Analýza systémů založená na modelech
2022/2023

Domácí úloha 1

Pavel Šesták (xsesta07)

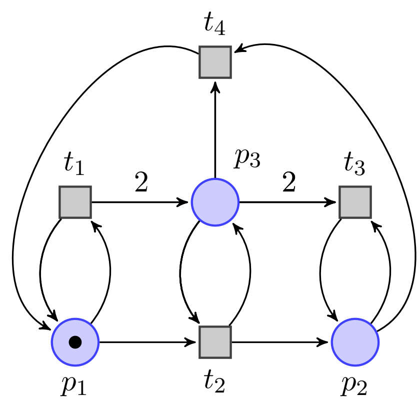
Brno, 23. března 2023

Obsah

Seznam obrázků

Seznam tabulek

MBA 2023
Úloha 1



Obrázek 1: P/T-síť.

Příklad 1. Uvažujte P/T-síť z Obrázku 1:

1. Sestrojte strom dosažitelných značení. S jeho využitím určete a odůvodněte, zda:
   (a) je P/T-síť bezpečná, omezená,
   (b) jsou značení $M_1 = (0, 1, 101)$, $M_2 = (1, 1, 102)$, $M_3 = (1, 0, 102)$ pokrytelná a dosažitelná (pokud je to možné na základě stromu rozhodnout),
   (c) může/nemůže být P/T-síť živá.
2. Vypočtěte P-invarianty. S jejich využitím určete a odůvodněte, zda
   (a) je P/T-síť striktně konzervativní, je konzervativní vzhledem k nějakému váhovému vektoru (pokud ano, tak uveďte příklad takového vektoru).
3. Vypočtěte T-invarianty.
   (a) Co lze s vypočtených T-invariantů určit o živosti sítě a proč?
   (b) Určete a odůvodněte, zda jsou vektory $v_1 = (4, 3, 1, 3)$ a $v_2 = (1, 1, 1, 1)$ T-invarianty a zda jsou realizovatelné.

Oprašte lineární algebru vypočtěte invarianty ručně. Dostupné nástroje můžete použít pro kontrolu.

4 bodů

Příklad 2. Dokažte, že ze stromu dos. značení není možné určit živost sít'ě (pan učitel to na cvičení mohl mít špatně). *1 bod*

Příklad 3. V nástroji Netlab nebo Pipe (Netlab je spolehlivější) modelujte synchronizaci *dvou* procesů Lamportovým algoritmem, jak je popsán na `https://en.wikipedia.org/wiki/Lamport%27s_bakery_algorithm#Pseudocode`. Modelujte systém odpovídající paralelnímu spuštění Thread(1) a Thread(2). Použijte modelovací techniky, kdy místa v síti indikují vykonávané programové řádky a hodnoty proměnných. Pole délky 2 modelujte jako dvojici proměnných. Předpokládejte, že integer má hodnotu maximálně 1, takže celočíselné sčítaní je sčítání ve zbytkové třídě $\mathbb{Z}_2$. Snažte se o co největší přehlednost modelu, použijte textových označení míst a přechodů.

Proveďte dostupné analýzy a interpretujte výsledky. Zejména zodpovězte následující otázky a zdůvodněte odpovědi.

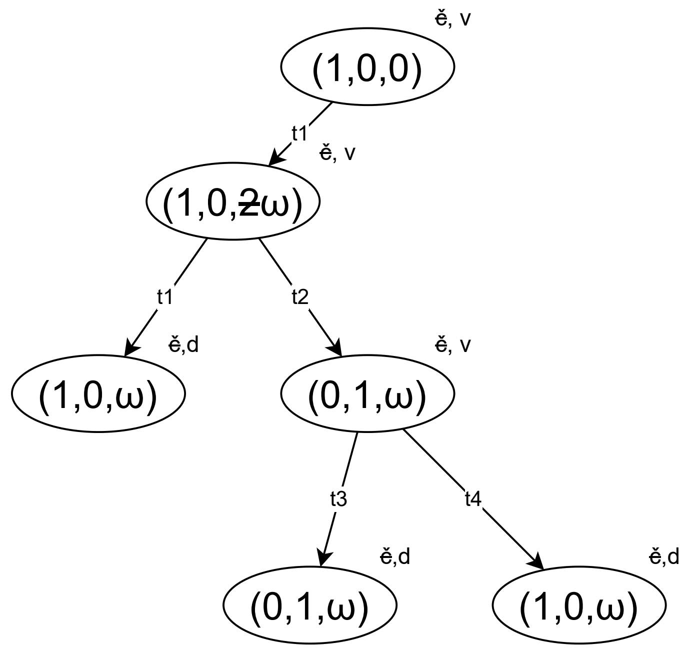
1. Garantuje protokol vzájemné vyloučení (t.j., procesy nemohou být současně v kritické sekci)?
2. Je možné, že protokol garantuje nemožnost uváznutí?

