

(a) Direct left recursion

Grammar:
 $A \leftarrow (A \text{ 'x'}) / \text{'x'};$

Parse tree for input: xxx

```

└─ A : 0+3 : "xxx"
  └─ A 'x' : 0+3 : "xxx"
    └─ A : 0+2 : "xx"
      └─ A 'x' : 0+2 : "xx"
        └─ A : 0+1 : "x"
          └─ A 'x' : 0+1 : "x"
            └─ 'x' : 1+1 : "x"
              └─ 'x' : 2+1 : "x"
                └─ 'x' : 2+1 : "x"
  
```

(b) Indirect left recursion

Grammar:
 $A \leftarrow B / \text{'x'};$
 $B \leftarrow (A \text{ 'y'}) / (A \text{ 'x'})$

Parse tree for input: xxxyx

```

└─ A : 0+4 : "xxxyx"
  └─ B : 0+4 : "xxxyx"
    └─ A 'x' : 0+4 : "xxxyx"
      └─ A : 0+3 : "xxy"
        └─ B : 0+3 : "xxy"
          └─ A 'y' : 0+3 : "xxy"
            └─ A : 0+2 : "xx"
              └─ B : 0+2 : "xx"
                └─ A 'x' : 0+2 : "xx"
                  └─ A : 0+1 : "x"
                    └─ A 'x' : 0+1 : "x"
                      └─ 'x' : 1+1 : "x"
                        └─ 'x' : 2+1 : "x"
                          └─ 'y' : 2+1 : "y"
                            └─ 'x' : 3+1 : "x"
  
```

(c) Input-dependent left recursion

Grammar:
 $A \leftarrow B / \text{'z'};$
 $B \leftarrow (\text{'x'} A) / (A \text{ 'y'})$

Parse tree for input: xxzyyy

```

└─ A : 0+6 : "xxzyyy"
  └─ B : 0+6 : "xxzyyy"
    └─ 'x' A : 0+6 : "xxzyyy"
      └─ 'x' : 0+1 : "x"
        └─ A : 1+5 : "xzyyy"
          └─ B : 1+5 : "xzyyy"
            └─ 'x' A : 1+5 : "xzyyy"
              └─ 'x' : 1+1 : "x"
                └─ A : 2+4 : "zyyy"
                  └─ B : 2+4 : "zyyy"
                    └─ A 'y' : 2+4 : "zyyy"
                      └─ A : 2+3 : "zy"
                        └─ B : 2+3 : "zy"
                          └─ A 'y' : 2+3 : "zy"
                            └─ A : 2+2 : "zy"
                              └─ B : 2+2 : "zy"
                                └─ A 'y' : 2+2 : "zy"
                                  └─ A : 2+1 : "z"
                                    └─ 'z' : 2+1 : "z"
                                      └─ 'y' : 3+1 : "y"
                                        └─ 'y' : 4+1 : "y"
                                          └─ 'y' : 5+1 : "y"
  
```

(d) Input-dependent left recursion

Grammar:
 $A \leftarrow \text{'x'?} (A \text{ 'y'} / A / \text{'y'})$

Parse tree for input: xxyyy

```

└─ A : 0+5 : "xxyyy"
  └─ 'x'? : 0+1 : "x"
    └─ 'x' : 0+1 : "x"
      └─ (A 'y') / A / 'y' : 1+4 : "xyyy"
        └─ A : 1+4 : "xyyy"
          └─ 'x'? : 1+1 : "x"
            └─ 'x' : 1+1 : "x"
              └─ (A 'y') / A / 'y' : 2+3 : "yyy"
                └─ A : 2+3 : "yyy"
                  └─ 'x'? : 2+0 : ""
                    └─ (A 'y') / A / 'y' : 2+3 : "yyy"
                      └─ A 'y' : 2+3 : "yyy"
                        └─ A : 2+2 : "yy"
                          └─ 'x'? : 2+0 : ""
                            └─ (A 'y') / A / 'y' : 2+2 : "yy"
                              └─ A 'y' : 2+2 : "yy"
                                └─ A : 2+1 : "y"
                                  └─ 'x'? : 2+0 : ""
                                    └─ (A 'y') / A / 'y' : 2+1 : "y"
                                      └─ 'y' : 2+1 : "y"
                                        └─ 'y' : 3+1 : "y"
                                          └─ 'y' : 4+1 : "y"
  
```

(e) Interwoven left recursion (3 cycles)

Grammar:
 $S \leftarrow E;$
 $E \leftarrow F \text{ 'n'} / \text{'n'};$
 $F \leftarrow E \text{ '+' } I^* / G \text{ '-' };$
 $G \leftarrow H \text{ 'm'} / E;$
 $H \leftarrow G \text{ 'l'};$
 $I \leftarrow (' A+ ')';$
 $A \leftarrow \text{'a'}$

Parse tree for input: nlm-n+(aaa)n

```

└─ E : 0+12 : "nlm-n+(aaa)n"
  └─ F 'n' : 0+12 : "nlm-n+(aaa)n"
    └─ F : 0+11 : "nlm-n+(aaa)"
      └─ E '+' I* : 0+11 : "nlm-n+(aaa)"
        └─ E : 0+5 : "nlm-n"
          └─ F 'n' : 0+5 : "nlm-n"
            └─ F : 0+4 : "nlm-"
              └─ G '-' : 0+4 : "nlm-"
                └─ G : 0+3 : "nlm"
                  └─ H 'm' : 0+3 : "nlm"
                    └─ H : 0+2 : "nl"
                      └─ G : 0+1 : "n"
                        └─ E : 0+1 : "n"
                          └─ 'n' : 0+1 : "n"
                            └─ 'l' : 1+1 : "l"
                              └─ 'm' : 2+1 : "m"
                                └─ '-': 3+1 : "-"
                                  └─ 'n' : 4+1 : "n"
                                    └─ '+' : 5+1 : "+"
                                      └─ I* : 6+5 : "(aaa)"
                                        └─ I : 6+5 : "(aaa)"
                                          └─ '(' : 6+1 : "("
                                            └─ A+ : 7+3 : "aaa"
                                              └─ A : 7+1 : "a"
                                                └─ A : 8+1 : "a"
                                                └─ A : 9+1 : "a"
                                              └─ ')' : 10+1 : ")"
                                                └─ 'n' : 11+1 : "n"
  
