MO-21 Gramatiky, jazyky a prekladače.

1/ Základné pojmy

Abeceda je ľubovoľná neprázdna konečná množina, ktorej prvky nazývame **písmená**. Konečné postupnosti písmen nazývame slová, Prázdna postupnosť písmen sa nazýva prázdne slovo a označujeme ju ε. **Dĺžka slova** $w=a_1....a_n$ je počet písmen v slove w=n. Ak $u= a_1....a_n$ a $v= b_1....b_m$ tak $u.v= a_1....a_n b_1....b_m$ je zreťazenie slov.

Množinu všetkých slov v abecede Σ označujeme Σ^*. **Jazykom v abecede** Σ rozumieme ľubovoľnú množinu slov v abecede Σ tj. ľubovoľnú podmnožinu množiny Σ^*.

Jazyk definovaný gramatikou G je množina L(G) (tj, jazyk nad gramatikou G) všetkých vetných foriem v terminálovej abecede.

Pr 1. Nech $\Sigma=\{a,b,c\}$. Potom $\Sigma^*=\{\varepsilon, a,b,c,aa,bb,.,aba, .\}$. Jazykmi sú napr.: ε, $\{a\}$, $\{abaa, bbab\}$, $\{a^n, b^n, c^n\}$.

Základné operácie s jazykmi sú: zjednotenie, prienik, rozdiel, doplnok, zreťazenie.

Jednou z možností, ako opísať jazyk, je udať spôsob, ktorým sa dajú vytvoriť, vygenerovať, všetky slová tohto jazyka , tj. udať gramatiku.

2/ Gramatiky

DEF: Gramatika je štvorica $G=(N, T, P, \delta)$, kde N, T – sú abecedy neterminálnych, terminálnych symbolov, P je konečná množina pravidiel a δ je počiatočný symbol.

Pr 2. G1=({A,B}, {a,b}, P, A). P: 1/ $A \to aB$, 2/ $A \to BBA$, 3/ $B \to \varepsilon$, 4/ $B \to aA$, 5/ $B \to aba$. Nájdite L(G1).

Pr 3. G2=({A,B}, {a,b}, P, A). P: 1/$A \to aBb$, 2/$aB \to aaBb$, 3/$B \to \varepsilon$. Nájdite L(G2).

Pr 4. G3=({E, T, F}, {a, +, *, (,)}, P, E)

P: 1/ $E \to E + T \mid T$,

2/ $T \to T * F \mid F$

3/ $F \to (E) \mid a$

Nájdite L(G3).

Gramatika je

regulárna, ak sú všetky jej pravidlá tvaru $A \to aB$ alebo $A \to b$, kde $A,B \in N$, $a,b \in T$.

bezkontextová, ak sú všetky jej pravidlá tvaru $A \to b$, kde $A \in N$, $b \in T$.

kontextová, ak pre každé pravidlo $u \to$ platí $|u| <= |v|$, tj, dĺžka slova u je menšia alebo rovná v

neobmedzená, ak na pravidlá nie sú kladené žiadne požiadavky

Potom poznáme **triedy jazykov** regulárnych, bezkontextových, kontextových, neobmedzených.

Druhým spôsobom na „opísanie" vlastností nového jazyka je navrhnúť (udať) zariadenie, ktoré by bolo schopné rozhodovať o slovách, či patria (nepatria) do daného jazyka. V ďalšom sa budeme zaoberať takýmito zariadeniami.

3/ Konečné automaty - KA

DEF: Deterministický konečný automat je pätica $A= (K, \Sigma, \delta, q_0, F)$, kde K je konečná množina stavov, Σ je vstupná abeceda, δ je prechodová funkcia δ: $K x \Sigma \Rightarrow K$, q_0 je počiatočný stav, $F \subseteq K$ je množina koncových stavov.

Konečný automat si môžeme predstavovať ako zariadenie s konečnostavovou riadiacou jednotkou a vstupnou páskou, na ktorej je jedna čítacia hlava a tá sa môže posúvať len zľava doprava. Zariadenie pracuje v taktoch, pričom sa v každom takte (kroku) na základe stavu riadiacej jednotky a čítaného symbolu zmení stav riadiacej jednotky a pohne čítacou hlavou doprava na ďalšie písmeno vstupného slova.

