RuleExpression

```
alias Separator = RuleExpression;

syntax RuleExpression
{
  | Sequence                  = LeftRule=RuleExpression ^ 10 RightRules=(RuleExpression ^ 10)+
  | Not                      = "!" RuleExpression ^ 20
  | And                      = "&" RuleExpression ^ 20
  | Optional                  = RuleExpression "?" precedence 30
  | ZeroOrMany                = RuleExpression "*" precedence 30
  | OneOrMany                 = RuleExpression "+" precedence 30
  | FieldName                 = Name "=" RuleExpression ^ 11
  | Char                     = CharLiteral
  | String                   = StringLiteral
  | Rounds                    = "(" RuleExpression ")"
  | Call                     = QualifiedName BindingPower=("^" Number)?
  | ZeroOrManyWithSeparator       = "(" RuleExpression ";" Separator ")" "*"
  | ZeroOrManyWithHangingSeparator = "(" RuleExpression ";" Separator ";" "?" ")" "*"
  | OneOrManyWithSeparator        = "(" RuleExpression ";" Separator ")" "+"
  | OneOrManyWithHangingSeparator  = "(" RuleExpression ";" Separator ";" "?" ")" "+"
}
```

`RuleExpression` describes the grammar of the syntax rules and token rules (regular and extensible).

See also

Name
RuleAlias
SimpleRule
VoidRule
ExtensionRule
Number
RuleExpression.Sequence
RuleExpression.And
RuleExpression.Not
RuleExpression.Optional
RuleExpression.FieldName
RuleExpression.Char
RuleExpression.String
RuleExpression.Rounds
RuleExpression.Call
RuleExpression.ZeroOrMany
RuleExpression.OneOrMany
RuleExpression.ZeroOrManyWithSeparator
RuleExpression.ZeroOrManyWithHangingSeparator
RuleExpression.OneOrManyWithSeparator
RuleExpression.OneOrManyWithHangingSeparator