LR 파싱에서 충돌 해결을 위한 Shift 우선 전략

이용석, 황이규
전북대학교 컴퓨터과학과

Shift-first Strategy for Resolving Conflicts in the LR Parsing

Yong-seok Lee, Yi-Gyu Hwang
Dept. of Computer Science, Chonbuk National University

요약

LR 파싱은 프로그래밍 언어를 위한 빠른 파싱 방법을 제공한다. 그러나 이 방법의 단점은 자연어와 같은 다양한 모호성을 가지는 문법에 적합하지 못하다. 모호성을 가지는 문법은 파싱 테이블 상에서 충돌을 야기하게 되는데 이를 해결하는 방법에 대한 연구가 많이 있어 왔다. 문장이 길어질 경우 구문 분석 도중 이러한 모호성이 파싱 효율에 큰 영향을 미치게 되는데, 본 논문에서는 Shift 우선 전략으로 LR 파싱의 효율적인 특징을 유지하면서 이러한 충돌을 해결할 수 있음을 보인다.

1. 서론

LR파싱 알고리즘[1]은 프로그래밍 언어를 위한 효과적인 파싱 방법으로 널리 알려져 있다. 이 방법은 다음 상태에서 어떠한 동작을 수행해야 될지를 파싱 테이블에 의해 결정적으로 유도되는 Shift-Reduce 파싱 방법이다. 그러나 자연어는 프로그래밍 언어에 비해 훨씬 복잡한 언어적 현상을 포함하고 있기 때문에 문장의 분석 도중 한 상태에서 명확히 다음 상태를 결정할 수 없는 경우가 많다. 이러한 경우를 파싱 테이블내의 충돌(conflict)이라고 한다. 이러한 충돌이 곧 모호성을 야기한다. 자연어를 분석하는데 가장 큰 장애는 이러한 모호성의 발생을 들 수 있다.

파싱 테이블내에서 충돌의 종류는 Shift-Shift, Reduce-Reduce, Shift-Reduce의 3가지로 나눌 수 있다. 이중 자연어 문장을 분석하는데 있어서 가장 문제가 되는 충돌은 Shift-Reduce이다. 이러한 자연어의 모호성을 LR 파싱 방법 안에서 효과적으로 허용하기 위한 연구가 Tomita의 일반화된 LR 파싱 방법(Generalized LR)[4]이다. 이 방법은 구문 분석 도중 발생하는 충돌을 모두 허용하여 그래프의 형태로 구문 분석 트리를 유지한다. 이를 위해 파싱을 위한 자료 구조로 GSS(Graph-Structured Stack)을 이용한다. 본 논문에서는 자연어 문법에서 발생할 수 있는 다양한 충돌과 이를 해결하기 위한 방법, LR 파싱을 확장하여 shift-reduce 충돌을 reduce-reduce 충돌로 전환시켜 구문정보 외적인 지식에 의한 해결이 가능하도록 하는 방법에 대하여 기술한다.

2. 문법의 충돌과 모호성

아래와 같은 입력문장을 구문 분석하기 위한 문법을 고려해 보면 그림 1과 같다.

입력 문장 : I saw a man in the park.

① S -> NP VP
② S -> S PP
③ NP -> *n
④ NP -> *det *n
⑤ NP -> NP PP
⑥ PP -> *prep NP
⑦ VP -> *v NP

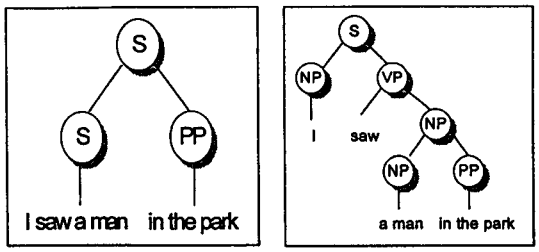
그림 1 모호한 문법의 예

그림 1에 나타나는 문법을 바탕으로 문장을 구문 분석하기 위해 파싱 테이블을 구성해 보면 그림 2와 같다.

