

컴파일러의 기초: Homework 3 for Problem Solving (Due: 4/15 (수))

Problem 1. (20)

다음의 Grammar 에 대하여 SLR(1) parser 를 만드려고 한다.

- $S' \rightarrow S$
- $S \rightarrow Aa \mid Bb$
- $A \rightarrow Ac \mid \epsilon$
- $B \rightarrow Bc \mid \epsilon$

(a) LR(0) 의 start state 를 만들고 A 와 B 의 $Follow()$ set 을 구하여 SLR(1) grammar 가 안됨을 확인하시오. (5)

(b) 위의 grammar 가 LR(1) 인가를 확인 하시오. (5)

(c) Shift-Reduce Parsing 에서 왜 left-recursive 한 grammar 가 유리한 가를 설명하시오 (Hint: c^* 를 recognize 하는 right-recursive grammar 를 이용하여 Shift/Reduce Parsing 을 할 경우를 생각해 보시오. 만약 수 백만개의 c 로 구성된 input 을 parsing 하는 경우 어떤 일이 생기는 가?). (5)

(d) 위의 production 을 SLR(1) grammar 가 되게 하려면 A 와 B 의 production 을 어떻게 바꾸어야 하는가? 왜 그런가? (5)

Problem 2. (30)

강의 시간에 다룬 dangling-else 문제를 가진 다음의 ambiguous grammar G 에 대하여 답하시오.

- $S' \rightarrow S$
- $S \rightarrow i E t S$
- $S \rightarrow i E t S e S$
- $S \rightarrow a$
- $E \rightarrow b$

(a) LR(1) DFA 를 만들고 start state 로부터 breath-first 순서로 나열 하시오 (5).

(b) State 를 merge 하여 LALR(1) DFA 를 만드시오. (5).

(c) 위의 LR(1) DFA 와 LALR(1) DFA 에서 $S \rightarrow i E t S \cdot$ 와 $S \rightarrow i E t S \cdot e S$ 만을 포함하는 state 를 찾아 해당 parsing table row 를 만드시오. (5)

(d) 위의 LALR(1) parsing table row 에서 conflict 를 가진 entry 에 대하여 dangling else 가 innermost if 에 bound 되도록 action 을 선택 하시오 (5).

(e) (d) 에서 outermost if 에 bound 되도록 action 을 선택하고 이 경우 G 의 language 를 제대로 parse 하지 못함을 예를 들어 보이시오 (5).

(f) (e) 의 선택을 LR(1) parsing table 에 적용할 경우 올바르게 G 의 language 를 parse 하는지 확인하고 설명 하시오 (5).