Przecięcie języków bezkontekstowych i dopełnienie języka bezkontekstowego

Dopełnienie języka bezkontekstowego albo przecięcie dwóch języków bezkontekstowych nie musi być językiem bezkontekstowym.

Przykład: język $\{a^n b^n c^n : n \in \mathbb{N}\}$ nie jest bezkontekstowy (co można wykazać korzystając z lematu o pompowaniu). Język ten jednak jest przecięciem dwóch języków bezkontekstowych $\{a^n b^m c^m : n, m \in \mathbb{N}\}$ ¡ $\{a^n b^n c^m : n, m \in \mathbb{N}\}$

Zadanie 7: Podaj gramatykę bezkontekstową generującą język:

- {aibjck : 3i = 2j+k },

S → aX | aSccc | aaYbbcc | Y | ε
X → **Y***bc* | *ccc*
Y → *aa***Y***bbb* | *aabbb*

Praca domowa

Odwrotna notacja polska (ONP) to sposób zapisu wyrażeń, w którym najpierw podajemy argumenty, a potem operację. Jeżeli wiadomo ile argumentów mają operacje (a w przypadku operacji arytmetycznych tak jest -- mają one po dwa argumenty), to w ONP nie są potrzebne nawiasy. Na przykład, wyrażenie 2*(3+5) zapisane w ONP ma postać $2\ 3\ 5\ +\ *$. Nota bene, wyrażenie zapisane w ONP to gotowy program dla maszyny stosowej, patrz Ćwiczenia.

Napisz specyfikację (dla Lex'a i Bison'a) analizatora, który wczyta wyrażenie arytmetyczne i wypisze je w ONP.

```
%token NUM

%%

input  : /* nic */
       | input line
       ;

line   :'\n'
       | exp '\n' { printf ("\n"); }
       ;

exp    : exp '+' sum{ printf("+ ");     }
       | exp '-' sum       { printf("- ");     }
       | sum
       ;

sum    :sum '*' fct {printf("* ");      }
       |sum '/' fct {printf("/ ");      }
       |fct
       ;

fct    : NUM        {printf("%d ", $1); }
       |'(' expt ')'
       ;

%%

main()
{
  yyparse();
}

yyerror(char *s)
{
    fprintf(stderr, "%s\n", s);
}
```