그래픽 요소를 사용한 파서 제작 도구의 설계

정호욱, 최종명, 유재우
숭실대학교 컴퓨터학과

A Design of the Parser Generator using Graphic Components

Ho-Uk Jung, Jong-Myung Choi, Chae-Woo Yoo
Department of Computing, Soongsil university

요 약

프로그램을 개발하다보면 많은 분야에서 파싱 작업을 필요로 하게된다. 이러한 필요에의해 개발되는 파서는 Lex와 Yacc과 같은 도구에 의해 생성되는 경우도 있지만, 도구의 사용이 어렵기 때문에 프로그램으로 직접 작성하는 경우도 많다. GUI 방식으로 아이콘을 이용해서 파서를 작성할 수 있다면, 파서를 필요로하는 많은 프로그램 개발에서 손쉽게 사용될 수 있을 것이다. 이 논문에서는 아이콘을 이용해서 파서를 생성할 수 있는 방법에 대해 기술하고 있다

1. 서론

프로그램 개발과 관련된 어플리케이션 프로그램들 중에는 상태나 사건의 진행에 따라서 다음 동작을 결정하는 프로그램들이 있다 사건이나 상태의 진행에 따라서 동작을 결정하는 프로그램은 그 사건이나 상태를 파싱할 필요가 있다. 원래 파싱이란 주로 프로그래밍 언어의 구문과 의미 분석을 위해 많이 사용하던 기술로서 여러 가지 방법으로 수행될 수 있고 그 정보 또한 여러 가지 형태로 나타내어질 수 있다.

파서는 파싱을 수행하는 프로그램이다. 파서를 개발하는 도구 중 가장 보편적으로 사용되는 것이 Lex[1]와 Yacc[1] 유틸리티이다 Lex는 입력 파일을 정해진 토큰으로 나누어주는 어휘 분석기로 많이 사용되며 Yacc은 Lex가 넘겨준 토큰의 정보를 기지고 구문 분석하여 처리된 결과를 트리 형태로 만들어 주는 프로그램이다. 파서를 제작할 때 관련 도구를 이용하면 프로그램을 더욱 쉽게 제작할 수 있다 하지만 도구를 사용하기면 도구 사용법을 익히이 한다는 단점이 있다

지금의 파서 제직 도구와 달리 아이콘의 결합으로 파서를 구현 할 수 있다면 사용자는 쉽게 파서를 제작할 수 있을 것이고 그 응용 범위도 넓어질 것이다 본 논문에서는 그래픽 요소를 이용하여 파서를 제작할 수 있는 파서 생성 도구를 설계하였다

2. 파싱의 이용분야

파싱은 주로 자연언어 처리나 컴파일러를 개발하는데 많이 이용되는 기술이다. 하지만 컴파일러 개발 이외에도 다양한 분야에서 이용될 수 있으며, 실제로 많은 응용프로그램들에서 내부적으로 파싱 기술을 이용한다.

SMPLISH[2]와 같은 전자 사전 시스템에서는 영문서의 어휘와 문법을 분석하여 단어나 문장의 의미를 알아내기 위하여 lexer와 parser 생성기를 내부적으로 사용하고 있다

인터넷에서 사용되는 HTML이나 XML, SGML 문서를 파싱하거나 LaTeX 문서를 HTML등으로 파싱하여 변환할 수도 있다. 그리고 네트웍에서 사용되는 MIME[5] 같은 프로토콜이나 URL(Uniform Resource Locators)[4]등을 파싱할 수도 있다

CORBA의 IDL(Interface Definition Language)[6]을 파싱하여 stub와 코드를 생성하는 프로그램도 있으며, 기존의 언어로 작성된 프로그램을 새로운 언어로 바꾸어 주는 재공학 분야에서도 이용된다 재공학 분야에서 사용되는 프로그램의 예로는 TXL[3]과 REFINE[3]등이 있다