Model by měl být spustitelný v počítačových učebnách. Odevzdejte jej do informačního systému do termínu odevzdání úkolu.

5 bodů

1 Úloha 1

1.1 Strom dosažitelných značení



Obrázek 1: Strom dosažitelných značení

1.1.1 Je P/T-síť bezpečná, omezená?

Podle stromu dosažitelných značení síť není bezpečná, jelikož obsahuje přechod *t1*, který generuje značky do místa *p3* a v stromu dosažitelných značení se tedy objevuje znak ω, který reprezentuje možnost do nekonečna přidávat značky. Bezpečnost sítě umožňuje v každém místě maximálně jednu značku.

Tato P/T-síť není ani omezená, jelikož by muselo $\exists C \in \mathbb{N}$ takové, že $\forall p \in P : M(p) < C$. Tato vlastnost lze zase zkontrolovat z stromu dosažitelného značení. Strom dosažitelných značení by nesměl obsahovat žádný symbol ω.

1.1.2 Jsou zadaná značení pokrytelná a dosažitelná?

- $M_1 =$ (0,1,101)
- $M_2 =$ (1,1,102)
- $M_3 =$ (1,0,102)

Ze stromu dosažitelných značení lze určit pokrytí, nicméně dosažitelnost nelze obecně ze stromu pokrytí určit. Značení M_1 je pokrytelné jelikož je pokryto ve značení (0,1,ω). Značení M_2 není pokryté žádným uzlem stromu a nedá se tedy do takového stavu dostat. Značení M_3 je pokryté značením (1,0,ω).

1.1.3 Je P/T-síť živá?

Živost sítě se nedá určit ze stromu pokrytí, nicméně zadaná P/T-síť není živá. Přechod t_4 není živý. Pro běh se sekvencí přechodů: $t_1^n t_2 t_3^n$ se stanou všechny přechody neproveditelné a P/T-síť se zasekne se značením (0,1,0).

1.2 Vypočtěte P invarianty

K výpočtu P invariantů budeme potřebovat matici změn N.

$$N = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 0 & -1 & 0 & +1 \\ 0 & +1 & 0 & -1 \\ +2 & 0 & -2 & -1 \end{pmatrix} \end{array}$$

Matici N transponujeme a odstraníme lineárně závislé řádky.

$$N^T_{reduced} = \begin{pmatrix} 0 & 0 & 2 \\ -1 & +1 & 0 \\ 1 & -1 & -1 \end{pmatrix}$$

Dále řešíme soustavu rovnic $N^T_{reduced} * i = 0$. Řešíme například Gaussovou eliminační metodou.

$$eliminated = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$result = \begin{array}{c} p1 \\ p2 \\ p3 \end{array} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Podle výsledného vektoru je zřejmý vztah mezi místem p_1 a p_2. Počet značek v těchto dvou místech je vždy roven jedné. Obecně se jedná o T-invariant (k,k,0), kde $k \in N^+$.

1.2.1 Analýza konzervativnosti sítě

Zadaná P/T-síť není striktně konzervativní, jelikož v počátečním stavu je součet značek ve všech místech roven jedné, ale již po provedení přechodu t_1 je součet značek v P/T-síti roven třem. Je zřejmé, že do stavu p_3 si mohu negenerovat libovolný počet značek.
Zadaná P/T-síť je konzervativní vzhledem k váhovému vektoru w = (1,1,0) a platí:

$$\forall M \in [M_0\rangle : \sum_{i=1}^{n} w_i * M(p_i) = \sum_{i=1}^{n} w_i * M_0(p_i) = 1$$

1.3 Vypočtěte T invarianty

T invariant se počítá obdobně jako P invariant, jen počítá s netransponovanou maticí změn.

$$N = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 0 & -1 & 0 & +1 \\ 0 & +1 & 0 & -1 \\ +2 & 0 & -2 & -1 \end{pmatrix} \end{array}$$

$$eliminated = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{pmatrix} 2 & 0 & -2 & -1 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$results = \begin{array}{c} t_1 \\ t_2 \\ t_3 \\ t_4 \end{array} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 0 \\ 2 \end{pmatrix}$$

Dva výsledné vektory jsou T-invarianty v této P/T-síti.

