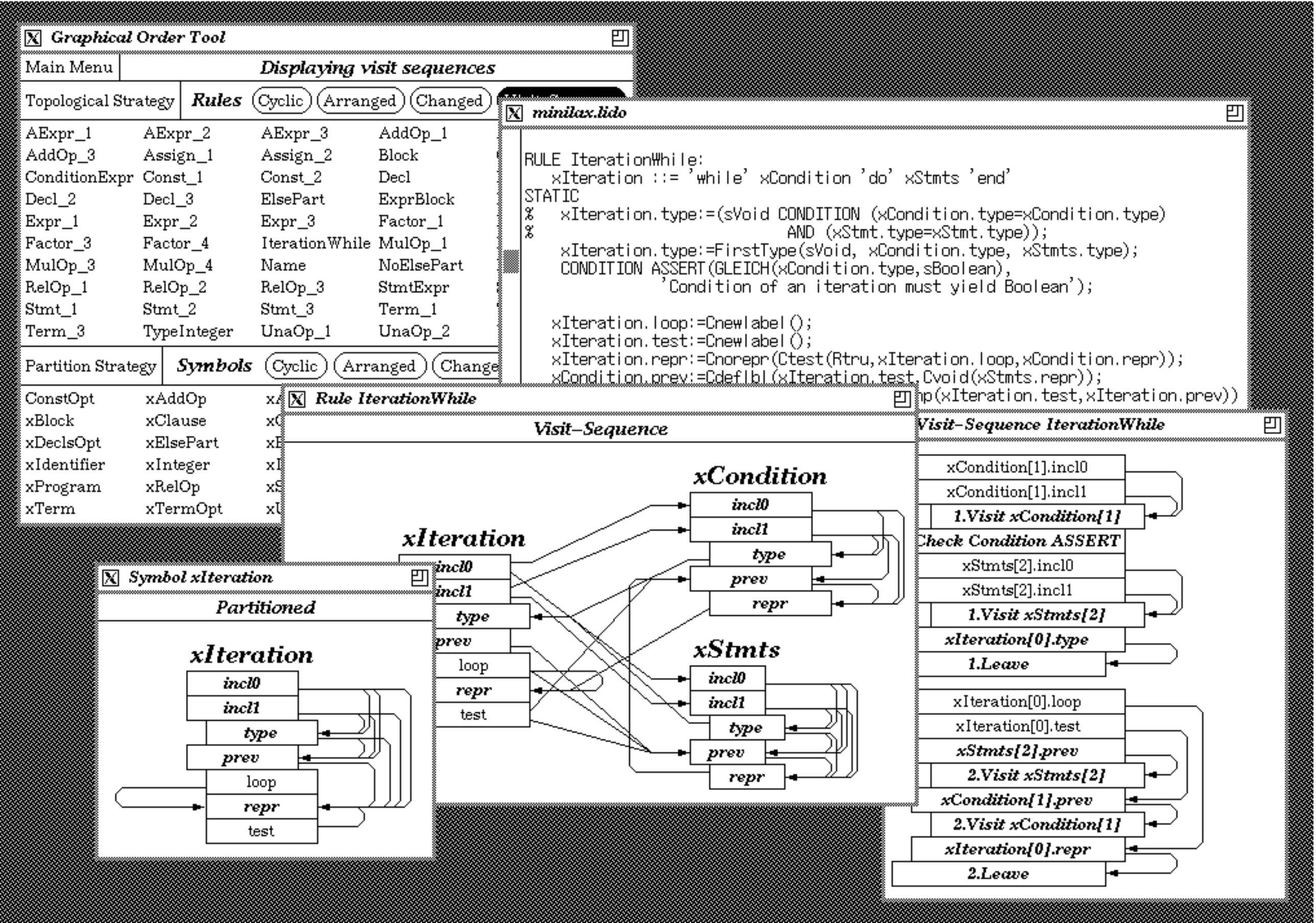
Graphical Order Tool
Main Menu
Displaying visit sequences
Topological Strategy
Rules
Cyclic
Arranged
Changed
AExpr_1 AExpr_2 AExpr_3 AddOp_1
AddOp_3 Assign_1 Assign_2 Block
ConditionExpr Const_1 Const_2 Decl
Decl_2 Decl_3 ElsePart ExprBlock
Expr_1 Expr_2 Expr_3 Factor_1
Factor_3 Factor_4 IterationWhile MulOp_1
MulOp_3 MulOp_4 Name NoElsePart
RelOp_1 RelOp_2 RelOp_3 StmtExpr
Stmt_1 Stmt_2 Stmt_3 Term_1
Term_3 TypeInteger UnaOp_1 UnaOp_2
Partition Strategy
Symbols
Cyclic
Arranged
ConstOpt xAddOp
xBlock xClause
xDeclsOpt xElsePart
xIdentifier xInteger
xProgram xRelOp
xTerm xTermOpt
minilax.lido
RULE IterationWhile:
xIteration ::= 'while' xCondition 'do' xStmts 'end'
STATIC
% xIteration.type:=(sVoid CONDITION (xCondition.type=xCondition.type)
% AND (xStmt.type=xStmt.type));
xIteration.type:=FirstType(sVoid, xCondition.type, xStmts.type);
CONDITION ASSERT(GLEICH(xCondition.type,sBoolean),
'Condition of an iteration must yield Boolean');
xIteration.loop:=Cnewlabel();
xIteration.test:=Cnewlabel();
xIteration.repr:=Cnorepr(Ctest(Rtru,xIteration.loop,xCondition.repr));
xCondition.prev:=Cdeflbl(xIteration.test,Cvoid(xStmts.repr));
Rule IterationWhile
Visit-Sequence
xCondition
incl0
incl1
type
prev
repr
xIteration
incl0
incl1
type
prev
loop
repr
test
xStmts
incl0
incl1
type
prev
repr
Symbol xIteration
Partitioned
xIteration
incl0
incl1
type
prev
loop
repr
test
Visit-Sequence IterationWhile
xCondition[1].incl0
xCondition[1].incl1
1.Visit xCondition[1]
Check Condition ASSERT
xStmts[2].incl0
xStmts[2].incl1
1.Visit xStmts[2]
xIteration[0].type
1.Leave
xIteration[0].loop
xIteration[0].test
xStmts[2].prev
2.Visit xStmts[2]
xCondition[1].prev
2.Visit xCondition[1]
xIteration[0].repr
2.Leave