

HotspotQueryLanguage

Rules

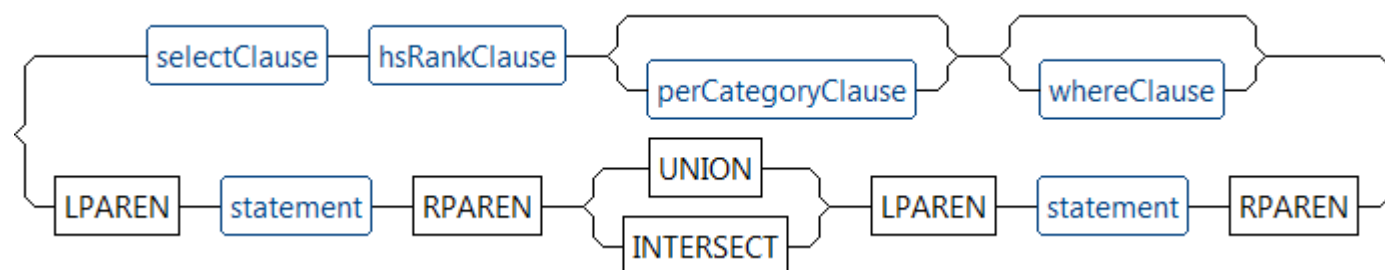
- statement
- selectClause
- hsRankClause
- perCategoryClause
- catRankClause
- quantity
- percent
- whereClause
- aggregateExpression
- minFunc
- maxFunc
- avgFunc
- sumFunc
- countFunc
- arithExpression
- term
- factor
- signedAtom
- arithAtom
- signedNumber
- variable
- logicalExpression
- orOperand
- andOperand
- logicalAtom
- relationalExpression
- comparison
- arithCompare
- textCompare
- text
- inclusion
- numberList
- textList
- SELECT
- HOTSPOTS
- TOP
- RANKED
- OF
- PER
- GROUP_UNIT

statement Top

Text notation:

```
statement : selectClause hsRankClause perCategoryClause? whereClause? | LPAREN statement RPAREN (UNION | INTERSECT) LPAREN statement RPAREN ;
```

Visual notation:

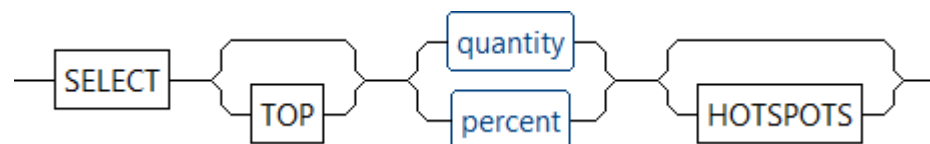


selectClause Top

Text notation:

```
selectClause : SELECT TOP? (quantity | percent) HOTSPOTS? ;
```

Visual notation:



hsRankClause Top

Text notation:

```
hsRankClause : RANKED BY arithExpression (ASCENDING | DESCENDING)? ;
```

- BY
- FROM
- WHERE
- AND
- OR
- NOT
- IN
- TRUE
- FALSE
- UNION
- INTERSECT
- ASCENDING
- DESCENDING
- MIN
- MAX
- AVG
- SUM
- COUNT
- A
- B
- C
- D
- E
- F
- G
- H
- I
- J
- K
- L
- M
- N
- O
- P
- Q
- R
- S
- T
- U
- V
- W
- X
- Y
- Z

Visual notation:

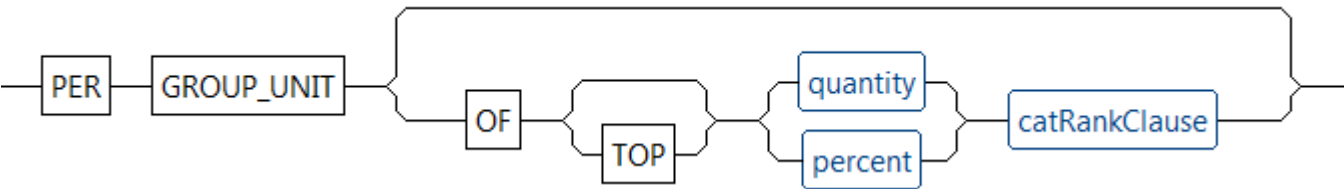


`perCategoryClause` Top

Text notation:

`perCategoryClause` : *PER* *GROUP_UNIT* (*OF TOP?* (*quantity* | *percent*) *catRankClause*)? ;

Visual notation:



`catRankClause` Top

Text notation:

`catRankClause` : *RANKED BY* *aggregateExpression* (*ASCENDING* | *DESCENDING*)? ;

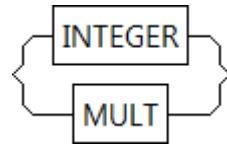
Visual notation:



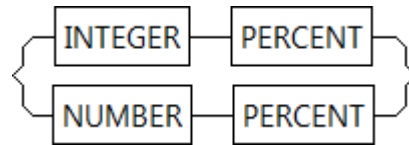
`quantity` Top

Text notation:

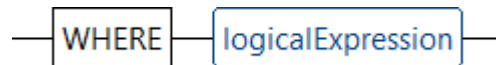
```
quantity : INTEGER | MULT ;
```

Visual notation:**percent** [Top](#)**Text notation:**

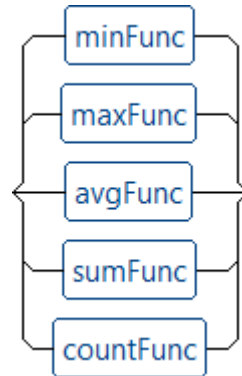
```
percent : INTEGER PERCENT | NUMBER PERCENT ;
```

Visual notation:**whereClause** [Top](#)**Text notation:**

```
whereClause : WHERE logicalExpression ;
```

Visual notation:**aggregateExpression** [Top](#)**Text notation:**

```
aggregateExpression : minFunc | maxFunc | avgFunc | sumFunc | countFunc ;
```

Visual notation:**minFunc** [Top](#)**Text notation:**

```
minFunc : MIN LPAREN arithExpression RPAREN ;
```

Visual notation:**maxFunc** [Top](#)**Text notation:**

```
maxFunc : MAX LPAREN arithExpression RPAREN ;
```

Visual notation:**avgFunc** [Top](#)**Text notation:**

```
avgFunc : AVG LPAREN arithExpression RPAREN ;
```

Visual notation:



sumFunc Top

Text notation:

```
sumFunc : SUM LPAREN arithExpression RPAREN ;
```

Visual notation:

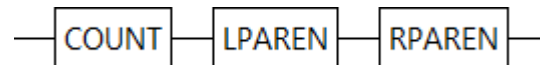


countFunc Top

Text notation:

```
countFunc : COUNT LPAREN RPAREN ;
```

Visual notation:

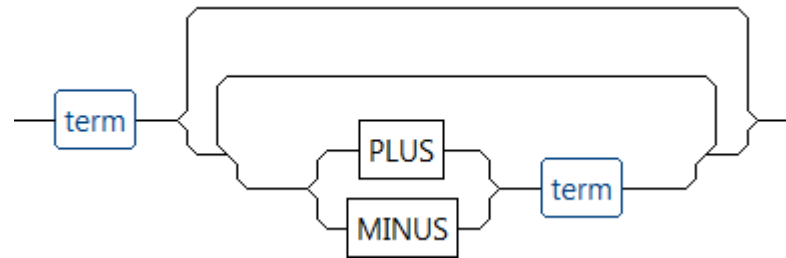


arithExpression Top

Text notation:

```
arithExpression : term ((PLUS | MINUS) term)* ;
```

Visual notation:

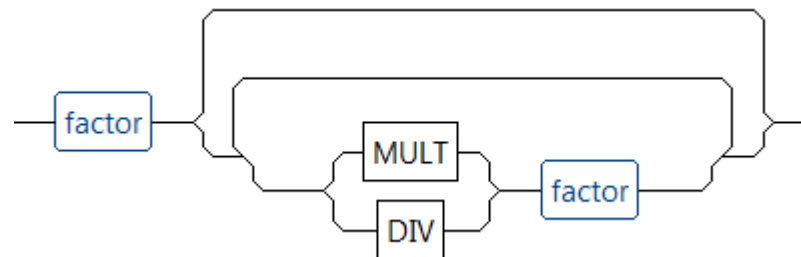


term Top

Text notation:

```
term : factor ((MULT | DIV) factor)* ;
```

Visual notation:

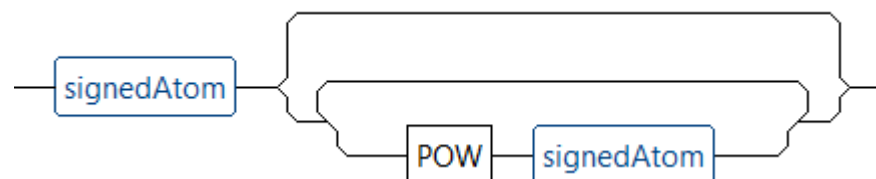


factor Top

Text notation:

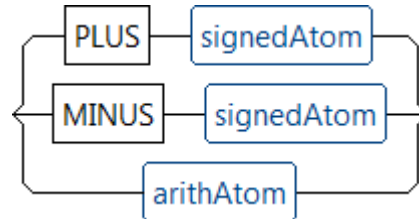
```
factor : signedAtom (POW signedAtom)* ;
```

Visual notation:

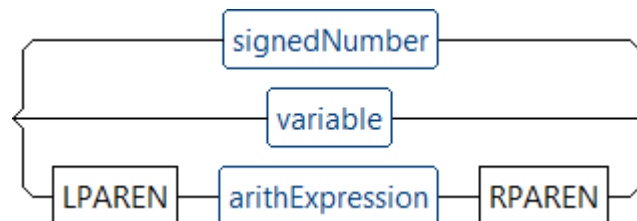


signedAtom [Top](#)**Text notation:**

```
signedAtom : PLUS signedAtom | MINUS signedAtom | arithAtom ;
```

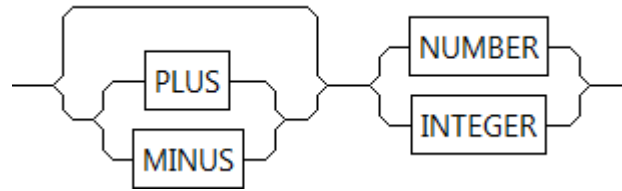
Visual notation:**arithAtom** [Top](#)**Text notation:**

```
arithAtom : signedNumber | variable | LPAREN arithExpression RPAREN ;
```

Visual notation:**signedNumber** [Top](#)**Text notation:**

```
signedNumber : (PLUS | MINUS)? (NUMBER | INTEGER) ;
```

Visual notation:



variable Top

Text notation:

```
variable : ID ;
```

Visual notation:

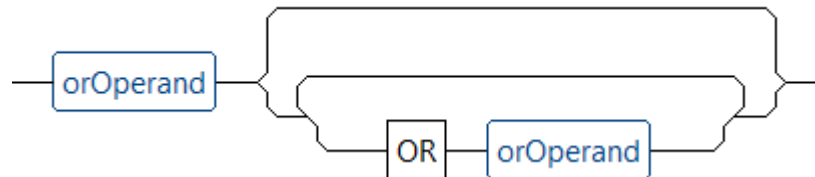


logicalExpression Top

Text notation:

```
logicalExpression : orOperand (OR orOperand)* ;
```

Visual notation:

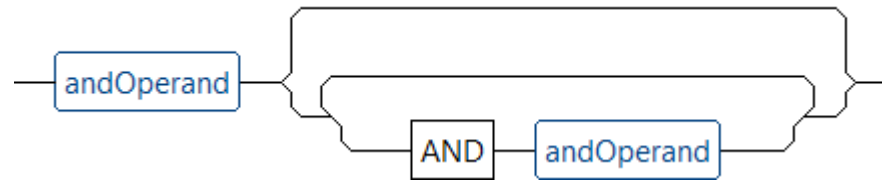


orOperand Top

Text notation:

```
orOperand : andOperand (AND andOperand)* ;
```

Visual notation:

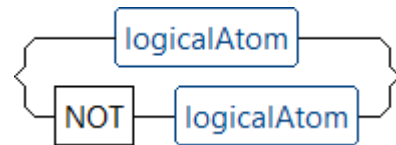


andOperand Top

Text notation:

```
andOperand : logicalAtom | NOT logicalAtom ;
```

Visual notation:

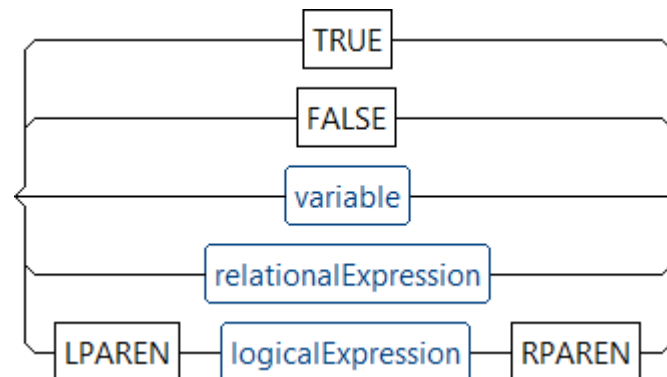


logicalAtom Top

Text notation:

```
logicalAtom : TRUE | FALSE | variable | relationalExpression | LPAREN logicalExpression RPAREN ;
```

Visual notation:

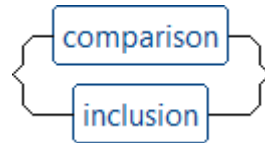


relationalExpression Top

Text notation:

```
relationalExpression : comparison | inclusion ;
```

Visual notation:



comparison Top

Text notation:

```
comparison : arithCompare | textCompare ;
```

Visual notation:

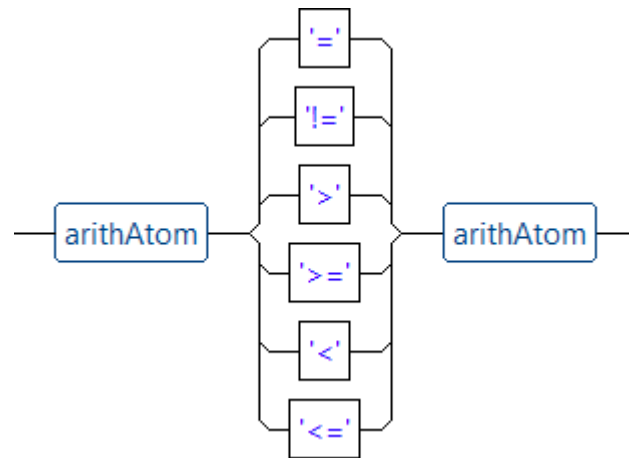


arithCompare Top

Text notation:

```
arithCompare : arithAtom ('=' | '!=' | '>' | '>=' | '<' | '<=') arithAtom ;
```

Visual notation:

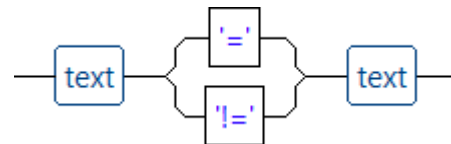


textCompare [Top](#)

Text notation:

```
textCompare : text ( '=' | '!=' ) text ;
```

Visual notation:

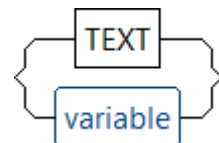


text [Top](#)

Text notation:

```
text : TEXT | variable ;
```

Visual notation:

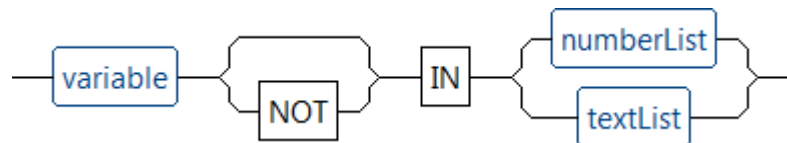


inclusion Top

Text notation:

```
inclusion : variable NOT? IN (numberList | textList) ;
```

Visual notation:

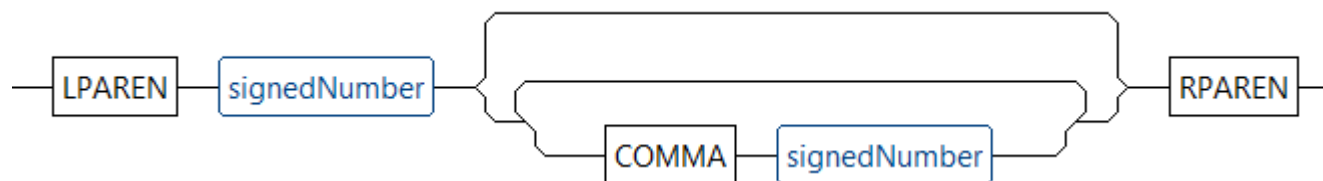


numberList Top

Text notation:

```
numberList : LPAREN signedNumber (COMMA signedNumber)* RPAREN ;
```

Visual notation:

