

What is Information Flow Analysis?

Information Flow Analysis, also called variable dependency analysis, is a study of the interdependencies of the program variables. LDRA Testbed analyses these dependencies on a procedure-by-procedure basis is a valid Boolean expression. The explicit syntax can be customised, the standard form corresponding to that of languages such as Ada, C and Pascal.

Information Flow Analysis Example

A variable A is said to depend on another variable B in a procedure if there is a path such that the value of B can cause the value of A to change. Intermediate variables do not appear in lists of dependencies. Only input variables appear in such lists. For example, if variable B is an intermediate variable which depends on C as in:

```
B := C; A := B;
```

then variable A depends on C (and not on B). Different types of dependency are distinguished:

Strongly dependent: If a variable A is defined then it always depends on the variable B, i.e. the value of A depends on B for **every** path containing an assignment to A in the procedure. Such as:

```
A = B + 1
```

Weakly dependent: The variable A sometimes depends on the variable B, i.e. there is at least one path through the procedure on which A is defined with reference to B, and there is another path where A is defined without reference to B.

Such as:

```
if (condition) A = B + 1
```

Conditionally dependent: The variable A does not directly depend on the variable B, but the value of the variable B does influence the value of A by changing the control flow paths. Such as:

```
if (B > 0) A = 0
```

Additionally, two types of definition can be identified:

Strongly defined: A variable is strongly defined if it always acquires a value, i.e. the variable's value is computed on every path through the procedure.

Weakly defined: A variable is weakly defined if it may acquire a value, i.e. the variable is therefore not computed on at least one path through the procedure.

Where users know what the dependencies should be they can insert this information into their code by means of a comment such as (for Ada):

```
--LDRA_INFOFLOW < output variable >[text]([< input variable > {,< input variable >}])
```

where < output variable > is the name of an output variable, [text] is commentary and < input variable > is the name of an input variable on which the output variable depends. If the comment takes up more than one line then the second and subsequent lines must also be legal comments, i.e. for Ada, start with --.

For example:

```
--LDRA_INFOFLOW xi (a,b,c)
--LDRA_INFOFLOW hiy depends on (y,z)
--LDRA_INFOFLOW DGSZ ()
--LDRA_INFOFLOW G5Y depends on (P,Q,R)
--LDRA_INFOFLOW MANY depends on (ONE,TWO,
--THREE,FOUR)
```

LDRA Testbed compares this predicted dependency with the actual dependency and reports the results in the Information Flow Analysis Report. In order to make the previous ideas more explicit, consider the following code:

```

Tbrowse
File Edit Configure View Window Help
program infoflow(input);
var a,b,c,d,e,f,g: integer;
begin
  read (a,c);
  if (c=1)
  then
  begin
    b:=a+7; d:=c+a; e:=a+1;
  end
  else
  begin
    d:=c+2; e:=1;
    if (c=2)
    then
    f:=a+1
    else
    f:=b+1;
  end;
end;
end;

```

For Help, press F1 Ln 53, Col 23 NUM

Example Code for Information Flow Analysis

Dead statements, i.e. statements which do not contribute to output values are reported, so the corresponding Information Flow Analysis would yield:

Strongly defined variables:		
Variable	Strongly directly Dependent	Weakly directly Dependent
a	directly depends on no other variables	
c	directly depends on no other variables	
d	c	a
e		a

Weakly defined variables:		
Variable	Strongly directly Dependent	Weakly directly Dependent
b	a	
f		a b

Strongly defined variables:		
Variable	Strongly conditionally Dependent	Weakly conditionally Dependent
a	conditionally depends on no other variables	
c	conditionally depends on no other variables	
d	c	
e	c	

Weakly defined variables:		
Variable	Strongly conditionally Dependent	Weakly conditionally Dependent
b	c	
f	c	

For Help, press F1 Ln 53, Col 23 NUM

Information Flow Analysis Results for the sample code featured above.

Obtaining Further Information

For further information on this particular feature of TBSafe and its availability please complete: the [LDRA reply form](#) or email info@ldra.com.



Certificate Number FM 26376

www.ldra.com



LDRA UK & Worldwide

Portside, Monks Ferry,
Wirral, CH41 5LH
Tel: +44 (0)151 649 9300
e-mail: info@ldra.com

LDRA Technology Inc.

Lake Amir Office Park, 1250 Bayhill Drive Suite # 360
San Bruno CA 94066 Tel: (650) 583 8880
e-mail: info@ldra.com

LDRA Technology Pvt. Ltd

#2989/1B, 3rd Floor, 12th Main, 80 Feet Road,
HAL II Stage, Bangalore- 560008. Near BSNL Building
Tel: +91 80 4080 8707
e-mail: india@ldra.com