Grammar to Finite Automaton Converter

Course: Formal Languages & Finite Automata

Author: Ernu Catalina

Theory:

In the study of formal languages and finite automata, one fundamental concept is the conversion of a context-free grammar (CFG) into a finite automaton (FA). This process involves translating the rules and symbols of a grammar into states and transitions in an automaton, facilitating the recognition of valid strings in the language defined by the grammar.

Objectives:

- Understand the principles of context-free grammars and finite automata.
- Implement a conversion mechanism from CFGs to FAs.
- Demonstrate the generation of valid strings based on a given grammar.
- Verify the acceptance of input strings by the generated finite automaton.
- Implementation Description

Grammar Class:

Represents a context-free grammar with non-terminals, terminals, and productions.

FiniteAutomaton Class:

Models a finite automaton with states, alphabet, transitions, initial state, and accepting states.

String Generation:

Generates valid strings based on the given grammar.

Grammar to FA Conversion:

Converts a context-free grammar to a finite automaton.

String Acceptance Checking:

Checks whether input strings are accepted by the generated finite automaton.

Code Snippets

```
# Grammar definition
non_terminals = {'S', 'B', 'L'}
terminals = {'a', 'b', 'c'}
productions = {
    'S': ['aB'],
    'B': ['bB', 'cL'],
    'L': ['cL', 'aS', 'b']
}
grammar = Grammar(non_terminals, terminals, productions)

# Generating and printing valid strings
valid_strings = generate_strings(grammar, 5)
print("Valid Strings:")
for string in valid_strings:
    print(string)

# Convert Grammar to Finite Automaton
fa = FiniteAutomaton()
fa.convert_from_grammar(grammar)
print("\nFinite Automaton:")
print(fa)

# Checking strings with Finite Automaton
input_strings = ["abbc", "cabc", "acbcc"]
for string in input_strings:
    if fa.check_string(string):
        print(f"'{string}' is accepted.")
    else:
        print(f"'{string}' is not accepted.")
```

Conclusions

The implemented converter successfully converts context-free grammars into finite automata and demonstrates the generation and acceptance checking of strings within the defined language.

References

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Introduction to Automata Theory, Languages, and Computation. Addison-Wesley.
Sipser, M. (2012). Introduction to the Theory of Computation. Cengage Learning.