

Temporální logiky

Radek Mařík

ČVUT FEL

Katedra telekomunikační techniky, K13132

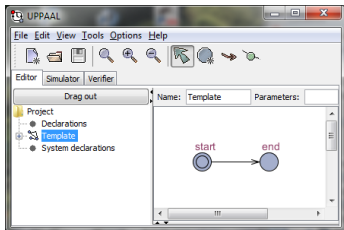
6. prosince 2017



- 1 **System UPPAAL**
 - Postup modelování a ověřování
- 2 **Základy temporálních logik**
 - Cesty výpočtu a čas
 - CTL* logika
 - CTL logika
- 3 **UPPAAL**
 - Specifikace požadavků v UPPAAL
- 4 **Hra NIM**
 - Specifikace požadavků hry NIM



Tvorba automatu [UPP09]



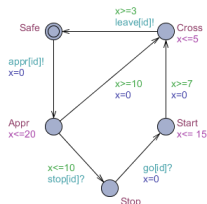
Automat

- počáteční pozice (dvojitá kružnice)
- "Add Location" pro přidání pozice
- "Selection Tool" pro pojmenování pozice
- "Add Edge" pro přidání hrany, prohnutí hran pomocí myši v okolí konců
- dolní tabulka "Position" a "Description" pro analýzu chyb

Kompozice systému ^[UPP09]

Systém

- **Systém** ... síť paralelních časovaných automatů (procesů).
- **Proces** ... instance parametrizovaného vzoru.

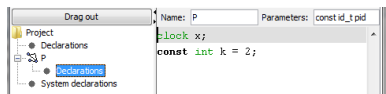
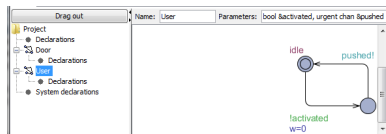


Proces

- **Pozice** ...
 - jméno,
 - invarianty
- **Hrany** ...
 - podmínky stráží ($x \geq 7$),
 - synchronizace ($go[id]?$),
 - přiřazení ($x = 0$),



Popis vzoru (template) [UPP09]



Parametrizovaný časový automat

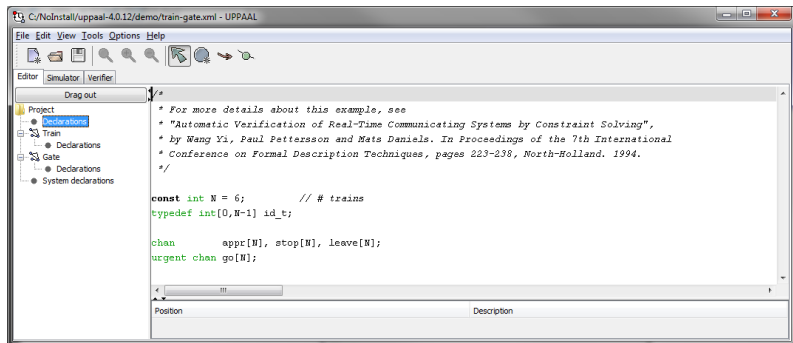
- jméno,
- parametry,

Lokální deklarace

- proměnné,
- synchronizační kanály,
- konstanty



Popis systému [UPP09]



Globální deklarace

- globální celočíselné proměnné,
- globální hodiny,
- synchronizační kanály,
- konstanty

Definice systému ^[UPP09]

```
bool activated1, activated2;  
urgent chan pushed1, pushed2;  
urgent chan closed1, closed2;  
  
Door1 = Door(activated1, pushed1, closed1, closed2);  
Door2 = Door(activated2, pushed2, closed2, closed1);  
User1 = User(activated1, pushed1);  
User2 = User(activated2, pushed2);  
  
system Door1, Door2, User1, User2;
```

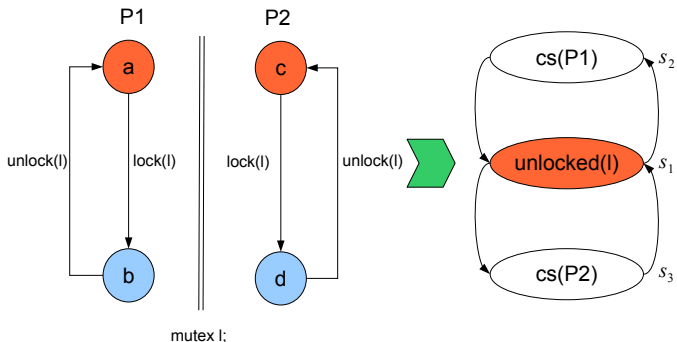
Přřazení procesů

- deklarace instancí procesu,
- vzory s úplně/částečně specifikovanými parametry,

