IV113 Validace a verifikace

Ověřování modelu pro CTL a logiky větvícího se času

Jiří Barnat

Lineární x Větvící se čas

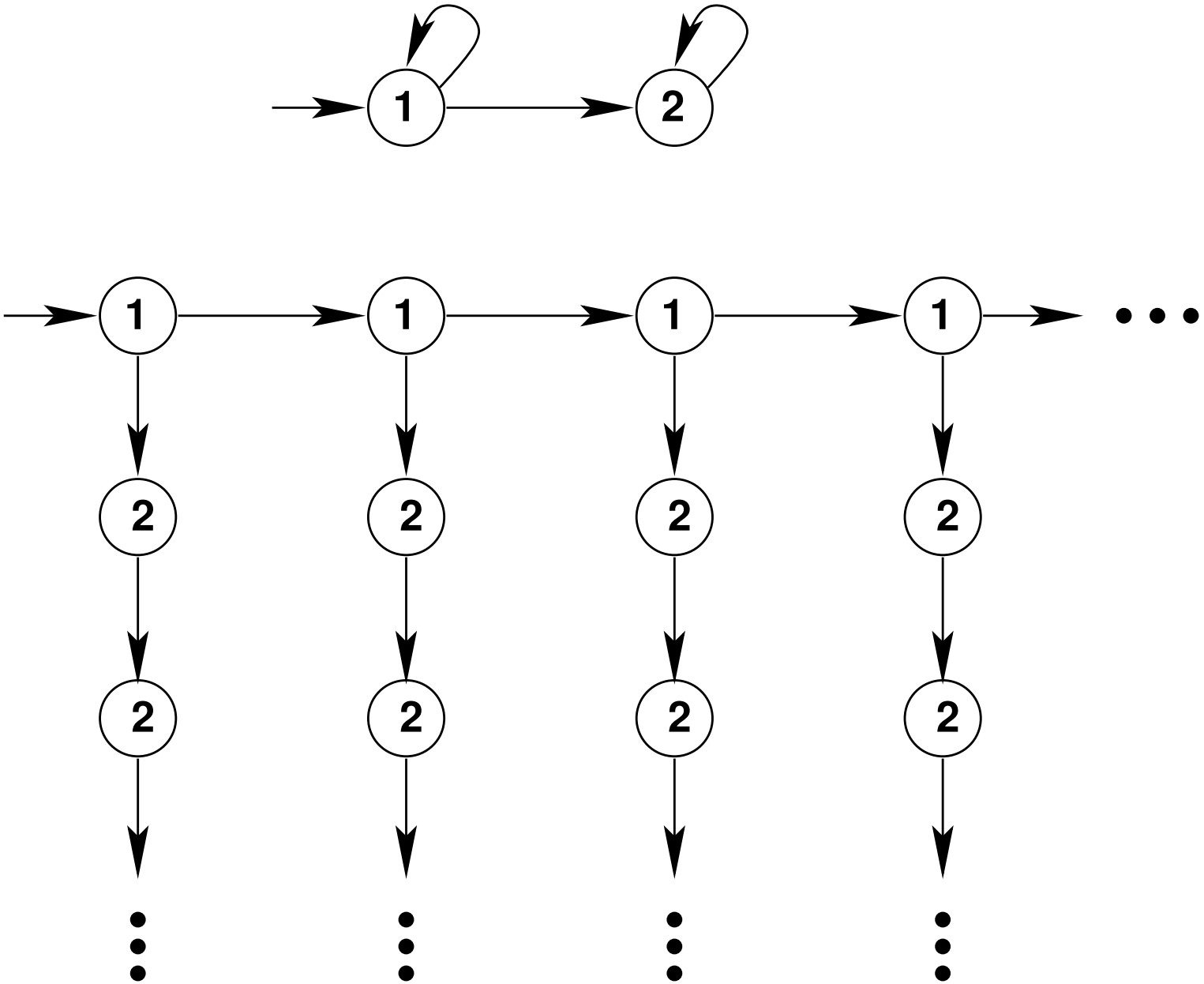
Pnueli, 1977

- Systém lze chápat jako množinu sekvencí stavů – **běhů**.
- Vlastnosti systému lze vymezit vlastnostmi běhů.
- Vlastnosti běhů lze popsat temporální logikou **lineárního času**.

