1 Lecture

- Formally, a small-step operational semantics is a five-tuple $(CF, \Rightarrow, FC, IF, OF)$, where
  - CF is a set of configurations,
  - $\Rightarrow$ is a binary relation over configurations ($\Rightarrow \subseteq CF \times CF$),
  - FC is a set of final configurations ($FC \subseteq CF$),
  - $IF : Program \times Input \to CF$ is an input function and
  - $OF : FC \to Output$ is an output function.
- An irreducible configuration is also called normal form. Each normal form is either a final configuration or a stuck state.
- For a given program and its input, small-step operational semantics either produces output value(s), stuck state(s) or no state whatsoever.
- A relation $R \subseteq A \times A$ is confluent iff $\forall a_1, a_2, a_3 \in A\colon a_1 R^* a_2 \wedge a_1 R^* a_3 \Rightarrow \exists a_4 \in A\colon a_2 R^* a_4 \wedge a_3 R^* a_4$. Confluency implies single normal form.
- A strongly normalising (or terminating) transition relation produces normal form via every path. A weakly normalising transition relation produces normal form via at least one path.
- Well-typed programs don't get stuck. In special cases (universality and strong normalisation don't mix well), well typed programs also always terminate, i.e. they always produce an output value.

2 Seminar

1. Given an irreflexive ($\forall a \in A\colon (a, a) \notin R$) antisymmetric ($\forall (a, a') \in A\colon (a, a') \in R \Rightarrow (a', a) \notin R$) relation R over a set A and an element $x \in A$, write down the expression specifying the number of its normal forms.
2. Given an irreflexive ($\forall a \in A\colon (a, a) \notin R$) antisymmetric ($\forall (a, a') \in A\colon (a, a') \in R \Rightarrow (a', a) \notin R$) relation R over a set A and an element $x \in A$, write down the expression specifying the length of any path from x to any of its normal forms.
3. Given an irreflexive ($\forall a \in A\colon (a, a) \notin R$) antisymmetric ($\forall (a, a') \in A\colon (a, a') \in R \Rightarrow (a', a) \notin R$) relation R over a set A and an element $x \in A$, write down the expression specifying the length of the shortest path from x to any of its normal forms.
4. Given an irreflexive ($\forall a \in A\colon (a, a) \notin R$) antisymmetric ($\forall (a, a') \in A\colon (a, a') \in R \Rightarrow (a', a) \notin R$) relation R over a set A and an element $x \in A$, write down the expression specifying the length of the shortest path from x to its nearest normal form.

5. Given an irreflexive ($\forall a \in A\colon (a, a) \notin R$) antisymmetric ($\forall (a, a') \in A\colon (a, a') \in R \Rightarrow (a', a) \notin R$) relation R over a set A, write down the expression specifying the length of the longest path from an element of A to its nearest normal form.

6. Given an irreflexive ($\forall a \in A\colon (a, a) \notin R$) antisymmetric ($\forall (a, a') \in A\colon (a, a') \in R \Rightarrow (a', a) \notin R$) relation R over a set A and an element $x \in A$, write down the expression specifying the next element on the shortest path from x to its nearest normal form.

7. Write down a definition of a function which takes a program and returns its copy with all constant (sub)expressions folded.

8. Write down a parameterised formula which decides whether the program supplied as a parameter at least once reads from an uninitialised variable.

9. Write down a definition of a function which takes a program and returns its copy with all common nontrivial subexpressions extracted to commands.