

Architectural Design Recovery Using Source Transformation

*James R. Cordy
Kevin A. Schneider*

Queen's University / Legasys Corporation

*Workshop on Software Architecture
CASE'95, Toronto
July 1995*

Motivation

- necessary to handle legacy systems
- can't trust anything but source code

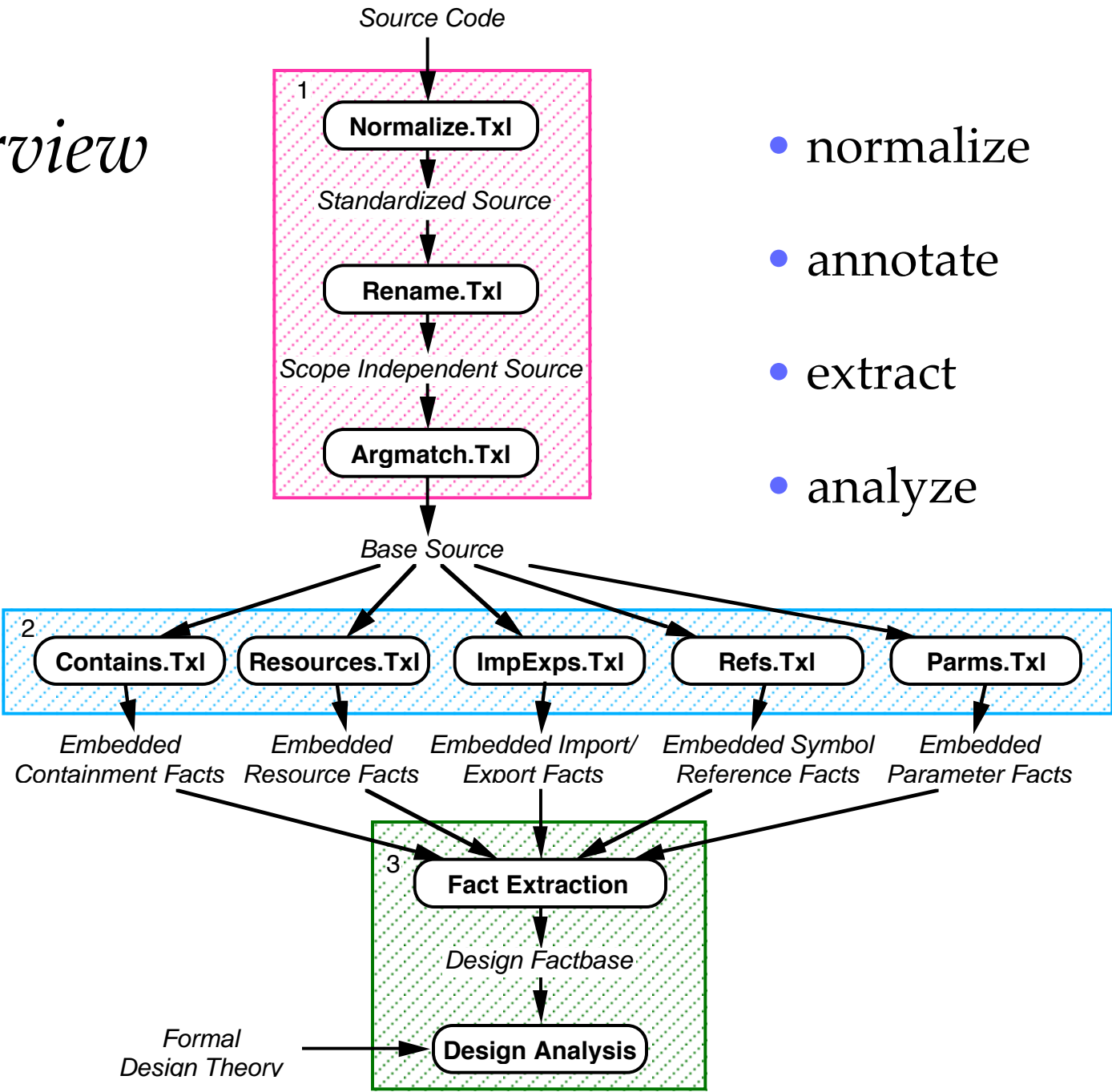
Method

- design ER schema for architecture theory
- identify minimum basis relations
- model legacy system as instance of the theory
- derive basis relations from source using conditional structural patterns on source
- query / visualize using strong transitive query engines to get higher level abstract relations

