3.3. Abstraktní automaty

V rámci této podkapitoly se budeme zabývat pouze dvěma typy abstraktních automatů : automatem konečným a automatem zásobníkovým - v jejich provedení bez výstupu . Cílem podkapitoly je syntéza nejjednodušších případů automatů , jako modelů syntaktických analyzátorů formálních jazyků.

Def.3.3.1.: Konečným automatem rozumíme pětici

$$A_k = (Q , X , \delta , q_0 , F) , \qquad (13.3)$$

kde $Q = \{q_0,\dots,q_n\}$ je konečná množina vnitřních stavů ,

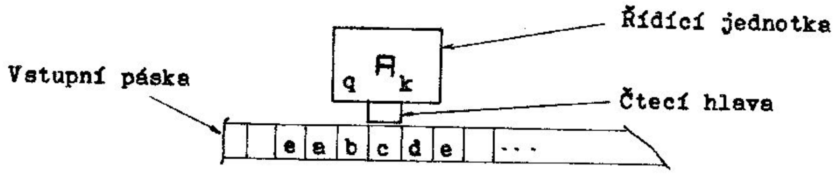
X je konečná vstupní abeceda ,

δ je tzv. přechodová funkce , obvykle typu $\delta : Q \times (X \cup \{e\}) \longrightarrow \mathcal{B}(Q)$,

q_0 je počáteční stav automatu ,

F je množina koncových stavů , $F \subseteq Q$. ∎

Omezíme se na automaty,jejichž vstupní abeceda je abeceda symbolů.
Základní představa činnosti konečného automatu jako syntaktického analyzátoru slov některého formálního jazyka vychází z přístupné myšlenkové konstrukce znázorněné na obr.9.3.



Obr.9.3.

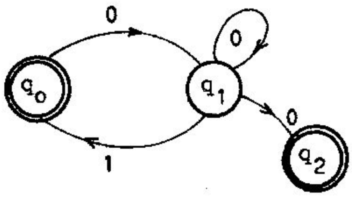
Automat zpracovává slovo w (zapsané sekvenčně na "vstupní pásce") .Slovo w je od ostatních předchozích slov odděleno alespoň jedním prázdným symbolem e (a je tedy také alespoň jedním prázdným symbolem e ukončeno). Automat sice prázdný symbol e zpravidla nezpracovává v tom smyslu,že by pomocí přechodové funkce změnil vnitřní stav,ale předpokládá se,že prázdný symbol respektuje (rozpoznává) , dokonce i to ,jestli je e na začátku nebo na konci slova w.Po rozpoznání e před prvním symbolem slova w se konečný automat nastaví čtecí hlavou nad první symbol a dále pak vykonává opakovaně sled dvou operací: podle vstupního symbolu a podle vnitřního stavu přechodová funkce δ nastaví nový vnitřní stav ; automat se přesune čtecí hlavou nad další vstupní symbol (obvykle doprava).Až do zastavení. Konečný automat se zastaví buď již někde uvnitř slova w (není definován přechod do dalšího stavu) nebo na konci slova po rozpoznání e .
O tom,zda analyzované slovo je ve smyslu jistého formálního jazyka korektní ,rozhoduje varianta zastavení spolu s druhem stavu automatu v situaci zastavení.Ze čtyř logicky přípustných možností je analyzované slovo ko-

rektní právě tehdy, jestliže automat zastaví po průchodu celým slovem w a nachází se v některém z koncových stavů $q \in F$.

