Task documentation DKA: Determinization of finite automaton in Python 3 for IPP 2013/2014
Name and surname: Marek Milkovič
Login: xmilko01

1 Assignment

The task was to create a script for determinization of the finite automaton, including the removal of ε-rules and lexical, syntactic and semantic validation of the input finite automaton.

2 Design

2.1 Lexical analysis

Since the syntax of writing finite automata is not much predictable, because of the presence of comments and previously uknown count of whitespaces, it is hard to parse out lexical units (tokens) with just regular expressions. The design of two-step lexical analysis provides solution for this problem. The steps are:

1. Finite automaton which parse out individual sets (sets of 5-tuple) from the input finite automaton.
2. Regular expressions which parse out individual members of the sets

2.2 Syntactic analysis

Syntactic analysis is performed with cooperation of lexical analysis, where the token from the lexical analyzer is compared with the expected token type. The part of the syntax analysis is also performed directly inside the lexical analyzator because the later analyzation would require additional parsing because of the way how the information are stored.

2.3 Semantic analysis

Semantic analysis is mixed into the lexical and syntactic analysis and is performed when it all required data for semantic analyzation are parsed.

2.4 Epsilon rules

Construction of ε-closure and removal of ε-moves is perfmored in a single algorithm using the stack of states to visit and the list of the already visited states. For the currently iterated state, all ε-moves are removed and replaced by regular moves using the stack. Currently iterated state is then put into the already visited states. Then the next state is iterated.

2.5 Determinization

Algorithm for determinization of finite automaton is slightly different than common algorithm. It doesn't iterate over the symbols in the alphabet but instead it iterates states, starting from the starting state. The structure of the automaton in the script is made that every state contains dictionary of rules with key as an accepted symbol and value as all the states where it can move with this rule. Iterating over this dictionary provides easy way how to distiguish non-determinism and removes it by merging states together. As the removal of epsilon rules, determinization also uses stack of the states to visit and the list of the already visited states. Since every state contains the list of the states it was made up with, it is easy to decide which states to visit to build up deterministic finite automaton.