```

(f) Interwoven left recursion (2 cycles)

Grammar:
 $M \leftarrow L;$
 $L \leftarrow P \text{ ".x" / 'x' };$
 $P \leftarrow P \text{ "(n)" / L; }$

Parse tree for input: x.x(n)(n).x.x

```

└─ L : 0+13 : "x.x(n)(n).x.x"
  └─ P ".x" : 0+13 : "x.x(n)(n).x.x"
    └─ P : 0+11 : "x.x(n)(n).x"
      └─ L : 0+11 : "x.x(n)(n).x"
        └─ P ".x" : 0+11 : "x.x(n)(n).x"
          └─ P "(n)" : 0+9 : "x.x(n)(n)"
            └─ P : 0+6 : "x.x(n)"
              └─ P "(n)" : 0+6 : "x.x(n)"
                └─ P : 0+3 : "x.x"
                  └─ L : 0+3 : "x.x"
                    └─ P ".x" : 0+3 : "x.x"
                      └─ P : 0+1 : "x"
                        └─ L : 0+1 : "x"
                          └─ P ".x" : 0+1 : "x"
                            └─ "(n)" : 3+3 : "(n)"
                            └─ ".x" : 6+3 : ".x"
                            └─ ".x" : 9+2 : ".x"
                            └─ ".x" : 11+2 : ".x"
  
```

(g) Explicit left associativity

Grammar:
 $E \leftarrow E \text{ '+' } N / N;$
 $N \leftarrow [0-9]+;$

Parse tree for input: 0+1+2+3

```

└─ E : 0+7 : "0+1+2+3"
  └─ E '+' N : 0+7 : "0+1+2+3"
    └─ E : 0+5 : "0+1+2"
      └─ E '+' N : 0+5 : "0+1+2"
        └─ E : 0+3 : "0+1"
          └─ E '+' N : 0+3 : "0+1"
            └─ E : 0+1 : "0"
              └─ N : 0+1 : "0"
                └─ [0-9] : 0+1 : "0"
                └─ '+' : 1+1 : "+"
                └─ E : 2+5 : "1+2+3"
                  └─ N '+' E : 2+5 : "1+2+3"
                    └─ N : 2+1 : "1"
                      └─ [0-9] : 2+1 : "1"
                      └─ '+' : 3+1 : "+"
                      └─ E : 4+3 : "2+3"
                        └─ N '+' E : 4+3 : "2+3"
                          └─ N : 4+1 : "2"
                            └─ [0-9] : 4+1 : "2"
                            └─ '+' : 5+1 : "+"
                            └─ E : 6+1 : "3"
                              └─ N : 6+1 : "3"
                                └─ [0-9] : 6+1 : "3"
  
```

(h) Explicit right associativity

Grammar:
 $E \leftarrow N \text{ '+' } E / N;$
 $N \leftarrow [0-9]+;$

Parse tree for input: 0+1+2+3

```

└─ E : 0+7 : "0+1+2+3"
  └─ N '+' E : 0+7 : "0+1+2+3"
    └─ N : 0+1 : "0"
      └─ [0-9] : 0+1 : "0"
      └─ '+' : 1+1 : "+"
      └─ E : 2+5 : "1+2+3"
        └─ N '+' E : 2+5 : "1+2+3"
          └─ N : 2+1 : "1"
            └─ [0-9] : 2+1 : "1"
            └─ '+' : 3+1 : "+"
            └─ E : 4+3 : "2+3"
              └─ N '+' E : 4+3 : "2+3"
                └─ N : 4+1 : "2"
                  └─ [0-9] : 4+1 : "2"
                  └─ '+' : 5+1 : "+"
                  └─ E : 6+1 : "3"
                    └─ N : 6+1 : "3"
                      └─ [0-9] : 6+1 : "3"
  
```

(i) Ambiguous associativity

Grammar:
 $E \leftarrow E \text{ '+' } E / N;$
 $N \leftarrow [0-9]+;$

Parse tree for input: 0+1+2+3

```

└─ E : 0+7 : "0+1+2+3"
  └─ E '+' E : 0+7 : "0+1+2+3"
    └─ E : 0+1 : "0"
      └─ N : 0+1 : "0"
        └─ [0-9] : 0+1 : "0"
        └─ '+' : 1+1 : "+"
        └─ E : 2+5 : "1+2+3"
          └─ E '+' E : 2+5 : "1+2+3"
            └─ E : 2+1 : "1"
              └─ N : 2+1 : "1"
                └─ [0-9] : 2+1 : "1"
                └─ '+' : 3+1 : "+"
                └─ E : 4+3 : "2+3"
                  └─ E '+' E : 4+3 : "2+3"
                    └─ E : 4+1 : "2"
                      └─ N : 4+1 : "2"
                        └─ [0-9] : 4+1 : "2"
                        └─ '+' : 5+1 : "+"
                        └─ E : 6+1 : "3"
                          └─ N : 6+1 : "3"
                            └─ [0-9] : 6+1 : "3"
  
```