

# Verifikace Modelů a UPPAAL

Radek Mařík

Czech Technical University  
Faculty of Electrical Engineering  
Department of Telecommunication Engineering  
Prague CZ

November 7, 2017



## 1 Úvod

- Motivace
- Úvod do verifikace modelů
- Formální popis

## 2 UPPAAL - Přehled

- Základní vlastnosti
- Architektura

## 3 UPPAAL - Vybrané vlastnosti

- Systém a proces
- Manuál

# Outline

## 1 Úvod

- Motivace
- Úvod do verifikace modelů
- Formální popis

## 2 UPPAAL - Přehled

- Základní vlastnosti
- Architektura

## 3 UPPAAL - Vybrané vlastnosti

- Systém a proces
- Manuál

# Garantování správného chování systémů [Cam10]

- zvyšování složitosti softwarových systémů,
- chyby vedou ke ztrátám
  - finančním,
  - na lidských životech.
- typicky problém systémů s kritickou bezpečností
  - letadla,
  - satelity,
  - lékařská zařízení.



# Cíle kladené na požadavky <sup>[Cam10]</sup>

- Demonstrace, že požadavky jsou
  - správné,
  - úplné,
  - přesné,
  - konzistentní,
  - testovatelné.

# Metody zajištění kvality <sup>[Cam10]</sup>

- **Testování a Simulace** poskytuje pouze pravděpodobnostní zajištění.
- **Verifikace za běhu** . . . technika kombinující formální verifikaci s během programu.
- **Formální verifikace**. . . technika založená na formálních metodách stavějící na matematicky založených jazycích, které umožňují specifikaci a verifikaci systémů.
  - **Specifikace** . . . zapsání požadavků na systém v matematickém jazyku.
  - **Verifikace** . . . formální důkaz toho, že systém splňuje požadavky.



# Princip formální verifikace [Cam10, Če09]

## Vstupy

- (matematický) model systému,
  - formální model  $M$ ,
- specifikace požadavků kladených na systém
  - formule  $\varphi$  určité temporální logiky,

## Verifikace

- Ověření, že systém splňuje specifikaci.
  - rozhodnutí, zda-li  $M$  je modelem formule  $\varphi$ , tj.  $M \models \varphi$

# Typologie formální verifikace [Cam10]

## Techniky

- **Statická analýza** ... ověření chování programu, aniž by se musel spustit.
  - **Abstraktní statická analýza** ... založená na *abstraktní interpretaci* používající aproximační abstraktní reprezentace k ověřování přibližných vlastností složitých systémů
    - analýza ukazatelů v moderních kompilátorech.
  - **Ověřování modelů** ... úplné procházení dosažitelných stavů programu.
  - **Omezené ověřování modelů** ... úplné procházení dosažitelných stavů programu pouze do určité hloubky.
- **Dokazování vět** ... nalezení důkazu vlastnosti, kdy systém i jeho vlastnosti jsou vyjádřeny jako formule v nějaké matematické logice.





# Řešitelnost temporálně logických formalismů <sup>[Hol06]</sup>

## Ověřování modelů

- Ptáme se, zda daný systém splňuje požadovanou vlastnost.
- Tj. pro strukturu reprezentující systém je třeba zjistit, jestli je modelem zadané formule.
- využitelné pro verifikaci existujících programů.

## Splnitelnost formulí

- Problém rozhodnutí, zda existuje nějaký model zadané formule.
- využitelné při automatické syntéze programů.

# Outline

## 1 Úvod

- Motivace
- Úvod do verifikace modelů
- Formální popis

## 2 UPPAAL - Přehled

- Základní vlastnosti
- Architektura

## 3 UPPAAL - Vybrané vlastnosti

- Systém a proces
- Manuál

# Verifikace modelů <sup>[Cam10]</sup>

## Princip

- budování konečného modelu systému,
- kontrola, zda požadovaná vlastnost je modelem dodržena,
- založeno na úplném prohledání stavového prostoru.

## Základní vlastnosti

- manipulace s obrovskými prohledávacími prostory,
- odpověď je "ano" či "ne", v záporném případě systém poskytuje
  - protipříklad, tj. běh systému, který neodpovídá vlastnosti.
- analýza specifikace softwarových systémů.



# Verifikace modelů v praxi <sup>[Cam10]</sup>

## Aplikace

- ověření hardwaru (obvody),
- ověření protokolů,
- analýza specifikace softwarových systémů.