Representation

- ER factbase as Prolog source facts
- ASCII \Rightarrow easy represent, debug, portable
- each fact independent \Rightarrow extensible
- source \Rightarrow can use source transformation to derive & query

Overview



An Example

- actual steps in a design recovery from source
- example is source for a simple programming language interpreter (μ Turing) written in the Turing Plus language

Design Recovery Demo

Design Recovery Using Source Transformation

K.A. Schneider & J.R. Cordy
Advanced Software Design Technology Project
Software Technology Laboratory, Queen's University
July 1992

Step 0. Augment - add implicit library definitions to source

[1] 9557
[Hit return to

Step 0. Augment - add implicit library definitions to source

```
% Signatures of Turing+ implicit functions
external function chr (i : int) : string
external function ord (c : string) : int
external function strint (s : string) : int
external function intstr (i : int, w : int) : string
external function streal (s : string) : real
external function realstr (r : real, w : int) : string
external function length (s : string) : int
external function eof (i : int) : boolean
external function index (s : string, s2 : string) : int
external function max (i : int, i2 : int) : int
external function min (i : int, i2 : int) : int
external function repeat (s : string, i : int) : string
external function size (i : int) : int

% uTuring U1.00
% Copyright (c) 1984 James R. Cordy
put "/e[;H/e[JMicro-Turing Monitor U1.00"
put skip, "Wait ..."

% include "limits.i"
% uTuring Limits U1.00
% Copyright (c) 1984 James R. Cordy
"microturing.bsource" 3427 lines, 76936 characters
```

Design Recovery Demo

Step 1. Normalize - separate declarations, imports, exports; expand binds; etc.

TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston

Compiling Txl/normalize.Txl ...

Parsing microtuning.bsource ...

Transforming ...

[unwindVarList]

[unwindBindList]

[unwindParmList]

[removeForwardP

[removeForwardP

[mergeModuleStu

[placeImportsAn

[sortSubprogram

[mergeProcedure

[mergeFunctionS

[mergeForwardPr

[mergeForwardFu

[unparImportLis

[unwindImportLi

[unparExportLis

[unwindExportLi

Step 1. Normalize - separate declarations, imports, exports; expand binds; etc

```
    tprocedure, tend, tendif, tendloop, tendfor, telsif, telseif, telse,
    tinvariant, texit, texitwhen, tarray, tof, tint, tboolean, tstring,
    tthen, twhen, tnot, tdiv, tmod, tand, tor, ttrue, tfalse, tchr, tord,
    tlength)
```

```
  for kw : 1..nkeywords
    var ix : int
    enterident (ix, keyword (kw))
    identtable (ix).token := keywordtoken (kw)
```

```
  end for
```

```
  import error
```

```
  export identindex
```

```
  export identtoken
```

```
  export identident
```

```
  export enterident
```

```
  export reinitialize
```

```
end ident
```

```
module scan
```

```
  const cblank := 0
```

```
  const cletter := 1
```

```
  const cdigit := 2
```

```
  const cspecialchar := 3
```

```
  const cquote := 4
```


Design Recovery Demo

```
Step 2. Rename - name all entities to reflect scope
TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston
Compiling Tx1/rename.Tx1 ...
Parsing microturing.normalized ...
Transforming ...
[nameModules 'MAIN]
[nameForwardProcedures 'MAIN]
[nameProcedures 'MAIN]
```

