AUTOMATY A GRAMATIKY

11

Kontextové uzávěrové vlastnosti
Turingův stroj
Rekurzivně spočetné jazyky
Kódování, enumerace

Pavel Surynek

Univerzita Karlova v Praze

Matematicko-fyzikální fakulta
Katedra teoretické informatiky a matematické logiky

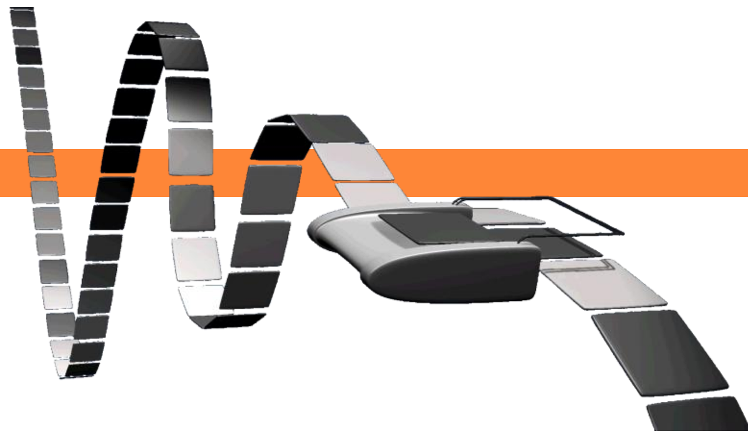
Kontextové **uzávěrové vlastnosti** (1)

- kontextové jazyky jsou uzavřené na konečná **sjednocení**
  - kontextové gramatiky $G_1=(V_N^1, V_T, S_1, P_1)$ a $G_2=(V_N^2, V_T, S_2, P_2)$, kde $V_N^1 \cap V_N^2 = \emptyset$
  - položme $G = (V_N^1 \cup V_N^2 \cup \{S'\}, V_T, S', P1 \cup P2 \cup \{S' \to S_1 | S_2\})$
    - S' je nový neterminál
      - ošetřit přepis původních počátečních symbolů na λ
  - platí $L(G) = L(G_1) \cup L(G_2)$
- kontextové jazyky jsou uzavřené na **konkatenaci**
  - položme $G = (V_N^1 \cup V_N^2 \cup \{S'\}, V_T, S', P_1 \cup P_2 \cup \{S' \to S_1.S_2\})$
    - S' je nový neterminál
      - ošetřit přepis původních počátečních symbolů na λ
  - platí $L(G) = L(G_1).L(G_2)$
- kontextové jazyky jsou uzavřené na **iteraci**
  - položme $G = (V_N^1 \cup \{S', S''\}, V_T, S', P_1 \cup \{S' \to \lambda \mid S'',\ S'' \to S''S_1 \mid S_1\})$
    - S', S'' jsou nové neterminály
      - je třeba dávat pozor na výskyt počátečního neterminálu vpravo (u bezkontextových netřeba)
  - platí $L(G) = L(G_1)^*$
- kontextové jazyky jsou uzavřené na **zrcadlový obraz**
  - položme $G = (V_N^1, V_T, S_1, \{v^R \to w^R \mid v \to w \in P_1\})$
  - platí $L(G) = L(G_1)^R$

Kontextové **uzávěrové vlastnosti** (2)

- kontextové jazyky jsou uzavřené na **kontextovou substituci**
  - substituce f: $X \rightarrow 2^{Y^*}$ je <u>kontextová substituce</u>, jestliže $f(x)$ je kontextový jazyk pro každé $x \in X$
    - máme $G_x = (V_N^x, Y, S_x, P_x)$, že $f(x) = L(G_x)$ pro každé $x \in X$
      - V_N^x jsou po dvou disjunktní
  - mějme $G = (V_N, X, S, P)$, zajímá nás kontextovost $f(L(G))$
  - položíme $G' = (V_N \cup \bigcup_{x \in X} V_N^x \cup X, Y, S, P \cup \bigcup_{x \in X} P^x \cup \{x \rightarrow S_x \mid x \in X\})$
    - případně ošetříme přepis neterminálů jiných než S na λ
    - platí $L(G') = f(L(G))$
- kontextové jazyky jsou **uzavřené na doplňky** a **konečné průniky**
  - velmi hluboký výsledek (řešilo se asi 20 let)
    - věta Immerman–Szelepcsényi

