Rekurzivní sestup

Thursday, May 30, 2013 8:41 AM

https://class.coursera.org/compilers/lecture/25

Top Down

Shora

Zleva doprava

Rekurzivní sestup nebo také **rekurzivní sestupný parser** postupuje shora dolů a je sestaven ze vzájemně se volajících procedur. Každá taková procedura obvykle implementuje jedno přepisovací pravidlo gramatiky. (Kromě něj do top-down parserů patří prediktivní parsery založené na LL(k) gramatikách.)

Prediktivní parser je rekurzivně sestupný parser, který nevyžaduje zpětné kroky. Je ho možné vytvořit jen pro LL(k) gramatiky, což jsou bezkontextové gramatiky, pro které existuje kladné *k*, které umožní rekurzivně sestupnému parseru rozhodnout se, které přepisovací pravidlo použít na základě *k* dalších načtených symbolů. LL(k) gramatiky vylučují mnohoznačnost a levou rekurzi. Jakákoliv bezkontextová gramatika může být transformována na ekvivalentní nelevorekurzivní gramatiku, ale odstranění levé rekurze ne vždy vede k LL(k) gramatice. Prediktivní parser běží v lineárním čase.

Rekurzivní sestup s návratem je technika určování použitého produkčního pravidla zkoušením všech pravidel. Není limitován na LL(k) gramatiky, ale nemá zaručeno skončit, pokud gramatika není LL(k). Může vyžadovat exponenciálni čas pro svůj běh.

Princip

Hlavní myšlenka je taková, že pro každý neterminál gramatiky je implementována příslušná funkce v programu.

- každému neterminálnímu symbolu *A* odpovídá procedura *A*
- tělo procedur je dáno pravými stranami pravidel pro *A*
- pravé strany musí být rozlišitelné na základě symbolů vstupního řetězce
- je-li rozpoznána pravá strana, pak v případě neterminálního symbolu vyvolá *A* proceduru pro rozpoznaný neterminální symbol, v případě terminálního symbolu, ověří *A* jeho přítomnost ve vstupním řetězci a zajistí přečtení dalšího znaku ze vstupu
- rozpoznané pravidlo analyzátor oznámí (např. jeho číslo)
- chybnou strukturu vstupního řetězce oznámí chybovým hlášením

Terminál na pravé straně je porovnán s dalším vstupním symbolem. Pokud se shodují, přejde se na další vstupní symbol a na další symbol na pravé straně. V opačném případě je nahlášena chyba.

O neterminál na pravé straně je postaráno voláním příslušné funkce. Po jejím vykonání se pokračuje dalším symbolem na pravé straně.

Pokud na pravé straně už nejsou žádné symboly, funkce končí (function returns).

Takto se postupně volají všechny funkce, až se nakonec opět ocitneme ve funkci pro startovací symbol, která byla zavolána jako první. Ta také oznamuje úspěšný průběh, pokud se prošel celý vstupní řetězec.

```
    void A() {
1)      Choose an A-production, A → X1X2 ··· Xk;
2)      for ( i = 1 to k ) {
3)          if ( Xi is a nonterminal )
4)              call procedure Xi();
5)          else if ( Xi equals the current input symbol a )
6)              advance the input to the next symbol;
7)          else /* an error has occurred */;
        }
    }
```

Pseudokód výše je nedeterministický, protože začíná volbou A-přepisovacího pravidla, které se používá blíže nepopsaným způsobem.

Příklad: http://info.lu2.name/soubory/prekl_06_604.pdf

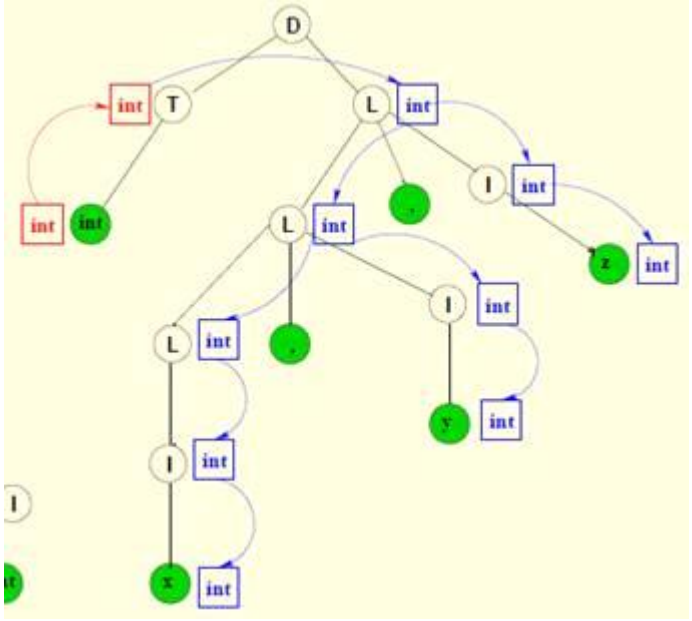
Obecně může rekurzivní sestup potřebovat backtracking, tzn. někdy je třeba se vrátit a opakovaně číst vstup. Backtracking je však potřeba zřídkakdy (dá se eliminovat volbou vhodné gramatiky - tedy zavedením jistých omezení). Kód výše neumožňuje backtracking, bylo by je nutno modifikovat – v řádce 7 se pak vrátit na řádku 1 a zvolit jiné pravidlo, popř. nahlásit chybu, když už žádné další nejde použít.

Sémantické zpracování

Při rekurzivním sestupu se může provádět také sémantické zpracování. Sémantické zpracování zahrnuje vyhodnocení atributů symbolů v derivačním stromu. Atributy = vlastnosti gramatických symbolů nesoucí sémantickou informaci (hodnota, adresa, typ, scope, spojitost mezi formálními a skutečnými parametry apod.).

Způsoby vyhodnocení:

1. procházením stromem od listů ke kořenu = **syntetizované atributy**
2. procházením stromem od rodiče k potomkovi, od staršího bratra k mladšímu = dědičné atributy (např vícenásobné deklarace v C – int x,y,z)



Je nutné doplnit procedury lex. analýzy (LA) i syntakt. analýzy (SA) takto:

- LA bude předávat s přečteným vstupním symbolem i jeho atributy.
- procedury SA pro neterminály doplnit o:
  - vstupní parametry odpovídající dědičným atributům
  - výstupní parametry odpovídající syntetizovaným atributům
  - zavést lokální proměnné pro uložení atributů pravostranných symbolů
  - před vyvoláním procedury korespondujícího neterminálu z pravé strany vypočítat hodnoty

jeho dědičných atributů
    - na konec procedury popisující pravou stranu pravidla zařadit příkazy vyhodnocující syntetizované atributy

Vlastnosti

- pro metodu rekurzivního sestupu, tj. analýza shora dolů, se používají *LL* gramatiky (levorekurzivní se nedají použít, protože by se program mohl zacyklit voláním pořád té samé procedury)
- jednoduchá *LL* gramatika je taková gramatika, kde levou stranu tvoří právě jeden neterminální symbol a kde každá pravá strana začíná terminálním symbolem
- navíc musí platit, že např. pro pravidla *A->...* jsou počáteční symboly různé
- obecná *LL* gramatika nemá omezení, ale musí pro ni existovat rozkladová tabulka