Step 2. Rename - name all entities to reflect scope

```
[nameForwardFu
[nameFunctions
[nameExternalFu
[nameExternalFu
[nameExternalPr
[nameExternalPr
[nameExportedCo
[nameModuleVars
[nameProcedure
[nameFunctionU
[nameExternalU
[nameExternalCo
[nameVariables
[nameConstants

procedure MAIN_error (MAIN_error_msg : string)
    if not MAIN_currentline_1.MAIN_error then
        MAIN_currentline_1.MAIN_error := true
        MAIN_currentline_1.tokens (MAIN_currentptr_1).MAIN_error := true
        MAIN_currenterror_1 := MAIN_error_msg
    end if
end MAIN_error

module MAIN_ident
    var MAIN_ident_identtable_1 : array 1..MAIN_maxidents_1 of
        record
            ident : string (MAIN_maxstringlength_1)
            token : int
        end record
    for MAIN_ident_ix_2 : 1..MAIN_maxidents_1
        MAIN_ident_identtable_1 (MAIN_ident_ix_2).token := MAIN_tidentifier_1
        MAIN_ident_identtable_1 (MAIN_ident_ix_2).ident := ""
    end for

    function MAIN_ident_identindex (MAIN_ident_identindex_ident : string) : int
        for MAIN_ident_identindex_ix_1 : 1..MAIN_maxidents_1
            if MAIN_ident_identtable_1 (MAIN_ident_identindex_ix_1).
                MAIN_ident_identindex_ident = MAIN_ident_identindex_ident then
```

Design Recovery Demo

Step 3. Argmatch - annotate arguments with formal var/const attributes
(Yields standardized base source)

TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston

Compiling Tx1/argmatch.Tx1 ...

Parsing microturing.renamed ...

Transforming ...

[attachProcFormalNamesToArguments C]

[attachFuncFormalNamesToArguments C]

19.0u 0.0s 0:20

Step 3. Argmatch - annotate arguments with formal var/const attributes

[1] 9582

[Hit return to

```
MAIN_tif_1, MAIN_tloop_1, MAIN_tfor_1, MAIN_tput_1, MAIN_tget_1,  
MAIN_tprocedure_1, MAIN_tend_1, MAIN_tendif_1, MAIN_tendloop_1,  
MAIN_tendfor_1, MAIN_telsif_1, MAIN_telseif_1, MAIN_telse_1,  
MAIN_tinvariant_1, MAIN_texit_1, MAIN_texitwhen_1, MAIN_tarray_1,  
MAIN_tof_1, MAIN_tint_1, MAIN_tboolean_1, MAIN_tstring_1, MAIN_tthen_1  
, MAIN_twhen_1, MAIN_tnot_1, MAIN_tdiv_1, MAIN_tmod_1, MAIN_tand_1,  
MAIN_tor_1, MAIN_ttrue_1, MAIN_tfalse_1, MAIN_tchr_1, MAIN_tord_1,  
MAIN_tlength_1)
```

```
for MAIN_ident_kw_1 : 1..MAIN_ident_nkeywords_1
```

```
  var MAIN_ident_ix_1 : int
```

```
  MAIN_ident_enterident (MAIN_ident_ix_1 : var
```

```
    MAIN_ident_enterident_index_, MAIN_ident_keyword_1 (MAIN_ident_kw_1) :  
    MAIN_ident_enterident_ident)
```

```
  MAIN_ident_idenntable_1 (MAIN_ident_ix_1).token :=
```

```
  MAIN_ident_keywordtoken_1 (MAIN_ident_kw_1)
```

```
end for
```

```
import MAIN_error
```

```
export MAIN_ident_identindex
```

```
export MAIN_ident_idenntoken
```

```
export MAIN_ident_identident
```

```
export MAIN_ident_enterident
```

```
export MAIN_ident_reinitialize
```

```
end MAIN_ident
```

```
"microturing.base" 3375 lines, 148623 characters
```

Design Recovery Demo

```
Step 4a. Scope analysis - infer containment relation facts
TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston
Compiling Tx1/rename.Tx1 ...
Parsing microturing.normalized ...
Transforming ...
[nameModules 'MAIN]
[nameForwardProcedures 'MAIN]
[nameProcedures 'MAIN]
```

