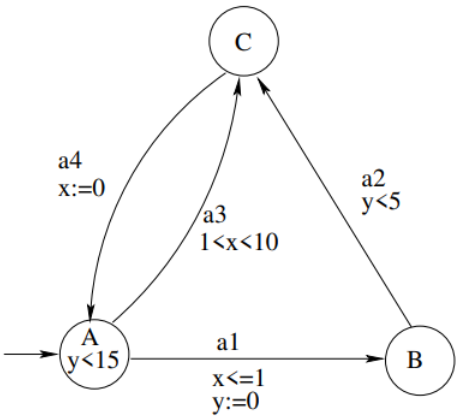
xmarci10

1	2	3	Σ

Úloha č. 2

(Termín odovzdania 16.04.2020)

Všetky definície a vety použité v tejto práci sú z prednášok predmetu MBA [1, 2].



Obr. 1: Časovaný automat $\mathcal{A}_1$

Príklad 1. Uvažujme automat $\mathcal{A}_1$ na obrázku 1:

(a) Obsahuje tento automat *zeno beh* ? Dokážte, alebo vyvráťte.

(b) Obsahuje tento automat *timelock* ? Ak áno uveďte beh vedúci do timelocku.

(*2 body*)

Riešenie

(1a) Pri riešení tejto úlohy budeme vychádzať z definície hovoriacej o tom, čo to zeno beh vlastne je a taktiež z lemmy hovoriacej o podmienkach pre neexistenciu zeno behu:

Definícia 6 [1]: Nech $\mathcal{A}$ je časovaný automat a $\rho = c_1 \xrightarrow{s_1} c_2 \xrightarrow{s_2} c_3 \xrightarrow{s_3} \dots$ jeho *časovo konvergentný nekonečný beh*. ρ nazveme **zeno behom**, ak ρ obsahuje nekonečné množstvo diskrétnych krokov.

Lemma 7 (Podmienka neexistencie zeho behu) [1]: Časový automat $\mathcal{A}$, kde pre každý riadiaci cyklus

$$l_0 \xrightarrow{g_1,a_1,R_1} l_1 \xrightarrow{g_2,a_2,R_2} \dots \xrightarrow{g_n,a_n,R_n} l_n = l_0$$

existujú hodiny $x \in \mathcal{C}$, také že:

- $x \in R_i$ pre nejaké $0 < i \leq n$ (hodiny x sú aspoň raz resetované)
- Existuje konštanta $c \in \mathbb{N}^+$ taká že $\nu(x) < c \rightarrow \nu(x) \not\models g_i$ pre nejaké $0 < i \leq n$ (aspoň jeden krok cyklu vyžaduje beh času)

V časovanom automate na obrázku 1 existujú dva riadiace cykly, ktoré majú tvar požadovaný Lemmou 7:

cyklus 1 $A \to C \to A$

- hodiny x sú resetované $(x := 0)$
- podmienka $1 < x \Rightarrow c = 1$

cyklus 2 $A \to B \to C \to A$

- hodiny x aj hodiny y sú resetované $(x := 0,\ y := 0)$
- neexistuje konštanta c podľa Lemmy 7 $\Rightarrow$ žiadny krok cyklu si nevyžaduje beh času

Podmienka neexistencie zeno behu **nie je splnená**.

Príklad zeno behu v časovanom automate z obrázku 1:

$$(A; [0,0]) \xrightarrow{a_1} (B; [0,0]) \xrightarrow{a_2} (C; [0,0]) \xrightarrow{a_4} (A; [0,0]) \xrightarrow{a_1} \dots$$

Táto sekvencia diskrétnych krokov sa môže opakovať donekonečna bez nutnosti behu času a tak tento beh spĺňuje Definíciu 6.

(1b) Definícia 5 (Timelock) [1]: Nech $c = (l, \nu)$ je konfigurácia časovaného automatu $\mathcal{A}$. $Paths_{div}(c)$ potom označuje množinu časovo divergentných behov zo stavu c. Konfiguráciu c nazveme **timelock** ak $Paths_{div}(c) = \emptyset$.

