Úloha 2: Časované automaty

1. příklad

- **Automat $\mathcal{A}_1$ obsahuje *zeno běh*.** Např. běh $\rho = (A; x = 0, y = 0) \xrightarrow{\mathtt{a}_1} (B; x = 0, y = 0) \xrightarrow{\mathtt{a}_2} (C; x = 0, y = 0) \xrightarrow{\mathtt{a}_4} (A; x = 0, y = 0) \xrightarrow{\mathtt{a}_1} \dots$ je zeno běh, protože je to *časově konvergentní nekonečný běh* ($Exectime(\rho) = 0$), který obsahuje *nekonečné množství diskrétních kroků*. **Neplatí zde také *podmínka neexistence zeno běhu***, protože pro *řídící cyklus* $A \xrightarrow{g_1,\mathtt{a}_1,R_1} B \xrightarrow{g_2,\mathtt{a}_2,R_2} C \xrightarrow{g_4,\mathtt{a}_4,R_4} A$ neexistují hodiny, pro které by alespoň jeden krok cyklu vyžadoval běh času. Jinými slovy, $\nexists\, x \in \mathcal{C} : \exists\, i \in \{1, 2, 4\} : \exists\, c \in \mathbb{N}^+ : \nu(x) < c \Rightarrow \nu(x) \not\models g_i$. □

- **Automat $\mathcal{A}_1$ obsahuje *timelock*. Běh vedoucí do timelocku je např. následující: $(A; x = 0, y = 0) \xrightarrow{10} (A; x = 10, y = 10)$.** *Konfigurace* $c = (A; x = 10, y = 10)$ je timelock, protože $Paths_{div}(c) = \emptyset$. Z této konfigurace už není možné přejít do žádného jiného místa *diskrétním krokem*, protože přechod do místa B je podmíněn predikátem $x \leq 1$ a přechod do místa C je podmíněn predikátem $1 < x < 10$. Tyto nejsou splněny, protože $x = 10$. Je zde pouze možné provádět *nekonečné množství časových kroků*, které ale vždy konvergují k číslu 15, protože je v místě A *invariant* $y < 15$. Žádný *časově divergentní běh* tedy z této konfigurace není možné provést. □

2. příklad

Dále v tomto příkladu budou uvažovány *časované automaty* definovány následovně:
$\mathcal{A} = (Loc, Act, \mathcal{C}, \hookrightarrow, Loc_0, Inv, AP, L, Loc_{acc})$, kde:

- Loc je konečná množina míst,
- Act je konečná množina událostí,
- $\mathcal{C}$ je konečná množina hodin,
- $\hookrightarrow \subseteq Loc \times CC(\mathcal{C}) \times Act \times 2^{\mathcal{C}} \times Loc$ je konečná množina přechodů,
  - $CC(\mathcal{C}) = \{\bigwedge G \mid G \subseteq ACC(\mathcal{C})\}$ je množina podmínek,
  - $ACC(\mathcal{C}) = \{x \bowtie c \mid x \in \mathcal{C} \wedge c \in \mathbb{N}\}$ je množina *atomických podmínek*, kde $\bowtie \in \{<, \leq, =, \geq, >\}$,
- $Loc_0 \subseteq Loc$ je množina počátečních míst,
- $Inv \subseteq Loc \times CC(\mathcal{C})$ je konečná množina *invariantů*,
- AP je konečná množina *atomických výroků*,
- $L \subseteq Loc \times 2^{AP}$ je ohodnocení míst,
- $Loc_{acc} \subseteq Loc$ je množina koncových míst.

Dále jsou také uvažovány *jazyky časovaných automatů* $\mathcal{L}(\mathcal{A}) = \{w \mid w \textbf{ je přijato automatem } \mathcal{A}\}$ nad *konečnými slovy* $w \in (Act \times \mathbb{R}^+)^*$.

Důkaz uzavřenosti časovaných automatů vůči operaci *sjednocení*:

- Mějme dva jazyky časovaných automatů $L_{\mathcal{A}_1}$ a $L_{\mathcal{A}_2}$. Dokažme, že jejich sjednocení $L_{\mathcal{A}_1} \cup L_{\mathcal{A}_2}$ je opět jazyk časovaných automatů.

- Protože $L_{\mathcal{A}_1}$ a $L_{\mathcal{A}_2}$ jsou jazyky časovaných automatů, tak existují časované automaty $\mathcal{A}_1$ a $\mathcal{A}_2$, které přijímají jazyky $L_{\mathcal{A}_1}$ a $L_{\mathcal{A}_2}$, tj. $L(\mathcal{A}_1) = L_{\mathcal{A}_1}$ a $L(\mathcal{A}_2) = L_{\mathcal{A}_2}$.
- Sestrojením časovaného automatu $\mathcal{A}_\cup$, který bude přijímat jazyk $L_{\mathcal{A}_1} \cup L_{\mathcal{A}_2}$, tj. $L(\mathcal{A}_\cup) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$, dokážeme, že jazyk $L_{\mathcal{A}_1} \cup L_{\mathcal{A}_2}$ je jazyk časovaných automatů a že operace sjednocení je uzavřena na časovaných automatech.

