Technische Universität Wien
Compilers and Languages
Prof. J. Knoop

A-1040 Wien
Argentinierstr. 8
Tel.: 01-58801-18510

# “ Analysis and Verification (185.276, VU 2.0, ECTS 3.0) ” SS 2019

## Assignment 6 22 May 2019

---

**Exercise 1** : (2+(2+2+2)+4+(4+4) Points)

*Constant analysis* (e.g., for *simple constants*) aims at accurately computing the value of a program term. Following the example of the constant analysis for simple constants, a *sign analysis* shall be specified, which aims at computing the sign of program variables (but not their exact values).

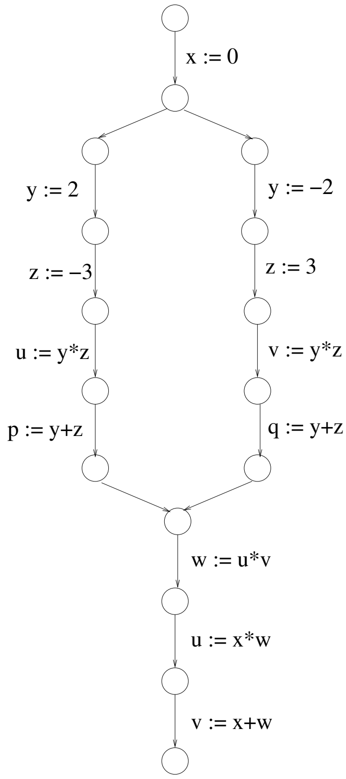
More detailed, the sign analysis shall be able to distinguish the following cases for variables:

- No information about the sign (represented by $\perp$).
- The information exact zero (represented by 0).
- The information smaller or equal zero (represented by $-0$).
- The information greater or equal zero (represented by $0+$).
- The information smaller or greater or equal zero (represented by $-0+$).

Then:

1. Provide an order of the five elements $\perp, 0, -0, 0+$, and $-0+$ such that it specifies a complete lattice represented by its Hasse diagram.
2. Based on this lattice, specify the three components of a sign analysis, which computes as accuratly as possible the sign of all variables occurring in a program, i.e., the sign of all program variables:
   2.1 DFA lattice (as complete lattice of abstract DFA states where variables take values out of $\{\perp, 0, -0, 0+, -0+\}$).
   2.2 (Local abstract) DFA semantics $[\![\ ]\!]_{sa}$ for assignments and tests.
   2.3 Largest possible reasonable set of initial assertions (as subset of the set of abstract program states).
3. Is your DFA semantics $[\![\ ]\!]_{sa}$ monotonic, distributive, additive? Prove your claim.
4. What is the
   4.1 *MaxFP* solution
   4.2 *MinFP* solution

   for the least possible initial assertion for the following program? Annotate the program points with the DFA information of the *MaxFP* and *MinFP* solution.




**<u>Exercise 2</u>** : (4+6 Points)

Consider the *availability analysis* (for a term $t$ (Variant 1)) of Chapter 7.9.1 with the initial assertion 'false', the analysis for simple constants of Chapter 7.9.2 with the initial assertion $\sigma_\perp$, and an arbitrary but fixed flow graph node $n$.

What does it mean obviously (anschaulich), say, what can be derived about the paths, which starting from start node **s** reach node $n$ ('for all such paths hold...', 'for some/for at least one such path holds...'), if in case of the

1. availability analysis the
   (a) *MOP* solution at node $n$ is 'true' for $t$, what if it is 'false.'
   (b) *JOP* solution at node $n$ is 'true' for $t$, what if is 'false.'
2. simple constants analysis
   (a) *MOP* solution at node $n \neq \mathbf{s}$
   (b) *JOP* solution at node $n \neq \mathbf{s}$

   has value

   - $\perp$
   - $z$ with $z \in \mathbb{Z}$
   - $\top$

   for an arbitrary but fixed variable $v$?

---

**Submission**: Wednesday, 29 May 2019, before the lecture.