앞에서 언급한 프로그램 외에도 수많은 어플리케이션들이 파싱 기법을 이용하여 개발되고 있다

3. 그래픽 중심의 파서 제작 도구의 필요성

응용 프로그램 개발에서 파싱을하는 파서를 제작하는 방법은 Lex나 Yacc과 같은 도구들을 사용하는 방법과 이러한 도구를 사용하지 않고 독자적인 방법으로 파싱을 하는 방법이 있다 파서 제작은 수준 높은 프로그래밍 언어와 컴파일러의 지식을 필요로 하며 프로그래밍 기술 또한 필요하다 하지만 이러한 지식과 기술을 갖추지 못한 사람도 파서를 제작할 수 있도록 도와주는 파서 제작 도구들이 있다 파서 개발 도구들을 이용하면 파서를 더욱 쉽게 제작할 수 있다는 장점이 있지만 새로운 도구를 익혀야 한다는 단점도 존재한다 Java 응용프로그램을 개발하려는 사용자가 Yacc과 같은 프로그램을 사용하려면 Yacc의 사용법과 문법을 배워야 한다 만약 Yacc에서의 규칙을 플로우 차트나 아이콘등의 연결로서 표현한다면 사용자가 더욱 쉽게 이해할 수 있을 것이다 또한 Visual C++나 delphi에서 아이콘과 그림의 조합만으로 실행 코드를 완성해주는 것처럼 규칙에 대한 아이콘과 그림의 연결만으로 실행 가능한 코드를 생성해 준다면 초보자도 쉽게 파서를 제작할 수 있을 것이다.

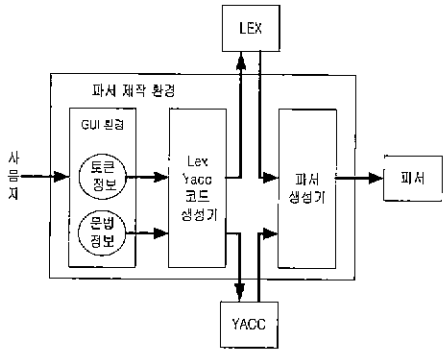
파싱 기술을 이용하는 많은 응용 프로그램들은 필요에 따라서 Lex와 Yacc이나 그 외에 다양한 어휘분석과 구문분석도구들을 이용하였다. 그러나 파싱 기술을 사용하는 모든 응용 프로그램이 모두 이러한 도구들을 사용하는 것은 아니다 개발하는 프로그램의 특성에 따라서 또는 도구 사용의 어려움으로 인하여 이용하지 않을 수도 있다 만약 파서 제작 도구들의 복잡성을 줄이고 더욱 사용하기 쉬운 환경을 제공한다면 파서를 제작하기 위해서 도구를 더욱 쉽게 사용할 수 있을 것이며 사용범위도 더욱 늘어날 것이다.

4. 그래픽 요소를 사용한 파서 제작 도구의 설계

4.1 시스템의 구조

[그림 1]에서 보는 것처럼 파서 제작 도구는 내부적으로 사용자 인터페이스와 그래픽 요소가 표현되는 GUI 환경과, Lex와 Yacc 소스 코드를 생성하는 Lex Yacc 코드 생성기, 그리고 파서를 생성하는 파서 생성기로 나누어진다

사용자는 파서 제작 도구가 제공하는 GUI환경 안에서 그래픽 요소를 이용하여 토큰 정보와 문법 정보를 만든다 Lex Yacc 코드 생성기는 사용자에 의해 만들어진 정보를 가지고 Lex와 Yacc의 입력 화일을 만든다 파서 제작 도구는 만들어진 입력 화일을 가지고 Lex와 Yacc 유틸리티를 호출하여 컴파일 한다 Lex와 Yacc으로 컴파일 하여 생성된 파일을 파서 생성기가 넘겨서 완성된 파서를 사용자에게 제공한다.



[그림 1] 그래픽 요소를 이용한 파서 제작 도구

사용자는 파서 제작 도구의 GUI 환경에서 그래픽 요소의 연결만으로 원하는 파서를 얻을 수 있을 것이다

4.2 Lex 코드에 사용될 그래픽 요소의 설계

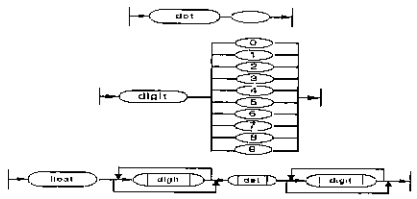
Lex에서는 토큰을 분리하기 위해서 정규표현을 사용한다 Lex는 정규표현을 위하여 많은 기본 기호를 가지고 있다 하지만 Lex의 정규표현에서 사용하는 기호를 모두 사용하지 않고 기본적인 표현의 조합으로 모든 정규 표현식을 나타낼 수 있다 다음의 [표 1]은 정규 표현식을 나타내기 위한 그래픽 요소들이다

그래픽 요소	설명
	시작
	끝
exp	(exp)+ exp가 1번 또는 계속나옴
exp	(exp)? exp가 0번 또는 1번 나옴
exp	(exp)* exp가 0번 또는 계속나옴
token	token이 이미 정의된 표현
token	token이 새로 정의 하는 표현
exp	일반 기호나 문자