Clarke & Emerson, 1980

- Systém lze chápat jako strom možných pokračování, tzv. **výpočetní strom**. V každém okamžiku chodu systému existuje (konečně mnoho) možných pokračování (budoucích stavů).
- Systém lze vymezit vlastnostmi výpočetního stromu.
- Vlastnosti stromu lze popsat temporální logikou **větvícího se času**.

Systém a výpočetní strom z iniciálního stavu



Logika CTL
(Computation Tree Logic)

CTL neformálně

Možné výpočty

- Je-li dán výpočetní strom a jeden z jeho vrcholů, pak podstrom určený daným vrcholem udává všechny možné běhy, které systém z daného stavu může provést.
- O každém jednom takovém běhu mluvíme jako o možném výpočtu (možné budoucnosti).

CTL formule umožňují

- Specifikovat vlastnosti stavů pomocí atomických propozic.
- Kvantifikovat přes možné výpočty z daného stavu.
- Omezovat množinu možných výpočtů pomocí (kvantifikovaných) LTL operátorů.

Příklad

- $\varphi \equiv EF(a)$
- Je možné provést výpočet, ve kterém jednou bude platit a.

Syntax CTL

Nechť AP je množina atomických propozic. Pak

- Je-li $p \in AP$, pak p je formule.
- Je-li φ formule, pak $\neg\varphi$ je formule.
- Jsou-li φ a ψ formule, pak $\varphi \vee \psi$ je formule.
- Je-li φ formule, pak $EX\,\varphi$ je formule.
- Jsou-li φ a ψ formule, pak $E[\varphi\,U\,\psi]$ je formule.
- Jsou-li φ a ψ formule, pak $A[\varphi\,U\,\psi]$ je formule.

Alternativní zápis (Backus-Naur form)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid EX\,\varphi \mid E[\varphi\,U\,\varphi] \mid A[\varphi\,U\,\varphi]$$

Syntaktické zkratky

Standardní

- Klasické syntaktické zkratky výrokové logiky
- Syntaktické zkratky z LTL
  - $F\,\varphi \equiv \mathit{true}\,U\,\varphi$
  - $G\,\varphi \equiv \neg F\,\neg\varphi$

Odvozené temporální operátory CTL

- $EF\,\varphi \equiv E[\mathit{true}\,U\,\varphi]$
- $AF\,\varphi \equiv A[\mathit{true}\,U\,\varphi]$
- $EG\,\varphi \equiv \neg AF\,\neg\varphi$
- $AG\,\varphi \equiv \neg EF\,\neg\varphi$
- $AX\,\varphi \equiv \neg EX\,\neg\varphi$

Modely CTL formulí

Model CTL formule

- Je dána množina atomických propozic AP.
- Modelem CTL formule je stav $s \in S$ Kripkeho struktury $M = (S, T, I, s_0)$.

Připomenutí

- Běh v Kripkeho struktuře je maximální cesta začínající v daném (iniciálním) stavu.
- Na konečné běhy nahlížíme jako na nekonečné, které vzniknou opakováním posledního stavu.

Značení

- Nechť $s \in S$ je stav Kripkeho struktury $M = (S, T, I, s_0)$.
- $P_M(s) = \{\pi \mid \pi \text{ je běh začínající ve stavu } s\}$

Sémantika CTL

Předpoklady

- Je dána množina atomických propozic AP.
- Je dán stav $s \in S$ Kripkeho struktury $M = (S, T, I, s_0)$.
- φ, ψ jsou syntakticky správné CTL formule.
- $p \in AP$ je atomická propozice.

Sémantika

$$
\begin{aligned}
s \models p \quad &\text{iff} \quad p \in I(s) \\
s \models \neg\varphi \quad &\text{iff} \quad \neg(s \models \varphi) \\
s \models \varphi \vee \psi \quad &\text{iff} \quad s \models \varphi \text{ or } s \models \psi \\
s \models EX\,\varphi \quad &\text{iff} \quad \exists \pi \in P_M(s).\pi(1) \models \varphi \\
s \models E[\varphi\, U\, \psi] \quad &\text{iff} \quad \exists \pi \in P_M(s).(\exists k \geq 0.(\pi(k) \models \psi \text{ and} \\
&\qquad\qquad \forall 0 \leq i < k.\pi(i) \models \varphi)) \\
s \models A[\varphi\, U\, \psi] \quad &\text{iff} \quad \forall \pi \in P_M(s).(\exists k \geq 0.(\pi(k) \models \psi \text{ and} \\
&\qquad\qquad \forall 0 \leq i < k.\pi(i) \models \varphi))
\end{aligned}
$$

Příklad

Vyjádřete pomocí CTL formule

- Je možné dosáhnout stav, ve kterém platí a, ale neplatí b.
- Pokud systém obdrží žádost *Req*, pak v konečném čase vygeneruje potvrzení *Ack*.
- V každém možném výpočtu nekonečně mnohokrát platí b.
- Vždy je možné systém restartovat (dosáhnout stavu *Restart*).

