

0.1 Lexical analyzer

A lexical analyzer reads the input file, and returns a series of tokens based based on the input. More specifically it is the scanner in the lexical analyzer which does this. These tokens are matched by rules, usually described by regular expressions. An example of such grammar rules can be seen on 0.1. Formally a token consists of two parts: The token type, and the token value. As an example the IDENT token seen on 0.2 has the token type IDENT and the value 'c'.

Terminal	Regular expression
dcl	"[a - z]"
assign	"="
digit	"[0 - 9]+"
endassign	";"
blank	" "+

Table 0.1: Sample token specification

This specification of tokens, would be used by the scanner to determine how tokens looks, and thereby which text-elements are tokens.

For example the lines of code seen on ?? might be read as the tokens seen on figure 0.2.

Token	Lexeme
IDENT	c
ASSIGN	=
DIGIT	42
SEMICOLON	;

Table 0.2: Example of tokens

The scanner produces a stream of tokens, which is returned to the parser, which checks if the tokens conforms to the language-specification.