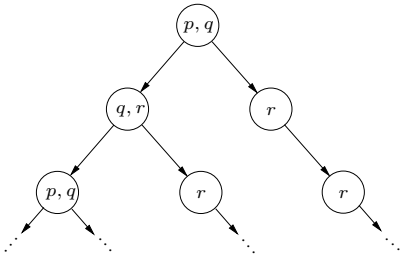
Definíció. Egy $s \in S$ *állapotból kiinduló úton* egy olyan $s_0, s_1, s_2, \ldots$ *végtelen sorozat*ot értünk, melyre $s = s_0$ és bármely $i \geq 0$ esetén $s_i R s_{i+1}$ $((s_i, s_{i+1}) \in R)$.

Példa Kripke struktúrára

$AP = \{p, q, r\}$



Ugyanez ”kihajtogatva”:



Végtelen fa, utak.

Concurrent rendszer, mint Kripke struktúra

- $V = \{v_1, \ldots, v_n\}$ a *változók* halmaza, melyek értékeiket egy véges D halmazból veszik fel. (Legtöbbször $D = \{0, 1\} = \{hamis, igaz\}$.)
- Atomi állítások: $v = d$ alakú egyenlet, ahol $v \in V$ és $d \in D$. Amennyiben $D = \{0, 1\}$, akkor $v = 1$ helyett v-t, $v = 0$ helyett $\neg v$-t írunk.
- *Állapot*ok: az $s : V \to D$ kiértékelések.
- Legyen adott egy φ_0 elsőrenű formula V változókkal. (Minden $s : V \to D$ állapotra $\varphi_0[s] = 1$ vagy $\varphi_0[s] = 0$.)
- Legyen $V' = \{v'_1, \ldots, v'_n\}$ és $\varrho(V, V')$ egy elsőrendű formula a $V \cup V'$ változókkal. (Akkor bármely két $s, s' : V \to D$ állapotra $\varrho(V, V')[s, s'] = 1$ vagy $\varrho(V, V')[s, s'] = 0$, ahol $s'(v'_i) = s'(v_i),\ 1 \leq i \leq n$.)

Concurrent rendszer, mint Kripke struktúra

Definíció. A φ_0-ból és ϱ-ból származtatott Kripke struktúrán azt az $M = (S, S_0, R, L)$ Kripke struktúrát értjük, melyre

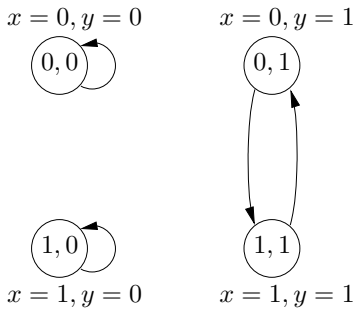
- $S = \{s \mid s : V \to D\}$,
- $S_0 = \{s \in S \mid \varphi_0[s] = 1\}$,
- bármely $s, s' \in S$ esetén $sRs' \iff \varrho[s, s'] = 1$ ($s'(v'_i) = s'(v_i)$ minden $1 \leq i \leq n$-re),
- bármely $s \in S$-re $L(s) = \{v = d \mid s(v) = d\}$.

Concurrent rendszer, mint Kripke struktúra

Példa. Legyen $V = \{x, y\}, D = \{0, 1\}$,

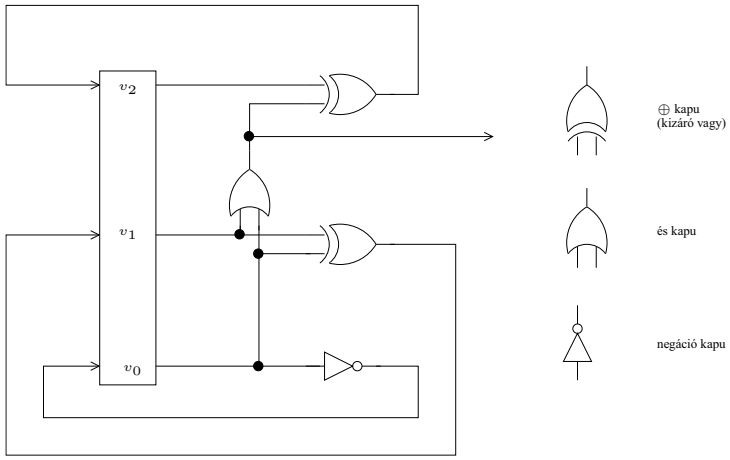
- $\varphi_0 = ((x = 1) \wedge (y = 1))$
- $\varrho = ((x' = (x + y) \bmod 2) \wedge (y' = y))$

Akkor M a következő Kripke struktúra:



Példák Kripke struktúrákra

1) Digitális hálózatok



$V = \{v_0, v_1, v_2\}, \quad D = \{0, 1\}$

Az átmeneti reláció leírása: $v_0' = \neg v_0$, $v_1' = v_0 \oplus v_1$, $v_2' = (v_0 \wedge v_1) \oplus v_2$

Példák Kripke struktúrákra

a) Szinkron működés esetén (minden változó egyszerre változik)

$\varrho_0(V, V') = (v_0' \leftrightarrow \neg v_0)$

$\varrho_1(V, V') = (v_1' \leftrightarrow v_0 \oplus v_1)$

$\varrho_2(V, V') = (v_2' \leftrightarrow (v_0 \wedge v_1) \oplus v_2)$

$\varrho(V, V') = \varrho_0 \wedge \varrho_1 \wedge \varrho_2$

Általában

$\varrho_i = (v_i' \leftrightarrow f_i(V))$, ahol $v_i' = f_i(V),\ 0 \leq i \leq n-1$

$\varrho = \varrho_0 \wedge \ldots \wedge \varrho_{n-1}$

b) Aszinkron működés esetén (egyszerre csak egy változó változik) általában

$\varrho_i = (v_i' \leftrightarrow f_i(V)) \wedge (\bigwedge_{j \neq i} v_j' = v_j)$, ahol $v_i' = f_i(V),\ 0 \leq i \leq n-1$

$\varrho = \varrho_0 \vee \ldots \vee \varrho_{n-1}$

Példák Kripke struktúrákra

P szekvenciális program, aszinkron működés

Először megcímkézzük az utasításokat:

- P bemenetének címkéje m, kimenet címkéje m'
- minden utasítás kimenő címkéje = a következő utasítás bemenő címkéje
- Ha $P = P_1; P_2$ akkor $P^L = P_1^L; l'' : P_2^L$
- Ha $P = \textbf{if}\ \ b\ \textbf{then}\ \ P_1\ \textbf{else}\ \ P_2\ \textbf{endif}$, akkor $P^L = \textbf{if}\ \ b\ \textbf{then}\ \ l_1 : P_1^L\ \textbf{else}\ \ l_2 : P_2^L\ \textbf{endif}$
- Ha $P = \textbf{while}\ \ b\ \textbf{do}\ \ P_1\ \textbf{endwhile}$, akkor $P^L = \textbf{while}\ \ b\ \textbf{do}\ \ l_1 : P_1^L\ \textbf{endwhile}$
- Ha P egyszerű utasítás ($v := e$, $\textbf{skip}$, stb.), akkor $P^L = P$.

Példák Kripke struktúrákra

- V: a program változóinak halmaza
- pc: a lépésszámláló (program counter), egy további speciális változó, melynek értékei a program címkéi
- diszjunkt másolataik: V' és pc'
- segédfüggvény: minden $Y \subseteq V$-re $same(Y) = \bigwedge_{y \in Y} (y' = y)$
- kezdőértékek: $pre : V \to D$

A program Kripke struktúrájának megadása

Kezdőállapot: $\varphi_0 = pre(V) \wedge (pc = m)$

Átmenet: $\varrho(V, pc, V', pc') = C(m, P^L, m')$, ahol $C(l, P^L, l')$ a következő (rekurzív definícióval megadott) elsőrendű formula:

Példák Kripke struktúrákra

$\underline{C(l, P^L, l')}$:

- $C(l, P_1; l'' : P_2, l') = C(l, P, l'') \vee C(l'', P_2, l')$
- $C(l,\ \textbf{if}\ \ b\ \textbf{then}\ \ l_1 : P_1\ \textbf{else}\ \ l_2 : P_2\ \textbf{endif}\ , l') =$
  $(pc = l \wedge pc' = l_1 \wedge b \wedge same(V)) \vee (pc = l \wedge pc' = l_2 \wedge (\neg b) \wedge same(V)) \vee$
  $C(l_1, P_1, l') \vee C(l_2, P_2, l')$
- $C(l,\ \textbf{while}\ \ b\ \textbf{do}\ \ l_1 : P_1\ \textbf{endwhile}\ , l') =$
  $(pc = l \wedge pc' = l_1 \wedge b \wedge same(V)) \vee (pc = l \wedge pc' = l' \wedge (\neg b) \wedge same(V)) \vee$
  $C(l_1, P_1, l)$
- $C(l, v = e, l') = (pc = l \wedge pc' = l' \wedge (v' = e) \wedge same(V - \{v'\}))$
- $C(l,\ \textbf{skip}\ , l') = (pc = l \wedge pc' = l' \wedge same(V))$

Példák Kripke struktúrákra

3) Egyidejű (concurrent) programok

P = **cobegin** $P_1 \parallel P_2 \parallel \cdots \parallel P_n$ **coend** ahol $P_1, P_2, \ldots, P_n$ processzusok, amelyek párhuzamosan hajthatók végre az aszinkron működési elv alapján.

- V_i a P_i változóinak a halmaza
- $V = V_1 \cup \cdots \cup V_n$
- pc_i a P_i programszámlálója, lehet $\perp$ is (definiálatlan)
- pc a programszámláló, értéke lehet $\perp$ is
- diszjunkt másolataik : V_i', V', pc_i', pc'

Példák Kripke struktúrákra

Egyidejű programok címkézése: a szekvenciális programok címkézését kiterjesztjük a következő szabállyal:

Ha $P = \textbf{cobegin}\ \ P_1 \parallel P_2 \parallel \cdots \parallel P_n\ \textbf{coend}$, akkor
$P^L = \textbf{cobegin}\ \ l_1 : P_1^L l_1' \parallel \cdots \parallel l_n : P_n^L l_n'\ \textbf{coend}$

Egyidejű program Kripke struktúrájának megadása

- Kezdőállapot
  $\varphi_0 = pre(V) \wedge pc = m\ \wedge\ \left(\bigwedge_{i=1}^n pc_i = \perp\right)$
- Átmenet: ugyanaz, mint szekvenciális program esetén, ahol a fenti P utasításra
  $C(l, P^L, l') = (pc = l \wedge pc_1' = l_1 \wedge \cdots \wedge pc_n' = l_n \wedge pc' = \perp)$
  $\vee \left((pc = \perp \wedge pc_1 = l_1' \wedge \cdots \wedge pc_n = l_n' \wedge pc' = l') \wedge \left(\bigwedge_{i=1}^n (pc_i' = \perp)\right)\right)$
  $\vee \left(\bigvee_{i=1}^n \left(C(l_i, P_i^L, l_i') \wedge same(V - V_i) \wedge same(PC - \{pc_i\})\right)\right)$

Példák Kripke struktúrákra

Osztott változók kezelése: a V_i halmazok metszete nem üres

Folyamat szinkronizáló utasítások szükségesek.

$\textbf{wait}(b)$ utasítás:

$$C(l, \textbf{wait}(b), l') = (pc = l \wedge pc' = l \wedge \neg b \wedge same(V)) \vee (pc = l \wedge pc' = l' \wedge b \wedge same(V))$$

$\textbf{lock}(v)$ utasítás:

$$C(l, \textbf{lock}(v), l') = (pc = l \wedge pc' = l \wedge v = 1 \wedge same(V)) \vee (pc = l \wedge pc' = l' \wedge v = 0 \wedge v' = 1 \wedge same(V))$$

$\textbf{unlock}(v)$ utasítás:

$$C(l, \textbf{unlock}(v), l') = (pc = l \wedge pc' = l' \wedge v' = 0 \wedge same(V) \setminus \{v\})$$

Példák Kripke struktúrákra

4) Kölcsönös kizárás (mutual exclusion)

P = m : **cobegin** $P_0 \parallel P_1$**coend** m'

P_0 : l_0 : **while** *True* **do**
NC_0 : **wait** $(turn = 0)$;
$CR_0 : turn = 1$;
endwhile
l_0'

P_1 : l_1 : **while** *True* **do**
NC_1 : **wait** $(turn = 1)$;
$CR_1 : turn = 0$;
endwhile
l_1'

NC : nem kritikus szekció
CR : kritikus szekció
P_0 és P_1 egyszerre nem lehetnek a kritikus szekcióban !

Példák Kripke struktúrákra

Atomi változók

pc (a P program számlálója) : $m, m', \perp$
pc_i (a P_i program számlálója, $i = 0, 1$) : $l_i, l_i', NC_i, CR_i, \perp$
$PC = \{pc, pc_0, pc_1\}$
$V = V_0 = V_1 = \{turn\}$

P_0 és P_1 egyszerre nem lehetnek a kritikus szekcióban :

$$\neg(pc_0 = CR_0 \wedge pc_1 = CR_1)$$

Példák Kripke struktúrákra

- $\varphi_0(V, PC) = (pc = m) \wedge (pc_0 = \perp) \wedge (pc_1 = \perp)$
- $\varrho(V, PC, V', PC')$ a következő négy formula diszjunkciója

1. $(pc = m) \wedge (pc_0' = l_0) \wedge (pc_1' = l_1) \wedge (pc' = \perp)$
2. $(pc_0 = l_0') \wedge (pc_1 = l_1') \wedge (pc' = m') \wedge (pc_0' = \perp) \wedge (pc_1' = \perp)$
3. $C(l_0, P_0, l_0') \wedge same(V - V_0) \wedge same(PC - \{pc_0\}) \equiv$
$C(l_0, P_0, l_0') \wedge same(pc, pc_1)$
4. $C(l_1, P_1, l_1') \wedge same(pc, pc_0)$

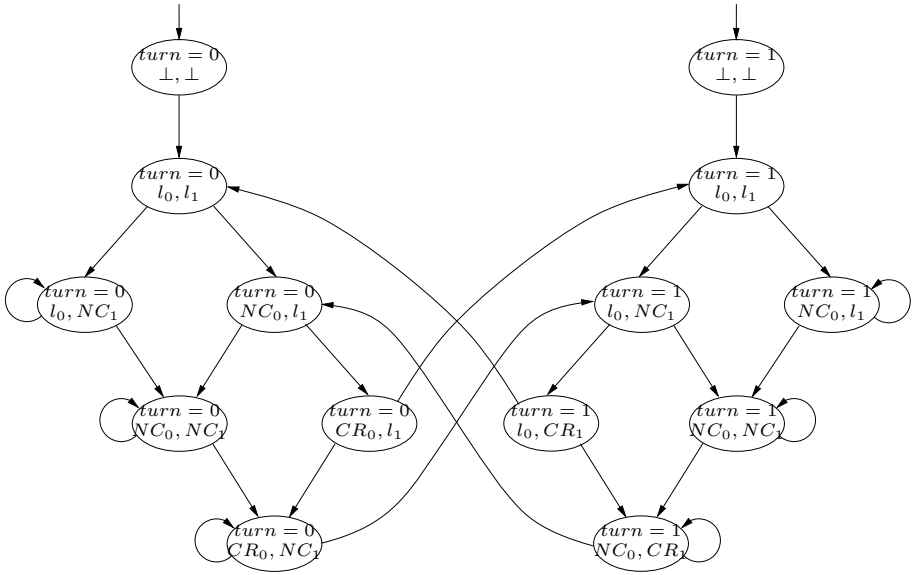
ahol $i = 0, 1$-re $C(l_i, P_i, l_i')$ a következő 5 formula diszjunkciója

Példák Kripke struktúrákra

$i = 0, 1$-re $C(l_i, P_i, l_i')$ a következő 5 formula diszjunkciója :

1. $(pc_i = l_i) \wedge (pc_i' = NC_i) \wedge$ *True* $\wedge same(turn)$
2. $(pc_i = NC_i) \wedge (pc_i' = CR_i) \wedge turn = i \wedge same(turn)$
3. $(pc_i = CR_i) \wedge (pc_i' = l_i) \wedge turn' = (i + 1) mod 2$
4. $(pc_i = NC_i) \wedge (pc_i' = NC_i) \wedge turn \neq i \wedge same(turn)$
5. $(pc_i = l_i) \wedge (pc_i' = l_i') \wedge$ *False* $\wedge same(turn)$

Példák Kripke struktúrákra



Példák Kripke struktúrákra

A kölcsönös kizárás fenti Kripke strukúrájából látható, hogy a két folyamat nem lehet ugyanaban az időben kritikus szekcióban (mivel nincs olyan állapot, amiben $pc_0 = CR_0$ és $pc_1 = CR_1$ egyszerre teljesül).

Ugyanakkor a program nem zárja ki, hogy az egyik folyamat soha nem hajthatja végre a kritikus szekcióját, míg a másik végtelen sokszor. (Éhen halás problémája.)

Állapotrobbanás

A modell alkotás módszereiből látható, hogy a modell mérete általában exponenciális függvénye a modellezéshez szükséges változók számának (pl. n Boole változó esetén a Kripke struktúra mérete 2^n lehet).

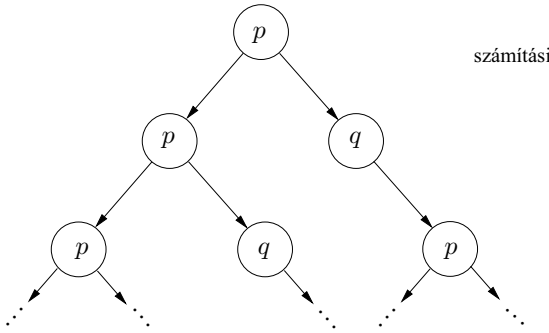
Ezért nagyon fontos a Kripke struktúra hatékony (tártakarékos) ábrázolása, pl. rendezett bináris döntési diagramokkal (OBDD), ld. később.

Temporális logika

Számítási fák tulajdonságait írjuk le.

Számítási fa: egy végtelen fa, amelyet a Kripke struktúra egy állapotából történő széthajtogatással kapunk.

Kripke struktúra

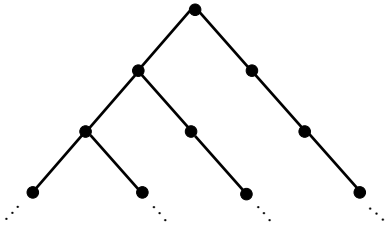


számítási fa

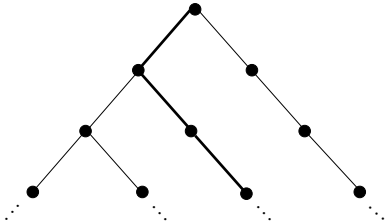
CTL* = Computation Tree Logic (Számítási fa logika)

Útkvantorok: a számítási fa egy pontjából kiinduló utakra vonatkoznak

$\mathbf{A}=$ **A**ll, minden út



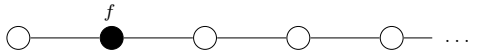
$\mathbf{E}=$ **E**xists, van olyan út



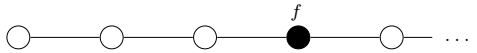
CTL* = Számítási fa logika

Időoperátorok: egy konkrét útra vonatkoznak

$\mathbf{X}=$ ne**X**t time: az út következő pontján teljesül valami



$\mathbf{F}=$ **F**uture, eventually: az út valamely pontján teljesül valami

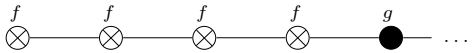


$\mathbf{G}=$ always, **G**lobally: az út minden pontján

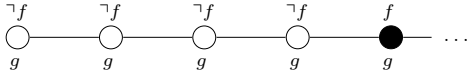
CTL* = Számítási fa logika

Időoperátorok: egy konkrét útra vonatkoznak

$\mathbf{U}=$ **U**ntil: van egy olyan pont, amelyen teljesül valami és eddig a pontig teljesül egy másik valami



$\mathbf{R}=$ **R**elease: ha valami nem teljesül egy pontig, akkor a következő pontban teljesül egy másik valami



CTL* = Számítási fa logika

Szintaxis

Legyen AP az atomi változók halmaza

Állapotformulák

- ha $p \in AP$, akkor p állapotformula
- ha f, g állapotformulák, akkor $\neg f,\ f \vee g,\ f \wedge g$ is állapotformulák
- ha f útformula, akkor $\mathbf{E}f$ és $\mathbf{A}f$ állapotformula

Útformulák

- ha f állapotformula, akkor f útformula is
- ha f, g útformulák, akkor $\neg f,\ f \vee g,\ f \wedge g,\ \mathbf{X}f,\ \mathbf{F}f,\ \mathbf{G}f,\ f\,\mathbf{U}\,g,\ f\,\mathbf{R}\,g$ is útformulák

CTL* = Számítási fa logika

Szemantika

Legyen $M = (S, R, L)$ egy Kripke struktúra. Egy $\pi = s_0, s_1, \ldots$ út és $i \geq 0$ esetén $\pi^i = s_i, s_{i+1}, \ldots$.