ÁNO zadaný automat z obrázku 1 **obsahuje timelock**. Príklad timelock-u je napríklad konfigurácia $(A; [14, 14])$. Prvá odchádzajúca hrana zo stavu A je $A \xrightarrow{a_1, x \leq 1, y:=0} B$, no tento krok nemožno vykonať kvôli guard-u $x \leq 1$. Ďalšia odchádzajúca hrana je $A \xrightarrow{a_3, 0<x<10} C$, ktorú ale taktiež nemožno kvoli jej guardu vykonať. Zároveň nám guard v lokácii A nedovoľuje neobmedzený beh času a tak $Paths_{div}((A; [14, 14])) = \emptyset$.

Príklad behu vedúceho do timelock-u.

$$(A; [0,0]) \xrightarrow{14.0} (A; [14, 14])$$

Príklad 2. Dokážte, že jazyky časovaných automatov[1] sú uzatvorené voči operácii zjednotenia a konkatenácie.

(*2 body*)

Riešenie

2. Dôkazy oboch vlastností budú ukázané metódou, kedy to, že jazyky časovaných automatov sú uzatvorené voči nejakej operácii ukážeme konštrukciou automatu $\mathcal{A}$ takého, že $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \oplus \mathcal{L}(\mathcal{A}_2)$, kde $\oplus \in \{\cdot, \cup\}$.

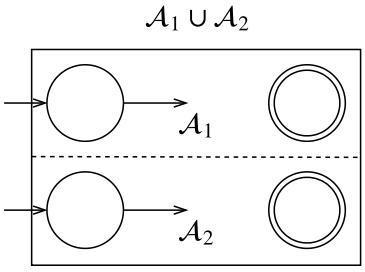
Teorém 2.1. *Jazyky časovaných automatov sú uzatvorené voči operácii zjednotenia.*

Dôkaz. Nech $\mathcal{A}_1 = (Loc_1,\ Act,\ \mathcal{C}_1,\ \hookrightarrow_1,\ Loc_{0_1},\ Inv_1,\ AP_1,\ L_1,\ Loc_{acc_1})$ a $\mathcal{A}_2 = (Loc_2,\ Act,\ \mathcal{C}_2,\ \hookrightarrow_2,\ Loc_{0_2},\ Inv_2,\ AP_2,\ L_2,\ Loc_{acc_2})$ sú časované automaty prijímajúce jazyky $\mathcal{L}(\mathcal{A}_1)$ a $\mathcal{L}(\mathcal{A}_2)$. Bez ujmy na obecnosti uvažujme, že $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$ a $Loc_1 \cap Loc_2 = \emptyset$.

Nech $\mathcal{A} = (Loc,\ Act,\ \mathcal{C},\ \hookrightarrow,\ Loc_0,\ Inv,\ AP,\ L,\ Loc_{acc})$ je časovaný automat definovaný nasledovne:

- $Loc = Loc_1 \cup Loc_2$
- $Loc_{acc} = Loc_{acc_1} \cup Loc_{acc_2}$
- $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$
- $Inv = Inv_1 \cup Inv_2$
- $\hookrightarrow = \hookrightarrow_1 \cup \hookrightarrow_2$
- $AP = AP_1 \cup AP_2$
- $Loc_0 = Loc_{0_1} \cup Loc_{0_2}$
- $L = L_1 \cup L_2$

Z definície automatu $\mathcal{A}$ je vidieť, že jazyk $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$.



Obr. 2: Zjednotenie dvoch časovaných automatov. Oba časované automaty budú bežať paralelne. V prípade, že časované slovo $w \in \mathcal{L}(\mathcal{A}_1)$, bude prijaté automatom $\mathcal{A}_1$ a ak naopak platí, že $w \in \mathcal{L}(\mathcal{A}_2)$ príjme ho automat $\mathcal{A}_2$.