Příklad 18.3.:

Je dán konečný automat $A_k = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_0, q_2\})$ s přechodovou funkcí δ :

$$\delta(q_0,0) = q_1 ,$$
$$\delta(q_1,1) = q_0 ,$$
$$\delta(q_1,0) = q_1 ,$$
$$\delta(q_1,0) = q_2 .$$



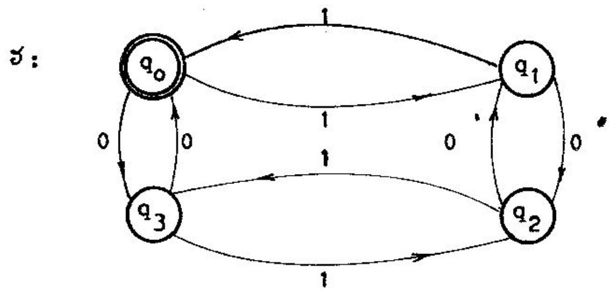
(Přechodová funkce může být zadána pouze diagramem s označenými koncovými stavy.)

Snadno se přesvědčíme, že slova w_1=e , w_2=0101 , w_3=001 , w_4=00100 automat přijímá jako korektní.

Předložený automat je ovšem nedeterministický . Vidíme to z posledních dvou řádků přechodové funkce. Stejný vstupní symbol a stejný vnitřní stav vedou k různým stavům. Důsledkem toho je skutečnost, že k některému slovu w existují dvě různé analýzy, z nichž jedna slovo přijímá jako správné a druhá jako nesprávné. Tuto situaci posuzujeme ovšem benevolentně : slovo w je korektní, jestliže nedeterministický konečný automat může nalézt alespoň jedinou korektní analýzu slova w. (Znamená to ovšem několikerý průchod slovem w.)

Příklad 19.3.:

Je dán konečný automat $A_k = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_0\})$ s přechodovou funkcí δ na obr.10.3.



Obr.10.3.

Automat rozpoznává jako korektní slova typu : w_1 = 000011 , w_2 = 1001 , ...
Automat je deterministický.

Množinu všech slov, které konečný automat analyzuje jako korektní označme $L(A_k)$. Ukazuje se, že množina $L(A_k)$ má úzký vztah k formálnímu jazyku generovanému regulární gramatikou. Obecně platí velmi silné tvrzení : Ke každé regulární gramatice lze sestrojit konečný automat tak, že $L(G)=L(A_k)$.

Syntéza konečného automatu k regulární gramatice

Je dána pravá regulární gramatika $G = (N,T,S,P)$ s množinou pravidel P $p = (L,R)$, kde $L \in N$, R je typu aA nebo e, $a \in T$, $A \in N$.

Konečný automat $A_k = (Q,X,\delta,q_0,F)$, pro který platí $L(A_k) = L(G)$ sestrojíme v následujících krocích :

1. $Q = N$,
2. $X = T$,
3. Pro každé pravidlo typu $A \Rightarrow aB$ z P platí : $B \in \{\delta(A,a)\}$.
4. $q_0 = S$.
5. $F = \{B \mid \text{pro každé pravidlo typu } B \Rightarrow e \text{ z } P\}$. ∎

(Zkonstruovaný automat vychází obecně nedeterministický.)

Příklad 20.3.:

Je dána pravá regulární gramatika $G = (\{B,C,S\}, \{0,1\}, S, P)$ kde P obsahuje pravidla

$p_1 : S \Rightarrow 0B \mid e$, $\qquad p_3 : C \Rightarrow e$.

$p_2 : B \Rightarrow 0B \mid 1S \mid 0C$,

Sestrojte konečný automat A_k, pro který platí $L(A_k) = L(G)$.

Postupujeme podle kroků popsané konstrukce :

1. $q_0 = S$, $q_1 = B$, $q_2 = C$.
2. $X = \{0,1\}$.
3. $\delta(q_0,0) = q_1$,

   $\delta(q_1,0) \in \{q_1,q_2\}$,

   $\delta(q_1,1) = q_0$,
4. $q_0 = S$.
5. $F = \{q_0,q_2\}$.

Diagram přechodové funkce δ je znázorněn v příkladu 18.3.