Legyen f egy állapotformula és $s \in S$ egy állapot. f szerinti indukcióval definiáljuk az "M az s állapotban kielégíti f-et" fogalmát: $M, s \models f$

- $M, s \models p \iff p \in L(s)$
- $M, s \models \neg f_1 \iff$ nem teljesül, hogy $M, s \models f_1$
- $M, s \models f_1 \vee f_2 \iff M, s \models f_1$ vagy $M, s \models f_2$
- $M, s \models f_1 \wedge f_2 \iff M, s \models f_1$ és $M, s \models f_2$
- $M, s \models \mathbf{E} f_1 \iff$ van olyan s-ből kiinduló π út, amelyre $M, \pi \models f_1$
- $M, s \models \mathbf{A} f_1 \iff$ az s-ből kiinduló minden π útra $M, \pi \models f_1$

Ha nem okoz félreértést, akkor $M, s \models f$ és $M, \pi \models f$ helyett csak $s \models f$-et és $\pi \models f$-et írunk.

CTL* = Számítási fa logika

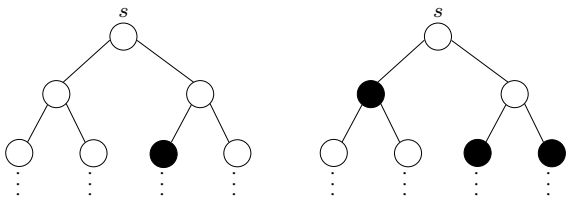
Szemantika

Legyen π egy állapotból kiinduló út és f egy útformula. f szerinti indukcióval definiáljuk az "M π útja kielégíti f-et" fogalmát: $M, \pi \models f$

- Ha f állapotformula is, akkor
  $M, \pi \models f \iff \pi = s_0, s_1, \ldots$ és $M, s_0 \models f$
- $M, \pi \models \neg f_1$, $M, \pi \models f_1 \vee f_2$, $M, \pi \models f_1 \wedge f_2$ a szokásos módon
- $M, \pi \models \mathbf{X} f_1 \iff M, \pi^1 \models f_1$
- $M, \pi \models \mathbf{F} f_1 \iff$ van olyan $i \geq 0$, hogy $M, \pi^i \models f_1$
- $M, \pi \models \mathbf{G} f_1 \iff$ minden $i \geq 0$-ra $M, \pi^i \models f_1$
- $M, \pi \models f_1 \, \mathbf{U} \, f_2 \iff$
  van olyan $k \geq 0$, hogy $M, \pi^k \models f_2$ és minden $0 \leq i < k$-ra $M, \pi^i \models f_1$
- $M, \pi \models f_1 \, \mathbf{R} \, f_2 \iff$
  minden $j \geq 0$-ra, ha tetszőleges $0 \leq i < j$-re $M, \pi^i \not\models f_1$, akkor $M, \pi^j \models f_2$

CTL* Számítási fa logika

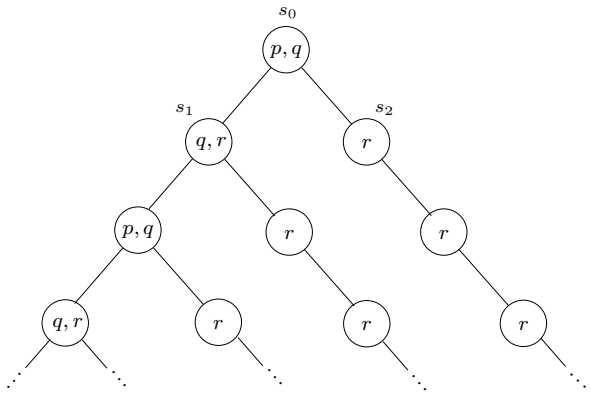
Tipikus példák



$M, s \models \mathbf{EF} g$ $\quad$ $M, s \models \mathbf{AF} g$

$M, s \models \mathbf{EG} g$ $\quad$ $M, s \models \mathbf{AG} g$

CTL* Számítási fa logika



1. $M, s_0 \models p \wedge q$
2. $M, s_0 \models \neg r$
3. $M, s_0 \models \mathbf{EX}(q \wedge r)$
4. $M, s_0 \models \neg \mathbf{AX}(q \wedge r)$
5. $M, s_0 \models \neg \mathbf{EF}(p \wedge r)$
6. $M, s_2 \models \mathbf{EG} r$
7. $M, s_2 \models \mathbf{AG} r$
8. $M, s_0 \models \mathbf{AF} r$
9. $M, s_0 \models \mathbf{E}((p \wedge q) \mathbf{U} r)$
10. $M, s_0 \models \mathbf{A}(p \mathbf{U} r)$

CTL* Számítási fa logika

Ekvivalencia

Tetszőleges f és g útformulák esetén $f \equiv g \iff \forall M = (S, R, L)$ Kripke struktúra és $\forall (s \in S)$ esetén $M, s \models f \iff M, s \models g$

(Útformulákra hasonlóan.)

Ekvivalens formulák

$f \wedge g \equiv \neg(\neg f \vee \neg g)$

$f\mathbf{R}g \equiv \neg(\neg f \mathbf{U} \neg g)$

$\mathbf{F}f \equiv True\mathbf{U}f$, ahol $True$ egy azonosan igaz formula

$\mathbf{G}f \equiv \neg\mathbf{F}(\neg f)$

$\mathbf{A}f \equiv \neg\mathbf{E}(\neg f)$

Következmény: A $\vee, \neg, \mathbf{X}, \mathbf{U}$ és $\mathbf{E}$ operátorok adekvát (teljes) rendszert alkotnak, azaz bármely CTL* formula átalakítható olyan, az eredetivel ekvivalens formulává, amely csak a fenti operátorokat tartalmazza.

A CTL($\subseteq$ CTL*) logika = branching time logic

Csak olyan CTL* formulákat engedünk meg, amelyekben az $\mathbf{X}, \mathbf{F}, \mathbf{G}, \mathbf{U}$ és $\mathbf{R}$ időoperátorokat közvetlenül megelőzi valamely útkvantor.

Állapotformulák

- ha $p \in AP$, akkor p állapotformula
- ha f, g állapotformulák, akkor $\neg f, f \vee g, f \wedge g$ is állapotformulák
- ha f útformula, akkor $\mathbf{E}f$ és $\mathbf{A}f$ állapotformula

Útformulák

- ha f, g állapotformulák, akkor $\mathbf{X}f, \mathbf{F}f, \mathbf{G}f, f\mathbf{U}g$ és $f\mathbf{R}g$ útformulák

Az LTL($\subseteq$ CTL*) logika = linear time logic

Csak $\mathbf{A}f$ alakú állapotformulákat engedünk meg.

Állapotformulák

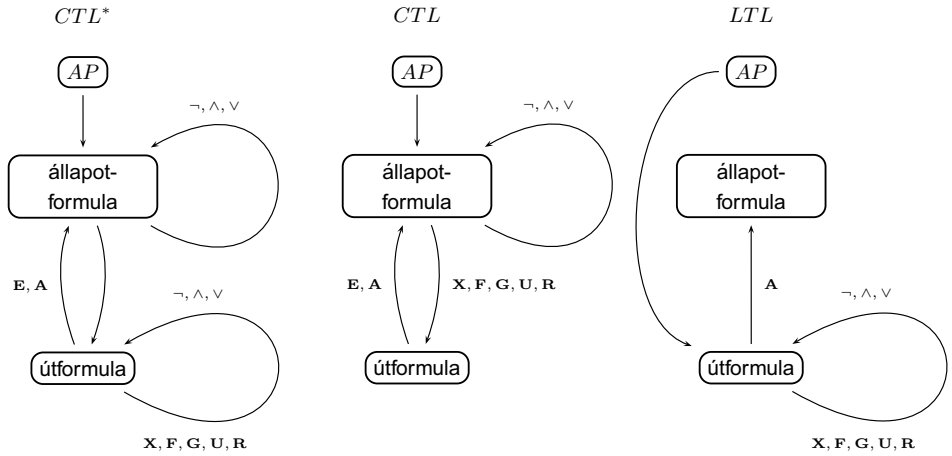
- ha f útformula, akkor $\mathbf{A}f$ állapotformula

Útformulák

- ha $p \in AP$, akkor p útformula
- ha f, g útformulák, akkor $\neg f, f \vee g, f \wedge g, \mathbf{X}f, \mathbf{F}f, \mathbf{G}f, f\mathbf{U}g$ és $f\mathbf{R}g$ útformulák

Megjegyzés. Az LTL logikának a következő definíciója is szokásos: (1) nem engedjük meg az $\mathbf{A}$ operátort (tehát csak útformulák vannak), ugyanakkor (2) az mondjuk, hogy $M, s \models f$, ha minden s-ből kiinduló π útra teljesül, hogy $M, \pi \models f$. Nyilvánvalóan a két definíció kifejező ereje azonos.

A CTL*, CTL és LTL logikák szintaxisának összehasonlítása



A CTL*, CTL és LTL logikák szintaxisának összehasonlítása

Az ábrából látható, hogy (egy adott AP esetén) mind a CTL, mind az LTL (állapot- és út)formulák halmaza valódi része a CTL* (állapot- és út)formulák halmazának. Továbbá, a CTL és az LTL formulák halmaza összehasonlíthatatlan, ugyanakkor a metszetük nem üres.

Ugyanazen tartalmazás áll fenn szemantikailag is. Tehát pl. megadható olyan CTL* formula (pl. az $\mathbf{E}(\mathbf{GF}p)$), amely nem ekvivalens egyetlen CTL és egyetlen LTL formulával sem, stb. A szemantikai tartalmazásokat és összehasonlíthatatlanságot igazoló formulák a következő ábrán láthatók. A megfelelő szemantikai állítások igazolása nem mindig egyszerű.

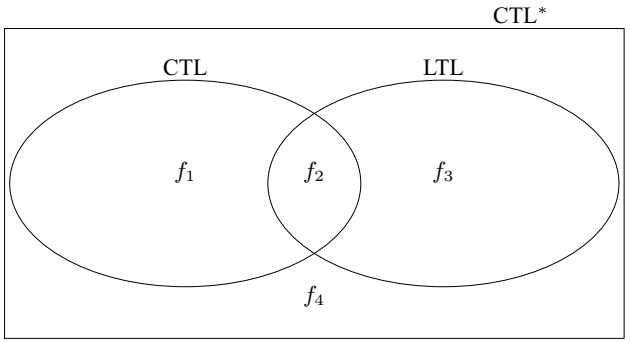
A CTL*, CTL és LTL logikák kifejező erejének összehasonlítása

$f_1 = \mathbf{AGEF}p$

$f_2 = \mathbf{AG}(p \rightarrow \mathbf{AF}q)$ CTL-ben és $f_2 = \mathbf{AG}(p \rightarrow \mathbf{F}q)$ LTL-ben

$f_3 = \mathbf{A}(\mathbf{GF}p \rightarrow \mathbf{F}q)$

$f_4 = \mathbf{E}(\mathbf{GF}p)$



Azonosságok CTL-ben

A CTL-ben 10 alapvető operátor van:

$\mathbf{AX}$, $\mathbf{EX}$, $\mathbf{AF}$, $\mathbf{EF}$, $\mathbf{AG}$, $\mathbf{EG}$, $\mathbf{AU}$, $\mathbf{EU}$, $\mathbf{AR}$, $\mathbf{ER}$

Tétel. Az $\mathbf{EX}$, $\mathbf{EG}$, $\mathbf{EU}$ operátorhalmaz teljes.

Bizonyítás.

$$\begin{aligned}
\mathbf{AX}f &\equiv \neg\mathbf{EX}(\neg f)\\
\mathbf{EF}f &\equiv \mathbf{E}(\mathit{True}\ \mathbf{U}\ f)\\
\mathbf{AG}f &\equiv \neg\mathbf{EF}(\neg f)\\
\mathbf{AF}f &\equiv \neg\mathbf{EG}(\neg f)
\end{aligned}$$

$$\begin{aligned}
\mathbf{A}(f\,\mathbf{U}\,g) &\equiv \neg\mathbf{E}(\neg g\,\mathbf{U}\,(\neg f \wedge \neg g)) \wedge \neg\mathbf{EG}(\neg g)\\
&\equiv \neg(\mathbf{E}(\neg g\,\mathbf{U}\,(\neg f \wedge \neg g)) \vee \mathbf{EG}(\neg g))\\
\mathbf{A}(f\,\mathbf{R}\,g) &\equiv \neg\mathbf{E}(\neg f\,\mathbf{U}\,\neg g)\\
\mathbf{E}(f\,\mathbf{R}\,g) &\equiv \neg\mathbf{A}(\neg f\,\mathbf{U}\,\neg g)
\end{aligned}$$

Azonosságok CTL-ben

Tétel. Az $\mathbf{EX}$, $\mathbf{AU}$, $\mathbf{EU}$ operátorhalmaz teljes.

Bizonyítás. Az $\mathbf{EX}$, $\mathbf{EG}$, $\mathbf{EU}$ teljes, továbbá

$$\begin{aligned}
\mathbf{EG}f &\equiv \neg\mathbf{AF}(\neg f)\\
&\equiv \neg\mathbf{A}(\mathit{True}\ \mathbf{U}\,\neg f)
\end{aligned}$$

További teljes operátorhalmazok:

- $\mathbf{AG}$, $\mathbf{AU}$, $\mathbf{AX}$
- $\mathbf{AF}$, $\mathbf{EU}$, $\mathbf{EX}$

Korrekt (fair) számítási utak

Vannak olyan esetek, amikor nem minden számítási út érdekel bennünket, hanem csak az ún. *korrekt útak*. Ez CTL-ben nem fejezhető ki (csak CTL*-ban), ezért módosítjuk a CTL szemantikáját.

Korrekt Kripke struktúra: $M = (S, R, L, F)$, ahol $F \subseteq \mathcal{P}(S)$, a *korrektségi korlátozások* halmaza.

Egy $\pi = s_0, s_1, \ldots$ út esetén legyen

$$inf(\pi) = \{s \mid \text{végtelen sok } i\text{-re } s = s_i\}$$

Egy út *korrekt*, ha minden $P \in F$-re

$$P \cap inf(\pi) \neq \emptyset$$

Az $M, s \models f$ helyett $M, s \models_F f$-et írunk.

Korrekt számítási utak

A $\models_F$ definíciója ugyanaz, mint a $\models$ definíciója, kivéve az

$$M, s \models p$$

$$M, s \models \mathbf{E}(f_1)$$

$$M, s \models \mathbf{A}(f_1)$$

eseteket, amelyek a következőkre változnak:

$M, s \models_F p \iff p \in L(s)$ és létezik s-ből kiinduló korrekt út

$M, s \models_F \mathbf{E}(f_1) \iff$ létezik olyan, s-ből kiinduló π korrekt út, amelyre $M, \pi \models_F f_1$

$M, s \models_F \mathbf{A}(f_1) \iff$ minden, az s-ből kiinduló π korrekt útra $M, \pi \models_F f_1$.

A modell ellenőzés alapfeladata

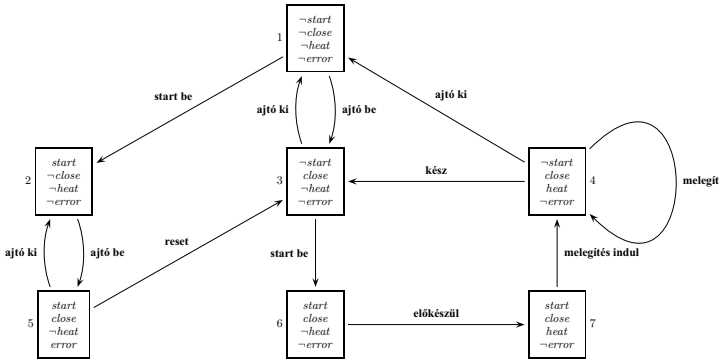
1) Adott: $M = (S, R, L)$ Kripke struktúra, $s_0 \in S$ állapot és f CTL (LTL) formula. Kérdés: teljesül-e $M, s_0 \models f$?
2) Adott: $M = (S, R, L)$ Kripke struktúra és f CTL (LTL) formula. Feladat: határozzuk meg az
   $$\{s \in S \mid M, s \models f\}$$
   halmazt!

$1) \Rightarrow 2)$ (mivel véges számú állapot van)

$2) \Rightarrow 1)$ $s_0 \in \{s \in S \mid M, s \models f\}$?

Példa modell ellenőrzésre

A mikrohullámú sütő modellje



Kérdés: mely állapotokban teljesül az

$$\mathbf{AG}(\text{start} \rightarrow \mathbf{AF}\ \text{heat})$$

formula?

CTL modell ellenőrzés címkézéssel

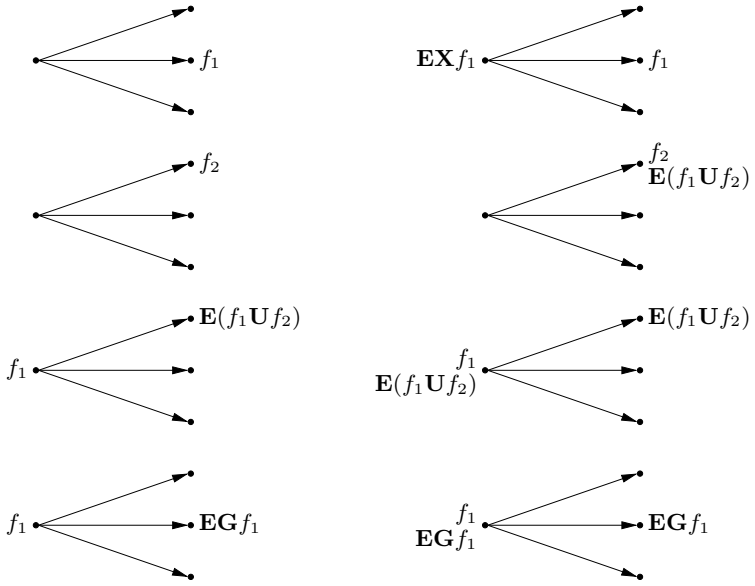
A modell ellenőrzés alapfeladatát a következőképpen oldjuk meg:

Adott $M = (S, R, L)$ Kripke struktúra és f formula esetén minden $s \in S$-re meghatározzuk a $lab(s)$ halmazt, amely f azon f' részformuláiból áll, melyekre $M, s \models f'$! Így

$$M, s \models f \iff f \in lab(s)$$

CTL modell ellenőrzés címkézéssel

Példák: $\mathbf{EX}$, $\mathbf{EU}$, $\mathbf{EG}$



CTL modell ellenőrzés címkézéssel

A címkézési eljárást pszeudokód formában adjuk meg, arra alapozva, hogy $\mathbf{EX}, \mathbf{EU}$ és $\mathbf{EG}$ teljes operátorhalmazt alkotnak a CTL-ben.

A következő pszeudokódok egy $M = (S, R, L)$ Kripke struktúrára és egy f formulára vonatkoznak.

<u>Előkészítő rész:</u>

forall $s \in S$ **do** $lab(s) := \emptyset$;

CTL modell ellenőrzés címkézéssel

procedure $MC(f)$;
begin
case
$f \in AP$: **forall** $s \in S$ **do**
if $f \in L(s)$ **then** $lab(s) = lab(s) \cup \{f\}$ **endforall** ;
$f = \neg f_1 : MC(f_1)$; **forall** $s \in S$ **do**
if $f_1 \notin lab(s)$ **then** $lab(s) = lab(s) \cup \{f\}$ **endforall** ;
$f = f_1 \vee f_2 : MC(f_1); MC(f_2)$; **forall** $s \in S$ **do**
if $(f_1 \in lab(s)$ **or** $f_2 \in lab(s))$ **then**
$lab(s) = lab(s) \cup \{f\}$ **endforall** ;
$f = \mathbf{EX}f_1 : MC(f_1)$;
$Check\mathbf{EX}(f_1)$;
$f = \mathbf{E}(f_1\mathbf{U}f_2) : MC(f_1); MC(f_2)$;
$Check\mathbf{EU}(f_1, f_2)$;
$f = \mathbf{EG}(f_1) : MC(f_1)$;
$Check\mathbf{EG}(f_1)$;
endcase
end

CTL modell ellenőrzés címkézéssel

procedure $Check\mathbf{EX}(f)$;
var s, t, U;
begin
  $U = \{t \in S \mid f \in lab(t)\}$
  forall $s \in S$ **do**
  if $(\exists t \in U) : s\,R\,t$ **then**
    $lab(s) = lab(s) \cup \{\mathbf{EX}f\}$
  endforall
end

CTL modell ellenőrzés címkézéssel

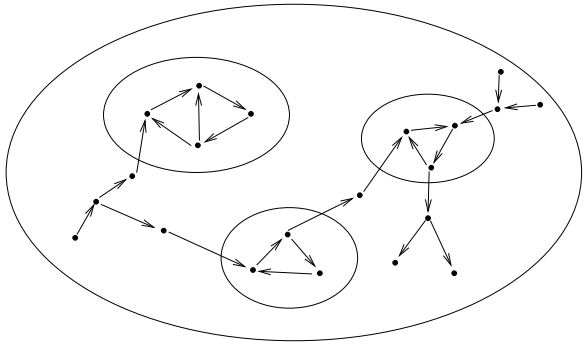
procedure $Check\mathbf{EU}(f_1, f_2)$;
var s, t, U, V;
begin
  $U = \{s \mid f_2 \in lab(s)\}$; $V = \{s \mid f_1 \in lab(s)\}$;
  forall $s \in U$ **do**
    $lab(s) = lab(s) \cup \{\mathbf{E}(f_1\mathbf{U}f_2)\}$
  endforall
  while $U \neq \emptyset$ **do** legyen $s \in U$
    forall $t \in V$ **do**
      if tRs **and** $\mathbf{E}(f_1\mathbf{U}f_2) \notin lab(t)$ **then**
        begin $lab(t) = lab(t) \cup \{\mathbf{E}(f_1\mathbf{U}f_2)\}$
          $U = U \cup \{t\}$
        end
    endforall
    $U = U - \{s\}$
  endwhile
end

CTL modell ellenőrzés címkézéssel

A Check$\mathbf{EG}$ eljárás egy irányított gráf erősen összefüggő komponenseinek (SCC) meghatározásán alapul.

