

Automaty a gramatiky

TIN071

Marta Vomlelová

marta@ktiml.mff.cuni.cz
<http://ktiml.mff.cuni.cz/~marta>

February 21, 2024

Organizační záležitosti

- Přednáška:

- moodle <https://dl1.cuni.cz/course/index.php?categoryid=337>
 - 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 sestrojít automaty a gramatiky
- zařijete 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)determinizmu,
 - Napište definici, formulujte větu, popište ideu důkazu, algoritmus,
 - zařadte jazyk do Chomského hierarchie a svou odpověď dokažte.

Požadavky ke zkoušce

- **Zápočet je nutnou podmínkou účasti na zkoušce.**
- Zkouška sestává z písemného testu a ústní části.
- **Písemný test** bude sestávat ze dvou až tří otázek, které korespondují sylabu přednášky, ověřují schopnosti získané na cvičení a znalost definic, vět a algoritmů z přednášky.
- ! **Po dobu písemného testu 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 písemného testu, zpravidla budete dotázáni na vysvětlení-zdůvodnění-příklady k tvrzením v písemné části. Ústní část může být doplněna otázkou v rozsahu sylabu přednášky s písemným testem nesouvisející.

Zdroje a literatura

- 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. Chytil: *Automaty a gramatiky*, SNTL Praha, 1984
- ⇒ moodle <https://dl1.cuni.cz/course/index.php?categoryid=337>
- kde jsou tyto slajdy
 - moodle testy (které ale netestují zdůvodnění a důkazy).
- ⇒ cvičení.

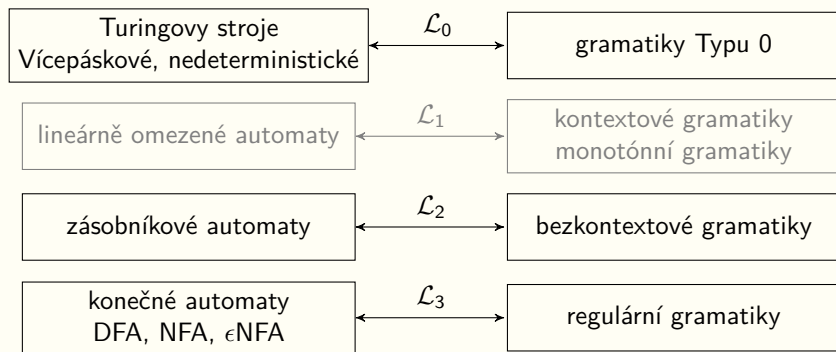
Pohled do historie

- 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.

Cíl přednášky

- 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 $P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXPTIME$.

Automaty a gramatiky – dva způsoby popisu



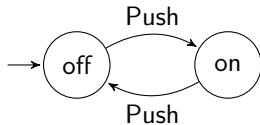
Praktické využití

- 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.

Jednoduché příklady konečných automatů

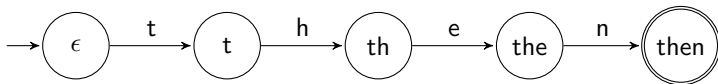
- Návrh a verifikace integrovaných obvodů.

Konečný automat modelující spínač on/off .



- Lexikální analýza

Konečný automat rozpoznávající slovo then.



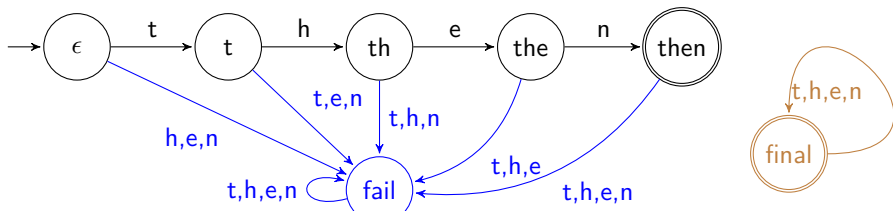
Definition 1.1 (Deterministický konečný automat)

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 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 dvojíým kruhem či šipkou '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 $\forall s \in \Sigma: \delta(\text{final}, s) = \text{final}$.

