

Prostorová složitost, Shrnutí

Jakub Bulín

Adapted from the Czech lecture slides by Marta Vomlelová.

Translation, minor modifications, and any errors are mine.

KTIML MFF UK

Organizační záležitosti

- Přednáška:
 - ▶ moodle <https://dl1.cuni.cz/course/view.php?id=5119>
 - ★ login jako do SIS
 - ▶ video nahrávky přednášek z roku 2019
<https://is.mff.cuni.cz/prednasky/prednaska/NTIN071/1>
 - ★ login jako do SIS
 - ★ poslední dvě přednášky jsou nové, nejsou tam.
- Cvičení:
 - ▶ vyzkoušíte si prakticky sestrojit automaty a gramatiky
 - ▶ zažijete příklady, což je něco jiného, než je přečíst,
 - ▶ potřebujete zápočet, který udělují **výhradně** cvičící.
- Zkouška:
 - ▶ **Zápočet je nutnou podmínkou účasti na zkoušce** (kromě předtermínů).
 - ▶ Písemná příprava a ústní část
 - ▶ Porozumění látce + schopnost formalizace
 - ★ Orientace v Chomského hierarchii, automatech, gramatikách, (ne)determinismu,
 - ★ Napište definici, formulujte větu, popište ideu důkazu, algoritmus,
 - ★ zařaďte jazyk do Chomského hierarchie a svou odpověď dokažte.
- Komunikace v konzultačních hodinách: Úterý 14:00 v S303.
 - ▶ Jindy bez záruky. Na maily se snažím odpovídat, na dotazy po přednášce také.

Požadavky ke zkoušce

- **Zápočet je nutnou podmínkou účasti na zkoušce.**
 - Zkouška sestává z písemné přípravy a ústní části.
 - **Písemná příprava** bude sestávat ze dvou až tří otázek, které korespondují syllabu přednášky, ověřují schopnosti získané na cvičení a znalost definic, vět a algoritmů z přednášky.
- ! Po dobu písemné přípravy musí být veškeré přinesené poznámky, přípravy, mobily, počítače apod. uloženy v uzavřeném batohu.** V případě opomenutí poznámek na židli, stole, otevřeném batohu apod. je zkouška okamžitě hodnocena 'nevyhověl' a student v ní dále nepokračuje.
- **Požadavky ústní části** Ústní část bude vycházet z písemné přípravy, zpravidla budete dotázáni na vysvětlení-zdůvodnění-příklady k tvrzením v písemné části. Ústní část může být doplněna otázkou v rozsahu syllabu přednášky s písemnou přípravou nesouvisející.

- J.E. Hopcroft, R. Motwani, J.D. Ullman: *Introduction to Automata Theory, Languages, and Computations*, Addison–Wesley
 - M. Sipser: Introduction to the Theory of Computation, Cengage Learning, 2013
 - M. Chytíl: *Automaty a gramatiky*, SNTL Praha, 1984
 - ⇒ moodle <https://dl1.cuni.cz/course/view.php?id=5119>
 - ▶ kde jsou tyto slajdy
 - ⇒ kde je dotazník k textové verzi slajdů
 - ▶ moodle testy (které ale netestují zdůvodnění a důkazy).
- ⇒ cvičení.