Model Checking CTL

Definice problému

Model checking CTL

- Je dána Kripkeho struktura $M = (S, T, I, s_0)$.
- Je dána CTL formule φ.
- Problém: Platí, že $M, s_0 \models \varphi$?

Alternativně

- Je dána Kripkeho struktura $M = (S, T, I, s_0)$.
- Je dána CTL formule φ.
- Problém: Spočítat množinu $\{s \mid M, s \models \varphi\}$.

Pojmenování

- Výše uvedené přístupy se někdy také označují jako
  - Local model-checking problém — $M, s_0 \models \varphi$.
  - Global model-checking problém — $\{s \mid M, s \models \varphi\}$.
- Neplést s vlastností algoritmů.
  - Local algorithm for global model-checking.

Algoritmus pro CTL Model-Checking — Idea

Pozorování

- Znám-li pro každý stav platnost formulí φ a ψ, snadno odvodím platnost formulí $\neg\varphi$, $\varphi \vee \psi$, $EX\,\varphi$, ...

Idea algoritmu pro CTL Model Checking

- Je dána Kripkeho struktura $M = (S, T, I)$ a formule φ.
- Spočítám značkovací funkci $label : S \to 2^{2^{\varphi}}$, která o každém stavu $s \in S$ Kripkeho struktury M řekne, jaké podformule formule φ platí v daném stavu.
- Platí, že $s_0 \models \varphi \iff \varphi \in label(s_0)$.
- Funkci *label* budu počítat postupně pro jednotlivé podformule formule φ, a to od nejjednodušších podformulí (atomické propozice) ke složitějším (až po podformuli φ).

Podformule CTL formule

Podformule formule φ

- Je dána CTL formule φ.
- Množinu všech podformulí formule φ označujeme 2^{φ}.
- Množina 2^{φ} je definována induktivně dle struktury φ.

Definice 2^{φ}

1) $\varphi \in 2^{\varphi}$ (φ je podformule φ)
2) Jestliže $\eta \in 2^{\varphi}$ a
   - $\eta \equiv \neg\psi$, pak $\psi \in 2^{\varphi}$ (ψ je podformule φ)
   - $\eta \equiv \psi_1 \vee \psi_2$, pak $\psi_1, \psi_2 \in 2^{\varphi}$ (ψ_1, ψ_2 jsou podformule φ)
   - $\eta \equiv EX\,\psi$, pak $\psi \in 2^{\varphi}$ (ψ je podformule φ)
   - $\eta \equiv E[\psi_1\, U\, \psi_2]$, pak $\psi_1, \psi_2 \in 2^{\varphi}$ (ψ_1, ψ_2 jsou podformule φ)
   - $\eta \equiv A[\psi_1\, U\, \psi_2]$, pak $\psi_1, \psi_2 \in 2^{\varphi}$ (ψ_1, ψ_2 jsou podformule φ)
3) Žádná jiná formule není podformulí φ.

Transformace formule

Pozorování

- Je snazší prokazovat, platnost existenčně kvantifikovaných modálních operátorů než platnost univerzálně kvantifikovaných modálních operátorů.
- Pro účely verifikace CTL formule φ nad daným Kripkeho systémem M, vyjádříme formuli φ v modifikovaném tvaru.

Alternativní základní syntax CTL

- $\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid EX\,\varphi \mid E[\varphi\, U\, \varphi] \mid EG\,\varphi$

Příklad

- Jak se vyjádří $EG\,\varphi$ v původní základní syntaxi CTL?
- Jak se definují podformule CTL formule φ je-li φ zapsána pomocí alternativní syntax?

