

Input Language Grammar

We'll use a slightly modified version of the simple expression grammar from *bnf1.txt*:

$$S ::= \{ E1 \}$$
$$E ::= F E1$$
$$F ::= \text{Num} \mid \text{Id}$$
$$E1 ::= E1a \mid \text{Empty}$$
$$E1a ::= + E$$

Semantics Specification

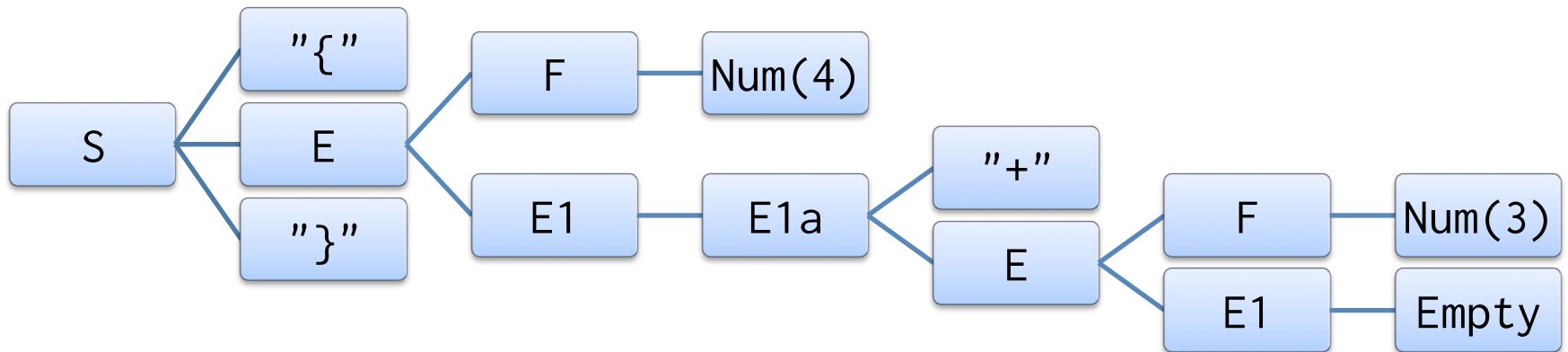
```
{  
  "atomic_mapping": {  
    "+": "ADD",  
    "{": "PRINT"  
  },  
  "reduction_symbols": ["S", "E"]  
}
```

Input Program

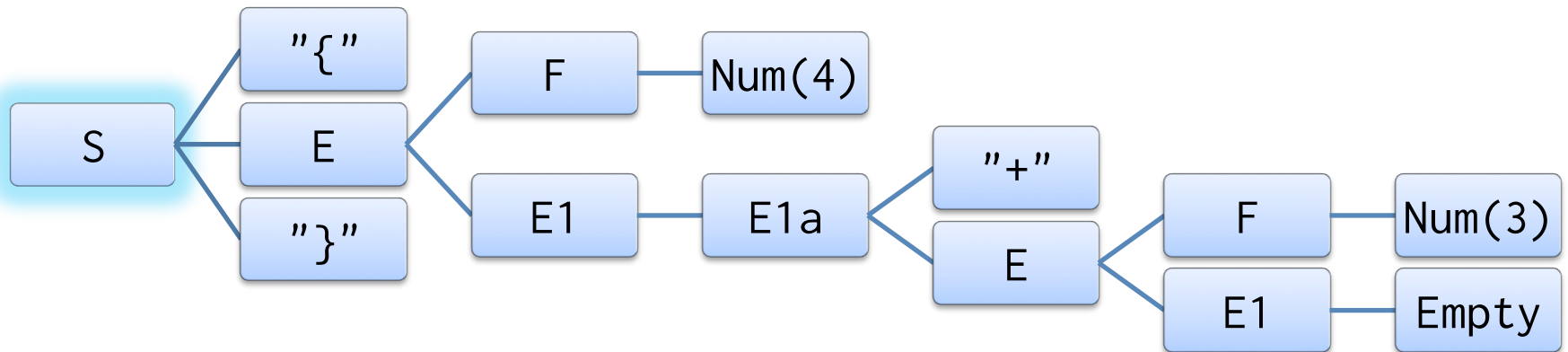
We'll use a very simple input:

$$\{ \ 4 \ + \ 3 \ }$$

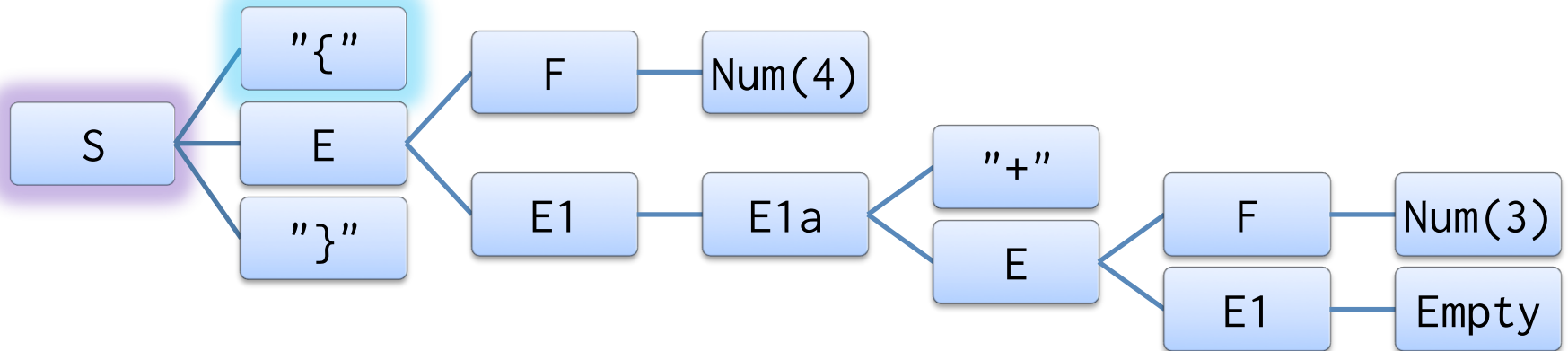
Parse Tree



Parse Tree

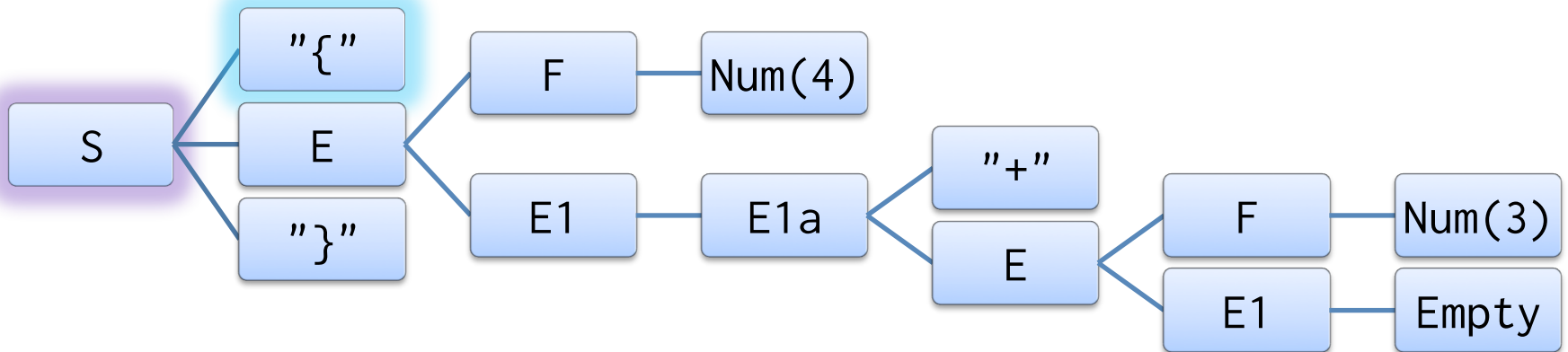


Parse Tree

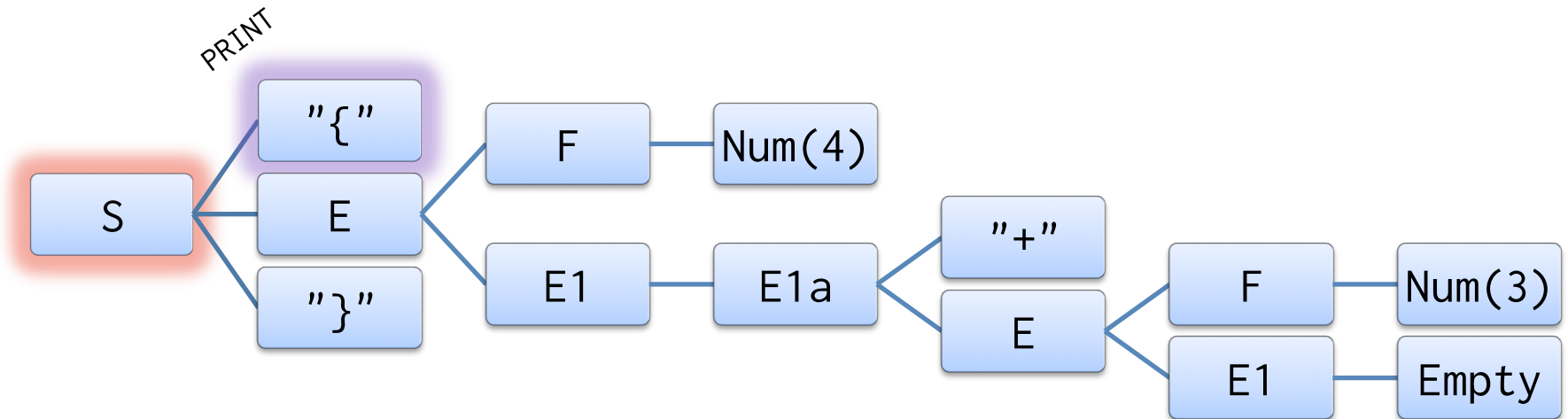


Parse Tree

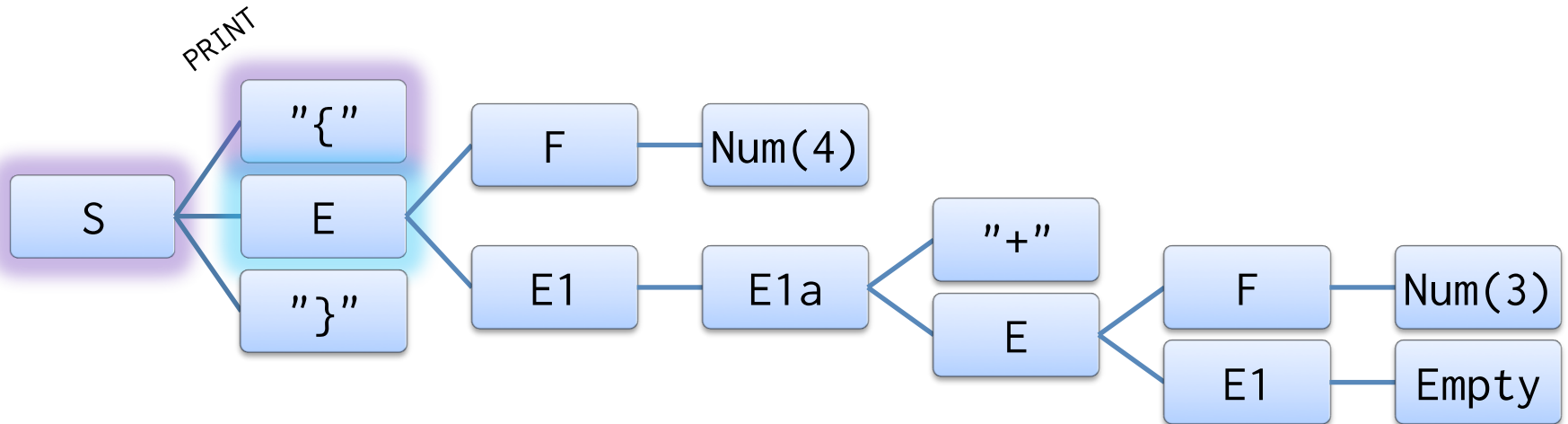
"}": "PRINT"



