

Języki formalne i metody kompilacji

— wymagania na zaliczenie

Stefan Sokołowski

Elbląg, wersja z września 2024

Spis skrócony

1	Co trzeba <i>wiedzieć</i> (głównie z wykładu)	2
1.1	Gramatyki bezkontekstowe oraz prawostronnie liniowe	2
1.2	Wyrażenia regularne	2
1.3	Drzewa wyvodu	3
1.4	Deterministyczne automaty skończone	3
1.5	Maszyny ze stosem	3
1.6	Działania na językach	3
1.7	Analizator leksykalny (skaner)	3
1.8	Rekursywny zstępujący analizator składniowy (parser)	4
1.9	Wstępująca analiza składniowa przez tablice pierwszeństwa	4
1.10	Programowanie poziomu maszyny (język wewnętrzny)	4
1.11	Organizacja pamięci dla wykonania skompilowanego programu	4
1.12	Dwa etapy tłumaczenia	4
1.13	Gramatyki atrybutowe	5
2	Co trzeba <i>umieć</i> (głównie z laboratorium)	5
2.1	Czytanie i rozumienie gramatyki bezkontekstowej	5
2.2	Wyrażenia regularne	5
2.3	Drzewa wyvodu	5
2.4	Automaty	5
2.5	Maszyny ze stosem	5
2.6	Działania na językach	6
2.7	Analizator leksykalny (skaner)	6
2.8	Rekursywny zstępujący analizator składniowy (parser)	6
2.9	Wstępująca analiza składniowa przez tablice pierwszeństwa	6
2.10	Programowanie poziomu maszyny (język wewnętrzny)	6

Szczegóły wymagań

Ten przedmiot przede wszystkim należy **rozumieć**, nie można się go nauczyć na pamięć. Wstępnie potrzebne są:

- myślenie w stylu matematycznym — chociaż bez żadnej specjalnej dziedziny matematycznej;
- umiejętność programowania
 - w języku C łącznie z rekursją, typami wyliczeniowymi, strukturami i wskaźnikami; lub
 - w języku Java łącznie z rekursją, typami wyliczeniowymi, klasami i obiektami; lub
- staranność.

Dla wyjaśnienia trudności/wątpliwości zapraszam na konsultacje; szczegóły w
<http://iis.ans-elblag.pl/~stefan/Dydaktyka/kontakt-ANS.html>

1 Co trzeba *wiedzieć* (głównie z wykładu)

1.1 Gramatyki bezkontekstowe oraz prawostronnie liniowe

- Pojęcia alfabetu, słowa, języka.
- Terminale, nieterminale, produkcje, nieterminał początkowy.
- Generowanie słów przez gramatykę, generowany język.
- Produkcje rekursywne (i wzajemnie rekursywne) jako źródło nieskończonych języków.
- Gramatyki bezkontekstowe a języki bezkontekstowe; gramatyki prawostronnie liniowe a języki regularne.
- Równoważność gramatyk (równoważne: generują ten sam język).
- Nieformalne kryteria nieregularności języka.
- Przykłady języków bezkontekstowych ale nieregularnych.

1.2 Wyrażenia regularne

- Definicja wyrażenia regularnego i generowanie słów przez takie wyrażenie.
- Związek wyrażeń regularnych z językami regularnymi.
- Rola wyrażeń regularnych w wyszukiwaniu i zastępowaniu w tekście.

1.3 Drzewa wyvodu

- Pojęcie drzewa wyvodu słowa w gramatyce.
- Istnienie drzewa wyvodu a należenie słowa do języka.
- Trywialność drzew wyvodu dla gramatyk prawostronnie liniowych.
- Nieformalny związek między kształtem gramatyki, kształtem drzew wyvodu i pierwszeństwem operatorów.

1.4 Deterministyczne automaty skończone

- Definicja automatu, stanu, funkcji przejścia.
- Akceptowanie słowa przez automat, język akceptowany.
- Związek automatu z językami regularnymi, wyrażeniami regularnymi i gramatykami prawostronnie liniowymi.

1.5 Maszyny ze stosem

- Definicja maszyny ze stosem i jej (częściowej) funkcji przejścia.
- Akceptowanie słowa przez maszynę ze stosem (trzy warunki akceptacji).
- Związek maszyny ze stosem z gramatykami bezkontekstowymi.
- Prosta i odwrotna „polska” notacja beznawiasowa Łukasiewicza.

1.6 Działania na językach

- Suma, złączenie i iteracja (domknięcie Kleene’go) języka.
- Algebraiczne własności działań na językach.
- Rozwiązywanie równań językowych postaci

$$X = L \mid X M$$

gdzie L i M to dane języki.

1.7 Analizator leksykalny (skaner)

- Pojęcie leksemu, podział ciągu znaków na wejściu na leksemy.
- Automat akceptujący leksemy i jego funkcja przejścia.
- Programistyczne powody wyodrębniania etapu analizy leksykalnej.

