

Automaty a formální jazyky

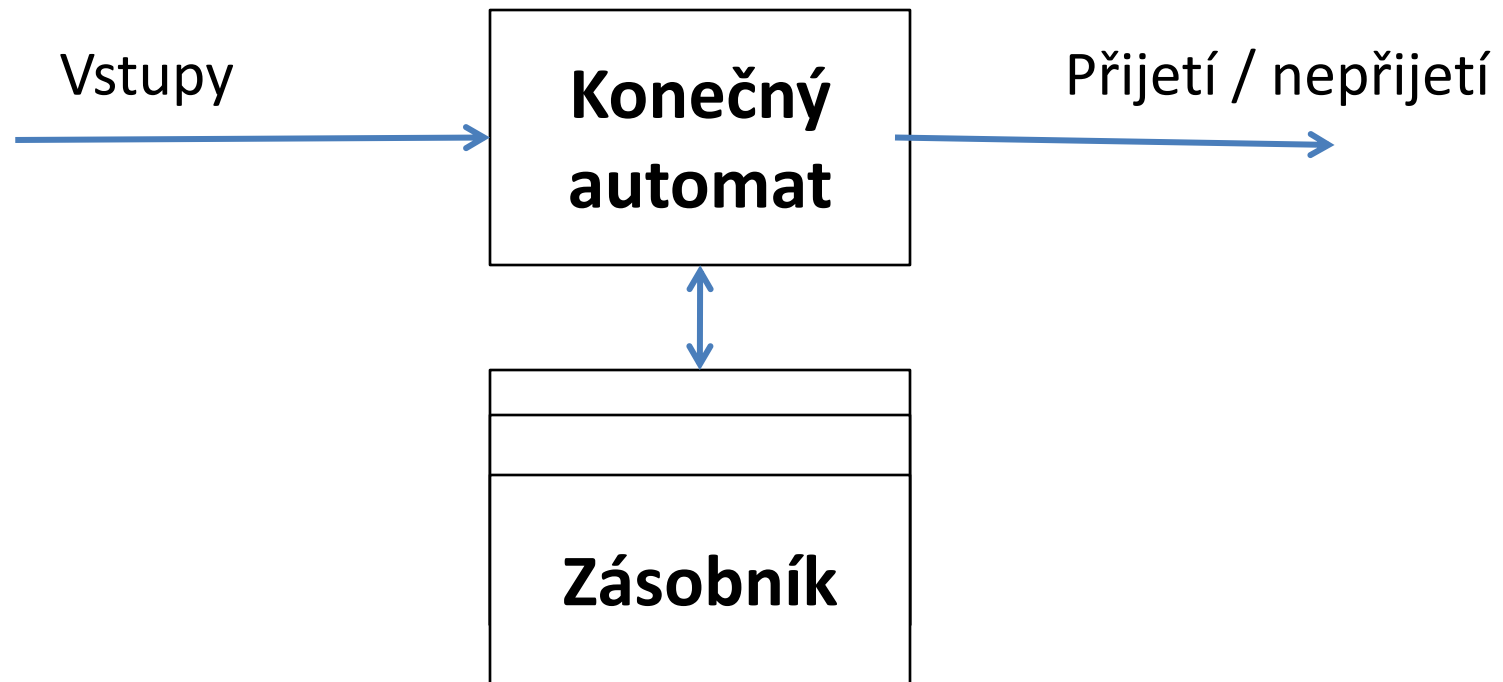
Přednáška V.

Zásobníkový automat

Zásobníkový automat – základní vlastnosti

- Mají „paměť“ - zásobník
- Paměť typu LIFO
- Rozpoznávají bezkontextové jazyky

Zásobníkový automat - princip



Zásobníkový automat – princip II.

- Konečný automat se „rozhoduje“ podle aktuálního stavu a vstupu
- Zásobníkový automat:
 - Bereme v úvahu vždy: stav, vstup, zásobník
 - Výsledkem je: stav, zásobník
 - Nemusí číst vstup
- Jeden krok zásobníkového automatu:
 1. Čtení vstupu nebo prázdný symbol místo vstupu („spontánní přechod“); čtení zásobníku
 2. Přechod do nového stavu (může být stejný jako ten předcházející)
 3. Úprava zásobníku (*vždy čte jeden symbol*, uložit lze libovolný počet – i žádný symbol)

Zásobníkový automat – formální definice

Zásobníkový automat je uspořádaná sedmice

$A = (Q, \Sigma, Y, \delta, q_0, Z_0, F)$ kde:

Q je konečná množina (*vnitřní stavy*)