Def.3.3.2.: Zásobníkový automat je reprezentován jako sedmice

$$A_z = (Q, X, Z, \delta, q_0, Z_0, F), \tag{14.3}$$

kde Q je konečná množina vnitřních stavů,

X je konečná abeceda vstupních symbolů,

Z je konečná abeceda zásobníku,

δ je přechodová funkce , jako zobrazení

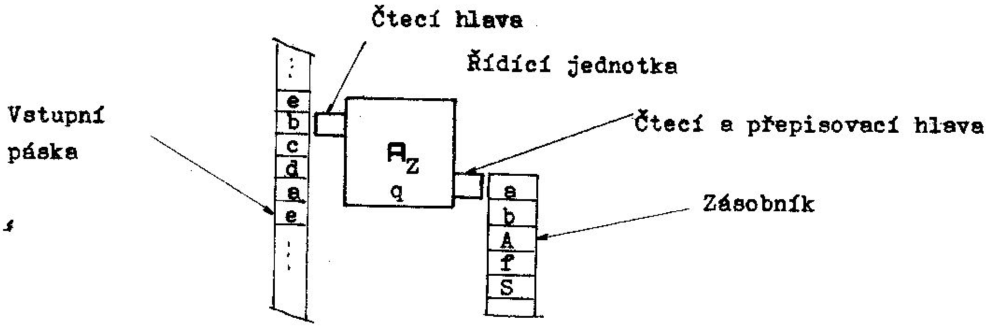
$$\delta : Q \times (X \cup \{e\}) \times Z \longrightarrow \mathcal{B}(Q \times Z^{*}) , \tag{15.3}$$

q_0 je počáteční stav automatu ,

Z_0 je počáteční symbol zásobníku ,

F je množina koncových stavů , $F \subseteq Q$. ∎

Činnost zásobníkového automatu se obvykle vysvětluje pomocí myšlenkového modelu znázorněného na obr.11.3.Vlastní analýza slov je v principu obdobou funkce konečného automatu: automat startuje od prázdného symbolu na počátku slova w , z výchozího stavu q_0 a od počáteční značky zásobníku Z_0 (na "dně" zásobníku).



Obr.11.3.

Přechodová funkce zásobníkového automatu je ovšem podstatně zajímavější: trojici (okamžitý stav , vstupní symbol , symbol na vrcholu zásobníku) přiřazuje dvojici (následující stav , řetěz zásobníkových symbolů)."Manipulace"se zásobníkem,kterou v každém kroku automatu přechodová funkce provádí,vypadá tak,jakoby z vrcholu zásobníku byl odňat symbol a byl nahražen řetězem symbolů.Protože definicí 3.3.2. není předepsána disjunktnost množin X a Z , lze tuto operaci vykládat "lidsky" také tak,že automat některé symboly do zásobníku "odkládá" a zpětně si zase některé "vybírá". Také rozpoznávání korektnosti analyzovaného slova w je důmyslnější. Opět v něm hraje roli zastavení automatu , dále pak stav automatu , ale navíc i obsah zásobníku.Podle toho pak rozlišujeme dva typy zásobníkových automatů (ale také dva typy jazyků přijímaných zásobníkovými automaty) :

$A_Z(0)$ ... w je korektní,jestliže automat zastaví po průchodu celým slovem, v některém (nikoli nutně koncovém) stavu a po úplném vyprázdnění zásobníku (tj. včetně Z_0) ,

$A_Z(F)$ ... w je korektní,jestliže automat zastaví po průchodu celým slovem, v některém z koncových stavů a s nikoli nutně prázdným zásobníkem.

Obecně platí,že jazyk přijímaný automatem $A_Z(0)$ nemusí být přijímaný automatem $A_Z(F)$ - a také opačně.

Příklad 21.3.:

Je dán zásobníkový automat $A_Z(0)$:

$$A_Z(0) = (\{q_1,q_2\} , \{0,1\} , \{Z_o,A\} , \delta , q_1 , Z_o , \{\emptyset\})$$

s přechodovou funkcí δ :

1. $\delta(q_1,0,Z_o) = (q_1,AZ_o)$,
2. $\delta(q_1,0,A) = (q_1,AA)$,
3. $\delta(q_1,1,A) = (q_2, e)$,
4. $\delta(q_2,1,A) = (q_2, e)$,
5. $\delta(q_2,e,Z_o) = (q_2,e)$.