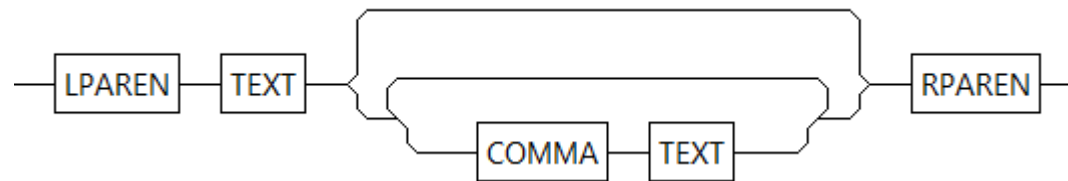


textList Top

Text notation:

```
textList : LPAREN TEXT (COMMA TEXT)* RPAREN ;
```

Visual notation:



SELECT Top

Text notation:

`SELECT : S E L E C T ;`

Visual notation:

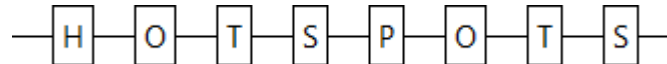


HOTSPOTS Top

Text notation:

`HOTSPOTS : H O T S P O T S ;`

Visual notation:

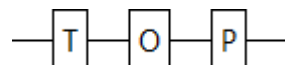


TOP Top

Text notation:

`TOP : T O P ;`

Visual notation:

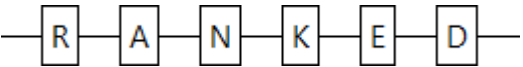


RANKED Top

Text notation:

RANKED : R A N K E D ;

Visual notation:



OF Top

Text notation:

OF : O F ;

Visual notation:

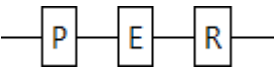


PER Top

Text notation:

PER : P E R ;

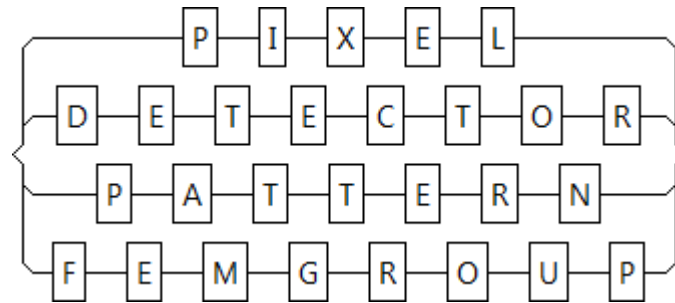
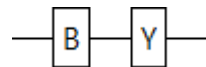
Visual notation:

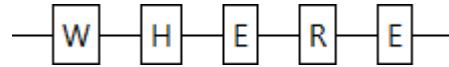


GROUP_UNIT Top

Text notation:

GROUP_UNIT : P I X E L | D E T E C T O R | P A T T E R N | F E M G R O U P ;

Visual notation:**BY** [Top](#)**Text notation:**`BY : B Y ;`**Visual notation:****FROM** [Top](#)**Text notation:**`FROM : F R O M ;`**Visual notation:****WHERE** [Top](#)**Text notation:**`WHERE : W H E R E ;`**Visual notation:**

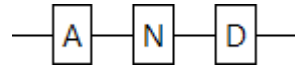


AND Top

Text notation:

AND : A N D ;

Visual notation:

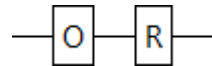


OR Top

Text notation:

OR : O R ;

Visual notation:



NOT Top

Text notation:

NOT : N O T ;

Visual notation:

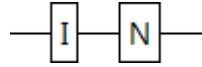


IN Top

Text notation:

IN : *I N* ;

Visual notation:

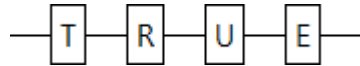


TRUE Top

Text notation:

TRUE : *T R U E* ;

Visual notation:

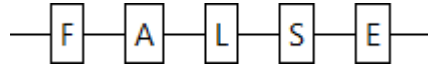


FALSE Top

Text notation:

FALSE : *F A L S E* ;

Visual notation:



UNION Top

Text notation:

UNION : *U N I O N* ;

Visual notation:



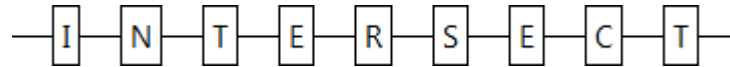
INTERSECT

[Top](#)