[표 1] 정규 표현식의 그래픽 요소

● **그래픽요소를 이용한 정규표현의 예제**

다음의 그림은 실수형 숫자에 대한 정규표현을 [표 1]에 나타난 그래픽 요소를 이용하여 표현한 것이다



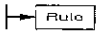
[그림 2] 실수형 숫자에 대한 표현 예

그림 2에서는 먼저 dot라는 token으로 .을 나타내고 digit이라는 token은 0에서 9사이의 수를 그리고 digit과 dot의 조합으로 실수형 숫자를 표현하는 float를 나타낼 수 있다

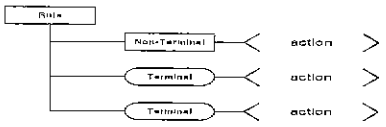
4.3 Yacc 코드에 사용될 그래픽 요소의 설계

4.3.1 yacc 문법의 시각적 표현

- 시작 심볼 파싱의 시작이 되는 Non-terminal을 지정한다. 시작 심볼에 대한 그래픽 요소는 아래 그림과 같다

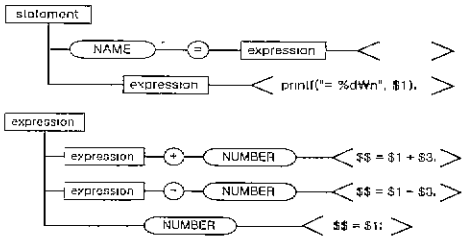


- rule : 문법을 서술하는 부분으로 이것을 이용하여 파싱이 이루어진다
- terminal and token lex에서 추출된 토큰과 yacc에서 파싱에 필요한 토큰을 모두 포함한다
- non-terminal : rule을 표현할 때 왼쪽부분에 나오는 것으로 이것은 오른쪽 부분에 있는 토큰과 non-terminal이 왼쪽에 나오는 하나의 non-terminal로 reduce된다.
- action 문법이 정해진 rule에 맞게 파싱되면 취하는 행동을 지시하는 곳이다. 사용자가 정의한 함수도 호출할 수 있다



4.3.2 그래픽 요소를 이용한 간단한 계산기 예제

다음은 그래픽 요소를 이용하여 덧셈과 뺄셈이 가능한 계산기의 문법[3]을 구현한 예이다



[그림 3] 계산기에 대한 문법 예

[그림 3]에서 NAME은 문자로 이루어진 토큰이고 NUMBER는 숫자로 이루어진 토큰이다 NAME과 NUMBER는 사용자에 의해서 정의되었다고 가정하였다. statement는 NAME = expression 또는 expression으로 이루어지며, expression은 expression +/- NUMBER 혹은 NUMBER로 이루어 진다. 또한 각각의 규칙에 대한 액션에 의해서 입력 숫자와 연산자를 가지고 계산된 결과를 출력한다.[1]

5 결론

파싱은 많은 응용 프로그램에서 사용되는 기술이다 파싱을 하기 위해서는 파서를 제작해야 하는데 이때 도구를 사용할 수 있다 많은 파서 제작 도구들이 있으나 이 도구들이 파싱을 하는 응용 프로그램에서 보편적으로 사용되지는 않는다 만약 더욱 이해하기 쉽고 사용하기 편한 파서 제작 도구가 개발 된다면 사용자들이 쉽게 파서를 제작할 수 있을 것이다

본 연구에서는 그래픽 요소의 결합만으로 파서를 제작할 수 있는 도구를 설계 하였다. 이 도구를 이용한다면 전문적인 지식이 없는 사용자도 쉽게 파서를 제작할 수 있을 것이며 파싱을 필요로 하는 많은 응용분야에서 사용이 가능할 것이다.

참고문헌

[1] John R. Levine, Tony Mason & Doug Brown. *lex & yacc*, O'Reilly & Associates, Inc. 1995.

[2] Yacc++ and The Language objects Library, Compiler Resources, Inc
Available http://www.cs.jcu.edu.au/~alison/TONY/yacc++.html

[3] Alex Sellink, Chris Verhoef, *Current Parsing Techniques in Software Renovation Considered Harmful*, 1998
Available http://adam.wins.uva.nl/~x/ref/ref.html

[4] RFC 1808 Network Working Group Request for Comments 1808 Category : standards Track: R Fielding UC Irvine, 1995.

[5] Building and Parsing MIME Messages
Available http://devedge.netscape.com/docs/manuals/messaging/msgjava/asd3j.html

[6] Lexer and Parser Generators
Available http://www.cs.jcu.edu.au/~alison/TONY/lexparse.html