

ACKNOWLEDGEMENT ABSTRACT

A parser is a program that takes a string as an input and checks if it is in the set of strings defined by some grammar and that grammar is built into the parser.

The parser generator is a program that takes a grammar as an input and produces the parser for the language described by that grammar. To build a parser there are two things which must be considered; first the grammar of a language; second the formalism of the parser which is a tool for implementing the grammar.

The aim of this research is to build a parser generator for natural language, this parser will help the user who works on natural language field in terms of time and effort.

The Parser Generator System (PGS) has been designed using two formalisms. These are Definite Clause Grammar (DCG) and Augmented Transition Networks (ATN) since they are largely used in natural language analyzing.

The system could be run on an IBM computer and its compatible using Turbo Prolog version 2.0 and its toolbox.

PGS is designed as a collection of subsystems and each subsystem has a certain function. These subsystems are:

- 1- the user interface
- 2- the checker
- 3- the help
- 4- the generator
- 5- the tester
- 6- the modifier