

Scanning:

Core task: To implement a state machine which is driven by characters from the input program. First, define a token. A token should have a type and a lexical element, which is its actual string. Then, the states and transitions of the state machine should be defined. States should be final or non-final, when no more transitions can be taken, the state machine should be on a final state, or a lexical error is returned.

Transition involves the current state, the input character and the next state. We need to group input characters to a set to better define transitions.

Scanning comments is somewhat special

Parsing

The context free grammar was taken from the 1st edition of the Java specification reference and then fed to the Jlalr1 program to produce the parse rules. The parse rules is essentially a state machine that generates a parse tree.

Weeding

Special requirements listed in the assignment description is implemented in the weeding stage. The weeding function traverses the parse tree to find unique patterns that are not supported by the language.

However, our team overlooked the difference between the Joos1W language and the Java language constituted by the context free grammar described above, so many features that are not supported by the Joos language was not picked out by the weeder.

AST Building

The design of the ASTree is mainly copied from the eclipse API documentation.