The scanner begins operation started from state S, after recognizing a digit makes transition to state 1, the first nondigit encountered causes a transition to the state R (d → “0|1|…|9”).
Identifier and unsigned integer

- **S**
  - Digit
  - Letter
  - Letter/digit

- **1**
  - Digit
  - Any other

- **2**
  - Digit
  - Any other

- **R**
  - Any other
Procedure SCAN(token, type)
{char and class are global}
  token ← empty
  prevstate ← start
  currstate ← statetable[prevstate, class]
  while currstate ≠ ‘R’ do
    token ← token, char {concatenation}
    GETCHAR(char, class)
    prevstate ← currstate
    currstate ← statetable[prevstate, class]
    type ← TYPE(prevstate)

<table>
<thead>
<tr>
<th>state</th>
<th>digit</th>
<th>letter</th>
<th>other</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lexical class

State-Transition Table
Rules for recognizing unsigned numbers

unsigned_real ↔ unsigned_integer fraction
    | unsigned_integer scale_factor
    | unsigned_integer fraction scale_factor

unsigned_integer ↔ digit+

fraction ↔ ‘.’ unsigned_integer

scale_factor ↔ ‘E’ [sign] unsigned_integer

sign ↔ ‘+’|’-’
Unsigned numbers

Diagram:
- S to 1 with d
- 1 to 3 with d
- 3 to 5 with d
- 5 to R with d
- R to 7 with d
- 7 to other with d
- 5 to other with d
- 4 to other with E
- 3 to other with E
- 1 to other with d
- 4 to 6 with +/–
- 6 to 7 with d
Problems to solve

- Classification of tokens (keywords, identifiers, operators, etc.),
- Identifying of tokens – the token has to be isolated from its neighbors (delimiters, etc.),
  
  for free-format languages 3.14*NUM

- Sometimes lookahead is needed - cannot be consumed because it might be part of another token,
  
  Do 10 I = 1,5  or  DO 10 I = 1.5

- not reserved keywords
  
  if then then then = else; else else = then;

- Some statements can creates new keyword in the language
  
  typedef int new_type;

- And others