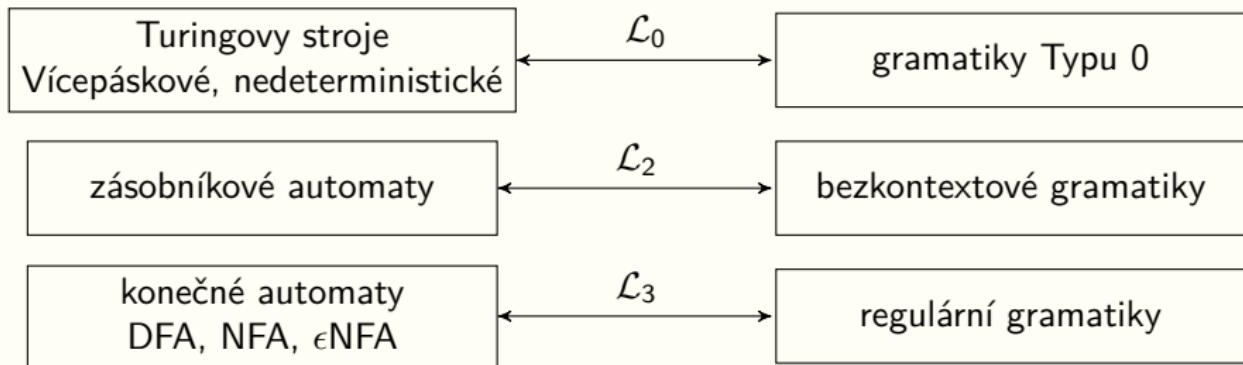
Pohled do historie

- Počátky
 - ▶ první formalizace pojmu algoritmus Ada, Countess of Lovelace 1852
 - ▶ intenzivněji až s rozvojem počítačů ve druhé čtvrtině 20. století
 - ▶ co stroje umí a co ne?
 - ▶ Church, Turing, Kleene, Post, Markov
- Polovina 20. století
 - ▶ neuronové sítě (1943)
 - ▶ konečné automaty (Finite Automata) (Kleene 1956)
neuronové sítě \approx FA)
- 60. léta 20. století
 - ▶ gramatiky (Chomsky)
 - ▶ zásobníkové automaty
 - ▶ formální teorie konečných automatů.
- 80. léta 20. století
 - ▶ popularita tříd časové a prostorové složitosti.



- Osvojit si abstraktní model výpočetních zařízení,
- vnímat, jak drobné změny v definici vedou k velmi odlišným třídám,
- zažít skutečnost algoritmicky nerozhodnutelných problémů,
- rychlý úvod do složitosti $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$.

Automaty a gramatiky – dva způsoby popisu

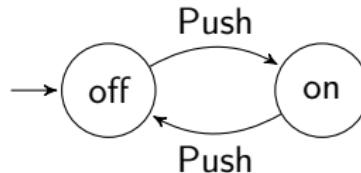


Praktické využití

- Zamyšlení nad korektností programu, algoritmu, překladače,
- zpracování přirozeného jazyka,
- překladače:
 - ▶ lexikální analýza,
 - ▶ syntaktická analýza,
- návrh, popis, verifikace hardware
 - ▶ integrované obvody
 - ▶ stroje
 - ▶ automaty
- realizace pomocí software
 - ▶ hledání výskytu slova v textu (grep)
 - ▶ verifikace systémů s konečně stavy.

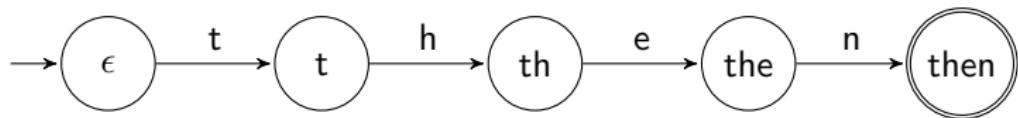
- Návrh a verifikace integrovaných obvodů.

Konečný automat modelující spínač on/off .



- Lexikální analýza

Konečný automat rozpoznávající slovo then.



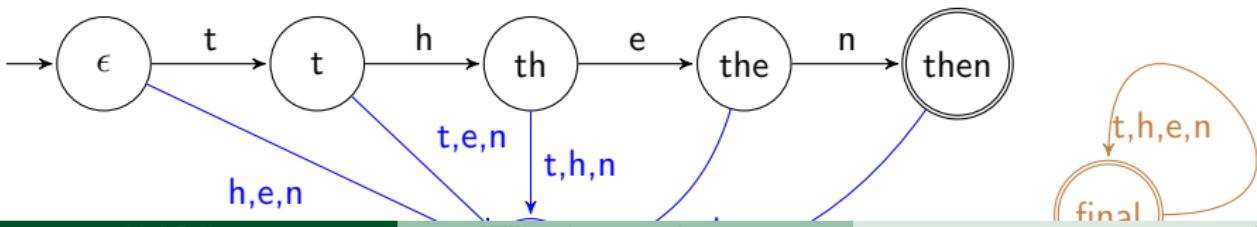
Definice 1.1: Deterministický konečný automat

