

RPAL Phrase Structure:

```
Rpal -> E;
E     -> 'let' D 'in' E           => 'let'
      -> 'fn' Vb+ '.' E         => 'lambda'
      -> Ew;
Ew    -> T 'where' Dr           => 'where'
      -> T;
```

# Tuple Expressions #####

```
T     -> Ta ( ',' Ta )+         => 'tau'
      -> Ta ;
Ta    -> Ta 'aug' Tc           => 'aug'
      -> Tc ;
Tc    -> B '->' Tc '|' Tc      => '->'
      -> B ;
```

# Boolean Expressions #####

```
B     -> B 'or' Bt              => 'or'
      -> Bt ;
Bt    -> Bt '&' Bs              => '&'
      -> Bs ;
Bs    -> 'not' Bp              => 'not'
      -> Bp ;
Bp    -> A Rl A                => 'rl'
      -> A ;
Rl    -> 'gr' | '>'              => 'gr'
      -> 'ge' | '>='             => 'ge'
      -> 'ls' | '<'              => 'ls'
      -> 'le' | '<='             => 'le'
      -> 'eq'                    => 'eq'
      -> 'ne'                    => 'ne';
```

# Arithmetic Expressions #####

```
A     -> A '+' At               => '+'
      -> A '-' At               => '-'
      -> '+' At                  => 'neg'
      -> '-' At
      -> At ;
At    -> At '*' Af             => '*'
      -> At '/' Af              => '/'
      -> Af ;
Af    -> Ap '**' Af            => '**'
      -> Ap ;
Ap    -> Ap '@' '<IDENTIFIER>' R => '@'
      -> R ;
```

```

# Rators And Rands #####

R      -> R Rn                      => 'gamma'
      -> Rn ;

Rn     -> '<IDENTIFIER>'
      -> '<INTEGER>'
      -> '<STRING>'
      -> 'true'                      => 'true'
      -> 'false'                   => 'false'
      -> 'nil'                     => 'nil'
      -> '(' E ') '
      -> 'dummy'                   => 'dummy' ;

# Definitions #####

D      -> Da 'within' D              => 'within'
      -> Da ;

Da     -> Dr ( 'and' Dr )+          => 'and'
      -> Dr ;

Dr     -> 'rec' Db                  => 'rec'
      -> Db ;

Db     -> Vl '=' E                  => '='
      -> '<IDENTIFIER>' Vb+ '=' E   => 'function_form'
      -> '(' D ') ' ;

# Variables #####

Vb     -> '<IDENTIFIER>'
      -> '(' Vl ') '
      -> '(' ') '                    => '()';

Vl     -> '<IDENTIFIER>' list ',,'   => ',, '?;

```