Definice systému

- seznam procesů systému,

Přechody mezi konfiguracemi v Kripkeho struktuře [Voj10]



Cesta v Kripkeho struktuře ^[Voj10]

Cesta

- **Cesta** $\pi \dots$ v Kripkeho struktuře M je nekonečná sekvence stavů $\pi = s_0s_1s_3 \dots$ taková, že $\forall i \in \mathbb{N}. R(s_i, s_{i+1})$.
- $\Pi(M, s) \dots$ množina všech cest v M , které začínají v $s \in S$
- Sufix π^i cesty $\pi = s_0s_1s_3 \dots s_i s_{i+1} s_{i+2}$ je cesta $\pi^i = s_i s_{i+1} s_{i+2}$ začínající v s_i .
- $s_j = \pi[i]$



Pojem času ^[Voj10]

Abstrakce času

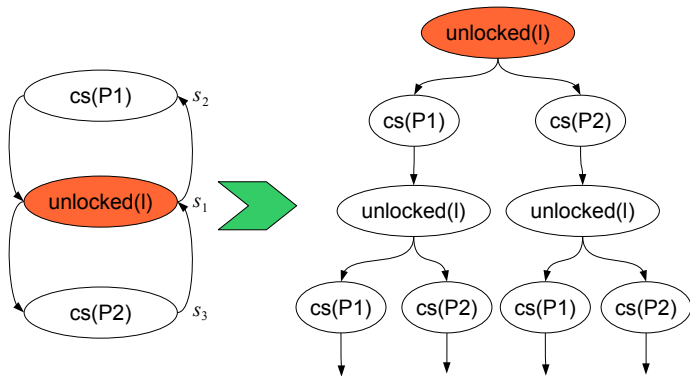
- **Logický čas** ... pracuje s (*částečným*) *uspořádáním* stavů/událostí v chování systému.
- **Fyzický čas** ... *měření* doby uběhlou mezi dvěma stavy/události.

Čas ve verifikaci modelů

- **Lineární čas** ... dovoluje se vyjadřovat pouze o dané *lineární trase* ve stavovém prostoru.
 - Na všech trasách, x musí být následováno y .
 - Na všech trasách, x musí být následováno y nebo z .
- **Větvící se čas** ... dovoluje kvantifikovat (existenčně i univerzálně) možné budoucnosti počínaje daným stavem. Na stavový prostor se pohlíží jako na rozvinutý *nekonečný strom*.
 - Existuje trasa, kde následující stav je x .

Výpočetní strom ^[Voj10]

Popisuje vlastnosti výpočetního stromu



CTL* formule ^[Voj10]

Skládá se z

- atomické výroky
- logické spojky
- kvantifikátory cest
- temporální operátory



CTL* kvantifikátory a operátory [Wik10, Voj10]

Kvantifikátory cest

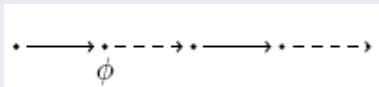
popisují strukturu větvení výpočetního stromu

- E ... existuje cesta výpočtu z daného stavu.
- A ... pro všechny cesty výpočtů z daného stavu.

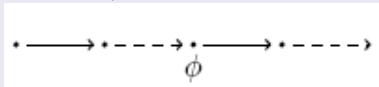
Temporální operátory

určují vlastnosti cesty ve výpočetním stromu

- $X\varphi$ (next time, \bigcirc)... vlastnost φ platí ve druhém (následujícím) stavu cesty..



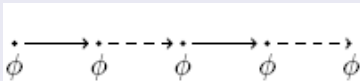
- $F\varphi$ (in future, \diamond)... vlastnost φ platí v nějakém stavu cesty.



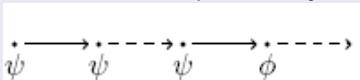
CTL* operátory [Wik10, Voj10]

Temporální operátory

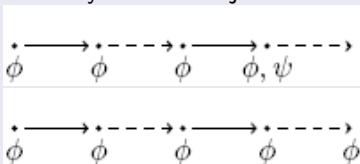
- $G\varphi$ (globally, \square). . . vlastnost φ platí ve všech stavech cesty.



- $\psi U\varphi$ (until). . . vlastnost φ platí v nějakém stavu cesty a vlastnost ψ platí přinejmenším ve všech předcházejících stavech této cesty.