Deterministický konečný automat (DFA) $A = (Q, \Sigma, \delta, q_0, F)$ sestává z:

- ① konečné množiny **stavů**, zpravidla značíme Q
- ② konečné neprázdné množiny **vstupních symbolů (abecedy)**, značíme Σ
- ③ **přechodové funkce**, zobrazení $Q \times \Sigma \rightarrow Q$, značíme δ , která bude reprezentovaná hranami grafu (na hraně seznam $\subset \Sigma$) nebo tabulkou
- ④ **počátečního stavu** $q_0 \in Q$, vede do něj šipka 'odnikud',
- ⑤ a **množiny koncových (přijímajících) stavů** (*final states*) $F \subseteq Q$, označených dvojitým kruhem či šípkou 'ven'.

Úmluva: Pokud pro některou dvojici stavu a písmene není definovaný přechod, přidáme nový stav *fail* a přechodovou funkci doplníme na totální přidáním šipek do *fail*.

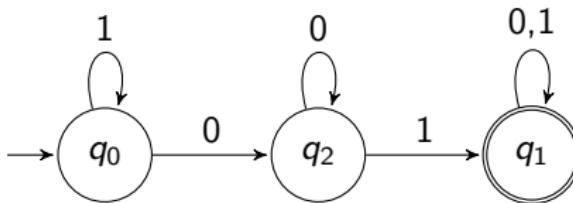
Pokud je množina F prázdná, a je vyžadovaná neprázdná, přidáme do ní i Q nový stav *final* do kterého vedou jen přechody z něj samého $\forall s \in \Sigma: \delta(\text{final}, s) = \text{final}$.



Example 1.1

Automat A přijímající $L = \{x01y : x, y \in \{0, 1\}^*\}$.

- Stavový diagram (graf) Automat $A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$.



tabulka

- - ▶ řádky: stavy + přechody
 - ▶ sloupce: písmena vstupní abecedy

δ	0	1
$\rightarrow q_0$	q_2	q_0
$*q_1$	q_1	q_1
q_2	q_2	q_1

Abeceda, slova, jazyky

Definice 1.2: Slovo, $\epsilon, \lambda, \Sigma^*, \Sigma^+$, jazyk

Mějme neprázdnou množinu symbolů Σ .

- **Slovo, řetězec** je konečná (i prázdná) posloupnost symbolů $s \in \Sigma$, **prázdné slovo** se značí ϵ nebo λ .
- **Množinu všech slov v abecedě Σ** značíme Σ^* ,
- množinu všech neprázdných slov v značíme Σ^+ .
- **jazyk** $L \subseteq \Sigma^*$ je množina slov v abecedě Σ .

Definice 1.3: operace zřetězení, mocnina, délka slova

Nad slovy Σ^* definujeme operace:

- **zřetězení slov** $u.v$ nebo uv
- **mocnina** (počet opakování) u^n ($u^0 = \epsilon$, $u^1 = u$, $u^{n+1} = u^n.u$)
- **délka slova** $|u|$ ($|\epsilon| = 0$, $|auto| = 4$).
- **počet výskytů** $s \in \Sigma$ ve slově u značíme $|u|_s$ ($|zmrzlina|_z = 2$).

Rozšířená přechodová funkce

Definice 1.4: rozšířená přechodová funkce

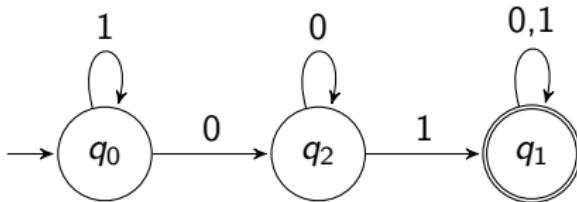
Mějme přechodovou funkci $\delta : Q \times \Sigma \rightarrow Q$.

Rozšířenou přechodovou funkci $\delta^* : Q \times \Sigma^* \rightarrow Q$ (tranzitivní uzávěr δ) definujeme induktivně:

- $\delta^*(q, \epsilon) = q$
- $\delta^*(q, wx) = \delta(\delta^*(q, w), x)$ pro $x \in \Sigma, w \in \Sigma^*$.