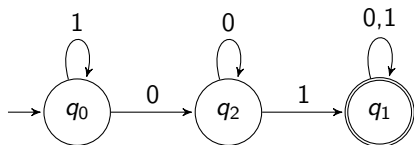


Popis konečného automatu

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

Definition 1.2 (Slovo, $\epsilon, \lambda, \Sigma^*, \Sigma^+$, jazyk)

Mějme neprázdnou množinu symbolů Σ .

- **Slovo** 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 .

Definition 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

Definition 1.4 (rozšířená přechodová funkce)

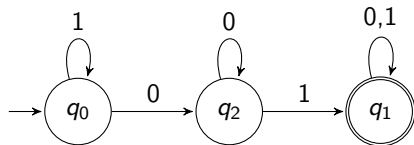
Mějme přechodovou funkci $\delta : Q \times \Sigma \rightarrow Q$.

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^* .

$$\delta^*(q_0, 1100) = q_2, \delta^*(q_0, 1100111111111001) = q_1$$



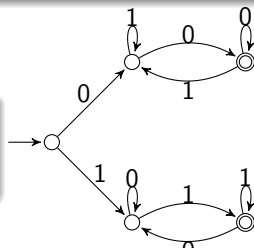
Jazyky rozpoznatelné konečnými automaty

Definition 1.5 (jazyky rozpoznatelné konečnými automaty, 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 jazyky)

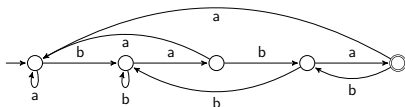
- $L = \{w \mid w = xux, w \in \{0, 1\}^*, x \in \{0, 1\}, u \in \{0, 1\}^*\}$.



Příklady regulárních jazyků

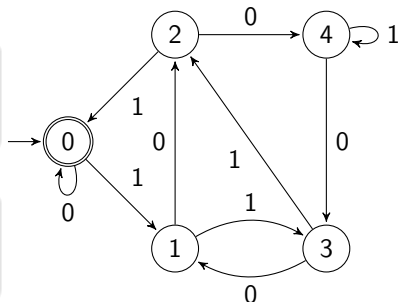
Example 1.3 (regulární jazyk)

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



Example 1.4 (regulární jazyk)

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



Example 1.5 (!Neregulární jazyk)

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

Iterační (pumping) lemma pro regulární jazyky

Theorem 1.1 (Iterační (pumping) lemma pro regulární jazyky)

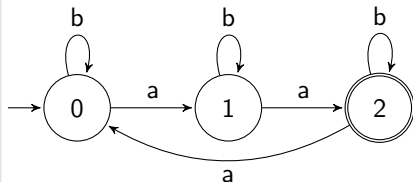
Mějme regulární jazyk L . Pak existuje konstanta $n \in \mathbb{N}$ (závislá na L) tak že každé $w \in L$; $|w| \geq n$ můžeme rozdělit na tři části, $w = xyz$, že:

- $y \neq \epsilon$
- $|xy| \leq n$
- $\forall k \in \mathbb{N}_0$, slovo xy^kz je také v L .

Example 1.6

- Lemma řeklo: $n = 3$.
- $abbbba = a(b)bbba$;
 $\forall i \geq 0$; $a(b)^i bbba \in L(A)$.
- $aaaaba = (aaa)aba$;
 $\forall i \geq 0$; $(aaa)^i aba \in L(A)$.
- aa nelze pumpovat,
ale $|aa| < n$.

Automat A

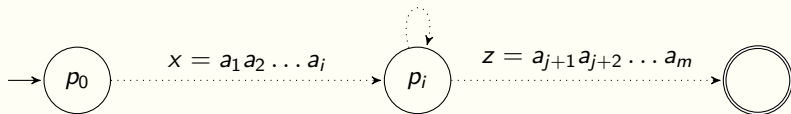


Důkaz iteračního lematu pro regulární jazyky

Proof: iteračního lematu pro regulární jazyky

- Mějme regulární jazyk L , pak existuje DFA A s n stavy, že $L = L(A)$.
- Vezměme libovolné slovo $a_1 a_2 \dots a_m = w \in L$ délky $m \geq n$, $a_i \in \Sigma$.
- Definujme: $\forall i \ p_i = \delta^*(q_0, a_1 a_2 \dots a_i)$. Platí $p_0 = q_0$.
- Máme $n + 1$ p_i a n stavů, některý se opakuje, vezměme první takový, tj. $(\exists i, j)(0 \leq i < j \leq n \ \& \ p_i = p_j)$.
- Definujme: $x = a_1 a_2 \dots a_i$, $y = a_{i+1} a_{i+2} \dots a_j$, $z = a_{j+1} a_{j+2} \dots a_m$, tj. $w = xyz$, $y \neq \epsilon$, $|xy| \leq n$.

