

## CD23 Programming Language LL(1) Grammar

NPROG      <program> ::= CD23 <id> <global> <funcs> <mainbody>

NGLOB      <globals> ::= <consts> <types> <arrays>

Special     <consts> ::= constants <initlist> | ε

NLIST      <initlist> ::= <init> <opt\_initlist>

Special     <opt\_initlist> ::= , <initlist> | ε

We applied left factoring to this rule because it wasn't LL(1)

NINIT      <init> ::= <id> is <expr>

Special     <types> ::= types <typelist> | ε

Special     <arrays> ::= arrays <arrdecls> | ε

NFUNCS     <funcs> ::= <func> <funcs> | ε

NMAIN      <mainbody> ::= main <slist> begin <stats> end CD23 <id>

NSDLST     <slist> ::= <decl\_prefix> <sdecl> <opt\_slist>

Special     <opt\_slist> ::= , <slist> | ε

We applied left factoring to this rule because it wasn't LL(1)

NYPEL      <typelist> ::= <type> <opt\_typelist>

Special     <opt\_typelist> ::= <typelist> | ε

NRTYPE     <type> ::= <structed> is <fields> end

NATYPE     <type> ::= <typeid> is array [<expr>] of <structid> end

NFLIST     <fields> ::= <decl\_prefix> <sdecl> <opt\_fields>

Special     <opt\_fields> ::= , <fields> | ε

Special     <decl\_prefix> ::= <id> :

NSDECL     <sdecl> ::= <decl\_prefix> <stype>

NALIST     <arrdecls> ::= <decl\_prefix> <arrdecl> <opt\_arrdecl>

Special     <opt\_arrdecl> ::= , <arrdecls> | ε

NARRD      <arrdecl> ::= <decl\_prefix> <typeid>

NFUND      <func> ::= func <id> (<plist>):<rtype> <funcbody>

Special     <rtype> ::= <stype> | void

Special     <plist> ::= <params> | ε

NPLIST      <params> ::= <param> <opt\_params>

Special     <opt\_params> ::= , <params> | ε

NSIMP      <param> ::= <decl\_prefix> <sdecl>

We applied left factoring to these rule because it weren't LL(1)

We applied left factoring to this rule because it wasn't LL(1)

NARRP <param> ::= <decl\_prefix> <arrdecl>

NARRC <param> ::= const <decl\_prefix> <arrdecl>

Special <funcbody> ::= <locals> begin <stats> end

Special <locals> ::= <dlist> | ε

NDLIST <dlist> ::= <decl> <opt\_dlist>

Special <opt\_dlist> ::= , <dlist> | ε

Special <decl> ::= <decl\_prefix> <sdecl> | <decl\_prefix> <arrdecl>

Special <stype> ::= integer | real | Boolean

NSTATS <stats> ::= <stat>; <opt\_stats> | <strstat> <opt\_stats>

Special <opt\_stats> ::= <stats> | ε

Special <strstat> ::= <forstat> | <ifstat>

Special <stat> ::= <reptstat> | <asgnstat> | <iostat>

Special <stat> ::= <callstat> | <returnstat>

NFORL <forstat> ::= for (<asgnlist> ; <bool>) <stats> end

NREPT <reptstat> ::= repeat (<asgnlist>) <stats> until <bool>

Special <asgnlist> ::= <alist> | ε

NASGNS <alist> ::= <asgnstat> <opt\_alist>

Special <opt\_alist> ::= , <alist> | ε

NIFTH <ifstat> ::= if (<bool>) <stats> <opt\_else> end

NIFTE <opt\_else> ::= else <stats> | ε

Special <asgnstat> ::= <var> <asgnop> <bool>

NASGN <asgnop> ::= =

NPLEQ <asgnop> ::= +=

NMNEQ <asgnop> ::= -=

NSTEQ <asgnop> ::= \*=

NDVEQ <asgnop> ::= /=

NINPUT <iostat> ::= In >> <vlist>

NOUTP <iostat> ::= Out << <prlist> <opt\_line>

Special <opt\_line> ::= ε

NOUTL <opt\_line> ::= <<Line

NOUTL <iostat> ::= Out << Line

We applied left factoring to these rules because it weren't LL(1)

We applied left factoring to these rules because it weren't LL(1)

We applied left factoring to this rule because it wasn't LL(1)

We applied left factoring to these rules because they weren't LL(1)

We applied left factoring to these rules because they weren't LL(1).