Egy irányított gráf valamely C részgráfja erősen összefüggő komponens, ha olyan maximális részgráf, melyben bármely $a, b \in C$ csúcsokra a elérhető b-ből és b elérhető a-ból.

C nemtriviális, ha egynél több csúcsból áll, vagy ha egy olyan csúcsból áll, amelyből hurokél vezet önmagába.



CTL modell ellenőrzés címkézéssel

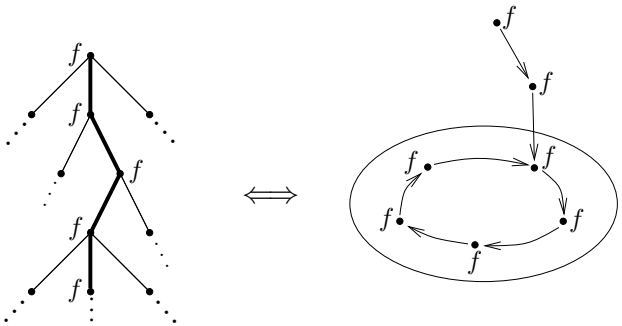
Legyen $M = (S, R, L)$ egy Kripke struktúra és f egy formula. Jelöljük $M' = (S', R', L')$-vel M azon részstruktúráját, amelyre

- $S' = \{s \in S \mid M, s \models f\}$
- $R' = R \cap (S' \times S')$
- L' az L S'-re való megszorítása

Lemma. Tetszőleges $s \in S$-re $M, s \models \mathbf{EG}f$ akkor és csak akkor áll fenn, ha a következő két feltétel teljesül:

1. $s \in S'$
2. az (S', R') irányított gráfnak van olyan C nemtriviális SCC-je, hogy s-ből út vezet valamely $t \in C$-be.

CTL modell ellenőrzés címkézéssel



$s \models \mathbf{EG} f \iff s \in S'$ és az (S', R') irányított gráfnak van olyan C nemtriviális SCC-je, hogy s-ből út vezet valamely $t \in C$-be.

Az előbbi lemma alapján $Check\mathbf{EG}$ a következő

procedure $Check\mathbf{EG}(f_1)$;
var S', T, SCC, s, t;
begin
  $S' = \{s \in S \mid f_1 \in lab(s)\}$;
  $SCC = \{C \mid C$ nemtriviális SCC S'-ben$\}$;
  $T = \bigcup_{C \in SCC} C$;
  forall $s \in T$ **do** $lab(s) := lab(s) \cup \{\mathbf{EG}(f_1)\}$ **endforall**
  while $T \neq \emptyset$ **do** legyen $s \in T$;
    forall $t \in S'$ **do**
      if $t\,R\,s$ **and** $\mathbf{EG}(f_1) \notin lab(t)$ **then**
        begin $lab(t) = lab(t) \cup \{\mathbf{EG}(f_1)\}$;
          $T = T \cup \{t\}$;
        end
    endforall
    $T = T - \{s\}$;
  endwhile
end

CTL modell ellenőrzést címkézéssel

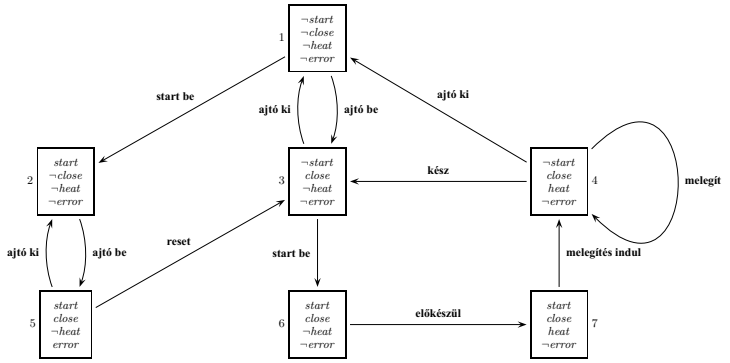
Időbonyolultsági kérdések

A $Check\mathbf{EU}$ és a $Check\mathbf{EG}$ időbonyolultsága $O(\mid S \mid + \mid R \mid)$. Jelölje $\mid f \mid$ egy formula részformuláinak számát.

Tétel. Létezik olyan algoritmus, amely tetszőleges $M = (S, R, L)$ Kripke struktúrára, $s \in S$ és f CTL formula esetén eldönti, hogy $M, s \models f$ teljesül-e. Az algoritmus időbonyolultsága $O(\mid f \mid (\mid S \mid + \mid R \mid))$.

Példa modell ellenőrzésre

A mikrohullámú sütő modellje



Kérdés: mely állapotokban teljesül az

$$\mathbf{AG}(\text{start} \rightarrow \mathbf{AF}\ \text{heat})$$

formula?

Példa címkézessel történő CTL modell ellenőrzésre

A mikrohullámú sütő mely állapotában teljesül

$$\mathbf{AG}(\text{start} \rightarrow \mathbf{AF}\ \text{heat})\ ?$$

Jelölés: tetszőleges f formula esetén: $S(f) = \{s \in S \mid f \in lab(s)\}$.

Először átalakítjuk a formulát:

$$\begin{aligned}
&\mathbf{AG}(\text{start} \rightarrow \mathbf{AF}\ \text{heat}) \\
\equiv\ &\mathbf{AG}(\neg\text{start} \vee \mathbf{AF}\ \text{heat}) \\
\equiv\ &\neg\mathbf{EF}\neg(\neg\text{start} \vee \mathbf{AF}\ \text{heat}) \\
\equiv\ &\neg\mathbf{EF}(\text{start} \wedge \neg(\mathbf{AF}\ \text{heat})) \\
\equiv\ &\neg\mathbf{EF}(\text{start} \wedge \mathbf{EG}\neg\ \text{heat})
\end{aligned}$$

1) $S(\text{start}) = \{2, 5, 6, 7\}$
2) $S(\neg\text{heat}) = \{1, 2, 3, 5, 6\}$
3) $S(\mathbf{EG}\neg\text{heat}) : S' = S(\neg\text{heat}) = \{1, 2, 3, 5, 6\}$, $SCC = \{1, 2, 3, 5\}$

Mivel 6-ból nem vezet út SCC-be az (S', R') gráfban, ezért
$S(\mathbf{EG}\neg\text{heat}) = \{1, 2, 3, 5\}$

Példa címkézéssel történő CTL modell ellenőrzésre

A mikrohullámú sütő mely állapotaira teljesül $\mathbf{AG}(start \rightarrow \mathbf{AF}heat)$?

4) $S(start \wedge \mathbf{EG}\neg heat) = \{2, 5\}$
5) Tetszőleges f-re $\mathbf{EF}f \equiv \mathbf{E}(\mathit{True}\ \mathbf{U} f)$, ezért $\mathbf{EF}f = S(f) \cup \{$az összes olyan állapot, amelyből út vezet $S(f)$ valamely elemébe$\}$. $S(\mathbf{EF}(start \wedge \mathbf{EG}\neg heat)) = \{1, 2, 3, 4, 5, 6, 7\}$
6) $S(\neg\mathbf{EF}(start \wedge \mathbf{EG}\neg heat)) = \emptyset$

Tehát a mikrohullámú sütő egyetlen állapotában (és így a kezdőállapotban) sem teljesül, hogy $\mathbf{AG}(start \rightarrow \mathbf{AF}heat)$.

Korrekt CTL modell ellenőrzés címkézéssel

Definíció. Legyen $M = (S, R, L, F)$ egy korrekt Kripke strukúra, C pedig S-nek egy erősen összefüggő komponense. Azt mondjuk, hogy C *korrekt F-re vonatkozóan,* ha minden $P \in F$-re $C \cap P \neq \emptyset$.

1) Megadjuk a $Check Fair\mathbf{EG}(f)$ eljárást.
   Legyen $M' = (S', R', L', F')$, ahol
   - $S' = \{s \in S \mid M, s \models f\}$
   - $R' = R \cap (S' \times S')$
   - L' az L megszorítása S'-re
   - $F' = \{P \cap S' \mid P \in F\}$

Lemma. Tetszőleges $s \in S$-re, $M, s \models_F \mathbf{EG}f$ pontosan akkor, ha a következő két feltétel teljesül:

- $s \in S'$
- az (S', R') irányított gráfnak van olyan C nemtriviális korrekt SCC-je (F'-re vonatkozóan), hogy s-ből út vezet valamely $t \in C$-be.

Korrekt CTL modell ellenőrzés címkézéssel

Ennek ismeretében, a $Check Fair\mathbf{EG}(f)$ eljárás annyiban különbözik a $Check\mathbf{EG}(f)$ eljárástól, hogy az SCC definícióját kicseréljük az $SCC = \{C \mid C$ nemtriviális korrekt SCC S'-ben$\}$ definícióra.

2) A $Check Fair\mathbf{EX}$ és $Check Fair\mathbf{EU}$ eljárások megadása végett vezessük be a $\mathbf{fair}$ atomi állítást: tetszőleges $s \in S$-re
   $M, s \models \mathbf{fair} \iff$ van s-ből induló korrekt út.

Következésképpen $\mathbf{fair} = \mathbf{EG}(\mathit{True})$, ahol $\mathbf{EG}(\mathit{True})$ korrekt ($\models_F$) szemantikáját kell venni. Ezért az S-beli állapotoknak a $\mathbf{fair}$ állítással való megcímkézése elvégezhető a $Check Fair\mathbf{EG}(\mathit{True})$ eljáráshívással.

Korrekt CTL modell ellenőrzés címkézéssel

Ezek után $CheckFair\mathbf{EX}(f)$ a következő lesz:

$CheckFair\mathbf{EG}(\mathit{True})$

$Check\mathbf{EX}(f \wedge \mathbf{EG}(\mathit{True}))$

$CheckFair\mathbf{EU}(f_1, f_2)$ a következő lesz:

$CheckFair\mathbf{EG}(\mathit{True})$

$Check\mathbf{EU}(f_1, f_2 \wedge \mathbf{EG}(\mathit{True}))$

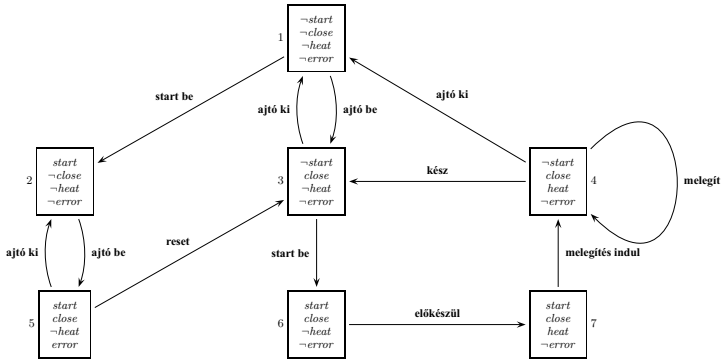
Az $MCFair$ eljárás a következőképpen adható meg:

Korrekt CTL modell ellenőrzés címkézessel

procedure $MCFair(f)$
begin $CheckFair\mathbf{EG}(\mathit{True})$;
  case
    $f \in AP$: **forall** $s \in S$ **do**
    if $f \in L(s)$ **and** $\mathbf{EG}(\mathit{True}) \in lab(s)$ **then**
      $lab(s) = lab(s) \cup \{f\}$ **endforall** ;
    $f = \neg f_1 : MCFair(f_1)$;
    forall $s \in S$ **do**
      if $f_1 \notin lab(s)$ **then** $lab(s) = lab(s) \cup \{f\}$;
    endforall ;
    $f = f_1 \vee f_2$: hasonlóan
    $f = \mathbf{EX} f_1 : MCFair(f_1); Check\mathbf{EX}(f_1 \wedge \mathbf{EG}\mathit{True})$;
    $f = \mathbf{E}(f_1 \mathbf{U} f_2) : MCFair(f_1); MCFair(f_2)$;
      $Check\mathbf{EU}(f_1, f_2 \wedge \mathbf{EG}(\mathit{True}))$;
    $f = \mathbf{EG}(f_1)$:
      $MCFair(f_1); CheckFair\mathbf{EG}(f_1)$;
  endcase
end

Példa modell ellenőrzésre

A mikrohullámú sütő modellje



Példa címkézéssel történő CTL modellvizsgálatra

A mikrohullámú sütő példában vezessük be az $F = \{P\}$ korrektségi korlátozást, ahol $P = \{s \mid M, s \models start \wedge close \wedge \neg error\} = \{6, 7\}$.

Mely állapotban teljesül a korrekt szemantika mellett $\mathbf{AG}(start \rightarrow \mathbf{AF} heat)$?

1) $S(start) = \{2, 5, 6, 7\}$
2) $S(\neg heat) = \{1, 2, 3, 5, 6\}$
3) $S(\mathbf{EG}\neg heat)$ korrekt számítása: $S' = \{1, 2, 3, 5, 6\}$, $SCC = \emptyset$, mert nincs olyan SCC, mely tartalmazza 6-ot vagy 7-et. Ezért $S(\mathbf{EG}\neg heat) = \emptyset$
4) $S(start \wedge \mathbf{EG}\neg heat) = \emptyset$
5) $S(\mathbf{EF}(start \wedge \mathbf{EG}\neg heat)) = \emptyset$
6) $S(\neg\mathbf{EF}(start \wedge \mathbf{EG}\neg heat)) = S(\mathbf{AG}(start \rightarrow \mathbf{AF} heat)) = \{1, 2, 3, 4, 5, 6, 7\}$

Korrekt modell ellenőrzés címkézéssel

Tétel. Létezik olyan algoritmus, mely tetszőleges $M = (S, R, L, F)$ korrekt Kripke struktúra, $s \in S$ és f CTL formula esetén eldönti, hogy $M, s \models_F f$ teljesül-e. Az algoritmus időbonyolultsága $O(|f|(|S| + |R|)|F|)$.

LTL modell ellenőrzés tablóval

LTL szintaxis

Állapotformulák

- ha f útformula, akkor $\mathbf{A}f$ állapotformula

Útformulák

- ha $p \in AP$, akkor p útformula
- ha f, g útformulák, akkor $\neg f, f \vee g, f \wedge g, \mathbf{X}f, \mathbf{F}f, \mathbf{G}f, f\mathbf{U}g, f\mathbf{R}g$ útformulák

Adott $M = (S, R, L)$ Kripke struktúra és $s \in S$ esetén igaz-e $M, s \models \mathbf{A}f$?

Mivel $\mathbf{A}f \equiv \neg\mathbf{E}\neg f$, ezért elegendő csak az $M, s \models \mathbf{E}f$ igazságértékét megvizsgálni. **PSPACE** teljes probléma!

LTL modell ellenőrzés tablóval

Az útformulákra vonatkozó

$$\begin{aligned} \mathbf{F}f &\equiv \mathit{True}\ \mathbf{U}f \\ \mathbf{G}f &\equiv \neg\mathbf{F}\neg f \\ f_1\mathbf{R}f_2 &\equiv \neg(\neg f_1\mathbf{U}\neg f_2) \end{aligned}$$

azonosságok miatt elegendő csak az $\mathbf{X}$ és $\mathbf{U}$ időoperátorokkal foglalkozni. Algoritmust adunk $M, s \models \mathbf{E}f$ eldöntésére, ahol $M = (S, R, L)$.

Definíció. Tetszőleges f formula $Cl(f)$ lezárása a legszűkebb olyan halmaz, melyre teljesülnek:

- $f \in Cl(f)$
- $\neg f_1 \in Cl(f) \iff f_1 \in Cl(f)$ (ahol $\neg\neg f_1 = f_1$)
- ha $f_1 \vee f_2 \in Cl(f)$, akkor $f_1, f_2 \in Cl(f)$
- ha $\mathbf{X}f_1 \in Cl(f)$, akkor $f_1 \in Cl(f)$
- ha $\neg\mathbf{X}f_1 \in Cl(f)$, akkor $\mathbf{X}\neg f_1 \in Cl(f)$
- ha $f_1\mathbf{U}f_2 \in Cl(f)$, akkor $f_1, f_2, \mathbf{X}(f_1\mathbf{U}f_2) \in Cl(f)$

LTL modell ellenőrzés tablóval

Definíció. Egy $A = (s_A, K_A)$ alakú párt *atom*nak nevezünk, ahol $s_A \in S, K_A \subseteq Cl(f) \cup AP$ és az alábbi feltételek teljesülnek:

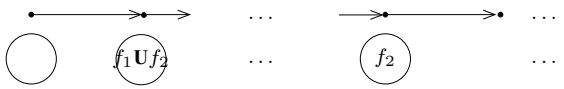
- minden $p \in AP$-re $p \in K_A \iff p \in L(s_A)$
- minden $f_1 \in Cl(f)$-re $f_1 \in K_A \iff \neg f_1 \notin K_A$
- minden $f_1 \vee f_2 \in Cl(f)$-re $f_1 \vee f_2 \in K_A \iff f_1 \in K_A$ vagy $f_2 \in K_A$
- minden $\neg\mathbf{X}f_1 \in Cl(f)$-re $\neg\mathbf{X}f_1 \in K_A \iff \mathbf{X}\neg f_1 \in K_A$
- minden $f_1\mathbf{U}f_2 \in Cl(f)$-re $f_1\mathbf{U}f_2 \in K_A \iff f_2 \in K_A$ vagy ($f_1 \in K_A$ és $\mathbf{X}(f_1\mathbf{U}f_2) \in K_A$)

K_A: olyan maximális, konzisztens formulahalmaz, mely s_A címkézésével is konzisztens.

LTL modell ellenőrzés tablóval

Definíció. Legyen G a következő gráf. Csúcsai az atomok és (A, B) akkor és csak akkor él, ha $s_A \, R \, s_B$ és minden $\mathbf{X}f_1 \in Cl(f)$ formula esetén
$\mathbf{X}f_1 \in K_A \iff f_1 \in K_B$

Definíció. G-ben $\mathbf{U}$-teljesítő útnak nevezünk egy olyan π utat, melyre teljesül, hogy ha $f_1 \mathbf{U} f_2 \in K_A$ valamely π-beli A atomra, akkor π-ben van egy olyan A-ból elérhető B atom, melyre $f_2 \in K_B$.



LTL modell ellenőrzés tablóval

Lemma: $M, s \models \mathbf{E}f$ akkor és csak akkor, ha G-ben van egy olyan (s, K) csúcs, amelyre $f \in K$ és amelyből kiindul egy $\mathbf{U}$ teljesítő út.

Bizonyítás: "$\Longleftarrow$"

Legyen $(s, K) = (s_0, K_0), (s_1, K_1), \ldots$ egy $\mathbf{U}$-teljesítő út, amelyre $f \in K$. Ekkor $\pi = s_0, s_1, \ldots$ $(s = s_0)$ egy út M-ben.

Azt kell megmutatni, hogy $M, \pi \models f$.

Állítás: minden $g \in Cl(f)$-re, $i \geq 0$-ra $\pi^i \models g \iff g \in K_i$.

g szerinti indukció:

- ha $g \in AP$, akkor az atom definíciója szerint
  $g \in K_i \iff g \in L(s_i) \iff \pi^i \models g$.
- ha $g = \neg h_1$, akkor $\pi^i \models g \iff \pi^i \not\models h_1 \iff h_1 \notin K_i \iff g \in K_i$
- ha $g = h_1 \vee h_2$, akkor $\pi^i \models g \iff \pi^i \models h_1$ vagy $\pi^i \models h_2 \iff h_1 \in K_i$ vagy $h_2 \in K_i \iff g \in K_i$

LTL modell ellenőrzés tablóval

- ha $g = \mathbf{X}h_1$, akkor
  $\pi^i \models g \iff \pi^{i+1} \models h_1 \iff h_1 \in K_{i+1} \iff \mathbf{X}h_1 \in K_i$
  (G definíciója miatt) $\iff g \in K_i$
- Tegyük fel, hogy $g = h_1 \mathbf{U} h_2 \in K_i$. Ekkor van olyan $j \geq i$, melyre $h_2 \in K_j$. Ha $h_2 \in K_i$ is teljesül, akkor (az indukciós feltevés miatt) $\pi^i \models h_2$ és így $\pi^i \models g$.

  Ha $h_2 \notin K_i$, akkor (az atom definíciója miatt) $h_1 \in K_i$ és $\mathbf{X}g \in K_i$. Akkor a G definíciója miatt $g \in K_{i+1}$.

  Kapjuk, hogy minden $i \leq k < j$-re $h_1 \in K_k$. Az indukciós feltevés miatt $\pi^j \models h_2$ és minden $i \leq k < j$-re $\pi^k \models h_1$. Kapjuk, hogy $\pi^i \models h_1 \mathbf{U} h_2 = g$.

  Tegyük fel, hogy $\pi^i \models g$. Legyen $j \geq i$ a legkisebb olyan szám, amelyre $\pi^j \models h_2$ és minden $i \leq k < j$-re $\pi^k \models h_1$. Az induikciós feltevés miatt $h_2 \in K_j$ és minden $i \leq k < j$-re $h_1 \in K_k$.