Pozn. Pokud se v textu objeví δ aplikované na slova, míní se tím δ^* .

$$\delta^*(q_0, 1100) = q_2, \delta^*(q_0, 110011111111001) = q_1$$



Jazyky rozpoznatelné konečnými automaty

Definice 1.5: jazyky rozpoznatelné DFA, regulární jazyky

- **Jazykem rozpoznávaným (akceptovaným, přijímaným)**

deterministickým konečným automatem $A = (Q, \Sigma, \delta, q_0, F)$ nazveme jazyk $L(A) = \{w \mid w \in \Sigma^* \text{ & } \delta^*(q_0, w) \in F\}$.

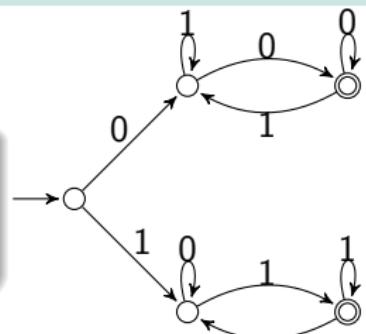
- Slovo w je **přijímáno** automatem A , právě když $w \in L(A)$.

- Jazyk L je **rozpoznatelný** konečným automatem, jestliže existuje konečný automat A takový, že $L = L(A)$.

- Třídu jazyků rozpoznatelných konečnými automaty označíme \mathcal{F} , nazveme **regulární jazyky**.

Example 1.2 (regulární jazyk)

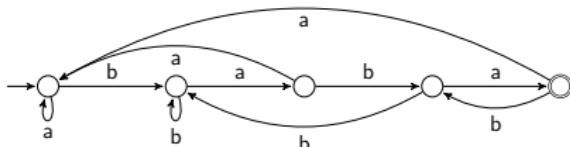
- $L = \{w \mid w = xux, w \in \{0, 1\}^*, x \in \{0, 1\}, u \in \{0, 1\}^*\}$.



Příklady regulárních jazyků

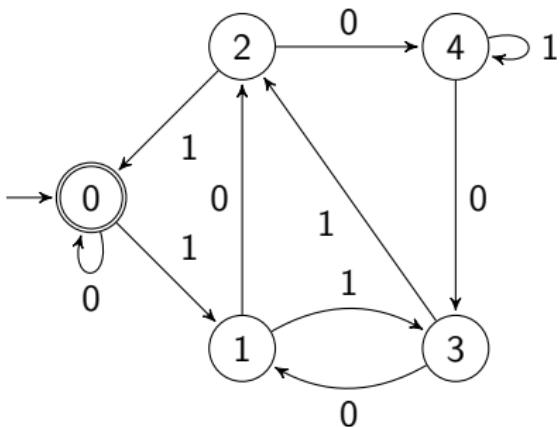
Example 1.3 (regulární jazyk)

- $L = \{w \mid w = ubaba, w \in \{a, b\}^*, u \in \{a, b\}^*\}$.



Example 1.4 (regulární jazyk)

- $L = \{w \mid w \in \{0, 1\}^* \& w \text{ je binární zápis čísla dělitelného } 5\}$.



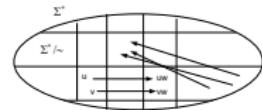
Example 1.5 (!N)Eregulární jazyk)

- $L = \{0^n 1^n \mid w \in \{0, 1\}^*, n \in \mathbb{N}\}$
NENÍ regulární jazyk.

Definice 1.6: kongruence

Mějme konečnou abecedu Σ a relaci ekvivalence \sim na Σ^* (reflexivní, symetrická, tranzitivní). Potom:

- \sim je **pravá kongruence**, jestliže
$$(\forall u, v, w \in \Sigma^*) u \sim v \Rightarrow uw \sim vw.$$
- je **konečného indexu**, jestliže rozklad Σ^*/\sim má konečný počet tříd.
- Třídu kongruence \sim obsahující slovo u značíme $[u]_\sim$, resp. $[u]$.