Teorém 2.2. *Jazyky časovaných automatov sú uzatvorené voči operácii konkatenácie.*

Dôkaz. Nech $\mathcal{A}_1 = (Loc_1,\ Act,\ \mathcal{C}_1,\ \hookrightarrow_1,\ Loc_{0_1},\ Inv_1,\ AP_1,\ L_1,\ Loc_{acc_1})$ a $\mathcal{A}_2 = (Loc_2,\ Act,\ \mathcal{C}_2,\ \hookrightarrow_2,\ Loc_{0_2},\ Inv_2,\ AP_2,\ L_2,\ Loc_{acc_2})$ sú časované automaty prijímajúce jazyky $\mathcal{L}(\mathcal{A}_1)$ a $\mathcal{L}(\mathcal{A}_2)$. Bez ujmy na obecnosti uvažujme, že $Loc_1 \cap Loc_2 = \emptyset$.

Nech $\mathcal{A} = (Loc,\ Act,\ \mathcal{C},\ \hookrightarrow,\ Loc_0,\ Inv,\ AP,\ L,\ Loc_{acc})$ je časovaný automat definovaný nasledovne:

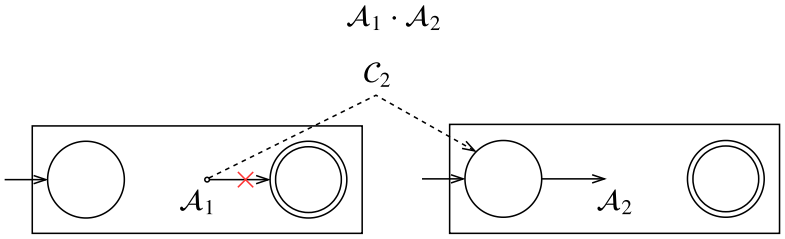
[1] Uvažujte jazyky nad konečnými slovami s množinou koncových stavov Loc_{acc}

- $Loc = Loc_1 \cup Loc_2$
- $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$
- $Loc_0 = Loc_{0_1}$
- $Loc_{acc} = Loc_{acc_2}$
- $Inv = Inv_1 \cup Inv_2$
- $AP = AP_1 \cup AP_2$
- $L = L_1 \cup L_2$
- $\hookrightarrow := \hookrightarrow_1 \cup \hookrightarrow_2$
  $\cup \{(q, g, a, r \cup \mathcal{C}_2, i) : i \in Loc_{0_2} \wedge \exists f \in Loc_{acc_1} : (q, g, a, r, f) \in \hookrightarrow_1\}$
  $\setminus \{(q, g, a, r, f) \in \hookrightarrow_1 : f \in Loc_{acc_1}\}$

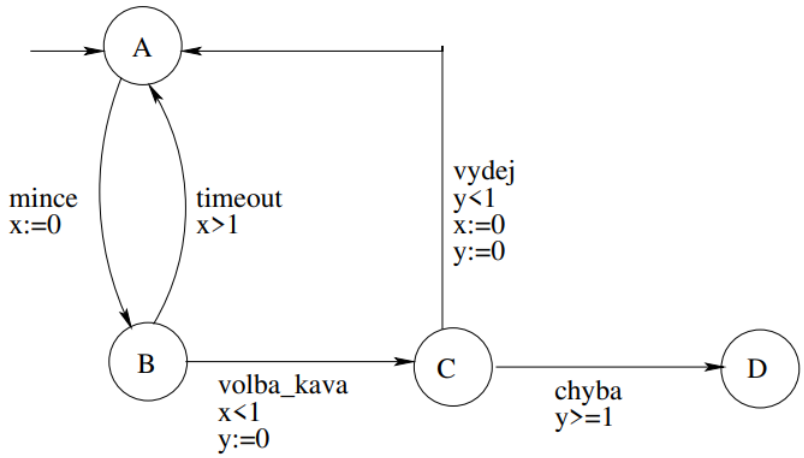
Z definície automatu $\mathcal{A}$ je vidieť, že:

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2) = \{uv : u \in \mathcal{L}(\mathcal{A}_1) \wedge v \in \mathcal{L}(\mathcal{A}_2)\}$$

.