Text notation:

`INTERSECT : I N T E R S E C T ;`

Visual notation:



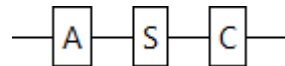
ASCENDING

[Top](#)

Text notation:

`ASCENDING : A S C ;`

Visual notation:



DESCENDING

[Top](#)

Text notation:

`DESCENDING : D E S C ;`

Visual notation:



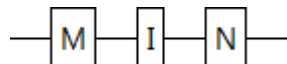
MIN

[Top](#)

Text notation:

`MIN : M I N ;`

Visual notation:

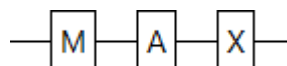


MAX Top

Text notation:

MAX : M A X ;

Visual notation:

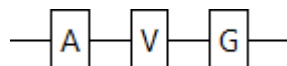


AVG Top

Text notation:

AVG : A V G ;

Visual notation:

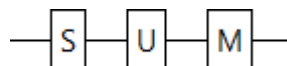


SUM Top

Text notation:

SUM : S U M ;

Visual notation:

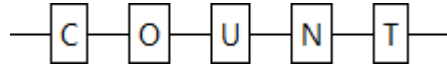


COUNT Top

Text notation:

COUNT : C O U N T ;

Visual notation:

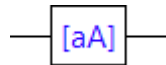


A Top

Text notation:

A : [aA] ;

Visual notation:



B Top

Text notation:

B : [bB] ;

Visual notation:

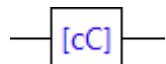


C Top

Text notation:

C : [cC] ;

Visual notation:

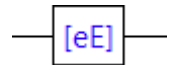


D Top**Text notation:**

D : [dD] ;

Visual notation:**E** Top**Text notation:**

E : [eE] ;

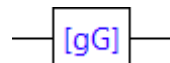
Visual notation:**F** Top**Text notation:**

F : [fF] ;

Visual notation:**G** Top**Text notation:**

G : [gG] ;

Visual notation:

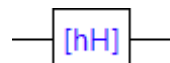


H Top

Text notation:

`H : [hH] ;`

Visual notation:

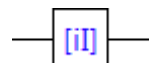


I Top

Text notation:

`I : [iI] ;`

Visual notation:

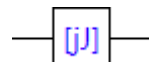


J Top

Text notation:

`J : [jJ] ;`

Visual notation:

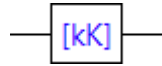


K Top

Text notation:

$K : [kK] ;$

Visual notation:



L Top

Text notation:

$L : [lL] ;$

Visual notation:

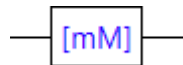


M Top

Text notation:

$M : [mM] ;$

Visual notation:

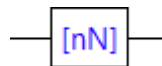


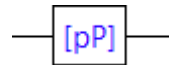
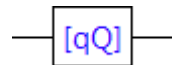
N Top

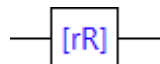
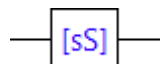
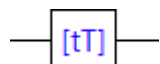
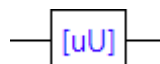
Text notation:

$N : [nN] ;$

Visual notation:



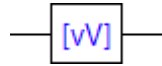
O Top**Text notation:** $O : [oO] ;$ **Visual notation:****P** Top**Text notation:** $P : [pP] ;$ **Visual notation:****Q** Top**Text notation:** $Q : [qQ] ;$ **Visual notation:****R** Top**Text notation:** $R : [rR] ;$

Visual notation:**S** Top**Text notation:**`S : [rR] ;`**Visual notation:****T** Top**Text notation:**`T : [sS] ;`**Visual notation:****U** Top**Text notation:**`U : [tT] ;`**Visual notation:****V** Top

Text notation:

$V : [vV] ;$

Visual notation:



W Top

Text notation:

$W : [wW] ;$

Visual notation:

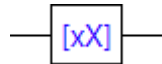


X Top

Text notation:

$X : [xX] ;$

Visual notation:

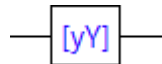


Y Top

Text notation:

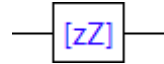
$Y : [yY] ;$

Visual notation:



Z **Top****Text notation:**

z : [**zZ**] ;

Visual notation:Generated by: [ANTLR 4 IDE](#). Copyright (c) 2013 [Edgar Espina](#)