1.3.1 Vztah T-invariantů k živosti sítě

Existence T-invariant v omezené P/T-síti zaručuje živost sítě. Pro neomezené sítě toto tvrzení neplatí a existence T-invariantu je nutnou, nikoli však postačující podmínkou živosti PT-sítě. Pro omezené sítě jsme tedy díky T-invariantům schopni rozhodovat o živosti. Naše síť není omezená, tudíž o živosti za pomocí T-invariantů nelze rozhodnout.

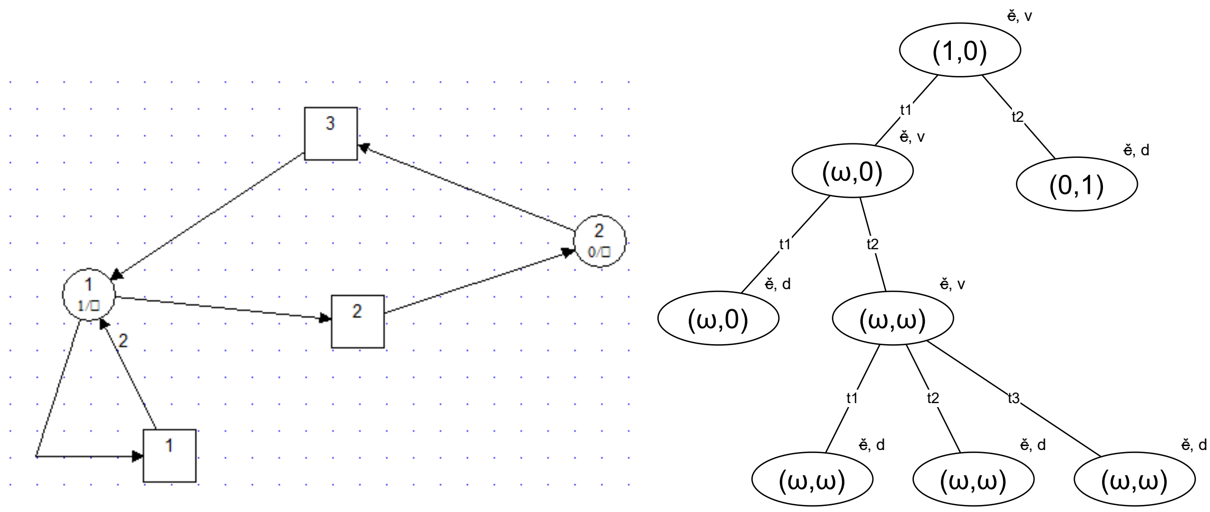
1.3.2 Analýza zadaných T-invariantů

- v1 = (4,3,1,3)
- v2 = (1,1,1,1)