# Přístupy verifikace modelů <sup>[Cam10]</sup>

## Temporální verifikace modelů

- použití temporální logiky (vyjádření času),
- systémy modelovány jako přechodové systémy s konečným počtem stavů.

## Automatový přístup

- specifikace i model vyjádřen jako automaty,
- oba automaty se porovnávají
  - jazyková inkluze,
  - zjemňující uspořádání,
  - pozorovací ekvivalence.



# Výhody/nevýhody verifikace modelů [Cam10]

## Výhody

- úplná automatizace,
- vysoká rychlost,
- možnost verifikace i částečných specifikací,
- produkuje protipříklady.

## Nevýhody

- problém exploze stavů,
  - binární rozhodovací diagramy (BDD),
  - nástroje jsou schopny zvládnout systémy s 100 – 200 stavovými proměnnými
  - je možné zvládnout systémy s  $10^{120}$  stavy.



# Rozšíření metod verifikace modelů <sup>[Bie08]</sup>

## Odstraňování konečnosti

- spojité proměnné,
- spojitý čas,
- práce s pravděpodobnostmi,
- parametrizace velikosti či počtu komponent,
- náhrada konečných automatů zásobníkovými automaty.

# Temporální logika <sup>[Bie08]</sup>

## Vyšetření sekvenčního či temporálního chování systému

- reaktivní, distribuované či paralelní systémy,
- A. Pnueli upozornil na tuto myšlenku jako první,

## Ověřované vlastnosti

- **Bezpečnost** . . . vlastnost stanovující, že určitá chyba či katastrofický stav není dosažitelný.
  - všechny dosažitelné stavy splňují určitý invariant.
- **Živost** . . . něco jednou nastane,
- **Férovost** . . . ,





# Outline

## 1 Úvod

- Motivace
- Úvod do verifikace modelů
- **Formální popis**

## 2 UPPAAL - Přehled

- Základní vlastnosti
- Architektura

## 3 UPPAAL - Vybrané vlastnosti

- Systém a proces
- Manuál

# Stavový prostor [Če09]

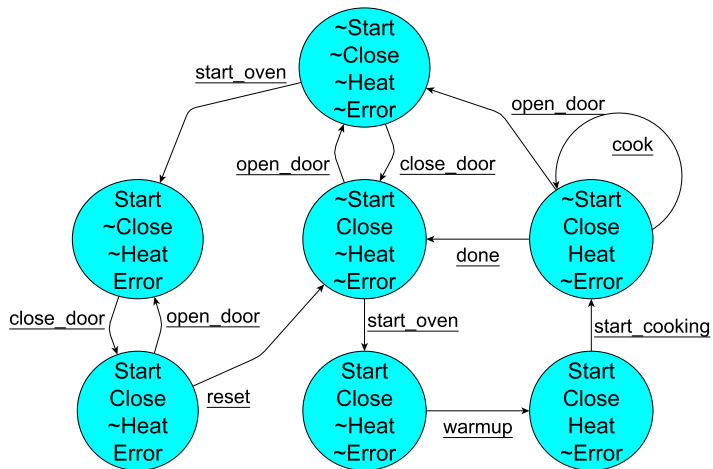
## Aplikovatelné jen na konečné stavové prostory

- Verifikovat se dají pouze ty parametry modelu, které jsou specifikovány.
- Stavový prostor lze formalizovat za použití atomických výroků a Kripkeho struktury

## Atomické výroky

- základní tvrzení popisující daný systém
  - výrazy,
  - konstanty,
  - predikátové symboly.
- Každý atomický výrok je algoritmicky rozhodnutelný na základě daného stavu.
- Stav ... ohodnocení všech proměnných.

## Kripkeho struktura - mikrovlnná trouba [?]



# Kripkeho struktura <sup>[Če09]</sup>

Kripkeho struktura je typ nedeterministického konečného automatu.

## Kripkeho struktura

- Je dána množina atomických propozic  $AP$ .
- Kripkeho struktura je trojice  $(S, T, \mathcal{I})$ , kde
  - $S$  je konečná množina stavů,
  - $T \subseteq S \times S$  je přechodová relace,
  - $\mathcal{I} : S \rightarrow 2^{AP}$  je interpretace  $AP$ .