DEF: Konfigurácia konečného automatu A je dvojica (q, w), kde $q \in K$, $w \in \Sigma$. Skutočnosť, že automat je v konfigurácii (q, w) znamená, že automat je v stave q a w je neprečítaná časť slova, pričom hlava je na prvom písmene zostávajúceho slova.

DEF: Krok KA je relácia $(q, aw) \Rightarrow (p, w)$, ak $\delta(q, a) = p$, kde $a \in \Sigma$, $w \in \Sigma^*$.

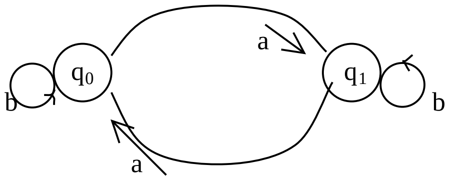
DEF: Jazyk rozpoznávaný KA je množina $L(A) = \{ w \mid (q_0, w) \Rightarrow (q, \varepsilon)\ q \in F\}$. Postupnosť konfigurácií nazývame výpočet automatu na slove w.

Pr5. Nech $A=(\{q_0, q_1\}, \{a,b\}, \delta, q_0, \{q_1\})$, kde $\delta(q_0, a)=q_1$, $\delta(q_0, b)=q_0$, $\delta(q_1, a) =q_0$, $\delta(q_1, b) =q_1$.

Prechodové funkcie je možné zadať ešte pomocou:
1/ Prechodovej tabuľky

δ	a	b
q_0	q_1	q_0
q_1	q_0	q_1

2/ Prechodový diagram



Výpočet automatu pre slovo aababb je: $(q_0, aababb) \Rightarrow (q_1, ababb) \Rightarrow (q_0, babb) \Rightarrow (q_0, abb) \Rightarrow (q_1, bb) \Rightarrow (q_1, b) \Rightarrow (q_1, \varepsilon)$ a keďže q1 je povolený koncový stav, slovo aababb je akceptované.

Pr6. Zistite, či bude akceptované slovo **aabb** pomocou KA z príkladu 5.

Pr7. Akú množinu slov rozpozná automat $A = (K, \Sigma, \delta, q_0, F)$, kde $K=\{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F=\{q_0\}$ a prechodová funkcia je daná:

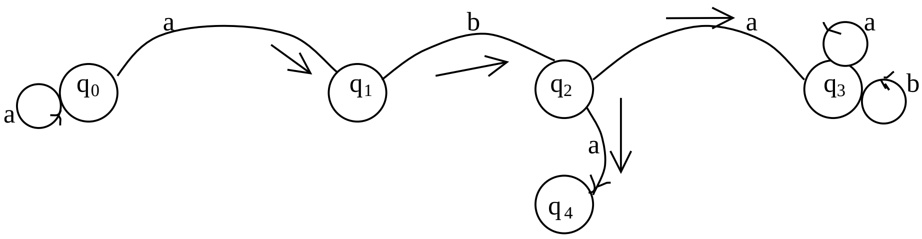
$\delta(q_0, a) =q_0$,$\delta(q_0, b) =q_1$,
$\delta(q_1, a) =q_2$, $\delta(q_1, b) =q_1$,
$\delta(q_2, a) =q_1$, $\delta(q_2, b) =q_2$

Pr8. Zostavte automat, ktorý rozpoznáva množinu všetkých slov v abecede {a, b}, ktoré obsahujú párny počet písmen **a** a nepárny počet písmen **b**.