Example 1.6 (Pravá kongruence)

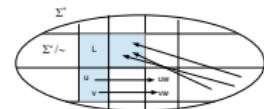
- Relace \sim_{end} 'končí stejným písmenem' je pravá kongruence,
 - ▶ pokud $ux \sim_{end} vx$, pak i $uxw \sim_{end} vxw$.
- Relace \sim_{fl} 'končí stejně jako začíná' je ekvivalence, $aa \sim_{fl} bb$, ale $aaa \not\sim_{fl} bba$, tedy není pravá kongruence.
- Relace $\sim_{||}$ 'mají stejný počet znaků' není konečného indexu.

Myhill–Nerodova věta

Věta 1.1: Myhill–Nerodova věta

Nechť L je jazyk nad konečnou abecedou Σ . Potom následující tvrzení jsou ekvivalentní:

- a) L je rozpoznatelný konečným automatem,
- b) existuje pravá kongruence \sim konečného indexu nad Σ^* tak, že L je sjednocením jistých tříd rozkladu Σ^* / \sim .



Důkaz: Důkaz Myhill–Nerodovy věty \Rightarrow

a) \Rightarrow b); tj. automat \Rightarrow pravá kongruence konečného indexu

- definujeme $u \sim v \equiv \delta^*(q_0, u) = \delta^*(q_0, v)$.
- je to ekvivalence (reflexivní, symetrická, transitivní)
- je to pravá kongruence (z definice δ^*)
- má konečný index (konečně mnoho stavů)

$$L = \{w \mid \delta^*(q_0, w) \in F\} = \bigcup_{q \in F} \{w \mid \delta^*(q_0, w) = q\} = \bigcup_{q \in F} [w \mid \delta^*(q_0, w) = q]_\sim$$

Theorem (Myhill–Nerodova věta - 'kopie')

Nechť L je jazyk nad konečnou abecedou Σ . Potom následující tvrzení jsou ekvivalentní:

- L je rozpoznatelný konečným automatem,
- existuje pravá kongruence \sim konečného indexu nad Σ^* tak, že L je sjednocením jistých tříd rozkladu Σ^* / \sim .

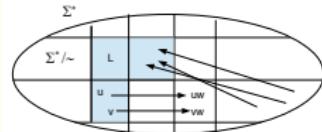
Důkaz: Důkaz Myhill–Nerodovy věty \Leftarrow

b) \Rightarrow a); tj. pravá kongruence konečného indexu \Rightarrow automat

- abeceda automatu vezmeme Σ
- za stavy Q volíme třídy rozkladu Σ^* / \sim
- počáteční stav $q_0 \equiv [\epsilon]$
- koncové stavy $F = \{c_1, \dots, c_n\}$, kde $L = \bigcup_{i=1, \dots, n} c_i$
- přechodová funkce $\delta([u], x) = [ux]$ (je korektní z def. pravé kongruence).
- $L(A) = L$

$$w \in L \Leftrightarrow w \in \bigcup_{i=1, \dots, n} c_i \Leftrightarrow w \in c_1 \vee \dots \vee w \in c_n \Leftrightarrow [w] = c_1 \vee \dots \vee [w] = c_n \Leftrightarrow [w] \in F \Leftrightarrow w \in L(A)$$

$$\delta^*([\epsilon], w) = [w]$$



□

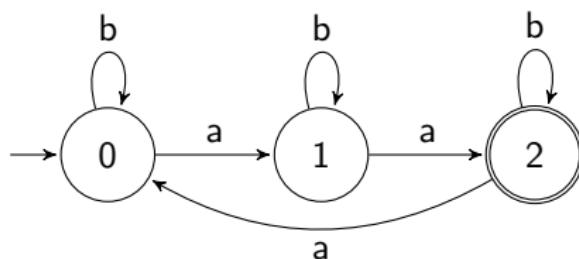
Použití Myhill–Nerodovy věty: Konstrukce automatů

Example 1.7

Sestrojte automat přijímající jazyk

$L = \{w \mid w \in \{a, b\}^* \& |w|_a = 3k + 2\}$, tj. obsahuje $3k + 2$ symbolů a .

- $|u|_x$ značí počet symbolů x ve slově u
- definujme $u \sim v \equiv (|u|_a \bmod 3 = |v|_a \bmod 3)$
- třídy ekvivalence $0, 1, 2$
- L odpovídá třídě 2
- a – přechody do následující třídy
- b – přechody zachovávají třídu.