  Indirekt úton megmutatjuk, hogy $g \in K_i$.

LTL modell ellenőrzés tablóval

Tegyük fel, hogy $g \notin K_i$. Mivel $h_1 \in K_i$ (és $g \notin K_i$), ezért az atom definíciója miatt $\mathbf{X}g \notin K_i$, azaz $\mathbf{X}\neg g \in K_i$. A G definíciója miatt $\neg g \in K_{i+1}$, azaz $g \notin K_{i+1}$. Az érvelést folytatva kapjuk, hogy $g \notin K_j$, ami ellentmondás, mert $h_2 \in K_j$.

"$\Longrightarrow$"

Tegyük fel, hogy $M, s \models \mathbf{E}f$, azaz van egy olyan $\pi = s_0, s_1, \ldots$ $(s = s_0)$ út, amelyre $\pi \models f$. Legyen $i \geq 0$-re $K_i = \{g | g \in Cl(f)$ és $\pi^i \models g\}$.

Ekkor a következők teljesülnek:

- (s_i, K_i) atom. (Pl. adott $g \in Cl(f)$ esetén
  $g \in K_i \iff \pi^i \models g \iff \pi^i \not\models \neg g \iff \neg g \notin K_i$.)
- G-ben van él (s_i, K_i)-ből (s_{i+1}, K_{i+1})-be:
  $\mathbf{X}g \in K_i \iff \pi^i \models \mathbf{X}g \iff \pi^{i+1} \models g \iff g \in K_{i+1}$.

LTL modell ellenőrzés tablóval

- $(s_0, K_0), (s_1, K_1), \ldots$ $\mathbf{U}$-teljesítő út, mert $h_1 \mathbf{U} h_2 \in K_i \iff \pi^i \models h_1 \mathbf{U} h_2 \Longrightarrow (\exists j \geq i) \pi^j \models h_2 \iff (\exists j \geq i) h_2 \in K_j$.

Definíció: G egy erősen összefüggő C komponensét $\mathbf{U}$-teljesítőnek nevezzük, ha minden $A \in C$ atom és minden $f_1 \mathbf{U} f_2 \in K_A$ esetén van olyan $B \in C$, melyre $f_2 \in K_B$.

Lemma: G-ben egy (s, K) atomból akkor és csak akkor indul ki $\mathbf{U}$-teljesítő út, ha G-ben van olyan $\mathbf{U}$-teljesítő erősen összefüggő C komponens, hogy valamely $B \in C$-re (s, K)-ból út vezet B-be.

Bizonyítás: "$\Longrightarrow$"

Legyen C' az (s, K)-ból kiinduló $\mathbf{U}$-teljesítő út azon pontjainak halmaza, amelyek végtelen sokszor előfordulnak. Legyen C egy C'-t tartalmazó erősen összefüggő komponens, ekkor (s, K)-ból vezet út C-be.

LTL modell ellenőrzés tablóval

Megmutatjuk, hogy C $\mathbf{U}$-teljesítő. Legyen evégett $h_1 \mathbf{U} h_2 \in K_B$ valamely $B = (s_B, K_B) \in C$-re. Mivel C erősen összefüggő, van egy út B-ből C' egy D pontjába. Ha h_2 szerepel valahol ezen az úton, akkor C $\mathbf{U}$-teljesítő. Ha viszont h_2 nem szerepel ezen az úton, akkor a G definíciója miatt $h_1 \mathbf{U} h_2$ az út minden pontján - és így C'-nek a D pontján is - szerepel. Mivel C' egy $\mathbf{U}$-teljesítő út egy része, van egy (a D pontból elérhető) további pontja, amelyben szerepel h_2. Tehát ismét azt kapjuk, hogy C $\mathbf{U}$-teljesítő.

"$\Longleftarrow$"

Legyen C egy $\mathbf{U}$-teljesítő erősen összefüggő komponens és tfh. (s, K)-ból út vezet egy $B \in C$ pontba. Nyilvánvaló, hogy meg tudunk adni egy B-ből kiinduló $\mathbf{U}$-teljesítő utat. Ha viszont $h_1 \mathbf{U} h_2 \in K_D$ az (s, K)-ból B-be vezető út egy D pontjában, akkor az atom definíciója miatt vagy $h_2 \in K_D$ vagy $\mathbf{X}(h_1 \mathbf{U} h_2) \in K_D$ és ezért $h_1 \mathbf{U} h_2$ szerepel a D-re következő pontban. Így folytatva azt kapjuk, hogy h_2 szerepel a B-be vezető úton vagy $h_1 \mathbf{U} h_2$ szerepel magában a B pontban. Mindkét esetben kapunk egy (s, K)-ból kiinduló $\mathbf{U}$-teljesítő utat.

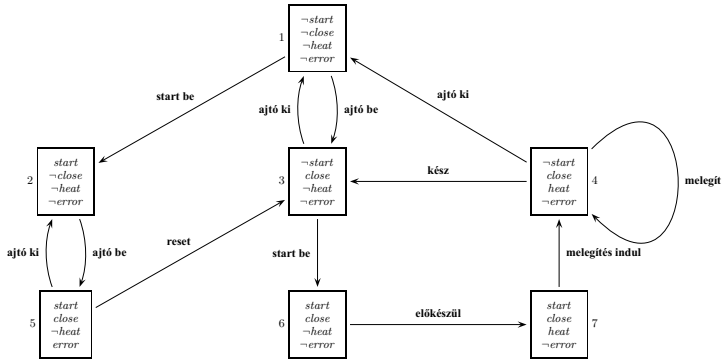
LTL modell ellenőrzés tablóval

Tétel: $M, s \models \mathbf{E} f \iff G$-ben van egy olyan (s, K) csúcs, amelyre $f \in K$ és G-nek van olyan $\mathbf{U}$-teljesítő erősen összefüggő C komponense, hogy (s, K)-ból út vezet C valamely elemébe.

Időbonyolultság: $O((|S| + |R|) \cdot 2^{O(|f|)})$.

Példa modell ellenőrzésre

A mikrohullámú sütő modellje



Példa tablóval történő LTL modellvizsgálatra

A mikrohullámú sütő mely állapotaiban teljesül az $\mathbf{A}((\neg heat)\mathbf{U}\ close)$ formula?

(Nem tud melegíteni, amíg az ajtó be nincs zárva.)

Mivel $\neg\mathbf{A}((\neg heat)\mathbf{U}\ close) \equiv \mathbf{E}\neg((\neg heat)\mathbf{U}\ close)$, az utóbbi formulát vizsgáljuk.

Legyen $f = (\neg heat)\mathbf{U}\ close$.

1) $Cl(\neg f) = \{\neg f, f, \mathbf{X}f, \neg\mathbf{X}f, \mathbf{X}\neg f, heat, \neg heat, close, \neg close\}$

2) Atomok konstrukciója
   $f \in K_A \iff close \in K_A$ vagy $\neg heat \in K_A$ és $\mathbf{X}f \in K_A$.

Csak azokat az atomi változókat vizsgáljuk, amelyek előfordulnak f-ben!

Példa tablóval történő LTL modellv ellenőrzésre

Az 1 és 2 állapotok tartalmazzák a $\neg close$ és a $\neg heat$ atomi állításokat. Ezért az 1 és 2 állapotokkal összekapcsolhatók a

$K_1' = \{\neg close, \neg heat, f, \mathbf{X}f\}$ és $K_1'' = \{\neg close, \neg heat, \neg f, \mathbf{X}\neg f, \neg\mathbf{X}f\}$

halmazok. Tehát $(1, K_1'), (2, K_1'), (1, K_1''), (2, K_1'')$ atomok.

A 3, 5 és 6 állapotok tartamazzák a $close$ és a $\neg heat$ atomi állításokat. Ezen állapotok összekapcsolhatók a

$K_2' = \{close, \neg heat, f, \mathbf{X}f\}$ és $K_2'' = \{close, \neg heat, f, \mathbf{X}\neg f, \neg\mathbf{X}f\}$

halmazokkal. Tehát minden $i \in \{3, 5, 6\}$-re (i, K_2') és (i, K_2'') atomok.

Példa tablóval történő LTL modell ellenőrzésre

A 4 és 7 állapotok tartamazzák a $close$ és a $heat$ atomi állításokat. Ezen állapotok összekapcsolhatók a

$K_3' = \{close, heat, f, \mathbf{X}f\}$ és $K_3'' = \{close, heat, f, \mathbf{X}\neg f, \neg\mathbf{X}f\}$

halmazokkal. Tehát minden $i \in \{4, 7\}$-re (i, K_3') és (i, K_3'') atomok.

3) A G gráf konstrukciója. Például

$$(1, K_1') \to (2, K_1')$$

átmenet, mivel a mikrohullámú sütő modellben $1\mathbf{R}2$, továbbá $\mathbf{X}f \in K_1'$ és $f \in K_1'$. Hasonló okok miatt

$$(1, K_1'') \to (2, K_1'')$$

is átmenet. Ugyanakkor nincs átmenet $(1, K_1')$-ből $(2, K_1'')$-be, mert $\mathbf{X}f \in K_1'$, de $f \notin K_1''$.

Példa tablóval történő LTL modell ellenőrzésre

Így megkonstruálható a teljes gráf.

A főtétel szerint egy $1 \le s \le 7$ állapotra $M, s \models \neg f$ akkor és csak akkor, ha van egy olyan (s, K) alakú atom, melyre $\neg f \in K$ és (s, K)-ból út vezet G-ben egy $\mathbf{U}$-teljesítő erősen összefüggő C komponens valamely pontjába.

Ilyen (s, K) pár nem lesz, ezért minden s-re

$M, s \not\models \neg f$

$\iff M, s \not\models \mathbf{E}\neg f$

$\iff M, s \models \mathbf{A}f.$

CTL operátorok fixpont jellemzése

Definíció: Legyen S egy halmaz, $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ pedig egy leképezés. Akkor

1) τ monoton, ha minden $P, Q \subseteq S$-re, ha $P \subseteq Q$, akkor $\tau(P) \subseteq \tau(Q)$,
2) τ $\cup$-folytonos, ha minden $P_1 \subseteq P_2 \subseteq \dots$ sorozatra $\tau(\bigcup_{i=1}^{\infty} P_i) = \bigcup_{i=1}^{\infty} \tau(P_i)$,
3) τ $\cap$-folytonos, ha minden $P_1 \supseteq P_2 \supseteq \dots$ sorozatra $\tau(\bigcap_{i=1}^{\infty} P_i) = \bigcap_{i=1}^{\infty} \tau(P_i)$,
4) P a τ egy fixpontja, ha $\tau(P) = P$,
5) P a τ legkisebb (legnagyobb) fixpontja, ha P fixpont, és τ minden további Q fixpontjára $P \subseteq Q$ $(Q \subseteq P)$ teljesül,
6) minden $P \subseteq S$-re $\tau^0(P) = P$ és minden $i \geq 0$-ra $\tau^{i+1}(P) = \tau(\tau^i(P))$.

CTL operátorok fixpont jellemzése

Lemma: Ha $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ $\cup$-folytonos, akkor monoton is.

Bizonyítás: Tegyük fel, hogy $P \subseteq Q$. Akkor

$\tau(P \cup Q) = \tau(P) \cup \tau(Q)$ $\quad$ (τ $\cup$-folytonos)

$\tau(P \cup Q) = \tau(Q)$ $\quad$ (mert $P \subseteq Q$)

Tehát $\tau(P) \cup \tau(Q) = \tau(Q) \iff \tau(P) \subseteq \tau(Q)$.

Lemma: Ha $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ $\cap$-folytonos, akkor monoton is.

Bizonyítás: Tegyük fel, hogy $P \subseteq Q$. Akkor

$\tau(P \cap Q) = \tau(P) \cap \tau(Q)$ $\quad$ (τ $\cap$-folytonos)

$\tau(P \cap Q) = \tau(P)$ $\quad$ (mert $P \subseteq Q$)

Tehát $\tau(P) \cap \tau(Q) = \tau(P) \iff \tau(P) \subseteq \tau(Q)$.

CTL operátorok fixpont jellemzése

Lemma: Ha S véges és $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ monoton, akkor τ $\cup$-folytonos és $\cap$-folytonos is.

Bizonyítás: Az $\cup$-folytonosságot bizonyítjuk. Legyen $P_1 \subseteq P_2 \subseteq \dots \subseteq S$. Mivel S véges, ezért van olyan j_0, hogy minden $j \geq j_0$-ra $P_j = P_{j_0}$ és minden $1 \leq j < j_0$-ra $P_j \subseteq P_{j_0}$. Tehát $\bigcup_i P_i = P_{j_0}$.

Másrészt minden $j \geq j_0$-ra $\tau(P_j) = \tau(P_{j_0})$ és, mivel τ monoton, minden $j < j_0$-ra $\tau(P_j) \subseteq \tau(P_{j_0})$.

Tehát $\bigcup_i \tau(P_i) = \tau(P_{j_0}) = \tau(\bigcup_i P_i)$.

A $\cap$-folytonosság hasonlóan bizonyítható.

Következmény: Legyen S véges halmaz, és $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ egy leképezés. Ekkor τ monoton $\iff$ τ $\cup$-folytonos $\iff$ τ $\cap$-folytonos.

CTL operátorok fixpont jellemzése

Lemma: Legyen $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ egy leképezés.

1) Ha τ $\cup$-folytonos, akkor $\bigcup_{i=0}^{\infty} \tau^i(\emptyset)$ a τ legkisebb fixpontja.
2) Ha τ $\cap$-folytonos, akkor $\bigcap_{i=0}^{\infty} \tau^i(S)$ a τ legnagyobb fixpontja.

Bizonyítás: 1)

$\tau(\bigcup_{i=0}^{\infty} \tau^i(\emptyset)) = \bigcup_{i=0}^{\infty} \tau(\tau^i(\emptyset)) = \bigcup_{i=1}^{\infty} \tau^i(\emptyset) = \bigcup_{i=0}^{\infty} \tau^i(\emptyset)$

Tehát $\bigcup_{i=0}^{\infty} \tau^i(\emptyset)$ fixpont.

Legyen P a τ fixpontja.

$\emptyset \subseteq P \Longrightarrow \tau^i(\emptyset) \subseteq \tau^i(P) = P$, minden $i \geq 0$-ra $\Longrightarrow \bigcup_{i=0}^{\infty} \tau^i(\emptyset) \subseteq P$.

A 2) állítás hasonlóan bizonyítható.

CTL operátorok fixpont jellemzése

Tétel: Legyen S egy n elemű halmaz és $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ egy monoton leképezés. Akkor $\tau^n(\emptyset)$ a τ legkisebb és $\tau^n(S)$ a τ legnagyobb fixpontja.

Bizonyítás: Mivel τ monoton, ezért minden $i \leq n$-re $\tau^i(\emptyset) \subseteq \tau^n(\emptyset)$.
Másrészt mivel S n elemű, ezért minden $j > n$-re $\tau^j(\emptyset) = \tau^n(\emptyset)$. Tehát

$$\bigcup_{i=0}^{\infty} \tau^i(\emptyset) = \tau^n(\emptyset)$$

Hasonlóan kapjuk, hogy

$$\bigcap_{i=0}^{\infty} \tau^i(S) = \tau^n(S)$$

Jelölés: τ legkisebb (legnagyobb) fixpontját $\mu Z.\tau$-val ($\nu Z.\tau$-val) jelöljük.

CTL operátorok fixpont jellemzése

A következők egy tetszőleges $M = (S, R, L)$ Kripke struktúrára vonatkoznak.

Jelölés: Tetszőleges f CTL formulára legyen

$[\![f]\!] = \{s \in S \mid M, s \models f\}$.

Akkor teljesülnek a következők:

1) Ha $f \in AP$, akkor $[\![f]\!] = \{s \in S \mid f \in L(s)\}$
2) $[\![\neg f]\!] = S \setminus [\![f]\!]$
3) $[\![f_1 \vee f_2]\!] = [\![f_1]\!] \cup [\![f_2]\!]$
4) $[\![\mathbf{EX}f]\!] = \{s \in S \mid (\exists t) : (sRt \wedge t \in [\![f]\!])\}$
5) $[\![\mathbf{EG}f]\!] = [\![f]\!] \cap \{s \in S \mid (\exists t) : (sRt \wedge t \in [\![\mathbf{EG}f]\!])\}$!
6) $[\![\mathbf{E}(f\mathbf{U}g)]\!] = [\![g]\!] \cup ([\![f]\!] \cap \{s \in S | (\exists t) : (sRt \wedge t \in [\![\mathbf{E}(f\mathbf{U}g)]\!])\})$!

CTL operátorok fixpont jellemzése

$[\![\mathbf{EG}f]\!]$ kiszámítása

Legyen $f \wedge \mathbf{EX} : \mathcal{P}(S) \to \mathcal{P}(S)$ a következő függvény:

$f \wedge \mathbf{EX}(Z) = [\![f]\!] \cap \{s \in S | (\exists t) : (sRt \wedge t \in Z)\}$.

Tétel: $f \wedge \mathbf{EX}(Z)$ monoton és $[\![\mathbf{EG}f]\!] = \nu Z.f \wedge \mathbf{EX}(Z)$.

Bizonyítás: 1) $f \wedge \mathbf{EX}(Z)$ monoton. Legyen $P_1 \subseteq P_2$

$s \in f \wedge \mathbf{EX}(P_1)$

$\iff s \models f \wedge (\exists t) : (sRt \wedge t \in P_1)$

$\implies s \models f \wedge (\exists t) : (sRt \wedge t \in P_2)$

$\iff s \models f \wedge \mathbf{EX}(P_2)$

2) $[\![\mathbf{EG}f]\!]$ az $f \wedge \mathbf{EX}(Z)$ fixpontja.

$s_0 \in [\![\mathbf{EG}f]\!]$

$\iff$ van $s_0, s_1, \ldots$ út, hogy $(\forall k) : s_k \models f$

$\iff s_0 \models f$ és $s_1 \models \mathbf{EG}f \iff s_0 \models f$ és $s_0 \models \mathbf{EXEG}f$

$\iff s_0 \in [\![f]\!] \cap [\![\mathbf{EXEG}f]\!] = (f \wedge \mathbf{EX})([\![\mathbf{EG}f]\!])$

CTL operátorok fixpont jellemzése

3) $\nu Z.f \wedge \mathbf{EX}(Z) = [\![\mathbf{EG}f]\!]$

a) $[\![\mathbf{EG}f]\!] \subseteq \nu Z.f \wedge \mathbf{EX}(Z)$ (ez a legnagyobb)

b) $s \in \nu Z.f \wedge \mathbf{EX}(Z) \implies s \in (f \wedge \mathbf{EX}(Z))(\nu Z.f \wedge \mathbf{EX}(Z))$

$\implies s \models f \wedge (\exists t) : (sRt \wedge t \in \nu Z.f \wedge \mathbf{EX}(Z))$

$\implies$ van $s = s_0, s_1, \ldots$ út, hogy minden $k \geq 0$-ra $s_k \models f$

$\implies s \in [\![\mathbf{EG}f]\!]$

Tehát $\nu Z.f \wedge \mathbf{EX}(Z) \subseteq [\![\mathbf{EG}f]\!]$.

4) Mivel $f \wedge \mathbf{EX}(Z)$ monoton és S véges,

$\nu Z.f \wedge \mathbf{EX}(Z) = (f \wedge \mathbf{EX})^n(S)$, ahol n az S elemeinek száma.

CTL operátorok fixpont jellemzése

$[\![\mathbf{E}(f\mathbf{U}g)]\!]$ kiszámítása

Legyen $g \vee (f \wedge \mathbf{EX}) : \mathcal{P}(S) \to \mathcal{P}(S)$ a következő függvény

$g \vee (f \wedge \mathbf{EX}(Z)) = [\![g]\!] \cup ([\![f]\!] \cap \{s | (\exists t) : (sRt \wedge t \in Z)\})$

Tétel: $g \vee (f \wedge \mathbf{EX}(Z))$ monoton és $[\![\mathbf{E}(f\mathbf{U}g)]\!] = \mu Z.g \vee (f \wedge \mathbf{EX}(Z))$.

Bizonyítás: 1) $g \vee (f \wedge \mathbf{EX}(Z))$ monoton (ugyanúgy mint az előbb).

2) $[\![\mathbf{E}(f\mathbf{U}g)]\!]$ a $g \vee (f \wedge \mathbf{EX}(Z))$ fixpontja.

$s_0 \in [\![\mathbf{E}(f\mathbf{U}g)]\!]$

$\iff$ van $s_0, s_1, \ldots$ út, hogy $s_0 \models g$ vagy ($s_0 \models f$ és $s_1 \models \mathbf{E}(f\mathbf{U}g)$)

$\iff$ $s_0 \models g$ vagy ($s_0 \models f$ és $s_0 \models \mathbf{EXE}(f\mathbf{U}g)$)

$\iff$ $s_0 \in [\![g]\!] \cup ([\![f]\!] \cap [\![\mathbf{EXE}(f\mathbf{U}g)]\!])$

$\iff$ $s_0 \in g \vee (f \wedge \mathbf{EX}([\![\mathbf{E}(f\mathbf{U}g)]\!]))$.