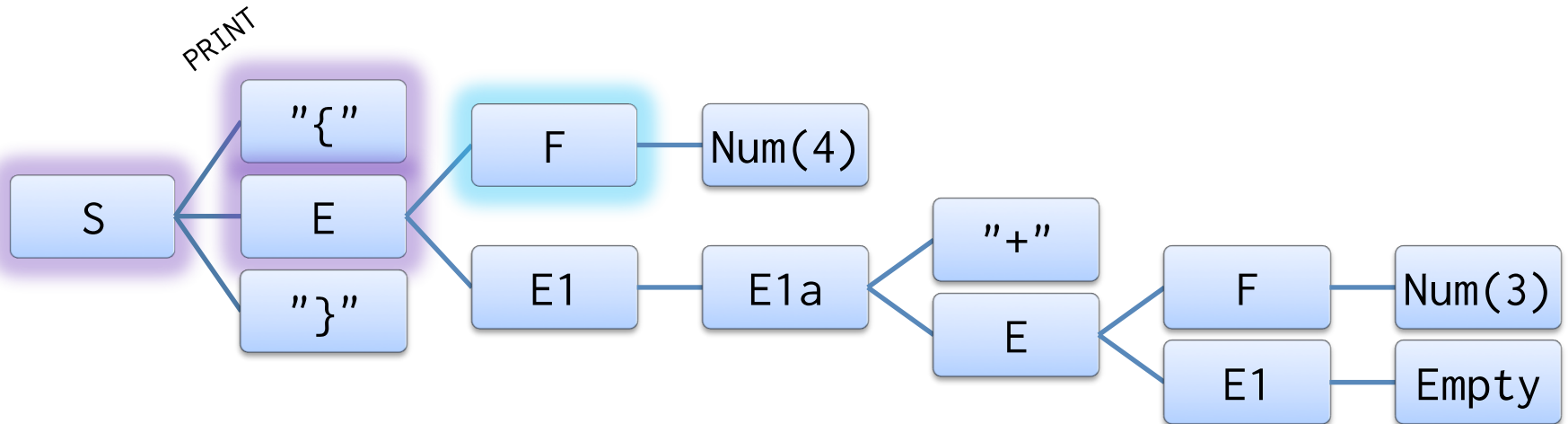
Parse Tree



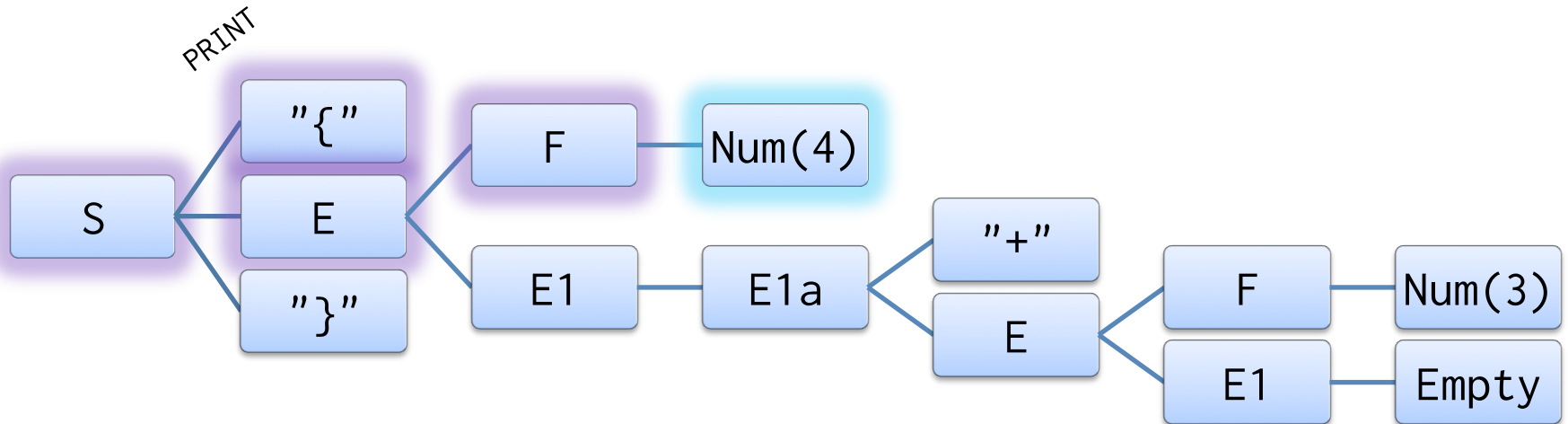
Parse Tree



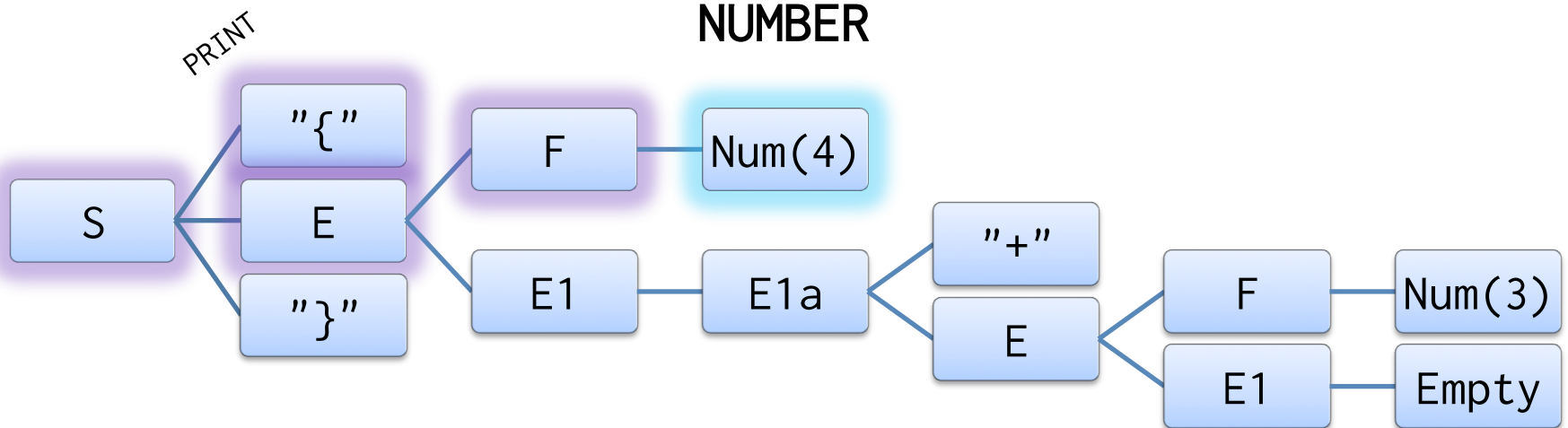
Parse Tree



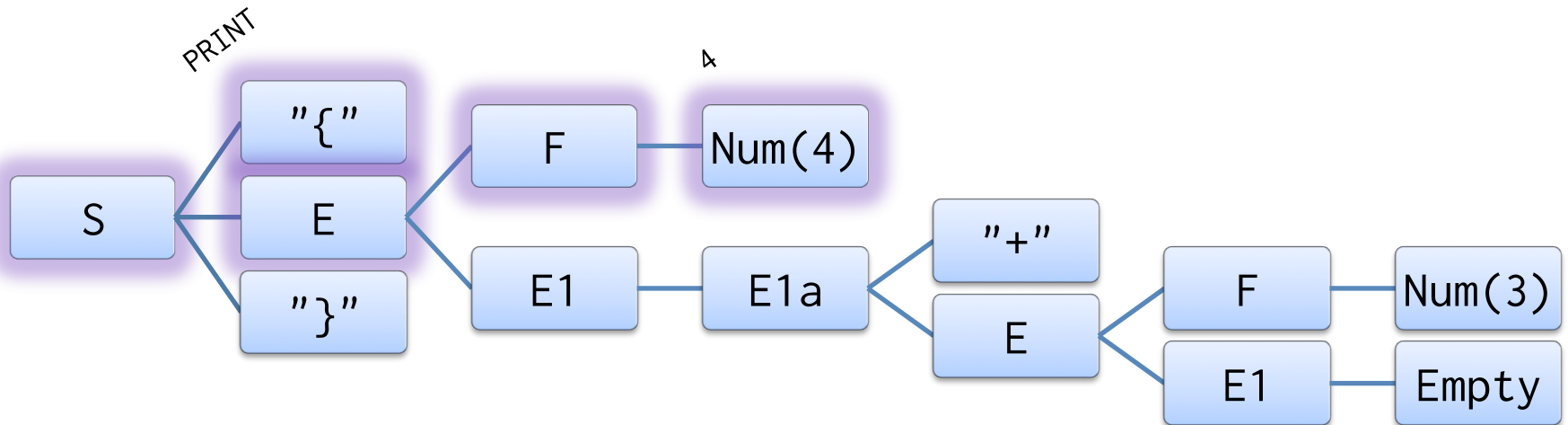
Parse Tree



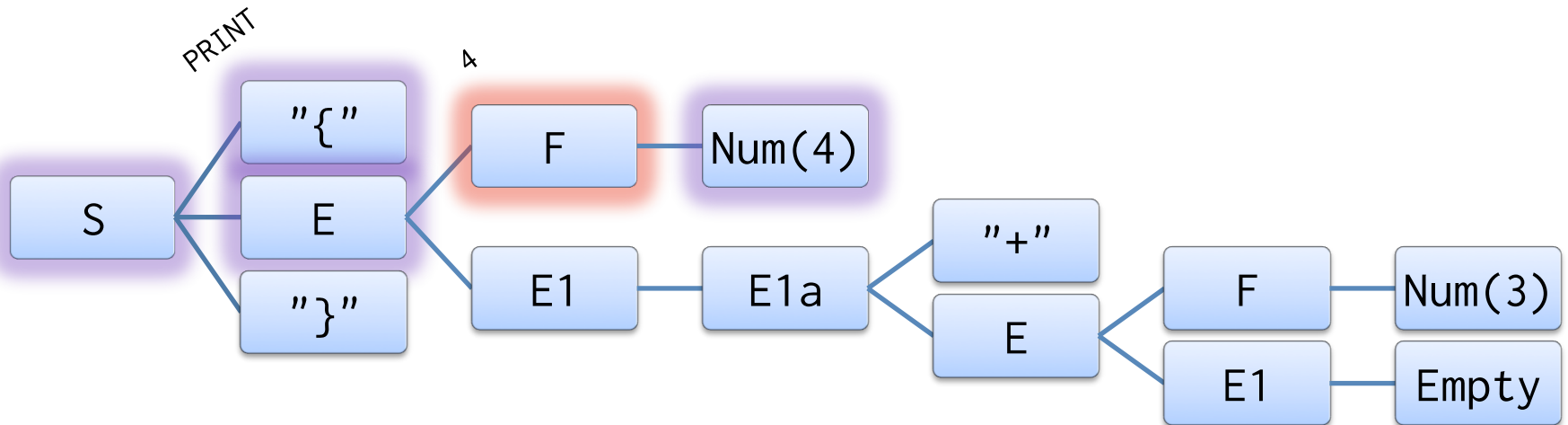
Parse Tree



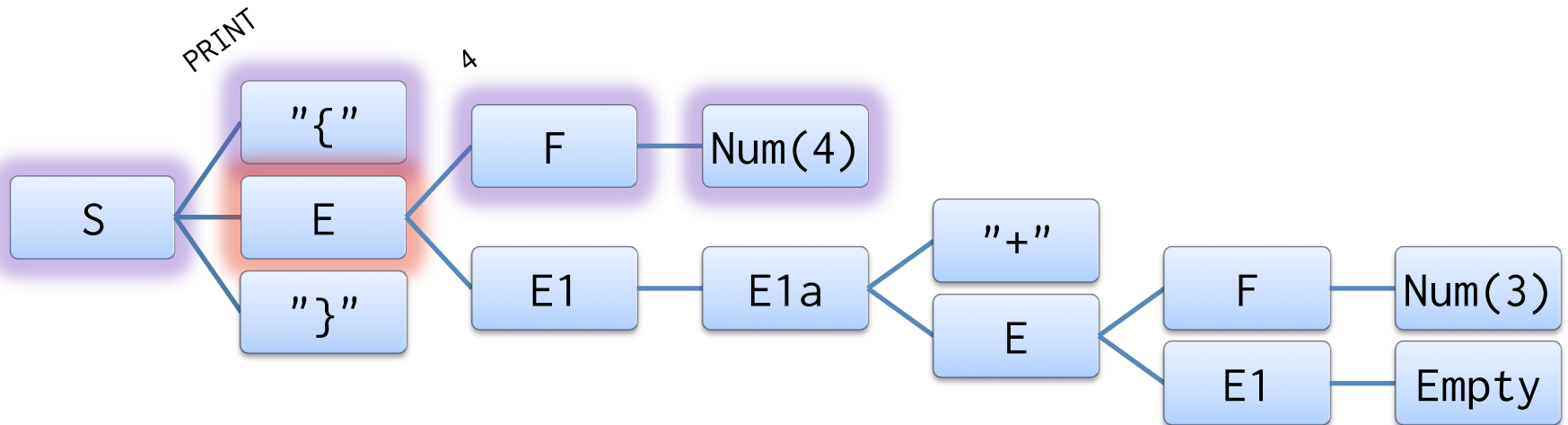
Parse Tree



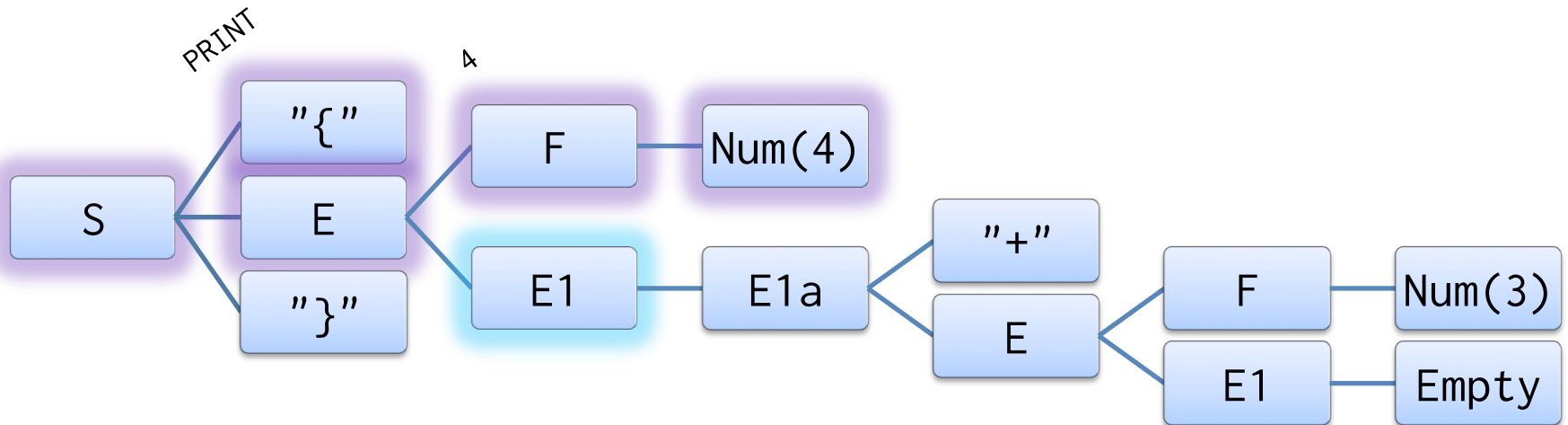