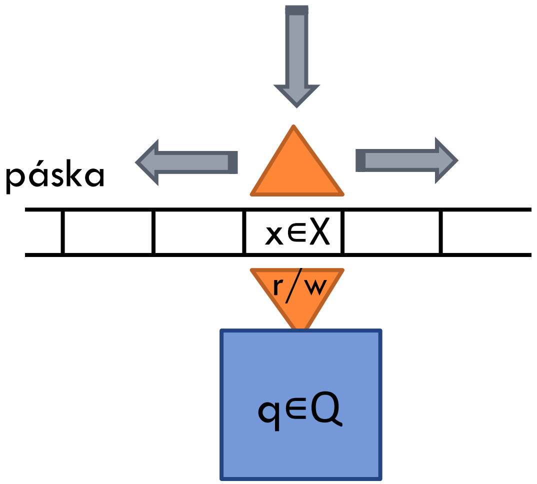
Turingův stroj (1)



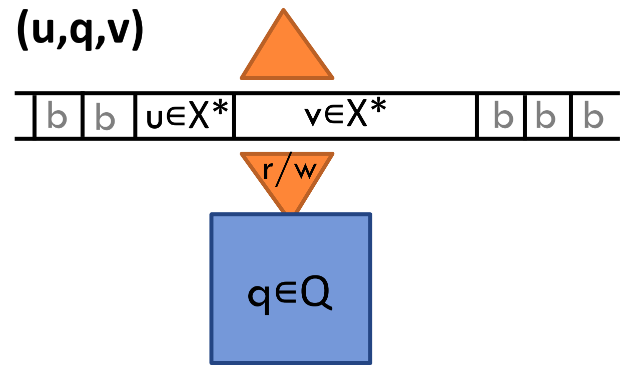
- **Turingův stroj**
  - automat, který může přepisovat pásku
  - páska je potenciálně nekonečná
    - v konečném čase stroj použije její konečnou část
- historická vsuvka
  - 30. léta 20. století
    - konečná výpočetní zařízení, formalizace algoritmu a problému
      - Turingův stroj, λ-kalkulus, RAM (*random access machine*)
  - osobnosti
    - Turing, Kleene, Markov, Post, Church
  - Churchova hypotéza (metavěta)
    - všechna konečná výpočetní zařízení jsou ekvivalentní, tj. ekvivaletní Turingovu stroji
      - nedokázáno ani nevyvráceno

Turingův stroj (2)