Algoritmus sestrojení časovaného automatu $\mathcal{A}_\cup$:

Vstup: Časovaný automat $\mathcal{A}_1 = (Loc_1, Act_1, \mathcal{C}_1, \hookrightarrow_1, Loc_{0_1}, Inv_1, AP_1, L_1, Loc_{acc_1})$ a časovaný automat $\mathcal{A}_2 = (Loc_2, Act_2, \mathcal{C}_2, \hookrightarrow_2, Loc_{0_2}, Inv_2, AP_2, L_2, Loc_{acc_2})$, kde bez újmy na obecnosti předpokládáme, že $Loc_1 \cap Loc_2 = \emptyset$.

Výstup: Časovaný automat $\mathcal{A}_\cup = (Loc, Act, \mathcal{C}, \hookrightarrow, Loc_0, Inv, AP, L, Loc_{acc})$, kde $L(\mathcal{A}_\cup) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$.

Metoda:

1. $Loc = Loc_1 \cup Loc_2$.
2. $Act = Act_1 \cup Act_2$.
3. $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$.
4. $\hookrightarrow = \hookrightarrow_1 \cup \hookrightarrow_2$.
5. $Loc_0 = Loc_{0_1} \cup Loc_{0_2}$.
6. $Inv = Inv_1 \cup Inv_2$.
7. $AP = AP_1 \cup AP_2$.
8. $L = L_1 \cup L_2$.
9. $Loc_{acc} = Loc_{acc_1} \cup Loc_{acc_2}$.

□

Důkaz uzavřenosti časovaných automatů vůči operaci *konkatenace*:

- Mějme dva jazyky časovaných automatů $L_{\mathcal{A}_1}$ a $L_{\mathcal{A}_2}$. Dokažme, že jejich konkatenace $L_{\mathcal{A}_1}.L_{\mathcal{A}_2}$ je opět jazyk časovaných automatů.
- Protože $L_{\mathcal{A}_1}$ a $L_{\mathcal{A}_2}$ jsou jazyky časovaných automatů, tak existují časované automaty $\mathcal{A}_1$ a $\mathcal{A}_2$, které přijímají jazyky $L_{\mathcal{A}_1}$ a $L_{\mathcal{A}_2}$, tj. $L(\mathcal{A}_1) = L_{\mathcal{A}_1}$ a $L(\mathcal{A}_2) = L_{\mathcal{A}_2}$.
- Sestrojením časovaného automatu $\mathcal{A}_\circ$, který bude přijímat jazyk $L_{\mathcal{A}_1}.L_{\mathcal{A}_2}$, tj. $L(\mathcal{A}_\circ) = L(\mathcal{A}_1).L(\mathcal{A}_2)$, dokážeme, že jazyk $L_{\mathcal{A}_1}.L_{\mathcal{A}_2}$ je jazyk časovaných automatů a že operace konkatenace je uzavřena na časovaných automatech.

Algoritmus sestrojení časovaného automatu $\mathcal{A}_\circ$:

Vstup: Časovaný automat $\mathcal{A}_1 = (Loc_1, Act_1, \mathcal{C}_1, \hookrightarrow_1, Loc_{0_1}, Inv_1, AP_1, L_1, Loc_{acc_1})$ a časovaný automat $\mathcal{A}_2 = (Loc_2, Act_2, \mathcal{C}_2, \hookrightarrow_2, Loc_{0_2}, Inv_2, AP_2, L_2, Loc_{acc_2})$, kde bez újmy na obecnosti předpokládáme, že $Loc_1 \cap Loc_2 = \emptyset$.

Výstup: Časovaný automat $\mathcal{A}_\circ = (Loc, Act, \mathcal{C}, \hookrightarrow, Loc_0, Inv, AP, L, Loc_{acc})$, kde $L(\mathcal{A}_\circ) = L(\mathcal{A}_1).L(\mathcal{A}_2)$.

Metoda:

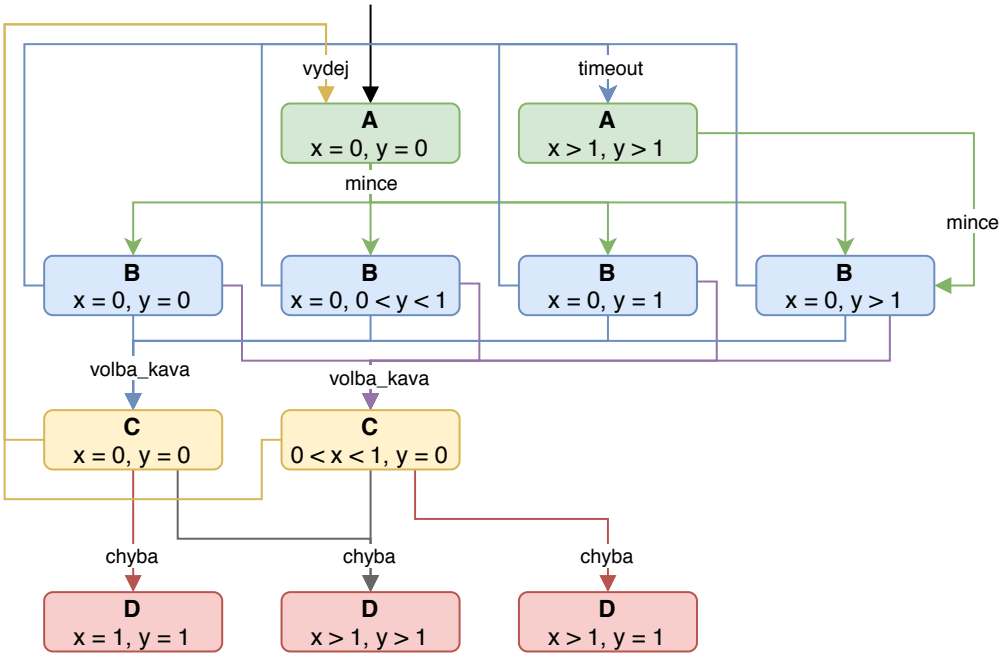
1. $Loc = Loc_1 \cup Loc_2$.
2. $Act = Act_1 \cup Act_2$.