Parse Tree



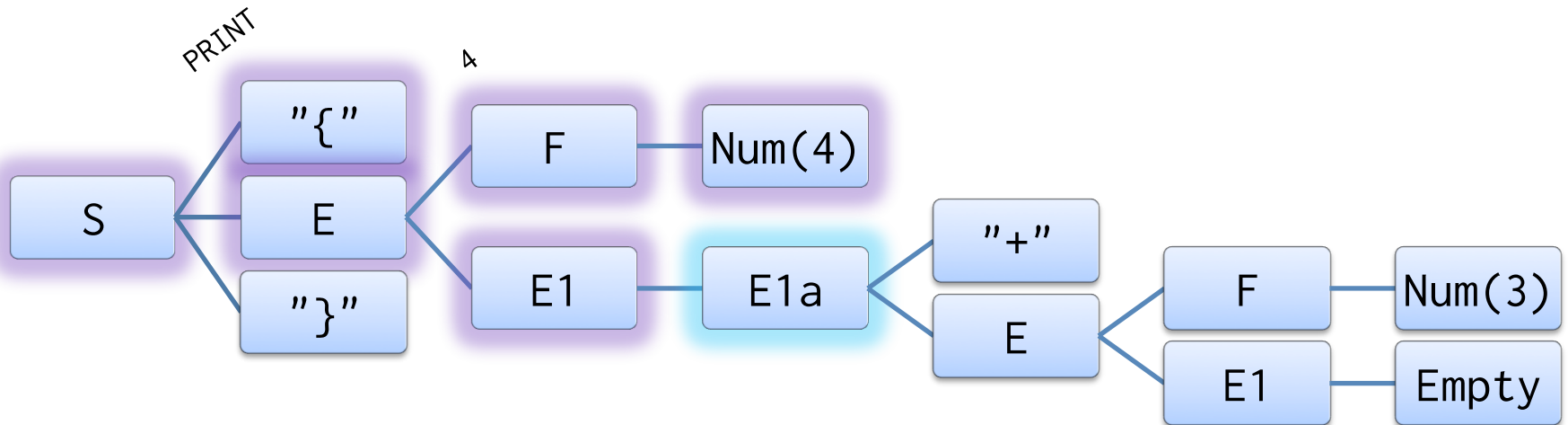
Parse Tree



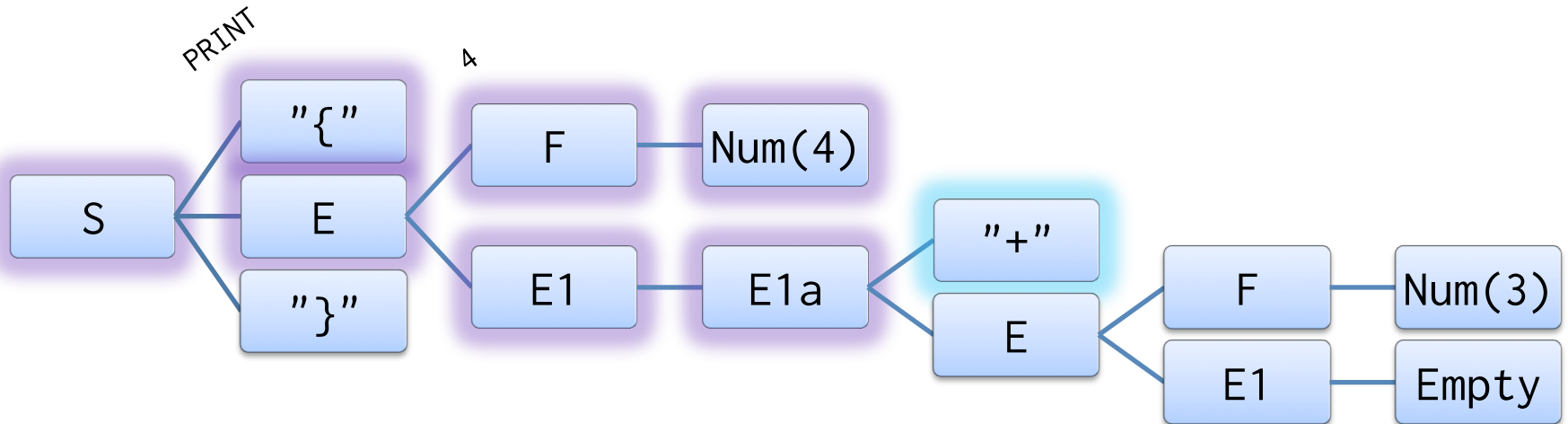
Parse Tree



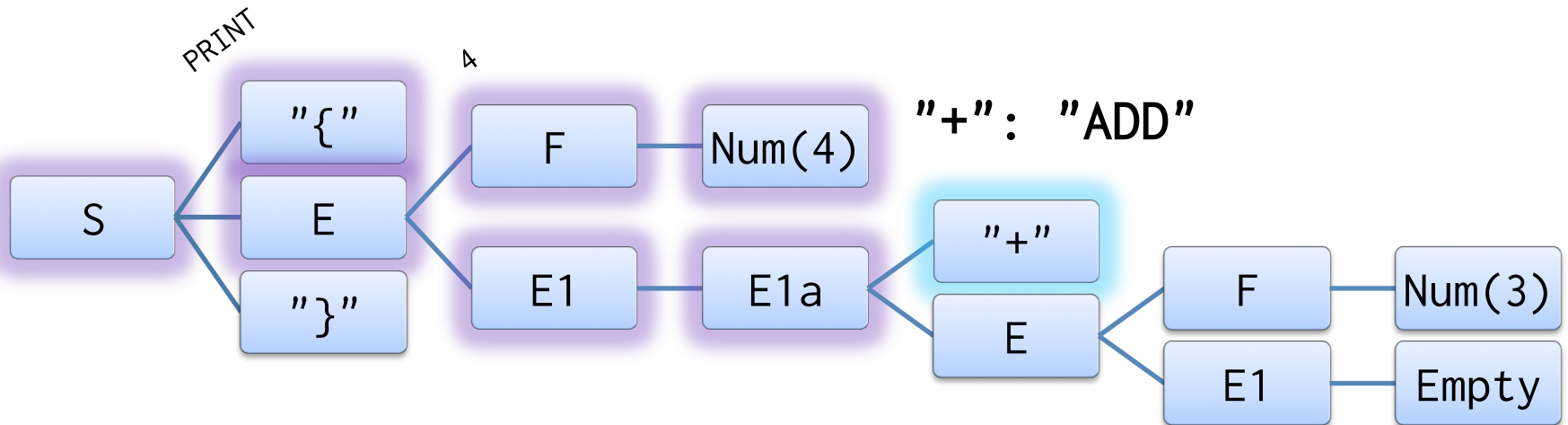
Parse Tree



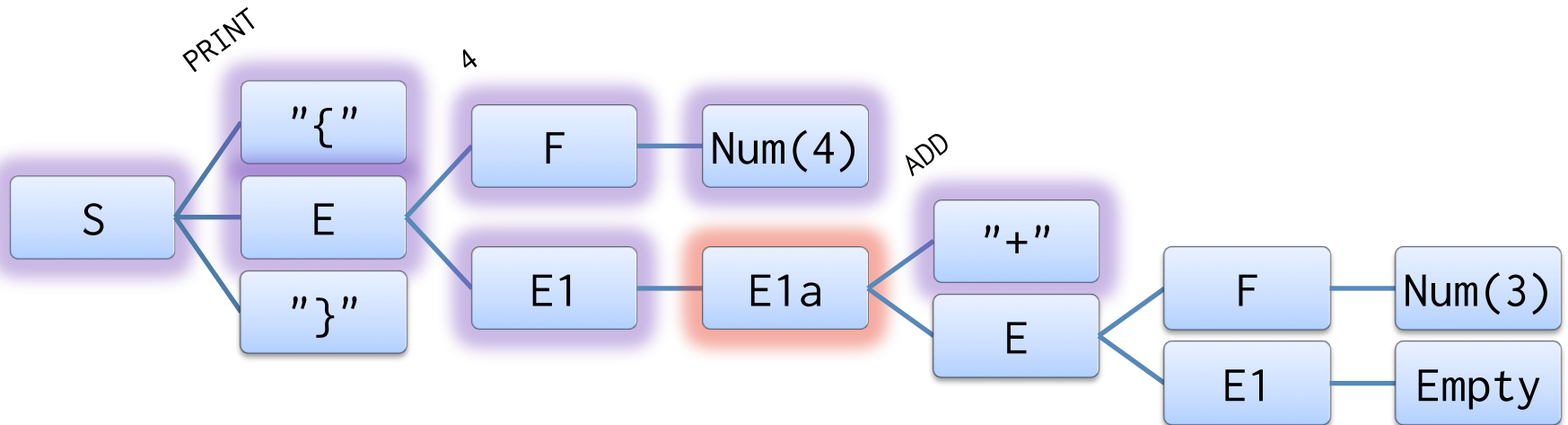
Parse Tree



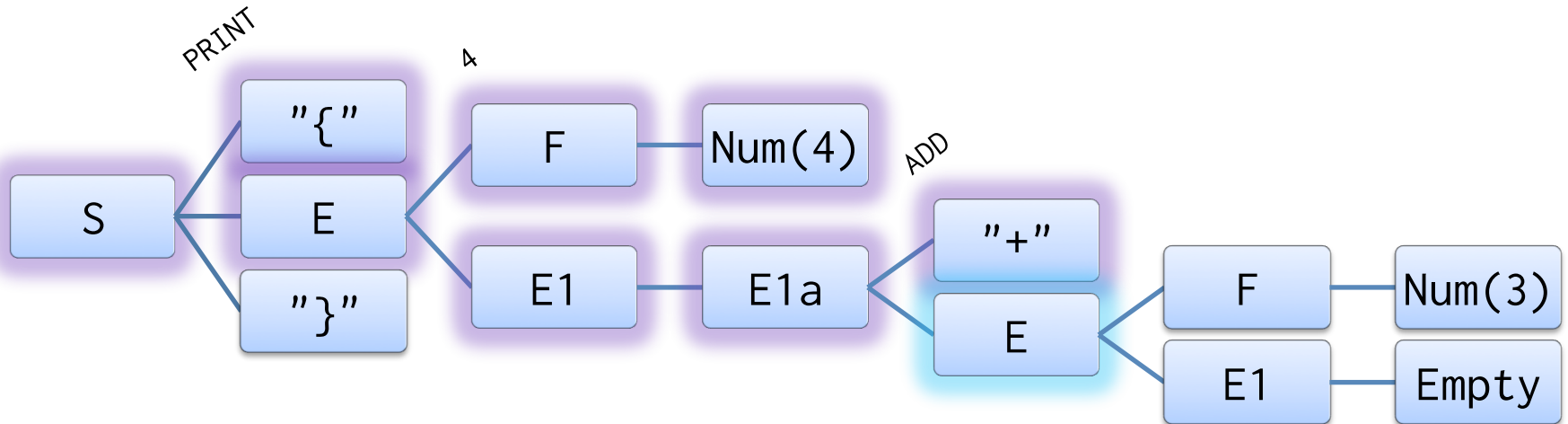
Parse Tree



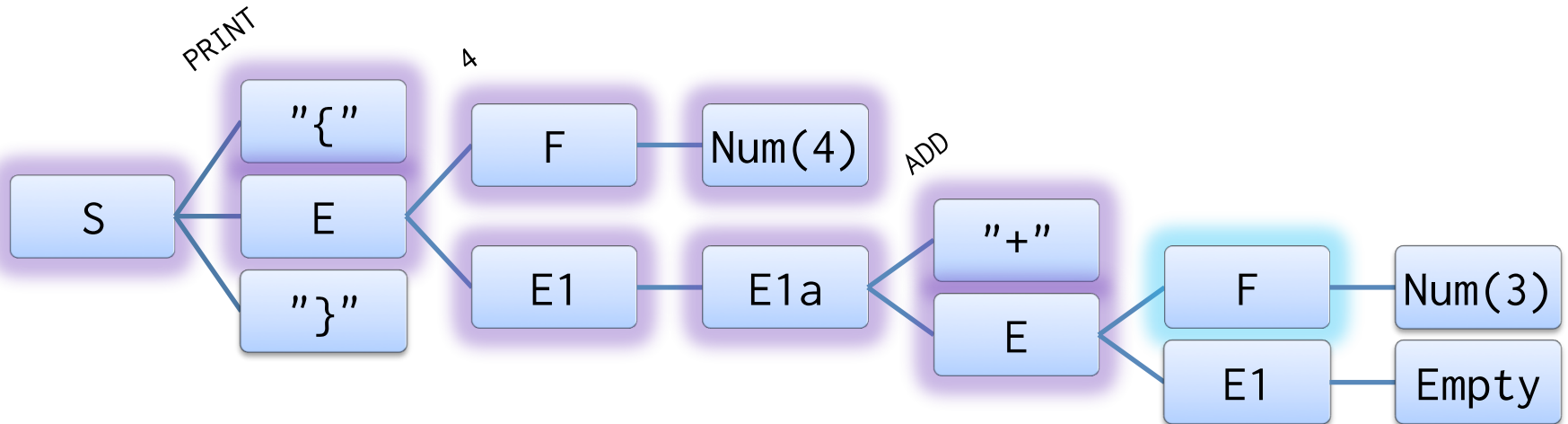
Parse Tree