Algoritmus pro CTL Model-Checking

VSTUP: Kripkeho struktura $M = (S, T, I, s_0)$, CTL formule φ.
VÝSTUP: *True*, pokud $s_0 \models \varphi$, jinak *False*.

proc CTLMC(φ, M)
  label := I
  Solved := $AP \cap 2^{\varphi}$
  while $\varphi \notin$ Solved do
    foreach ($\eta \in \{\neg\psi_1, \psi_1 \vee \psi_2, EX\,\psi_1, E[\psi_1\,U\,\psi_2], EG\,\psi_1 \mid \psi_1, \psi_2 \in Solved\}$)do
      if ($\eta \in 2^{\varphi}$ and $\eta \notin$ Solved)
        then *label* := *updateLabel*(η, *label*, M)
          Solved := Solved $\cup \{\eta\}$
      fi
    od
  od
  return ($\varphi \in$ *label*(s_0))
end

Algoritmus pro CTL Model-Checking — updateLabel()

proc updateLabel(η, *label*, M)
  if ($\eta \equiv E[\psi_1 \, U \, \psi_2]$)
    then return *checkEU*(ψ_1, ψ_2, *label*, M)
  fi
  if ($\eta \equiv EG \, \psi$)
    then return *checkEG*(ψ, *label*, M)
  fi
  foreach (s $\in S$)do
    if ($\eta \equiv \neg\psi$ and $\psi \notin$ *label*(s)) or
      ($\eta \equiv \psi_1 \vee \psi_2$ and ($\psi_1 \in$ *label*(s) $\vee \, \psi_2 \in$ *label*(s))) or
      ($\eta \equiv EX \, \psi$ and ($\exists t \in \{t \mid (s,t) \in T\}$ takové, že $\psi \in$ *label*(t)))
      then *label*(s) := *label*(s) $\cup \{\eta\}$
    fi
  od
  return *label*
end

Algoritmus pro označení stavů podformulí $E[\psi_1 \, U \, \psi_2]$

VSTUP: Kripkeho struktura $M = (S, T, I)$,
Značkovací funkce $label : S \to 2^{\varphi}$, korektní vůči ψ_1 a ψ_2
VÝSTUP: Značkovací funkce $label : S \to 2^{\varphi}$, korektní vůči $E[\psi_1 \, U \, \psi_2]$

```
proc checkEU(ψ1, ψ2, label, M)
  Q := {s | ψ2 ∈ label(s)}
  foreach ( s ∈ Q)do
    label(s) := label(s) ∪ {E[ψ1 U ψ2]}
  od
  while (Q ≠ ∅) do
        choose s ∈ Q
        Q := Q ∖ {s}
        foreach ( t ∈ {t | T(t,s)}) do          /* all immediate predecessors */
          if (E[ψ1 U ψ2] ∉ label(t) ∧ ψ1 ∈ label(t))
            then label(t) := label(t) ∪ {E[ψ1 U ψ2]}
                 Q := Q ∪ {t}
          fi
        od
  od
  return label
end
```

Silně souvislé komponenty

Podgraf

- Nechť $G = (V, E)$ je graf, tj. $E \subseteq V \times V$.
- Graf $G' = (V', E')$ nazveme podgrafem grafu G pokud platí $V' \subseteq V$ a $E' = E \cap V' \times V'$.

Podgraf $C = (V', E')$ grafu $G = (V, E)$ se nazývá

- silně souvislá komponenta, pokud $\forall u, v \in V'$ platí, že $(u, v) \in E'^*$ a $(v, u) \in E'^*$.
- maximální silně souvislá komponenta (SCC), pokud C je silně souvislá komponenta a pro každé $v \in (V \setminus V')$ platí, že $(V' \cup \{v\}, E \cap (V' \cup \{v\} \times V' \cup \{v\}))$ není silně souvislá komponenta.
- netriviální silně souvislá komponenta, pokud C je silně souvislá komponenta a $E' \neq \emptyset$.