Ne-regulární jazyk

Example 1.8 (Ne-regulární jazyk)

Jazyk $L_{a^+ b^i c^i} = \{ u | u = a^+ b^i c^i \vee u = b^i c^j, + \in \mathbb{N}_{>0}, i, j \in \mathbb{N}_0 \}$ není regulární (Myhill–Nerodova věta).

Důkaz: Jazyk $L_{a^+ b^i c^i}$ není regulární

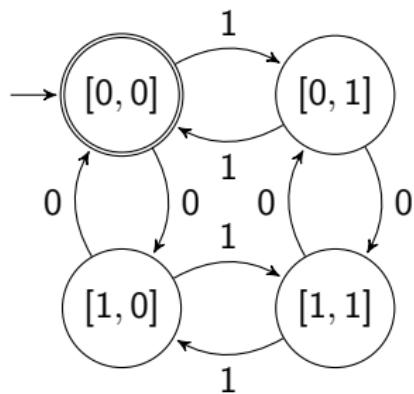
- Důkaz sporem: Předpokládejme, že L je regulární
⇒ pak existuje pravá kongruence \sim_L konečného indexu m , L je sjednocení některých tříd Σ^*/\sim_L
- vezmeme množinu řetězců $S = \{ab, abb, abbb, \dots, ab^{m+1}\}$
- existují dvě slova $i \neq j$, která padnou do stejné třídy
 - $i \neq j$ $ab^i \sim_L ab^j$
 - přidáme c^i $ab^i c^i \sim_L ab^j c^i$ \sim je kongruence
 - spor $ab^i c^i \in L \text{ & } ab^j c^i \notin L$ s 'L je sjednocení některých tříd Σ^*/\sim_L

Příklad - 'součin' automatů

Example 1.9

$L = \{w \mid w \in \{0, 1\}^*, |w|_0 = 2k \& |w|_1 = 2\ell, k, \ell \in \mathbb{N}_0\}$, tj.

- sudý počet 0
- a zároveň sudý počet 1.



δ	0	1
$* \rightarrow [0, 0]$	$[1, 0]$	$[0, 1]$
$[0, 1]$	$[1, 1]$	$[0, 0]$
$[1, 0]$	$[0, 0]$	$[1, 1]$
$[1, 1]$	$[0, 1]$	$[1, 1]$

Příklad (špatného) protokolu pro elektronický převod peněz

- Tři zúčastnění: zákazník, obchod, banka.
- Pro jednoduchost jen jedna platba (soubor 'money').

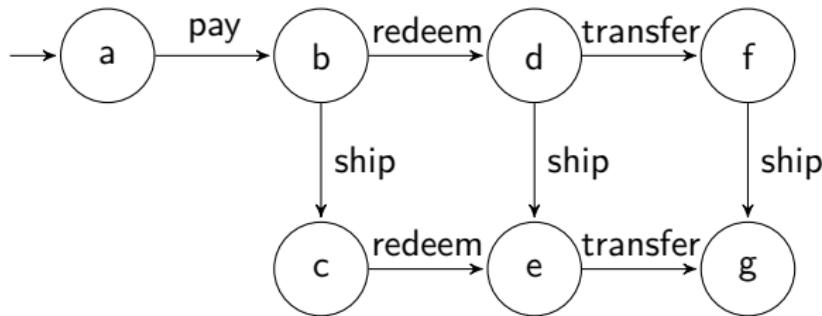
Example 1.10

Zákazník poskytne obchodu číslo kreditní karty, obchod si vyžádá peníze od banky a poše zboží zákazníkovi. Zákazník má možnost zablokovat kartu a žádat zrušení transakce.

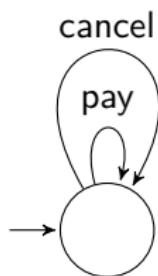
Pět událostí:

- ▶ Zákazník může zadat číslo karty **pay**.
- ▶ Zákazník může kartu zablokovat **cancel**.
- ▶ Obchod může poslat **ship** zboží zákazníkovi.
- ▶ Obchod může vyžádat **redeem** peníze od banky.
- ▶ Banka může převést **transfer** peníze obchodu.