CTL operátorok fixpont jellemzése

3) $[\![\mathbf{E}(f\mathbf{U}g)]\!] = \mu Z.(g \vee (f \wedge \mathbf{EX}(Z)))$

a) $\mu Z.(g \vee (f \wedge \mathbf{EX}(Z))) \subseteq [\![\mathbf{E}(f\mathbf{U}g)]\!]$

b) $s \in [\![\mathbf{E}(f\mathbf{U}g)]\!]$

$\Longrightarrow$ van olyan $s = s_0, s_1, \ldots$ út és $j \geq 1$, hogy $s_j \models g$ és $\forall (l < j) : s_l \models f$.

Állítás: $s \in (g \vee (f \wedge \mathbf{EX}))^j(\emptyset)$

Bizonyítás: j szerinti indukcióval

$\Longrightarrow s \in \bigcup_i (g \vee (f \wedge \mathbf{EX}))^i(\emptyset) = \mu Z.(g \vee (f \wedge \mathbf{EX}(Z)))$.

4) Mivel $g \vee (f \wedge \mathbf{EX}(Z))$ monoton és S véges,

$\nu Z.g \vee (f \wedge \mathbf{EX}(Z)) = (g \vee (f \wedge \mathbf{EX}(Z)))^n(\emptyset)$, ahol n az S elemeinek száma.

CTL operátorok fixpont jellemzése

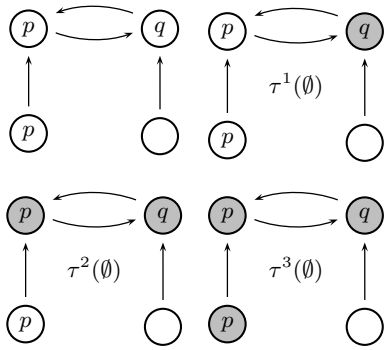
A legkisebb és legnagyobb fixpontok kiszámítása monoton $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ leképezés esetén.

function Lfp(τ : leképezés)
  var Q, Q';
  $Q = \emptyset$; $Q' = \tau(Q)$;
  while $(Q \neq Q')$ **do**
    $Q = Q'$; $Q' = \tau(Q)$;
  endwhile
  return (Q);
endfunction
function Gfp(τ : leképezés)
  var Q, Q';
  $Q = S$; $Q' = \tau(Q)$;
  while $(Q = Q')$ **do**
    $Q = Q'$; $Q' = \tau(Q)$;
  endwhile
  return (Q);
endfunction

CTL operátorok fixpont jellemzése

Számítsuk ki a $[\![\mathbf{E}(p\mathbf{U}q)]\!] = \bigcup_i (q \vee (p \wedge \mathbf{EX}))^i(\emptyset)$ halmazt, ahol

$q \vee (p \wedge \mathbf{EX}(Z)) = [\![q]\!] \cup ([\![p]\!] \cap \{s | (\exists t) : (s\mathbf{R}t \wedge t \in Z)\})$.



Szimbolikus CTL modell ellenőrzés

A címkézős CTL modellvizsgálat időbonyolultsága lineáris mind a gráf mind a formula méretében számolva. Ugyanakkor a modell mérete gyakorlati feladatoknál nagyon nagy lesz. Valójában a modell mérete exponenciális függvénye az atomi változók számának (állapotrobbanás).
Ezért célszerű a Kripke struktúráknak egy gazdaságos reprezentációját választani: rendezett bináris döntési diagramok (OBDD). Az OBDD-kal támogatott modellvizsgáltot szimbolikus modellvizsgálatnak nevezzük.

Bináris döntési diagramok (BDD)

Bináris döntési fa (BDF)

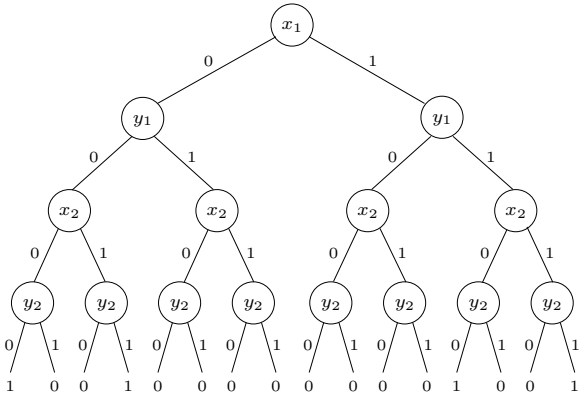
- teljesen kiegyensúlyozott véges, bináris fa
- az ugyanazon nemterminális (nem levél) szinten lévő csúcsok ugyanazon x változóval vannak megcímkézve
- minden terminális(levél) szinten lévő csúcs 0-val vagy 1-gyel van megcímkézve

Az n változós Boole függvények $(\varphi : \{0,1\}^n \to \{0,1\})$ és az n változót tartalmazó BDF-ek bijektív módon megfeleltethetünk egymásnak.

φ igazságtáblája	$\leftrightarrow$	φ BDF reprezentációja
$x = 0$	$\sim$	x bal fia
$x = 1$	$\sim$	x jobb fia

Bináris döntési diagramok (BDD)

Példa. A $\varphi(x_1, y_1, x_2, y_2) = (x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2)$ 4 változós Boole függvénynek megfelelő BDF.



Példák:

$$\varphi(0,1,0,1) = 0$$
$$\varphi(0,0,1,1) = 1$$

Bináris döntési diagramok (BDD)

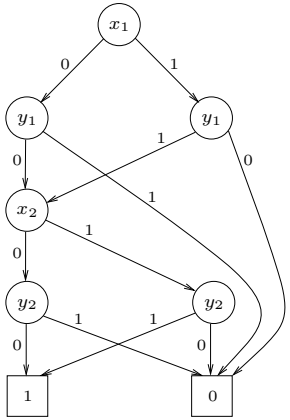
A BDF redundáns ábrázolási forma, mivel azonos részfákat többszörösen ábrázol.
A Bináris döntési diagram (BDD)

- egy gyökérponttal rendelkező irányított véges aciklikus gráf
- minden nemterminális d csúcs egy $var(d)$ változóval van megcímkézve és belőle 2 él indul ki, a $low(d)$ és a $high(d)$ csúcsokba
- minden terminális d csúcs (nincs kimenő éle) egy $value(d) \in \{0,1\}$ értékkel van megcímkézve

Minden BDF egyben BDD is.
Egy BDD alkalmas arra, hogy az izomorf részgráfokat csak egy példányban tárolja.

Bináris döntési diagramok (BDD)

Példa. A $\varphi(x_1, y_1, x_2, y_2) = (x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2)$ függvény ábrázolása BDD-vel.



$$\varphi(0,1,0,1) = 0$$
$$\varphi(0,0,1,1) = 1$$

Bináris döntési diagramok (BDD)

Definíció. Legyen D egy BDD, melyben legfeljebb az $x_1, \ldots, x_n$ változók szerepelnek és amelynek a gyökere d. A D által reprezentált $\varphi_d(x_1, \ldots, x_n)$ Boole függvényt a következőképpen definiáljuk.

(i) Ha d terminális csúcs (D-nek csak egy csúcsa van), akkor
$\varphi_d(x_1, \ldots, x_n) = value(d)$

(ii) Ha d nemterminális csúcs és $var(d) = x_i$, akkor
$\varphi_d(x_1, \ldots, x_n) = (\neg x_i \wedge \varphi_{low(d)}(x_1, \ldots, x_n)) \vee (x_i \wedge \varphi_{high(d)}(x_1, \ldots, x_n))$

Két BDD ekvivalens, ha ugyanazt a Boole függvényt reprezentálják.

Bináris döntési diagramok (BDD)

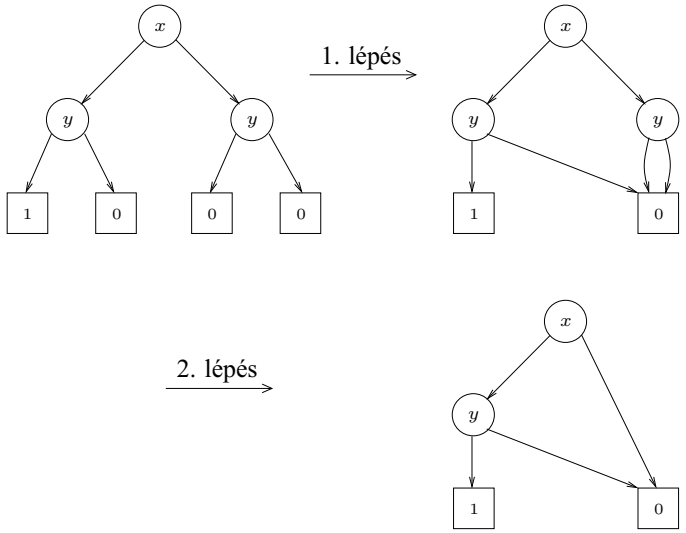
BDD redukáló algoritmus ($Reduce$)
Input. Egy D BDD
Output. Egy D-vel ekvivalens BDD.

1. Ha D tartalmaz 0 csúcsot (1 csúcsot), akkor egy d kivételével töröljük valamennyit és a törölt csúcsokba vezető éleket irányítsuk át d-be.
2. Ha egy d nemterminális csúcs mindkét kimenő éle ugyanazon d' csúcsba vezet, akkor töröljük a d csúcsot és a bele vezető éleket irányítsuk át d'-be.
3. Ha d és d' csúcsok izomorf részgráfok gyökerei, akkor töröljük a d' gyökerű részgráfot és a bele vezető éleket irányítsuk át d-be.

Egy BDD *redukált*, ha a redukáló algoritmus változatlanul hagyja.

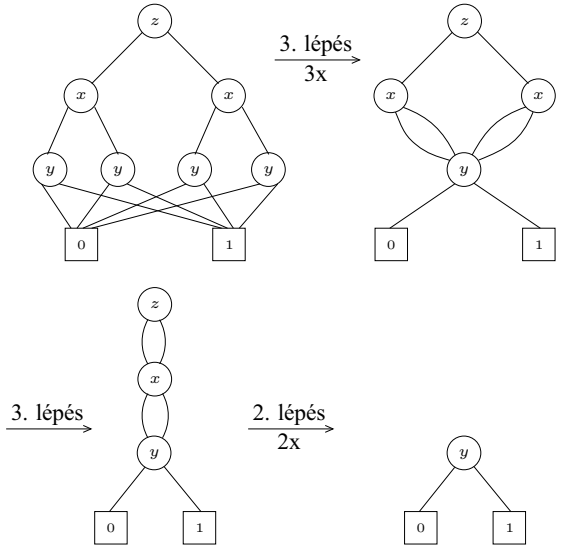
Bináris döntési diagramok (BDD)

Példák a $Reduce$ algoritmus alkalmazására.



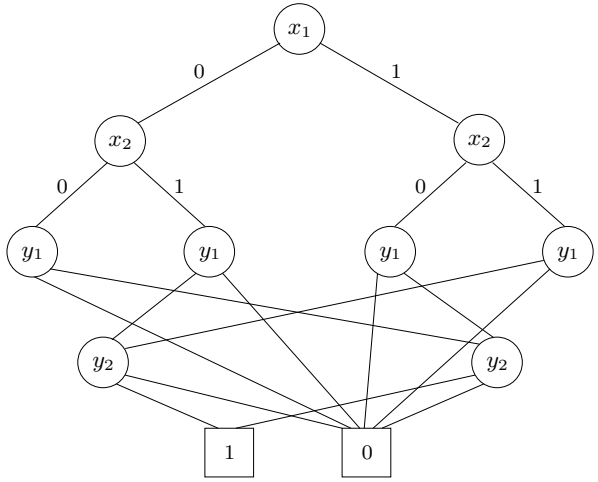
Bináris döntési diagramok (BDD)

Példák a $Reduce$ algoritmus alkalmazására.



Bináris döntési diagramok (BDD)

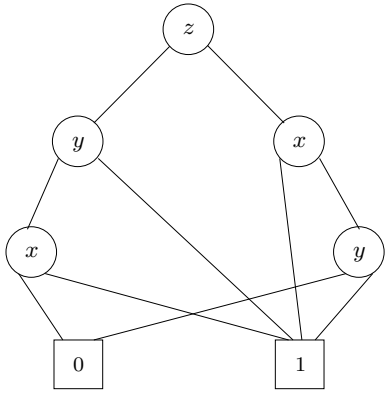
Egy Boole függvény BDD-vel való reprezentációja függ a változók sorrendjétől. Például, a $\varphi(x_1, y_1, x_2, y_2) = (x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2)$ függvényt reprezentáló BDD a következő is lehet.



Bináris döntési diagramok (BDD)

Definíció. Egy D BDD-t rendezettnek nevezünk (OBDD), ha megadható a benne szereplő változóknak egy olyan $x_1, \ldots, x_n$ rendezése, hogy bármely, D gyökértől valamely levélig vezető úton a változók a megadott sorrendben követik egymást.

Példa. Az eddig szereplő valamennyi BDD egyben OBDD is. A Következő BDD nem rendezett:



Bináris döntési diagramok (BDD)

Definíció. A D_1 és D_2 OBDD-k ugyanúgy rendezettek, ha mindkettő a változók egy és ugyanazon $x_1, \ldots, x_n$ sorrendjére nézve rendezett.

Tétel. A D_1 és D_2 ugyanúgy rendezett, redukált OBDD-k akkor és csak akkor reprezentálják ugyanazon Boole függvényt, ha izomorfak.

$$\varphi_1 = \varphi_2 \iff D_1 \text{ és } D_2 \text{ izomorfak}$$

Más szóval, egy adott φ függvény és a változók egy adott sorrendje esetén, a φ-t ezen sorrendre vonatkozóan rendezett, redukált OBDD izomorfizmus erejéig egyértelmű.

Bináris döntési diagramok (BDD)

A redukált BDD-k alkalmazása

1. Egy $\varphi(x_1, \ldots, x_n)$ Boole függvény akkor és csak akkor nem függ x_i-től, ha a φ-t reprezentáló bármely redukált BDD-ben nem szerepel x_i.
2. $\varphi(x_1, \ldots, x_n) = \psi(x_1, \ldots, x_n)$ akkor és csak akkor, ha az őket reprezentáló, ugyanazon rendezés szerint rendezett, redukált OBDD-k izomorfak (megegyeznek).
3. $\varphi(x_1, \ldots, x_n)$ érvényes $\iff$ az őt reprezentáló, redukált BDD az $\boxed{1}$ BDD.
4. $\varphi \to \psi$ akkor és csak akkor érvényes, ha a $\varphi \wedge (\neg\psi)$-t reprezentáló redukált BDD $\boxed{0}$.

OBDD-kre vonatkozó algoritmusok

1. $Reduce(D)$ (redukált, lásd fentebb)
2. $Restrict(D, x_i, b)$, ahol $b \in \{0, 1\}$. Megkonstuálja a $\varphi(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n) = \varphi \mid_{x_i=b}$ függvényt reprezentáló OBDD-t, ahol φ a D által reprezentált Boole függvény. Minden olyan d csúcsra, amelyből él vezet egy x_i-vel címkézett d' csúcsba, irányítsuk át az élet $low(d')$-be, ha $b = 0$ és $high(d')$-be, ha $b = 1$. Ezután hívjuk meg a $Reduce$ eljárást.
3. $Apply(D_1, D_2, *)$, ahol $*$ egy kétváltozós Boole függvény, D_1, D_2 pedig ugyanúgy rendezett OBDD-k. Kiszámolja a $\varphi_1(x_1, \ldots, x_n) * \varphi_2(x_1, \ldots, x_n)$ Boole függvényt reprezentáló OBDD-t, ahol φ_1 és φ_2 a D_1 és D_2 által reprezentált Boole függvények.

OBDD-kre vonatkozó algoritmusok

Az $Apply(D_1, D_2, *)$ a következő tételen alapul.
Tétel. (Shannon kiterjeszés) Tetszőleges φ Boole függvényre és x változóra

$$\varphi = (\neg x \wedge \varphi \mid_{x=0}) \vee (x \wedge \varphi \mid_{x=1})$$

Tehát $\varphi_1 * \varphi_2 =$
$(\neg x_i \wedge (\varphi_1 \mid_{x_i=0} * \varphi_2 \mid_{x_i=0})) \vee (x_i \wedge (\varphi_1 \mid_{x_i=1} * \varphi_2 \mid_{x_i=1}))$

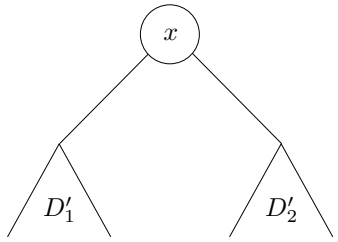
Az $Apply(D_1, D_2, *)$ eljárás: tegyük fel, hogy D_1 és D_2 gyökere d_1 és d_2

- ha d_1 és d_2 terminálisok, akkor az eredmény a $value(d_1) * value(d_2)$ OBDD
- d_1 és d_2 nemterminálisok, $var(d_1) = var(d_2) = x$

OBDD-kre vonatkozó algoritmusok

- d_1 és d_2 nemterminálisok, $var(d_1) = var(d_2) = x$

Az eredmény az



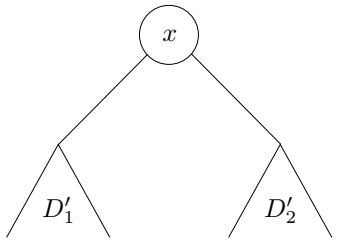
OBDD, ahol

D'_1 a $\varphi_1 \mid_{x=0} * \varphi_2 \mid_{x=0}$,
D'_2 a $\varphi_1 \mid_{x=1} * \varphi_2 \mid_{x=1}$

függvényt reprezentáló OBDD-k.

OBDD-kre vonatkozó algoritmusok

- $var(d_1) = x$ és (d_2 terminális vagy $var(d_2) = x'$), ahol $x' > x$. Ekkor φ_2 nem függ x-től (azaz D_2-ben nincs x_2 csúcs). Az eredmény az



  OBDD, ahol D'_1 a $\varphi_1 \mid_{x=0} * \varphi_2$, D'_2 a $\varphi_1 \mid_{x=1} * \varphi_2$ függvényt reprezentáló OBDD

- ($var(d_1) = x$ vagy d_1 terminális) és $var(d_2) = x'$,ahol $x > x'$. Az eredményt az előző esetekkel analóg módon kapjuk.

OBDD-kre vonatkozó algoritmusok

4. $Exist(D, x)$
   Tetszőleges φ Boole függvény esetén

$$\exists x.\varphi = \varphi \mid_{x=0} \vee \varphi \mid_{x=1}$$

   $Exist(D, x)$ a $\exists x.\varphi$ függvényt reprezentáló OBDD, ahol D a φ-t reprezentáló OBDD.

   Tehát $Exist(D, x) = Apply(Restrict(D, x, 0), Restrict(D, x, 1), \vee)$.

   Általánosítása
   $\exists x_{i_1} \dots \exists x_{i_K}.\varphi = \exists x_{i_1}(\dots \exists x_{i_K}.\varphi) \dots)$, röviden $\exists \overline{x}.\varphi$, ahol $\overline{x} = x_{i_1} \dots x_{i_K}$.

Kripke struktúrák reprezentálása OBDD-kkel

Legyen $M = (S, R, L)$ egy Kripke struktúra és legyen $x_1, \dots, x_n$ az atomi állítások ($\mathbf{AP}$) egy tetszőleges, de rögzített sorrendje.

1. S elemeit és részhalmazait n változós Boole függvényként fogjuk fel, majd OBDD-vel reprezentáljuk.
   - $s \in S$-hez rendeljük hozzá a $\varphi_s = v_1 \wedge \dots \wedge v_n$ Boole függvényt, ahol

$$v_i = \begin{cases} x_i & \text{ha } x_i \in L(s) \\ \neg x_i & \text{ha } x_i \notin L(s) \end{cases}$$

     Megjegyzés. A hozzárendelés csak akkor lesz injektív, ha $s_1 \neq s_2$-ből következik $L(s_1) \neq L(s_2)$. Ez azonban új atomi állítások hozzávételével mindig elérhető.

Kripke struktúrák reprezentálása OBDD-kkel

- $T = \{s_1, \dots, s_k\} \subseteq S$-hez rendeljük hozzá a $\varphi_T = \varphi_{s_1} \vee \dots \vee \varphi_{s_k}$ Boole függvényt, ahol φ_{s_i} az s_i állapothoz rendelt Boole függvény. Az így kapott Boole függvényeket reprezentáljuk OBDD-kkel.

2. R-et ugyancsak Boole függvényként fogjuk fel. Vegyük az $x_1, \dots, x_n$ változók $x'_1, \dots, x'_n$ diszjunkt másolatait. Minden $(s, s') \in R$ átmenethez rendeljük hozzá a

$$\varphi_s(x_1, \dots, x_n) \wedge \varphi_{s'}(x'_1, \dots, x'_n)$$

   $2n$ változós Boole függvényt. Ezután R-hez rendeljük hozzá a

$$\varphi_R = \bigvee_{(s,s') \in R} \varphi_s(x_1, \dots, x_n) \wedge \varphi_{s'}(x'_1, \dots, x'_n)$$

   Boole függvényt. Az így kapott φ_R Boole függvényt reprezentáljuk OBDD-vel.