## Rozšířená Kripkeho struktura

- je čtveřice  $(S, T, \mathcal{I}, s_0)$ , kde
  - $(S, T, \mathcal{I})$  je Kripkeho struktura,
  - $s_0$  je počáteční stav.



# Kripkeho přechodový systém [Če09]

Máme-li danou množinu  $Act$  akcí proveditelných programem, můžeme Kripkeho struktury rozšířit o označení přechodu.

## Kripkeho přechodový systém

- je pětice  $(S, T, \mathcal{I}, s_0, L)$ , kde
  - $(S, T, \mathcal{I}, s_0)$  je rozšířená Kripkeho struktura,
  - $L : T \rightarrow Act$  je značkovací funkce.



# Outline

- 1 Úvod
  - Motivace
  - Úvod do verifikace modelů
  - Formální popis
- 2 UPPAAL - Přehled
  - Základní vlastnosti
  - Architektura
- 3 UPPAAL - Vybrané vlastnosti
  - Systém a proces
  - Manuál

# Nástroj v kostce <sup>[UPP10]</sup>

## Nástroj integrující prostředí

- pro modelování,
- simulaci,
- a verifikaci,
- reálných systémů.

## Vývojové týmy

- **Uppsala** University, Švédsko,
- **Aalborg** University, Dánsko.

# Nástroj v kostce <sup>[UPP10]</sup>

## Nástroj integrující prostředí

- pro modelování,
- simulaci,
- a verifikaci,
- reálných systémů.

## Vývojové týmy

- **Uppsala** University, Švédsko,
- **Aalborg** University, Dánsko.



# Modely systémů <sup>[UPP10]</sup>

## Vlastnosti modelů

- sada nedeterministických procesů
- s konečnou řídicí strukturou a
- reálnými hodinami,
- komunikující pomocí kanálů nebo
- sdílených proměnných

# Implementace <sup>[UPP10]</sup>

## Hlavní návrhová kritéria

- výkonnost,
  - vyhledávací stroj *za letu*
  - symbolické techniky
- snadno použitelné.
- diagnostický záznam
  - může být generován verifikátorem a přehráván simulátorem

## Dostupnost

- První verze v roce 1995
- Současná verze je 4.0.12
- grafická rozhraní jsou implementována v Java
- verifikátor je implementován v C++
- dostupné pro Linux, SunOS, MS Windows (95/98/NT/2000/XP/Vista/7)

# Implementace <sup>[UPP10]</sup>

## Hlavní návrhová kritéria

- výkonnost,
  - vyhledávací stroj *za letu*
  - symbolické techniky
- snadno použitelné.
- diagnostický záznam
  - může být generován verifikátorem a přehráván simulátorem

## Dostupnost

- První verze v roce 1995
- Současná verze je 4.0.12
- grafická rozhraní jsou implementována v Java
- verifikátor je implementován v C++
- dostupné pro Linux, SunOS, MS Windows (95/98/NT/2000/XP/Vista/7)

# Průmyslové studie <sup>[UPP10]</sup>

## Případové studie

- audio/video protokol
  - komunikace mezi audio/video komponentami pomocí jediné sběrnice
- protokol vysílání s ohraničeným opakováním,
- protokol pro vyhnutí se kolizím
  - média založená na Eternetu
- řadič spojky automobilů,
- protokol řízení audio komponent (Philips)
- TDMA(Time Division Multiple Access) protokol mechanismu start-up
  - synchronizace 3 komunikujících stanic s libovolného počátečního stavu.

## Typické aplikace

- řadiče reálného času,
- komunikační protokoly.

# Průmyslové studie <sup>[UPP10]</sup>

## Případové studie

- audio/video protokol
  - komunikace mezi audio/video komponentami pomocí jediné sběrnice
- protokol vysílání s ohraničeným opakováním,
- protokol pro vyhnutí se kolizím
  - média založená na Eternetu
- řadič spojky automobilů,
- protokol řízení audio komponent (Philips)
- TDMA(Time Division Multiple Access) protokol mechanismu start-up
  - synchronizace 3 komunikujících stanic s libovolného počátečního stavu.

## Typické aplikace

- řadiče reálného času,
- komunikační protokoly.

# Outline

- 1 Úvod
  - Motivace
  - Úvod do verifikace modelů
  - Formální popis
- 2 UPPAAL - Přehled
  - Základní vlastnosti
  - **Architektura**
- 3 UPPAAL - Vybrané vlastnosti
  - Systém a proces
  - Manuál

# Komponenty systému <sup>[UPP10]</sup>

## Jazyk popisu

- jazyk nedeterministických podmíněných příkazů
- jednoduché datové typy (ohraničená celá čísla, pole, atd.)
- sítě automatů s hodinami a datovými proměnnými.