Algoritmus pro označení stavů podformulí $EG\,\psi$

VSTUP: Kripkeho struktura $M = (S, T, I, s_0)$,
Značkovací funkce $label : S \to 2^{\varphi}$, korektní vůči ψ
VÝSTUP: Značkovací funkce $label : S \to 2^{\varphi}$, korektní vůči $EG\,\psi$

proc checkEG(ψ, $label$, M)
  S' := {s | $\psi \in label$(s)}
  SCC := {C | C je netriviální SCC grafu $G' = (S', T \cap S' \times S')$}
  Q := $\bigcup_{C \in SCC} \{s \mid s \in C\}$
  foreach (s ∈ Q)do
    $label$(s) := $label$(s) ∪ {$EG\,\psi$}
  od
  while Q ≠ ∅ do
      choose s ∈ Q
      Q := Q ∖ {s}
      foreach (t ∈ ($S' \cap \{t \mid T(t, s)\}$))do /* *all immediate predecessors in* S' */
        if $EG\,\psi \notin label$(t)
          then $label$(t) := $label$(t) ∪ {$EG\,\psi$}
               Q := Q ∪ {t}
        fi
      od
  od
end

Složitost algoritmu pro ověřování formule CTL

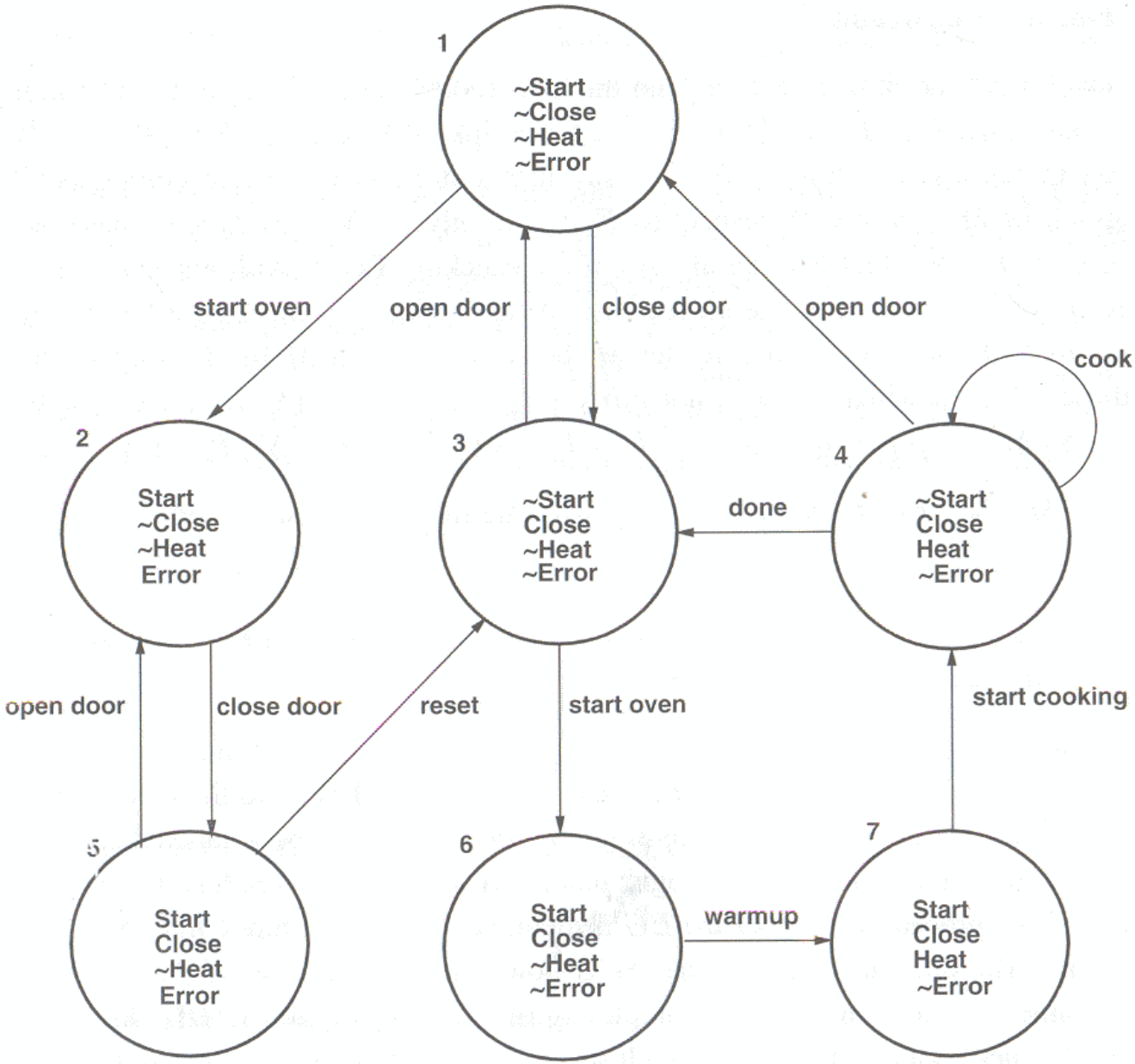
Pozorování

- Každá CTL formule φ má nejvýše $\mid \varphi \mid$ různých podformulí.
- Rozklad každého podgrafu grafu $G = (S, T)$ na SCC lze provést v čase $\mathcal{O}(\mid S \mid + \mid T \mid)$.
- Každé volání funkce *updateLabel* skončí v čase $\mathcal{O}(\mid S \mid + \mid T \mid)$.

