Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7

(parse tree here)

(activation stack here)

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

```
Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i
```

▲25 + 13 * 5 + 7

Angelic recursive descent parser:

```
▲Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

▲25 + 13 * 5 + 7

Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) { ▲Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) { Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

▲25 + 13 * 5 + 7

Expr( )

Angelic recursive descent parser:

```
▲Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

▲25 + 13 * 5 + 7

Expr( )
Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
                ▲
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

▲25 + 13 * 5 + 7

Expr( )
Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


▲Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

▲25 + 13 * 5 + 7

Expr( )
Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { ▲match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
T ▲

Expr( )
Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}
```
▲

```
Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
  T ▲
  E

Expr( )
Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); } ▲
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
T ▲
E

Expr( )
Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
  T   ▲
  E

Expr( )
Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
T ▲
E

Expr( )
Expr( )
Term( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
T ▲
E

Expr( )
Expr( )
Term( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); } ▲
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
  T   T ▲
     E

Expr( )
Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); } ▲
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
  T   T ▲
  E

Expr( )
Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
 T   T   ▲
    E

Expr( )
Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int);}
    else { match(int); }
}
```
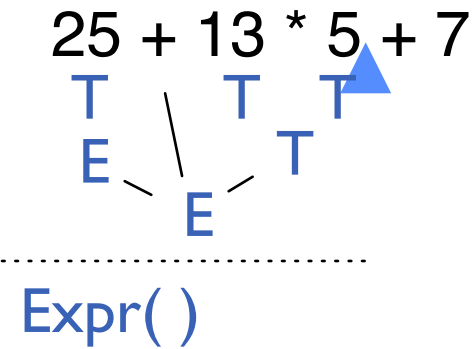▲

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
T   T   T
E   T

Expr( )
Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term();}
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```
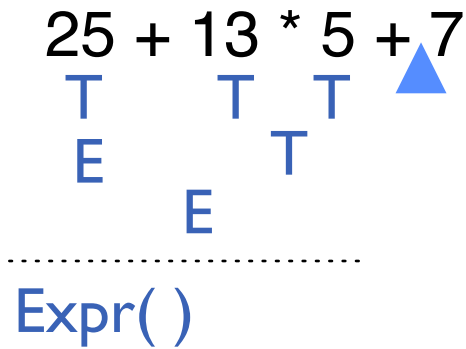
Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
T   T   T
E   T
   E

Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```
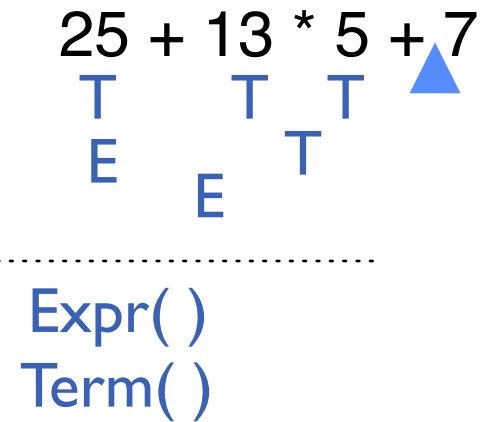
Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7
 T   T   T   ▲
 E   T
         E

Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}
```
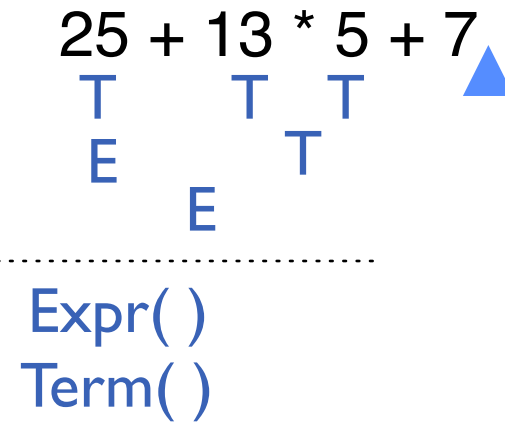▲
```
Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

```
25 + 13 * 5 + 7
 T       T   T    ▲
 E          T
     E
```

Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { ▲match(int); }
}
```
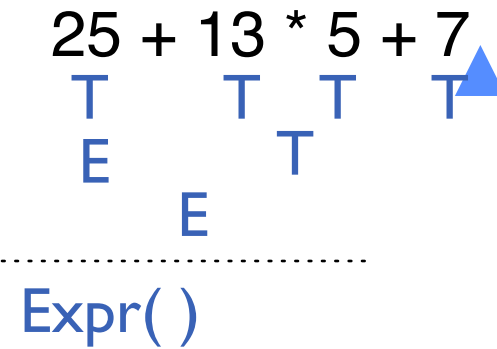
Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

```
25 + 13 * 5 + 7
 T       T  T
 E          T
      E
```

Expr( )
Term( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

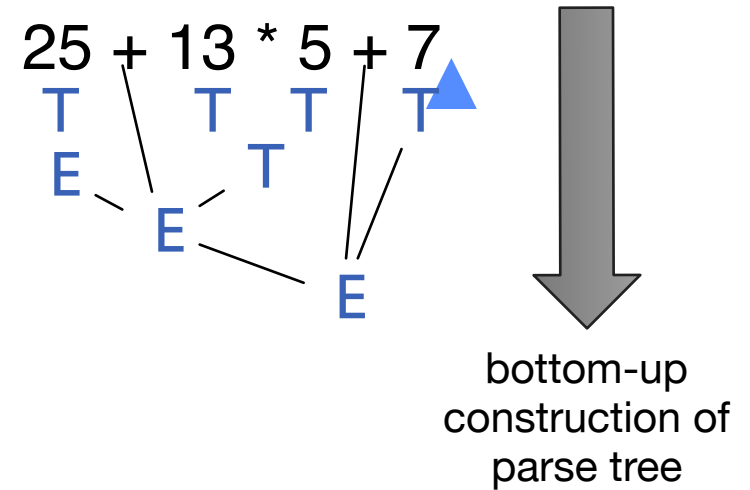Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

```
25 + 13 * 5 + 7
 T      T  T    T ▲
 E         T
      E
```

Expr( )

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }  ▲
    else { Term(); }
}


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```

Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

25 + 13 * 5 + 7

bottom-up construction of parse tree

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}

Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```
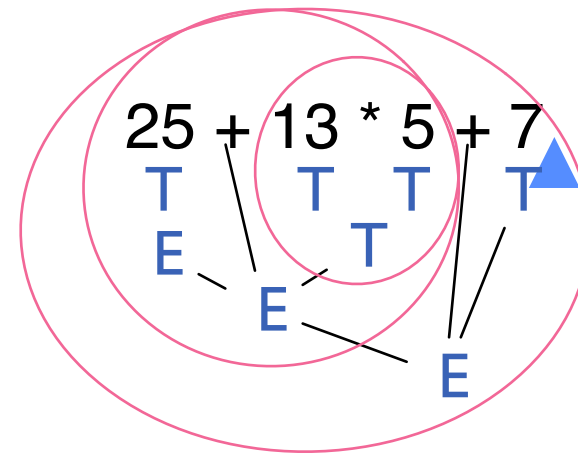
Expr ::= Expr + Term
Expr ::= Term
Term ::= Term * i
Term ::= i

$25 + 13 * 5 + 7$

Angelic recursive descent parser:

```
Tree Expr( ) {
    if (magic) {  Expr(); match('+'); Term(); }
    else { Term(); }
}  ▲


Tree Term( ) {
    if (magic) {  Term(); match('*'); match(int); }
    else { match(int); }
}
```