## Simulátor

- vyšetřování možných dynamických běhu nějakého systému,
- detekce vad modelů před jeho verifikací,
- umožňuje analýzu záznamů běhů vedoucích k nežádaným stavům.

## Verifikátor modelu

- prověření všech možností dynamického chování modelu,
- kontrola invariantů a živosti prohledáváním stavového prostoru,
- dosažitelnost symbolických stavů reprezentovaných omezeními.

# Komponenty systému <sup>[UPP10]</sup>

## Jazyk popisu

- jazyk nedeterministických podmíněných příkazů
- jednoduché datové typy (ohraničená celá čísla, pole, atd.)
- sítě automatů s hodinami a datovými proměnnými.

## Simulátor

- vyšetřování možných dynamických běhu nějakého systému,
- detekce vad modelů před jeho verifikací,
- umožňuje analýzu záznamů běhů vedoucích k nežádaným stavům.

## Verifikátor modelu

- prověření všech možností dynamického chování modelu,
- kontrola invariantů a živosti prohledáváním stavového prostoru,
- dosažitelnost symbolických stavů reprezentovaných omezeními.



# Komponenty systému <sup>[UPP10]</sup>

## Jazyk popisu

- jazyk nedeterministických podmíněných příkazů
- jednoduché datové typy (ohraničená celá čísla, pole, atd.)
- sítě automatů s hodinami a datovými proměnnými.

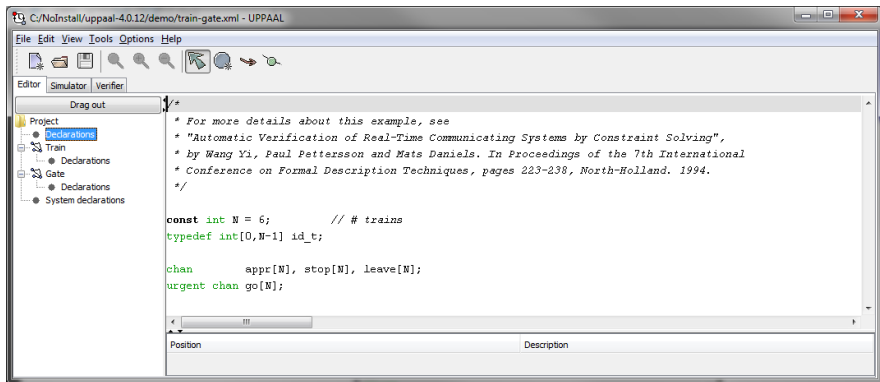
## Simulátor

- vyšetřování možných dynamických běhu nějakého systému,
- detekce vad modelů před jeho verifikací,
- umožňuje analýzu záznamů běhů vedoucích k nežádaným stavům.

## Verifikátor modelu

- prověření všech možností dynamického chování modelu,
- kontrola invariantů a živosti prohledáváním stavového prostoru,
- dosažitelnost symbolických stavů reprezentovaných omezeními.