Celková složitost

- Algoritmus *CTLMC* má časovou složitost $O(\mid \varphi \mid (\mid S \mid + \mid T \mid))$.
- Algoritmus *CTLMC* má prostorovou složitost $O(\mid \varphi \mid\mid S \mid)$.

Příklad: Mikrovlnná trouba $AG(Start \implies AF(Heat))$



Příklad: Mikrovlnná trouba $AG(Start \implies AF(Heat))$

Přepis formule $\varphi \equiv AG(Start \implies AF(Heat))$

- $AG(Start \implies AF(Heat))$
- $AG(\neg(Start \wedge \neg AF(Heat)))$
- $AG(\neg(Start \wedge EG(\neg Heat)))$
- $\neg EF(Start \wedge EG(\neg Heat))$
- $\neg E[true\ U\ (Start \wedge EG(\neg Heat))]$

Platnost podformulí $[S(\varphi) = \{s \mid s \models \varphi\}]$

- $S(Start) = \{2, 5, 6, 7\}$
- $S(Heat) = \{4, 7\}$
- $S(\neg Heat) = \{1, 2, 3, 5, 6\}$
- $S(EG(\neg Heat)) = \{1, 2, 3, 5\}$
- $S(Start \wedge EG(\neg Heat)) = \{2, 5\}$
- $S(E[true\ U\ (Start \wedge EG(\neg Heat))]) = \{1, 2, 3, 4, 5, 6, 7\}$
- $S(\neg E[true\ U\ (Start \wedge EG(\neg Heat))]) = \emptyset$

Logika CTL*

CTL* jako rozšíření CTL

Pozorování

- V logice CTL není možné omezit množinu možných výpočtů libovolnou *LTL* formulí. Tj. každý modální operátor LTL musí být bezprostředně předcházen kvantifikátorem.

Logika CTL*

- Logika větvícího se času stejně jako logika CTL.
- Množiny možných běhů lze omezit libovolnou *LTL* formulí.
- V syntax logiky CTL* vystupují kvantifikátory cest jako samostatné operátory.

Příklad

- $A[p \wedge X(\neg p)]$ je formule CTL*, ale není to formule CTL.

Syntax CTL*

Typy CTL* formulí

- Operátory E a A jsou samostatné, proto existují v CTL* formule jejichž modelem je běh Kripkeho struktury.
- Aplikací operátorů E a A vznikají z formulí jejichž modelem je běh Kripkeho struktury, formule, jejichž modelem je stav Kripkeho struktury.
- Rozlišujeme tedy formule stavu a formule cesty.

Syntax CTL*

formule stavu	$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid E\,\psi$
formule cesty	$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid X\,\psi \mid \psi\,U\,\psi$

Sémantika CTL*

Předpoklady

- Je dána množina atomických propozic AP, $p \in AP$.
- Je dána Kripkeho struktura $M = (S, T, I)$.
- φ_1, φ_2 jsou CTL* formule stavu, ψ_1, ψ_2 formule cesty.

Sémantika

$$
\begin{array}{rcl}
M, s \models p & \text{iff} & p \in I(s) \\
M, s \models \neg\varphi_1 & \text{iff} & \neg(M, s \models \varphi_1) \\
M, s \models \varphi_1 \vee \varphi_2 & \text{iff} & M, s \models \varphi_1 \text{ or } M, s \models \varphi_2 \\
M, s \models E\,\psi_1 & \text{iff} & \exists \pi \in P_M(s).\pi \models \psi_1 \\
M, \pi \models \varphi_1 & \text{iff} & M, \pi(0) \models \varphi_1 \\
M, \pi \models \neg\psi_1 & \text{iff} & \neg(M, \pi \models \psi_1) \\
M, \pi \models \psi_1 \vee \psi_2 & \text{iff} & M, \pi \models \psi_1 \text{ or } M, \pi \models \psi_2 \\
M, \pi \models X\,\psi_1 & \text{iff} & M, \pi^1 \models \psi_1 \\
M, \pi \models \psi_1\,U\,\psi_2 & \text{iff} & \exists k \geq 0.(M, \pi^k \models \psi_2 \text{ and} \\
 & & \forall 0 \leq i < k.M, \pi^i \models \psi_1)
\end{array}
$$