State	*det	*n	*v	*prep	$	NP	PP	VP	S
0	sh3	sh4				2			1
1				sh6	acc		5		
2			sh7	sh6			9	8	
3		sh10							
4			re3	re3	re3				
5				re2	re2				
6	sh3	sh4				11			
7	sh3	sh4				12			
8				re1	re1				
9			re5	re5	re5				
10			re4	re4	re4				
11			re6	re6,sh6	re6		9		
12				re7,sh6	re7		9		

그림 2 파싱 테이블

여기에서 상태 11과 12에서 전치사 *prep를 보고 두 가지 동작을 수행 할 수 있다. 이것은 영문법에서 자주 발생하는 전치사구 접속(PP attachment)와 관련이 있다. 이러한 경우 전치사가 동사와 관련되는지, 또는 명사와 관련되는지 모호성이 발생하게 된다. 두 가지 경우를 파스트리로 그려보면 다음과 같다.



이러한 모호성을 해결하기 위한 기존의 연구는 여러 가지가 있었다. 그 대표적인 예를 몇 가지 들면 첫째, Right Association(RA)[2]을 들 수 있다. RA는 "구구조가 결합할 때 다른 정보의 도움이 없는 한, 구(Phrase)는 가능한 한 부분 분석된 오른쪽에 붙는다"라는 원칙을 일반화한 것이다. 예를 들어 아래와 같은 문장을 생각해 보면,

"Tom said that Bill had taken the cleaning out yesterday."

에서 "yesterday"가 전체 문장과 관계가 있는지 "Bill had taken the cleaning out"과 관계가 있는지 모호할 경우, 이미 분석된 구문 구조중 오른쪽에 있는 "Bill had taken the cleaning out"와 관계가 있다고 가정하는 것이다. 이 경우는 LR 파싱상의 파싱 테이블에서 한 상태에서 shift와 reduce충돌이 발생했음을 알 수 있다. 일반적인 경우 이러한 문제는 shift를 선택함으로써 해결한다. 그러나 여기에서 무조건 shift를 선택하고 reduce를 고려하지 않기 때문에 일반적이지 못하다. 위에서 "I saw a man in the park" 예제의 경우, 후자를 선택했음을 알 수 있다.

두 번째 방법으로 Minimal Attachment (MA)가 있다. MA는 "다른 정보의 도움이 없는 한, 구(Phrase)는 분석의 복잡도를 최소화하는 쪽에 붙는다"라는 원칙을 일반화한 것이다. 예를 들면,

"John bought the book for Susan."

라는 문장이 있을 경우, "for Susan"이 "bought"와 관계가 있는지 또는 "the book"과 관계가 있는지 모호할 경우, 복잡도를 최소화하는 "bought"와 관계가 있는 것으로 한다. 이것은 reduce-reduce충돌과 관련이 있는 것으로 파싱되는 스택내에서 reduce되는 심볼이 많은 쪽으로 진행한다.

또 다른 방법으로는 모호성이 발생하는 경우, 모두를 허용하는 GLR 방법을 사용하는 것이다. 그림 3은 "I saw a man in the park"를 그림 1의 문법으로 GLR 방법을 이용하여 구문 분석하는 과정을 보여주고 있다.

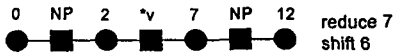


그림 3 파싱 과정(1)

상태 12에서 *prep(in)을 보고 두 가지 동작(reduce 7, shift 6)을 할 수 있다. 이러한 경우 GLR은 두 동작 모두를 행한다. 두 동작 모두를 행한 후의 모습은 그림 4와 같다.

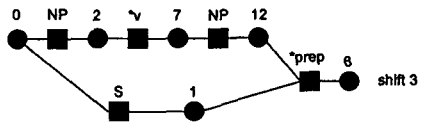


그림 4 파싱 과정(2)

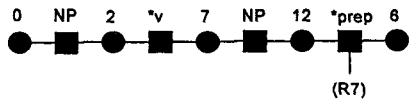
GLR의 경우, 문장이 길어지면 구문 분석을 위해 유지해야 하는 GSS가 부담으로 작용한다.

자연어에서 발생하는 모호성의 대부분은 shift-reduce 충돌에서 나타나는데, 본 논문에서는 shift-reduce 충돌시 우선적으로 shift를 선택하며, 이때 선택되지 않은 reduce를 버리지 않고 보존하며 이러한 방법이 LR 파싱 방법 안에서 효과적으로 수행될 수 있음을 보인다.