# Systémový editor <sup>[UPP10]</sup>

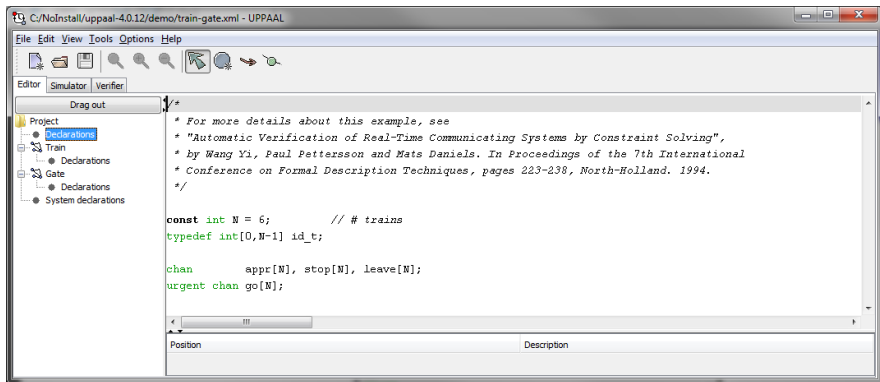


## Editor

- tvorba grafického i textového popisu systémů



# Systémový editor <sup>[UPP10]</sup>

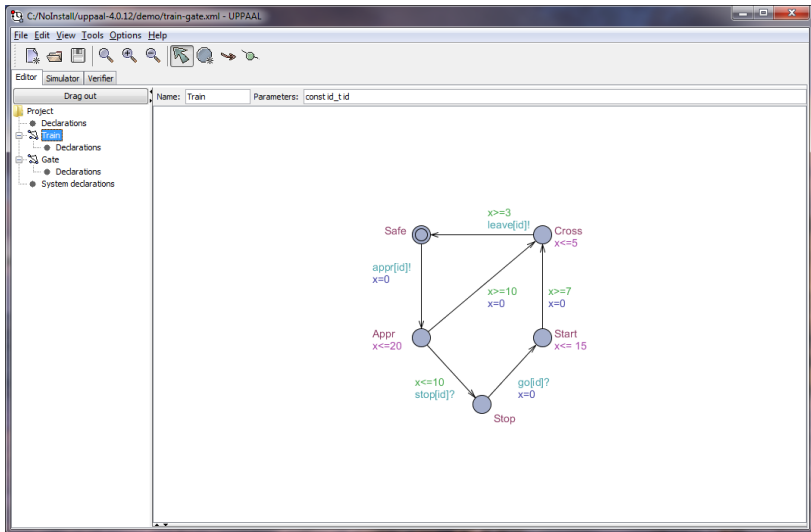


## Editor

- tvorba grafického i textového popisu systémů



# Grafický systémový editor <sup>[UPP10]</sup>



Grafický simulátor <sup>[UPP10]</sup>

## Simulátor

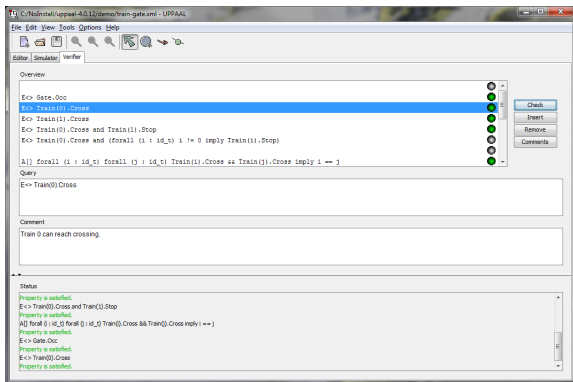
- grafická vizualizace a záznam možného dynamického chování popisu systému,
- sekvence symbolických stavů systému,
- možnost vizualizace trasy generované verifikátorem.

Grafický simulátor <sup>[UPP10]</sup>

## Simulátor

- grafická vizualizace a záznam možného dynamického chování popisu systému,
- sekvence symbolických stavů systému,
- možnost vizualizace trasy generované verifikátorem.

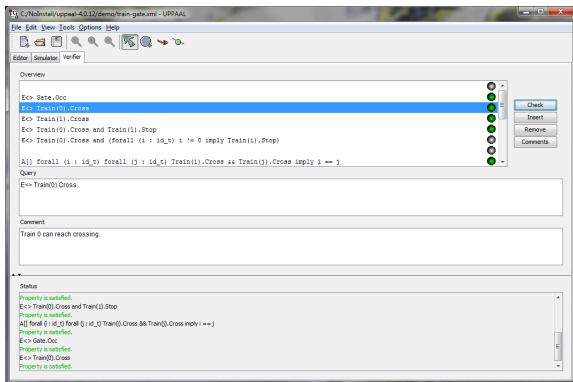
## Verifikátor [UPP10]



## Verifikátor

- Editor specifikace požadavků,
- Stroj verifikátoru modelu
  - automatické ověření živosti a ohraničené živosti pomocí dosažitelnosti v symbolickém stavovém prostoru.

## Verifikátor [UPP10]



## Verifikátor

- Editor specifikace požadavků,
- Stroj verifikátoru modelu
  - automatické ověření živosti a ohraničené živosti pomocí dosažitelnosti v symbolickém stavovém prostoru.