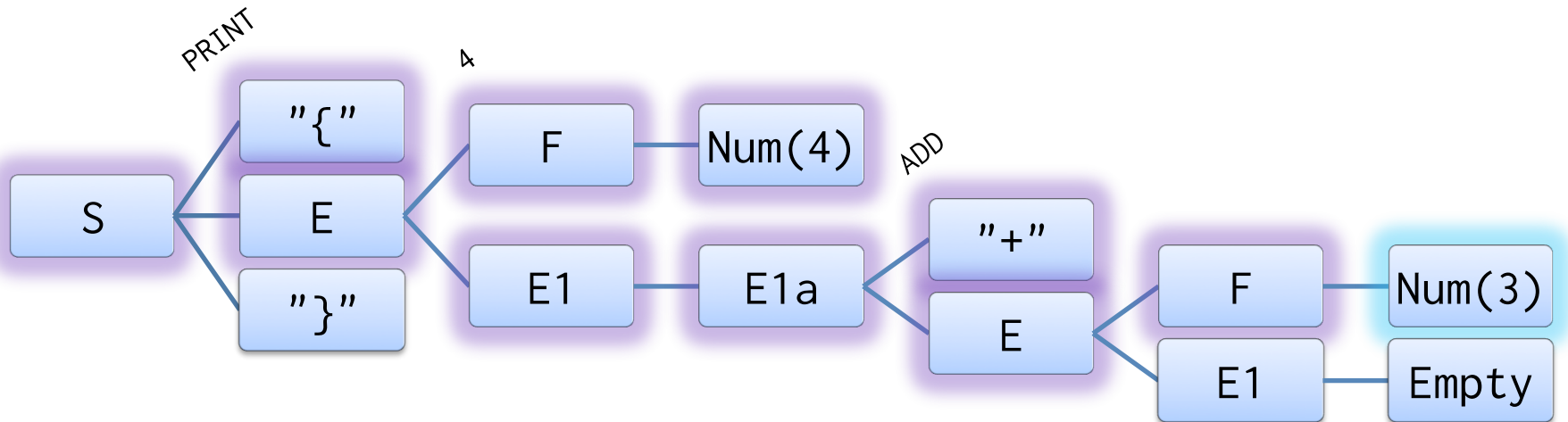
Parse Tree



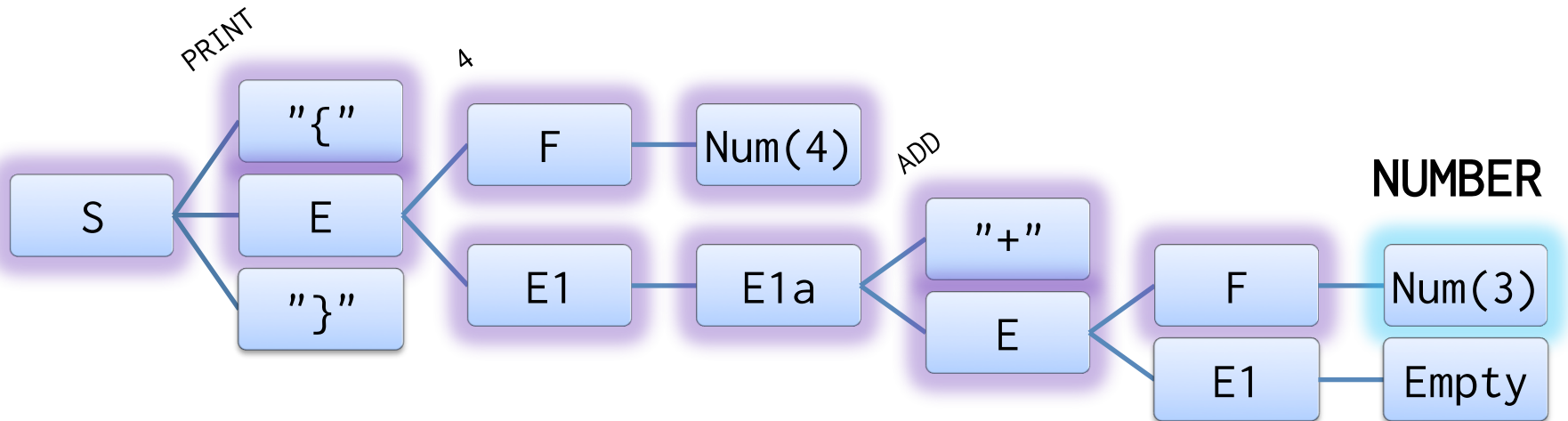
Parse Tree



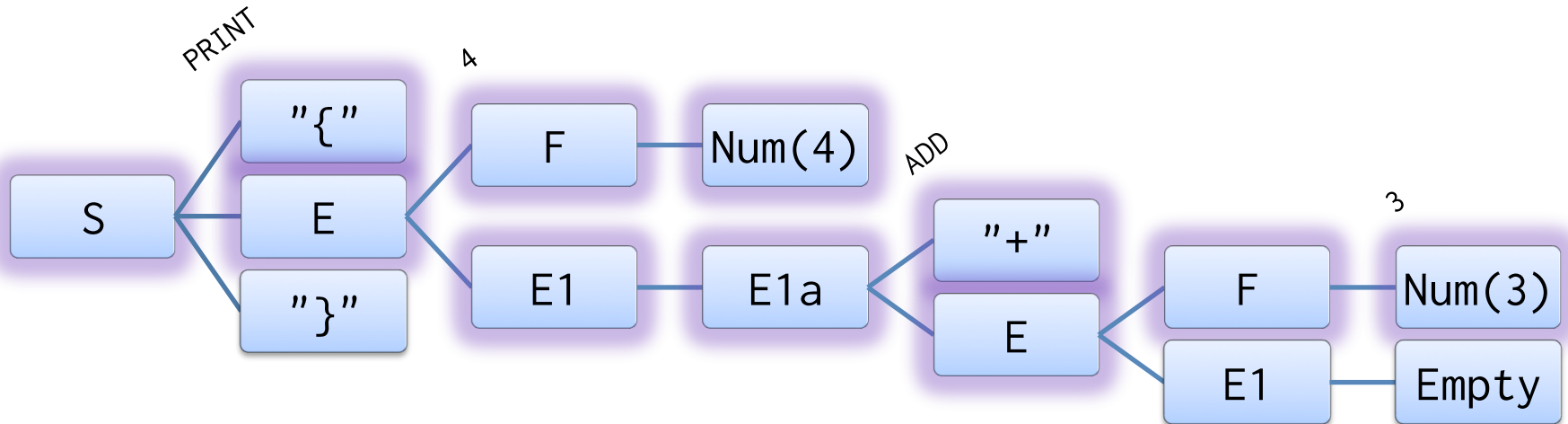
Parse Tree



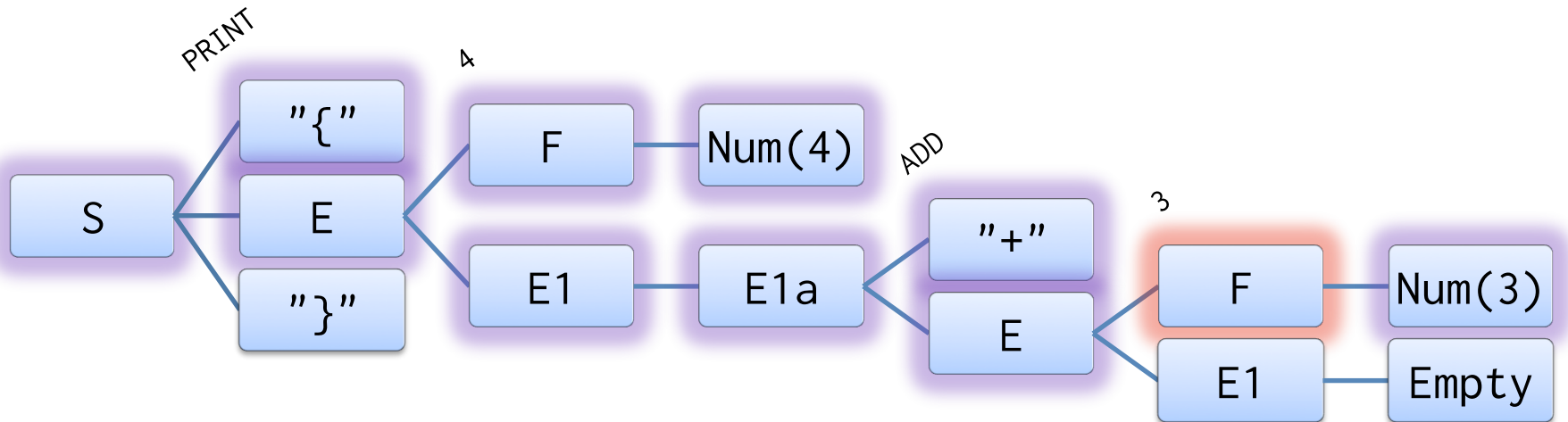
Parse Tree



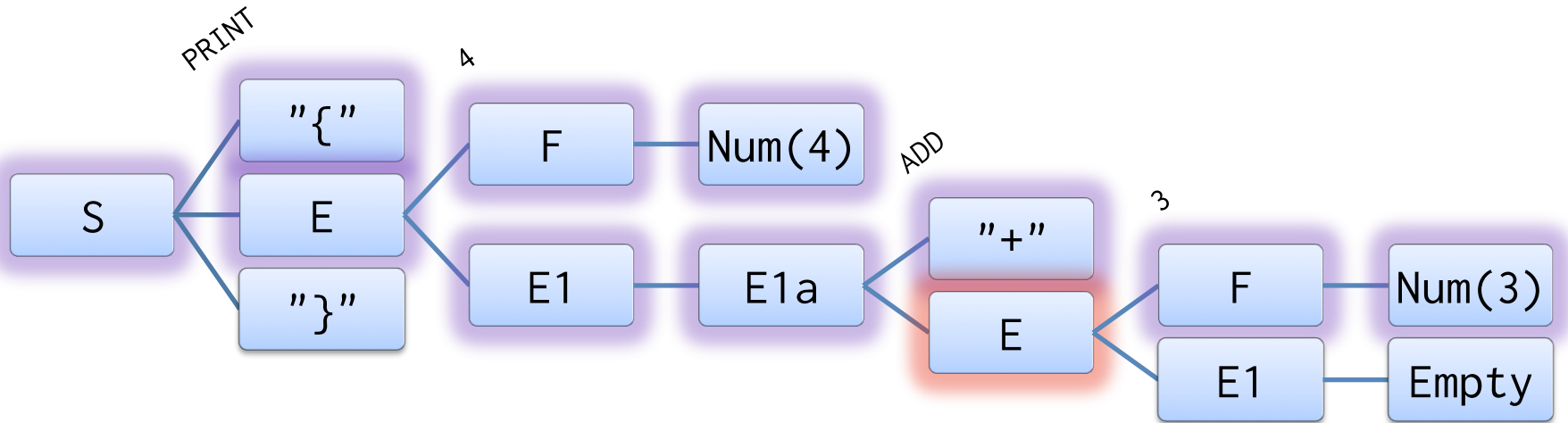
Parse Tree



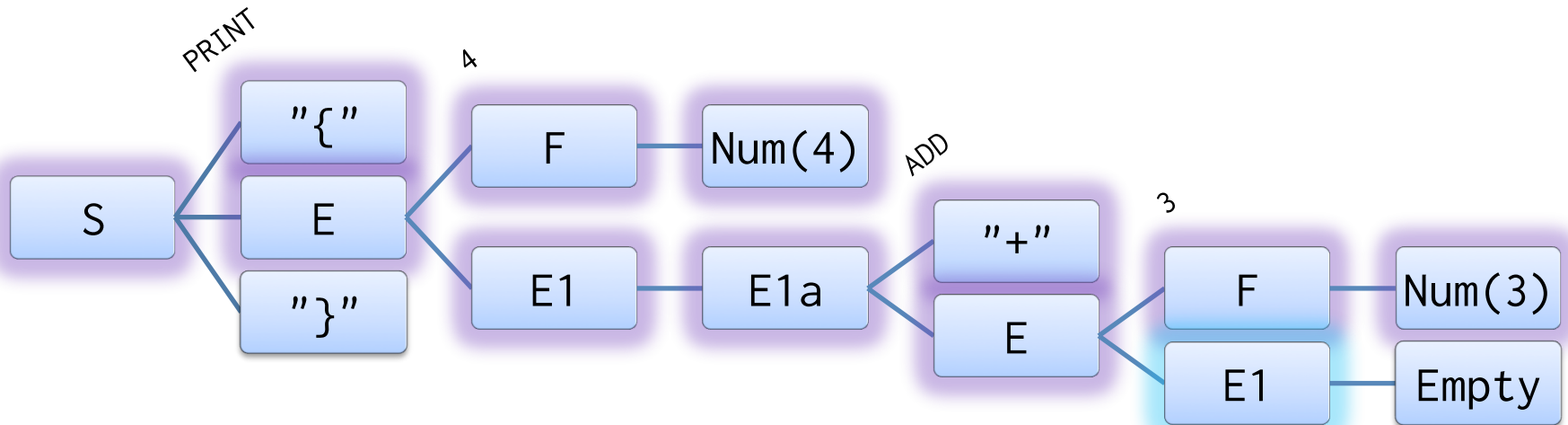
Parse Tree



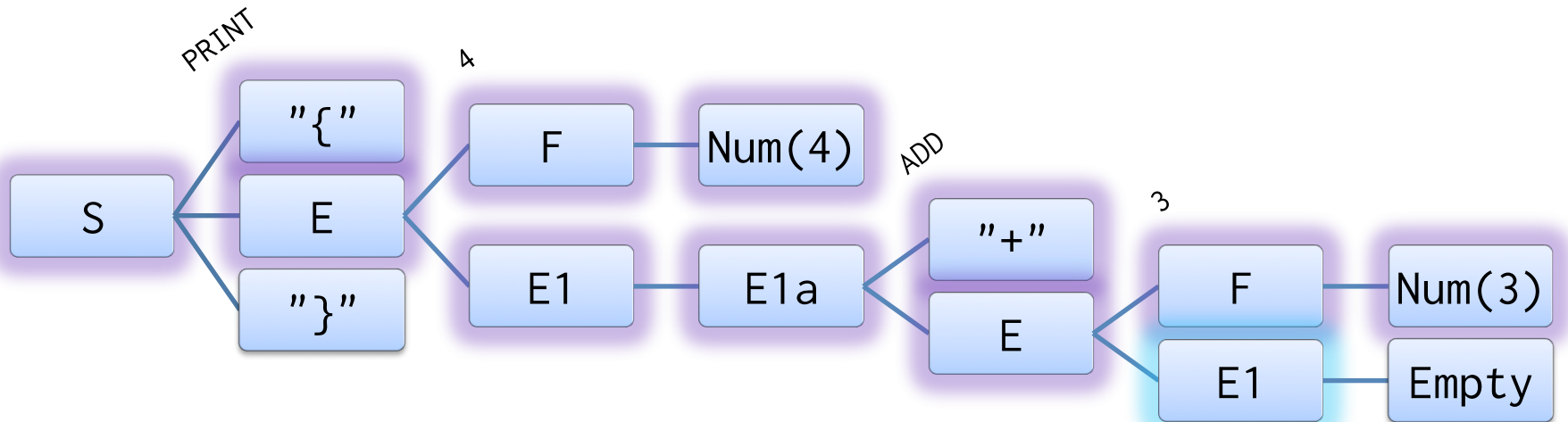
