

Sprawozdanie z zajęć laboratoryjnych z przedmiotu

Języki Formalne i Kompilatory

Wojciech Lulek index:136280

Generatory Flex i Bison – ćwiczenia Beata Jankowska

Zadanie 6.1a Zbiór danych jest niepustym zbiorem wierszy, z których każdy składa się z liczb nieujemnych bez znaku:

- a) całkowitych zapisanych w układzie dziesiętnym, np. 3256,
- b) całkowitych zapisanych w układzie szesnastkowym, np. 43C3,
- c) rzeczywistych zapisanych w układzie dziesiętnym w formacie stałopozycyjnym, np. 12.93,
- d) rzeczywistych zapisanych w układzie dziesiętnym w formacie

zmiennoprzecinkowym, z cechą koniecznie poprzedzoną znakiem, np. 23.64E+12.

Poszczególne wiersze są niepuste i mają jednorodną postać, tzn. znajdujące się w nich liczby należą do tej samej kategorii (a, b, c lub d). Każda liczba w wierszu jest poprzedzona dowolną (niezerową) liczbą spacji lub znaków tabulacji. Skonstruuj wyrażenie regularne/wzorzec definiujące opisany zbiór danych.

Wyrażenie regularne:

- a) $[0-9]^+$
- b) $0x[A-F[0-9]^*]^*$
- c) $[0-9]^+\backslash\.[0-9]\{2\}[^E]$
- d) $[0-9]^+\backslash\.[0-9]^+E\backslash[0-9]^+$

Zadanie 6.1b Zdefiniuj gramatykę bezkontekstową generującą zbiór zbiorów danych scharakteryzowanych w zadaniu 6.1a.

Mała legenda:

DEC- liczby zapisane dziesiętnie

HEX- liczby zapisane szesnastkowo

STP- liczby zapisane w układzie dziesiętnym w formacie stałopozycyjnym

STD- liczby zapisane w układzie dziesiętnym w formacie stałopozycyjnym z dopełnieniem

Alfabet: $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,x,.,E+,E-, \backslash n\}$

Symbole terminalne: $\{CYFRA, LITERA\}$

Symbole nieterminalne: S (startowy), CYFRY, LITERY, CYFRLITERA, DEC, HEX, STP, STD

Produkcje:

$S \rightarrow DEC S \mid HEX S \mid STP S \mid STD S \mid \varepsilon$

$CYFRY \rightarrow CYFRA CYFRY \mid CYFRA$

$LITERY \rightarrow LITERA LITERY \mid LITERA$

$CYFRLITERA \rightarrow CYFRA CYFRLITERA \mid LITERA CYFRLITERA \mid CYFRA \mid LITERA$

$DEC \rightarrow CYFRY \mid DEC DEC$

$HEX \rightarrow '0' 'x' CYFRLITERA \mid HEX HEX$

STP-> DEC ‘,’ DEC| STP STP

STD-> DEC ‘E-’ CYFRY| DEC ‘E+’ CYFRY| STD STD

Zadanie 6.1c Skonstruuj w języku Bison przetwornik weryfikujący poprawność budowy przedmiotowego zbioru danych i generujący na wyjściu ciąg liczb, w odpowiednich formatach, będących sumami liczb przechowywanych w kolejnych wierszach zbioru.

Kod z rozwiązaniem zadania umieszczony jest na githubie pod linkiem:

https://github.com/Zerkles/jfik_bj

Wnioski:

Rozwiązanie podpunktu pierwszego nie powinno stanowić większego problemu dla każdego kto zna przynajmniej podstawy działania wyrażeń regularnych, jedynym utrudnieniem w tym zadaniu było stworzenie takiego wyrażenia regularnego, które reaguje tylko na jeden „typ” danych i nie reaguje na inne bądź części innych wyrażeń co doprowadziłoby do fatalnych błędów.

W zadaniu 6.1b zdefiniowanie alfabetu, symboli terminalnych oraz nieterminalnych nie stanowiło większego problemu w przeciwieństwie do produkcji, których zaplanowanie i logiczne rozpisanie zajęło nieco więcej czasu.

W ostatnim zadaniu dzięki uprzednio stworzeniu języka oraz odpowiednich wyrażeń regularnych mogliśmy się zabrać wprost za programowanie. Program wychwytyjący odpowiednie sekwencje danych napisany pod program lex. Generalnie jedynym jego zadaniem jest zwracanie odpowiednich tokenów do programu napisanego pod program bison. W programie bison natomiast na samym początku mam załączone niezbędne biblioteki, zadeklarowane kontenery odpowiedniego typu przechowujące aktualną sumę wiersza podanego na wejściu oraz zdefiniowaną funkcję konwertującą liczby szesnastkowe na dziesiętne. W głównej części programu obok zdefiniowanych produkcji mamy wykonywane odpowiednie dla danego typu operacje sumowania, a ostatecznie po wykryciu końca linii wyświetlania sumy danego wiersza. Z racji braku zaimplementowanych w języku c operacji dodawania liczb szesnastkowych, są one konwertowane na liczby dziesiętne, dodawane a następnie ponownie przed wyświetleniem konwertowane do postaci hexadecymalnej.