# Proving of non-ambiguity of control statements productions

To prove this, the program segment that will be parsed is

rpt is (hisNum | !myNum) ?

{

rpt is !herNum & ourNum ?

hisNum : 2;

no

{

herNum : 1 + (2 % 3);

}

}

no rpt is myNum < 100 ?

{

myNum : myNum + 1;

}

It is a common occurrence in programming, the nested loops.

Leftmost derivation

* <ctr>
* rpt <cnd>
* rpt is <ex> ? <S> <cndp>
* rpt is <md><exp> ? <S> <cndp>
* rpt is <pow><mdp><exp> ? <S> <cndp>
* rpt is <neg><powp><mdp><exp> ? <S> <cndp>
* rpt is <term><powp><mdp><exp> ? <S> <cndp>
* rpt is (<ex>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (<md><exp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (<pow><mdp><exp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (<neg><powp><mdp><exp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (<term><powp><mdp><exp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID<powp><mdp><exp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID<mdp><exp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | <md>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | <mdp><exp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | <pow><mdp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | <neg><powp><mdp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | !<term><powp><mdp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | !ID <powp><mdp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | !ID <mdp>)<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | !ID )<powp><mdp><exp> ? <S> <cndp>
* rpt is (ID | !ID )<mdp><exp> ? <S> <cndp>
* rpt is (ID | !ID )<exp> ? <S> <cndp>
* rpt is (ID | !ID ) ? <S> <cndp>
* rpt is (ID | !ID ) ? {<SL>} <cndp>
* rpt is (ID | !ID ) ? {<S><SLP>} <cndp>
* rpt is (ID | !ID ) ? {<cnt><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt <cnd><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is <ex> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is <md><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is <pow><mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is <neg><powp><mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !<term><powp><mdp><exp> ? <S> <cndp><SLP>}<cndp>
* rpt is (ID | !ID ) ? {rpt is !ID<powp><mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID<mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID<exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & <md><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & <pow><mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & <neg><powp><mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & <term><powp><mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID<powp><mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID<mdp><exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID<exp> ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? <S> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? <A> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : <AP> <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : <ex>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : <md><exp>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : <pow><mdp><exp>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : <neg><powp><mdp><exp>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : <term><powp><mdp><exp>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst <powp><mdp><exp>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst <mdp><exp>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst <exp>; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; <cndp><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no <S><SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {<SL>}<SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {<S><SLP>}<SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {<A><SLP>}<SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : <AP> <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : <ex>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : <md> <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : <pow> <mdp> <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : <neg><powp> <mdp> <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : <term><powp> <mdp> <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst <powp> <mdp> <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID/ | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst <mdp> <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +<md><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +<pow><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +<neg><powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +<term><powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(<ex>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(<md><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(<pow><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(<pow><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(<neg><powp><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(<term><powp><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst<powp><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst <mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst <exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % <md><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % <pow><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % <neg><powp><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % <term><powp><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst <powp><mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst <mdp><exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst <exp>)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst)<powp><mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst) <mdp><exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst) <exp>; <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); <SLP>} <SLP>} <cndp>
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); } <SLP>} <cndp>

rpt is (ID | !ID ) ?

{

The grammar produced at this point, already has the same upper part of the sample program segment.

rpt is !ID & ID ?

ID : numcnst;

no

{

ID : numcnst +(numcnst % numcnst);

}

}

no

<S>

* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {<SL>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {<S><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {<cnt><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt <cnd> <SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is <ex> ? <S> <cndp> <SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is <md><exp> ? <S> <cndp> <SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is <pow><mdp><exp> ? <S> <cndp> <SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is <neg><powp><mdp><exp> ? <S> <cndp> <SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is <term><powp><mdp><exp> ? <S> <cndp><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID <powp><mdp><exp> ? <S> <cndp><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID <mdp><exp> ? <S> <cndp><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID <exp> ? <S> <cndp><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < <md><exp> ? <S> <cndp><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < <pow><mdp><exp> ? <S> <cndp><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < <neg><powp><mdp><exp> ? <S> <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < <term><powp><mdp><exp> ? <S> <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst <powp><mdp><exp> ? <S> <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst <mdp><exp> ? <S> <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst <exp> ? <S> <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? <S> <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {<SL>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {<S><SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {<A><SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : <AP> <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : <ex>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : <md><exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : <pow><mdp><exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : <neg> <powp> <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : <term> <powp> <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID <powp> <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID +<md><exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + <pow> <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + <neg> <powp> <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + <term> <powp> <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst <powp> <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst <mdp> <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst <exp>; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; <SLP>} <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } <cndp> <SLP> }
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } <SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } <S><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } <D><SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str <V> ; <SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str <dt>,<V> ;<SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str ID <dtp>,<V> ;<SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str ID ,<V> ;<SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str ID ,<dt> ;<SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str ID, ID<dtp> ;<SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str ID, ID : <dterm> ;<SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str ID, ID : stringliteral ;<SLP>}
* rpt is (ID | !ID ) ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst +(numcnst % numcnst); }} no {rpt is ID < numcnst ? {ID : ID + numcnst; } str ID, ID : stringliteral ;}

rpt is (ID | !ID ) ?

