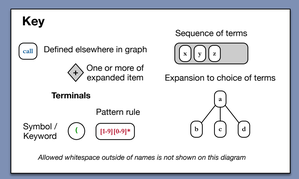
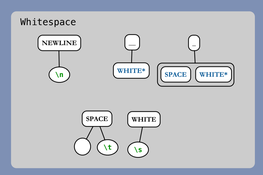
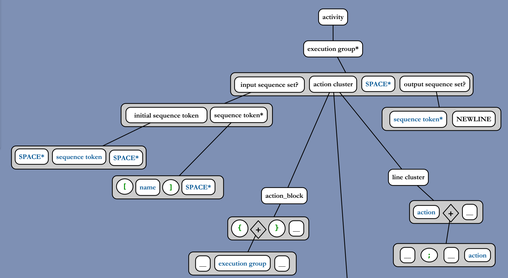
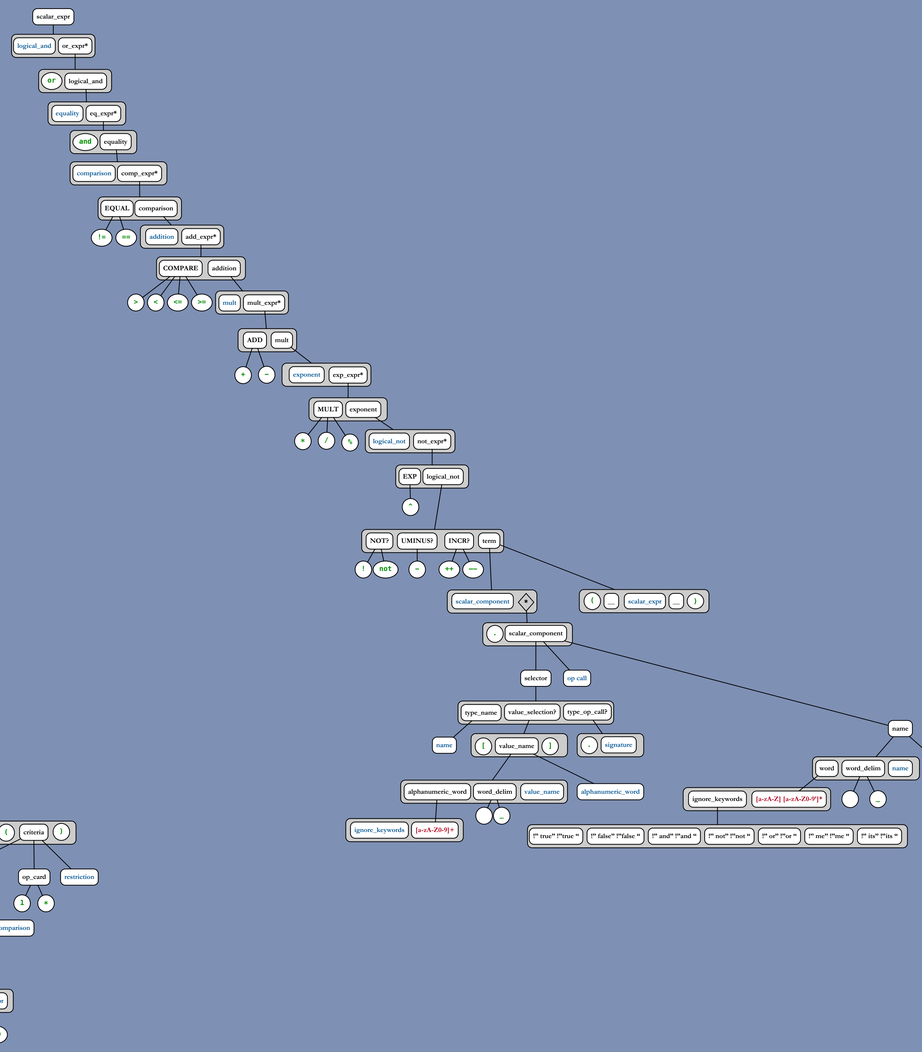

Key
call
Defined elsewhere in graph
+
One or more of expanded item
Terminals
Symbol / Keyword
(
Pattern rule
[1-9][0-9]*
Sequence of terms
x
y
z
Expansion to choice of terms
a
b
c
d
Allowed whitespace outside of names is not shown on this diagram


Whitespace
NEWLINE
\n
_
WHITE*
-
SPACE
WHITE*
SPACE
WHITE
_
\t
\s


activity
execution group*
input sequence set?
action cluster
SPACE*
output sequence set?
initial sequence token
sequence token*
sequence token*
NEWLINE
SPACE*
sequence token
SPACE*
(
name
)
SPACE*
action_block
line cluster
action
+
_
(
+
)
_
_
execution group
_
_
;
_
action
action
_

iteration
<<
inst_set
>>
action_block
conditional action
case action
enum_var
?
(
case+
)
name
enum_var
next_enum_val*
:
execution group
.
name
.
enum_var
guard action
guard
?
execution group
false_block?
bool_var
(
bool_expr
)
:
execution group
inst_delete
!=
inst_set
op call
name
signature
(
param_list?
)
param
param
,
param_list
name
type_qualifier?
:
name
signal
event_name
-->
destination
name
:
inst_set
duration?
@
scalar_expr
migration
inst_set?
>>
new_inst_spec
assignment
assign_scalar
scalar_variable
type_dec?
=
scalar_expr
name
::
type_name
name
assign_table
assign_inst_set
inst_set_var
assign_inst_op
inst_set
:=
<-:=
me
inst_source
selection?
path
class_name
op_call
inst_creation
inst_set_var
variable
hop+
/
rel_access
enum
rel_phrase
hippity_hop
class_name
R
[1-9][0-9]*
name
~
~!
link_action
link_source?
link_spec+
inst_set
&
enum
link_target
inst_set
assoc_class_link?
,
inst_set
inst_creation
=
new_inst_spec
class_name
attr_init_clause?
(
attr_init_list?
)
attr_init
,
attr_init_list
attr_init
attr_name
:
scalar_expr
(
criteria
)
op_card
,
restriction
op_card
restriction
1
*
attr_comparison
,
restriction
attr_comparison
op_order?
reflex?
attr_name
comparison
+*
-^
itr.
name
op_compare
scalar_expr
:
>
<
>=
<>
==


scalar_expr
logical_and
or_expr*
or
logical_and
equality
eq_expr*
and
equality
comparison
comp_expr*
EQUAL
comparison
!=
==
addition
add_expr*
COMPARE
addition
<
<
<=
>=
mult
mult_expr*
ADD
mult
+
-
exponent
exp_expr*
MULT
exponent
*
/
%
logical_not
not_expr*
EXP
logical_not
^
NOT?
UMINUS?
INCR?
term
!
not
-
++
~~
scalar_component
(
_
scalar_expr
_
)
.
scalar_component
selector
op call
type_name
value_selection?
type_op_call?
name
[
value_name
]
.
signature
alphanumeric_word
word_delim
value_name
alphanumeric_word
ignore_keywords
[a-zA-Z0-9]+
_
-
name
word
word_delim
name
word
ignore_keywords
[a-zA-Z]
[a-zA-Z0-9]*
_
-
i" true" i"true "
i" false" i"false "
i" and" i"and "
i" not" i"not "
i" or" i"or "
i" me" i"me "
i" its" i"its "