Parse Tree



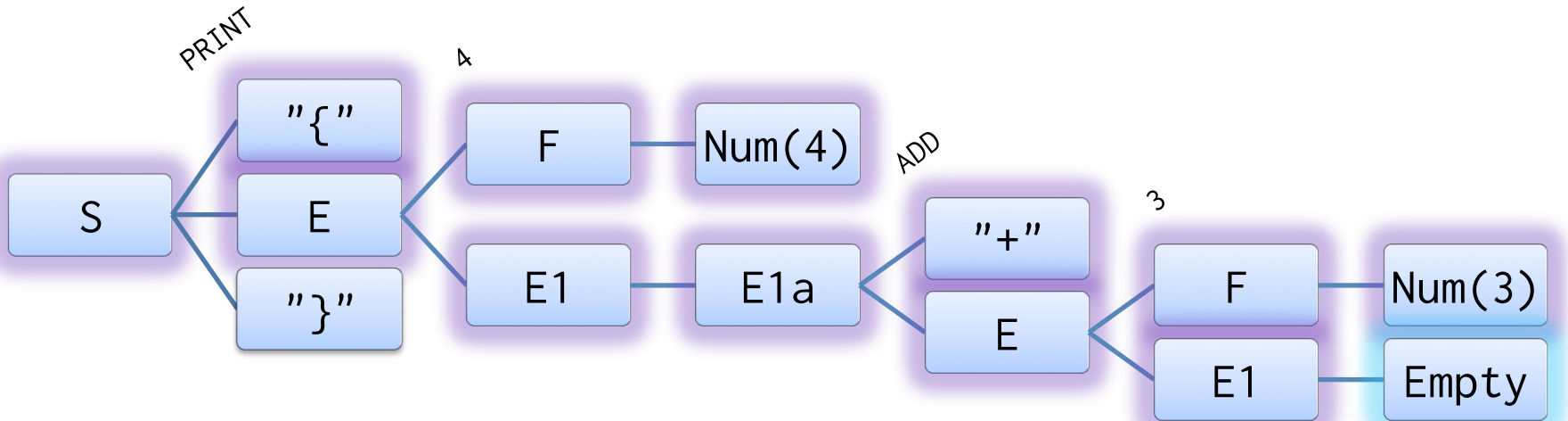
Parse Tree



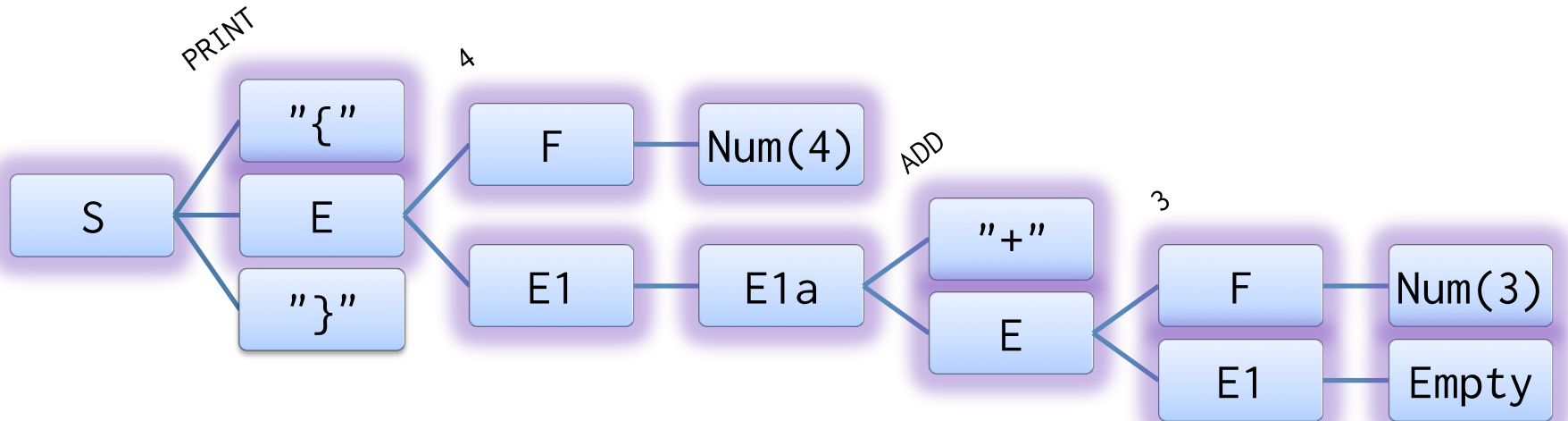
Parse Tree



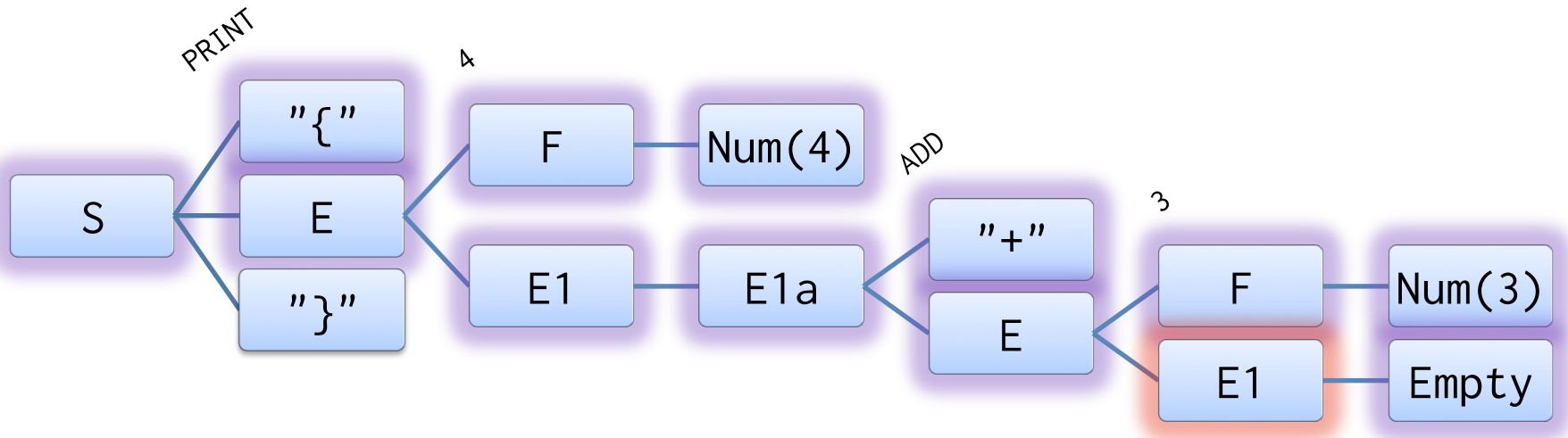
Parse Tree



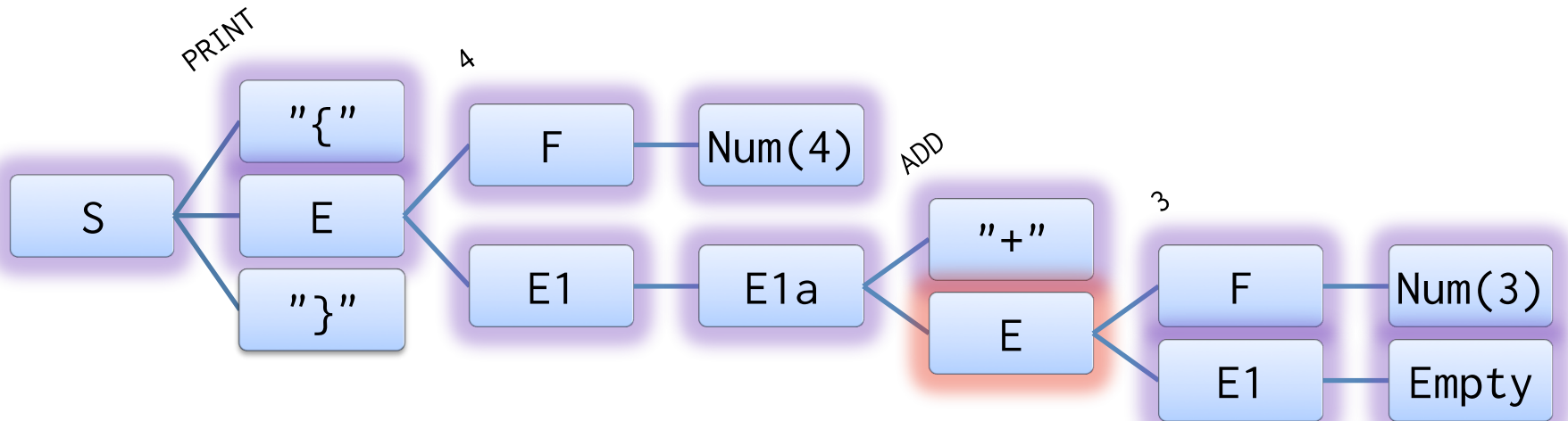
Parse Tree



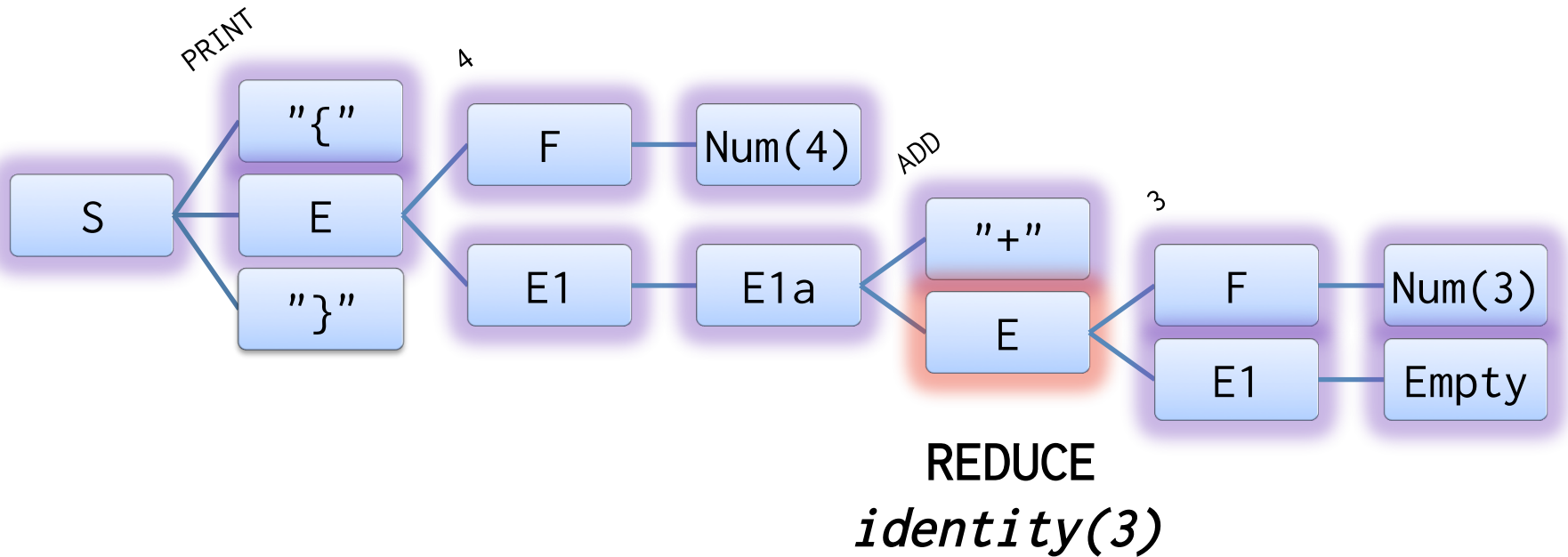
Parse Tree



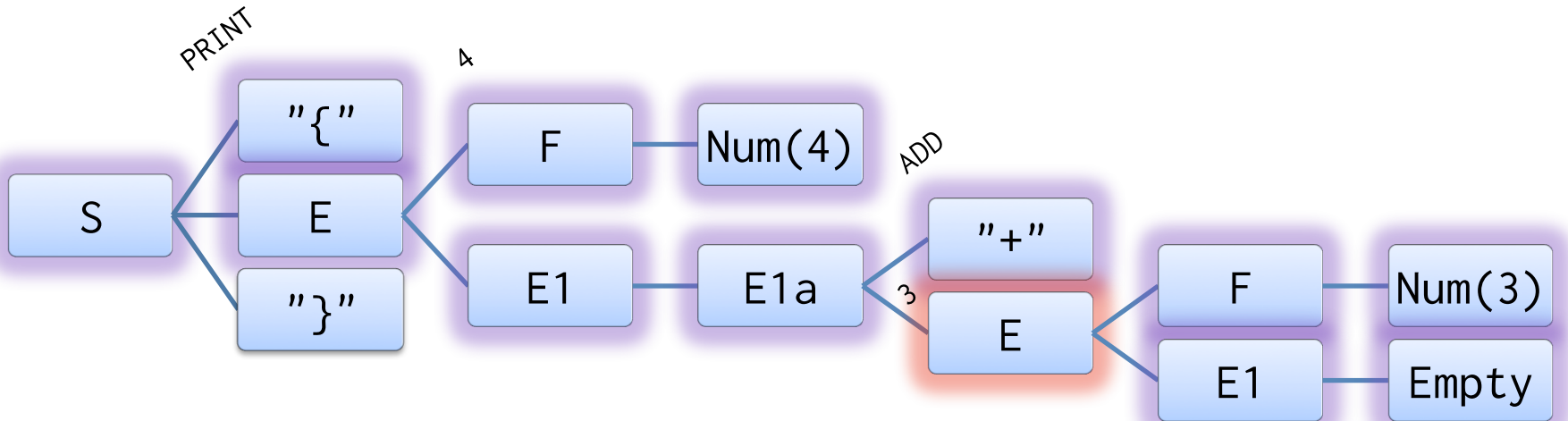
Parse Tree