Nedeterministické konečné automaty

Definícia je skoro rovnaká, rozdiel je v prechodovej funkcii. Táto neurčuje nasledujúci stav jednoznačne, ale udá množinu stavov, do ktorých sa môže automat dostať. Zmena je aj v tom (vzhľadom na argument ε-prázdne slovo), že automat môže zmeniť stav aj bez posunu hlavy (ak prečítal ε). Navyše sa podľa tejto definície môže dostať automat do stavu, keď nemôže vykonávať žiadnu činnosť (sťaby zmrzol???).
Čiže výpočet končí vtedy, **keď existuje aspoň jeden výpočet na slove w, zabezpečujúci koncový stav.**

Pr9. Nech $A=(\{q_0, q_1, q_2, q_3, q_4\}, \{a,b\}, \delta, q_0, \{q_1, q_3\})$, kde
$\delta(q_0, a) =\{q_0, q_1\}$,$\delta(q_0, b)=\Phi$,
$\delta(q_1, b)=\{q_2\}$,
$\delta(q_2, a)=\{q_3, q_4\}$, $\delta(q_2, b) =\Phi$,
$\delta(q_3, a) =\delta(q_3, b) =\{q_3\}$,
$\delta(q_4, a) =\delta(q_4, b) =\Phi$, $\delta(q_4, \varepsilon) =\Phi$,



Existujú dva výpočty na slove **aba**, ale len jeden končí v koncovom stave.

Na to, aby v nedeterministickom KA slovo **w nepatrilo do L(A),** nesmie existovať výpočet na **w**, alebo všetky výpočty musia končiť v stavoch rôznych od koncových!!!

DEF: Nedeterministický zásobníkový automat je sedmica A= (K, Σ, Γ , δ, q_0, Z_0, F), kde K je konečná množina stavov, Σ je vstupná abeceda, Γ je zásobníková abeceda, δ je prechodová funkcia δ:Kx($\Sigma \vee \varepsilon$) x $\Gamma \Rightarrow 2^K$, q_0 je počiatočný stav, $Z_0 \in \Gamma$ je začiatočný symbol v zásobníku , F$\subseteq$ K je množina koncových stavov.
Zásobníkový automat si môžeme predstaviť ako konečný automat, ktorý má k dispozícii pomocnú pamäť vo forme zásobníka (tj. vždy vie, čo si naposledy uchoval). Pozor, hovoríme o nedeterministickom automate.

DEF: Konfigurácia zásobníkového automatu je trojica (q, w, γ), kde q $\in$K, w$\in\Sigma^*$, $\gamma\in\Gamma^*$. Konfigurácia reprezentuje skutočnosť, že automat je v stave q, zostatok vstupu je w a obsah zásobníka je γ. Čítacia hlava je na prvom písmenku slova w a dostupný symbol v zásobníku je posledný symbol γ.

Pr10. Nech A=({q_0 ,q_1 ,q_2},{a,b},{Z_0, Z}, δ, q_0, Z_0, q_2), kde: $\delta(q_0, a, Z_0) = \{(q_0, Z_0Z)\}$,
$\delta(q_0, a, Z) = \{(q_0, ZZ)\}$,
$\delta(q_0, \varepsilon, Z) = \{(q_1, Z)\}$,
$\delta(q_1, b, Z) = \{(q_1, \varepsilon)\}$,
$\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$,
Vykonajte výpočet automatu na slove **aaabbb.**

4/ Turingove stroje a lineárne ohraničené automaty.
Na konečné a zásobníkové automaty sa môžeme pozerať ako na modely reálnych počítačov. Sú však napriek výhodám pre bežné použitie dosť obmedzujúce. Nasledujúci typ automatu, ktorý podľa všeobecne uznávanej hypotézy (Turingova teória) dokáže to, čo skutočný počítač sa nazýva Turingov stroj. Môžeme si ho predstaviť ako konečný automat, ktorý môže svojou hlavicou nielen čítať, ale aj zapisovať a vie ňou pohybovať v dvoch smeroch. Okrem toho môže vstupnú pásku podľa potreby ľubovoľne predlžovať smerom doprava.

Def: Turingov stroj je šestica A= (K, Σ, Γ , δ, q_0, F), kde K je konečná množina stavov, Σ je vstupná abeceda, Γ je abeceda páskových symbolov, δ je prechodová funkcia δ:Kx $\Gamma \Rightarrow$ K x (Γ-{B}) x {-1,1}, kde B je prázdny symbol, q_0 je počiatočný stav, F$\subseteq$ K je množina koncových stavov.
Konfigurácia Turingovho stroja je trojica (q, w, i), kde **q** $\in$ K, **w**$\in$ x (Γ-{B}) a **i**>=1 je prirodzené číslo.