3. Shift 우선 방법

이 장에서 우리는 shift-reduce 충돌의 해결을 위한 방법을 제안한다[5, 6]. 본 논문에서는 문법적 제약의 표현을 위한 다양한 기호와 문자열들을 표현하기 위해 [1]의 방법을 사용한다. shift-reduce 충돌이란 위에서 설명한 바와 같이 문장을 분석하는 한 상태에서 주어진 단어를 가지고 shift와 reduce 동작을 모두 행할 수 있는 상태를 말한다. 그림 3의 경우, shift 동작과 reduce 동작을 모두 행할 수 있다. 이와 같은 경우를 shift-reduce 충돌이라 한다. 본 논문에서 제안하는 방법을 설명하기 전에 몇 가지 용어를 아래와 같이 정의한다.

δ-shift : shift-reduce 충돌이 발생할 때, shift 우선 전략에 의해 LR파서 스택에 shift된 문법 심볼은 shift-reduce 충돌이 있었다는 것을 표시하기 위해 δ-shift를 태그한다.



(lookahead : the, action : shift 3)
그림 5 δ-shift의 예(3)

그림 5의 경우, 그림 3에서 shift 우선 방법에 의해 [shift 6]을 선택하고 [reduce 7]은 δ-shift된 경우를 나타낸다.

δ-전파 : A -> β에 의해 문법 스트링 β가 reduce 될 때, 만약 β의 가장 왼쪽 심볼이 δ-shift를 가지면 축약된 심볼 A는 마찬가지로 δ-shift로 태그된다. 즉, δ-shift는 전파된다.

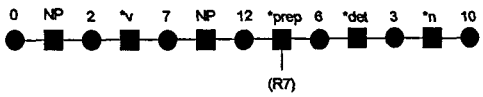
순수 축약(pure reduction)과 비순수 축약

(impure reduction) : A -> β 규칙에 의해 문법 스트링 β가 축약될 때, 만일 가장 왼쪽 심볼을 제외한 β의 모든 심볼이 δ-shift를 가지지 않는다면 우리는 이 축약을 순수 축약이라 부른다. 비순수 축약은 순수 축약이 아닌 축약을 말한다.

순수하게 결정적인 파싱 : 순수 축약만을 사용하여 파싱이 결정적으로 진행되는 파싱을 말한다.

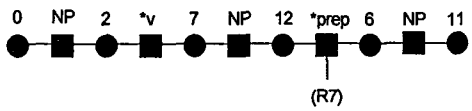
강한 심볼 : 자연어의 LR파싱에서, 상당히 모호함에도 불구하고, 강한 심볼로 파싱될 수 있는 요소가 존재한다. 이 요소들을 강한 절이라 하면, 강한 절의 부분적 요소는 또 다른 강한 절을 만들기 위해 다른 강한 절의 부분적 요소와 결합할 수 없다. 예를 들어, 다음 문장을 보면, 그림 1의 입력 문장에서 "I", "saw", "a man", "in", "the park"는 모두 강한 절이다. 그러므로, "man in", "in the"는 모두 강한 절이 될 수 없다.

문장 "I saw a man in the park"를 예로 계속 파싱해 가보면 그림 5에서 "the"를 보고 "shift 3", "park"를 보고 "shift 10" 을 한 후 그림 6과 같다.



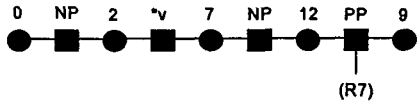
(lookahead : $, action : Reduce 4)
그림 6 파싱과정(4)

그림 6에서 '$'를 보고 reduce 과정을 반복하게 된다.



(lookahead : $, action : Reduce 6)
그림 7 파싱 과정(5)

그림 7에서 "reduce 6"을 하게 되는데, 이것이 앞에서 정의한 순수 축약이 된다. 순수 축약의 경우, 동작을 행해도 δ-shift에 의해 연기된 축약(Reduce 7)에는 영향을 미치지 않는다. 그림 8은 그림 7에서 δ-전파가 된 것을 알 수 있다.



(lookahead : $, action : Reduce 5)
그림 8 파싱 과정(6)

그림 8에서 "reduce 5"를 행할 때, 비순수 축약이 발생하게 되는데, 여기서 연기된 축약 "reduce 7"을 할지 현재 축약인 "reduce 5"를 행할 지 결정해야 한다. 즉, 앞에서 shift-reduce 충돌을 해결하기 위해 shift우선을 설정한 후 이것이 결국 reduce-reduce 충돌로 바뀌게 된 것이다. 그림 8에서 "reduce 5"를 선택할 경우, 일반적인 LR파싱 방법과 같이 진행을 하면 된다. 이 경우 PP가 NP에 접속되는 경우를 나타낸다. 그러나 연기된 축약(reduce 7)을 선택할 경우, 기존 LR 파싱 방법과 다른 메커니즘이 필요하다. 우리는 이 메커니즘을 의사 파싱이라 정의한다.

