

```

fractal
:
FRACTAL '{ orbit color }' eof
;

orbit
:
ORBIT '[' complex ',' complex ']' '[' statevariablelist ']' '{ trap* begin? loop end? }'
;

color
:
COLOR '[' colorargb ']' '{ palette* colorinit? colorrule* }'
;

begin
:
BEGIN '{ beginstatement }'
;

loop
:
LOOP '[' INTEGER ',' INTEGER ']' '(' conditionexp ')' '{ loopstatement }'
;

end
:
END '{ endstatement }'
;

trap
:
TRAP VARIABLE '[' complex ']' '{ pathop* }'
;

pathop
:
PATHOP_0POINTS ';'
|
PATHOP_1POINTS '(' complex ')' ';'
|
PATHOP_2POINTS '(' complex ',' complex ')' ';'
|
PATHOP_3POINTS '(' complex ',' complex ',' complex ')' ';'
;

beginstatement
:
statement*
;

loopstatement
:
statement*
;

```

endstatement

```
:  
  statement*  
;
```

statement

```
:  
  VARIABLE '=' expression ';'?  
|  
  IF '(' conditionexp ')' '{' statement* '}' ELSE '{' statement* '}'  
|  
  IF '(' conditionexp ')' '{' statement* '}'  
|  
  IF '(' conditionexp ')' statement  
|  
  STOP ';'?  
;
```

statevariable

```
:  
  'real' VARIABLE  
|  
  'complex' VARIABLE  
|  
  VARIABLE  
;
```

statevariablelist

```
:  
  statevariable  
|  
  statevariablelist ',' statevariable  
;
```

simpleconditionexp

```
:  
  expression ('=' | '<' | '>' | '<=' | '>=' | '<>') expression  
|  
  VARIABLE '?' expression  
|  
  VARIABLE '~?' expression  
|  
  JULIA  
|  
  '(' conditionexp ')'  
;
```

conditionexp

```
:  
  simpleconditionexp  
|  
  conditionexp2  
|  
  conditionexp '!' conditionexp2  
;
```

conditionexp2
:
simpleconditionexp
|
conditionexp3
|
conditionexp2 '^' conditionexp3
;

conditionexp3
:
simpleconditionexp
|
conditionexp4
|
conditionexp3 '&' conditionexp4
;

conditionexp4
:
simpleconditionexp
|
'~' conditionexp4
;

simpleexpression
:
constant
|
variable
|
real
|
function
|
'(' expression ')'
|
'|' expression '|'
|
'<' expression '>'
|
'<' expression ';' expression '>'
;

expression
:
simpleexpression
|
complex
|
expression2
|
expression '+' expression2
|
expression2 '+' expression
|

expression '-' expression2
;

expression2
:
simpleexpression
|
expression3
|
expression2 '*' expression2
|
'-' expression2
|
'+' expression2
|
'|' '*'? expression2
|
expression2 '*'? '|'
;

expression3
:
simpleexpression
|
expression4
|
expression3 '/' expression3
;

expression4
:
simpleexpression
|
expression4 '^' expression4
;

function
:
'time' '(' ')'
|
('mod' | 'mod2' | 'pha' | 're' | 'im') '(' expression ')'
|
('cos' | 'sin' | 'tan' | 'acos' | 'asin' | 'atan') '(' expression ')'
|
('log' | 'exp' | 'sqrt' | 'abs' | 'ceil' | 'floor' | 'square' | 'saw' | 'ramp') '(' expression ')'
|
('pow' | 'atan2' | 'hypot' | 'max' | 'min' | 'pulse') '(' expression ',' expression ')'
;

constant
:
'e'
|
'pi'
|
'2pi'

