

# IFJ projekt: Překladač jazyka IFJ20

## **Tým 056, varianta II**

František Nečas, Ondřej Ondryáš, David Chocholatý

Fakulta informačních technologií VUT v Brně

Božetěchova 1/2, 612 00 Brno-Královo Pole



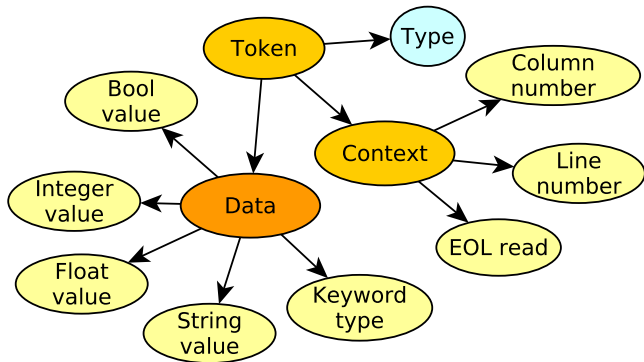
24. ledna 2021

- David Chocholatý
  - scanner, chybové zprávy na stderr, integrační testy překladače, LL(1) gramatika, Makefile, hashovací tabulka
- František Nečas
  - syntaktická analýza (rekurzivní sestup a precedenční SA), základní sémantické kontroly, optimalizace kódu, tvorba vnitřní reprezentace
- Ondřej Ondryáš
  - datové struktury pro vnitřní reprezentaci kódu (graf řízení toku a abstraktní syntaktický strom), generátor cílového kódu, kostra pro testování, testy pro scanner a parser

- Git (GitHub)
- Discord
- Virtuální schůzky
- Pull request, code review
- Párové programování

- Všechna rozšíření
  - BOOLTHEN, BASE, FUNEXP, MULTIVAL, UNARY
- Lexikální analyzátor jako konečný automat
- Jednoprůchodová syntaktická analýza
  - Rekurzivní sestup pro kostru kódu
  - Precedenční SA pro výrazy včetně přiřazení a volání funkcí
- Techniky odloženého odvozování datových typů (suplující druhý průchod SA)
- Optimalizace vnitřní reprezentace kódu
- Generování kódu – vyhodnocování na zásobníku, zkrácené vyhodnocování logických výrazů

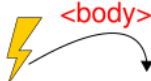
- Deterministický konečný automat
- Řešení EOL pravidel
- Struktura tokenů
- Rozšíření BASE



- Jednoprůchodová syntaktická analýza
- Implementace
  - rekurzivní sestup
  - precedenční syntaktická analýza
- Provádění přidružených sémantických akcí

- Zotavení z chyby na základě pevných klíčů (nový řádek)

```
1 package main
2
3 func foo(test float) {
4     for ; j := 5 ; {
5     }
6 }
```



```
parser: error: Line 3, col 15:
expected float64, int, string or bool keyword, got identifier
precedence_parser: error: Line 4:
expected pure expression (no definitions or assignments)
```

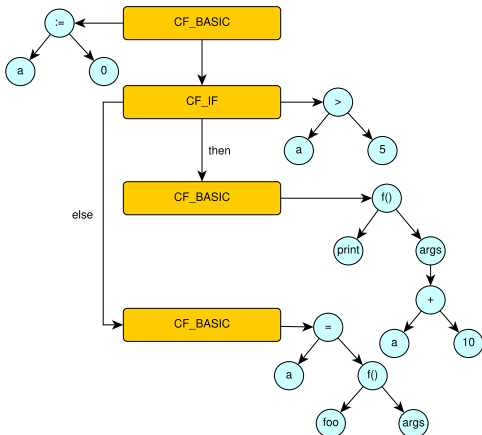
- Zpracovává výrazy, včetně přiřazení a definic
  - Čárka jako operátor s nejnižší prioritou
  - Příznak definičních pravidel
- Detekce konce výrazu (kompatibilní s Go)
- Jako sémantické akce vytváří abstraktní syntaktický strom a provádí typové kontroly



- Popis struktury programu
  - Lineární + větvení
- Metadata programu a funkcí
- Čtyři typy vrcholů reprezentujících příkazy
- Vrcholy obsahují odkazy na AST

```

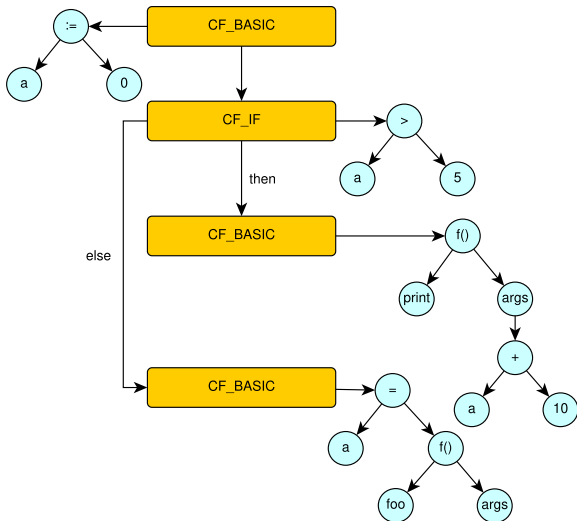
1 a := 0
2 if a > 5 {
3   print(a + 10)
4 } else {
5   a = foo()
6 }
    
```



