

Egzamin wstępny na studia drugiego stopnia. Programowanie.

Za zadanie można otrzymać 100 punktów.

Gramatyka bezkontekstowa jest w postaci normalnej Chomsky'ego (CNF), jeżeli wszystkie produkcje mają postać: $A \rightarrow a$ lub $A \rightarrow BC$, gdzie A, B, C są symbolami nieterminalnymi, natomiast a jest symbolem terminalnym.

Część 1. (10p) Napisz gramatykę bezkontekstową G_1 z jednym symbolem nieterminalnym, która generuje zbiór palindromów nad alfabetem $\{a, b\}$ (czyli takich słów, które brzmią tak samo odczytane normalnie oraz wspak. Napisz gramatykę G_2 w postaci CNF generującą ten sam język.

Część 2. (15p) Przedstaw rekurencyjną procedurę (w pseudokodzie) sprawdzania dla gramatyki w postaci CNF zadanej przez zbiór produkcji P , czy dane słowo w należy do języka generowanego przez tę gramatykę. Skoncentruj się na elegancji i czytelności.

Część 3. (10p) Czy przedstawiona przez Ciebie rekurencyjna procedura może być podstawą dla utworzenia efektywnego parsera? Odpowiedź krótko uzasadnij.

W dalszej części wybierz jeden z następujących języków: C, C++, C#, Java, Pasca lub Python. Symbole terminalne będziemy reprezentować jako znaki.

Część 4. (8p) Przedstaw, jak w wybranym języku reprezentować słowa, symbole nieterminalne, produkcje oraz gramatykę.

Część 5. (25p) Zdefiniuj w wybranym języku funkcję `CNFParse`, która bierze jako argument gramatykę G , słowo w oraz symbol nieterminalny A , a zwraca wartość logiczną, mówiącą, czy słowo w da się wyprowadzić w gramatyce G z symbolu A . Funkcja powinna działać w czasie wielomianowym (pewne punkty można otrzymać również za „projekt” implementacji lub jej szkic).

Część 6. (7p) Powiedz, jak najwygodniej reprezentować symbole nieterminalne, słowa, symbole nieterminalne, produkcje oraz gramatykę w Prologu.

Część 7. (25p) Napisz predykat, `CNFParse(G, S, W)`, który nie zawodzi wtedy i tylko wtedy, gdy w gramatyce G (w postaci CNF, reprezentowanej zgodnie z poprzednim zadaniem) z symbolu S da się wyprowadzić słowo W . Absolutnym priorytetem jest zwięzłość i elegancja kodu. Efektywność nie ma znaczenia. Powiedz krótko, jak można by przyspieszyć ten program zachowując w miarę możliwości jego strukturę (czas wielomianowy uznajemy za satysfakcjonujący w tej części).