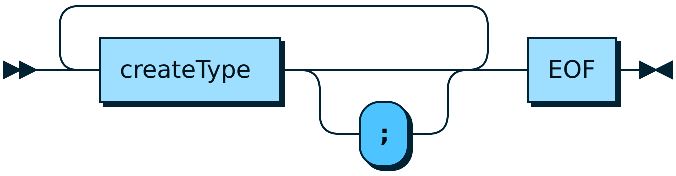
PG-Schema Grammar

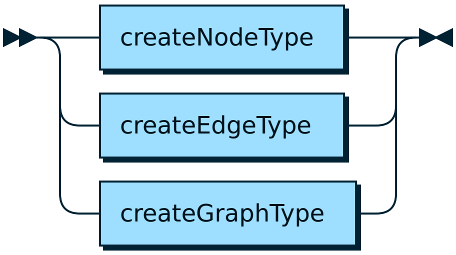
pgs:



```
pgs      ::= ( createType ';'? )+ EOF
```

no references

createType:

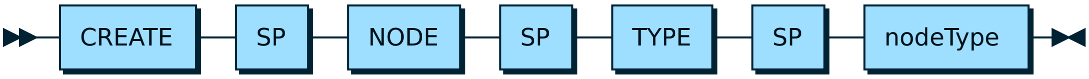


```
createType
         ::= createNodeType
           | createEdgeType
           | createGraphType
```

referenced by:

- pgs

createNodeType:

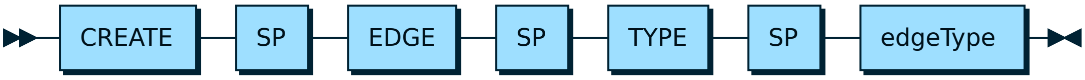


```
createNodeType
         ::= CREATE SP NODE SP TYPE SP nodeType
```

referenced by:

- createType

createEdgeType:

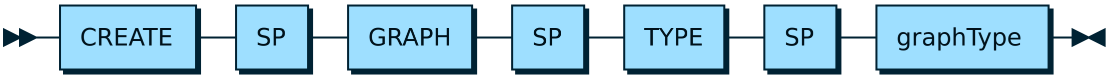


```
createEdgeType
         ::= CREATE SP EDGE SP TYPE SP edgeType
```

referenced by:

- createType

createGraphType:

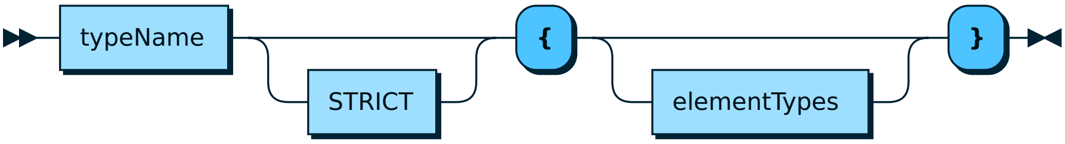


```
createGraphType
         ::= CREATE SP GRAPH SP TYPE SP graphType
```

referenced by:

- createType

graphType:

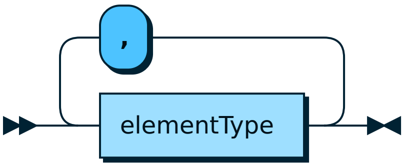


```
graphType
         ::= typeName STRICT? '{' elementTypes? '}'
```

referenced by:

- createGraphType

elementTypes:

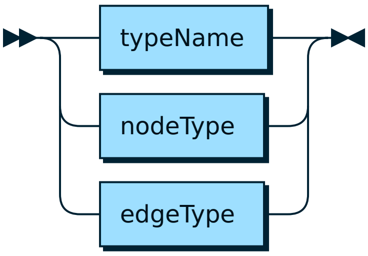


```
elementTypes
         ::= elementType ( ',' elementType )*
```

referenced by:

- graphType

elementType:



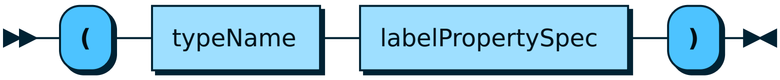
```
elementType
         ::= typeName
```

```
             | nodeType
             | edgeType
```

referenced by:

- elementTypes

nodeType:

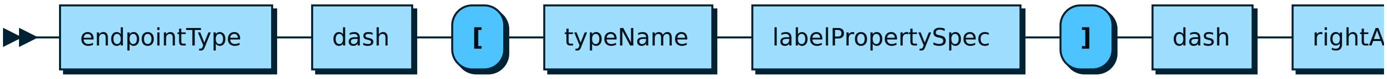


```
nodeType ::= '(' typeName labelPropertySpec ')'
```

referenced by:

- createNodeType
- elementType

edgeType:

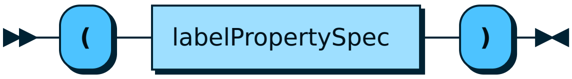


```
edgeType ::= endpointType dash '[' typeName labelPropertySpec ']' dash rightArrowHead endpointT
```

referenced by:

- createEdgeType
- elementType

endpointType:

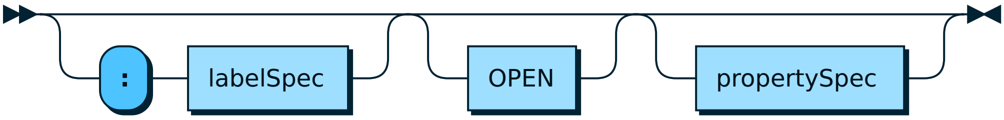


```
endpointType
         ::= '(' labelPropertySpec ')'
```

referenced by:

- edgeType

labelPropertySpec:



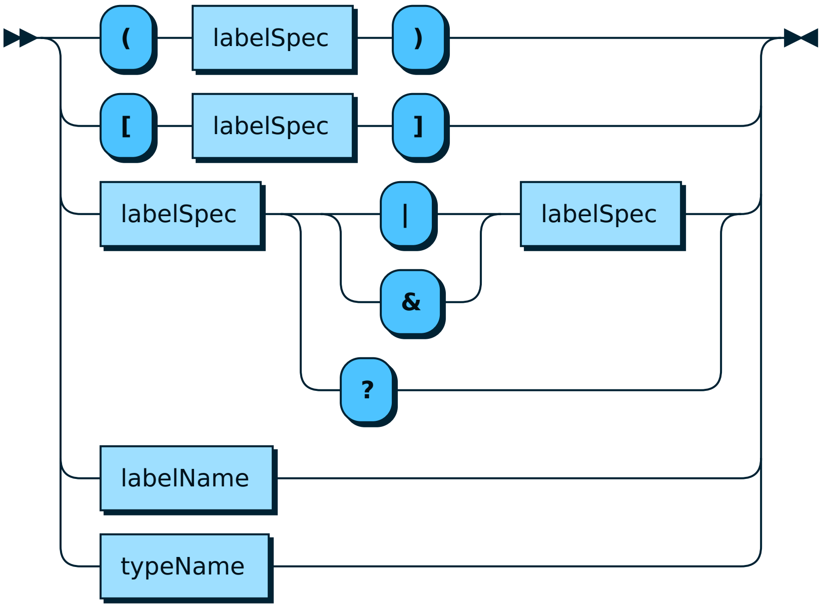
```
labelPropertySpec
         ::= ( ':' labelSpec )? OPEN? propertySpec?
```

referenced by:

- edgeType

- endpointType
- nodeType

labelSpec:

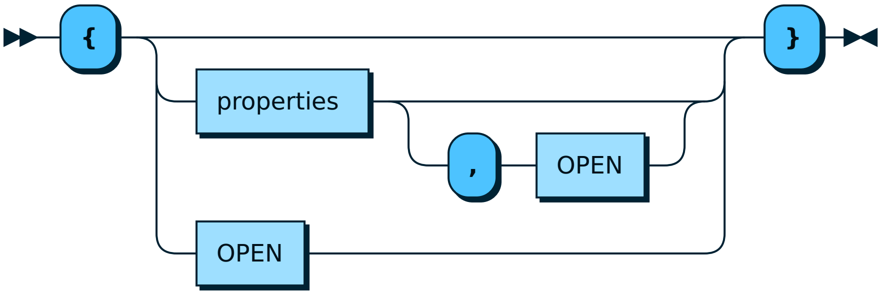


```
labelSpec
         ::= '(' labelSpec ')'
           | '[' labelSpec ']'
           | labelSpec ( ( '|' | '&' ) labelSpec | '?' )
           | labelName
           | typeName
```

referenced by:

- labelPropertySpec
- labelSpec

propertySpec:

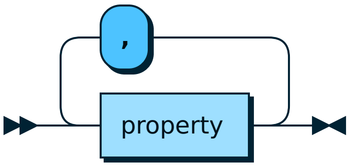


```
propertySpec
         ::= '{' ( properties ( ',' OPEN )? | OPEN )? '}'
```

referenced by:

- labelPropertySpec

properties:

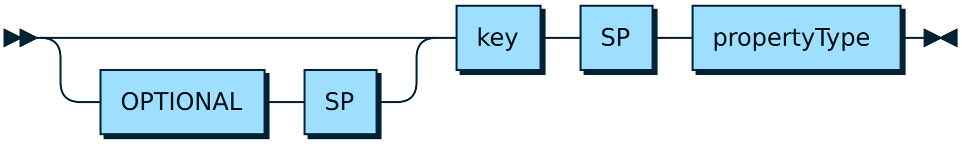


```
properties
         ::= property ( ',' property )*
```

referenced by:

- propertySpec

property:



```
property ::= ( OPTIONAL SP )? key SP propertyType
```

referenced by:

- properties

propertyType:

StringLiteral

```
propertyType
         ::= StringLiteral
```

referenced by:

- property

key:

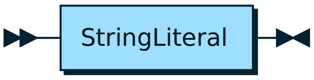
StringLiteral

```
key      ::= StringLiteral
```

referenced by:

- property

labelName:



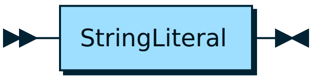
```
labelName
```

```
         ::= StringLiteral
```

referenced by:

- labelSpec

typeName:

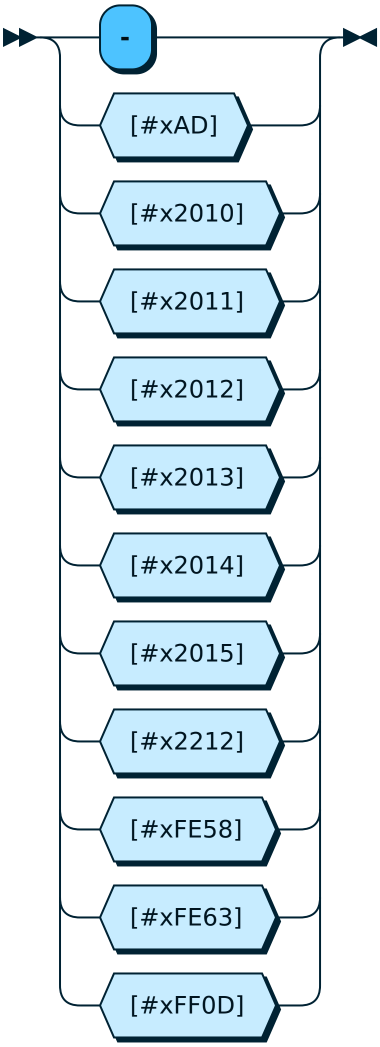


```
typeName ::= StringLiteral
```

referenced by:

- edgeType
- elementType
- graphType
- labelSpec
- nodeType

dash:

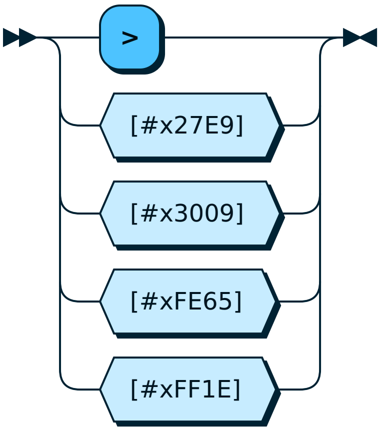


```
dash     ::= [-#xAD#x2010#x2011#x2012#x2013#x2014#x2015#x2212#xFE58#xFE63#xFF0D]
```

referenced by:

- edgeType

rightArrowHead:

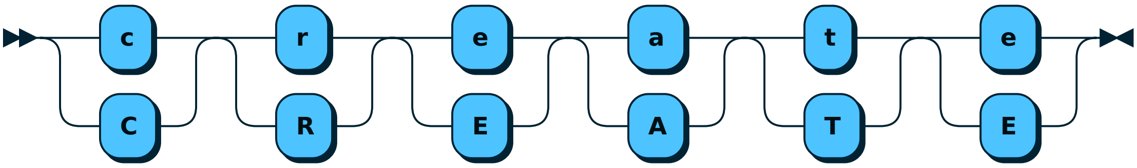


```
rightArrowHead
         ::= [>#x27E9#x3009#xFE65#xFF1E]
```

referenced by:

- edgeType

CREATE:

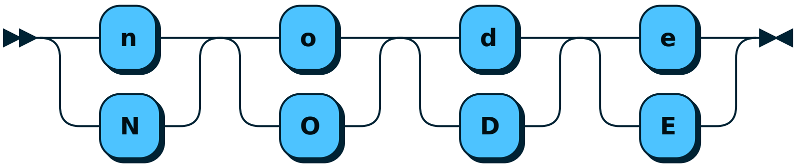


```
CREATE   ::= ( 'c' | 'C' ) ( 'r' | 'R' ) ( 'e' | 'E' ) ( 'a' | 'A' ) ( 't' | 'T' ) ( 'e' | 'E'
```

referenced by:

- createEdgeType
- createGraphType
- createNodeType

NODE:

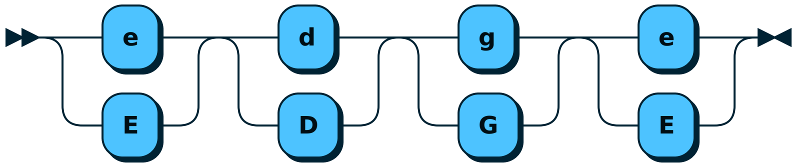


```
NODE     ::= ( 'n' | 'N' ) ( 'o' | 'O' ) ( 'd' | 'D' ) ( 'e' | 'E' )
```

referenced by:

- createNodeType

EDGE:

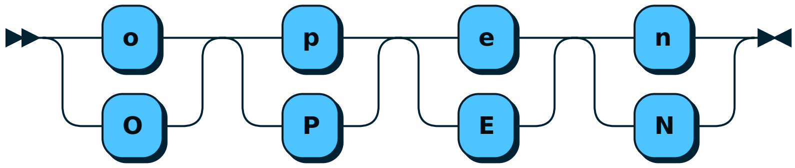


```
EDGE     ::= ( 'e' | 'E' ) ( 'd' | 'D' ) ( 'g' | 'G' ) ( 'e' | 'E' )
```

referenced by:

- createEdgeType

OPEN:

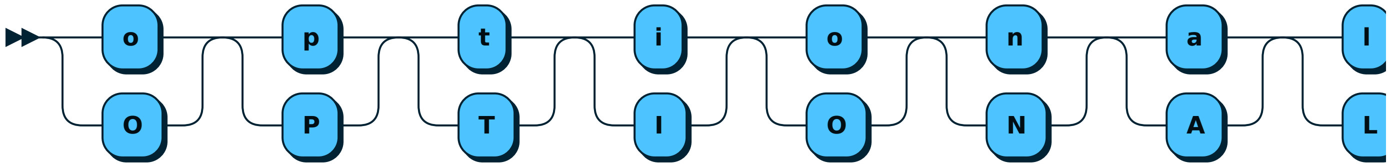


```
OPEN     ::= ( 'o' | 'O' ) ( 'p' | 'P' ) ( 'e' | 'E' ) ( 'n' | 'N' )
```

referenced by:

- labelPropertySpec
- propertySpec

OPTIONAL:

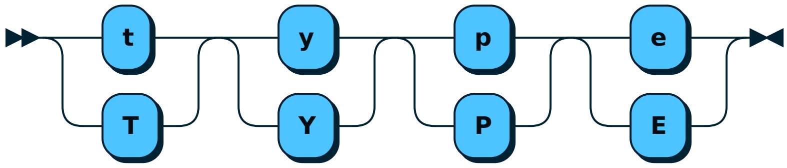


```
OPTIONAL ::= ( 'o' | 'O' ) ( 'p' | 'P' ) ( 't' | 'T' ) ( 'i' | 'I' ) ( 'o' | 'O' ) ( 'n' | 'N'
```

referenced by:

- property

TYPE:

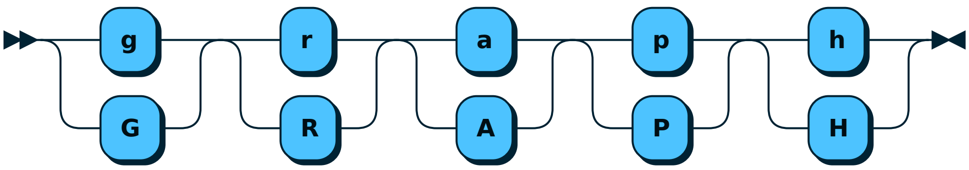


```
TYPE     ::= ( 't' | 'T' ) ( 'y' | 'Y' ) ( 'p' | 'P' ) ( 'e' | 'E' )
```

referenced by:

- createEdgeType
- createGraphType
- createNodeType

GRAPH:

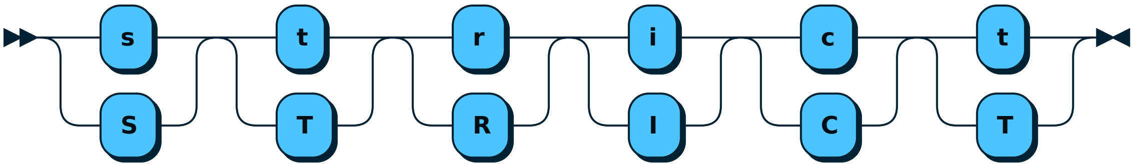


```
GRAPH    ::= ( 'g' | 'G' ) ( 'r' | 'R' ) ( 'a' | 'A' ) ( 'p' | 'P' ) ( 'h' | 'H' )
```

referenced by:

- createGraphType

STRICT:

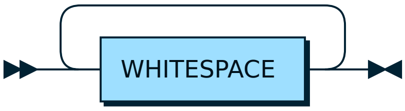


```
STRICT   ::= ( 's' | 'S' ) ( 't' | 'T' ) ( 'r' | 'R' ) ( 'i' | 'I' ) ( 'c' | 'C' ) ( 't' | 'T'
```

referenced by:

- graphType

SP:

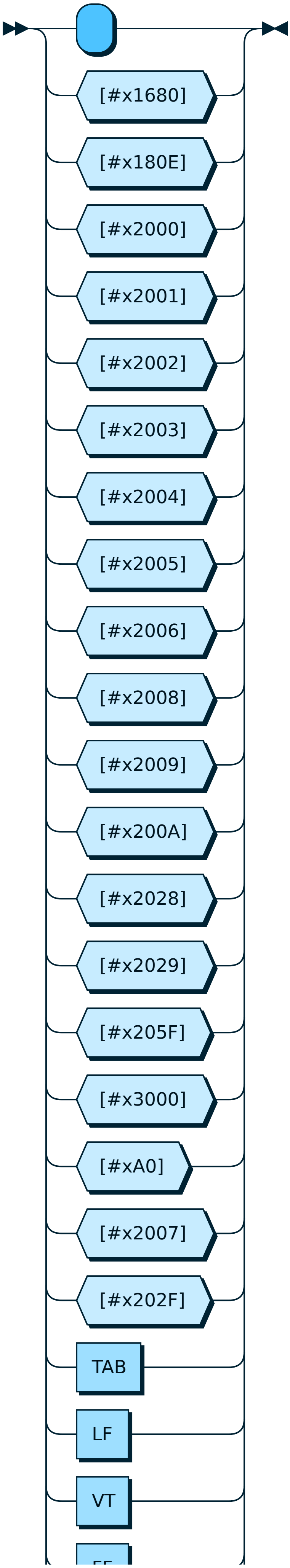


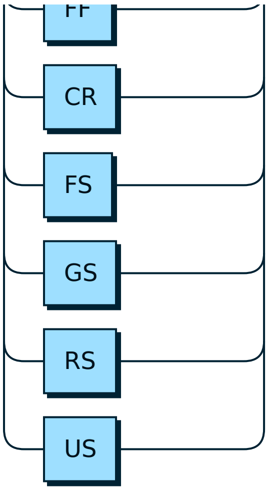
```
SP       ::= WHITESPACE+
```

referenced by:

- createEdgeType
- createGraphType
- createNodeType
- property

WHITESPACE:


[#x1680]
[#x180E]
[#x2000]
[#x2001]
[#x2002]
[#x2003]
[#x2004]
[#x2005]
[#x2006]
[#x2008]
[#x2009]
[#x200A]
[#x2028]
[#x2029]
[#x205F]
[#x3000]
[#xA0]
[#x2007]
[#x202F]
TAB
LF
VT

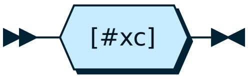


```
WHITESPACE
         ::= [ #x1680#x180E#x2000#x2001#x2002#x2003#x2004#x2005#x2006#x2008#x2009#x200A#x2028#x
           | TAB
           | LF
           | VT
           | FF
           | CR
           | FS
           | GS
           | RS
           | US
```

referenced by:

- SP

FF:

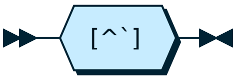


```
FF       ::= #xc
```

referenced by:

- WHITESPACE

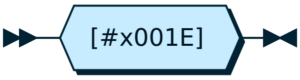
EscapedSymbolicName_0:



```
EscapedSymbolicName_0
         ::= [^`]
```

no references

RS:



```
RS       ::= #x001E
```

referenced by:

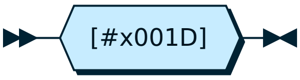
- WHITESPACE

StringLiteral_1:

```
StringLiteral_1
         ::= [^'\]
```

no references

GS:

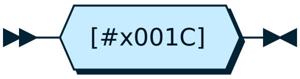


```
GS       ::= #x001D
```

referenced by:

- WHITESPACE

FS:

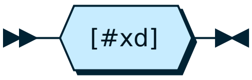


```
FS       ::= #x001C
```

referenced by:

- WHITESPACE

CR:

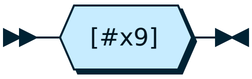


```
CR       ::= #xd
```

referenced by:

- WHITESPACE

TAB:



```
TAB      ::= #x9
```

referenced by:

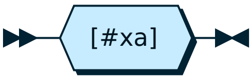
- WHITESPACE

StringLiteral_0:

```
StringLiteral_0
         ::= [^"\]
```

no references

LF:

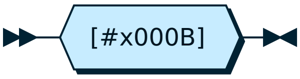


```
LF       ::= #xa
```

referenced by:

- WHITESPACE

VT:

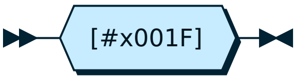


```
VT       ::= #x000B
```

referenced by:

- WHITESPACE

US:

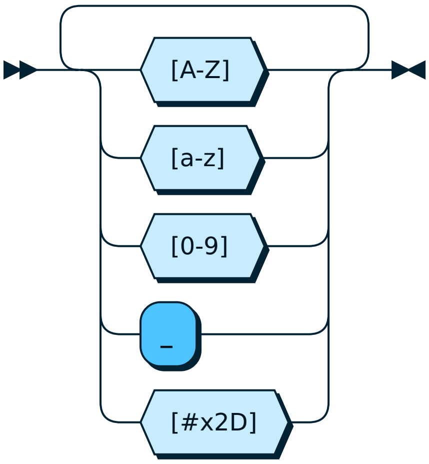


```
US       ::= #x001F
```

referenced by:

- WHITESPACE

StringLiteral:

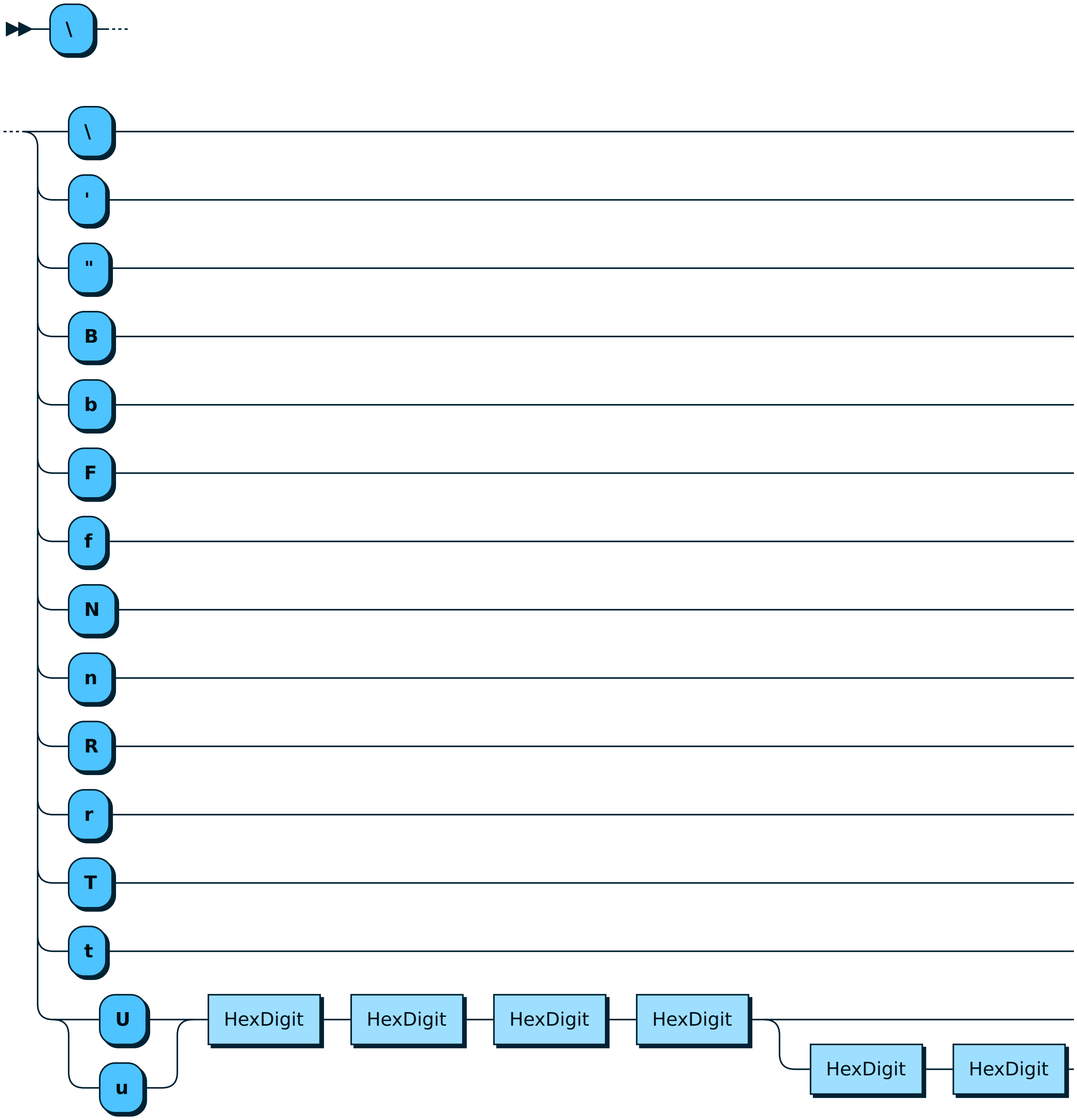


```
StringLiteral
         ::= [A-Za-z0-9_#x2D]+
```

referenced by:

- key
- labelName
- propertyType
- typeName

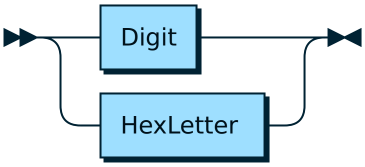
EscapedChar:



```
EscapedChar
         ::= '\' ( '\' | "'" | '"' | 'B' | 'b' | 'F' | 'f' | 'N' | 'n' | 'R' | 'r' | 'T' | 't'
             HexDigit ( HexDigit HexDigit HexDigit HexDigit )? )
```

no references

HexDigit:



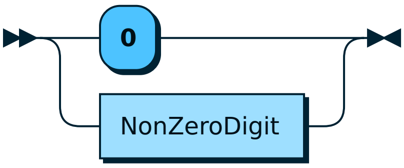
```
HexDigit ::= Digit
```

```
              | HexLetter
```

referenced by:

- EscapedChar

Digit:

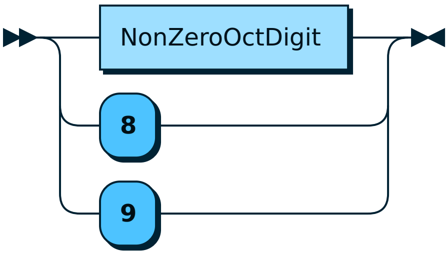


```
Digit    ::= '0'
              | NonZeroDigit
```

referenced by:

- HexDigit

NonZeroDigit:

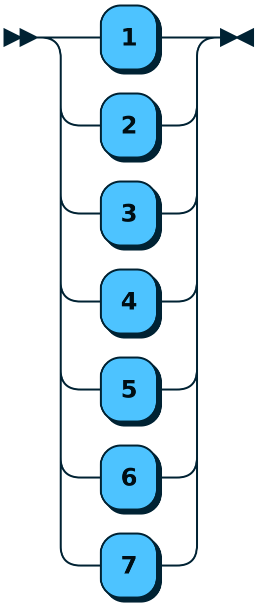


```
NonZeroDigit
         ::= NonZeroOctDigit
              | '8'
              | '9'
```

referenced by:

- Digit

NonZeroOctDigit:

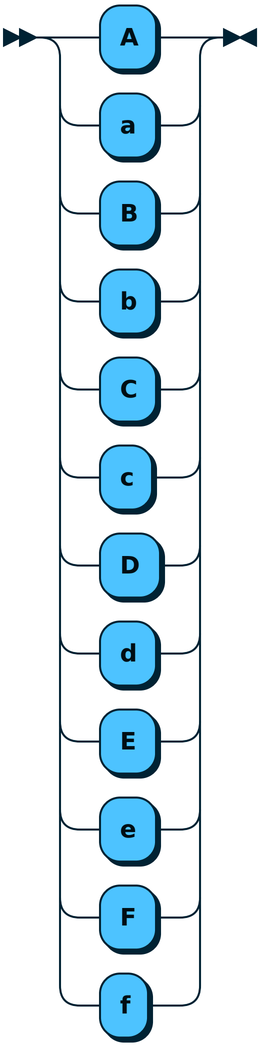


```
NonZeroOctDigit
         ::= '1'
           | '2'
           | '3'
           | '4'
           | '5'
           | '6'
           | '7'
```

referenced by:

- NonZeroDigit

HexLetter:



```
HexLetter
         ::= 'A'
           | 'a'
           | 'B'
           | 'b'
           | 'C'
           | 'c'
           | 'D'
           | 'd'
           | 'E'
           | 'e'
           | 'F'
           | 'f'
```

referenced by:

- HexDigit

EOF:

```
EOF      ::= $
```

referenced by:

- pgs

... generated by RR - Railroad Diagram Generator