$$y = a_{i+1} a_{i+2} \dots a_j$$



- Smyčka nad p_i se může opakovat libovolně krát a vstup je také akceptovaný. □

Použití pumping lemmatu

Example 1.7 (Pumping lemma jako hra s oponentem)

Jazyk $L_{eq} = \{w; |w|_0 = |w|_1\}$ slov se stejným počtem 0 a 1 není regulární.

Proof: Jazyk L_{eq} není regulární.

- Předpokládejme že L_{eq} je regulární. Vezměme n z pumping lemmatu.
- Zvolme $w = 0^n 1^n \in L_{eq}$.
- Rozdělime $w = xyz$ dle pumping lemmatu, $y \neq \epsilon$, $|xy| \leq n$.
- Protože $|xy| \leq n$ je na začátku w , obsahuje jen 0.
- Z pumping lemmatu: $xz \in L_{eq}$ (pro $k = 0$). To má ale méně 0 než 1, takže nemůže být v L_{eq} . □

Example 1.8

Jazyk $L = \{0^i 1^i; i \geq 0\}$ není regulární.

Aplikace pumping lemmatu 2

Example 1.9

Jazyk L_{pr} slov 1^p kde p je prvočíslo není regulární.

Proof: L_{pr} slov 1^p kde p je prvočíslo není regulární.

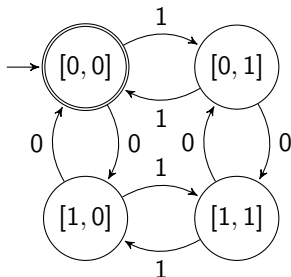
- Předpokládejme že L_{pr} je regulární. Vezměme n z pumping lemmatu. Zvolme prvočíslo $p \geq n + 2$, označme $w = 1^p$.
- Rozložme $w = xyz$ dle pumping lemmatu, nechť $|y| = m$. Pak $|xz| = p - m$.
- $xy^{p-m}z \in L_{pr}$ z pumping lemmatu, ale $|xy^{p-m}z| = |xz| + (p - m)|y| = p - m + (p - m)m = (m + 1)(p - m)$ není prvočíslo (žádný z činitelů není 1). □

Příklad - 'součin' automatů

Example 1.10

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

- 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').

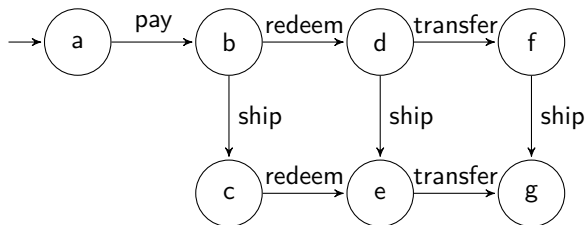
Example 1.11

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

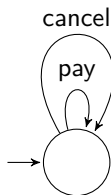
Pět událostí:

- 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.

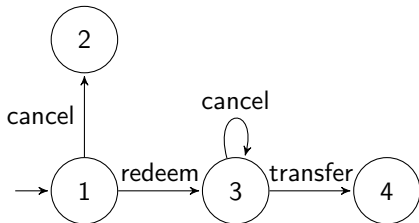
(Neúplný) konečný automat pro bankovní příklad



Obchod



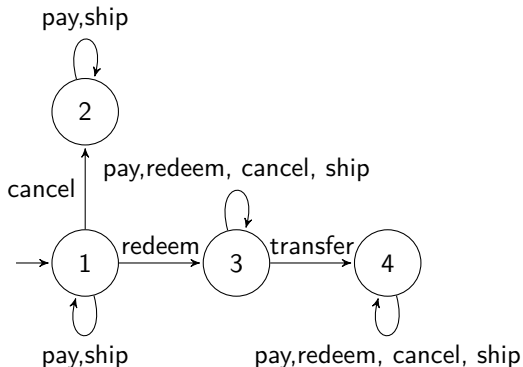
Zákazník



Banka

Hrana pro každý vstup

- Můžeme vyžadovat, aby automat provedl akci pro každý vstup. Obchod přidá hranu 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.