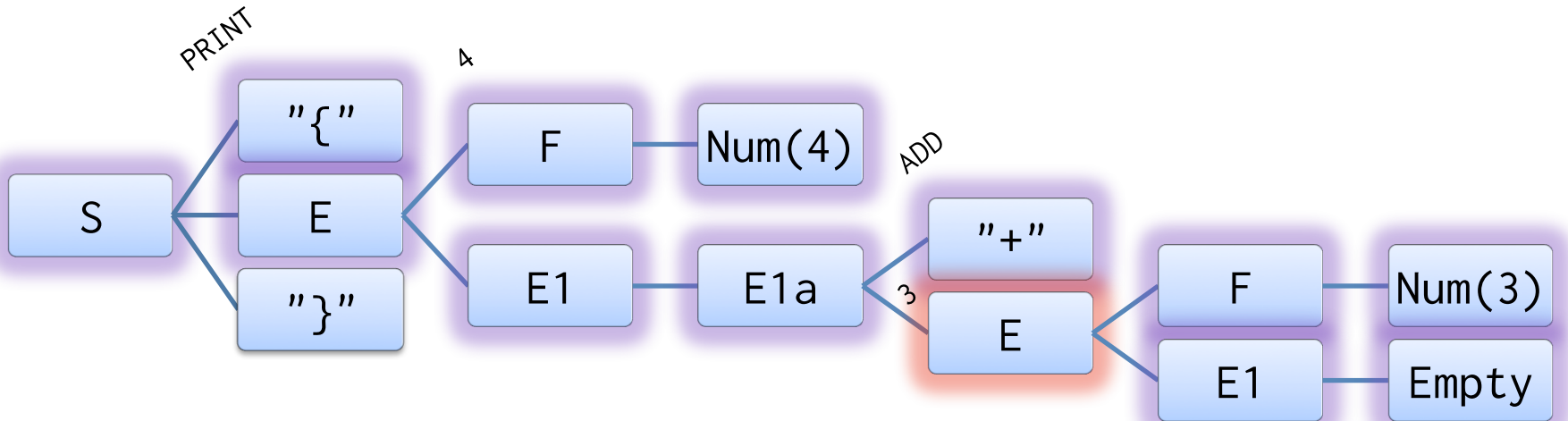
Parse Tree



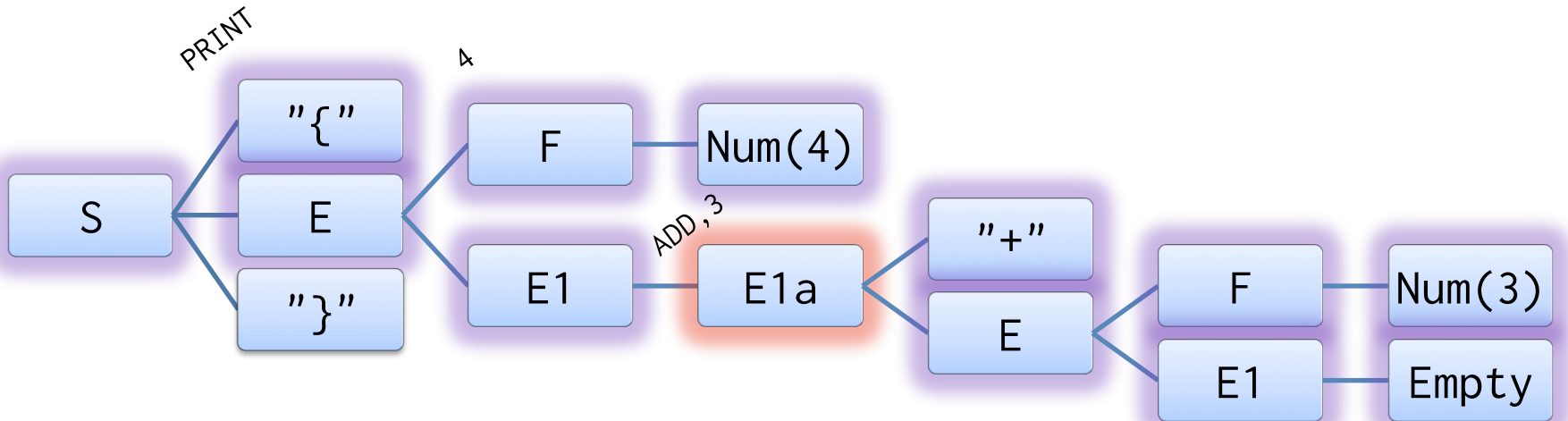
Parse Tree



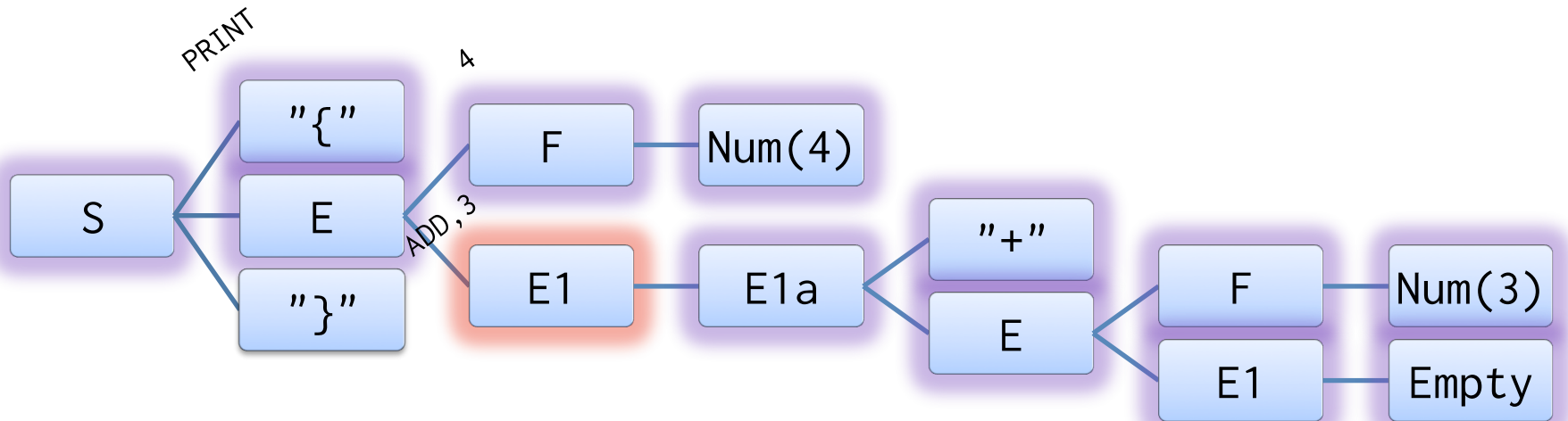
Parse Tree



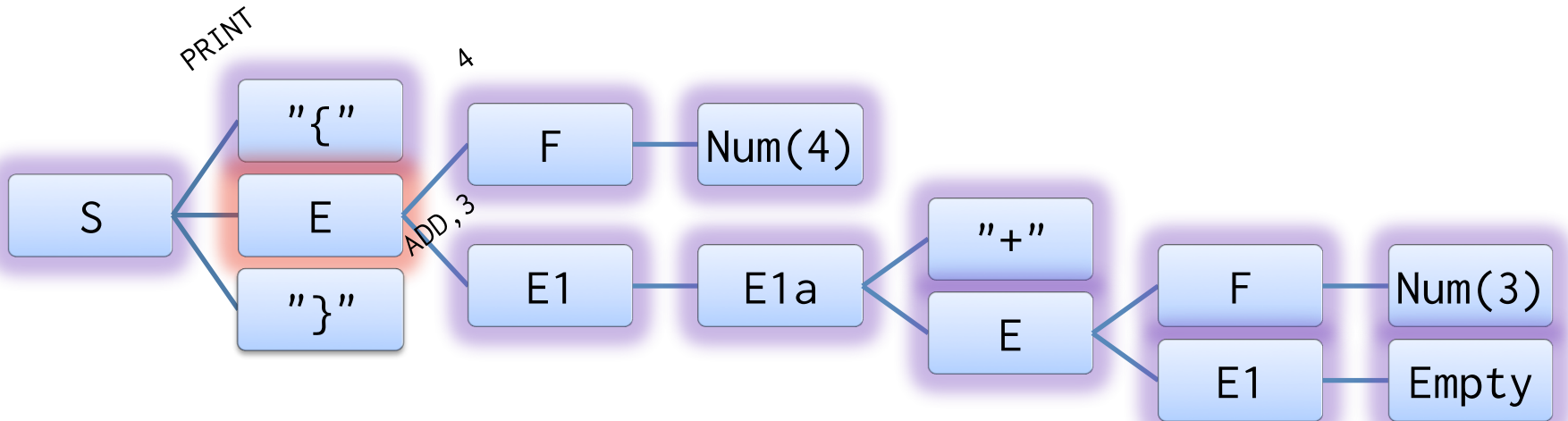
Parse Tree



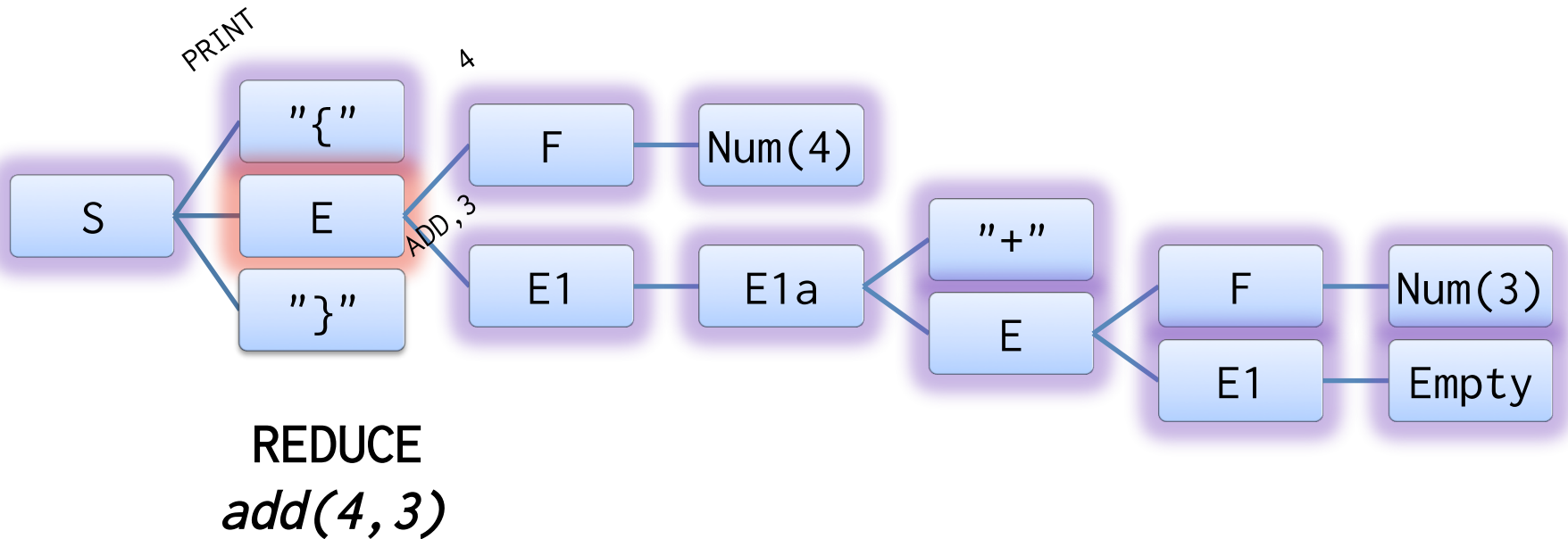
Parse Tree



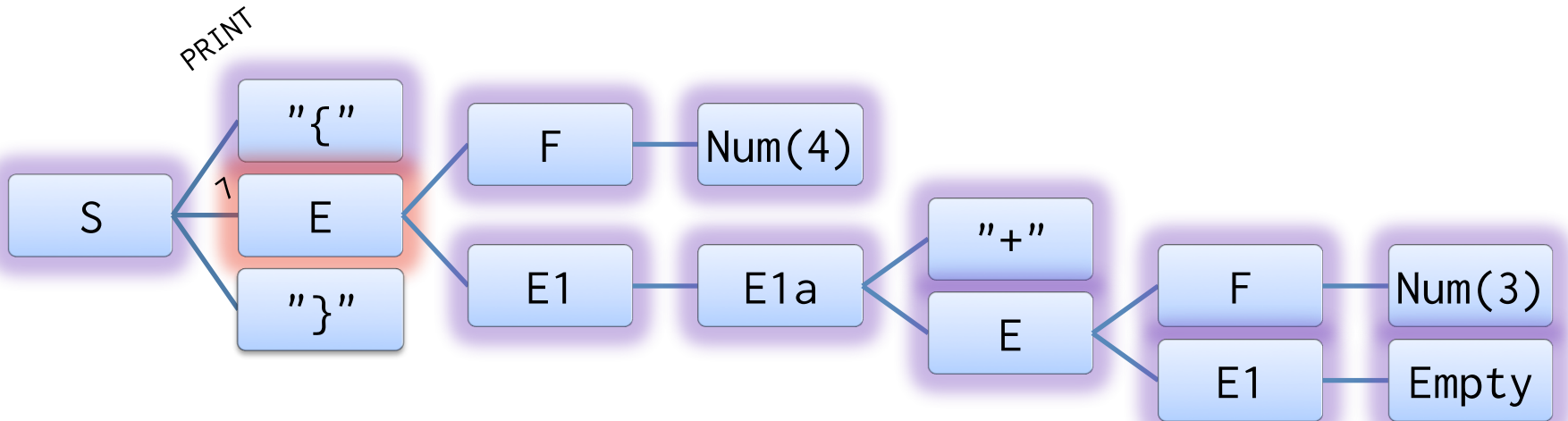
Parse Tree



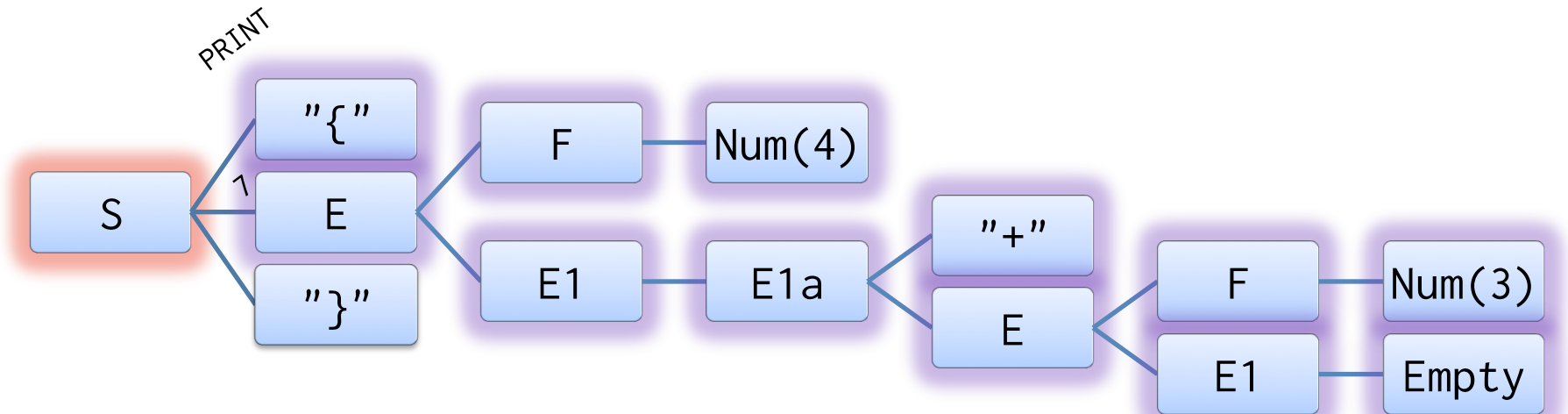