Obr. 3: Konkatenácia dvoch časovaných automatov. Automat $\mathcal{A}_1$ prečíta reťazec u, ale namiesto toho aby prešiel do koncového stavu, tak prejde do počiatočného stavu automatu $\mathcal{A}_2$ pričom resetuje všetky hodiny $\mathcal{C}_2$. Automat $\mathcal{A}_2$ potom príjme reťazec v a akceptuje.



Obr. 4: Časovaný automat $\mathcal{A}_2$

Príklad 3. Uvažujme automat $\mathcal{A}_2$ na obrázku 4 s množinou atomických predikátov $AP = \{init, error, run\}$ a funkciou L definovanou nasledovne:

$$L(A) = \{init, run\}, L(D) = \{error\}, L(B) = L(C) = \{run\}$$

(a) Zostavte abstrakciu založenú na regiónoch (stačí zostrojiť iba stavy dostupné z počiatočnej konfigurácie).

(b) Rozhodnite, či je dostupný stav v ktorom platí predikát error.

(c) Rozhodnite či platí $\mathcal{A}_2 \models \exists(run\ U^{<2}\ error)$.

(d) Rozhodnite či platí $(B, x = y = 0) \models \forall(run\ U^{<2}\ init)$.

Svoje tvrdenia zdôvodnite.

(*4 body*)

Riešenie

(3a) Abstrakcia založená na regiónoch pre časovaný automat $\mathcal{A}_2$ (viz Obr. 4) je zobrazená na obrázku 5. Zobrazené sú iba stavy dostupné z počiatočnej konfigurácie. Pri konštrukcii boli taktiež uvažované aj kroky, ktoré obsahujú nulový posun času.

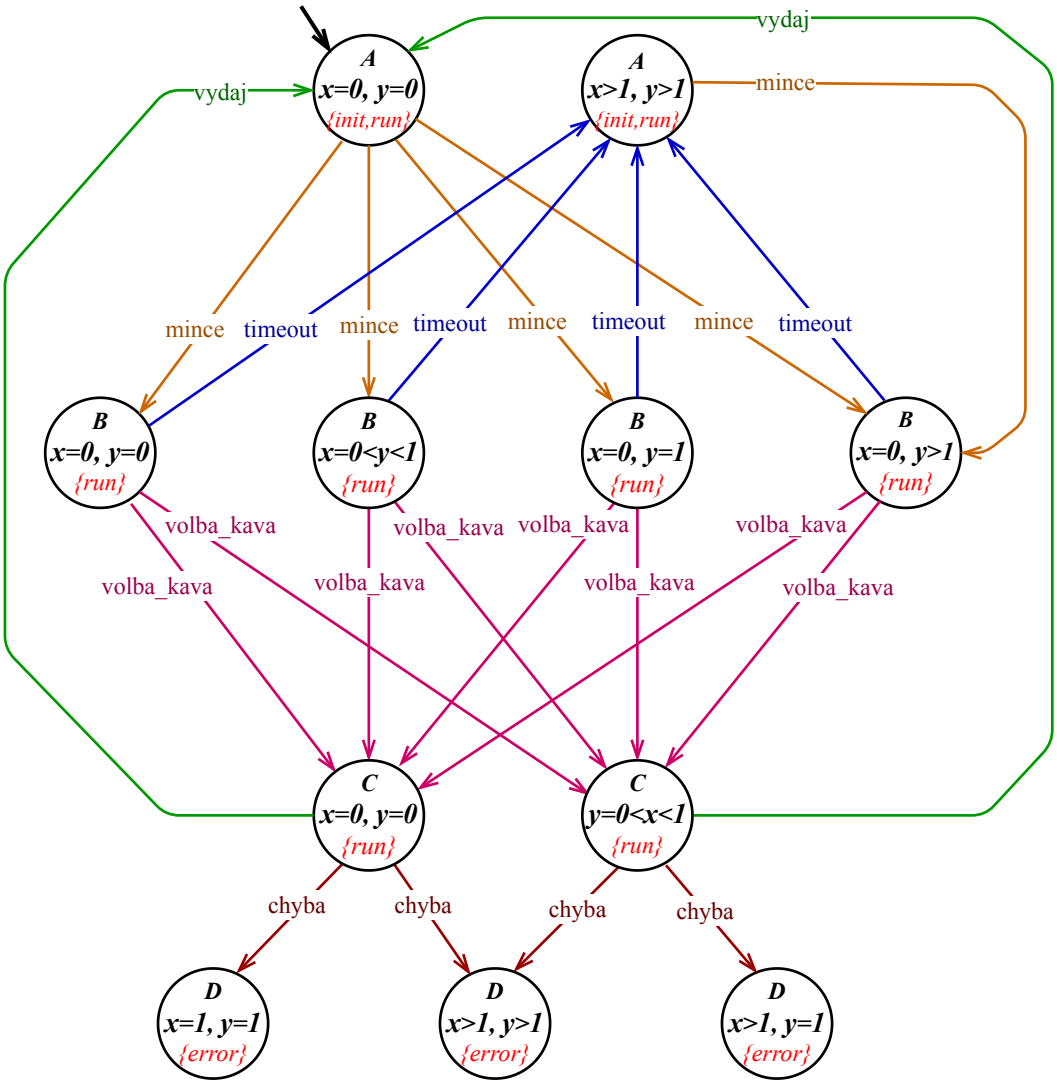
(3b) Túto vlastnosť môžme overiť na základe zkonštruovanej regiónovej abstrakcie. Na obrázku 5 je možné vidieť, že sú dosiahnuteľné až tri stavy v ktorých platí predikát *error*.