Note that if <opt\_line> is <<Line, the node is NOUTL, not NOUTP

NCALL      <callstat> ::= <id> (<elist>) | <id> ()

NRETN      <returnstat> ::= return void | return <expr>

NVLIST      <vlist> ::= <var> <opt\_vlist>

Special      <opt\_vlist> ::= , <vlist> | ε

We applied left factoring to NVLIST because it wasn't LL(1)

Special      <var> ::= <id> <opt\_expr>

NSIMV      <opt\_expr> ::= ε

Special      <opt\_expr> ::= [<expr>] <opt\_id>

NAELT      <opt\_id> ::= ε

NARRV      <opt\_id> ::= . <id>

We applied left factoring to these rules because they weren't LL(1).

Note that if <opt\_line> is <<Line, the node is NOUTL, not NOUTP

NEXPL      <elist> ::= <bool> <opt\_elist>

Special      <opt\_elist> ::= , <elist> | ε

NBOOL      <bool> ::= <rel> <bool'>

Special      <bool'> ::= <logop> <rel> <bool'> | ε

We applied left factoring to NEXPL because it wasn't LL(1)

We changed NBOOL because it was left recursive

NNOT      <rel> ::= ! <expr> <relop> <expr>

Special      <rel> ::= <expr> <opt\_rel>

Special      <opt\_rel> ::= <relop> <expr> | ε

We applied left factoring to this rule because it wasn't LL(1)

NAND      <logop> ::= &&

NOR      <logop> ::= ||

NXOR      <logop> ::= &|

NEQL      <relop> ::= ==

NNEQ      <relop> ::= !=

NGRT      <relop> ::= >

NLSS      <relop> ::= <

NLEQ      <relop> ::= <=

NGEQ      <relop> ::= >=

Special      <expr> ::= <term> <expr'>

NADD      <expr'> ::= +<term> <expr'>

NSUB      <expr'> ::= - <term> <expr'>

Special      <expr'> ::= ε

Special      <term> ::= <fact> <term'>

NMUL      <term'> ::= \*<fact> <term'>

We changed these rules because they were left recursive

We changed these rules because they were left recursive

NDIV  $\langle \text{term}' \rangle ::= / \langle \text{fact} \rangle \langle \text{term}' \rangle$

NMOD  $\langle \text{term}' \rangle ::= \% \langle \text{fact} \rangle \langle \text{term}' \rangle$

Special  $\langle \text{term}' \rangle ::= \epsilon$

Special  $\langle \text{fact} \rangle ::= \langle \text{exponent} \rangle \langle \text{fact}' \rangle$

NPOW  $\langle \text{fact}' \rangle ::= ^ \langle \text{exponent} \rangle \langle \text{fact}' \rangle \mid \epsilon$

Special  $\langle \text{exponent} \rangle ::= \langle \text{var} \rangle$

NILIT  $\langle \text{exponent} \rangle ::= \langle \text{intlitt} \rangle$

NFLIT  $\langle \text{exponent} \rangle ::= \langle \text{reallitt} \rangle$

Special  $\langle \text{exponent} \rangle ::= \langle \text{fncall} \rangle$

NTRUE  $\langle \text{exponent} \rangle ::= \text{true}$

NFALS  $\langle \text{exponent} \rangle ::= \text{false}$

Special  $\langle \text{exponent} \rangle ::= (\langle \text{bool} \rangle)$

NFCALL  $\langle \text{fncall} \rangle ::= \langle \text{id} \rangle (\langle \text{elist} \rangle) \mid \langle \text{id} \rangle ()$

NPRLST  $\langle \text{prlist} \rangle ::= \langle \text{printitem} \rangle \langle \text{opt\_prlist} \rangle$

Special  $\langle \text{opt\_prlist} \rangle ::= , \langle \text{prlist} \rangle \mid \epsilon$

Special  $\langle \text{printitem} \rangle ::= \langle \text{expr} \rangle$

NSTRG  $\langle \text{printitem} \rangle ::= \langle \text{string} \rangle$

We changed this rule  
because it was left recursive

We applied left  
factoring to this rule  
because it wasn't LL(1)