

Plan for Today

Error recovery goals

Review panic mode error recovery for predictive parsers

Panic mode for LR parsers

Error recovery using error symbol in productions

Error Handling Goals

Provide program with a list of as many errors as possible

Provide USEFUL error messages

- appropriate line and position information
- guidance for fixing the error

Avoid infinite loops or recursion

Add minimal overhead to the processing of correct programs

Find "all" errors in program before translation begins

Predictive parser for Float Assignment Grammar

```
void S() { switch (lookahead) {
  case ID:
    case EOF: // the 2 tokens in the FIRST(SemLise EOF)
      try { SemLise(); match(EOF); } catch { panic(S); } break;
    default: panic(S); break;
}}
void SemLise() { switch (lookahead) {
  case ID: // FIRST(Sem SemLise) = { ID }
    try { Sem(); SemLise(); } catch { panic(SemLise); } break;
  case EOF: // FOLLOW(SemLise) = { EOF }
    break;
  default: panic(SemLise); break;
}}
void Sem() { switch (lookahead) {
  case ID: try { match(ID); match(ASSIGN); match(FLOAT);
    } catch { panic(Sem); } break;
  default: panic(Sem); break;
}}
```

Grammar 3.1 from Tiger book

```
(0) S' -> S $
(1) S -> S ; S
(2) S -> id := E
(3) S -> print (L)
(4) E -> id
(5) E -> num
(6) E -> E + E
(7) E -> (S, E)
(8) L -> E
(9) L -> L, E
```

LR parse table (Table 3.19 from Tiger Book)

	id	num	print	;	,	+	:=	()	\$	S	E	L
1	r4		r7								g2		
2				r3						r			
3	r4		r7								g5		
4							r6						
5				r1	r1					r1			
6	r20	r10						r8				g11	
7								r9					
8	r4		r7								g12		
9	r20	r10						r8				g15	g14
10				r5	r5	r5			r5	r5			
11				r2	r2	r16				r2			
12				r3	r18								
13				r5	r5					r5			
14					r19					r13			
15					r8					r8			

CS453 Lecture

Error Recovery II

5

LR parse table (Table 3.19 from Tiger book) cont...

	id	num	print	;	,	+	:=	()	\$	S	E	L
16	r20	r10						r8				g17	
17				r6	r6	r16			r6	r6			
18	r20	r10						r8				g21	
19	r20	r10						r8				g23	
20				r4	r4	r4			r4	r4			
21									r22				
22				r7	r7	r7			r7	r7			
23					r9	r16			r9				

CS453 Lecture

Error Recovery II

6

Error recovery using an error symbol

```
exp -> ( error )
expr -> error ; exp
```

Steps taken when error occurs:

- (0) generate error indicating expected token(s)
- (1) pop off stack until have state with shift action for error token
- (2) shift the error token
- (3) throw away input tokens until hit token with non-error action
- (4) resume parsing

CS453 Lecture

Error Recovery II

7

Suggested Exercises

Show the Stack, Input, and Action table (see Figure 3.18 in handout) using the parse table on slides 5 and 6 where the parser is using panic mode recovery for the following inputs:

`:= b + c - ; $`

`(d := 5 + 6, 3) $`

`(()) $`

CS453 Lecture

Error Recovery II

8