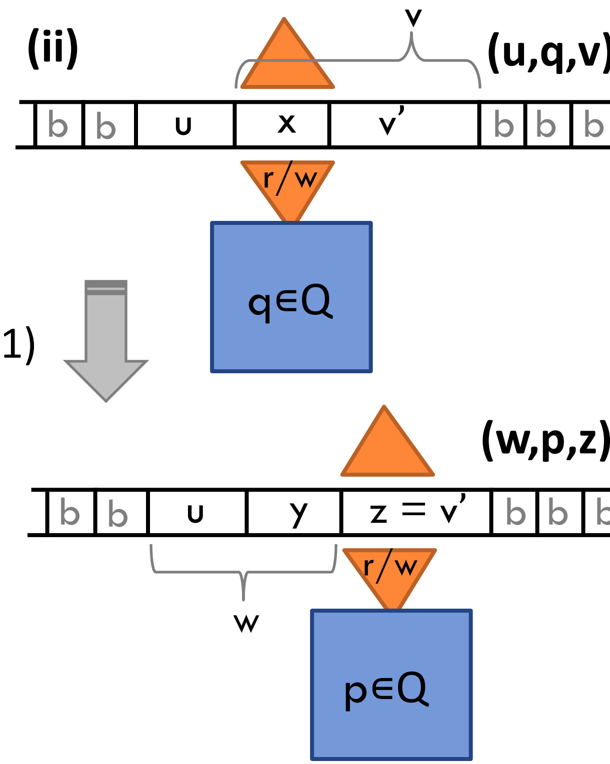
- Turingův stroj **formálně**
  - $T = (Q, X, \delta, q_0, b, F)$
    - Q - konečná neprázdná množina stavů
    - X - konečná neprázdná množina symbolů (abeceda)
    - $F \subseteq Q$ - množina přijímajících stavů
    - $\delta: (Q-F)\times X \rightarrow Q\times X\times\{-1, 0, +1\}$
      - přechodová funkce
        - pro stav a čtený symbol dává nový stav, symbol k zápisu a následující pozici na pásce
        - v přijímajícím stavu výpočet končí
    - $q_0 \in Q$ - počáteční stav
    - $b \in X$ - symbol pro prázdnou buňku na pásce
  - **předpoklady**
    - páska je směrem doprava nekonečná
    - vstupní slovo je napsané na levém konci pásky
      - zprava doplněné symbolem b do nekonečna
    - počáteční pozice je na prvním písmenu vstupního slova