;

variable

:
VARIABLE
;

real

:
'+'? (RATIONAL I INTEGER)
|
'-' (RATIONAL I INTEGER)
;

complex

:
'<' '+'? (RATIONAL I INTEGER) ',' '+'? (RATIONAL I INTEGER) '>'
|
'<' '+'? (RATIONAL I INTEGER) ',' '-' (RATIONAL I INTEGER) '>'
|
'<' '-' (RATIONAL I INTEGER) ',' '+'? (RATIONAL I INTEGER) '>'
|
'<' '-' (RATIONAL I INTEGER) ',' '-' (RATIONAL I INTEGER) '>'
|
'+'? (RATIONAL I INTEGER) '+' (RATIONAL I INTEGER) 'i'
|
'+'? (RATIONAL I INTEGER) '-' (RATIONAL I INTEGER) 'i'
|
'+'? (RATIONAL I INTEGER) 'i'
|
'-' (RATIONAL I INTEGER) '+' (RATIONAL I INTEGER) 'i'
|
'-' (RATIONAL I INTEGER) '-' (RATIONAL I INTEGER) 'i'
|
'-' (RATIONAL I INTEGER) 'i'
|
real
;

palette

:
PALETTE VARIABLE '{' paletteelement+ }'
;

paletteelement

:
[' colorargb '>' colorargb ',' INTEGER ',' expression ']' ';' |
[' colorargb '>' colorargb ',' INTEGER ']' ';' |
;

colorinit

:
INIT '{' colorstatement }'
;

```
colorstatement
:
  statement*
;
```

```
colorrule
:
  RULE '(' ruleexp ')' '[' (RATIONAL | INTEGER) ']' '{' colorexp '}'
;
```

```
ruleexp
:
  expression ('=' | '>' | '<' | '>=' | '<=' | '<>') expression
  |
  ruleexp ('&' | '|' | '^') ruleexp
;
```

```
colorexp
:
  expression
  |
  expression ',' expression ',' expression
  |
  expression ',' expression ',' expression ',' expression
  |
  VARIABLE '[' expression ']'
;
```

```
colorargb
:
  '(' (RATIONAL | INTEGER) ',' (RATIONAL | INTEGER) ',' (RATIONAL | INTEGER)
  ',' (RATIONAL | INTEGER) ')'
  |
  ARGB32
  |
  ARGB24
;
```

```
eof
:
  EOF
;
```

```
FRACTAL
:
  'fractal'
;
```

```
ORBIT
:
  'orbit'
;
```

```
TRAP
:
  'trap'
;
```

;

BEGIN

:

'begin'

;

LOOP

:

'loop'

;

END

:

'end'

;

INIT

:

'init'

;

IF

:

'if'

;

ELSE

:

'else'

;

STOP

:

'stop'

;

JULIA

:

'julia'

;

COLOR

:

'color'

;

PALETTE

:

'palette'

;

RULE

:

'rule'

;

ARGB32

```
:  
'#' ('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')  
('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')  
;
```

ARGB24

```
:  
'#' ('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')  
('0'..'9'|'a'..'f'|'A'..'F')('0'..'9'|'a'..'f'|'A'..'F')  
;
```

RATIONAL

```
:  
('0'..'9')+ '!' ('0'..'9')* '%'? | '!' ('0'..'9')+ '%'? | '0'..'9'+ '%!'?!'? ('0'..'9')+ '!' ('0'..'9')* ('e'|'E')  
(+'|-')? ('0'..'9')+ '%'? | '!' ('0'..'9')+ ('e'|'E') ('+'|-')? ('0'..'9')+ '%'? | ('0'..'9')+ ('e'|'E') ('+'|-')? ('0'..'9')+ '%'?  
;
```

INTEGER

```
:  
('0'..'9')+  
;
```

PATHOP_0POINTS

```
:  
'CLOSE'  
;
```

PATHOP_1POINTS

```
:  
'MOVETO'  
|  
'MOVEREL'  
|  
'MOVETOREL'  
|  
'LINETO'  
|  
'LINEREL'  
|  
'LINETOREL'  
;
```

PATHOP_2POINTS

```
:  
'ARCTO'  
|  
'ARCREL'  
|  
'ARCTOREL'  
|  
'QUADTO'  
|  
'QUADREL'  
|
```


'QUADTOREL'

;

PATHOP_3POINTS

:

'CURVETO'

|

'CURVEREL'

|

'CURVETOREL'

;

VARIABLE

:

('a'..'z'|'A'..'Z') ('a'..'z'|'A'..'Z'|'0'..'9')*

;

COMMENT

:

('/' ~ ('\n'|'\r')* '\r'? '\n' } | '/'* (.)? '*' { }) -> skip

;

WHITESPACE

:

(' '| '\t'| '\r'| '\n') -> skip

;