Step 4a. Scope analysis - infer containment relation facts

```
end MAIN_error
module MAIN_ident
  $ contains (MAIN, MAIN_ident) $
  var MAIN_ident_identtable_1 : array 1..MAIN_maxidents_1 of
    record
      ident : string (MAIN_maxstringlength_1)
      token : int
    end record
  $ contains (MAIN_ident, MAIN_ident_identtable_1) $
  for MAIN_ident_ix_2 : 1..MAIN_maxidents_1
    $ contains (MAIN_ident, MAIN_ident_ix_2) $
    MAIN_ident_identtable_1 (MAIN_ident_ix_2).token := MAIN_tidentifier_1
    MAIN_ident_identtable_1 (MAIN_ident_ix_2).ident := ""
  end for
  function MAIN_ident_identindex (MAIN_ident_identindex_ident
```

Design Recovery Demo

```
Step 4b. Resource identification - infer resource set facts
TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston
Compiling Tx1/resources.Tx1 ...
Parsing microtuning.base ...
Transforming ...
[varResources]
[conResources]
[pconResources]
[forconResource
[handlerconReso
[moduleResource
[procedureResou
[functionResour
[externalProced
[externalFuncti
23.0u 0.0s 0:24
[1] 9646
[Hit return to
```

Step 4b. Resource identification - infer resource set facts

```
        MAIN_currenterror_1 := MAIN_error_msg
    end if
end MAIN_error

module MAIN_ident

    $ module (MAIN_ident) $

    var MAIN_ident_identtable_1 : array 1..MAIN_maxidents_1 of
        record
            ident : string (MAIN_maxstringlength_1)
            token : int
        end record

    $ variable (MAIN_ident_identtable_1) $

    for MAIN_ident_ix_2 : 1..MAIN_maxidents_1

        $ constant (MAIN_ident_ix_2) $

        MAIN_ident_identtable_1 (MAIN_ident_ix_2).token := MAIN_tidentifier_1
        MAIN_ident_identtable_1 (MAIN_ident_ix_2).ident := ""
    end for
```

Design Recovery Demo

```
Step 4c. Import/export analysis - infer module import/export facts
TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston
Compiling Tx1/impexps.Tx1 ...
Parsing microturing.base ...
Transforming ...
[importsAndExportsInModules]
[importsInProcedures]
[importsInFunctions]
9.0u 0.0s 0:11 81% 0+0k 0+0i 0ef+0w
[1] 9659
[Hit return to
```

Step 4c. Import/export analysis - infer module import/export facts

```
    MAIN_ident_enterident (MAIN_ident_ix_1 : var
      MAIN_ident_enterident_index_, MAIN_ident_keyword_1 (MAIN_ident_kw_1) :
      MAIN_ident_enterident_ident)
    MAIN_ident_identtable_1 (MAIN_ident_ix_1).token :=
      MAIN_ident_keywordtoken_1 (MAIN_ident_kw_1)
  end for

  import (MAIN_error)

  $ imports (MAIN_ident, MAIN_error) $

  export (MAIN_ident_identindex)

  $ exports (MAIN_ident, MAIN_ident_identindex) $

  export (MAIN_ident_identtoken)

  $ exports (MAIN_ident, MAIN_ident_identtoken) $

  export (MAIN_ident_identident)

  $ exports (MAIN_ident, MAIN_ident_identident) $
```

Design Recovery Demo

```
Step 4d. Reference analysis - infer use relations
TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston
Compiling Tx1/references.Tx1 ...
Parsing microtuning.base ...
Transforming ...
[embedProcCalls 'MAIN]
[embedFuncCalls 'MAIN AllFuncs]
[embedVarParmRefs 'MAIN]
[embedPutRefs 'MAIN]
[embedGetRefs 'MAIN]
[getModuleRefs
[getProcedureRe
[getFunctionRef
17.0u 0.0s 0:20
[1] 9666
[Hit return to
```