Szimbolikus CTL modell ellenőrzés

Legyen $M = (S, R, L)$ egy Kripke struktúra és f egy CTL formula. A szimbolikus modellvizsgálat lényege, hogy mind S részhalmazait, mind R -et OBDD-kkel reprezentáljuk. A módszer előnye

1. nagy állapotszám és átmenetszám esetén is viszonylag kisméretű objektumokkal dolgozunk
2. $[\![f]\!]$ kiszámítható az OBDD-kkel megismert algoritmusok alkalmazásával

Jelölések

- D_R-rel jelöljük a φ_R-et (vagyis R-et) reprezentáló OBDD-t,
- tetszőleges $T \subseteq S$-re D_T-vel jelöljük a φ_T-t (vagyis T-t) reprezentáló OBDD-t.

Szimbolikus CTL modell ellenőrzés

Adott f formula esetén $D_{[\![f]\!]}$ kiszámítása:

1. Ha $f \in \mathbf{AP}$, akkor $D_{[\![f]\!]}$ az $\{s \in S \mid f \in L(s)\}$ halmazt reprezentáló OBDD.
2. $D_{[\![\neg f]\!]}$-et úgy kapjuk $D_{[\![f]\!]}$-ből, hogy benne a 0-t és 1-et felcseréljük.
3. $D_{[\![f \vee g]\!]} = Apply(D_{[\![f]\!]}, D_{[\![g]\!]}, \vee)$
4. Az $\mathbf{EX}f$ formula. Az $[\![f]\!]$-et reprezentáló $\varphi_{[\![f]\!]}$ OBDD-ben az $x_1, \ldots, x_n$ változókat nevezzük át $x'_1, \ldots, x'_n$-re. Akkor
$$\varphi_{[\![\mathbf{EX}f]\!]} = \exists \overline{x'} \varphi_R((\overline{x}, \overline{x'}) \wedge \varphi_{[\![f]\!]}(\overline{x'}))$$
Így
$$D_{[\![\mathbf{EX}f]\!]} = Exists(Apply(D_R, D_{[\![f]\!]}, \wedge), \overline{x'})$$
Hasonlóan, tetszőleges $Z \subseteq S$ esetén D_Z-ből meghatározható az
$$\mathbf{EX}(Z) = \{s \in S \mid (\exists t)(s\, R\, t \wedge t \in Z)\}$$
halmazt reprezentáló $D_{\mathbf{EX}(Z)}$ OBDD.

Szimbolikus CTL modell ellenőrzés

5. Az $\mathbf{EG}f$ formula.
   Mivel $[\![\mathbf{EG}f]\!] = \nu Z.(f \wedge \mathbf{EX}(Z))$,
   $D_{[\![\mathbf{EG}f]\!]}$ a következő, OBDD-kből álló sorozat határértéke lesz:
   (i) $D_0 =$ az S-et reprezentáló OBDD;
   (ii) $D_{n+1} = Apply(D_f, D'_n, \wedge)$, ahol
   $D'_n = Exists(Apply(D_R, D_n, \wedge), \overline{x'})$
6. Az $\mathbf{E}(f\mathbf{U}g)$ formula. Mivel $[\![\mathbf{E}(f\mathbf{U}g)]\!] = \mu Z.(g \vee (f \wedge \mathbf{EX}(Z)))$,
   $D_{[\![\mathbf{E}(f\mathbf{U}g)]\!]}$ a következő, OBDD sorozat határértéke lesz:
   (i) $D_0 =$ az $\emptyset$-t reprezentáló OBDD;
   (ii) $D_{n+1} = Apply(D_g, Apply(D_f, D'_n, \wedge), \vee)$, ahol
   $D'_n = Exists(Apply(D_R, D_n, \wedge), \overline{x'})$

Korrekt szimbolikus CTL modell ellenőrzés

Legyen $M = (S, R, L, F)$ egy korrekt Kripke struktúra, ahol $F = \{g_1, \ldots g_n\}$ most CTL formulák halmaza.
Egy π utat korrektnek nevezünk, ha minden $1 \leq i \leq n$-re
$$inf(\pi) \cap \{s \in S \mid s \models g_i\} \neq \emptyset,$$
vagyis a π úton végtelen sok olyan s állapot van, melyre $s \models g_i$.
(A korrekt út ekvivalens megfogalmazása.)
A $\models_F$ definícióját a korrekt út segítségével ugyanúgy definiáljuk, mint korábban. Tetszőleges f formula esetén legyen
$$[\![f]\!]_F = \{s \in S \mid M, s \models_F f\}.$$

Korrekt szimbolikus CTL modell ellenőrzés

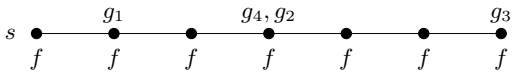
Ha $f \in \mathbf{AP}$, akkor $[\![f]\!]_F = \{s \in [\![f]\!] \mid$ van s-ből kiinduló korrekt út$\}$, továbbá

- $[\![\neg f]\!]_F = S - [\![f]\!]_F$,
- $[\![f \vee g]\!]_F = [\![f]\!]_F \cup [\![g]\!]_F$

A továbbiakban megadjuk $[\![\mathbf{EG}f]\!]_F$, $[\![\mathbf{EX}f]\!]_F$ és $[\![\mathbf{E}(f\mathbf{U}g)]\!]_F$ kiszámítási módját.

1. $[\![\mathbf{EG}f]\!]_F$ kiszámítása
   $[\![\mathbf{EG}f]\!]_F$ a legbővebb olyan $Z \subseteq S$ halmaz, melyre teljesül, hogy
   (i) minden $s \in Z$-re $s \models f$
   (ii) minden $s \in Z$-ből indul ki olyan véges (n hosszúságú), Z-beli elemekből álló szakasz, melyen minden $1 \leq i \leq n$-re van olyan állapot, mely kielégíti g_i-t.

$F = \{g_1, g_2, g_3, g_4\}$



Korrekt szimbolikus CTL modell ellenőrzés

$[\![\mathbf{EG}f]\!]_F$ fixpont jellemzése

Legyen $\mathbf{E}(f\mathbf{U}(Z \wedge g_k)) : \mathcal{P}(S) \to \mathcal{P}(S)$ a következő:
$\mathbf{E}(f\mathbf{U}(Z \wedge g_k)) = \{s \in S \mid$ van olyan $s = s_0, s_1, \ldots, s_k = s'$ szakasz, melyre $s' \in Z, s' \models g_k$ és minden $0 \leq j < k$-ra $s_j \models f\}$.
Tétel. $[\![\mathbf{EG}f]\!]_F = \nu Z.f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z \wedge g_k))))$
Bizonyítás.

a) $[\![\mathbf{EG}f]\!]_F$ az $f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z \wedge g_k))))$ leképezés fixpontja.
$s \in [\![\mathbf{EG}f]\!]_F$

$\iff$ van $s = s_0, s_1, \ldots, s_k = s'$ szakasz, melyen minden g_i teljesül valamely pontban, melynek minden pontjában teljesül f és melyre $s' \in [\![\mathbf{EG}f]\!]_F$.

$\iff$ $s \in f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}([\![\mathbf{EG}f]\!]_F \wedge g_k))))$

Korrekt szimbolikus CTL modell ellenőrzés

b) $[\![\mathbf{EG}f]\!]_F$ tartalmazza az $f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z \wedge g_k))))$ leképezés bármely fixpontját.
Legyen Z_0 egy tetszőleges fixpont. Akkor

$s \in Z_0$

$\iff$ $s \in f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z_0 \wedge g_k))))$

$\Rightarrow$ van $s = s_0, s_1, \ldots, s_k = s'$ szakasz, melyen minden g_i teljesül valamely pontban, melynek minden pontjában teljesül f és melyre $s' \in Z_0$

$\Rightarrow$ ugyanezt folytatva s'-vel, kapjuk, hogy van olyan s-ből kiinduló út, melyen minden g_i végtelen sokszor teljesül, és melynek minden pontján teljesül f.

$\Rightarrow$ $s \in [\![\mathbf{EG}f]\!]_F$.

Korrekt szimbolikus CTL modell ellenőrzés

$[\![\mathbf{EG}f]\!]_F$ kiszámítása:

Nyilvánvaló, hogy $f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z \wedge g_k))))$ monoton. Mivel S véges, ezért

$$\nu Z.f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z_0 \wedge g_k)))) = \bigcap_{i=1}^{\infty} f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(S \wedge g_k))))^i$$

A képletben szereplő CTL formulákat és S-et OBDD-kkel reprezentálva a fixpont iterációs módszerrel meghatározható. Mivel az $[\![\mathbf{E}(f_1\mathbf{U}f_2)]\!]$ alakú részformula maga is fixpont (lásd korábban), ezért minden egyes iterációs lépésben kiszámolunk egy másik fixpontot.

Korrekt szimbolikus CTL modell ellenőrzés

2. $[\![\mathbf{EX}f]\!]_F$ kiszámítása:
   Ismét bevezetjük a fair predikátumot, ahol
   $[\![fair]\!]_F = [\![\mathbf{EG}\mathit{True}\,]\!]_F$ Ezek után $[\![\mathbf{EX}f]\!]_F = [\![\mathbf{EX}(f \wedge fair)]\!]$,
   ahol $[\![\mathbf{EX}(f \wedge fair)]\!]$ halmazt a már megismert módon számítjuk ki szimbolikus modellvizsgálattal.
3. $[\![\mathbf{E}(f\mathbf{U}g)]\!]_F$ kiszámítása:
   $[\![\mathbf{E}(f\mathbf{U}g)]\!]_F = [\![\mathbf{E}(f\mathbf{U}(g \wedge fair))]\!]$,
   ahol $[\![\mathbf{E}(f\mathbf{U}(g \wedge fair))]\!]$ halmazt az ismert módon számítjuk ki szimbolikus modellvizsgálattal.

μ-kalkulus

A fixpontok hatékonyan használhatók modellvizsgálatra, mivel egy formulát kielégítő állapotok halmaza gyakran előáll úgy, mint egy $\mathcal{P}(S)$ feletti monoton leképezés valamilyen fixpontja. Például, mint láttuk $[\![\mathbf{EG}f]\!]_F = \nu Z.\, f \wedge (\bigwedge_{k=1}^{n} \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z \wedge g_k)))$, ahol $F = \{g_1, \ldots, g_n\}$ a korrektségi korlátozások halmaza.

Ugyanakkor a fixpontképzés nem írható le a CTL nyelvén. Ezért célszerű egy olyan nyelvet kifejleszteni, amiben leírható a fixpontképzés. Egy ilyen nyelv a μ-*kalkulus*.

μ-kalkulus

Kissé módosítjuk a Kripke struktúra fogalmát, az átmeneteket nevekkel látjuk el. AP feletti Kripke struktúrán egy $M = (S, T, L)$ hármast értünk, ahol

- S az állapotok halmaza
- $T \subseteq \mathcal{P}(S \times S)$ az átmenetek halmaza, tehát minden $a \in T$-re $a \subseteq S \times S$
- $L : S \to \mathcal{P}(AP)$

A további fogalmak, összefüggések az $M = (S, T, L)$ Kripke struktúrára vonatkoznak, $(s, t) \in a$ helyett $s \xrightarrow{a} t$-t írunk.

μ-kalkulus

Legyen $Var = \{Q_1, Q_2, \ldots\}$ a *részhalmazváltozó*k halmaza (értékeiket $\mathcal{P}(S)$-ből veszik fel).

Szintaxis

- Ha $p \in AP$ akkor p formula.
- Minden részhalmazváltozó formula.
- Ha f és g formulák, akkor $\neg g, f \wedge g, f \vee g$ is formulák.
- Ha f formula és $a \in T$ akkor $[a]f$ és $\langle a \rangle f$ formula.
- Ha $Q \in Var, f$ pedig Q-ban szintaktikusan monoton formula, akkor $\mu Q.f$ és $\nu Q.f$ is formulák. f szintaktikusan monoton Q-ban, ha Q minden f-beli előfordulása páros számú $(0, 2, 4, \ldots)$ negáció hatáskörébe esik.

μ-kalkulus

Szabad és kötött változók. Ha az f-ben szereplő szabad változók $Q_1, \ldots, Q_n$, akkor $f(Q_1, \ldots, Q_n)$-et írunk.

Tetszőleges $e : Var \to \mathcal{P}(S)$ kiértékelés, $Q \in Var,\ W \in P(S)$ esetén $e[Q \leftarrow W]$ kiértékelést a következőképpen definiáljuk:

$$e[Q \leftarrow W](Q') = \begin{cases} W & \text{ha } Q' = Q \\ e(Q') & \text{különben.} \end{cases}$$

Tetszőleges f, μ-kalkulusbeli formula és e kiértékelés esetén $[\![f]\!]_M e$-vel (vagy röviden csak $[\![f]\!]e$-vel) jelöljük azon S-beli állapotok halmazát, amelyek kielégítik f-et.

μ-kalkulus

Szemantika ($[\![f]\!]e$ definíciója)

- Ha $p \in AP$ akkor $[\![p]\!]e = \{s \mid p \in L(s)\}$
- Ha $Q \in Var$ akkor $[\![Q]\!]e = e(Q)$
- $[\![\neg f]\!]e = S - [\![f]\!]e$
- $[\![f \wedge g]\!]e = [\![f]\!]e \cap [\![g]\!]e$
- $[\![f \vee g]\!]e = [\![f]\!]e \cup [\![g]\!]e$
- $[\![\langle a \rangle f]\!]e = \{s \mid (\exists t)(s \xrightarrow{a} t \wedge t \in [\![f]\!]e)\}$
- $[\![[a] f]\!]e = \{s \mid (\forall t)(s \xrightarrow{a} t \Rightarrow t \in [\![f]\!]e)\}$
- $[\![\mu Q.f]\!]e$ a $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ leképezés legkisebb fixpontja, ahol $\tau(W) = [\![f]\!](e[Q \leftarrow W])$
- $[\![\nu Q.f]\!]e$ a $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ leképezés legnagyobb fixpontja, ahol $\tau(W) = [\![f]\!](e[Q \leftarrow W])$

μ-kalkulus

A legkisebb és legnagyobb fixpont létezését az biztosítja, hogy a τ leképezés monoton lesz. Valóban, ha $e \subseteq e'$ akkor

$$[\![f \wedge g]\!]e \subseteq [\![f \wedge g]\!]e'$$

$$[\![f \vee g]\!]e \subseteq [\![f \vee g]\!]e'$$

$$[\![\langle a \rangle f]\!]e \subseteq [\![\langle a \rangle f]\!]e'$$

$$[\![[a] f]\!]e \subseteq [\![[a] f]\!]e'$$

Továbbá a formulákban a negációk bevihetők a változókig a De Morgan szabályokkal és a következő szabályok alkalmazásával:

$$[\![\neg[a] f]\!]e = [\![\langle a \rangle \neg f]\!]e$$

$$[\![\neg\langle a \rangle f]\!]e = [\![[a] \neg f]\!]e$$

$$[\![\neg\mu Q.f(Q)]\!]e = [\![\nu Q.\neg f(Q)]\!]e$$

$$[\![\neg\nu Q.f(Q)]\!]e = [\![\mu Q.\neg f(Q)]\!]e$$

Negációmentes formulákat kapunk.

μ-kalkulus

Tehát minden fixpont operátor paraméterében levő formula monoton lesz, így minden lehetséges τ leképezés monoton lesz.

Mivel S véges, ezért a τ leképezések $\cup$- és $\cap$-folytonosak is lesznek, tehát

$$[\![\mu Q.f]\!]e = \bigcup_i \tau^i(\emptyset)$$

$$[\![\nu Q.f]\!]e = \bigcap_i \tau^i(S)$$

ahol $\tau(W) = [\![f]\!](e[Q \leftarrow W])$.

μ-kalkulus

A CTL beágyazása a μ-kalkulusba: minden f CTL formulához megadható egy vele szemantikailag ekvivalens $Tr(f)$ μ-kalkulusbeli formula.

$$\text{CTL} \xrightarrow{Tr} \mu\text{-kalkulus}$$

- $Tr(p) = p$
- $Tr(\neg f) = \neg Tr(f)$
- $Tr(f \wedge g) = Tr(f) \wedge Tr(g)$
- $Tr(\mathbf{EX}f) = \langle a \rangle Tr(f)$
- $Tr(\mathbf{EG}f) = \nu Z.(Tr(f) \wedge \langle a \rangle Z)$
- $Tr(\mathbf{E}(f\mathbf{U}g)) = \mu Z.(Tr(g) \vee (Tr(f) \wedge \langle a \rangle Z)$

Tétel. Tetszőleges $M = (S, R, L)$ Kripke struktúra és $s \in S$ esetén $M, s \models f \iff s \in [\![Tr(f)]\!]_M$ ahol $a = R$.

μ-kalkulus

Példák.

1) $Tr(\mathbf{EG}(\mathbf{E}(p\mathbf{U}q)) = \nu Y.(\mu Z.(q \vee (p \wedge \langle a \rangle Z)) \wedge \langle a \rangle Y)$
2) Korrekt modellvizsgálat az $F = \{h\}$ korrektségi feltétel mellett.

$$[\![\mathbf{EG}f]\!]_F = \nu Z.f \wedge \mathbf{EX}(\mathbf{E}(f\mathbf{U}(Z \wedge h)))$$

$$[\![\mathbf{E}(f\mathbf{U}(Z \wedge h))]\!] = \mu Y.((Z \wedge h) \vee (f \wedge \mathbf{EX}Y))$$

Tehát $[\![\mathbf{EG}f]\!]_F =$

$$\nu Z.(f \wedge \mathbf{EX}(\mu Y.((Z \wedge h) \vee (f \wedge \mathbf{EX}Y))) =$$

$$\nu Z.(f \wedge \langle a \rangle(\mu Y.((Z \wedge h) \vee (f \wedge \langle a \rangle Y)))$$

μ-kalkulus

Fixpontok kiszámítása

A naív algoritmus.

function $eval(f, e)$
  case
    $f = p$: **return** $\{s \mid p \in L(s)\}$;
    $f = Q$: **return** $e(Q)$;
    $f = \neg g$: **return** $S - eval(g, e)$;
    $f = g \wedge h$: **return** $eval(g, e) \cap eval(h, e)$;
    $f = \langle a \rangle g$: **return** $\{s \mid (\exists t)(s \xrightarrow{a} t \wedge t \in eval(g, e))\}$;
    $f = [a]g$: **return** $\{s \mid (\forall t)(s \xrightarrow{a} t \Rightarrow t \in eval(g, e))\}$;

μ-kalkulus

Fixpontok kiszámítása

$f = \mu Q.g(Q)$:
  $Q_{val} = \emptyset$;
  repeat
    $Q_{old} = Q_{val}$;
    $Q_{val} = eval(g, e[Q \leftarrow Q_{val}])$;
  until $Q_{val} = Q_{old}$;
  return Q_{val};

$f = \nu Q.g(Q)$:
  $Q_{val} = S$;
  repeat
    $Q_{old} = Q_{val}$;
    $Q_{val} = eval(g, e[Q \leftarrow Q_{val}])$;
  until $Q_{val} = Q_{old}$;
  return Q_{val};

endcase endfunction

μ-kalkulus

Fixpontok kiszámítása

A naív algoritmus időbonyolultsága.

- f a formula és $M = (S, T, L)$ a Kripke struktúra

- egy fixpont kiszámításához legfeljebb n lépés szükséges, ahol $n = |S|$

- a legbelső fixpontot kiszámító program n^k-szor fut le, ahol k a formula fixpont mélysége: $f = \mu Q_1.g_1(\ldots \nu Q_2.g_2(\ldots \mu Q_k.g_k(\ldots)\ldots))$

- minden egyes iterációs lépés időbonyolultsága $\mathcal{O}(|M| \cdot |f|)$, ahol $|M| = |S| + \sum_{a \in T} |a|$.

- tehát a teljes algoritmus időbonyolultsága $\mathcal{O}(|M| \cdot |f| \cdot n^k)$.

μ-kalkulus

Fixpontok kiszámítása

A naív algoritmus gyorsítható.

Ehhez zzükségünk lesz arra a tényre, hogy egy legkisebb (legnagyobb) fixpont kiszámítása esetében az iterációt bármely olyan halmazzal kezdhetjük, amely része a legkisebb fixpontnak (tartalmazza a legnagyobb fixpontot).

Lemma. Legyen $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$, $\cup$-folytonos leképezés. Ha $W \subseteq \mu Z.\tau(Z)$ akkor $\bigcup_i \tau^i(W) = \mu Z.\tau(Z)$.

Bizonyítás

$W \subseteq \mu Z.\tau(Z) \Rightarrow \forall i : \tau^i(W) \subseteq \tau^i(\mu Z.\tau(Z)) = \mu Z.\tau(Z)$
$\Rightarrow \bigcup_i \tau^i(W) \subseteq \mu Z.\tau(Z)$

$\emptyset \subseteq W \Rightarrow \forall i : \tau^i(\emptyset) \subseteq \tau^i(W) \Rightarrow \bigcup_i \tau^i(\emptyset) \subseteq \bigcup_i \tau^i(W) \Rightarrow \mu Z.\tau(Z) \subseteq \bigcup_i \tau^i(W)$

μ-kalkulus

Fixpontok kiszámítása

Egy formulában szereplő fixpontok kiszámítása gyorsítható abban az esetben, amikor egy fixpont operátor közvetlen hatáskörében egy ugyanolyan típusú fixpont szerepel.

Tekinstük például a következő formulát: $f = \mu Q_1.g_1(Q_1, \mu Q_2.g_2(Q_1, Q_2))$.