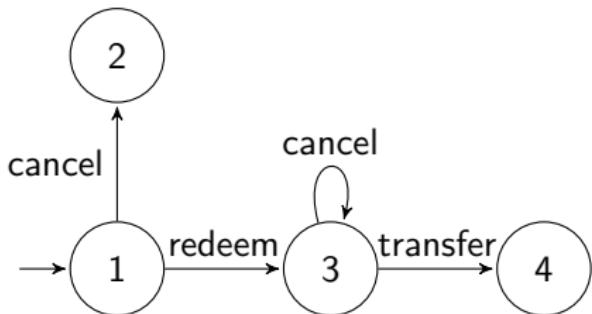
(Neúplný) konečný automat pro bankovní příklad



Obchod



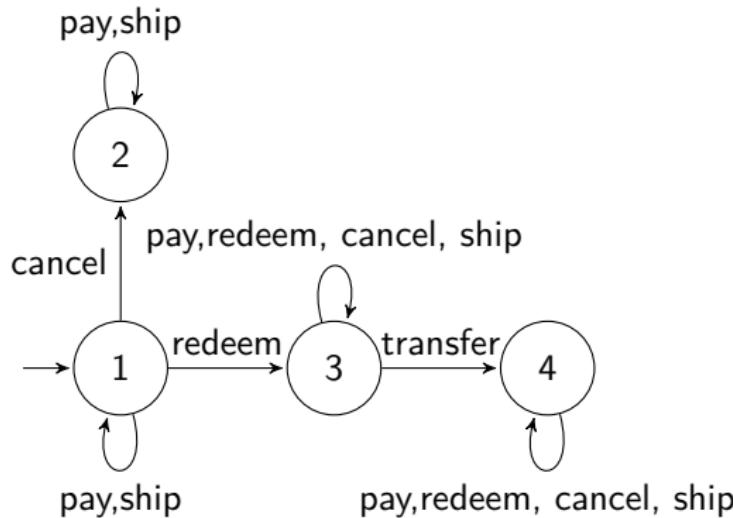
Zákazník



Banka

Hrana pro každý vstup

- Můžeme vyžadovat, aby automat provedl akci pro každý vstup. Obchod přidá hrany pro každý stav do sebe samého označenou *cancel*.
- Zákazník by neměl shodit bankovní automat opětovným zaplacením *pay*, proto přidáme smyčku *pay*. Podobně s ostatními akcemi.

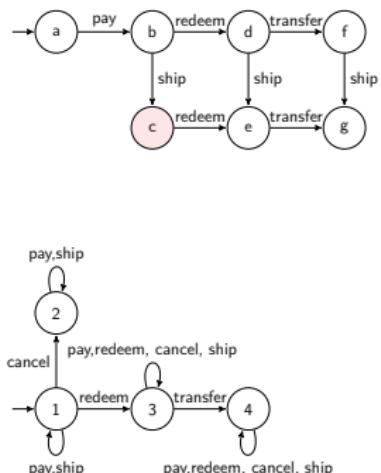
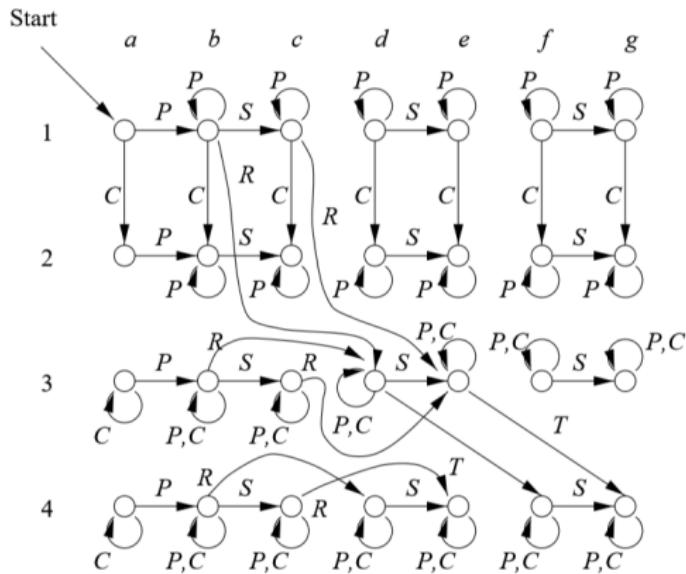


Úplnější automat pro banku.

