CD23 Programming Language LL(1) Grammar

<param> ::= <decl_prefix> <sdecl>

NSIMP

NPROG NGLOB <globals> ::= <consts> <types> <arrays> <consts> ::- constants <initlist> | ε Special NILIST <initlist> ::= <init> <opt_initlist> We applied left factoring to this rule because it wasn't LL(1) Special <opt_initlist> ::= , <initlist> | ε NINIT <init> ::= <id> is <expr> Special <types> ::= types <typelist> | ε <arrays> ::= arrays <arrdecls> | ε Special **NFUNCS** <funcs> ::= <func> <funcs $> | \epsilon$ <mainbody> ::= main <slist> begin <stats> end CD23 <id> **NMAIN NSDLST** <slist> ::= <decl prefix> <sdecl> <opt slist> We applied left factoring to this rule because it wasn't LL(1) Special <opt_slist> ::= , <slist> | ε NTYPEL <typelist> ::= <type> <opt typelist> Special <opt_typelist> ::= <typelist> | ε NRTYPE <type> ::= <structed> is <fields> end NATYPE <type> ::= <typeid> is array [<expr>] of <structid> end We applied left <fields> ::= <decl_prefix> <sdecl> <opt_fields> NFLIST factoring to these rule because it <opt_fields> ::= , <fields> | ε Special weren't LL(1) Special <decl_prefix> ::= <id> : <sdecl> ::= <stype> NSDECL We applied left NALIST <arrdecls> ::= <decl_prefix> <arrdecl> <opt_arrdecl> factoring to this rule because it wasn't LL(1) Special <opt_arrdecl> ::= , <arrdecls> | ε NARRD <arrdecl> ::= <typeid> NFUND <func> ::= func <id> (<plist>):<rtype> <funcbody> Special <rtype> ::= <stype> | void Special **NPLIST** <params> ::= <param> <opt params> Special <opt_params> ::= , <params> | ε

NARRP	<pre><param/> ::= <decl_prefix> <arrdecl></arrdecl></decl_prefix></pre>		We	applied left factoring to	
NARRC	<pre><param/> ::= const <decl_prefix> <arrdecl></arrdecl></decl_prefix></pre>	these rules because it weren't			
Special	<pre><funcbody> ::= <locals> begin <stats> end</stats></locals></funcbody></pre> LL(1)				
Special	<locals> ::= <dlist> ε</dlist></locals>	Г	Ma and	ind left factoring to	
NDLIST	<dlist> ::= <decl> <opt_dlist></opt_dlist></decl></dlist>	I	We applied left factoring to these rules because it weren't		
Special	<pre><opt_dlist> ::= , <dlist> ε</dlist></opt_dlist></pre>	L	LL(1)		
Special	<decl> ::= <decl_prefix> <sdecl> <decl_< td=""><td>prefix> <a< td=""><td>arrdecl></td><td></td></a<></td></decl_<></sdecl></decl_prefix></decl>	prefix> <a< td=""><td>arrdecl></td><td></td></a<>	arrdecl>		
Special	<stype> ::= integer real Boolean</stype>			We applied left	
NSTATS	<stats> ::= <stat>; <opt_stats> <strstat></strstat></opt_stats></stat></stats>	<opt_stat< td=""><td colspan="3">copt_stats> factoring to this</td></opt_stat<>	copt_stats> factoring to this		
Special	<pre><opt_stats> ::= <stats> ε</stats></opt_stats></pre>			rule because it wasn't LL(1)	
Special	<strstat> ::= <forstat> <ifstat></ifstat></forstat></strstat>				
Special	<stat> ::= <reptstat> <asngnstat> <iost< td=""><td>at></td><td></td><td></td></iost<></asngnstat></reptstat></stat>	at>			
Special	<stat> ::= <callstat> <returnstat></returnstat></callstat></stat>				
NFORL	<forstat> ::= for (<asgnlist> ; <bool>) <stats> end</stats></bool></asgnlist></forstat>				
NREPT	<repstat> ::= repeat (<asgnlist>) <stats> u</stats></asgnlist></repstat>	ntil <bool< td=""><td>></td><td></td></bool<>	>		
Special	<asgnlist> ::= <alist> ε</alist></asgnlist>				
NASGNS	<alist> ::= <asgnstat> <opt_alist></opt_alist></asgnstat></alist>		We a	applied left	
Special	<pre><opt_alist> ::= , <alist> ε factoring to these rules because</alist></opt_alist></pre>				
NIFTH	<ifstat> ::= if (<bool>) <stats> <opt_else></opt_else></stats></bool></ifstat>	<mark>end</mark>		weren't LL(1)	
NIFTE	<opt_else> ::= else <stats> ε</stats></opt_else>				
Special	<asgnstat> ::= <var> <asgnop> <bool></bool></asgnop></var></asgnstat>				
NASGN	<asgnop> ::= =</asgnop>				
NPLEQ	<asgnop> ::= +=</asgnop>				
NMNEQ	<asgnop> ::= -=</asgnop>				
NSTEQ	<asgnop> ::= *=</asgnop>				
NDVEQ	<asgnop> /=</asgnop>				
NINPUT	<iostat> ::= In >> <vlist></vlist></iostat>				
NOUTP	<iostat> ::= Out << <pri>iostat> ::= Out << <pri>iostat> copt_line></pri></pri></iostat>	We ann	lied left	: factoring to these	
Special	<opt_line> ::= ε</opt_line>	1		they weren't LL(1).	
NOUTL	<opt_line> ::= <<line< td=""><td>1</td><td colspan="3">Note that if <opt_line> is <<line,< td=""></line,<></opt_line></td></line<></opt_line>	1	Note that if <opt_line> is <<line,< td=""></line,<></opt_line>		
NOUTL	<iostat> ::= Out << Line</iostat>	the node is NOUTL, not NOUTP			