Step 4d. Reference analysis - infer use relations

```
        ).MAIN_error := true
        $ write_ref (MAIN_error, MAIN_currenterror_1) $
        :=
        $ read_ref (MAIN_error, MAIN_error_msg) $
    end if
end MAIN_error

module MAIN_ident
  var MAIN_ident_identtable_1 : array 1..
  $ read_ref (MAIN_ident, MAIN_maxidents_1) $
  of
    record
      ident : string (
        $ read_ref (MAIN_ident, MAIN_maxstringlength_1) $
      )
      token : int
    end record
  for MAIN_ident_ix_2 : 1..
  $ read_ref (MAIN_ident, MAIN_maxidents_1) $

    $ write_ref (MAIN_ident, MAIN_ident_identtable_1) $
```

Design Recovery Demo

Step 4e. Parameterization analysis - infer parameterization relations
TXL v7.7a8 (23.12/94) (c)1988-94 Queen's University at Kingston
Compiling Txl/parameters.Txl ...
Parsing microturing.base ...
Transforming ...
[varParameterFacts]
[constParameterFacts]
[varArgumentFacts]
[constArgumentFacts]

9.0u 0.0s 0:11
[1] 9679
[Hit return to

```
Step 4e. Parameterization analysis - infer parameterization relations
```

```
procedure MAIN_ident_enterident (  
  $ var_parameter (MAIN_ident_enterident_index_) $  
  ,  
  $ const_parameter (MAIN_ident_enterident_ident) $  
  )  
  pre LIBRARY_length (MAIN_ident_enterident_ident) <=  
    MAIN_maxstringlength_1  
  
  MAIN_ident_enterident_index_ := MAIN_ident_index (  
    $ const_argument (MAIN_ident_enterident_ident,  
      MAIN_ident_index_ident) $  
  )  
  if MAIN_ident_enterident_index_ = 0 then  
    for MAIN_ident_enterident_ix_1 : 1..MAIN_maxidents_1  
      if MAIN_ident_idtable_1 (MAIN_ident_enterident_ix_1).  
        MAIN_ident_enterident_ident = "" then  
        MAIN_ident_enterident_index_ := MAIN_ident_enterident_ix_1  
        MAIN_ident_idtable_1 (MAIN_ident_enterident_ix_1).  
          MAIN_ident_enterident_ident := MAIN_ident_enterident_ident  
        exit  
      end if  
    end for  
  end for
```

Design Recovery Demo

Step 5. Extract and merge fact sets
[1] 9693

The result of these transforms is a raw factbase from which more useful design relations, such as 'partof', 'uses', etc. are easily derived. For example,

```
uses(X,Y) :- read_ref(X,Y).  
uses(X,Y) :- write_ref(X,Y).
```

Step 5. Extract and merge fact sets

[Hit return to

```
calls(MAIN clear,ident).  
calls(MAIN clear,runstack).  
calls(MAIN crmode,LIBRARY gtty).  
calls(MAIN crmode,LIBRARY stty).  
calls(MAIN echo,LIBRARY gtty).  
calls(MAIN echo,LIBRARY stty).  
calls(MAIN edit,LIBRARY intstr).  
calls(MAIN edit,LIBRARY length).  
calls(MAIN edit,LIBRARY min).  
calls(MAIN edit,LIBRARY ord).  
calls(MAIN edit,LIBRARY repeat).  
calls(MAIN edit,MAIN crmode).  
calls(MAIN edit,MAIN echo).  
calls(MAIN edit,MAIN enter).  
calls(MAIN edit,MAIN execute).  
calls(MAIN edit,MAIN nocrmode).  
calls(MAIN edit,MAIN noecho).  
calls(MAIN edit,scan).  
calls(MAIN enter,LIBRARY intstr).  
calls(MAIN enter,LIBRARY length).  
calls(MAIN enter,LIBRARY ord).  
calls(MAIN enter,parse).  
calls(MAIN enter,scan).  
"microturing.facts" 2543 lines, 113576 characters
```