Součin automatů

- Součin automatů pro banku a obchod má stavy dvojice $B \times O$.
- Hrana v součinu automatů provádí paralelně akce v bance a obchodě. Pokud jednomu chybí akce, bude chybět i součinu automatů.

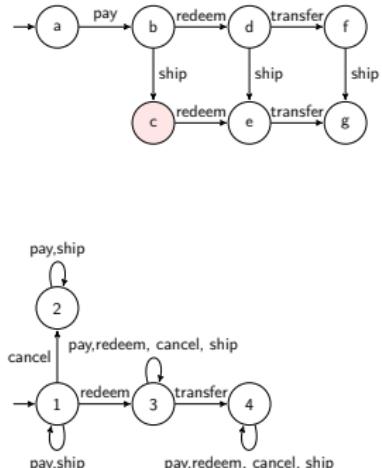
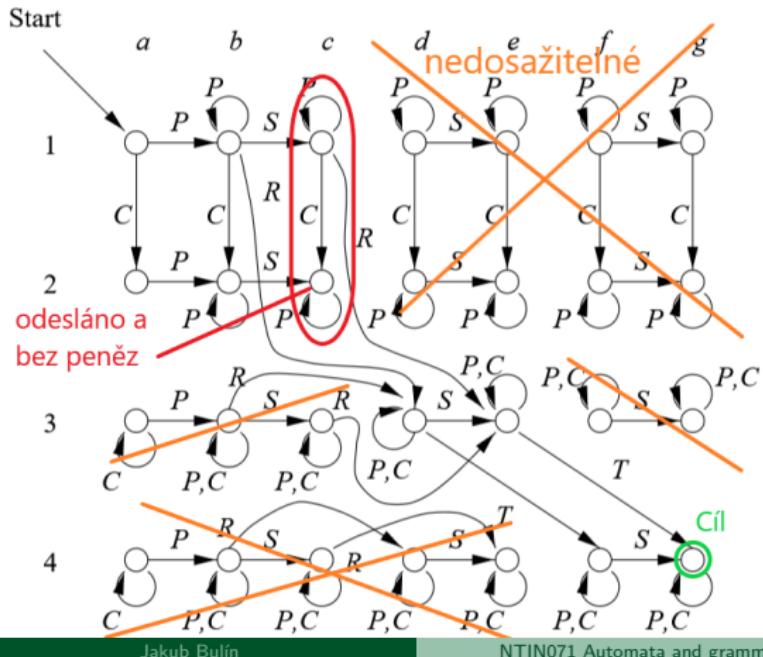
J.E. Hopcroft, R. Motwani, J.D. Ullman: *Introduction to Automata Theory, Languages, and Computations*, Addison-Wesley



Součin automatů

- Součin automatů pro banku a obchod má stavy dvojice $B \times O$.
- Hrana v součinu automatů provádí paralelně akce v bance a obchodě. Pokud jednomu chybí akce, bude chybět i součinu automatů.

J.E. Hopcroft, R. Motwani, J.D. Ullman: *Introduction to Automata Theory, Languages, and Computations*, Addison-Wesley



Dnes jsme probrali

- Definice
 - ▶ deterministického konečného automatu $A = (Q, \Sigma, \delta, q_0, F)$
 - ▶ jazyka $L \subseteq \Sigma^*$
 - ▶ jazyka rozpoznávaného konečným automatem
$$L(A) = \{w \mid w \in \Sigma^* \text{ & } \delta^*(q_0, w) \in F\}$$
- Myhill–Nerodova věta
- příklad důkazu ne-regulárnosti jazyka $(\{ab^ic^i \mid i \in \mathbb{N}\}, \{0^i1^i \mid i \in \mathbb{N}\})$
- příklady regulárních jazyků

Chtěli bychom dokázat, že L_u není rekurzivní. K tomu se nám bude hodit Postova věta, která má i daleko širší uplatnění.