Parsing 2
Simple Boolean Expression Grammar

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
Example Program

Prog → StmtSeq
StmtSeq → Stmt StmtSeq
StmtSeq → ε
Stmt → Id = Expr ;
Expr → Expr || Term
Expr → Term
Term → Term && Factor
Term → Factor
Factor → ! Factor
Factor → ( Expr )
Factor → Id
Factor → True
Factor → False

x = True;
y = False;
w = x || y;
Example Program
Top Down Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program

```
x = True;
y = False;
w = x || y;
```

```
Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
```
Example Program

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program

\[
x = \text{True}; \\
y = \text{False}; \\
w = x \lor y;
\]

```
Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
```

```
x = \text{True}; \\
y = \text{False}; \\
w = {\text{true}} \lor \text{false};
```
Example Program

```
x = True;
y = False;
w = x || y;
```
Example Program

```plaintext
x = True;
y = False;
w = x || y;
```

Problems:
- Prog -> StmtSeq
- StmtSeq -> Stmt StmtSeq
- StmtSeq -> ε
- Stmt -> Id = Expr;
- Expr -> Expr || Term
- Expr -> Term
- Term -> Term && Factor
- Term -> Factor
- Factor -> ! Factor
- Factor -> ( Expr )
- Factor -> Id
- Factor -> True
- Factor -> False

Diagram:

```
Prog -> StmtSeq
    |    |    |    |    |    |    |    |
    Stmt Seq  Stmt Seq  Stmt Seq  Stmt Seq  ε
    |    |    |    |    |    |    |    |
    (id, x) = Expr;
    |    |    |    |    |    |    |    |
    Term Seq  Stmt Seq  Term Seq  Term Seq  ε
    |    |    |    |    |    |    |    |
    (id, y) = Expr;
    |    |    |    |    |    |    |    |
    Term Seq  Stmt Seq  Term Seq  Term Seq  ε
    |    |    |    |    |    |    |    |
    (id, w) = Expr;
    |    |    |    |    |    |    |    |
    Term Seq  Stmt Seq  Term Seq  Term Seq  ε
    |    |    |    |    |    |    |    |
    (id, x)
    |    |    |    |    |    |    |    |
    True Seq  False Seq  Term Seq  Factor Seq  ε
    |    |    |    |    |    |    |    |
    True Seq  False Seq  Term Seq  Factor Seq  ε
    |    |    |    |    |    |    |    |
    (id, y)
    |    |    |    |    |    |    |    |
    Factor Seq  ε
    |    |    |    |    |    |    |    |
    Factor Seq  ε
    |    |    |    |    |    |    |    |
    Factor Seq  ε
    |    |    |    |    |    |    |    |
    Factor Seq  ε
```
Example Program

x = True;
y = False;
w = x || y;

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

(id, x) = Expr ;
(id, y) = Expr ;
(id, w) = Expr ;
Example Program

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program

x = True;
y = False;
w = x || y;

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
Example Program

\[
\text{Prog} \rightarrow \text{StmtSeq} \\
\text{StmtSeq} \rightarrow \text{Stmt} \text{ StmtSeq} \\
\text{StmtSeq} \rightarrow \epsilon \\
\text{Stmt} \rightarrow \text{Id} = \text{Expr} ; \\
\text{Expr} \rightarrow \text{Expr} | | \text{Term} \\
\text{Expr} \rightarrow \text{Term} \\
\text{Term} \rightarrow \text{Term} \&\& \text{Factor} \\
\text{Term} \rightarrow \text{Factor} \\
\text{Factor} \rightarrow \neg \text{Factor} \\
\text{Factor} \rightarrow ( \text{Expr} ) \\
\text{Factor} \rightarrow \text{Id} \\
\text{Factor} \rightarrow \text{True} \\
\text{Factor} \rightarrow \text{False}
\]

\[
x = \text{True}; \\
y = \text{False}; \\
w = x | | y;
\]
Example Program

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program

```
x = True;
y = False;
w = x || y;
```

```
Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
```

```
(id, x) = Expr ;
Stmt
(id, y) = Expr ;
Stmt
StmtSeq
(id, w) = Expr ;
Stmt
StmtSeq
ε
```

```
(id, x)
Term
(id, y)
Factor
True
```

```
(id, w)
Term
(id, y)
Factor
False
```

```
Expr
```

```
Term
```

```
Factor
```

```
(id, x)
```

```
(id, y)
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
Example Program