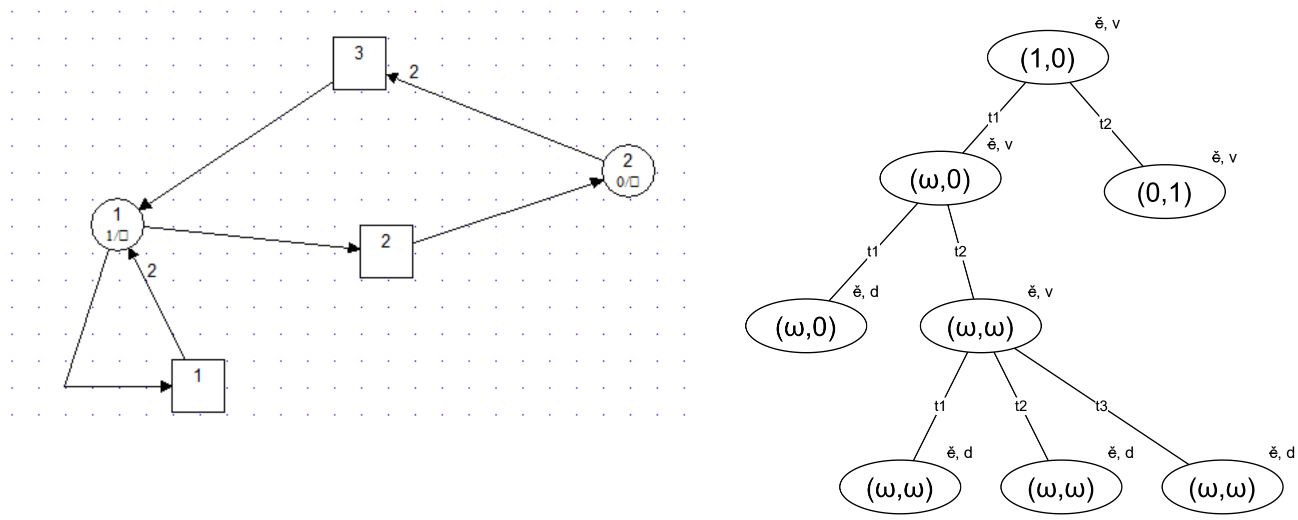
Vektor v1 není T-invariantem zadané P/T-sitě se specifikovaným počátečním značením. Zadaná sekvence přechodů je sice proveditelná, ale dostaneme se do značení $M_x = (1, 0, 3)$, což je různé od $M_0 = (1, 0, 0)$.
Vektor v2 není T-invariantem zadané P/T-sitě se specifikovaným počátečním značení, jelikož přechod t_1 přidá do místa p_3 právě dvě značky a pro realizaci přechodů t_3 a t_4 potřebujeme v místě p_3 alespoň tři značky, takže po provedení $t_1 t_2 t_3$ se dostaneme do deadlocku ve stavu (0,1,0).
Jednodušeji se dá říci, že se nejedná o lineárně závislé vektory na T-invariantech vypočtených v sekci 1.3. Jelikož se nejedná o T-invarinaty, tak nemůžou být ani realizovatelné.

2 Úloha 2

Pro důkaz, že ze stromu dosažitelných značení P/T-sítě není možné určit živost sítě použijeme důkaz sporem.



Obrázek 2: Petriho síť PT1 a její strom dosažitelných značení



Obrázek 3: Petriho síť PT2 a její strom dosažitelných značení

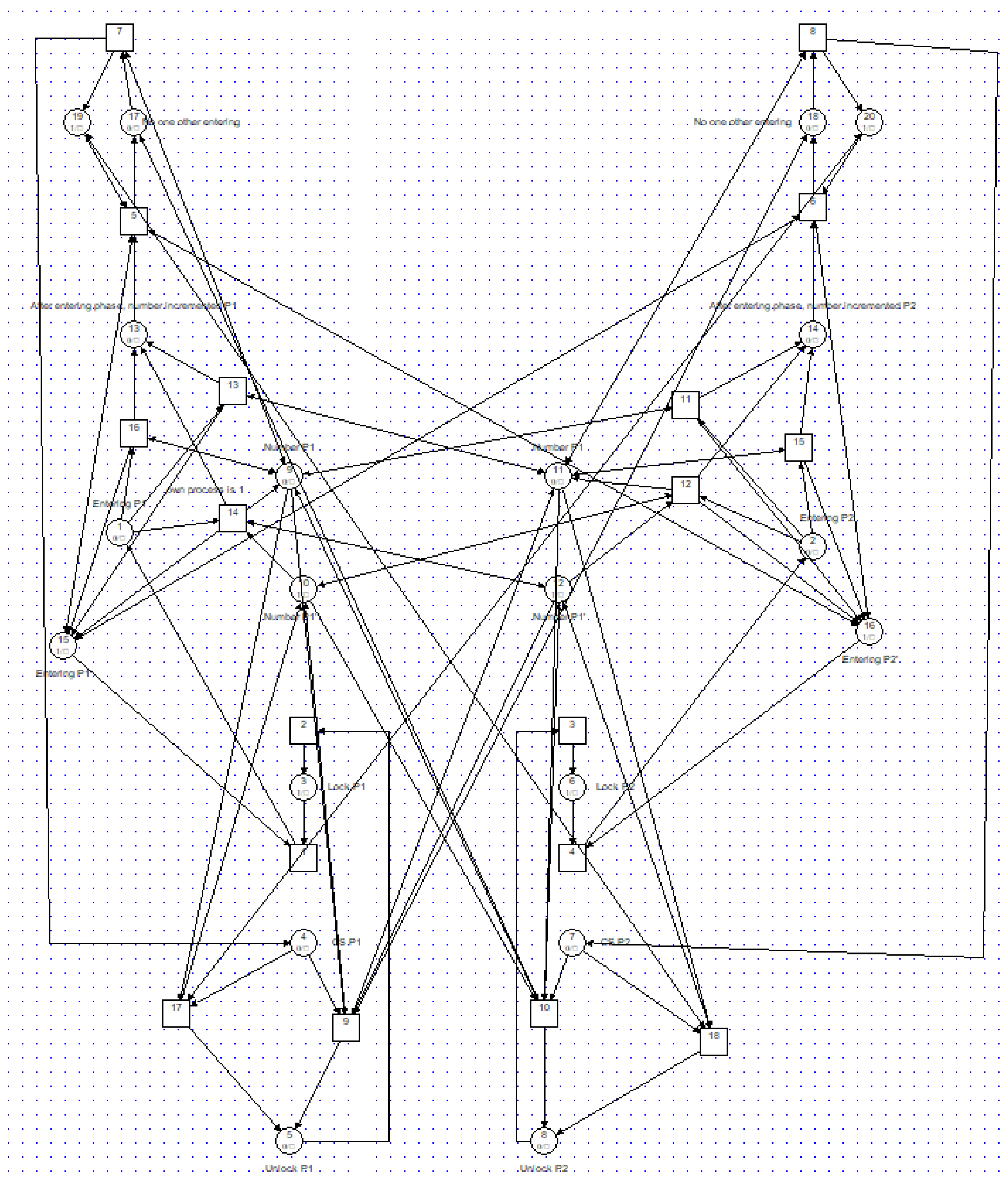
Stromy těchto dvou rozdílných Petriho sítí byly generovány pomocí DFS zleva do prava. Oba stromy jsou identické a pokud strom dosažitelných značení dokáže rozhodovat živost sítě, pak síť PT1 a PT2 musí být obě živé nebo neživé.