의사 파싱(Pseudo Parsing) : 연기된 축약을 한 후의 파싱을 위한 스택상에서 다음 동작은 현상태와 lookahead의 FIRST 심볼로 결정하는데 shift 동작일 경우, 다음 상태는 현상태와 다음 lookahead 심볼로 결정한다. 다른 경우는 일반 LR 파싱 방법과 같다.

그림 8을 예로 들어 설명해 보면 다음과 같다. 만일 "reduce 5"를 선택했다면 일반적인 방법을 통해 파싱을 해나가면 되지만 연기된 축약 "reduce 7"을 선택했을 때, 축약 후 의사 파싱을 행하게 된다(lookahead = VP).

다음 동작 : Action[2, FIRST[VP]] = shift 7
다음 상태 : GOTO[2, VP] = 8

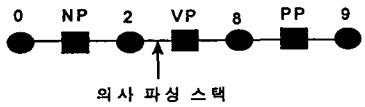


그림 9 의사 파싱 과정(1)

따라서 다음 상태는 다음과 같다.

다음 동작 : Action[8, FIRST[PP]] = reduce1
다음 상태 : GOTO[1, PP] = 9

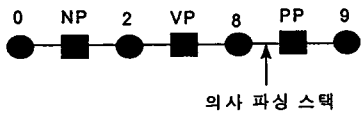


그림 10 의사 파싱 과정(2)

의사 파싱이 끝난 후의 상태를 살펴보면 아래와 같다.

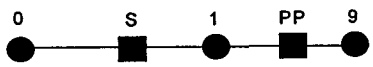


그림 11 의사 파싱이 끝난 후

따라서 연기된 축약이 선택될 경우, 이는 PP가 문장 전체와 관계가 있음을 의미한다. 결국 shift-reduce 충돌은 reduce-reduce 충돌로 전환될 수 있으며, reduce-reduce충돌은 의미 지식(의미 제약)이 충분하면 Shieber의 방법[3]에 의해 해결 가능하다. 또한 앞에서 정의한 강한 심볼도 결국 reduce-reduce 충돌 해결을 위한 정보원으로 활용될 수 있다. 예를 들어 "The soldier aimed the rifle at the target" 문장의 경우, "aim"과 "at"은 강한 결합을 이루고 있음을 알 수 있다. 따라서 이러한 정보원을 통해 불필요한 구문 트리를 제거함으로써 파싱의 효율 향상을 기대할 수 있다.

4. 결론

프로그래밍 언어를 기계로 처리하기 위한 방법 중 대표적인 LR 파싱 방법으로 자연어를 처리하고자 할 때 발생하는 문법 충돌의 유형을 살펴보고 이의 일반적인 해결 방법과 shift-reduce 충돌의 해결을 위한 shift 우선 전략에 대해 기술하였다. 이를 통해 구문 분석 도중 다양한 구문-의미정보 지식의 활용을 통해 모호성을 해결할 수 있음을 보였다.

참고 문헌

[1] Aho, A. V. and Ullman, J. D. The Theory of Parsing, Translation and Compiling vol. 1., Prentice-Hall, Englewood Cliffs, N. J., 1972.

[2] Pereira, Fernando C. N., "A new characterization of attachment preferences", Natural Language Parsing Systems, Cambridge University Press, pp.307-319, 1985.

[3] Shieber, S. M. "Sentence Disambiguation by a Shift-Reduce Parsing Technique", Proceedings of the Eighth International Joint Conference on AI, Vol.2, 1983.

[4] Tomita, M., Efficient Parsing for Natural Language : A Fast Algorithm for Practical Systems, Kluwer Academic Press, 1986.

[5] Yong-Seok Lee, Hideto Tomabechi, Jun-ichi Aoe, "A Shift-First Strategy for Interleaved LR Parsing", Information Sciences : An International Journal, Vol. 84, pp.1-14, 1995.

[6] Yong-Seok Lee, Hideto Tomabechi, Jun-ichi Aoe, "A Shift-First Strategy for Generalized LR Parsing", IEICE Transactions on Information and Systems, Vol. E77-D, No. 10, October, 1994.