1.8 Rekursywny zstępujący analizator składniowy (parser)

- Odpowiedniość między nieterminalami gramatyki a rekursywnymi funkcjami analizatora oraz między produkcjami a ciałami tych funkcji.
- Problem lewostronnej rekursji i sposoby jego „leczenia”.
- Błędy „twarde” i „miękkie” sygnalizowane przez funkcje parsera.
- Gramatyki nie-LL, do których metoda rekursywnej analizy zstępującej się nie nadaje.

1.9 Wstępująca analiza składniowa przez tablice pierwszeństwa

- Znaczenie relacji pierwszeństwa.
- Użycie tablicy pierwszeństwa do znalezienia podstawy redukcji.
- Generowanie relacji pierwszeństwa dla danej gramatyki.
- Zakres stosowalności analizy z pierwszeństwem.
- Poprawianie gramatyki na gramatykę z pierwszeństwem.
- Wykonanie analizy składniowej przy pomocy tablic pierwszeństwa z jawnym użyciem stosu.
- Automat wyszukujący właściwą redukcję dla znalezionej podstawy.
- Porównanie analizy z pierwszeństwem i rekursywnej analizy zstępującej.

1.10 Programowanie poziomej maszyny (język wewnętrzny)

- Uproszczona architektura komputera (adresowana pamięć i rejestr arytmetyczny) oraz podstawowe komendy niskiego poziomu (przesyłania, arytmetyczne, skoki).
- Adres komendy prosty i modyfikowany.
- Rozumienie działania komend.

1.11 Organizacja pamięci dla wykonania skompilowanego programu

- Rozmieszczenie zmiennych prostych, tablic i struktur.
- Paczki danych związane z wywołaniem funkcji (parametry i zmienne lokalne).
- Realizacja rekursji przez stos.

1.12 Dwa etapy tłumaczenia

- Od drzewa wyvodu do programu abstrakcyjnej maszyny stosowej.
- Od programu abstrakcyjnej maszyny stosowej do programu w języku maszyny.

1.13 Gramatyki atrybutowe

- Pojęcie gramatyki atrybutowej.
 - Tworzenie kompilatora przy użyciu programów FLEX i BISON.
-

2 Co trzeba *umieć* (głównie z laboratorium)

2.1 Czytanie i rozumienie gramatyki bezkontekstowej

- Rozpoznawanie, czy dane słowo należy do języka generowanego przez daną gramatykę.
- Podawanie przykładów (i kontrprzykładów) na słowa generowane przez gramatykę.
- Zapisywanie gramatyką języka zadanego nieformalnie.
- Rozpoznawanie, czy język zadany gramatyką bezkontekstową jest regularny (czyli czy posiada równoważną gramatykę prawostronnie liniową).

2.2 Wyrażenia regularne

- Konstruowanie wyrażenia regularnego równoważnego danej gramatyce prawostronnie liniowej oraz gramatyki prawostronnie liniowej równoważnej danemu wyrażeniu regularnemu.
- Wyszukiwanie i zastępowanie w tekście przy pomocy odpowiednio dobranych wyrażeń regularnych.

2.3 Drzewa wyvodu

- Konstruowanie drzewa wyvodu dla danego słowa i gramatyki.
- Poprawianie gramatyki w taki sposób, żeby wymusić pożądane pierwszeństwa operatorów.

2.4 Automaty

- Rozpoznawanie, czy dane słowo jest akceptowane przez dany automat.
- Konstruowanie automatu do danego języka regularnego opisanego nieformalnie.
- Konstruowanie gramatyki prawostronnie liniowej równoważnej danemu automatu, oraz automatu równoważnego danej gramatyce prawostronnie liniowej.

2.5 Maszyny ze stosem

- Rozpoznawanie, czy dane słowo jest akceptowane przez daną maszynę ze stosem.
- Konstruowanie maszyny ze stosem do danego języka bezkontekstowego zadanego nieformalnie.
- Konstruowanie gramatyki bezkontekstowej równoważnej danej maszynie ze stosem.
- Obliczanie wartości wyrażenia w odwrotnej notacji Łukasiewicza przy pomocy stosu.

2.6 Działania na językach

- Usuwanie lewostronnej rekursji (przez zamianę na iterację) z gramatyk bezkontekstowych.

2.7 Analizator leksykalny (skaner)

- Konstruowanie automatu akceptującego zadany nieformalnie język leksemów.
- Programowanie prostego skanera (przez modyfikację dostarczonego i omówionego programu w C).

2.8 Rekursywny zstępujący analizator składniowy (parser)

- Programowanie prostego pasera (przez modyfikację dostarczonego i omówionego programu w C).

2.9 Wstępująca analiza składniowa przez tablice pierwszeństwa

- Wyznaczanie zbiorów fst^+ i lst^+ dla nieterminali gramatyki.
- Wypełnianie tabeli pierwszeństwa relacjami biorącymi się z tych zbiorów.
- Konstrukcja automatu wyszukującego produkcję do danej podstawy redukcji.
- Programowanie prostego parsera z pierwszeństwem (przez modyfikację dostarczonego i omówionego programu w C).

2.10 Programowanie poziomu maszyny (język wewnętrzny)

- Ręczna symulacja programów w uproszczonym języku wewnętrznym.
 - Pisanie prostych programów w uproszczonym języku maszyny i ich uruchamianie na dostarczonym symulatorze.
-