- $\psi R\varphi$ (release). . . vlastnost φ musí platit do (a včetně) stavu, kdy začne platit vlastnost ψ , pokud takový stav existuje.



CTL* syntax ^[Voj10]

Nechť AP je neprázdňá množina atomických výroků.

Syntax stavových formulí, které jsou pravdivé v daném stavu

- Jestliže $p \in AP$, potom p je stavová formule.
- Jestliže φ a ψ jsou stavové formule, potom $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$ jsou stavové formule.
- Jestliže φ je běhová formule, potom $E\varphi$ a $A\varphi$ jsou stavové formule.

Syntax běhových formulí, které jsou pravdivé podél specifické cesty

- Jestliže φ je stavová formule, pak φ je také běhová formule.
- Jestliže φ a ψ jsou běhové formule, pak $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, $X\varphi$, $F\varphi$, $G\varphi$, $\varphi U\psi$ a $\varphi R\psi$ jsou běhové formule.

CTL* je množina stavových formulí generovaných výše uvedenými pravidly.



CTL* sémantika ^[Voj10]

- Nechť je dána Kripkeho struktura $M = (S, T, \mathcal{I}, s_0, L)$ nad množinou atomických výroků AP .
- Pro stavovou formuli φ nad AP , zapisujeme $M, s \models \varphi$ fakt, že φ platí v $s \in S$.
- Pro běhovou formuli φ nad AP , zapisujeme $M, \pi \models \varphi$ fakt, že φ platí podél cesty π v M .
- Nechť $s \in S$, π je cesta v M , φ_1, φ_2 jsou stavové formule nad AP , $p \in AP$, a ψ_1, ψ_2 jsou běhové formule nad AP . Pak relaci \models definujeme induktivně následovně:
 - $M, s \models p$ iff $p \in L(s)$.
 - $M, s \models \neg\varphi_1$ iff $M, s \not\models \varphi_1$.
 - $M, s \models \varphi_1 \vee \varphi_2$ iff $M, s \models \varphi_1$ nebo $M, s \models \varphi_2$.
 - $M, s \models \varphi_1 \wedge \varphi_2$ iff $M, s \models \varphi_1$ a zároveň $M, s \models \varphi_2$.
 - $M, s \models E\psi_1$ iff $\exists \pi \in \Pi(M, s). M, \pi \models \psi_1$.
 - $M, s \models A\psi_1$ iff $\forall \pi \in \Pi(M, s). M, \pi \models \psi_1$.



CTL* sémantika [Voj10]

- Pokračování definice relace \models :

- $M, \pi \models \varphi_1$ iff $M, s_0 \models \varphi_1, s_0 = \pi[0]$.
- $M, \pi \models \neg\psi_1$ iff $M, \pi \not\models \psi_1$.
- $M, \pi \models \psi_1 \vee \psi_2$ iff $M, \pi \models \psi_1$ nebo $M, \pi \models \psi_2$.
- $M, \pi \models \psi_1 \wedge \psi_2$ iff $M, \pi \models \psi_1$ a zároveň $M, \pi \models \psi_2$.
- $M, \pi \models X\psi_1$ iff $M, \pi^1 \models \psi_1$.
- $M, \pi \models F\psi_1$ iff $\exists i \geq 0. M, \pi^i \models \psi_1$.
- $M, \pi \models G\psi_1$ iff $\forall i \geq 0. M, \pi^i \models \psi_1$.
- $M, \pi \models \psi_1 U\psi_2$ iff $\exists i \geq 0. M, \pi^i \models \psi_2$
a zároveň $\forall 0 \leq j < i. M, \pi^j \models \psi_1$.
- $M, \pi \models \psi_1 R\psi_2$ iff $\forall i \geq 0. (\forall 0 \leq j < i. M, \pi^j \not\models \psi_1 \Rightarrow M, \pi^i \models \psi_2)$.



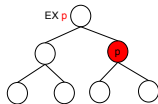
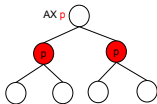
CTL* základní operátory ^[Voj10]

- Všechny CTL* operátory lze odvodit z \vee , \neg , X , U a E :
 - Nech $p \in AP$, $true \equiv p \vee \neg p$ (a $false \equiv \neg true$)
 - $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$,
 - $F\varphi \equiv true U \varphi$,
 - $G\varphi \equiv \neg F\neg\varphi$,
 - $\varphi R \psi \equiv \neg(\neg\varphi U \neg\psi)$,
 - $A\varphi \equiv \neg E\neg\varphi$.