Príklad behu vedúceho do stavu spĺňajúceho predikát *error* (čiže do stavu D) môže vyzerať nasledovne:

$$(A; [0,0]) \xrightarrow{mince} (B; [0,0]) \xrightarrow{volba\_kava} (C; [0,0]) \xrightarrow{1.0,chyba} (\mathbf{D}; [\mathbf{1}, \mathbf{1}])$$

Na overenie tejto úlohy bol taktiež použitý nástroj Uppaal[2]. Model časovaného automatu $\mathcal{A}_2$ v nástroji Uppaal je zobrazený na obrázku 6.

[2] Uppaal je integrované prostredie na modelovanie, validáciu a verifikáciu systémov pracujúcich v reálnom čase, ktoré sú modelované ako siete časovaných automatov. Viac informácií viz: `http://www.uppaal.org/`



Obr. 5: Abstrakcia založená na regiónoch pre časovaný automat z obrázku 4.

Pri overovaní, či je stav v ktorom platí predikát *error* (stav D) dostupný sme použili Uppall `Verifier`, kde sme zadali dotaz v tvare `E<>Process.D`[3]. Tento dotaz reprezentuje Uppall notáciu formuly $\exists \diamond Process.D$, ktorá môže byť voľne preložená ako „je možné dosiahnuť stav `D` v automate `Process` ?".

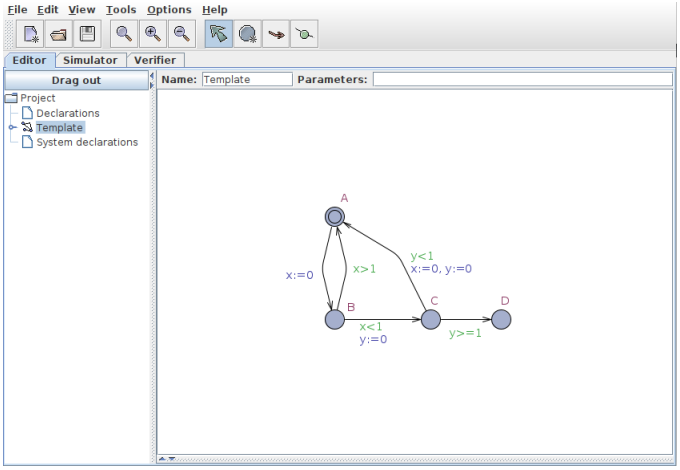
Výsledok z nástroja Uppall je možné vidieť na obrázku 7. Z obrázku je viditeľné, že spomínaná formula je platná a teda stav D (v ktorom platí predikát *error*) **je dosiahnuteľný**.

