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
- available in YACC

Lines and positions information
- in general how could it be improved?
- can we actually implement such improvements with SableCC?

Where should an overflow number error be handled?

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
- Find “all” errors in program before translation begins

Predictive parser using panic mode error recovery

```c
void S() {
    switch (tok) {
        case ID:
            case EOF:
                try {
                    StmList();
                } catch {
                    panic(FOLLOW(S));
                } break;
        default: print "error";
    panic(FOLLOW(S));
            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)**

<table>
<thead>
<tr>
<th>s1 n1 print</th>
<th>;</th>
<th>,</th>
<th>+</th>
<th>=</th>
<th></th>
<th></th>
<th></th>
<th>S</th>
<th>S</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>s4</td>
<td>s7</td>
<td></td>
<td></td>
<td></td>
<td>g2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>s4</td>
<td>s3</td>
<td></td>
<td></td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>s4</td>
<td>s7</td>
<td></td>
<td></td>
<td>g5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s6</td>
<td></td>
<td></td>
<td></td>
<td>g9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>r7</td>
<td>r7</td>
<td>r7</td>
<td>r7</td>
<td>r7</td>
<td>g13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>s20</td>
<td>s10</td>
<td></td>
<td></td>
<td>s8</td>
<td>g11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>s9</td>
<td></td>
<td></td>
<td></td>
<td>g12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>s4</td>
<td>s7</td>
<td></td>
<td></td>
<td>g15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>s20</td>
<td>s10</td>
<td></td>
<td></td>
<td>s8</td>
<td>g14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>r5</td>
<td>r5</td>
<td>r5</td>
<td>r5</td>
<td>g15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>r5</td>
<td>r5</td>
<td>r5</td>
<td>r5</td>
<td>g15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>r2</td>
<td>r2</td>
<td>s16</td>
<td>r2</td>
<td>g13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>s3</td>
<td>s18</td>
<td></td>
<td></td>
<td>g13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>s3</td>
<td>s18</td>
<td></td>
<td></td>
<td>g13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>r8</td>
<td>r8</td>
<td></td>
<td></td>
<td>g13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LR parse table (Table 3.19) cont...**

<table>
<thead>
<tr>
<th>s1 n1 print</th>
<th>;</th>
<th>,</th>
<th>+</th>
<th>=</th>
<th></th>
<th></th>
<th></th>
<th>S</th>
<th>S</th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>s20</td>
<td>s10</td>
<td></td>
<td></td>
<td></td>
<td>g17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>s20</td>
<td>s10</td>
<td>r6</td>
<td>r6</td>
<td>s16</td>
<td>r6</td>
<td>r6</td>
<td>g21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>s20</td>
<td>s10</td>
<td></td>
<td></td>
<td>s8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>s20</td>
<td>s10</td>
<td></td>
<td></td>
<td>s8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>r4</td>
<td>r4</td>
<td>r4</td>
<td>r4</td>
<td>g23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>s22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>r7</td>
<td>r7</td>
<td>r7</td>
<td>r7</td>
<td>r7</td>
<td>r7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>r9</td>
<td>s16</td>
<td>r9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Error recovery using an error symbol

- **exp** -> ( error )
- **exp**s -> 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

### Suggested Exercises

Write a predictive parser using panic mode error recovery for the grammar shown below.

- **(0)** S -> E $  
- **(1)** E -> B E'  
- **(2)** E' -> ‘or’ B E'  
- **(3)** E' ->  
- **(4)** B -> t l f

Show the Stack, Input, and Action table (see Figure 3.18) using the parse table in Figure 3.19 where the parser is using panic mode recovery for the following inputs:

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