Automat je deterministický. Prověřme jeho funkci na analýze slova $w = 0011$.

Krok číslo	Stav	Vstupní symbol	Vrchol zásobníku	Obsah zásobníku	Přechod číslo
1	q_1	0	Z_o	Z_o	1.
2	q_1	0	A	A Z_o	2.
3	q_1	1	A	A A Z_o	3.
4	q_2	1	A	A Z_o	4.
5	q_2	e	Z_o	Z_o	5.
6	q_2	e	e	e	-

Příklad 22.3.:

Je dán zásobníkový automat $A_Z(F)$:

$$A_Z(F) = (\{q_o,q_1,q_2\} , \{a,b\} , \{Z_o,a,b\} , \delta , q_o , Z_o , \{q_2\})$$

s přechodovou funkcí :

1. $\delta(q_o,a,Z_o) = (q_o,aZ_o)$,
2. $\delta(q_o,b,Z_o) = (q_o,bZ_o)$,
3. $\delta(q_o,a,a) \in \{(q_o,aa),(q_1,e)\}$,
4. $\delta(q_o,b,b) \in \{(q_o,bb),(q_1,e)\}$,
5. $\delta(q_o,a,b) = (q_o,ab)$,
6. $\delta(q_o,b,a) = (q_o,ba)$,
7. $\delta(q_1,a,a) = (q_1, e)$,
8. $\delta(q_1,b,b) = (q_1, e)$,
9. $\delta(q_1,e,Z_o) = (q_2, e)$.

Automat je <u>nedeterministický</u>. Ukažme postup analýzy slova $w = aabbaa$.

Krok číslo	Stav	Vstupní symbol	Vrchol zásobníku	Obsah zásobníku	Přechod číslo
1	q_0	a	Z_0	Z_0	1
2	q_0	a	a	a Z_0	3
3	q_0	b	a	a a Z_0	6
4	q_0	b	b	b a a Z_0	4
5	q_1	a	a	a a Z_0	7
6	q_1	a	a	a Z_0	7
7	q_1	e	Z_0	Z_0	9
8	q_2	e	e	e	-

Třída formálních jazyků generovaných bezkontextovými gramatikami je neobyčejně členitá. Jako modelů syntaktických analyzátorů těchto jazyků se vesměs předvádějí různé modifikace zásobníkových automatů.
Procvičíme pouze nejjednodušší konstrukci zásobníkového automatu k bezkontextové gramatice.

Syntéza zásobníkového automatu k dané bezkontextové gramatice

Je dána bezkontextová gramatika $G = (N,T,S,P)$, s pravidly $p = (L,R)$.
Zásobníkový automat $A_Z(0) = (Q , X , Z , \delta , q_0 , Z_0 , \{\emptyset\})$, pro který platí $L(A_Z(0)) = L(G)$, sestrojíme v následujících krocích :

1. $Q = \{q_1\}$,
2. $X = T$,
3. $Z = N \cup T \cup \{e\}$,
4. Pro všechna $p \in P$ sestrojíme základní přechody funkce tak, že
$$\delta(q_1,e,L) = (q_1,R) \quad . \qquad (16.3)$$
5. Ke každému terminálu $t \in T$ sestrojíme tzv. mazací přechod
$$\delta(q_1,t,t) = (q_1,e) \quad . \qquad (17.3)$$
6. $q_0 = q_1$.
7. $Z_0 = S$. ∎

Příklad 23.3.:

Je dána gramatika $G = (\{S,A\} , \{b,c,d\} , S , P)$ s pravidly

$p_1 : S \Rightarrow dSA \mid bAc$,

$p_2 : A \Rightarrow dA \mid c$.

