

section *ZFormalSyntax* **parents** *basic_toolkit*

[*GenName*, *OperatorTemplate*]

DeclName ::= *decl*⟨⟨*NAME*⟩⟩

RefName ::= *ref*⟨⟨*NAME*⟩⟩

Formals == seq₁ *NAME*

UPTO, *FFUN*, *FUN*, *MAPSTO* : *NAME*

MINUS, *PLUS*, *IN*, *DOM* : *NAME*

OVERRIDE : *NAME*

EQUAL, *GT* : *NAME*

NATURAL, *INTEGER* : *NAME*

STROKECHAR ::= *primestroke* | *outputstroke* | *inputstroke*

DECIMAL == 0.9

STROKE ::= *char*⟨⟨*STROKECHAR*⟩⟩ | *decimal*⟨⟨*DECIMAL*⟩⟩

Declaration ::= *variable*⟨⟨(seq₁ *DeclName*) × *Expression*⟩⟩ |
constant⟨⟨*DeclName* × *Expression*⟩⟩ |
expression⟨⟨*Expression*⟩⟩

Predicate ::= *newline*⟨⟨*Predicate* × *Predicate*⟩⟩ |
semicolon⟨⟨*Predicate* × *Predicate*⟩⟩ |
forall⟨⟨*SchemaText* × *Predicate*⟩⟩ |
exists⟨⟨*SchemaText* × *Predicate*⟩⟩ |
unique⟨⟨*SchemaText* × *Predicate*⟩⟩ |
equivalence⟨⟨*Predicate* × *Predicate*⟩⟩ |
implication⟨⟨*Predicate* × *Predicate*⟩⟩ |
disjunction⟨⟨*Predicate* × *Predicate*⟩⟩ |
conjunction⟨⟨*Predicate* × *Predicate*⟩⟩ |
negation⟨⟨*Predicate*⟩⟩ |
relationapplication⟨⟨*Relation*⟩⟩ |
schemapredicate⟨⟨*Expression*⟩⟩ |
TRUE |
FALSE |
parenthesizedpredicate⟨⟨*Predicate*⟩⟩

Relation ::= *rel*⟨⟨*NAME* × seq *Expression*⟩⟩

Application ::= *app*⟨⟨*NAME* × seq *Expression*⟩⟩

SchemaText ::= *schematext*⟨⟨opt *DeclPart* × opt *Predicate*⟩⟩

DeclPart ::= *declpart*⟨⟨seq₁ *Declaration*⟩⟩

csc == (*SchemaText* \ {*schematext*(⟨⟩, ⟨⟩)})

\{*e* : *Expression* • *schematext*(⟨*declpart*(⟨*expression*(*e*)), ⟨⟩)⟩

sc == *SchemaText* \ {*e* : *Expression* • *schematext*(⟨*declpart*(⟨*expression*(*e*)), ⟨⟩)⟩

Expression ::= *schemaforall*⟨⟨*SchemaText* × *Expression*⟩⟩ |
schemaexists⟨⟨*SchemaText* × *Expression*⟩⟩ |
schemaunique⟨⟨*SchemaText* × *Expression*⟩⟩ |
functionconstruction⟨⟨*SchemaText* × *Expression*⟩⟩ |
definitedescription⟨⟨*SchemaText* × *Expression*⟩⟩ |
substitution⟨⟨seq₁(*DeclName* × *Expression*) × *Expression*⟩⟩ |
schemaequivalence⟨⟨*Expression* × *Expression*⟩⟩ |
schemaimplication⟨⟨*Expression* × *Expression*⟩⟩ |
schemadisjunction⟨⟨*Expression* × *Expression*⟩⟩ |
schemaconjunction⟨⟨*Expression* × *Expression*⟩⟩ |
schemanegation⟨⟨*Expression*⟩⟩ |
conditional⟨⟨*Predicate* × *Expression* × *Expression*⟩⟩ |
composition⟨⟨*Expression* × *Expression*⟩⟩ |
piping⟨⟨*Expression* × *Expression*⟩⟩ |
hiding⟨⟨*Expression* × seq₁ *DeclName*⟩⟩ |
projection⟨⟨*Expression* × *Expression*⟩⟩ |
precondition⟨⟨*Expression*⟩⟩ |
product⟨⟨*Expression* × *Expression*⟩⟩ |
powerset⟨⟨*Expression*⟩⟩ |
functionapplication⟨⟨*Application*⟩⟩ |
application⟨⟨*Expression* × *Expression*⟩⟩ |
decoration⟨⟨*Expression* × *STROKE*⟩⟩ |
schemarenaming⟨⟨*Expression* × seq₁(*DeclName* × *DeclName*)⟩⟩ |
bindingselection⟨⟨*Expression* × *RefName*⟩⟩ |
tupleselection⟨⟨*Expression* × ℕ⟩⟩ |
bindingconstruction⟨⟨*Expression* × seq *STROKE*⟩⟩ |
reference⟨⟨*RefName*⟩⟩ |
genericinstantiation⟨⟨*RefName* × seq₁ *Expression*⟩⟩ |
numberliteral⟨⟨ℤ⟩⟩ |
setextension⟨⟨seq *Expression*⟩⟩ |
setcomprehension⟨⟨*SchemaText* × *Expression*⟩⟩ |
characteristicsetcomprehension⟨⟨*csc*⟩⟩ |
schemaconstruction⟨⟨*sc*⟩⟩ |
bindingextension⟨⟨seq(*DeclName* × *Expression*)⟩⟩ |
tupleextension⟨⟨{*s* : seq *Expression* | # *s* ≥ 2}⟩⟩ |
characteristicdefinitedescription⟨⟨*SchemaText*⟩⟩ |
parenthesizedexpression⟨⟨*Expression*⟩⟩ |
sequence⟨⟨seq *Expression*⟩⟩

Branch ::= *branch*⟨⟨*DeclName* × opt *Expression*⟩⟩
Freetype ::= *freetype*⟨⟨*NAME* × seq *Branch*⟩⟩

Paragraph ::= *giventypes*⟨⟨seq₁ *NAME*⟩⟩ |
axiomaticdescription⟨⟨*SchemaText*⟩⟩ |
schemadefinition⟨⟨*NAME* × *SchemaText*⟩⟩ |
genericaxiomaticdescription⟨⟨*Formals* × *SchemaText*⟩⟩ |
genericschemadefinition⟨⟨*NAME* × *Formals* × *SchemaText*⟩⟩ |
horizontaldefinition⟨⟨*DeclName* × *Expression*⟩⟩ |
generichorizontaldefinition⟨⟨*NAME* × *Formals* × *Expression*⟩⟩ |
genericoperatordefinition⟨⟨*GenName* × *Expression*⟩⟩ |
freetypes⟨⟨seq₁ *Freetype*⟩⟩ |
conjecture⟨⟨*Predicate*⟩⟩ |
genericconjecture⟨⟨*Formals* × *Predicate*⟩⟩ |
operatortemplate⟨⟨*OperatorTemplate*⟩⟩