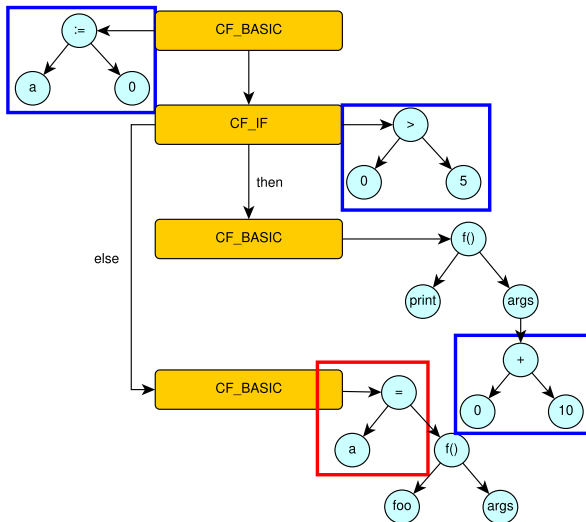
- Reprezentace výrazu, výstup precedenční SA
- Popisuje samotné výpočetní akce
- Různé typy uzlů (definice/přiřazení, volání funkcí, operace, termy, *seznam AST*)
- Zprostředkovává podstatnou část sémantických kontrol – princip **odvozování datových typů**
  - Syntaktický analyzátor při volání funkce ve výrazu ještě nemusel dojít k definici této funkce
  - Supluje význam druhého průchodu SA
    - Zpětně ověřuje správnost volání a typů

- Skládání konstant
  - Využití algebraických vlastností operací (odstranění dvojité negace)
- Propagace konstant
  - Sledování konstantních definic v rámci seznamu
  - Nahrazování proměnných za konstantu
- Odstranění mrtvého kódu
  - For cyklů s konstantní nepravdivou podmínkou
  - Else bloků vždy pravdivé podmínky
  - If bloků se vždy nepravdivou podmínkou

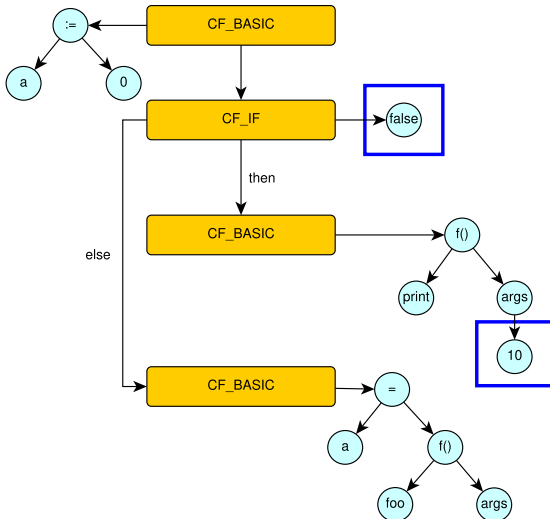
## Původní struktura kódu



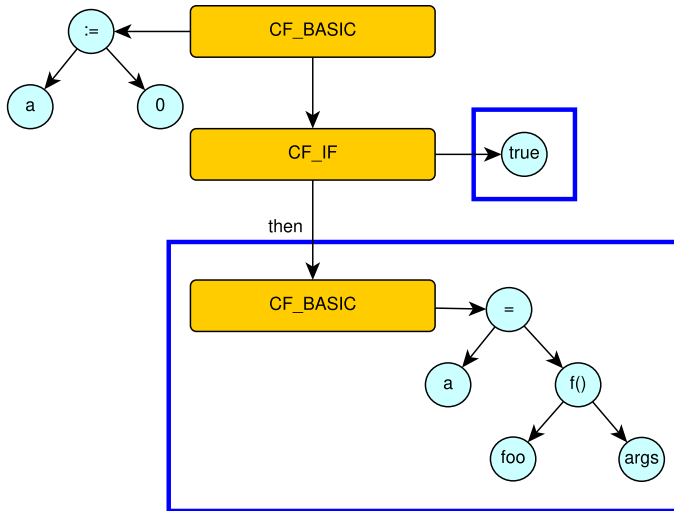
Po propagaci konstant



Po skládání konstant



Po odstranění mrtvého kódu



- Generování výrazů
  - Post-order rekurzivní průchody jednotlivými AST
  - Aritmetické výrazy generovány přímo na zásobníku
  - Logické výrazy generovány zkráceným vyhodnocováním
- Volání funkcí
  - Argumenty v dočasném rámci, návratové hodnoty na zásobníku (integrace do vyhodnocování výrazů)
  - Vestavěné funkce generovány *in-line*
- Rozsahy platnosti – svázáno s tabulkami symbolů, dekorované názvy proměnných
- Dopředné generování definic proměnných
- Počítání referencí (zahazování zbytečných definic)



- Lineárně vázaný seznam (jedno- i obousměrně vázaný)
  - Funkce, parametry a návratové hodnoty v grafu řízení toku
  - Konstanty při optimalizacích
  - Zásobníky, tabulky symbolů
- Hashovací tabulka
  - Tabulka symbolů
- Zásobník
  - Zásobník tabulek symbolů
  - Zásobník pro precedenční SA
- Strom
  - Graf řízení toku, abstraktní syntaktický strom
- Nafukovací pole
  - Dynamický řetězec pro načítání identifikátorů a řetězcových konstant

Diskuze

Děkujeme za pozornost.