Tegyük fel, hogy egy formulában a $Q_1, \ldots, Q_k$ részhalmazváltozók szerepelnek (Q_1 a legkülső, Q_k a legbelső). Akkor $Q_j^{i_1 \ldots i_j}$-vel jelöljük a Q_j ij-edik *approximáció*ját, miután kiszámoltuk a Q_l i_l-edik approximációját minden $1 \leq l < j$-re. $i_j = \omega$ jelöli a Q_j-re vonatkozó fixpontot.

Például Q_2^{30} a Q_2 fixpont nulladik approximációja ($\emptyset$ vagy S) a Q_1 harmadik approximációjának ismeretében, a Q_2^{31} a Q_2 fixpont első approximációja a Q_1 harmadik approximációjának ismeretében.

μ-kalkulus

Fixpontok kiszámítása

Példa. $f = \mu Q_1.g_1(Q_1, \mu Q_2.g_2(Q_1, Q_2))$. Legyen $\tau(Q_1) = \mu Q_2.g_2(Q_1, Q_2)$ (monoton)

$Q_1^0 = \emptyset$

Q_1^1-hez meg kell határozni: $\tau(Q_1^0) : Q_2^{00} = \emptyset, Q_2^{01}, \ldots, \underline{Q_2^{0\omega}}$ $Q_1^1 = g_1(Q_1^0, Q_2^{0\omega})$

Q_1^2-höz meg kell határozni: $\tau(Q_1^1) : Q_2^{10}, Q_2^{11}, \ldots, Q_2^{1\omega}$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \| \quad$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad ?$

$Q_1^0 \subseteq Q_1^1 \Rightarrow \tau(Q_1^0) = Q_2^{0\omega} \subseteq \tau(Q_1^1) = Q_2^{1\omega}$

Tehát $? = Q_2^{0\omega}$, vagyis a $\tau(Q_1^i) = Q_2^{i\omega}$ fixpont kiszámításakor a $Q_2^{i0} = \emptyset$ helyett elegendő a $Q_2^{i0} = Q_2^{(i-1)\omega}$ kezdőértékkel indulni.

Ezért $[\![f]\!]e$ kiszámítása $\mathcal{O}(n^2)$ lépés helyett $\mathcal{O}(n)$ lépésben elvégezhető, ahol $n = |S|$.

μ-kalkulus

Fixpontok kiszámítása

A gyorsabb algoritmus.

Definíció. Egy f formula *legkülső ν-részformulája (μ-részformulája)* f azon $\nu Q.g$ akakú ($\mu Q.g$ akakú) részformulája, amely semmilyen $\nu Q'.g'$ ($\mu Q'.g'$) alakú részformulának sem részformulája.

μ-kalkulus

Fixpontok kiszámítása

A gyorsabb algoritmus.

A fixpontok megelőző közelítését egy $A[1..N]$ vektorban tároljuk, ahol N a fixpontok száma. Kezdetben $A[i] = \emptyset$, ha Q_i legkisebb és $A[i] = S$, ha Q_i legnagyobb fixpont.

function $eval(f, e)$

case

$f = p$: **return** $\{s \mid p \in L(s)\}$;

$f = Q$: **return** $e(Q)$;

$f = \neg g$: **return** $S - eval(g, e)$;

$f = g \wedge h$: **return** $eval(g, e) \cap eval(h, e)$;

$f = \langle a \rangle g$: **return** $\{s \mid (\exists t)(s \xrightarrow{a} t \wedge t \in eval(g, e))\}$;

$f = [a]g$: **return** $\{s \mid (\forall t)(s \xrightarrow{a} t \Rightarrow t \in eval(g, e))\}$;

μ-kalkulus

Fixpontok kiszámítása

$f = \mu Q_i.g(Q_i)$:

forall $\nu Q_j.g'(Q_j)$ legkülső ν-részformulája g-nek

do $A[j] = S$; **endforall**

repeat

$Q_{old} = A[i]$;

$A[i] = eval(g, e[Q_i \leftarrow A[i]])$;

until $A[i] = Q_{old}$;

return $A[i]$;

$f = \nu Q_i.g(Q_i)$:

forall $\mu Q_j.g'(Q_j)$ legkülső μ-részformulája g-nek

do $A[j] = \emptyset$; **endforall**

repeat

$Q_{old} = A[i]$;

$A[i] = eval(g, e[Q_i \leftarrow A[i]])$;

until $A[i] = Q_{old}$;

return $A[i]$;

endcase endfunction

μ-kalkulus

Fixpontok kiszámítása

Általában $[\![f]\!]e$ kiszámításának időbonyolultsága a gyorsabb módszerrel $\mathcal{O}((|f|n)^d)$, ahol d az f formula alternáló mélysége.

Definíció.

- Minden AP-beli vagy Q-beli változó alternáló mélysége 0.
- $\neg f,\ \langle a \rangle f,\ [a]f$ alternáló mélysége ugyanaz, mint f alternáló mélysége.
- $f \vee g,\ f \wedge g$ alternáló mélysége f és g alternáló mélységének maximuma.
- $\mu Q.f$ alternáló mélysége $\max\{a, b, c\}$, ahol

  $a = 1$

  $b = f$ alternáló mélysége

  $c = 1 +$ (f $\nu Q'.g$ alakú részformulái alternáló mélységének maximuma).
- $\nu Q.f$ alternáló mélysége analóg.

μ-kalkulus

Szimbolikus modell ellenőrzés μ-kalkulussal

Legyen $M = (S, T, L)$ egy Kripke struktúra ($T \subseteq \mathcal{P}(S \times S)$). Akkor S elemei, részhalmazai, T elemei az ismert módon reprezentálhatók $OBDD$-kkel. Legyen tetszőleges

- $p \in AP$ esetén $D_p(x_1, \ldots, x_n)$ az $\{s \in S \mid p \in L(S)\}$ halmazt reprezentáló $OBDD$.
- $a \in T$ esetén $D_a = D_a(x_1, \ldots, x_n, x'_1, \ldots, x'_n)$ az a-t reprezentáló $OBDD$.
- f μ-kalkulus-formula és $e : \ Var \to P(S)$ leképezés esetén $D(f, e)$ az $[\![f]\!]e$ halmazt reprezentáló $OBDD$.

A cél $D(f, e)$ meghatározása.

μ-kalkulus

Szimbolikus modell ellenőrzés μ-kalkulussal

- $D(p, e) = D_p$
- $D(Q, e) = D_{e(Q)}$
- $D(\neg f, e) = \neg D(f, e)$
- $D(f \vee g, e) = D(f, e) \vee D(g, e)$
- $D(\langle a \rangle f, e) = \exists \overline{x'}(D_a(\overline{x}, \overline{x'}) \wedge D(f, e)(\overline{x'}))$
- $D([a]f, e) = D(\neg\langle a \rangle \neg f, e)$
- $D(\mu Q.f, e) = FIX(f, e, D_\emptyset)$
- $D(\nu Q.f, e) = FIX(f, e, D_1)$

ahol $D_\emptyset$ (D_1) az $\emptyset$-t (S-et) reprezentáló $OBDD$, továbbá alkalmaztuk a $D_1 \vee D_2 = apply(D_1, D_2, \vee)$, stb. rövidítéseket. FIX a következő eljárás:

μ-kalkulus

Szimbolikus modell ellenőrzés μ-kalkulussal

function $FIX(f, e, D_Q)$
  result-bdd $= D_Q$;
  repeat
    old-bdd $=$ result-bdd;
    result-bdd $= D(f, e(Q \leftarrow$ old-bdd$))$;
  until (old-bdd $=$ result-bdd)
  return (result-bdd);
endfunction

Automataelméleti megközelítés

Véges automaták végtelen szavakon

Definíciók

1) Legyen Σ egy véges ábécé. Akkor Σ^ω a Σ-ból képezhető végtelen (ω-hosszúságú) szavak halmaza.

   $$v \in \Sigma^\omega \iff v = a_0 a_1 \ldots, \text{ ahol } \forall i : a_i \in \Sigma$$

2) Véges (Büchi) automata $B = (Q, \Sigma, \delta, I, F)$, ahol
   - Q egy véges halmaz, az *állapotok* halmaza;
   - Σ az *input* ábécé;
   - $\delta \subseteq Q \times \Sigma \times Q$ az *átmenetek* halmaza;
   - $I \subseteq Q$ a *kezdőállapotok* halmaza;
   - $F \subseteq Q$ a *végállapotok* halmaza;

Automataelméleti megközelítés

3) Legyen $v = a_0 a_1 \ldots \in \Sigma^\omega$. B egy *futása* v-n egy olyan $\rho : \{0, 1, \ldots\} \to Q$ leképezés, melyre teljesül, hogy $\rho(0) \in I$ és minden $i \geq 0$-ra $(\rho(i), a_i, \rho(i+1)) \in \delta$.

$$\begin{array}{ccccccc} q_0 & \xrightarrow{a_0} & q_1 & \xrightarrow{a_1} & q_2 & \xrightarrow{a_2} & \ldots \\ \| & & \| & & \| & & \\ \rho(0) & & \rho(1) & & \rho(2) & & \end{array}$$

$inf(\rho) = \{q \in Q \mid \text{végtelen sok } i\text{-re } \rho(i) = q\}$.

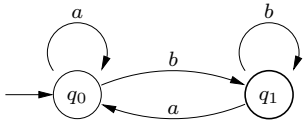
ρ *elfogadó*, ha $inf(\rho) \cap F \neq \emptyset$.

4) $v \in L(B) \iff B$-nek van elfogadó futása v-n.

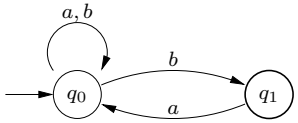
Automataelméleti megközelítés

Példák Büchi automatákkal felismerhető nyelvekre:

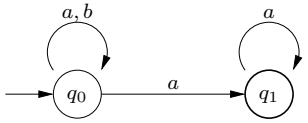
- végtelen sok b



- végtelen sok ba alakú rész-szó (végtelen sok ab alakú rész-szó)



- véges sok b

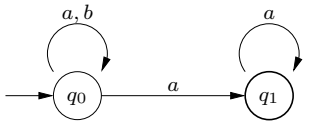


Automataelméleti megközelítés

Az $M = (Q, \Sigma, \delta, I, F)$ Büchi automata *determinisztikus*, ha $\delta : Q \times \Sigma \to Q$ egy leképezés.

Tétel. A determinisztikus Büchi automatákkal felismerhető nyelvek osztálya valódi része a nemdeterminisztikus Büchi automatákkal felismerhető nyelvek osztályának.

Bizonyítás.



A fenti $\Sigma = \{a, b\}$ feletti automata nemdeterminisztikus és pontosan azokat a $v \in \Sigma^\omega$ szavakat ismeri fel, ahol v-ben véges sok b szerepel.

Automataelméleti megközelítés

Belátjuk, hogy ez a nyelv nem ismerhető fel determinisztikus Büchi automatával.

Tegyük fel, hogy van olyan B' determinisztikus Büchi automata, amely ezt a nyelvet ismeri fel.

Akkor vannak olyan n_1, n_2, $n_3, \ldots$ számok, hogy az

$$a^{n_1},\ a^{n_1} b a^{n_2},\ a^{n_1} b a^{n_2} b a^{n_3}, \ldots$$

szavak a B' automatát végállapotba viszik, különben az

$$a^\omega,\ a^{n_1} b a^\omega,\ a^{n_1} b a^{n_2} b a^\omega, \ldots$$

szavakat nem ismerné fel. De akkor B' felismeri az

$$a^{n_1} b a^{n_2} b a^{n_3} \ldots$$

szót is, ami ellentmondás.

Automataelméleti megközelítés

Következmény. A determinisztikus Büchi automatákkal felismerhető nyelvek osztálya nem zárt a komplementer képzésre.

Bizonyítás. A $\Sigma = \{a, b\}$ feletti

- végtelen b-t tartalmazó ω-szavak halmaza felismerhető determinisztikus Büchi automatával (ld. példa)
- véges számú b-t tartalmazó ω-szavak halmaza nem ismerhető fel determinisztikus Büchi automatával (ld. az előző tételt).

Automataelméleti megközelítés

Tétel. A Büchi automatákkal felismerhető nyelvek osztálya zárt a komplementer képzésre.

Bizonyítás. Nagyon nehéz, $2^{\mathcal{O}(n \log n)}$ állapotra van szükség. Lásd

A. P. Sistla, M, Y. Vardi, P. Wolper, The complementation problem for Büchi automata with applications to temporal logic, *Theoretical Computer Science* 49: 161-177.

Automataelméleti megközelítés

Tétel. A Büchi automatákkal felismerhető nyelvek osztálya zárt a metszetre.

Bizonyítás. Legyenek

$$B_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$$
$$B_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$$

tetszőleges Büchi automaták.

Legyen $B = (Q_1 \times Q_2 \times \{0, 1, 2\}, \Sigma, \delta, I_1 \times I_2 \times \{0\}, F)$ az a Büchi automata, ahol

- $((p_1, p_2, i_1), a, (q_1, q_2, i_2)) \in \delta \iff$
  - $(p_1, a, q_1) \in \delta_1$ és $(p_2, a, q_2) \in \delta_2$
  - ha $i_1 = 0$ és $q_1 \in F_1$, akkor $i_2 = 1$
  - ha $i_1 = 1$ és $q_2 \in F_2$, akkor $i_2 = 2$
  - ha $i_1 = 2$, akkor $i_2 = 0$
  - $i_1 = i_2$ különben
- $F = Q_1 \times Q_2 \times \{2\}$

Automataelméleti megközelítés

Megjegyzés. $F = F_1 \times F_2$ nem jó: az F_1 és F_2-beli állapotok külön-külön előfordulhatnak végtelen sokszor úgy, hogy együtt csak véges sokszor (esetleg 0-szor) fordulnak elő.

$$\begin{pmatrix} \underline{p} \\ r \end{pmatrix} \begin{pmatrix} s \\ \underline{q} \end{pmatrix} \begin{pmatrix} \underline{p} \\ r \end{pmatrix} \begin{pmatrix} s \\ \underline{q} \end{pmatrix} \dots \qquad \begin{matrix} \underline{p} \in F_1 \\ \underline{q} \in F_2 \end{matrix}$$

Megjegyzés. Ha $F_1 = Q_1$, akkor az egyszerűbb

$$B = (Q_1 \times Q_2, \Sigma, \delta, I_1 \times I_2, Q_1 \times F_2)$$

automata is megfelelő, ahol

$$((p_1, p_2), a, (q_1, q_2)) \in \delta \iff (p_1, a, q_1) \in \delta_1 \wedge (p_2, a, q_2) \in \delta_2.$$

Automataelméleti megközelítés

Általánosított Büchi automatának mondjuk azt a

$$B = (Q, \Sigma, \delta, I, F)$$

rendszert, ahol

$F = \{P_1, \ldots, P_k\}$ és $\forall 1 \leq i \leq k$-ra $P_i \subseteq Q$.

Továbbá minden $v \in \Sigma^\omega$ szóra

$v \in L(B) \iff$
B-nek van olyan ρ futása v-n, hogy $\forall\, 1 \leq i \leq k$-ra $inf(\rho) \cap P_i \neq \emptyset$

Tétel. Az általánosított Büchi automatákkal felismerhető nyelvek osztálya megegyezik a Büchi automatákkal felismerhető nyelvek osztályával.

Automataelméleti megközelítés

Bizonyítás. Legyen $B = (Q, \Sigma, \delta, I, F)$, ahol $F = \{P_1, \ldots, P_k\}$.

Konstruáljuk meg a

$$B' = (Q \times \{0, 1, \ldots, k\}, \Sigma, \delta', I \times \{0\}, Q \times \{k\})$$

automatát, ahol

$((p, x), a, (q, y)) \in \delta' \iff$

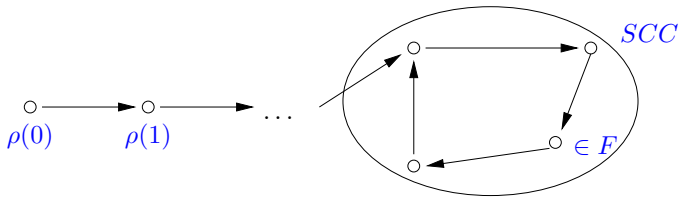
- $(p, a, q) \in \delta$,
- ha $q \in P_i \wedge x = i - 1$, akkor $y = i$
- ha $x = k$, akkor $y = 0$
- különben $x = y$

Ekkor teljesülni fog az $L(B) = L(B')$ egyenlőség.

Automataelméleti megközelítés

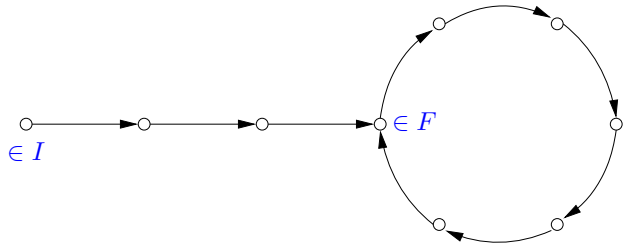
Tétel. Tetszőleges $B = (Q, \Sigma, \delta, I, F)$ Büchi automatára eldönthető, hogy $L(B) = \emptyset$ teljesül-e.

Bizonyítás. Legyen $v \in \Sigma^\omega$ egy végtelen szó és ρ egy futás v-n. Mivel Q véges, van olyan i_0, hogy a $\rho(i_0), \rho(i_0 + 1), \ldots$ állapotok mindegyike végtelen sokszor fordul elő a $\rho(i_0), \rho(i_0 + 1), \ldots$ sorozatban. Tehát a $\rho(i_0), \rho(i_0 + 1), \ldots$ állapotok B egy erősen összefüggő komponensének (SCC) részei. Tehát $L(B) \neq \emptyset \iff B$ elérhető állapotainak van olyan SCC-je, mely tartalmaz végállapotokat.



Automataelméleti megközelítés

Következmény. $L(B) \neq \emptyset \iff B$-nek van olyan elérhető $q \in F$ végállapota, mely egy körön helyezkedik el.



Tétel. B ezen tulajdonsága $\mathcal{O}(n^2)$ lépésben eldönthető, ahol $n = |Q|$.

Bizonyítás. Tegyük fel, hogy egy kezdőállapot van: $I = \{q_0\}$.

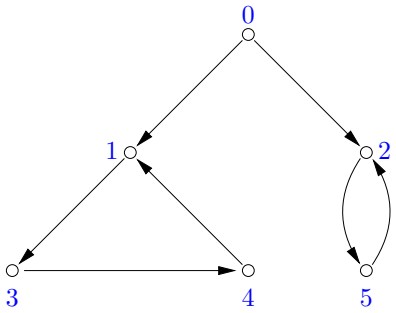
Két lépésben járunk el: I és II.

Automataelméleti megközelítés

I. Soroljuk fel a q_0-ból elérhető végállapotokat postorder sorrendben.

1) A q_0-ból elindulva futtassuk a mélységi keresés (DFS) algoritmust B átmenetgráfjára.

2) A DFS során akkor adjunk az outputra egy elérhető f végállapotot, amikor minden leszármazottját bejártuk (és nem amikor először találkozunk vele).



$4, 3, 1, 5, 2, 0$

Automataelméleti megközelítés

Lemma. Legyenek $q_1, \ldots, q_k$ a q_0-ból elérhető végállapotok postorder sorrendben. Ha van olyan $1 \leq i < j \leq k$, hogy q_i-ből út vezet q_j-be, akkor q_i rajta van egy körön.

Bizonyítás.

- Tegyük fel, hogy q_i-ből vezet út q_j-be.
- Az úton lévő csúcsok közül a DFS nem q_i-t érte el, legelsőnek (különben q_j megelőzné q_i-t a postorder sorrendben).
- Tehát az út valamely p csúcspontját a DFS előbb érte el, mint q_i-t.
- p nem előzheti meg q_i-t a postorderben (különben q_j is megelőzné q_i-t.)
- Tehát p a q_i egyik őse a DFS által meghatározott feszítőfában: p-ből van út q_i-be B átmenetgráfjában.
- Tehát q_i és p egy körön helyezkednek el.

Automataelméleti megközelítés

II. **Algoritmus**

Input: A végállapotok egy $q_1, \ldots, q_k$ postorder felsorolása.

for $i = 1$ **to** k **do**

- DFS q_i-ből
- ha a DFS eléri q_i-t, akkor **return** "kör"
- ha a DFS elér egy, a $q_1, \ldots, q_{i-1}$-ből végrehajtott DFS-ek által megjelölt csúcsot, akkor backtrack a DFS-ben (mert q_i nem lehet körön).
- **return** "nincs kör".

done

Automataelméleti megközelítés

Tétel. A II. algoritmus akkor és csak akkor ad vissza "kör"-t, ha a $q_1, \ldots, q_k$ állapotok valamelyike rajta van egy körön.

Bizonyítás. "$\Rightarrow$" Triviális.

"$\Leftarrow$"

- Legyen j a legkisebb olyan index, melyre q_j rajta van egy π körön.
- Állítás. π egyik csúcsa sem érhető el egy olyan q_i-ből, melyre $i < j$.
  Bizonyítás. Tegyük fel, hogy van ilyen i. Akkor a Lemma szerint van q_i-n átmenő kör. Ellentmondás, mert j minimális volt.
- Tehát a $q_1, \ldots, q_{j-1}$-ből kiinduló DFS-ek egyike sem jelölte meg π egyetlen csúcsát sem.
- Tehát a II. algoritmusban a q_j-ből kiinduló DFS eléri q_j-t és "kör" -t ad vissza.

