

Small Basic: Parser Generator

Table of Contents

[Overview](#)

[What's a Parser?](#)

[What's a Parser Generator?](#)

[Sample Programs](#)

[Parser Generator 0.4 for Graphics Interpreter](#)

[Parser Generator 1.4 for Code Block Formatter](#)

[See Also](#)

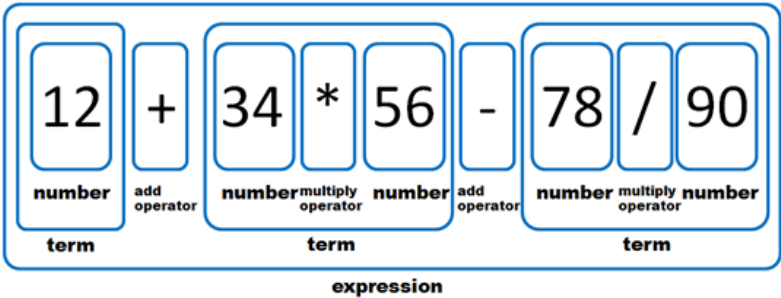
Overview

This article introduces Parser and Parser Generator written in Small Basic.

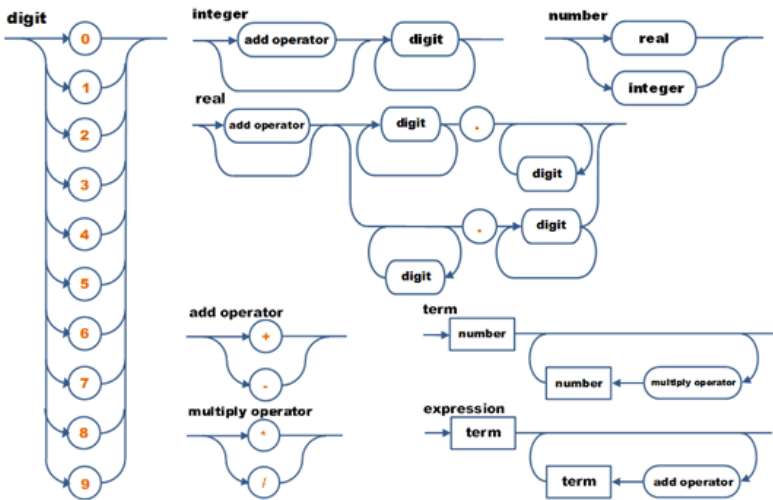
What's a Parser?

Parser is a program which analyzes program language syntax and does something. That will be a compiler, an interpreter, a program formatter, a program converter, or etc.

For example, calculator accepts expressions for it's input. The expression will be analyzed as follows. That is parsing.



Expression has it's syntax as following diagram.



This diagram can be converted as following form called EBNF (Extended Backus-Naur Form).

```
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
integer = [add operator], digit, {digit}
real = [add operator], (digit, {digit}, ".", {digit} | {digit}, ".", digit, {digit})
add operator = "+" | "-"
multiply operator = "*" | "/"
number = real | integer
```

```
term = number, {multiply operator, number}
expression = term, {add operator, term}
```

And this relationship will become program code of parsing. For example, syntax of "number" is

```
number = real | integer
```

The code of parsing "number" is as follows.

```
Sub Parse_GetNumber
  Parse_GetReal()
  If bMatched = "False" Then
    Parse_GetInteger()
  EndIf
EndSub
```

The whole program of calculator has been published as [BQJ710-0](#).

What's a Parser Generator?

Parser Generator is a program which generates parser program. The input of parser generator is a syntax of language.

Sample Programs

There are two sample parser generator programs. The first one generates Graphics Interpreter. The second one generates Code Block Formatter. These parser generators have subset of Small Basic language syntax for their input.

Parser Generator 0.4 for Graphics Interpreter

This program [XWX066](#) generates Small Basic Graphics interpreter. Save this program as "ParserGenerator04.smallbasic" and remove comment mark from lines with File object. This program generates a interpreter named "SBGIInterpreter04.smallbasic". Generated program has been uploaded as [XRZ138](#).

- Subroutines start with Exec_ , Lex_ and SBGI_ will be copied to the interpreter.
- Subroutines start with PG_ are parser generator itself.
- PG_Init is a core subroutine and contains syntax of Small Basic graphics as BNF format (but not strict BNF here).
- Subroutine SB_GetSub gets subroutine source code from given file name.

Language syntax is defined as BNF. So, many languages can be parsed by changing BNF, such as Small Basic (of course), old type of BASIC, any markup languages or BNF itself etc.

* BNF : Backus-Naur Form

Parser Generator 1.4 for Code Block Formatter

This program [ZBG977-4](#) generates Small Basic code block formatter. Generated program has been uploaded as [SKC235-2](#), and see the article "[Small Basic: How to Format a Code Block](#)" for it's detail. Following code is a BNF syntax of Small Basic written in PG_Init subroutine in this parser generator.

```
' initialize syntax array
syntax["state"] = "[<token>]"
syntax["token"] = "{<kw>|<op>|<mem>|<real>|<_str>|<var>|<rem>|[]}"
syntax["kw"] = "
{For|To|Step|EndFor|If|Then|ElseIf|Else|EndIf|While|EndWhile|Goto|Sub|EndSub}"
keywords
syntax["op"] = "[+|-|'*'|/|=|'<'>'<='>'<>'>|And|Or|(|,|):}"
syntax["mem"] = "<_name>[.]<_name>"
or event
syntax["var"] = "{<_name>['|'|']}"
syntax["rem"] = "_SQ*"
syntax["real"] = "[_<_num>[.<_num>]"
' statement
' token
' operator
' member (property)
' variable
' remark (comment)
' real number
```

See Also

[Small Basic Parser Generator](#) (an article in Small Basic Forum)