{

rpt is !ID & ID ?

ID : numcnst;

no

{

ID : numcnst + (numcnst % numcnst);

}

}

no

{

rpt is ID < numcnst ?

{

ID : ID + numcnst;

}

str ID, ID : stringliteral ;

}

Rightmost derivation

* <ctr>
* rpt <cnd>
* rpt is <ex> ? <S> <cndp>
* rpt is <ex> ? <S> no <S>
* rpt is <ex> ? <S> no {<SL>}
* rpt is <ex> ? <S> no {<S><SLP>}
* rpt is <ex> ? <S> no {<S><S><SLP>}
* rpt is <ex> ? <S> no {<S><S>}
* rpt is <ex> ? <S> no {<S><D>}
* rpt is <ex> ? <S> no {<S> str <V> ;}
* rpt is <ex> ? <S> no {<S> str <dt>,<V> ;}
* rpt is <ex> ? <S> no {<S> str <dt>,<dt> ;}
* rpt is <ex> ? <S> no {<S> str <dt>, ID<dtp> ;}
* rpt is <ex> ? <S> no {<S> str <dt>, ID:<dterm> ;}
* rpt is <ex> ? <S> no {<S> str <dt>, ID:stringliteral;}
* rpt is <ex> ? <S> no {<S> str ID<dtp>, ID:stringliteral;}
* rpt is <ex> ? <S> no {<S> str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {<cnt> str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt <cnd>} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? <S> <cndp>} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? <S> str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {<SL>} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {<S><SLP>} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {<S>} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {<A>} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<AP>} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<ex>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md><exp>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md>+<md><exp>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md>+<md>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md>+<pow><mdp>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md>+<pow>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md>+<neg><powp>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md>+<neg>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md>+<term>;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<md> + numcnst ;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<pow><mdp> + numcnst ;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<pow> + numcnst ;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<neg><powp> + numcnst ;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<neg> + numcnst ;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID:<term> + numcnst ;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <ex> ? {ID: ID + numcnst ;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md><exp> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < <md><exp> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < <md> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < <pow><mdp> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < <pow> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < <neg><powp> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < <neg> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < <term> ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <md> < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <pow><mdp> < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <pow> < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <neg><powp> < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <neg> < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is <term> < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? <S> no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {<SL>} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {<cnt>} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt <cnd>} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {<cnt>} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt <cnd>} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> <cndp>} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no <S>} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {<SL>}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {<S><SLP>}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {<S>}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {<A>}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <AP>}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <ex>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md><exp>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + <md><exp>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + <md>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + <pow><mdp>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + <pow>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + <neg><powp>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + <neg>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + <term>;}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<ex>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md><exp>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>%<md><exp>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>%<md>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>%<pow><mdp>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>%<pow>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>%<neg><powp>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>%<neg>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>%<term>);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<md>% numcnst );}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<pow><mdp>% numcnst );}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<pow> % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<neg><powp> % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<neg> % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (<term> % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <md> + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <pow><mdp> + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <pow> + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <neg><powp> + (numcnst % numcnst ); }} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <neg> + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : <term> + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <S> no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? <A> no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <AP> no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <ex>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <md><exp>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <md>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <pow><mdp>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <pow>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <neg><powp>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <neg>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : <term>; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <ex> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md><exp> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md>&<md><exp> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md>&<md> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md>&<pow><mdp> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md>&<neg><powp> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md>&<neg> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md>&<term> ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md>& ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md><exp> & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <md> & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <pow><mdp> & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <pow> & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <neg><powp> & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is <neg> & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is !<term> & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <ex> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md><exp> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | <md><exp> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | <md> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | <pow><mdp> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | <pow> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | <neg><powp> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | <neg> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | !<term> ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <md> | !ID ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <pow><mdp> | !ID ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <pow> | !ID ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <neg><powp> | !ID ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <neg> | !ID ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is <term> | !ID ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}
* rpt is ID | !ID ? {rpt is !ID & ID ? ID : numcnst; no {ID : numcnst + (numcnst % numcnst);}} no {rpt is ID < numcnst ? {ID: ID + numcnst;} str ID, ID:stringliteral;}

rpt is ID | !ID ?

{

rpt is !ID & ID ?

ID : numcnst;

no

{

ID : numcnst + (numcnst % numcnst);

}

}

no

{

rpt is ID < numcnst ?

{

ID : ID + numcnst;

}

str ID, ID : stringliteral;

}