Compiler Construction
2017

Prof. O. Nierstrasz, Mohammad Ghafari
Yuriy Tymchuk, Manuel Leuenberger

COMPILER CONSTRUCTION - EXAM

First name: ─────────────────

Last name: ─────────────────

Matrikel: ─────────────────

Date: Friday, 02.06.2017
Allowed material: this paper and a pen
Number of exercises: 3
Total points: 30

Important:
You have 90 minutes to solve the exam.
Some exercises are split over more than one page, so carefully read all exercises before proceeding. If you don't know something don't waste time, go ahead and if you have time go back and try to solve unanswered questions afterwards. You must answer each question briefly and to the point.

Compiler Construction
2017

Prof. O. Nierstrasz, Mohammad Ghafari
Yuriy Tymchuk, Manuel Leuenberger

Exercise 1 (10 Points)

Answer the following questions, any answer that exceeds 3 lines will not be corrected. Seriously!

a) Why is it important that a compiler supports separate compilation? **(2 Points)**

b) What is the difference between context-free and context-sensitive grammars? **(2 Points)**

c) Which kind of parsers have a problem with left-recursive grammars? What are the effects of the problem? **(2 Points)**

d) What is SSA and which optimizations rely on it?**(2 Points)**

e) Why does a generational garbage collector distinguish between old and new objects? **(2 Points)**

Exercise 2 (7 Points)

Consider a language over an 0 and 1 alphabet that always starts with 01 and ends with 10, and has no consecutive 0s. For example:

`0110, 01010, 0111111110, 010101010, 01111010110110, ...`

But not:

`010`

    a) Write a regular expression describing the language. **(2 Point)**

    b) Draw a corresponding non-deterministic finite state automaton (NFA). **(2 Points)**

Compiler Construction
2017

Prof. O. Nierstrasz, Mohammad Ghafari
Yuriy Tymchuk, Manuel Leuenberger

c) Derive a corresponding deterministic finite state automaton (DFA) from the NFA in the previous step. Show the intermediate steps in the construction. **(3 Points)**

Exercise 3 (13 Points)

Consider simplified boolean algebra with operations *or* ($\vee$), *and* ($\wedge$), *parentheses*, and literals T and F. $\wedge$ has a higher precedence than $\vee$. See the examples:

- T

- $T \vee F \wedge T$

- (T)

- $(T \vee F \vee F) \wedge (F)$

- $(F \wedge F) \wedge (F \vee T) \vee T \wedge F$

a) Write a context free grammar **(2 Points)**

Compiler Construction
2017

Prof. O. Nierstrasz, Mohammad Ghafari
Yuriy Tymchuk, Manuel Leuenberger

b) Write a context free grammar with precedence and without any left recursion: **(4 Points)**

Compiler Construction
2017

Prof. O. Nierstrasz, Mohammad Ghafari
Yuriy Tymchuk, Manuel Leuenberger

c) Draw a concrete syntax tree (CST) for the following example with respect to the grammar of the previous point. Example: $F \wedge (T \vee F) \vee T$. **(3 Points)**

Compiler Construction
2017

Prof. O. Nierstrasz, Mohammad Ghafari
Yuriy Tymchuk, Manuel Leuenberger

d) Explain how you would convert your CST into an AST. Provide the AST of the CST that you constructed for the previous step. **(2 Points)**

e) Describe an interpreter that evaluates the AST into $true$ or $false$. **(2 Points)**
   Here are some examples what are the expected results:

   - $T = true$
   - $T \vee F \wedge T = true$
   - $(T) = true$
   - $(T \vee F \vee F) \wedge (F) = false$
   - $(F \wedge F) \wedge (F \vee T) \vee T \wedge F = false$

Compiler Construction
2017

Prof. O. Nierstrasz, Mohammad Ghafari
Yuriy Tymchuk, Manuel Leuenberger

Points

Exercise 1

Task	Points	Score
1	2	
2	2	
3	2	
4	2	
5	2	
Total	**10**	

Exercise 2

Task	Points	Score
1	2	
2	2	
3	3	
Total	**7**	

Exercise 3

Task	Points	Score
1	2	
2	4	
3	3	
4	2	
5	2	
Total	**13**	