Wprowadzenie

Języki formalne i techniki translacji - Wykład 1

Maciek Gębala

8 października 2019

Notatki

Literatura

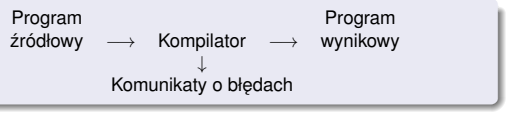
- A.V. Aho, R. Sethi, J.D. Ullman,
  Kompilatory. Reguły, metody i narzędzia,
  WNT, Warszawa 2002, (ISBN: 83-204-2656-1)
- J.E. Hopcroft, J.D. Ullman,
  Wprowadzenie do teorii automatów, języków i obliczeń,
  WNT, Warszawa 1994 (ISBN 83-01-11298-0)
- T.A. Sudkamp,
  Languages and Machines,
  Pearson, 2006, (ISBN: 978-81-317-1475-1)

Notatki

Proces translacji

Kompilator

Program czytający kod napisany w języku źródłowym i tłumaczący go na równoważny kod w języku wynikowym.



Notatki

Kompilacja typu analiza-synteza

- Analiza – rozłożenie programu na części składowe i stworzenie jego pośredniej reprezentacji.
- Synteza – przekształcenie reprezentacji pośredniej na program wynikowy.

Notatki

Przykłady narzędzi dokonujących analizy

- Edytory strukturalne
- Formatory kodu programu
- Kontrolery statyczne
- Interpretery

Notatki

Przykłady programów przypominających kompilatory

- Formatory tekstu
- Kompilatory do układów scalonych
- Interpretery zapytań

Notatki

Kontekst kompilatora

Notatki

Fazy kompilatora

Notatki

Fazy kompilatora

Analiza leksykalna

Ciąg znaków składający się na program źródłowy jest przekształcany w ciąg tokenów (symboli leksykalnych).

Analiza składniowa

Grupowanie symboli leksykalnych programu źródłowego w wyrażenia gramatyczne (tworzenie drzewa wyprowadzenia).

Analiza semantyczna

Kontrola programu pod względem poprawności semantycznej (np. kontrola typów) i zbieranie informacji do generowania kodu.

Notatki

Fazy kompilatora

Zarządzanie tablicą symboli

Zapamiętywanie identyfikatorów używanych w programie źródłowym i zbieranie informacji o różnych atrybutach tych identyfikatorów.

Wykrywanie i zgłaszanie błędów

Obsługa błędów w taki sposób aby nie przerywać kompilacji po pierwszym błędzie. Błędy wykrywane są w fazie analizy.

Notatki

Fazy kompilatora

Generowanie kodu pośredniego

Reprezentacja programu dla pewnej abstrakcyjnej maszyny (łatwa do utworzenia i tłumaczenia na program wynikowy).

Optymalizacja kodu

Poprawienie kodu pośredniego w taki sposób aby kod maszynowy działał szybciej.

Generowanie kodu wynikowego

Generowanie kodu wynikowego, najczęściej w asemblerze.

Notatki

Grupowanie faz

Przód kompilatora

Fazy zależne przede wszystkim od języka źródłowego i praktycznie niezależne od języka wynikowego. Zwykle składa się z analizatora leksykalnego, składniowego i semantycznego oraz tablicy symboli, generatora kodu pośredniego i obsługi błędów.

Tył kompilatora

Fazy zależne od maszyny docelowej a niezależne od języka źródłowego. Zwykle składa się z optymalizacji i generowania kodu z tablicą symboli i obsługą błędów.

Notatki

Języki formalne

Podstawowe definicje

Alfabet (Σ) – skończony zbiór symboli.
Słowo – skończony ciąg symboli z alfabetu.
Język – zbiór słów nad danym alfabetem ($\emptyset$ - język pusty).
ε – słowo puste.

Notatki

Deterministyczny Automat Skończony (DFA)

Deterministyczny automat skończony to uporządkowana piątka $(Q, \Sigma, \delta, q_0, F)$ gdzie
Q – skończony zbiór stanów,
Σ – skończony alfabet wejściowy,
δ – funkcja przejścia postaci $Q \times \Sigma \to Q$,
q_0 – stan początkowy,
$F \subseteq Q$ – zbiór stanów akceptujących.

Notacja

δ możemy rozszerzyć do $\hat{\delta} : Q \times \Sigma^* \to Q$ zgodną z definicją rekurencyjną $\hat{\delta}(q, \varepsilon) = q$ i $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$.

Automat akceptuje słowo w jeśli $\hat{\delta}(q_0, w) \in F$.

Notatki

Przykład

$M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\})$

δ	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Rysunek i analiza na tablicy

Notatki

Niedeterministyczny Automat Skończony (NFA)

Modyfikacja

Istnieje zero, jedno lub więcej przejść ze stanu przy tym samym symbolu wejściowym.

$$\delta : Q \times \Sigma \to 2^Q$$

Automat niedeterministyczny akceptuje słowo w jeżeli istnieje odpowiadający mu ciąg przejść ze stanu początkowego do stanu akceptującego.

Notatki

Przykład

$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_0, \{q_2, q_4\})$

δ	0	1
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	$\emptyset$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$

Rysunek i analiza na tablicy

Notatki

Niedeterministyczny Automat Skończony z ε-ruchami (NFA$_\varepsilon$)

Dopuszczamy przejścia między stanami bez symboli wejściowych.

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$$

Notatki

Wyrażenia regularne (RE)

Niech L, L_1, L_2 – języki nad alfabetem Σ. Wtedy

$$\begin{aligned} L_1 L_2 &= \{\, xy \,:\, x \in L_1 \wedge y \in L_2 \,\} && \text{(złożenie, konkatenacja)} \\ L^0 &= \{\varepsilon\} \\ L^{i+1} &= LL^i && \text{dla } i > 0 \\ L^* &= \bigcup_{i=0}^{\infty} L^i && \text{(domknięcie Kleene'ego)} \quad \left(L^+ = \bigcup_{i=1}^{\infty} L^i\right) \end{aligned}$$

Definicja

1. Wyrażenia regularne $\emptyset$, ε, a reprezentują odpowiednio język pusty, $\{\varepsilon\}$, $\{a\}$ dla $a \in \Sigma$.
2. Jeżeli r i s są wyrażeniami regularnymi reprezentującymi odpowiednio języki R i S to
   - $(r + s)$ reprezentuje język $R \cup S$,
   - (rs) reprezentuje język RS,
   - r^* reprezentuje język R^*.

Notatki

Przykład

$$(0+1)^*(00+11)(0+1)^*$$

$$(\varepsilon + 0 + 00)[(1+11)(0+00)]^*(\varepsilon + 1 + 11)$$

Notatki