Porovnání logik LTL, CTL a CTL*

Unifikace modelů

Pozorování

- Každá LTL formule je CTL* formule cesty.
- Každá CTL formule je CTL* formule stavu.
- Modelem CTL* formule cesty je běh Kripkeho struktury.
- Modelem CTL* formule stavu je stav Kripkeho struktury.
- Nevhodné pro účely porovnání.

Unifikace modelu

- Za účelem unifikace modelů definujeme, kdy CTL* formule cesty platí ve stavu Kripkeho struktury.
- Nechť ψ je CTL* formule cesty, pak

$$M, s \models \psi \quad \text{iff} \quad M, s \models A\,\psi$$

Motivace

Cíl

- Chceme zjistit, zda jsou vlastnosti (formule), které lze vyjádřit v jedné logice a nelze vyjádřit v jiné logice.
- Chceme zjistit, ve které logice lze vyjádřit víc vlastností.
- Chceme identifikovat vlastnosti, které nelze vyjádřit v jiné logice, tj. jak vypadá formule logiky $\mathcal{L}_1$, pro kterou neexistuje ekvivalentní formule logiky $\mathcal{L}_2$.

Ekvivalence formulí (i různých logik)

- Formule φ a ψ jsou ekvivalentní, právě když pro všechny modely $M = (S, T, I, s_0)$ a stavy $s \in S$ platí

$$M, s \models \varphi \quad \text{iff} \quad M, s \models \psi$$

Expresibilita – vyjadřovací síla

Shodná expresibilita

- Temporální logiky $\mathcal{L}_1$ a $\mathcal{L}_2$ jsou shodně expresibilní (mají stejnou vyjadřovací sílu), pokud pro všechny modely $M = (S, T, I, s_0)$ a stavy $s \in S$ platí

$$\forall \varphi \in \mathcal{L}_1.(\exists \psi \in \mathcal{L}_2.(M, s \models \varphi \iff M, s \models \psi)) \quad (1)$$
$$\wedge\ \forall \psi \in \mathcal{L}_2.(\exists \varphi \in \mathcal{L}_1.(M, s \models \varphi \iff M, s \models \psi)). \quad (2)$$

Menší expresibilita

- Pokud platí pouze tvrzení (1), tj. neplatí tvrzení (2), pak je logika $\mathcal{L}_1$ méně expresibilní (má menší vyjadřovací sílu) než logika $\mathcal{L}_2$.

Porovnání LTL, CTL, a CTL*

Tvrzení 1

- LTL a CTL jsou vyjadřovací silou neporovnatelné.
  1) $AG(EF(q))$ je CTL formule, kterou nelze vyjádřit v LTL.
  2) $FG(q)$ je LTL formule, kterou nelze vyjádřit v CTL.

Příklad – důkaz 1)

- Najděte dvě různé Kripkeho struktury a v nich identifikujte stavy, které jsou rozlišitelné CTL formulí $AG(EF(q))$ a přitom nejsou rozlišitelné žádnou LTL formulí (generují shodnou množinu běhů).

Příklad – intuice za 2) [důkaz jde nad rámec tohoto kurzu]

- Ukažte, že CTL formule $AF(AG(q))$ není ekvivalentní LTL formuli $FG(q)$.

Porovnání LTL, CTL, a CTL*

Důsledek 1

- CTL* je striktně více expresibilní než LTL.
  - Každá LTL formule je i formule CTL*.
  - CTL* formule $AG(EFq)$ není vyjádřitelná v LTL.

Důsledek 2

- CTL* je striktně více expresibilní než CTL.
  - Každá CTL formule je i formule CTL*.
  - CTL* formule $FG(q)$ není vyjádřitelná v CTL.

Pozorování

- Existují vlastnosti vyjádřitelné jak v LTL tak i v CTL.
  - CTL formule $A[p\,U\,q]$ je ekvivalentní LTL formuli $p\,U\,q$.