Konfigurácia reprezentuje skutočnosť, že Turingov stroj je v stave q, neprázdna časť pásky je w a hlavica je práve na i-tom symbole slova w. Pri svojej práci bude Turingov stroj prechádzať z konfigurácie do konfigurácie na základe prechodovej funkcie.
PRE KAŽDÝ ALGORITMUS EXISTUJE TURINGOV STROJ !!!

Pr11. Nech A= (K, Σ, Γ , δ, q_0, F), kde K ={q_0 ,q_1 ,q_2 ,q_3 ,q_4 ,q_5},Σ={a,b},Γ ={a, b, B, X, Y}, F={q_5}, kde:

$\delta(q_0, a) = (q_1, X, 1)$,
$\delta(q_1, a) = (q_1, a, 1)$,
$\delta(q_1, Y) = (q_1, Y, 1)$,
$\delta(q_1, b) = (q_2, Y, -1)$,
$\delta(q_2, Y) = (q_2, Y, -1)$,
$\delta(q_2, X) = (q_3, X, 1)$,
$\delta(q_2, a) = (q_4, a, -1)$,
$\delta(q_4, a) = (q_4, a, -1)$,
$\delta(q_4, X) = (q_0, X, 1)$,
$\delta(q_3, Y) = (q_3, Y, 1)$,
$\delta(q_3, B) = (q_5, Y, 1)$,

Vykonajte výpočet automatu na slove **aabb.**

$(q_0, aabb, 1) \Rightarrow (q_1, Xabb, 2) \Rightarrow (q_1, Xabb, 3) \Rightarrow (q_2, XaYb, 2) \Rightarrow (q_4, XaYb, 1) \Rightarrow (q_0, XaYb, 2) \Rightarrow (q_1, XXYb, 3) \Rightarrow (q_1, XXYb, 4) \Rightarrow (q_2, XXYY, 3) \Rightarrow (q_2, XXYY, 2) \Rightarrow (q_3, XXYY, 3) \Rightarrow (q_3, XXYY, 4) \Rightarrow (q_3, XXYY, 5) \Rightarrow (q_5, XXYYY, 6)$

Skúsme sa zamyslieť nad postupom práce Turingovho stroja.
Označí si najľavejší výskyt písmena **a** symbolom **X.** Potom presunie hlavicu doprava cez všetky znaky **a** a **Y** a označí si prvý výskyt znaku **b** symbolom **Y.**
V ďalšej fáze posunie hlavicu doľava nad najľavejšie **a** a činnosť opakuje.
V prípade, že už nezostali **a**, skontroluje písmena **b** a akceptuje slovo.

Napriek vyňatiu množstva učiva, ktoré sa v tejto teórii viaže na teóriu grafov a stromy, je pravdepodobné, že si viete predstaviť predchádzajúce príklady rozkreslené ako grafy (stromy), kde sa skúša, ktorá vetva (konár) nás privedie k predpokladanému cieľu (akceptácia slova). Toto skúšanie má tiež zabehnuté spôsoby a jeden z nich je skúmanie od najľavejších vetiev.

5/ Syntaktická analýza

Dôležitou problematikou v súvislosti so štúdiom jazykov je preklad. Dá sa povedať, že vieme prekladať z jazyka do jazyka , ak vieme pre každú vetu určiť jej ekvivalent. Opísať preklad potom znamená opísať (definovať) akúsi množinu dvojíc slov.