(3c) Definícia 5 [2]: Povieme, že automat splňuje formulu ϕ ($A \vDash \phi$) ak:

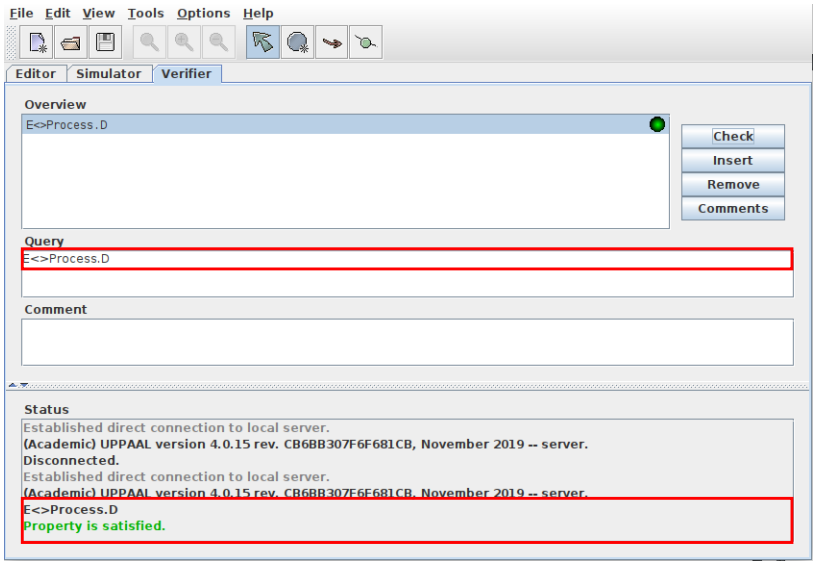
$$Init_{\mathcal{A}} \subseteq Sat(\phi)$$

kde $Init_{\mathcal{A}}$ je *množina počiatočných konfigurácií*: $Init_{\mathcal{A}} = \{(l, 0^{|\mathcal{C}|}) | l \in Loc_0)\}$ a $Sat(\phi)$ je *množina konfigurácií spĺňajúcich formulu* ϕ:
$Sat(\phi) = \{(l, \nu) | l \in Loc, \nu \in \mathbb{R}^{|\mathcal{C}|}_{\geq 0}, (l, \nu) \vDash \phi\}$

[3] `Process` je meno automatu $\mathcal{A}_2$ v prostredí Uppaal.



Obr. 6: Automat $\mathcal{A}_2$ z obrázku 4 v nástroji Uppaal. Modré výrazy znázorňujú reset hodín a zelené výrazy sú *guard*-y jednotlivých prechodov.



Obr. 7: Výsledok overenia úlohy **(3b)** v prostredí Uppaal.

Keďže automat $\mathcal{A}_2$ má iba jednu počiatočnú konfiguráciu môžme overované tvrdenie previesť na tvar:

$$(A, x = y = 0) \vDash \exists(run\ U^{<2}\ error)$$

Teraz z Definície 2 [2] hovoriacej o relácii splniteľnosti vieme, že konfigurácia s spĺňa formulu $\exists\phi$ ak $\pi \vDash \phi$ pre **nejakú** cestu $\pi \in Paths_{div}(s)$.

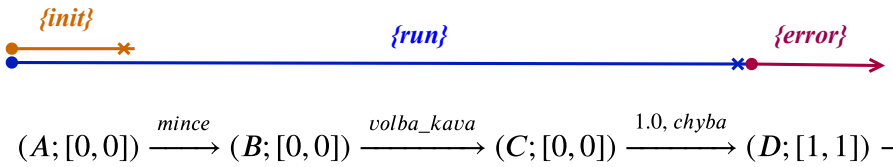
Takže aby zadané tvrdenie bolo pravdivé musí v časovanom automate $\mathcal{A}_2$ existovať časovo divergentná cesta z počiatočnej konfigurácie, ktorá spĺňa formulu $(run\, U^{<2}\, error)$.

