

Metody programowania 2016

Lista zadań nr 7

Na zajęcia 18 i 21 kwietnia 2016

Zadanie 1 (1 pkt). Wszystkie operatory w Pascalu są podzielone na cztery grupy pod względem priorytetu (od najmniejszego do największego):

< <= > <> >= > in
+ - or
/ div mod and
not

Operatory binarne łączą w lewo. Dodatkowo unarny operator - ma taki sam priorytet, jak binarne operatory addytywne +, - i or i może pojawić się przed liczbą i nawiasem otwierającym, ale nie po operatorze binarnym, np. -a oraz -(1+2) są poprawne, a+-b zaś — nie. Gramatyka bezkontekstowa Pascala nie rozróżnia wyrażeń różnych typów, lecz traktuje wszystkie operatory jednakowo (wyrażenia niepoprawne w sensie typów są odrzucane w późniejszych fazach kompilacji). Napisz gramatykę BNF opisującą wyrażenia w Pascalu.

Dla każdego z poniższych wyrażeń w Pascalu narysuj drzewo wyprowadzenia tego wyrażenia z powyższej gramatyki oraz abstrakcyjne drzewo rozbioru tego wyrażenia.

1. $i \geq 0$
2. $(i \geq 0) \text{ and not } p$
3. $i \geq 0 \text{ and not } p$
4. $(i \geq 0) \text{ and } (x < y)$

Zadanie 2 (1 pkt). Napisz w notacji BNF jednoznaczna gramatykę opisującą wyrażenia złożone z literałów całkowitoliczbowych, identyfikatorów, nawiasów i operatorów binarnych wymienionych w poniższej tabeli.

operator	łączność	priorytet
^	w prawo	4
*	w lewo	3
+	w lewo	2
<	niełączny	1
=	niełączny	1

Zadanie 3 (1 pkt). Literał zmiennopozycyjny w Pascalu to napis nad alfabetem

$$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, e, .\}$$

następującej postaci: na początku występuje opcjonalnie znak + lub -, następnie niepusty ciąg cyfr 0...9, potem opcjonalnie część ułamkowa, tj. kropka i niepusty ciąg cyfr i na końcu opcjonalnie wykładnik, tj. znak e, opcjonalnie znak + lub - i niepusty ciąg cyfr. Co najmniej jeden napis spośród części ułamkowej i wykładnika musi wystąpić (inaczej literał będzie tzw. literałem całkowitoliczbowym). Zdefiniuj gramatykę BNF opisującą język literałów zmiennopozycyjnych w Pascalu.

Zadanie 4 (1 pkt). Zaprojektuj algorytm, który dla podanej gramatyki bezkontekstowej rozstrzyga, czy język przez nią generowany jest niepusty.

Zadanie 5 (1 pkt). Słowo x nad alfabetem $\{ (,) \}$ jest ciągiem poprawnie rozstawionych nawiasów, jeśli $\text{balance}(x) = 0$ i $\text{balance}(y) \geq 0$, dla każdego prefiksu y słowa x , gdzie $\text{balance}(\epsilon) = 0$, $\text{balance}(z()) = \text{balance}(z) + 1$ oraz $\text{balance}(z)) = \text{balance}(z) - 1$. Dla przykładu $((()())())$ jest ciągiem poprawnie rozstawionych nawiasów, zaś $((()))()$ nim nie jest. Udowodnij, że gramatyka $G = \{ \{ (,) \}, \{ S \}, S, \{ S \rightarrow (S), S \rightarrow SS, S \rightarrow \epsilon \} \}$ generuje zbiór wszystkich ciągów poprawnie rozstawionych nawiasów.

Zadanie 6 (1 pkt). Napisz gramatykę bezkontekstową definiującą język

$$L = \{ a^n b^n \mid n \in \mathbb{N} \}.$$

Przepisz ją w postaci DCG. Pokaż, jak jest ona tłumaczona przez predykat `expand_term/2`. Użyj predykatu `phrase/2` by sprawdzić, który z ciągów

""
"aabb"
"aba"

należy do języka L . Ile nawrotów wykonuje program podczas sprawdzania? Pokaż, że Twój program także poprawnie generuje wszystkie słowa należące do języka L . Przerób gramatykę DCG tak, by analiza słów działała deterministycznie (bez głębokiego nawracania). Czy nowa gramatyka może być także wykorzystana do generowania słów? Dodaj do niej akcje semantyczne obliczające liczbę wystąpień liter a w podanym słowie.

Zadanie 7 (1 pkt). Sprawdź, do czego służy predykat `arithmetic_function/1` i wykorzystaj go, by zarejestrować funkcję arytmetyczną `!/1` obliczającą silnię podanej liczby. Zdefiniuj odpowiedni predykat `!/2`. Przejrzyj tabelę priorytetów operatorów standardowych i wybierz odpowiedni priorytet dla operatora `!/1`. Uczyń go operatorem postfiksowym tak, by można było z niego korzystać w następujący sposób:

?- N is 5!.
N = 120.

Powtórz czynności dla operatora `!/1` (iloczyn liczb nieparzystych od 1 do podanej liczby). Jakie trudności tu się pojawiają?