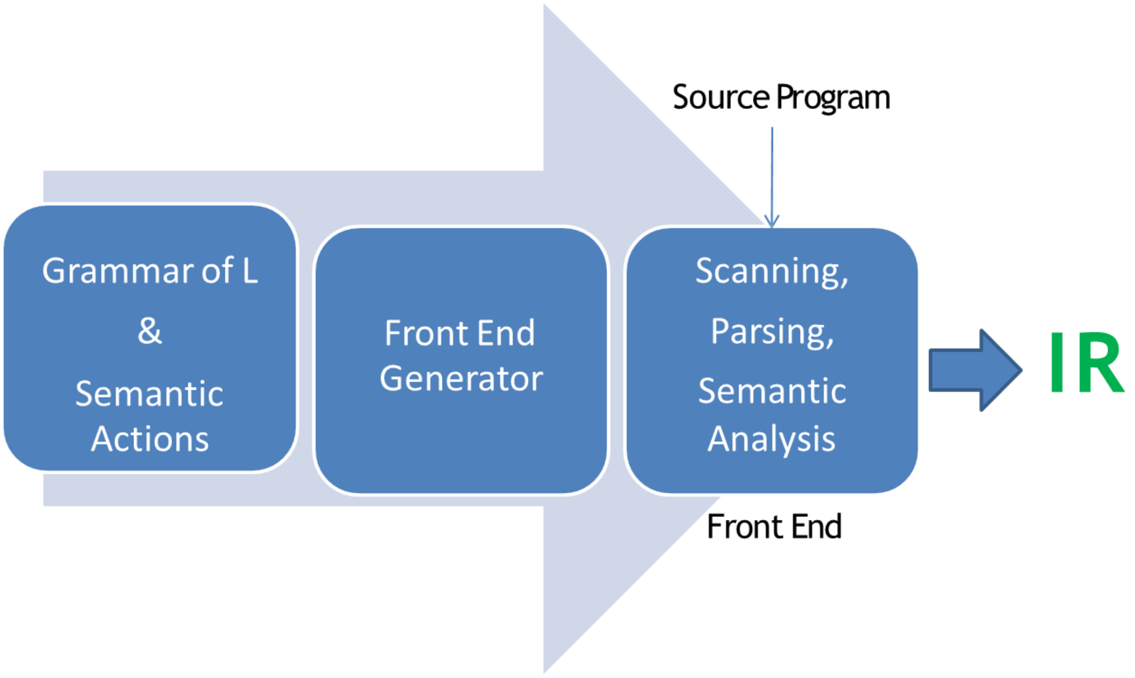
Language Processor Development Tools

- Writing of language processors is a well understood and repetitive process which ideally suits the program generation approach to software development.
- Set of language processor development tools (LPDTs) focusing on generation of the analysis phase of language processors.
- LPDT requires the following two inputs:
  - Specification of grammar of language L.
  - Specification of semantic actions to be performed in the analysis phase.

LPDT


Source Program
Grammar of L & Semantic Actions
Front End Generator
Scanning, Parsing, Semantic Analysis
IR
Front End

- It generates programs that perform lexical, syntax and semantic analysis of the source program and construct the IR.

- These programs collectively form the analysis phase of the language processor.

- Lexical analyzer generator LEX, and the parser generator YACC.

- The Specification consists of a set of "Translation Rules" of the form

  < string specification > { < semantic action > }

  where,

  - < semantic action > consist of C code.
  - The code is executed when a string matching

    < string specification > is encountered

    in the input.

- LEX and YACC generate C programs

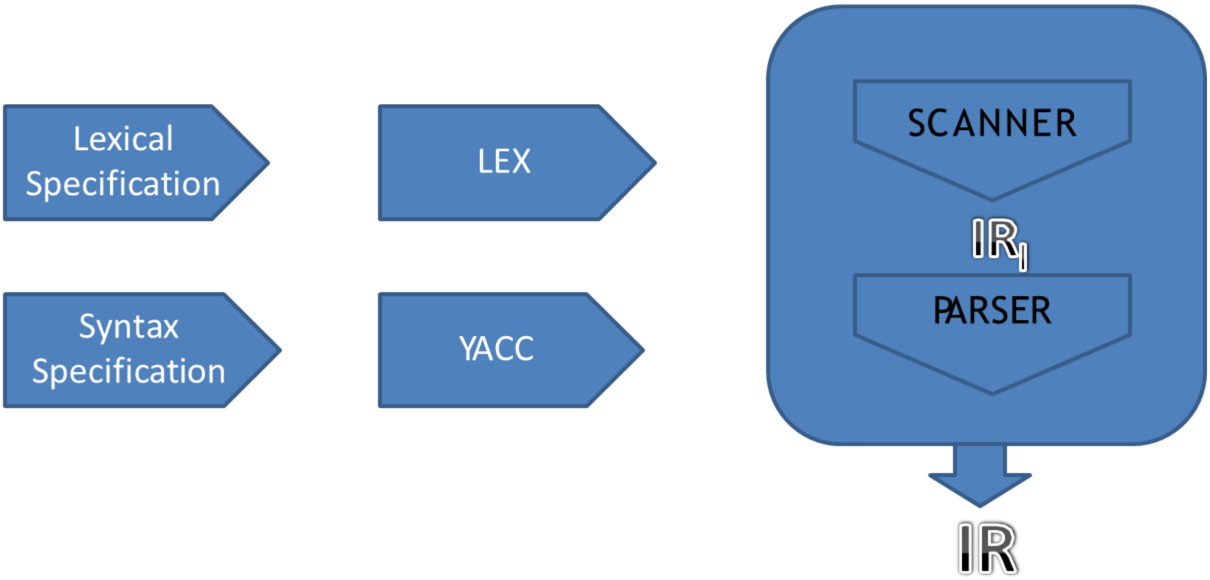
which contain the code for scanning and

parsing, respectively and the semantic actions

contained in the specification.

●A YACC generated parser can use a LEX generated scanner as a routine if the scanner and parser use same conventions concerning the representation of tokens.
A single pass compiler can be built using LEX and YACC if the semantic actions are aimed at generating target code instead of IR

Using LEX and YACC



Lexical Specification

LEX

Syntax Specification

YACC

SCANNER

IR_l

PARSER

IR

- LEX accepts an input specification which consists of two components.

◦1st is Specification of String i.e. in regular expression form. e.g id’s and constants.
2nd is Specification of Semantic Actions aimed at building an IR.

YACC

- Each string specification in the input to YACC resembles a grammar production.
- The actions associated with a string specification are executed when a reduction is made according to the specification.
- An attribute is associated with every non terminal symbol.
- The value of this attribute can be manipulated during parsing.
- The attribute can be given any user-designed structure.
- A symbol ‘$n’ in the action part of a translation rule refers to the attribute of the n’th symbol in the RHS of string specification.
- ‘$$’ represents the attribute of the LHS symbol of the string specification.