Automataelméleti megközelítés

Időbonyolultság

- Elérhető végállapotok: $\mathcal{O}(n)$
- A $q_1, \ldots, q_j$ állapotokból kiinduló DFS-ek (együttvéve) minden állapotot legfeljebb egyszer érnek el: $\mathcal{O}(n)$.

Tárigény

- Egy n mélységű verem a postorder részére
- Egy n hosszúságú bitvektor az elérés nyilvántartására

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

A Büchi automaták a következő modellvizsgálati probléma megoldásában használhatók.

Adott egy $M = (S, S_0, R, L)$ Kripke struktúra és egy f, LTL útformula.

Igaz-e, hogy minden $\pi = s_0, s_1, \ldots$ $(s_0 \in S_0)$ számítási útra $\pi \models f$?

A problémát visszavezetjük annak eldöntésére, hogy egy Büchi automata által felismert nyelv üres-e.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

A visszavezetés.

Definíció. Tetszőleges g LTL útformulához és $\sigma \in \mathcal{P}(AP)^\omega$ sorozat (számítás) esetén definiáljuk a $\sigma \models g$ fogalmát (Kripke struktúrától függetlenül).

- $\sigma \models p\,(\in AP) \iff p \in \sigma_0$
- $\sigma \models \neg g \iff \sigma \not\models g$
- $\sigma \models g_1 \wedge g_2 \iff \sigma \models g_1$ és $\sigma \models g_2$
- $\sigma \models \mathbf{X} g_1 \iff \sigma^1 \models g_1$
- $\sigma \models \mathbf{F} g_1 \iff$ van olyan $i \geq 0$, hogy $\sigma^i \models g_1$
- $\sigma \models \mathbf{G} g_1$, $\sigma \models g_1 \,\mathbf{U}\, g_2$ hasonlóan.

Legyen $models(g) = \{\sigma \in \mathcal{P}(AP)^\omega \mid \sigma \models g\}$.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

A visszavezetés.

1) Az $M = (S, S_0, R, L)$ Kripke struktúrát egy olyan B_M Büchi automatává transzformáljuk, melyre

$$L(B_M) = \{L(s_0)L(s_1)\ldots \mid s_0, s_1, \ldots \text{ számítási út és } s_0 \in S_0\}$$

(Tehát $L(B_M) \subseteq \mathcal{P}(AP)^\omega$.)

2) Az f formulát egy olyan B_f Büchi automatává transzformáljuk, melyre $L(B_f) = models(f)$.

$$\begin{aligned} &\forall \pi = s_0, s_1, \ldots (s_0 \in S_0),\, \pi \models f \\ &\iff \quad L(B_M) \subseteq L(B_f) \\ &\iff \quad L(B_M) \cap \overline{L(B_f)} = \emptyset \\ &\iff \quad L(B_M) \cap L(B_{\neg f}) = \emptyset. \end{aligned}$$

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

A visszavezetés.

- A Büchi automatákkal felismerhető nyelvek osztálya effektíven zárt a komplementerképzésre és a metszetre. Akkor megkonstruálható olyan B Büchi automata, amely az $L(B_M) \cap \overline{L(B_f)}$ nyelvet ismeri fel.
- B-ről eldönthető, hogy az $\emptyset$-t ismeri-e fel.
- Időbonyolultság: $\mathcal{O}(|S| \cdot 2^{\mathcal{O}(|f|)})$.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

1) Kripke struktúra átalakítása Büchi automatává.

Legyen $M = (S, R, S_0, L)$ egy kripke struktúra. Konstruáljuk meg a $B_M = (\Sigma, S \cup \{s_0\}, \delta, s_0, S \cup \{s_0\})$ Büchi automatát, ahol

- $\Sigma = \mathcal{P}(AP)$,
- $(s, a, t) \in \delta \iff (s, t) \in R \wedge a = L(t)$, minden $s, t \in S$-re,
- $(s_0, a, s) \in \delta \iff s \in S_0 \wedge a = L(s)$.

Nyilvánvaló, hogy

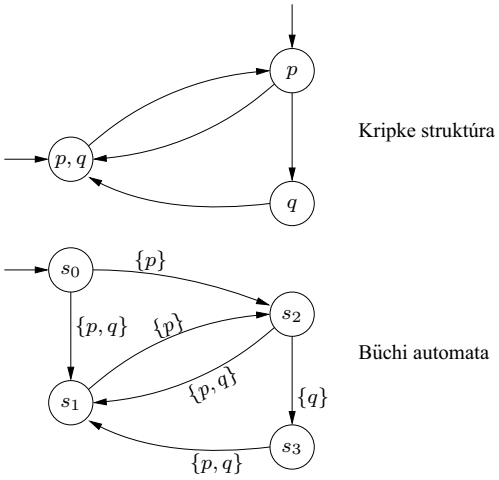
$$L(B_M) = \{L(s_0)L(s_1)\ldots \mid s_0, s_1, \ldots \text{ számítási út, } s_0 \in S_0\}.$$

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

Kripke struktúra átalakítása Büchi automatává.

Példa.



Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

2) Az f LTL útformula átalakítása B_f Büchi automatává.

Definíciók.

- Az f útformula *lezárásán* a
  $$Cl(f) = \{g, \neg g \mid g \text{ az } f \text{ részformulája}\}$$
  formulahalmazt értjük.
- Legyen $\sigma = \sigma_0\sigma_1 \ldots \in \mathcal{P}(AP)^\omega$ egy számítás.
  A σ-hoz és f-hez tartozó *(kielégítési) sorozaton* a
  $$satseq(\sigma, f) = satseq(\sigma, f, 0)satseq(\sigma, f, 1)\ldots$$
  sorozatot értjük, ahol
  $$satseq(\sigma, f, i) = \{g \in Cl(f) \mid \sigma^i \models g\}.$$

Észrevétel: $\sigma^i \models f \iff f \in satseq(\sigma, f, i)$.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatákkal

LTL útformula átalakítása Büchi automatává.

Cél: Olyan Büchi automata megkonstruálása, melynek elfogadó futásai az összes olyan

$$satseq(\sigma, f, 0) \xrightarrow{\sigma_0} satseq(\sigma, f, 1) \xrightarrow{\sigma_1} \dots$$

alakú sorozatok, ahol $f \in satseq(\sigma, f, 0)$.

Probléma: a $satseq(\sigma, f, i)$ állapotok megadása.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával

LTL formula átalakítása Büchi automatává

Definíció. Az f formula egy *pre-Hintikka* sorozata egy olyan $\alpha = \alpha_0\alpha_1 \dots \in \mathcal{P}(Cl(f))^\omega$ sorozat, amely kielégíti a következő feltételeket:

- Ha $f_1 \in Cl(f)$, akkor $\neg f_1 \in \alpha_i \iff f_1 \notin \alpha_i$.
- Ha $f_1 \wedge f_2 \in Cl(f)$, akkor $f_1 \wedge f_2 \in \alpha_i \iff f_1 \in \alpha_i$ és $f_2 \in \alpha_i$.
- Ha $\mathbf{X}f_1 \in Cl(f)$, akkor $\mathbf{X}f_1 \in \alpha_i \iff f_1 \in \alpha_{i+1}$.
- Ha $\mathbf{F}f_1 \in Cl(f)$, akkor $\mathbf{F}f_1 \in \alpha_i \iff f_1 \in \alpha_i$ vagy $\mathbf{F}f_1 \in \alpha_{i+1}$.
- Ha $\mathbf{G}f_1 \in Cl(f)$, akkor $\mathbf{G}f_1 \in \alpha_i \iff f_1 \in \alpha_i$ és $\mathbf{G}f_1 \in \alpha_{i+1}$.
- Ha $f_1\mathbf{U}f_2 \in Cl(f)$, akkor $f_1\mathbf{U}f_2 \in \alpha_i \iff f_2 \in \alpha_i$ vagy ($f_1 \in \alpha_i$ és $f_1\mathbf{U}f_2 \in \alpha_{i+1}$).

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával

LTL formula átalakítása Büchi automatává

Definíció. Az f formula egy pre-Hintikka-sorozata Hintikka-sorozat, ha teljesülnek a következő feltételek:

- Ha $\mathbf{F}f_1 \in \alpha_i$ akkor van olyan $j \geq i$ hogy $f_1 \in \alpha_j$
- Ha $\mathbf{G}f_1 \notin \alpha_i$ (vagy $\neg\mathbf{G}f_1 \in \alpha_i$) akkor van olyan $j \geq i$ hogy $f_1 \notin \alpha_j$ (vagy $\neg f_1 \in \alpha_j$)
- Ha $f_1\mathbf{U}f_2 \in \alpha_i$ akkor van olyan $j \geq i$ hogy $f_2 \in \alpha_j$

Definíció. Egy $\alpha = \alpha_0\alpha_1 \dots$ pre-Hintikka-sorozat illeszkedik egy $\sigma_0\sigma_1 \dots \in \mathcal{P}(AP)^\omega$ számításhoz, ha minden i-re $\sigma_i \subseteq \alpha_i$ és $(\alpha_i - \sigma_i) \cap AP = \emptyset$.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával

LTL formula átalakítása Büchi automatává

Tétel. $satseq(\sigma, f)$ az f formula egyetlen olyan Hintikka-sorozata, mely illeszkedik σ-hoz.

Bizonyítás. A $satseq(\sigma, f)$ definíciójából következik, hogy $satseq(\sigma, f)$ az f Hintikka-sorozata.

Megmutatjuk, hogy ha α az f egy olyan Hintikka-sorozata, mely illeszkedik σ-hoz, akkor $\alpha = satseq(\sigma, f)$: tetszőleges $g \in Cl(f)$-re és $i \geq 0$-ra $g \in \alpha_i \iff g \in satseq(\sigma, f, i)$.

A bizonyítást g szerinti indukcióval végezzük.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával

LTL formula átalakítása Büchi automatává

A bizonyítást g szerinti indukcióval végezzük.

(i) $g = p$ eset: $p \in \alpha_i \iff (\alpha$ illeszkedik σ-hoz$)$
$p \in \sigma_i \iff p \in satseq(\sigma, f, i)$

(ii) $g = g_1 \wedge g_2$ eset: $g_1 \wedge g_2 \in \alpha_i$
$\iff$ (Hintikka-sorozat definíciója) $g_1 \in \alpha_i$ és $g_2 \in \alpha_i$
$\iff$ (indukció feltevés) $g_1 \in satseq(\sigma, f, i)$ és $g_2 \in satseq(\sigma, f, i)$
$\iff$ (kielégítő sorozat definíciója) $g_1 \wedge g_2 \in satseq(\sigma, f, i)$

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával

LTL formula átalakítása Büchi automatává

(iii) $g = g_1 \mathbf{U} g_2$ eset: $g_1 \mathbf{U} g_2 \in \alpha_i$
$\iff$ (Hintikka-sorozat definíció, indukció) $g_2 \in \alpha_{i+k}$ és
$g_1 \in \alpha_i, \ldots, \alpha_{i+k-1}$ valamely $k \geq 0$-ra
$\iff$ (indukció feltevés) $g_2 \in satseq(\sigma_1, f, i+k)$ és
$g_1 \in satseq(\sigma, f, i), \ldots, satseq(\sigma, g, i+k-1)$
$\iff$ (kielégítő sorozat definíciója) $g_1 \mathbf{U} g_2 \in satseq(\sigma, f, i)$

Jelöljük a $satseq(\sigma, f)$ sorozatot $H(\sigma, f) = H(\sigma, f, 0), H(\sigma, f, 1), \ldots$-fel.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával

LTL formula átalakítása Büchi automatává

Cél. (átfogalmazás): Készítsünk olyan Büchi automatát, melynek

1) $\alpha_0 \xrightarrow{\sigma_0} \alpha_1 \xrightarrow{\sigma_1} \ldots$ egy futása $\iff \alpha = \alpha_0\alpha_1 \ldots$ az f-nek egy olyan pre-Hintikka-sorozata, mely illeszkedik σ-hoz.

2) egy $\alpha_0 \xrightarrow{\sigma_0} \alpha_1 \xrightarrow{\sigma_1} \ldots$ futása akkor és csak akkor elfogadó, ha $\alpha = H(\sigma, f)$ és $f \in H(\sigma, f, 0)$.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával

LTL formula átalakítása Büchi automatává

Konstrukció. Az f LTL formulához készítsük el a $B_f = (Q, \Sigma, \delta, I, F)$ általánosított Büchi automatát, ahol

- $\Sigma = \mathcal{P}(AP)$
- Q a $Cl(f)$ azon α részhalmazaiból áll, melyekre teljesülnek:
  - $\neg f_1 \in \alpha \iff f_1 \notin \alpha$
  - ha $f_1 \wedge f_2 \in Cl(f)$ akkor $f_1 \wedge f_2 \in \alpha \iff f_1 \in \alpha$ és $f_2 \in \alpha$
- $I = \{\alpha \mid f \in \alpha\}$
- $\alpha \xrightarrow{a} \beta$ egy átmenet $((\alpha, a, \beta) \in \delta)$ akkor és csak akkor, ha $a = \{p \in AP \mid p \in \alpha\}$ és teljesülnek a következők:

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

- Ha $\mathbf{X}f_1 \in Cl(f)$, akkor $\mathbf{X}f_1 \in \alpha \iff f_1 \in \beta$,
- Ha $\mathbf{F}f_1 \in Cl(f)$, akkor $\mathbf{F}f_1 \in \alpha \iff f_1 \in \alpha$ vagy $\mathbf{F}f_1 \in \beta$
- Ha $\mathbf{G}f_1 \in Cl(f)$, akkor $\mathbf{G}f_1 \in \alpha \iff f_1 \in \alpha$ és $\mathbf{G}f_1 \in \beta$
- Ha $f_1\mathbf{U}f_2 \in Cl(f)$, akkor $f_1\mathbf{U}f_2 \in \alpha \iff f_2 \in \alpha$, vagy ($f_1 \in \alpha$ és $f_1\mathbf{U}f_2 \in \beta$)

- F azon F_g alakú állapothalmazokból áll, ahol g az f egy $\mathbf{F}f_1$,$\neg\mathbf{G}f_1$ vagy $f_1\mathbf{U}f_2$ alakú részformulája, F_g pedig a következőképpen definiált halmaz:

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

- Ha $g = \mathbf{F}f_1$, akkor F_g azon állapotokból áll, melyek tartalmazzák f_1-et vagy $\neg\mathbf{F}f_1$-et
- Ha $g = \neg\mathbf{G}f_1$, akkor F_g azon állapotokból áll, melyek tartalmazzák $\neg f_1$-et vagy $\mathbf{G}f_1$-et
- Ha $g = f_1\mathbf{U}f_2$, akkor F_g azon állapotokból áll, melyek tartalmazzák f_2-t vagy $\neg(f_1\mathbf{U}f_2)$-t.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

Tétel. Legyen $B_f = (Q, \Sigma, \delta, I, F)$, az f-ből megkonstruált Büchi automata.

(1) $\alpha_0 \xrightarrow{\sigma_0} \alpha_1 \xrightarrow{\sigma_1} \dots$ akkor és csak akkor futása B_f-nek, ha α az f egy olyan pre-Hintikka sorozata, amely illeszkedik σ-hoz.

(2) A B_f egy $\alpha_0 \xrightarrow{\sigma_0} \alpha_1 \xrightarrow{\sigma_1} \dots$ futása akkor és csak akkor elfogadó, ha $\alpha = H(\sigma, f)$ és $f \in H(\sigma, f, 0)$.

Bizonyítás. (1) a definícióból következik

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

(2)“⇒” Tegyük fel, hogy $\alpha_0 \xrightarrow{\sigma_0} \alpha_1 \xrightarrow{\sigma_1} \dots$ elfogadó futás. Akkor (1) miatt $\alpha = \alpha_0\alpha_1 \dots$ az f olyan pre-Hintikka sorozata, amely illeszkedik σ-hoz. Tegyük fel, hogy α nem Hintikka sorozat, a következő esetek lehetségesek:

- $(\exists i \geq 0) : \mathbf{F}f_1 \in \alpha_i$ és $(\forall j \geq i)\ \neg f_1 \in \alpha_j$. Mivel α pre-Hintikka, ezért $(\forall j \geq i)\ \mathbf{F}f_1 \in \alpha_j$. Tehát $(\forall j \geq i)$: $\alpha_j \notin F_{\mathbf{F}f_1}$, vagyis a futás nem elfogadó, ami ellentmondás.
- $(\exists i \geq 0) : \neg\mathbf{G}f_1 \in \alpha_i$ és $(\forall j \geq i)$: $f_1 \in \alpha_j$. Továbbá, az is teljesül, hogy $(\forall j \geq i)$: $\neg\mathbf{G}f_1 \in \alpha_j$. (Ellenkező esetben, mivel α pre-Hintikka, azt kapnánk, hogy $\neg\mathbf{G}f_1 \in \alpha_i$.) Tehát $(\forall j \geq i)$: $\alpha_j \notin F_{\neg\mathbf{G}f_1}$, vagyis a futás nem elfogadó, ami ellentmondás.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

- $(\exists i \geq 0) : f_1 \mathbf{U} f_2 \in \alpha_i$ és $(\forall j \geq i)$: $\neg f_2 \in \alpha_j$.
  Ekkor, mivel α pre-Hintikka, minden $j \geq i$ esetén $f_1 \mathbf{U} f_2 \in \alpha_j$. Tehát tetszőleges $j \geq i$ indexre $f_2 \notin \alpha_j$ és $\neg(f_1 \mathbf{U} f_2) \notin \alpha_j$.
  Tehát minden $j \geq i$-re az α_j állapot nincs benne az $F_{f_1 \mathbf{U} f_2}$ végállapothalmazban és így a futás nem elfogadó, ami ellentmondás.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

(2)"⇐" Tegyük fel, hogy $\alpha = H(\sigma, f)$ és $f \in H(\sigma, f, 0)$. Akkor α a B_f egy futása (1) és az $f \in H(\sigma, f, 0)$ feltétel miatt.
Tegyük fel, hogy α nem elfogadó futás. Legyen F_g egy elfogadó állapothalmaz (g az f egy bizonyos részformulája) és tegyük fel, hogy van olyan $j \geq 0$, hogy $(\forall i \geq j) : \alpha_i \notin F_g$.

- $g = \mathbf{F} f_1$ eset. Ekkor
  - $(\forall i \geq j) : \neg f_1, \mathbf{F} f_1 \in \alpha_i$ ($F_{\mathbf{F} f_1}$ definíciója),
  - $(\forall i \geq j) : \neg f_1, \mathbf{F} f_1 \in satseq(\sigma, f, i)$ (mivel $H(\sigma, f) = satseq(\sigma, f)$),
  - $\neg \mathbf{F} f_1$, $\mathbf{F} f_1 \in satseq(\sigma, f, j)$ (satseq definíciója).
    Ellentmondás.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

- $g = \neg \mathbf{G} f_1$ eset. Ekkor
  - $(\forall i \geq j) : f_1, \neg \mathbf{G} f_1 \in \alpha_i$ ($F_{\neg \mathbf{G} f_1}$ definíciója),
  - $(\forall i \geq j) : \mathbf{G} f_1 \in satseq(\sigma, f, i)$ (mivel $H(\sigma, f) = satseq(\sigma, f)$),
  - $\neg \mathbf{G} f_1$, $\mathbf{G} f_1 \in satseq(\sigma, f, j)$ (satseq definíciója).
    Ellentmondás.
- $g = f_1 \mathbf{U} f_2$ eset. Ekkor
  - $(\forall i \geq j) : \neg f_2$, $(f_1 \mathbf{U} f_2) \in \alpha_i$ ($F_{f_1 \mathbf{U} f_2}$ definíciója),
  - $(\exists i \geq j) : f_2 \in \alpha_i$ (mivel $H(\sigma, f) = satseq(\sigma, f)$, továbbá satseq definíciója),
  - $(\exists i \geq j) : f_2, \neg f_2 \in \alpha_i$.
    Ellentmondás.

Automataelméleti megközelítés

LTL modell ellenőrzés Büchi automatával
LTL formula átalakítása Büchi automatává

Példa. $f = \mathbf{G}(p \Rightarrow \mathbf{F} q)$

- $Cl(f) = \{\mathbf{G}(p \Rightarrow \mathbf{F} q), p \Rightarrow \mathbf{F} q, p, \mathbf{F} q, q\}$
  $\cup$ ugyanezek negáltjai
- Néhány állapot
  $\{\mathbf{G}(p \Rightarrow \mathbf{F} q), p \Rightarrow \mathbf{F} q, \neg p, \mathbf{F} q, q\}$
  $\{\neg \mathbf{G}(p \Rightarrow \mathbf{F} q), p \Rightarrow \mathbf{F} q, p, \mathbf{F} q, \neg q\}$
  $\{\neg \mathbf{G}(p \Rightarrow \mathbf{F} q), \neg(p \Rightarrow \mathbf{F} q), p, \neg \mathbf{F} q, \neg q\}$
- Példa átmenetre:
  $\{\mathbf{G}(p \Rightarrow \mathbf{F} q), p \Rightarrow \mathbf{F} q, \neg p, \mathbf{F} q, q\} \xrightarrow{\{q\}} \{\mathbf{G}(p \Rightarrow \mathbf{F} q), p \Rightarrow \mathbf{F} q, p, \mathbf{F} q, q\}$

Vége...