# Outline

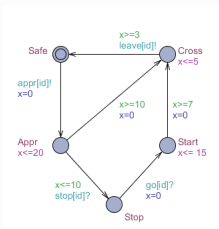
- 1 Úvod
  - Motivace
  - Úvod do verifikace modelů
  - Formální popis
- 2 UPPAAL - Přehled
  - Základní vlastnosti
  - Architektura
- 3 UPPAAL - Vybrané vlastnosti
  - **Systém a proces**
  - Manuál

# Výchozí principy <sup>[UPP09]</sup>

## Model

- Časový automaty
  - konečný stavový automat s hodinami,
  - čas je spojitý,
  - hodiny měří postup času.

## Vzory procesů - Automat



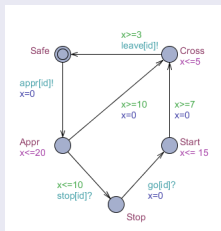
- pozice a hrany,
- symbolické proměnné a konstanty jako parametry,
- lokální proměnné a hodiny,
- daný proces je pak instancí VZORU.

# Výchozí principy <sup>[UPP09]</sup>

## Model

- Časový automaty
  - konečný stavový automat s hodinami,
  - čas je spojitý,
  - hodiny měří postup času.

## Vzory procesů - Automat



- pozice a hrany,
- symbolické proměnné a konstanty jako parametry,
- lokální proměnné a hodiny,
- daný proces je pak instancí VZORU.

# Časový automat <sup>[BDL05]</sup>

## Časový automat

- je šestice  $(L, \ell_0, C, A, E, \mathcal{I})$ , kde
  - $L$  je množina pozic,
  - $\ell_0 \in L$  je počáteční pozice,
  - $C$  je množina hodin.
  - $A$  je množina akcí, ko-akcí a interní  $\tau$ -akce,
  - $E \subseteq L \times A \times B(C) \times 2^C \times L$  je množina hran mezi pozicemi s akcí, stráží a množinou hodin, které se resetují, a
  - $\mathcal{I} : L \rightarrow B(C)$  přiřazuje invarianty k pozicím.

## Příklady

- $y := 0$  ... resetování hodin  $y$ ,
- $press?$  a  $press!$  ... označují akci a ko-akci (zde kanálovou synchronizaci).

# Časový automat <sup>[BDL05]</sup>

## Časový automat

- je šestice  $(L, \ell_0, C, A, E, \mathcal{I})$ , kde
  - $L$  je množina pozic,
  - $\ell_0 \in L$  je počáteční pozice,
  - $C$  je množina hodin.
  - $A$  je množina akcí, ko-akcí a interní  $\tau$ -akce,
  - $E \subseteq L \times A \times B(C) \times 2^C \times L$  je množina hran mezi pozicemi s akcí, stráží a množinou hodin, které se resetují, a
  - $\mathcal{I} : L \rightarrow B(C)$  přiřazuje invarianty k pozicím.

## Příklady

- $y := 0 \dots$  resetování hodin  $y$ ,
- $press?$  a  $press!$   $\dots$  označují akci a ko-akci (zde kanálovou synchronizaci).

# Hodiny časového automatu <sup>[BDL05]</sup>

## Hodiny

- **Ohodnocení hodin** je funkce  $u : C \rightarrow \mathbb{R}_{\geq 0}$  z množiny hodin do nezáporných reálných čísel.
- Nechť  $\mathbb{R}^C$  je množina všech ohodnocení hodin.
- Nechť  $u_0(x) = 0$  pro všechna  $x \in C$ .
- Zápis  $u \in \mathcal{I}(\ell)$  bude znamenat, že  $u$  splňuje  $\mathcal{I}(\ell)$ .
- Z daného stavu je možné provést přechod pomocí *akce* nebo *zpoždění*.

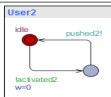
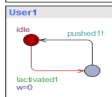
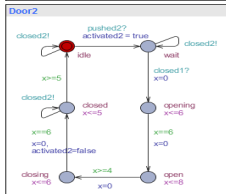
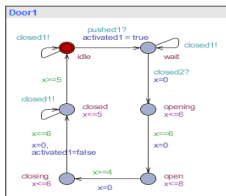