Výpočet Turingova stroje

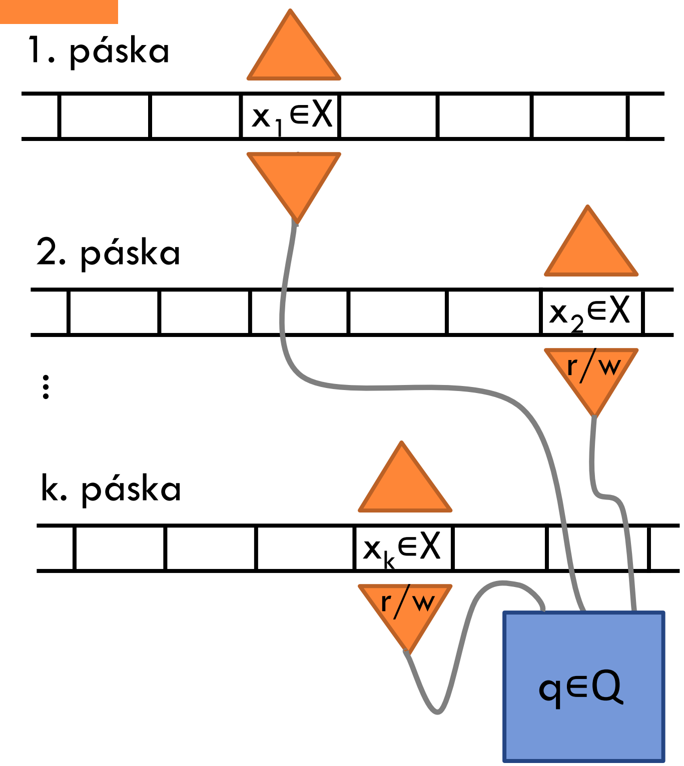


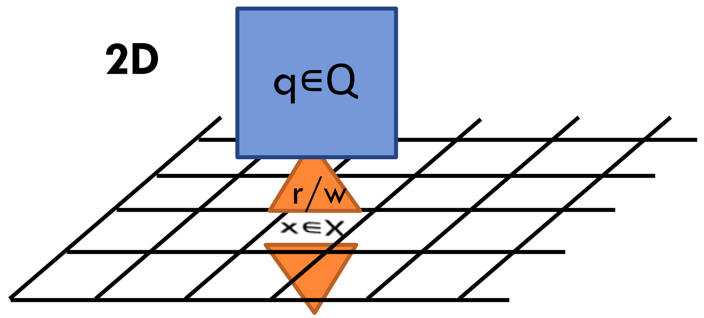
- **konfigurace** Turingova stroje
  - trojice (u,q,v), kde u,v∈X* a q∈Q
    - u.v je slovo na nejmenší souvislé části pásky obsahující symboly ≠ b a čtenou buňku
    - u je **část vlevo** od aktuální čtené pozice (kromě)
    - v je **část vpravo** od aktuální čtené pozice (včetně)
- **změna konfigurace** (u,q,v) na (w,p,z), jestliže
  - (i) v = x.v’ pro x∈X, w = u, z = y.v’ a δ(q,x)=(p,y,0), nebo
    - žádný posun
  - (ii) v = x.v’ pro x∈X, w = u.y, z = v’ a δ(q,x)=(p,y,+1), nebo
    - posun doprava
  - (iii) u= u’.x’, v = x.v’ pro x,x’∈X, w = u’, z = x’.y.v’ a δ(q,x)=(p,y,-1)
    - posun doleva
  - zapisujeme (u,q,v) $\vdash_T$ (w,p,z)
    - nutno ošetřit případy, kdy u=λ
- změna konfigurace na konečně mnoho kroků
  - (u,q,v) $\vdash_T^*$ (w,p,z)



Varianty Turingova stroje

- **nedeterministický**
  - přechodová funkce bude nedeterministická
    - $\delta: (Q\text{-}F)\times X \rightarrow 2^{Q\times X\times\{-1,\,0,\,+1\}}$
  - pojem změny konfigurace se upraví
    - místo $\delta(q,x) = \ldots$ budeme mít $\delta(q,x) \ni \ldots$
- **vícepáskový** (k pásek)
  - $\delta: (Q\text{-}F)\times X^k \rightarrow Q\times X^k\times\{-1, 0, +1\}^k$
    - případně $\delta: (Q\text{-}F)\times X^k \rightarrow 2^{Q\times X^k\times\{-1,\,0,\,+1\}^k}$ u nedeterministické varianty
  - vstupní slovo na první pásce
  - změna konfigurace analogicky
- **multi-dimenzionální**
  - páska má několik rozměrů (například 2D, 3D)
    - pohyb je možný ve všech směrech





Jazyky a Turingovy stroje

- slovo $w \in (X-\{b\})^*$ je **přijímáno** Turingovým strojem $T = (Q, X, \delta, q_0, b, F)$, jestliže
  - $(\lambda, q_0, w) \vdash_T^* (u,f,v)$ pro $u,v \in X^*$ a $f \in F$
    - někdy je vyžadováno smazání pásky, tj. $(\lambda, q_0, w) \vdash_T^* (\lambda,f,b)$
- jazyk L(T) přijímaný Turingovým strojem T
  - $L(T) = \{ w \mid w \in (X-\{b\})^* \wedge (\exists f \in F)\ (\lambda, q_0, w) \vdash_T^* (u,f,v) \}$
- jazyky přijímané Turingovými stroji jsou právě **rekurzivně spočetné** jazyky (typu 0) [*recursively enumerable language*]
  - platí s libovolnou variantou Turingova stroje (více pásek, více dimenzí, s nedeterminismem)
    - všechny varianty TS mají stejnou výpočetní sílu
    - lze se omezit na jednopáskový deterministický

Turingův stroj ⇒ gramatika

- mějme deterministický jednopáskový TS $T = (Q, X, \delta, q_0, b, F)$
  - hledáme gramatiku $G=(V_N,X,S,P)$ (bez omezení), že $L(G) = L(T)$
    - G vygeneruje pásku s prostorem pro výpočet a kopii vstupního slova, simuluje výpočet T nad vygenerovanou páskou (páska obsahuje zracadlový obraz vstupu)
      - $V_N = \{ X_x \mid x \in X \} \cup \{ Q_q \mid q \in Q \} \cup \{S, I, B, E\}$
        - I pomocný neterminál pro generování $w \in X^*$ a w^R, B pro generování prostoru na simulované pásce, E pro mazání simulační pásky
      - pravidla P
        - zahájení
          - $S \to I\, Q_{q0}\, B$
          - $I \to x\, I\, X_x \mid B$ pro $x \in X$
            - generování slova w a jeho kopie w^R
          - $B \to X_b\, B \mid X_b$
            - generování volného prostoru
        - simulace výpočtu nad w^R
          - $X_x\, Q_q\, X_y \to X_z\, Q_p\, X_y$ když $\delta(q,x)=(p,z,0)$
          - $X_x\, Q_q\, X_y \to Q_p\, X_z\, X_y$ když $\delta(q,x)=(p,z,+1)$
          - $X_x\, Q_q\, X_y \to X_z\, X_y\, Q_p$ když $\delta(q,x)=(p,z,-1)$
        - mazání pásky
          - $Q_f \to E$ pro $f \in F$ $\quad X\, E \to E$ pro $X \in V_N$ $\quad E\, X \to E$ pro $X \in V_N$ $\quad E \to \lambda$