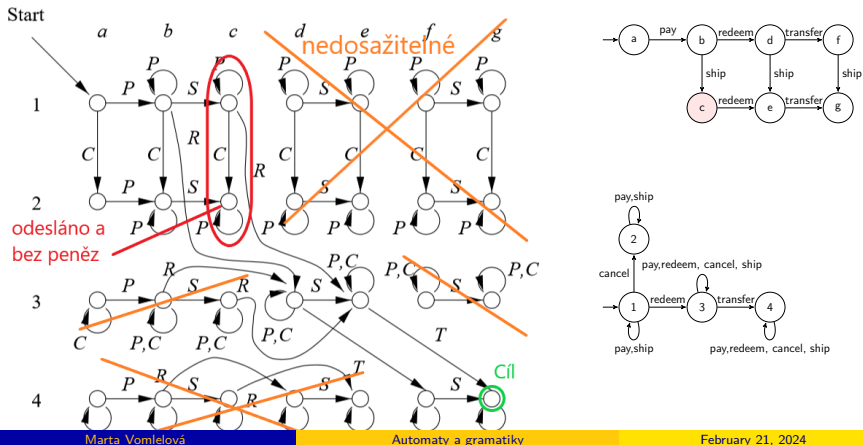


Úplnější automat pro banku.

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



Definition 1.6 (Dosažitelné stavy)

Mějme DFA $A = (Q, \Sigma, \delta, q_0, F)$ a $q \in Q$. Řekneme, že stav q je **dosažitelný**, jestliže existuje $w \in \Sigma^*$ takové, že $\delta^*(q_0, w) = q$.

Algorithm: Hledání dosažitelných stavů

Dosažitelné stavy hledáme iterativně.

- Začátek: $M_0 = \{q_0\}$.
- Opakuj: $M_{i+1} = M_i \cup \{q \mid q \in Q, (\exists p \in M_i, \exists x \in \Sigma) \delta(p, x) = q\}$
- opakuj dokud $M_{i+1} \neq M_i$.

Proof: Korektnost a úplnost

- Korektnost: $M_0 \subseteq M_1 \subseteq \dots \subseteq Q$ a každé M_i obsahuje pouze dosažitelné stavy.
- Úplnost:
 - nechť q je dosažitelný, tj. $(\exists w \in \Sigma^*) \delta^*(q_0, w) = q$
 - vezměme nejkratší takové $w = x_1 \dots x_n$ tž. $\delta^*(q_0, x_1 \dots x_n) = q$
 - zřejmě $\delta^*(q_0, x_1 \dots x_i) \in M_i$ (dokonce $M_i \setminus M_{i-1}$)
 - tedy $\delta^*(q_0, x_1 \dots x_n) \in M_n$, tedy $q \in M_n$.



Dnes jsme probrali

- Definice
 - deterministického konečného automatu $A = (Q, \Sigma, \delta, q_0, F)$
 - jazyka $L \subseteq \Sigma^*$
 - jazyka rozpoznávaného konečným automatem
$$L(A) = \{w \mid w \in \Sigma^* \ \& \ \delta^*(q_0, w) \in F\}$$
- iterační (pumping) lemma pro regulární jazyky
- příklad důkazu ne-regulárnosti jazyka 0^i1^i
- příklady regulárních jazyků
- způsob nalezení nedosažitelných stavů.

Přehled kapitol

- 1 Úvod, Iterační lemma pro reg. jazyky
- 2 Redukovaný DFA a ekvivalence automatů, stavů
- 3 Nedeterministické ϵ -NFA, Operace zachovávající regularitu
- 4 Regulární výrazy, Kleeneova věta, Substituce, Homomorfizmus
- 5 Dvousměrné FA, Mealy a Moore stroje
- 6 Gramatiky, Chomského hierarchie, víceznačnost
- 7 Chomského NF, Pumping Lemma pro CFL
- 8 CYK – náležení do CFL
- 9 Zásobníkové automaty, Deterministické PDA
- 10 Uzávěrové vlastnosti, Dykovy jazyky
- 11 Turingův stroj, rozšíření
- 12 Lineárně omezené automaty, Univerzální TM, Diagonální jazyk
- 13 Nerozhodnutelné problémy, Postův korespondenční p.
- 14 Časová složitost