Σ je konečná množina (*vstupní abeceda*)

Y je konečná množina (*zásobníková abeceda*)

δ je zobrazení - přechodová funkce:

$$Q \times (\Sigma \cup \{\varepsilon\}) \times Y \rightarrow (Q \times Y^*)$$

$q_0 \in Q$ (*počáteční stav*)

$Z_0 \in Y$ (*počáteční stav zásobníku*)

F je podmnožina Q (*rozpoznávací stavy*)

Situace ZA

- „Životní situaci“ určuje trojice **stav, vstup, zásobník**
- **Formálně: Situace** zásobníkového automatu A je uspořádaná trojice (q, u, v) :
 - $q \in Q$
 - $u \in \Sigma^*$ (zbytek vstupního slova)
 - $v \in Y^*$ (obsah zásobníku)
- Říkáme, že situace $E_1 = (q, au, Zv)$ **vede bezprostředně** na situaci $E_2 = (r, u, wv)$, jestliže $(r, w) \in \delta(q, a, Z)$. Značíme $E_1 \rightarrow E_2$.
- Říkáme, že situace E **vede** na situaci E_x , jestliže existuje posloupnost situací $E, E_1, E_2, \dots, E_n, E_x$ pro které platí:
 - $E \rightarrow E_1 \rightarrow E_2 \rightarrow \dots \rightarrow E_n \rightarrow E_x$
 - Značíme $E \rightarrow^* E_x$

Rozpoznávání jazyka

- **Přijímání koncovým stavem:** rozpoznávaný jazyk tvoří taková slova, po jejichž přečtení je automat v koncovém (rozpoznávacím) stavu. Označujeme $L(A)$.
- **Přijímání prázdným zásobníkem:** rozpoznávaný jazyk tvoří taková slova, po jejichž přečtení má automat prázdný zásobník. Označujeme $N(A)$.

Rozpoznávání jazyka II.

- Zásobníkový automat může rozpoznávat dva jazyky: $L(A)$ a $N(A)$.
- Rozpoznávání zásobníkem:
 - Nezajímá nás, v jakém stavu automat skončil!
 - Proto se často setkáme s automatem, který „nemá“ koncové stavy – $F = \emptyset$

Souvislost „L“ a „N“ I.

- **Věta:** Pro každý zásobníkový automat A1 existuje zásobníkový automat A2 takový, že:

$$N(A1) = L(A2)$$

(Neformálně: Když jde jazyk rozpoznat do prázdného zásobníku, najdeme i automat, co to bude umět do koncového stavu.)

Důkaz – princip: cesta od A1 k A2 – „přidáme“ koncový stav:

1. Zás. abecedu A1 rozšíříme o nový symbol, který bude na „dně“ zásobníku. Stavy A1 rozšíříme o nový stav – jediný rozpoznávací.
2. Zpracování slova stejné
3. Z libovolného z původních stavů přecházíme do rozpoznávacího stavu, jakmile narazíme na dno zásobníku.

Souvislost „L“ a „N“ II.

- **Věta:** Pro každý zásobníkový automat A1 existuje zásobníkový automat A2 takový, že:

$$L(A1) = N(A2)$$

(Neformálně: Když jde jazyk rozpoznat do konc. stavu, najdeme i automat, co to bude umět do prázdného zásobníku.)

Důkaz – princip: cesta od A1 k A2 – „přidáme“ koncový stav:

1. Zás. abecedu A1 rozšíříme o nový symbol, který „podloží zásobník“ - chrání zásobník proti předčasnému vyprázdnění.
2. Zpracování slova stejné
3. V původních koncových stavech smažeme zásobník.

ZA a bezkontextové jazyky I.

- **Věta:** Každý bezkontextový jazyk je přijímán zásobníkovým automatem do prázdného zásobníku.
- **Důkaz** – princip sestavení automatu z bezkontextové gramatiky:
 1. Vstupní abeceda terminály, zásobníková abeceda terminály a neterminály.
 2. Přepisovací pravidla gramatiky – přechodová funkce.
 3. Přidáme pravidlo pro „krácení“: stejný terminál na vstupu i vrcholu zásobníku.
 4. Stav je jen jeden (není podstatný).

ZA a bezkontextové gramatiky II.

- **Věta:** Ke každému zásobníkovému automatu lze sestavit bezkontextovou gramatiku generující jazyk přijímaný prázdným zásobníkem.
- **Důkaz:**
 - Má-li ZA jeden stav, je to analogické k předchozímu kroku.
 - Má-li ZA více stavů, nejprve jej převedeme na jednostavový.