NCALL	<callstat> ::= <id> (<elist>) <id> ()</id></elist></id></callstat>		
NRETN	<returnstat> ::= return void return <ex< th=""><th></th></ex<></returnstat>		
NVLIST	<vlist> ::= <var> <opt_vlist></opt_vlist></var></vlist>	We applied left factoring to NVLIST	
Special	<opt_vlist> ::= , <vlist> ε</vlist></opt_vlist>	because it wasn't LL(1)	
Special	<var> ::= <id> <opt_expr></opt_expr></id></var>		
NSIMV	<opt_expr> ::= ε</opt_expr>	We applied left factoring to the	
Special	<pre><opt_expr> ::= [<expr>] <opt_id></opt_id></expr></opt_expr></pre>	rules because they weren't LL(Note that if <opt_line> is <<lir is="" node="" not="" nout<="" noutl,="" td="" the=""></lir></opt_line>	
NAELT	<opt_id> ::= ε</opt_id>		
NARRV	<opt_id> ::= . <id></id></opt_id>		
NEXPL	<elist> ::= <bool> <opt_elist></opt_elist></bool></elist>	N/o applied left feetawing to	
Special	<opt_elist> ::= , <elist> ε</elist></opt_elist>	We applied left factoring to NEXPL because it wasn't LL(1) We changed NBOOL because it was left recursive	
NBOOL	<bool> ::= <rel> <bool'></bool'></rel></bool>		
Special	<bool'> ::= <logop> <rel> <bool'> ε</bool'></rel></logop></bool'>		
NNOT	<rel> ::= ! <expr> <relop> <expr></expr></relop></expr></rel>		
<mark>Special</mark>	<rel> ::= <expr> <opt_rel></opt_rel></expr></rel>	We applied left factoring to this rule	
Special	<opt_rel> ::= <relop> <expr> ε</expr></relop></opt_rel>	because it wasn't LL(1)	
NAND	<logop> ::= &&</logop>		
NOR	<logop> ::= </logop>		
NXOR	<logop> ::= & </logop>		
NEQL	<relop> ::= ==</relop>		
NNEQ	<relop> ::= !=</relop>		
NGRT	<relop> ::= ></relop>		
NLSS	<relop> ::= <</relop>		
NUEO			
NLEQ	<relop> ::= <=</relop>		
NGEQ	<relop> ::= <= <relop> ::= >=</relop></relop>		
	·		
NGEQ	<relop> ::= >=</relop>	We changed these rules because they were left	
NGEQ Special	<pre> <relop> ::= >= <expr> ::= <term> <expr'> </expr'></term></expr></relop></pre>		
NGEQ Special NADD	<pre><relop> ::= >=</relop></pre>	because they were left	
NGEQ Special NADD NSUB	<pre><relop> ::= >=</relop></pre>	because they were left	

NDIV <term'> ::= /<fact> <term'>

NMOD <term'> ::= % <fact> <term'>

Special <term'> ::= ε

Special <fact> ::= <exponent> <fact'>

NPOW <fact'> ::= ^ <exponent> <fact'> | ε

Special <exponent> ::= <var>

NILIT <exponent> ::= <intlit>

NFLIT <exponent> ::= <reallit>

Special <exponent> ::= <fncall>

NTRUE <exponent> ::= true

NFALS <exponent> ::= false

Special <exponent> ::= (<bool>)

NFCALL <fncall> ::= <id> (<elist>) | <id> ()

NPRLST <pri>vprlist> ::= <printitem> <opt_prlist>

Special <opt_prlist> ::= , <prlist> | ε

Special <printitem> ::= <expr>

NSTRG <printitem> ::= <string>

We changed this rule because it was left recursive

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