

Wprowadzenie

Języki formalne i techniki translacji - Wykład 1

Maciek Gębala

2 października 2018

Maciek Gębala

Wprowadzenie

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)

Maciek Gębala

Wprowadzenie

Proces translacji

Kompilator

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



Maciek Gębala

Wprowadzenie

Kompilacja typu analiza-synteza

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

Maciek Gębala

Wprowadzenie

Notatki

Notatki

Notatki

Notatki

Przykłady narzędzi dokonujących analizy

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

Maciek Gębala Wprowadzenie

Przykłady programów przypominających kompilatory

- Formatory tekstu
- Kompilatory do układów scalonych
- Interpretery zapytań

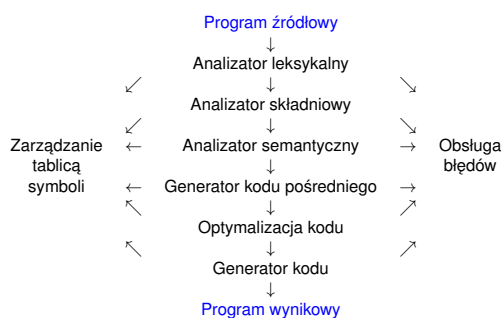
Maciek Gębala Wprowadzenie

Kontekst kompilatora



Maciek Gębala Wprowadzenie

Fazy kompilatora



Maciek Gębala Wprowadzenie

Notatki

Notatki

Notatki

Notatki

Fazy kompilatora

Analiza leksykalna

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

Analiza składowa

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.

Wprowadzenie

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.

Wprowadzenie

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.

Wprowadzenie

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.

Wprowadzenie

Notatki

[illegible]

Notatki

[illegible]

Notatki

[illegible]

Notatki

This image shows a full page of white paper with horizontal dashed lines, typical of primary school handwriting practice paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

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.

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 $Q \times \Sigma \rightarrow Q$,
- q_0 – stan początkowy,
- $F \subseteq Q$ – zbiór stanów akceptujących.

Notacja

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

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

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



Niedeterministyczny Automat Skończony (NFA)

Modyfikacja

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

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

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

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



Maciek Gębala

Wprowadzenie

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

Dopuszczamy przejścia między stanami bez symboli wejściowych.

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

Maciek Gębala

Wprowadzenie

Wyrażenia regularne (RE)

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

$$L_1 L_2 = \{xy : x \in L_1 \wedge y \in L_2\} \quad (\text{złożenie, konkatencja})$$

$$L^0 = \{\varepsilon\}$$

$$L^{i+1} = L L^i \quad \text{dla } i > 0$$

$$L^* = \bigcup_{i=0}^{\infty} L^i \quad (\text{domknięcie Kleene'ego}) \quad \left(L^+ = \bigcup_{i=1}^{\infty} L^i \right)$$

Definicja

- Wyrażenia regularne \emptyset , ε , a reprezentują odpowiednio język pusty, $\{\varepsilon\}$, $\{a\}$ dla $a \in \Sigma$.
- 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^* .

Maciek Gębala

Wprowadzenie

Przykład

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

Maciek Gębala

Wprowadzenie

Notatki

Notatki

Notatki

Notatki