Def:**Syntaktická analýza** slova **w** v bezkontextovej gramatike G je algoritmus, ktorý vytvorí postupnosť (čísiel) pravidiel v najľavejšom (najpravejšom) odvodení slova w.
Väčšina programovacích jazykov používa nejaký druh syntaxou riadeného prekladu. Sú dva základné spôsoby syntaktickej analýzy:
1/ *zhora nadol* – vychádzame z počiatočného neterminálu a snažíme sa prepisovať neterminály tak, aby sme dostali terminálové slovo. Základom je „uhádnuť", ktoré pravidlo treba v danom okamihu použiť. Keďže to zvyčajne nefunguje na prvý pokus, treba vymyslieť stromy odvodenia s možnými návratmi (a zabezpečiť, aby sa vedelo o všetkom čo už bolo preskúmané úspešne, či neúspešne) ((pyramídové hry))
2/ *zdola nahor* – syntaktický strom budujeme odspodu a snažíme sa nájsť to podslovo, ktoré je najviac vľavo a vzniklo prepísaním nejakého terminálu na neterminál (REDUKUJEME), až kým nedostaneme počiatočný neterminálny symbol.
Z hľadiska kompilátorov je výhodné, ak syntaktická analýza prebieha deterministicky.

6/ Kompilátory

Kedysi počítač interpretoval (pomocou HW) základné inštrukcie, písané pomocou znakov 0,1 (strojovým kódom). Uľahčenie práce priniesli vyššie programovacie jazyky, tieto však potrebujú prekladače do tvaru, ktorému rozumie procesor.

Prekladač je program, ktorý má na vstupe zdrojový text, napísaný v ľubovoľnom programovacom jazyku a na výstupe program, napísaný v inom programovacom jazyku (cieľovom jazyku). Poznáme niekoľko druhov prekladačov:
a/ **kompilátor** je prekladač, ktorého zdrojovým jazykom je programovací jazyk vyššej úrovne (Fortran, Pascal, Cobol, ...) a cieľovým jazykom je programovací jazyk nízkej úrovne (strojový kód alebo jazyk symbolických adries - JSA) .
b/ **assembler** je prekladač, ktorého zdrojovým jazykom je jazyk symbolických adries - JSA a cieľovým jazykom je strojový kód.
c/ **predprocesor –** je chápaný ako prekladač, ktorého zdrojovým aj cieľovým jazykom je programovací jazyk vyššej úrovne (nemusia byť rôzne).
d/ **interptréter** – existujú prekladače, ktoré transformujú zdrojový program do medzijazyka a ten sa dá vykonať prostredníctvom interprétra (snobol, basic)

Kompilátor je program, ktorý **transformuje zdrojový program vyššieho programovacieho jazyka** do **cieľového jazyka nízkej úrovne** (strojový kód alebo jazyk symbolických adries). Tento proces je značne komplikovaný a preto si ho rozložíme na jednotlivé fázy .

LA - lexikálny analyzátor združuje znaky (písmená, číslice,+,..) zdrojového jazyka do lexikálnych jednotiek-lexém a zadeľuje ich do skupín, ktoré navzájom súvisia. Bežnými skupinami lexém sú napr. vyhradené slová, identifikátory, operátory, oddeľovače atď.

SA - syntaktický analyzátor združuje lexémy do syntaktických štruktúr if . then .. else .. ;

GMJ - generátor medzijazyka transformuje vzniknutú štruktúru do jednoduchých inštrukcií medzijazyka (nie sú ešte špecifikované registre)

OK - optimalizátor kódu optimalizuje medzijazyk

GK - generátor kódu vytvára cieľový program už aj s prideľovaním pamäťových miest a registrov pre cieľový program (vytvára sa najťažšie)

ACH - analyzátor chýb zistí a ohlási chyby v zdrojovom programe, určia druh a miesto chyby, niektoré sa snažia chyby predvídať a opraviť

STI - spracovanie tabuliek informácií zbiera informácie o objektoch zdrojového programu a uchová v tabuľke symbolov napr. premenné, rozsahy polí.

```
           +----+
     +-----¦ LA +-----+,
     ¦     +----+     ¦
     ¦     +----+     ¦
     ¦+----¦ SA +---+ ¦
     ¦¦    +----+   ¦ |
     ¦¦      ¦      | |¦
   +----+  +----+  +----+
   ¦ACH +--¦GMJ +--¦STI ¦
   +----+  +----+  +----+
     ¦¦      ¦      ¦ ¦
     ¦¦    +----+   ¦ ¦
     ¦+----¦ OK +---+ ¦
     ¦     +----+     ¦
     ¦     +----+     ¦
     +-----¦ GK +-----+
```