Když zanalyzujeme síť P1 pak vidíme, že značky mohou kolovat mezi místy p_1 a p_2 pomocí sekvence přechodů $(t_2t_3)*$. Druhá možnost co lze v síti provést, je dogenerovat více značek pomocí přechodu T1. Je tedy zjevné, že síť PT1 je živá.

Jelikož stromy pokrytí jsou stejné, pak i PT2 by měla být živá. Ovšem jakmile se rozhodneme provést přechod t_2 a v p_1 bude pouze jedna značka, tak se nám již nepodaří dostat ze značení (0,1) a síť tedy nemůže být živá. (SPOR S PŘEDPOKLADEM)

3 Úloha 3

Petriho síť modelována v nástroji Netlab. Rozhraní Netlabu je již starší záležitostí, nicméně až na grafickou chybu při vytváření hran mezi místy a přechody a občasný pád u počítání reachability jsem na závažnější chybu nenarazil. Analýzy, které Netlab nabízí plně dostatečné pro potřeby tohoto úkolu. Dokonce jsem byl příjemně překvapen, že jsem díky analyzátorům narazil na chybu v návrhu, kterou se mi při ručním testování modelu nepovedlo odhalit. Uvítal bych export do SVG nebo PDF, jelikož pomocí nabízeného exportu do PNG se mi nepovedlo dosáhnout použitelných výsledků a obrázek modelu v tomto dokumentu byl pořízen pomocí print screenu. Práce se systémem je jinak intuitivní a příjemná. K dokonalosti pro dnešní dobu už by byl pouze webový kabátek, který by eliminoval problémy s instalací a stahováním desktopové aplikace.



Obrázek 4: Petriho síť pro Lamportův synchronizační algoritmus

3.1 Garantuje protokol vzájemné vyloučení?

Ano, protokol garantuje vzájemné vyloučení. Ověřit to lze pomocí stromu dosažitelných značení, naše kritické sekce jsou v místech p_4 a p_7. Ve vygenerovaném stromě se nevyskytuje dostupné značení kdy by v p_4 a p_7 zároveň byla značka.

3.2 Garantuje protokol nemožnost uváznutí?

Lamportův synchronizační protokol garantuje nemožnost uváznutí. Podle Netlab analyzátoru se v mém modelu Lamportova algoritmu žádný deadlock nenachází a síť by tedy měla být živá.