# Sémantika časového automatu [BDL05]

## Sémantika časového automatu

- Nechť  $(L, \ell_0, C, A, E, \mathcal{I})$  je časový automat.
- Sémantika ... přechodový systém s označením  $\langle S, s_0, \rightarrow \rangle$ , kde
- $S \subseteq L \times \mathbb{R}^C$  je množina stavů,
- $s_0 = (\ell_0, u_0)$  je počáteční stav,
- $\rightarrow \subseteq S \times (\mathbb{R}_{\geq 0} \cup A) \times S$  je přechodová relace taková, že
  - $(\ell, u) \xrightarrow{d} (\ell, u + d)$  if  $\forall d' : 0 \leq d' \leq d \implies u + d' \in \mathcal{I}(\ell)$
  - $(\ell, u) \xrightarrow{a} (\ell', u')$  if  $\exists e = (\ell, a, g, r, \ell') \in E$   
 $| e \in g, u' = [r \mapsto 0]u, u' \in \mathcal{I}(\ell')$
- $u + d$  zobrazuje každé hodiny  $x \in C$  na hodnotu  $u(x) + d$ , pro  $d \in \mathbb{R}_{\geq 0}$ ,
- $[r \mapsto 0]u$  označuje ohodnocení hodin, která mapuje každé hodiny  $v \in r$  na 0 a souhlasí s  $u$  nad  $C \setminus r$ .

# Specifikace systému [UPP09]

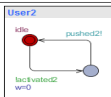
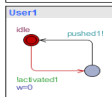
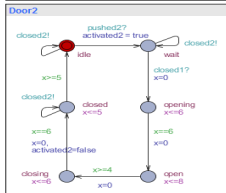
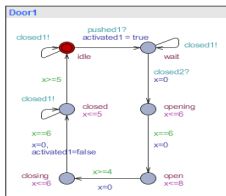


## Systém je kompozice souběžných procesů

- každý proces je modelován jako automat
- automat má množinu pozic,
- změny pozic se dějí pomocí hran/přechodů.
- stav systému je charakterizován pomocí aktuální
  - pozice každého automatu,
  - hodnot proměnných, a
  - stavu hodin.
- přechody je možné řídit pomocí stráží a synchronizací
- stráž je podmínka nad proměnnými a hodinami specifikující, kdy je přechod možný.



## Komunikace procesů [UPP09]



## Synchronizace

- synchronizace je mechanismus kdy dva procesy provedou současný přechod
  - 1 synchronizační kanál  $a$ ,
  - 2 první proces vyvolá přechod zprávou  $a!$
  - 3 druhý proces provede přechod přijetím zprávy  $a?$
- během přechodu je možné přiřadit do proměnných nebo resetovat hodiny.

# Síť časových automatů [BDL05]

## Sada automatů

- Společná množina hodin a akcí.
- $n$  časových automatů  $\mathcal{A}_i = (L_i, \ell_i^0, C, A, E_i, \mathcal{I}_i), 1 \leq i \leq n$
- poziční vektor  $\bar{\ell} = (\ell_1, \dots, \ell_n)$
- společná funkce invariantů  $\mathcal{I}(\bar{\ell}) = \bigwedge_i \mathcal{I}_i(\ell_i)$
- $\bar{\ell}[l'_i/l_i]$  ...  $i$ -tý element  $\ell_i$  vektoru  $\bar{\ell}$  je nahrazen  $l'_i$



# Sémantika sítě časových automatů <sup>[BDL05]</sup>

## Síť časových automatů

- $n$  časových automatů  $\mathcal{A}_i = (L_i, \ell_i^0, C, A, E_i, \mathcal{I}_i)$
- počáteční vektor pozic  $\bar{\ell}^0 = (\ell_1^0, \dots, \ell_n^0)$
- Sémantika ... přechodový systém s označením  $\langle S, s_0, \rightarrow \rangle$ , kde
- $S \subseteq (L_1 \times \dots \times L_n) \times \mathbb{R}^C$  je množina stavů,
- $s_0 = (\bar{\ell}_0, u_0)$  je počáteční stav,
- $\rightarrow \subseteq S \times S$  je přechodová relace taková, že
  - $(\bar{\ell}, u) \xrightarrow{d} (\bar{\ell}, u + d)$  if  $\forall d' : 0 \leq d' \leq d \implies u + d' \in \mathcal{I}(\bar{\ell})$ , a
  - $(\bar{\ell}, u) \xrightarrow{a} (\bar{\ell}[\ell'_i/\ell_i], u')$  if  $\exists \ell_i \xrightarrow{\tau g_i r_i} \ell'_i$   
 $| u \in g, u' = [r \mapsto 0]u, u' \in \mathcal{I}(\bar{\ell}[\ell'_i/\ell_i])$ ,
  - $(\bar{\ell}, u) \xrightarrow{a} (\bar{\ell}[\ell'_j/\ell_j, \ell'_i/\ell_i], u')$  if  $\exists \ell_i \xrightarrow{c?g_i r_i} \ell'_i$  a  $\ell_j \xrightarrow{c!g_j r_j} \ell'_j$   
 $| u \in (g_i \wedge g_j), u' = [r_i \cup r_j \mapsto 0]u, u' \in \mathcal{I}(\bar{\ell}[\ell'_j/\ell_j, \ell'_i/\ell_i])$ ,