nechť $w = x_1x_2\ldots x_n$

derivace: $S \Rightarrow_G^* \ldots$

$$x_1x_2\ldots x_n\, I\, X_{x_n} X_{x_{n-1}} \ldots X_{x_1} Q_{q0}\, B \Rightarrow_G$$
$$x_1x_2\ldots x_n\, B\, X_{x_n} X_{x_{n-1}} \ldots X_{x_1} Q_{q0}\, B \Rightarrow_G^* \ldots$$
$$x_1x_2\ldots x_n\, X_b \ldots X_b\, X_{x_n} X_{x_{n-1}} \ldots X_{x_1} Q_{q0}\, X_b \ldots X_b \Rightarrow_G^* \ldots$$
$$x_1x_2\ldots x_n\, u\, Q_f\, v \text{ pro } u,v \in \{X_x \mid x \in X\}^* \text{ a } f \in F \Rightarrow_G$$
$$x_1x_2\ldots x_n\, u\, E\, v \Rightarrow_G^* \ldots$$
$$x_1x_2\ldots x_n\, E\, v \Rightarrow_G^* x_1x_2\ldots x_n\, E \Rightarrow_G^* x_1x_2\ldots x_n$$

Gramatika ⇒ Turingův stroj

- mějme gramatiku (bez omezení) $G=(V_N,V_T,S,P)$
  - hledáme TS $T = (Q, X, \delta, q_0, b, F)$, že $L(T) = L(G)$
    - pracné po technické stránce, ukážeme **abstraktně**
  - konstrukce, z nichž na vyšší úrovni sestavíme T
    - TS T_1, že $L(T_1) = \{\#u\#v\# \mid u,v \in X^* \wedge u \Rightarrow_G v\}$, $\# \notin V_N \cup V_T$
      - na všech možných místech v *u* je vyzkoušena aplikace všech pravidel G
    - TS T_2, že $L(T_2) = \{ w \mid (\exists n \in \mathbb{N}_0)\ w = \#u_1\#u_2\#\ldots\#u_n\# \wedge u_1,\ldots,u_n \in (V_N \cup V_T)^* \wedge u_1 \Rightarrow_G u_2 \wedge u_2 \Rightarrow_G u_3 \wedge \ldots \wedge u_{n-1} \Rightarrow_G u_n \}$
      - T_2 bude založen na T_1
    - TS T_3 generující systematicky všechna slova nad $V_N \cup V_T \cup \{\#\}$ začínající #S#
      - T_3 vstupní slovo přepíše na následující slovo
    - spojením T_2 a T_3 vznikne požadovaný TS T

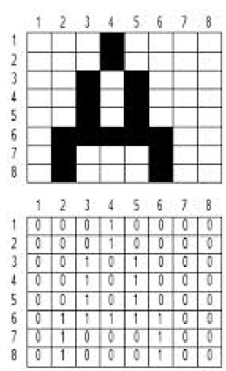
vstup: $z \in V_T^*$

na volné místo na pásce <u>zapiš</u> w = #S#
loop
  if $w \in L(T_2)$ **then**
    if w končí #z# **then** <u>přijmi</u> z
  w ← výsledek práce T_3 nad w