Podľa Definície 3 z [2] teda musí platiť, že (1) existuje časový okamih $t \in \langle 0, 2)$ v ktorom platí atomická podmienka *error* a zároveň (2) pre ľubovoľný časový okamih menší ako t platí formula $run \vee error$.

Príklad takéhoto behu je na obrázku 8. V tomto behu je časový okamih t v ktorom platí atomická podmienka *error* rovný 1, čiže patrí do intervalu $\langle 0, 2)$, a tým je

splnená prvá podmienka z Definície 3.

Zároveň platí aj druhá podmienka Definície 3, keďže pre ľubovoľný časový okamih menší ako 1 platí atomická podmienka *run*.



Obr. 8: Dôkaz, že časovaný automat $\mathcal{A}_2$ **spĺňa** formulu $\exists(run\ U^{<2}\ error)$. Krúžok značí začiatok platnosti atomickej podmienky, krížik znamená koniec platnosti atomickej podmienky a šípka trvanie jej platnosti.

(3d) Podobne ako v predchádzajúcom bode, na to aby sme rozhodli či platí

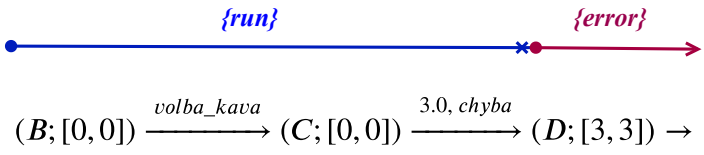
$$(B, x = y = 0) \vDash \forall(run\ U^{<2}\ init)$$

použijeme Definíciu 2 z [2]. Tentokrát pre zmenu použijeme časť hovoriacu o tom, že aby konfigurácia $(B, x = y = 0)$ spĺňovala formulu $\forall(run\ U^{<2}\ init)$ tak musí platiť, že $\pi \vDash (run\ U^{<2}\ init)$ pre **všetky** cesty $\pi \in Paths_{div}(B, x = y = 0)$.

To, že zadané tvrdenie nie je pravdivé ukážeme formou dokazovania protipríkladom. To znamená, že nájdeme časovo divergentný beh vedúci z konfigurácie $(B; [0, 0])$, ktorý nespĺňa formulu $(run\ U^{<2}\ init)$.

Príklad takéhoto behu je na obrázku 9. Z tohto behu môžme vidieť, že už prvá podmienka z Definície 3 (uvedenej vyššie) nie je splnená keďže neexistuje časový okamih z intervalu $\langle 0, 2)$ v ktorom by platila atomická formula *init*. Z toho vyplýva, že zadané tvrdenie **neplatí**.

To že všetky divergentné cesty z konfigurácie $(B; [0, 0])$ nespĺňajú formulu $(run\ U^{<2}\ init)$ môžme vidieť aj z regiónovej abstrakcie na obrázku 5. Môžme si všimnúť, že ak sa časovaný automat $\mathcal{A}_2$ raz dostane zo stavu $(B; [0, 0])$ do niektorého stavu spĺňajúceho predikát *error* už nikdy sa nedostane do stavu v ktorom by platil predikát *init*.



Obr. 9: Dôkaz, že konfigurácia $(B, x = y = 0)$ **nespĺňa** formulu $\forall(run\ U^{<2}\ init)$.

Literatúra

[1] Rogalewicz, A.: *Časované automaty - Abstrakce založená na regionech* (prednáška MBA). marec 2020.
URL `https://www.fit.vutbr.cz/study/courses/MBA/private/prednasky/TA-1.pdf`

[2] Rogalewicz, A.: *Časované automaty - Logika TCTL* (prednáška MBA). marec 2020.
URL `https://www.fit.vutbr.cz/study/courses/MBA/private/prednasky/TA-2.pdf`