Sestrojte zásobníkový automat $A_Z(0)$ tak,aby platilo $L(A_Z(0)) = L(G)$!

Postupujeme podle kroků popsané konstrukce :

1. $Q = \{q_1\}$,
2. $X = \{b,c,d\}$,
3. $Z = \{S,A\} \cup \{b,c,d\} \cup \{e\}$,
4. Základní přechody funkce δ :

$$\delta(q_1,e,S) \in \{(q_1,dSA),(q_1,bAc)\} ,$$

$$\delta(q_1,e,A) \in \{(q_1,dA),(q_1,c)\} .$$

5. Mazací přechody funkce δ :

$$\delta(q_1,b,b) = (q_1,e) ,$$

$$\delta(q_1,c,c) = (q_1,e) ,$$

$$\delta(q_1,d,d) = (q_1,e) .$$

6. $q_o = q_1$.
7. $Z_o = S$.

Sestrojený zásobníkový automat je nedeterministický . Pro danou gramatiku lze sestrojit také deterministický zásobníkový automat přijímající L(G). To ale neplatí obecně pro libovolnou bezkontextovou gramatiku.

Kontrolní otázky a úlohy

1. Promyslete algoritmickou realizaci konečného a zásobníkového automatu.Srovnejte s algoritmem vyčíslujícím hodnoty některé číselné funkce např.: $y = 4x + e^x$.
2. Konečný i zásobníkový automat jsme uvedli jako automaty bez výstupu. Připojíme-li k nim tzv. výstupní funkci $\lambda : Q \times X \longrightarrow \mathcal{B}(Y)$, resp. $\lambda : Q \times \{X \cup \{e\}\} \times Z \longrightarrow \mathcal{B}(Y)$,kde Y je konečná výstupní abeceda , získáme jistý typ diskrétních regulátorů. Zařaďte kombinační a sekvenční logické automaty,které znáte.
3. Pro třídu konečných automatů platí,že pro každý nedeterministický A_k lze sestrojit deterministický A_k , který přijímá tentýž regulární jazyk L(G). Naznačte formální matematické důvody,proč tato užitečná

vlastnost neplatí také pro třídu zásobníkových automatů.

4. Tak jako je snahou využít metody dokazování ekvivalence formálních gramatik, aniž bychom museli generovat celý formální jazyk, jsou vyvíjeny metody důkazu ekvivalence abstraktních automatů, aniž bychom prokazovali identitu přijímaných formálních jazyků. Vysvětlete podstatu těchto metod.

5. Sestrojte model syntaktického analyzátoru pro gramatiku G_3 z příkladu 16.3. Na několika slovech formálního jazyka prověřte jeho funkci.

6. Sestrojte model syntaktického analyzátoru pro formální jazyk generovaný gramatikou G definovanou v úloze 5. podkapitoly 3.2.2.

7. Určete obecné strukturní formule pro slova formálních jazyků rozpoznávaných konečnými automaty v příkladech 18.3. a 19.3.

8. Sestrojte alespoň dvě ekvivalentní gramatiky formálního jazyka přijímaného automatem z příkladu 22.3.!

9. Sestrojte zásobníkový automat pro bezkontextovou gramatiku zadanou v úloze 7. podkapitoly 3.2.2. Prověřte korektnost předepsaných slov.

10. Je dána gramatika $G = (\{S,A,B\}, \{a,b,c\}, S, P)$, kde P obsahuje pravidla :

$$p_1 : S \Rightarrow bA \mid cS \quad ,$$

$$p_2 : A \Rightarrow a \mid bS \mid cBB \quad ,$$

$$p_3 : B \Rightarrow bB \mid aAA \mid a \quad .$$

Sestrojte několik slov jazyka L(G) a pokuste se odhadnout obecný tvar. K dané gramatice sestrojte model syntaktického analyzátoru a prověřte vygenerovaná slova jazyka L(G) !