# Outline

- 1 Úvod
  - Motivace
  - Úvod do verifikace modelů
  - Formální popis
- 2 UPPAAL - Přehled
  - Základní vlastnosti
  - Architektura
- 3 UPPAAL - Vybrané vlastnosti
  - Systém a proces
  - **Manuál**

# Typy jazyka <sup>[BDL05]</sup>

## Typy

- **Konstanty** ... `const name value`, celočíselná hodnota.
- **Omezené celočíselné hodnoty** ... `int[min, max] name`, výchozí nastavení -32768 až 32768.
- **Pole** ... hodiny, kanály, konstanty, celočíselné proměnné  
`chan c[4]; clock a[2]; int[3,5] u[7];`.
- **Iniciátory** ... nastavení hodnot celočíselných proměnných a polí s celočíselnými proměnnými  
`int i := 2; int k[3] := {1, 2, 3};`.



# Speciální přechody <sup>[BDL05]</sup>

## Řídící elementy

- **Binární synchronizace** ... `chan c`, hrana `c!` a `c?`, nedeterministicky pár.
- **Broadcast synchronizace** ... `broadcast chan c`, jedna hrana `c!` se všemi možnými `c?`, neblokuje.
- **Urgentní synchronizace** ... `urgent chan c`.
  - Zpoždění není dovoleno, pokud je možný přechod na urgentním kanálu.
- **Urgentní pozice** ... Čas systému nemůže plynout, pokud se systém nachází v urgentní pozici.
- **Prováděcí pozice**
  - **Prováděcí stav** ... alespoň jedna z pozic je prováděcí.
  - Prováděcí stav se nemůže zpožďovat.
  - Následující přechod musí zahrnovat jednu výstupní hranu vedoucí z prováděcí pozice.

# Výrazy jazyka <sup>[BDL05]</sup>

## Hodiny, celočíselné proměnné a konstanty

- **Stráž** ... výsledkem je logická hodnota.
- **Synchronizace** ... synchronizační návěští `Expression!` nebo `Expression?` nebo prázdné.  
Výsledkem je kanál.  
Odkazovat může celá čísla, konstanty, kanály.
- **Přiřazení** ... výrazy oddělená čárkou.  
Odkazovat může hodiny, celočíselné proměnné, konstanty.  
Hodinám může přiřadit pouze celočíselné hodnoty.
- **Invariant** ... Konjunkce podmínek tvaru  $x < e$  nebo  $x \leq e$ , kde
  - $x$  je odkaz na hodiny,
  - $e$  se vyčíslí do celého čísla.Odkazovat může hodiny, celočíselné proměnné, konstanty.

# Literatura I

- [BDL05] [Gerd Behrmann, Alexandre David, and Kim G. Larsen](#). A tutorial on UPPAAL, updated 25th october 2005. Technical report, Department of Computer Science, Aalborg University, Denmark, October 2005.
- [Bie08] [Armin Biere](#). Tutorial on model checking, modelling and verification in computer science. In *Proc. 3rd Intl. Conf. on Algebraic Biology (AB'08). Lecture Notes in Computer Science (LNCS)*, volume 5147. Springer, 2008.
- [Cam10] [Alarico Campetelli](#). Analysis techniques: State of the art in industry and research. techreport TUM-I1008, Technische Universität München, April 2010.
- [Če09] [Jiří Čermák](#). Porovnání modelovacích schopností verifikačních nástrojů. Master's thesis, Masarykova univerzita, Fakulta informatiky, Brno, 2009.
- [Hol06] [Lukáš Holík](#). Rozhodnutelnost v temporálních logikách. Master's thesis, Masarykova univerzita, Fakulta informatiky, Brno, 2006.
- [UPP09] [UPPAAL 4.0: Small tutorial, November 2009](#).
- [UPP10] [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.

