

# Languages-beta: SL-Disambiguation \*

The PPlanCompS Project

SL-Disambiguation.cbs | PLAIN | PRETTY

---

*Language* "SL"

## A Disambiguation

Specification of disambiguation for parsing currently requires direct use of the corresponding SDF3 notation in comments, along with embedded productions in CBS notation. The well-formedness of the SDF3 notation is not checked. A future version of CBS is to include appropriate constructs for specifying disambiguation.

*Syntax SDF*

context-free syntax

```
expr ::= expr '+' expr {left}
expr ::= expr '/' expr {left}
expr ::= expr '*' expr {left}
expr ::= expr '-' expr {left}
expr ::= expr '==' expr {non-assoc}
expr ::= expr '<=' expr {non-assoc}
expr ::= expr '<' expr {non-assoc}
expr ::= expr '!=' expr {non-assoc}
expr ::= expr '>=' expr {non-assoc}
expr ::= expr '>' expr {non-assoc}
expr ::= expr '&&' expr {left}
expr ::= expr '||' expr {left}
expr ::= id '=' expr {right}
expr ::= expr '.' id '=' expr {right}
```

context-free priorities

```
{left:
expr ::= expr '.' id '(' expr-list? ')'
}
>
{right:
expr ::= '!' expr
}
>
{left:
expr ::= expr '*' expr
expr ::= expr '&&' expr
}
```

---

\*Suggestions for improvement: [plancomps@gmail.com](mailto:plancomps@gmail.com).  
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.

```

>
{non-assoc:
  expr ::= expr '==' expr
  expr ::= expr '<=' expr
  expr ::= expr '<' expr
  expr ::= expr '!=' expr
  expr ::= expr '>=' expr
  expr ::= expr '>' expr
}
>
{left:
  expr ::= expr '+' expr
  expr ::= expr '-' expr
  expr ::= expr '||' expr
}
>
{right:
  expr ::= id '=' expr
  expr ::= expr '.' id '=' expr
}

```

*Lexis SDF*

```

lexical syntax
id = "function" {reject}
id = "true" {reject}
id = "false" {reject}
id = "return" {reject}
id = "break" {reject}
id = "continue" {reject}
id = "while" {reject}
id = "if" {reject}
id = "else" {reject}

```

```

lexical restrictions
// Ensure greedy matching for lexicals
nat -/- [0-9]
id -/- [a-zA-Z0-9_]

```