Kódování, slova a čísla

- kódování různých datových typů vysoké úrovně
  - na nižší úrovni jen <u>řetězce</u> nebo přirozená <u>čísla</u>
    - **ASCII** či **Unicode řetězce** jsou reprezentovány posloupností bitů, tj. slovy nad {0, 1}
      - slova nad {0, 1} lze interpretovat jako přirozená čísla
        - interpretace $b_{n-1}\dots b_1 b_0 = \sum b_i 2^i + 2^n$, kde $b_i \in \{0,1\}$ pro $i=0,1,\dots,n-1$
      - máme pojem <u>i-tého řetězce</u> pro $i \in \mathbb{N}$
    - analogicky pro **bitmapové obrázky**
      - obrázek → kódování jako binární řetězec → číslo
      - máme pojem <u>i-tého obrázku</u>
    - **program** je jen vhodně interpretovaný řetězec
      - máme <u>i-tý program</u> (respektive <u>i-tý Turingův stroj</u>)
    - “program + bitmapové obrázky = počítačová **hra**”
      - máme <u>i-tou hru</u>
    - **důkaz** je posloupnost výrazů
      - <u>i-tý důkaz</u>

Př.:		
	101	13. binární řetězec
	0101	21. binární řetězec
	00101	37. binární řetězec



Množiny konečné a nekonečné

- **konečná** množina K
  - počet jejích prvků - **kardinalita** je číslo $\in \mathbb{N}_0$
    - neexistuje vzájemně jednoznačné zobrazení mezi K a její vlastní podmnožinou

Př.: {a, b, c} je konečná množina kardinality 3

- **nekonečná** množina N
  - existuje vzájemně jednoznačné zobrazení mezi N a její vlastní podmnožinou

Př.: $\mathbb{N}$ je nekonečná
$1 \leftrightarrow 2$
$2 \leftrightarrow 4$
$3 \leftrightarrow 6$
...

- **spočetná** množina S
  - existuje vzájemně jednoznačné zobrazení mezi S a $\mathbb{N}$
    - spočetné množiny jsou nekonečné

Př.: celá čísla $\mathbb{Z}$
$0 \leftrightarrow 1$
$-i \leftrightarrow 2i$
$i \leftrightarrow 2i+1$
...

Enumerace a jazyky (1)

- **enumerace** množiny M je vzájemně jednoznačné zobrazení mezi M a $\mathbb{N}$
  - máme tedy enumeraci řetězců, programů, obrázků, ...
- **lze enumerovat jazyky** na danou abecedou X (={0,1})?
  - **nikoli**
    - **pro spor** předpokládejme, že jazyky nad {0, 1} enumerovat lze
    - máme tedy pojem i-tého jazyka, označme jazyky L_i pro $i \in \mathbb{N}$
    - položme $L = \{ w \mid w \in \{0,1\}^* \wedge w$ je i-tý binární řetězec $\Rightarrow w \notin L_i \}$
      - L je nad {0, 1}, mělo by se tedy jednat o j-tý jazyk pro jisté $j \in \mathbb{N}$
      - uvažme j-tý binární řetězec v a otázku, zda $v \in L$
        - když $v \in L$, pak $v \notin L$ podle definice L
        - když $v \notin L$, pak $v \in L$ podle definice L

Enumerace a jazyky (2)

- byla použita **diagonalizace**
  - překlopíme hodnoty na diagonále
    - nová diagonála nemůže být řádkem
    - s každým řádkem se v jedné buňce liší
- předpoklad, že jazyky nad {0,1} lze enumerovat, byl chybný
  - máme více jazyků než programů (Turingových strojů)
    - existují jazyky, pro které nemáme program (Turingův stroj), který je přijímá
      - nikoli všechny jazyky jsou rekurzivně spočetné
    - nekonstruktivní přístup
      - jak takový jazyk vypadá?

řetězce

jazyky

	1	2	3	4	5	…
1	10	0	1	1	0	
2		10				
3			01			
4				01		
5					10	
…						