

Egzamin

Termin 1. 14.06.2019 12:15

1. Co to jest kompilator
2. Co to jest interpretator
3. Architektura kompilatora (części składowe kompilatora)
4. Zadania analizatora leksykalnego
5. Zadania analizatora syntaktycznego
6. Zadania analizatora semantycznego
7. Zadania optymalizatora
8. Zadania generatora kodu docelowego
9. Definicja gramatyki bezkontekstowej
10. Symbole terminalne i nieterminalne gramatyki
11. Co to jest wyprowadzenie
12. Co to jest drzewo parsowania, jego właściwości
13. Gramatyki jednoznaczne i niejednoznaczne
14. Operatory łączne lewostronnie i operatory łączne prawostronnie
15. Jak w gramatyce określamy pierwszeństwo operatorów.
16. Jak się tworzy gramatyka jednoznaczna dla wyrażeń arytmetycznych
17. Co to jest translacja sterowana składnią
18. Co to jest schemat translacji sterowanej składnią
19. Co to jest notacja postfiksowa
20. Co to jest atrybut syntezowany
21. Co to jest definicja sterowana składnią
22. Uwzględnienie akcji syntaktycznych w drzewie parsowania
23. Dlaczego analizator leksykalny jest tworzony osobno
24. Tokeny, Wzorce, Leksemy
25. Klasy tokenów
26. Usuwanie znaków białych podczas analizy leksykalnej
27. Co oznacza „czytanie z wyprzedzeniem”
28. Rozpoznawanie stałych
29. Rozpoznawanie identyfikatorów i słów kluczowych
30. Co to jest tablica haszująca

31. Co to jest alfabet
32. Co to jest napis, długość napisu
33. Prefiks, sufix napisu
34. Złączenie napisów
35. Podnoszenie napisu do potęgi
36. Definicja języka, operacje na językach (suma, złączenie, potęgowanie, domknięcie, domknięcie dodatnie)
37. Wyrażenia regularne
38. Definicje regularne
39. Rozszerzenia wyrażeń regularnych stosowanych w LEX
40. Wzorce leksemów dla liczb, identyfikatorów i instrukcji warunkowych
41. Diagram przejść
42. Jak korzystamy z diagramu przejść celem rozpoznawania leksemów
43. Narzędzie Lex, jak korzystamy z niego celem utworzenia analizatora leksykalnego
44. Struktura programu w języku Lex
45. Rozwiązywanie konfliktów w Lex
46. Deterministyczny automat skończony
47. Niedeterministyczny automat skończony
48. Jak korzystamy z automatu skończonego celem rozpoznawania leksemów
49. Symulacja automatu skończonego (DFA)
50. Na czym polega parsowanie zstępujące
51. Kiedy korzystamy z ϵ -produkcji w parsowaniu zstępującym
52. Co to jest parsowanie przewidujące
53. Definicja zbioru FIRST, jak korzystamy z tego zbioru podczas parsowania przewidującego
54. Co to jest rekurencja lewostronna
55. Wyeliminowanie rekurencji lewostronnej
56. Jak tworzymy pseudokod parsowania przewidującego
57. Projektowanie parsera przewidującego
58. Wyprowadzenia bezpośrednie i pośrednie
59. Forma zdaniowa gramatyki G
60. Zdanie gramatyki G

61. Język generowany przez gramatykę G
62. Wyprowadzenia lewostronne i prawostronne
63. Zbiór FIRST
64. Zbiór FOLLOW
65. Gramatyką LL(1), jakie warunki mają być spełnione aby gramatyka była gramatyką LL(1)
66. Tablica parsowania gramatyki LL(1)
67. Architektura parsera LL(1)
68. Algorytm parsowania za pomocą parsera LL(1)
69. Co to jest redukcja
70. Jak działa analiza wstępująca
71. Operacje shift i reduce
72. Parsowanie shift-reduce
73. Co to jest uchwyt
74. Parser LR (k)
75. Parsery SLR (k), LALR (1), GLR
76. Ogólna architektura parsera LR(1)
77. Automat PDA
78. Sytuacje LR(0)
79. Stany PDA
80. Architektura parsera LR(k)
81. Tablice parsowania
82. Gramatyka LR(k)
83. Konflikty w parserach LR(k)
84. Sposób działania parsera LR(1)
85. Generator parserów YACC, jak korzystamy z YACC'a celem utworzenia parsera
86. Specyfikacją dla programu YACC
87. Rozwiązywanie konfliktów reduce/reduce
88. Rozwiązywanie konfliktów shift/reduce
89. Cele analizy semantycznej
90. Przykłady błędów
91. Kontrola programu

92. Kontrola typów i generowanie kodu
93. Zakresy dynamiczne
94. Tablice symboliczne
95. Zastosowanie i Implementacja tablicy symboli
96. Systemy typów
97. Sposoby kontroli typów
98. Składnia abstrakcyjna i konkretna
99. Drzewa konkretne i abstrakcyjne
100. Zasady tworzenia prostego translatora: kolejność czynności niezbędnych do utworzenia prostego translatora.
101. Dostosowanie schematu translacji na notację polską
102. Pseudokod translacji na notację polską
103. Translator w języku Java
104. Translator W języku C
105. Translator w oparciu o Lex i Yacc
106. Dyrektywy SPIM
107. Sposoby adresowania
108. Operatory porównania
109. Instrukcje skoków
110. Wywołania systemowe
111. Przydział pamięci
112. Podstawowe dyrektywy SPIM
113. Przetwarzanie wyrażeń
114. Obliczenie adresów elementów tablic jedno- i dwuwymiarowych
115. Instrukcje warunkowe
116. Pętle
117. Wywołanie procedur
118. Wskaźniki SP i FP