Step 5. Extract and Merge Fact Sets

```
calls(MAIN exec execarraytype,MAIN exec execsimplotype).
calls(MAIN exec execassert,MAIN exec execerror).
calls(MAIN exec execassert,MAIN exec execexpression).
calls(MAIN exec execassert,runstack).
calls(MAIN exec execassignment,MAIN exec execerror).
calls(MAIN exec execassignment,MAIN exec execexpression).
:q/enterident/p
calls(MAIN ident enterident,LIBRARY length).
calls(MAIN ident enterident,MAIN error).
calls(MAIN ident enterident,MAIN ident identindex).
calls(MAIN ident,MAIN ident enterident).
const_argument(MAIN ident enterident ident,MAIN ident identindex ident).
const_argument(MAIN ident keyword,MAIN ident enterident ident).
const_parameter(MAIN ident enterident ident).
constant(MAIN ident enterident ix).
contains(MAIN ident enterident,MAIN ident enterident ident).
contains(MAIN ident enterident,MAIN ident enterident index_).
contains(MAIN ident enterident,MAIN ident enterident ix).
contains(MAIN ident,MAIN ident enterident).
exports(MAIN ident,MAIN ident enterident).
procedure(MAIN ident enterident).
read_ref(MAIN ident enterident,MAIN ident enterident ident).
read_ref(MAIN ident enterident,MAIN ident enterident index_).
read_ref(MAIN ident enterident,MAIN ident enterident ix).
read_ref(MAIN ident enterident,MAIN ident identtable).
read_ref(MAIN ident enterident,MAIN maxidents).
read_ref(MAIN ident enterident,MAIN maxstringlength).
var_argument(MAIN ident ix,MAIN ident enterident index_).
var_parameter(MAIN ident enterident index_).
write_ref(MAIN ident enterident,MAIN ident enterident index_).
write_ref(MAIN ident enterident,MAIN ident identtable).
[Hit return to continue]
```

Query Result

```
TXL Pro 1.0d5 (20.4/95) Copyright 1995 by Legasys Corp. & James R. Condy  
Compiling analyze.Txl ...  
Parsing factbase.analyze ...  
Transforming ...
```

```
1. 'enterident' is a procedure  
   contained in module 'ident'  
  
   contains const_parameter 'ident'  
   contains var_parameter 'index_'  
   contains constant 'ix'  
  
   called by module 'ident'  
   called by procedure 'scanline' of module 'scan'  
  
   called indirectly by procedure 'enter'  
   called indirectly by procedure 'edit'  
   called indirectly by program 'MAIN'  
   called indirectly by procedure 'execute'  
   called indirectly by procedure 'load'  
  
   calls function 'length' of library 'LIBRARY'  
   calls procedure 'error'  
   calls function 'identindex' of module 'ident'  
  
   reads const_parameter 'ident' of procedure 'enterident'  
   reads var_parameter 'index_' of procedure 'enterident'  
   reads constant 'ix' of procedure 'enterident'  
   reads variable 'identtable' of module 'ident'  
   reads pervasive_constant 'maxidents'  
   reads pervasive_constant 'maxstringlength'  
:
```

Query Result

```
writes var_parameter 'index_' of procedure 'enterident'  
writes variable 'identtable' of module 'ident'  
  
indirectly reads variable 'currentline'  
indirectly reads variable 'currentptr'  
indirectly reads const_parameter 'msg' of procedure 'error'  
indirectly reads const_parameter 'ident' of function 'identindex'  
indirectly reads constant 'ix' of function 'identindex'  
indirectly reads variable 'identtable' of module 'ident'  
indirectly reads pervasive_constant 'maxidents'  
  
indirectly writes variable 'currenterror'  
indirectly writes variable 'currentline'  
  
reads as value parameter constant 'keyword' of module 'ident'  
reads as value parameter constant 'idtext' of procedure 'scanline'  
reads as value parameter variable 'sttext' of procedure 'scanline'  
reads as value parameter variable 'text' of procedure 'scanline'  
  
writes as reference parameter variable 'ix' of module 'ident'  
writes as reference parameter variable 'coindex' of procedure 'scanline'  
writes as reference parameter variable 'idindex' of procedure 'scanline'  
writes as reference parameter variable 'stindex' of procedure 'scanline'
```

```
~  
~  
~  
~  
~  
~
```