```
x = True;
y = False;
w = x || y;
```
x = True;
y = False;
w = x || y;

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Term -> ( Expr )
Term -> Id
Term -> True
Term -> False

(id, x) = Expr ;
(id, y) = Expr ;
(id, w) = Expr ;
True
False
(id, x)
(id, y)
(id, y)
(id, x)
x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

\[
\begin{align*}
\text{Prog} & \rightarrow \text{StmtSeq} \\
\text{StmtSeq} & \rightarrow \text{Stmt} \text{ StmtSeq} \\
\text{StmtSeq} & \rightarrow \epsilon \\
\text{Stmt} & \rightarrow \text{Id} = \text{Expr} \\
\text{Expr} & \rightarrow \text{Expr} \mid \mid \text{Term} \\
\text{Expr} & \rightarrow \text{Term} \\
\text{Term} & \rightarrow \text{Term} \&\& \text{Factor} \\
\text{Term} & \rightarrow \text{Factor} \\
\text{Factor} & \rightarrow \neg \text{Factor} \\
\text{Factor} & \rightarrow ( \text{Expr} ) \\
\text{Factor} & \rightarrow \text{Id} \\
\text{Factor} & \rightarrow \text{True} \\
\text{Factor} & \rightarrow \text{False}
\end{align*}
\]

\[
\begin{align*}
x & = \text{True; } \\
y & = \text{False; } \\
w & = x \mid \mid y;
\end{align*}
\]
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

x = True;
y = False;
w = x || y;

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
Example Program
Bottom Parse

```markdown
Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
```

```
x = True;
y = False;
w = x || y;
```
x = True;
y = False;
w = x || y;

Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Fact -> ! Factor
Fact -> ( Expr )
Fact -> Id
Fact -> True
Fact -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

```
x = True;
y = False;
w = x || y;
```

```
Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Term -> ( Expr )
Term -> Id
Term -> True
Term -> False
Stmt -> (id, x) = Expr ;
Stmt -> (id, y) = Expr ;
Stmt -> (id, w) = Expr ;
```

```
(id, x) = Expr ;
(id, y) = Expr ;
(id, w) = Expr ;
```

```
(id, x)
(id, y)
(id, w)
```

```
true
false
```

```
ε
```

```
(id, x)
(id, y)
```

```
(Expr,term,factor)
```

```
Expr
```

```
Stmt
```

```
StmtSeq
```

```
Prog
```

```
StmtSeq
```

```
Stmt
```

```
Expr
```

```
Term
```

```
Factor
```

```
true
```

```
false
```

```
(id, x)
```

```
(id, y)
```

```
(id, w)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
```

```
(Expr,term,factor)
Example Program
Bottom Parse

x = True;
y = False;
w = x || y;

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False
x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

x = True;
y = False;
w = x || y;

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

(id, x) = Expr ;
(id, y) = Expr ;
(id, w) = Expr ;
(id, x)
(id, y)
(id, w)
ε
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
Example Program
Bottom Parse

Prog \rightarrow\ StmtSeq
StmtSeq \rightarrow\ Stmt\ StmtSeq
StmtSeq \rightarrow\ \epsilon
Stmt \rightarrow\ Id\ =\ Expr\ ;
Expr \rightarrow\ Expr\ \texttt{||}\ Term
Expr \rightarrow\ Term
Term \rightarrow\ Term\ \&\&\ Factor
Term \rightarrow\ Factor
Factor \rightarrow\ \texttt{!}\ Factor
Factor \rightarrow\ (\ Expr\ )
Factor \rightarrow\ Id
Factor \rightarrow\ True
Factor \rightarrow\ False

x = True;
y = False;
w = x \texttt{||} y;
Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False

x = True;
y = False;
w = x || y;
x = True;
y = False;
w = x || y;

Example Program
Bottom Parse

Prog -> StmtSeq
StmtSeq -> Stmt StmtSeq
StmtSeq -> ε
Stmt -> Id = Expr ;
Expr -> Expr || Term
Expr -> Term
Term -> Term && Factor
Term -> Factor
Factor -> ! Factor
Factor -> ( Expr )
Factor -> Id
Factor -> True
Factor -> False