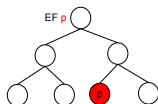
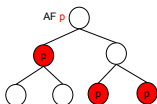


CTL syntaxe ^[Voj10]

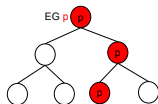
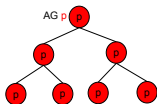
- CTL je sublogikou CTL*
 - běhové formule jsou omezeny na $X\varphi$, $F\varphi$, $G\varphi$, $\varphi U\psi$ a $\varphi R\psi$,
 - kde φ a ψ jsou stavové formule.
- Proto pouze 10 modálních CTL operátorů:
 - AX a EX



- AF a EF

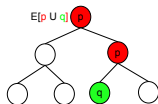
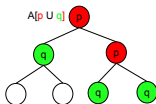


- AG a EG

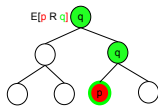
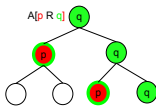


CTL modální operátory ^[Voj10]

- Modální CTL operátory:
 - AU a EU



- AR a ER



- Existují 3 základní CTL modální operátory - EX , EG a EU :

- $AX\varphi \equiv \neg EX\neg\varphi$

- $EF\varphi \equiv E[\text{true}U\varphi]$

- $AG\varphi \equiv \neg EF\neg\varphi$

- $AF\varphi \equiv \neg EG\neg\varphi$

- $A[\varphi U\psi]$

$$\equiv \neg E[\neg\psi U(\neg\varphi \wedge \neg\psi)] \wedge \neg EG\neg\psi$$

- $A[\varphi R\psi] \equiv \neg E[\neg\varphi U\neg\psi]$

- $E[\varphi R\psi] \equiv \neg A[\neg\varphi U\neg\psi]$



BNF gramatika specifikačního jazyka ^[UPP10]

BNF gramatika

- $A \rightarrow Expression$
- $E \langle \rangle Expression$
- $E \square Expression$
- $A \langle \rangle Expression$
- $Expression - - \rangle Expression$

Poznámky

- Žadný výraz nesmí mít postranní efekty.
- Výraz *process.location* testuje, zda určitý proces je v dané pozici.



Příklady specifikačního jazyka [UPP10]

BNF gramatika

- $A \square 1 < 2$
 - Invariantně $1 < 2$
- $E \langle \rangle p1.cs \text{ and } p2.cs$
 - Pravdivé, pokud systém může dosáhnout stavu, ve kterém procesy $p1$ a $p2$ jsou v jejich pozici cs
- $A \langle \rangle p1.cs \text{ simply not } p2.cs$
 - Invariantně proces $p1$ v pozici cs implikuje, že proces $p2$ **není** v pozici cs .
- $A \square \text{not deadlock}$
 - Invariantně, proces neobsahuje deadlock.



Jednoduchá varianta NIM

The Nim Number Game

Whoever takes the last proton wins!

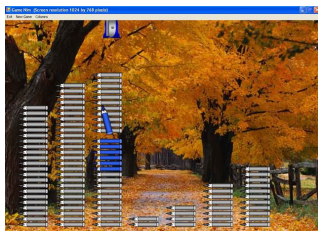
Press the "I'm ready! Let's start!" button to begin!



- NIM je hra založená na logice a strategii.
- Hrají 2 hráči.
- Hráč při svém tahu odstraní jednu až *MAX* (2) věci (zápalky, protony) z řady.
- Vyhrává ten hráč, který odstraní poslední věc.



Klasická varianta NIM



- NIM je hra založená na logice a strategii.
- Hrají 2 hráči.
- Hráči odebírají objekty z různých hromádek/řad.
- Hráč musí odstranit při svém tahu alespoň jeden objekt.
- Hráč při svém tahu odstraní libovolný počet objektů, které náležejí všechny k jedné hromádce.
- Základní varianty hry:
 - **Normální** ... Vyhrává ten hráč, který odstraní poslední věc.
 - **Prohra** ... Prohrává ten hráč, který odstraní poslední věc.



Literatura I



UPPAAL 4.0: Small tutorial, November 2009.



Tool environment for validation and verification of real-time systems (UPPAAL pamphlet).
<http://www.it.uu.se/research/group/darts/papers/texts/uppaal-pamphlet.pdf>, September 2010.



Tomas Vojnar.

Formal analysis and verification.

Lecture handouts, <http://www.fit.vutbr.cz/study/courses/FAV/public/>, August 2010.



Linear temporal logic.

http://en.wikipedia.org/wiki/Linear_temporal_logic, November 2010.