Parse Tree



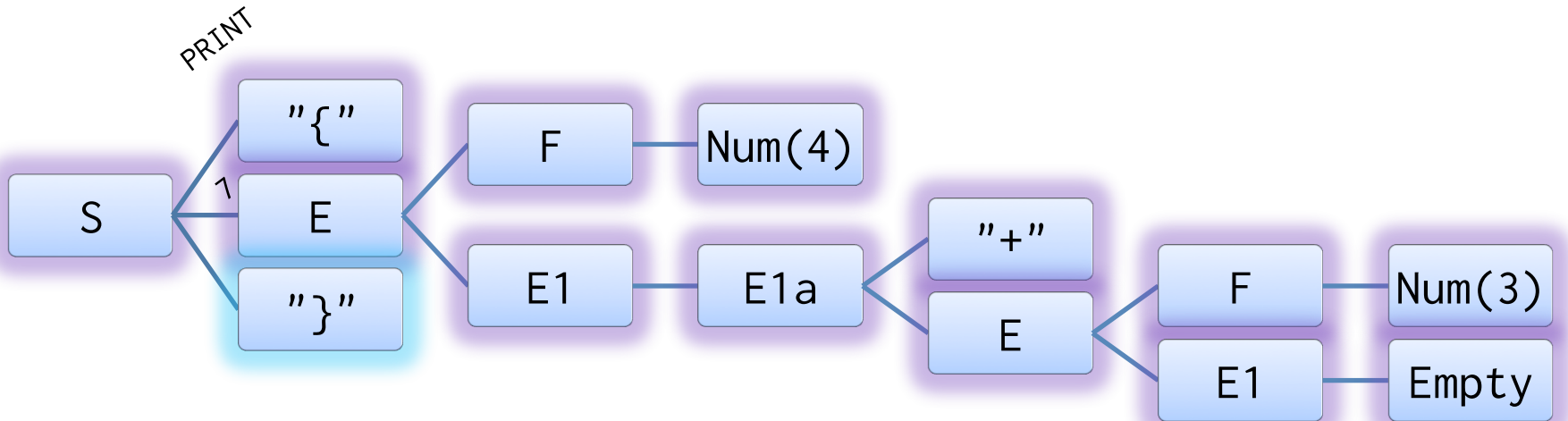
Parse Tree



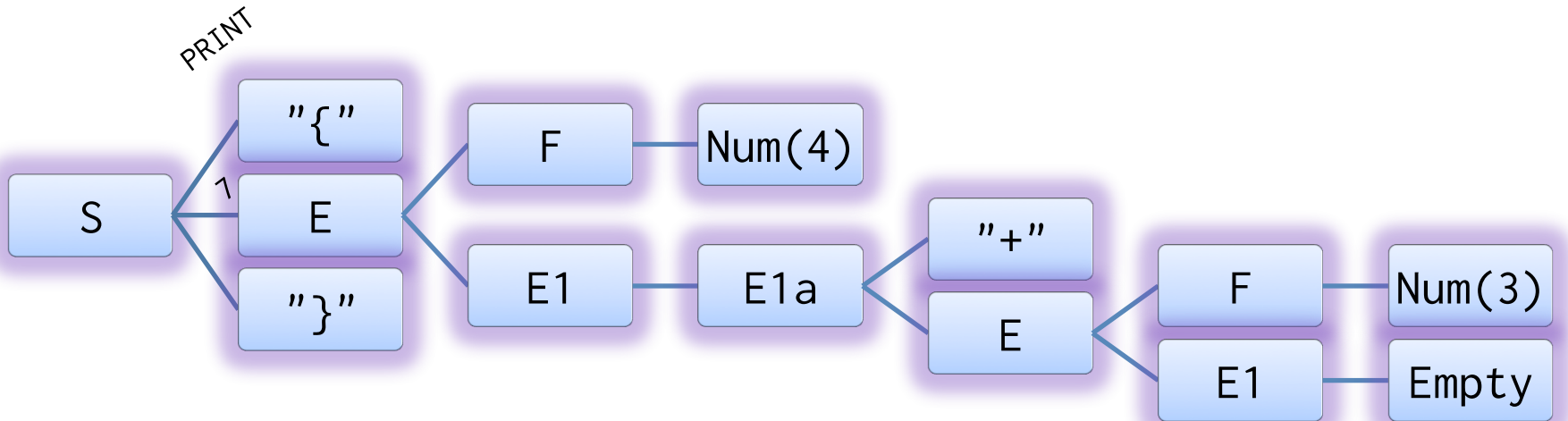
Parse Tree



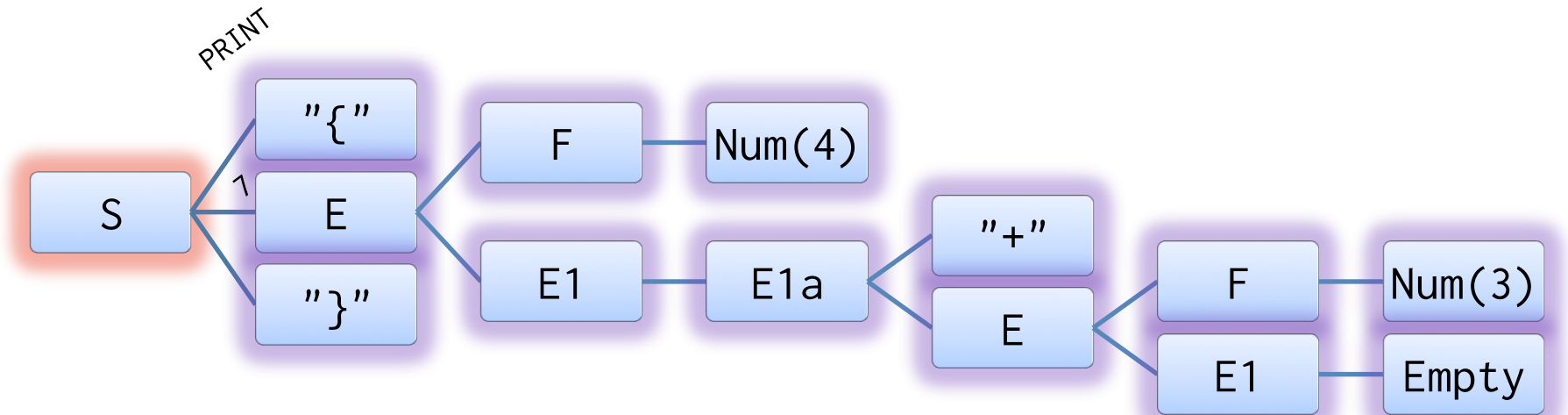
Parse Tree



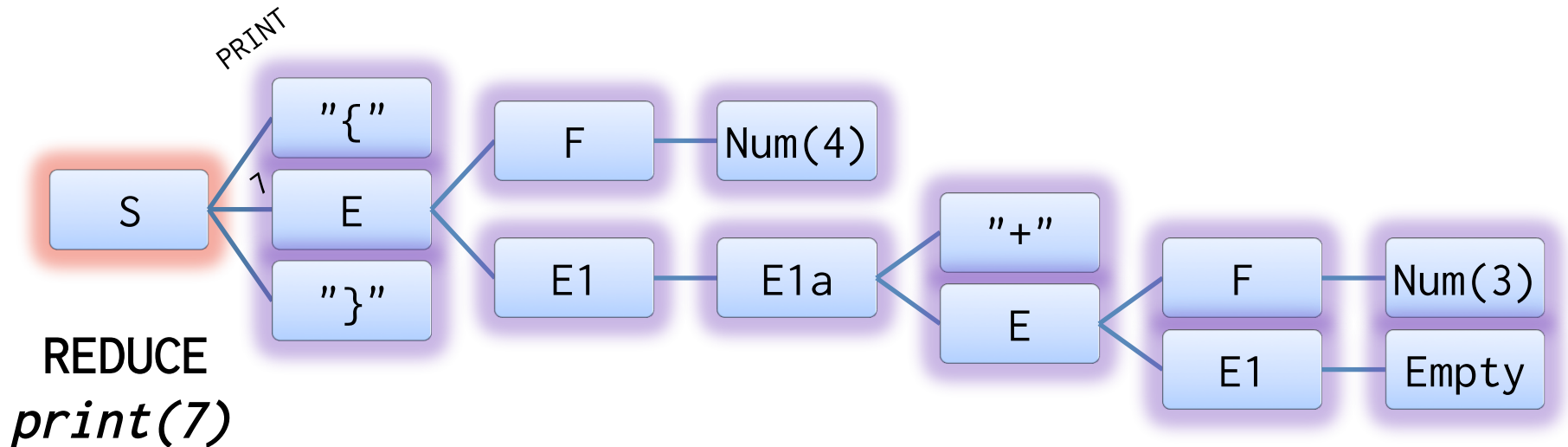
Parse Tree



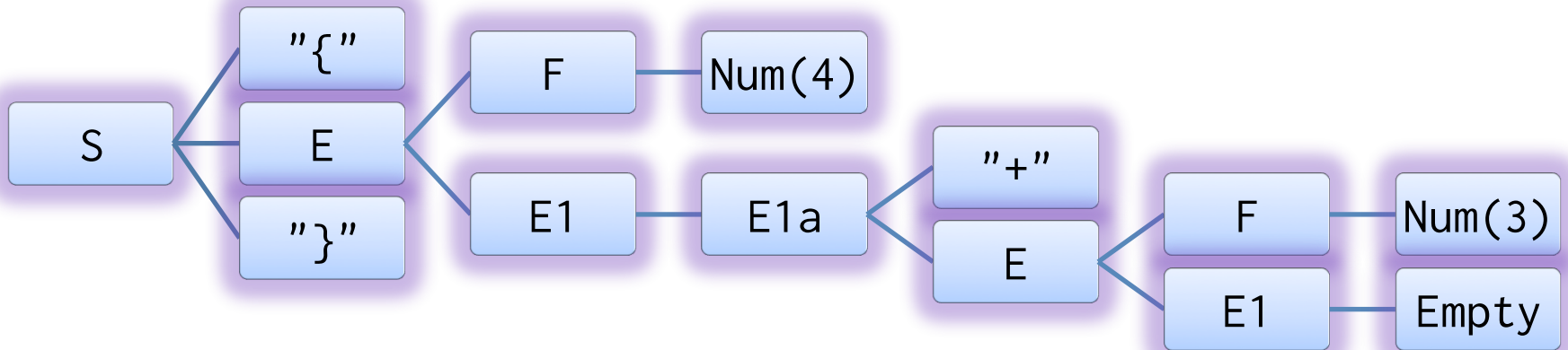
Parse Tree



Parse Tree



Parse Tree



Output: 7

Notes

- The answer you return should be the *output* of the program, i.e. the results of all the *print* commands.
- Reductions are applied using the operator found in the argument list. If no operator is found, use a default *identity* reduction (see slide 34).
- You need to figure out how to express default behaviors for terminals and nonterminals lacking specified semantics.