

Poznámky k prezentaci:

Github

Verzování(git)

Vzdálený repozitář(Github.com)

Rozdělená práce

Sledování postupu

Messenger

Časté osobní schůzky

- na základě konečného automatu
- Scanner dále komunikuje se syntaktickou analýzou prostřednictvím pomocné struktury , která představuje předávaný token, tato struktura obsahuje veškeré informace o konkrétním tokenu – jeho hodnotu, typ, alokovaný prostor, atd... Scanner postupně načítá jednotlivé znaky ze vstupu () dokud neprojde do koncového stavu, v tomto případě posílá hotový token, v opačném případě dochází k chybě s kódem 1.
- hlavní problém: indent, dedent
 - Pro zpracování a (odsazení) byla vytvořena pomocná struktura (indent stack), do kterého se při odsazování ukládá aktuální úroveň odsazení a počet mezer v odsazení, zároveň byla použita statická proměnná, která slouží k poznačení si, zda je aktuálně zpracováván token prvním na řádce, či nikoliv, tyto informace jsou následně využity pro odsazování a vracení se z odsazení.
- Pro ESCAPE sekvence je použita statická proměnná, díky které lze zkontrolovat, zda byl předchozí znak „\“ či nikoliv.
- pro většinu programů (bez výrazů) syntaktická analýza shoda dolů
 - vytvořili jsme si LL(1) gramatiku
 - neterminál = jedna funkce programu, potom na základě tokenů volíme, které pravidlo LL gramatiky zvolit – neočekávaný token = syntaktická chyba
- Syntaktická analýza je založená na LL gramatice, která je simulována pomocí rekurzivního sestupu
- při tokenu ID – není snadné rozhodnout, zda se jedná o výraz, přiřazení výrazu nebo přiřazení funkce
 - řešení: přednačení tokenu
- pro každý příkaz voláme příslušné funkce generátoru kódu
- Zpracování výrazů:
 - syntaktická analýza zdola nahoru
 - precendenční tabulka + funkce, která se opakovaně volá, dokud se výraz nezredukoval na výsledek

- implementace zásobníku pro ukládání operátorů a operandů
 - při každé redukci syntaktické + sémantické kontroly podvýrazů – ok = volání funkcí na generování kódu
 - jinak chyba
- Informace o proměnných a jejich typech jsou ukládány do tabulky symbolů. Stejně tak informace o funkcích, zda byly definovány, počet parametrů atp. Tyto informace jsou následně využívány ke kontrolám definic a redefinic.
 - při kontrole definic funkcí kontrolujeme, zda volaná funkce je již funkce definována dříve, navíc kontrolujeme i funkce, na kterých je tato závislá
 - pro globální proměnné ve funkci máme poznamenáno, zda byly využity, pokud se následně definuje proměnná stejného jména program skončí se sémantickou chybou