3. $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$.
4. $\hookrightarrow \subseteq Loc \times CC(\mathcal{C}) \times Act \times 2^{\mathcal{C}} \times Loc$ definována tak, že: $\forall l, l' \in Loc : \forall g \in CC(\mathcal{C}) : \forall a \in Act : \forall R \in 2^{\mathcal{C}} : (l, g, a, R, l') \in \hookrightarrow \Leftrightarrow (l, g, a, R, l') \in \hookrightarrow_1 \cup \hookrightarrow_2 \vee (l, g, a, R \cup \mathcal{C}, l') \in \hookrightarrow \Leftrightarrow (\exists l'' \in Loc_{acc_1} : (l, g, a, R, l'') \in \hookrightarrow_1 \wedge \exists l''' \in Loc_2 : \exists g' \in CC(\mathcal{C}) : \exists a' \in Act : \exists R' \in 2^{\mathcal{C}} : (l', g', a', R', l''') \in \hookrightarrow_2 \wedge l' \in Loc_{0_2})$.
5. $Loc_0 = Loc_{0_1}$.
6. $Inv = Inv_1 \cup Inv_2$.
7. $AP = AP_1 \cup AP_2$.
8. $L = L_1 \cup L_2$.
9. $Loc_{acc} = Loc_{acc_2}$.

□

3. příklad

***Abstrakce založená na regionech* automatu $\mathcal{A}_2$ (bez nedostupných stavů):**



- **Stav, ve kterém platí predikát $error$, je dostupný.** Existuje běh $(A; x = 0, y = 0) \xrightarrow{\texttt{mince}} (B; x = 0, y = 0) \xrightarrow{\texttt{volba\_kava}} (C; x = 0, y = 0) \xrightarrow{\texttt{1,chyba}} (D; x = 1, y = 1)$, kde $A \in Loc_0 \wedge error \in L(D)$. Dostupnost tohoto stavu s tímto predikátem je také vidět v regionové abstrakci.
- **Tvrzení $\mathcal{A}_2 \models \exists\,(run\ U^{<2}\ error)$ platí.** Platí, že $Int_{\mathcal{A}_2} \subseteq Sat(\phi)$, kde $\phi \equiv \exists\,(run\ U^{<2}\ error)$, $Int_{\mathcal{A}_2} = \{s = (A; x = 0, y = 0)\}$ a $s \models \phi$, protože existuje cesta $\pi \in Path_{div}(s)$, pro kterou platí $\pi \models run\ U^{<2}\ error$. Cesta π může být např. následující $\pi = (A; x = 0, y = 0) \xrightarrow{\texttt{mince}} (B; x = 0, y = 0) \xrightarrow{\texttt{volba\_kava}} (C; x = 0, y = 0) \xrightarrow{\texttt{1,chyba}} (D; x = 1, y = 1) \xrightarrow{\tau} (D; x = 1 + \tau, y = 1 + \tau) \xrightarrow{\tau} \ldots; \pi \models run\ U^{<2}\ error$, protože existuje časový okamžik $t = 1$ z intervalu $[0, 2)$, ve kterém platí formule $error$ ($error \in L(D)$) a pro libovolný časový okamžik menší než t platí $run \vee error$. Toto lze vypozorovat v regionové abstrakci.

- **Tvrzení $(B; x = 0, y = 0) \models \forall\, (run\, U^{<2}\, init)$ neplatí.** Neplatí totiž, že pro každou cestu $\pi \in Path_{div}(s)$ platí $\pi \models \phi$, kde $s = (B; x = 0, y = 0)$ a $\phi \equiv run\, U^{<2}\, init$. Např. pro cestu $\pi = (B; x = 0, y = 0) \xrightarrow{10} (B; x = 10, y = 10) \xrightarrow{\tau} (B; x = 10 + \tau, y = 10 + \tau) \xrightarrow{\tau} \ldots$ neplatí $\pi \models \phi$, protože zde neexistuje takový časový okamžik t z intervalu $[0, 2)$, ve kterém by platila formule $init$. Toto lze vypozorovat v regionové abstrakci.