Ast2Cfg - A Framework for CFG-Based Analysis and Visualisation of Ada Programs

Georg Kienesberger - Vienna University of Technology

FOSDEM'09
Free and Open Source Software Developers' European Meeting
7-8 February 2009 - Brussels, Belgium

1. Overview
2. CFG and AST
3. The Software
4. Live Demonstration of Ast2Cfg, Cfg2Dot
5. Examples

Overview

- Control Flow Graph (CFG) used in many analysis/optimisation methods

Overview

- Control Flow Graph (CFG) used in many analysis/optimisation methods
- ISO/IEC 15291:1999 - Ada Semantic Interface Specification (ASIS) implemented by ASIS-for-GNAT

Overview

- Control Flow Graph (CFG) used in many analysis/optimisation methods
- ISO/IEC 15291:1999 - Ada Semantic Interface Specification (ASIS) implemented by ASIS-for-GNAT
- traverse Abstract Syntax Tree (AST) and build CFG

Overview

- Control Flow Graph (CFG) used in many analysis/optimisation methods
- ISO/IEC 15291:1999 - Ada Semantic Interface Specification (ASIS) implemented by ASIS-for-GNAT
- traverse Abstract Syntax Tree (AST) and build CFG
- Framework

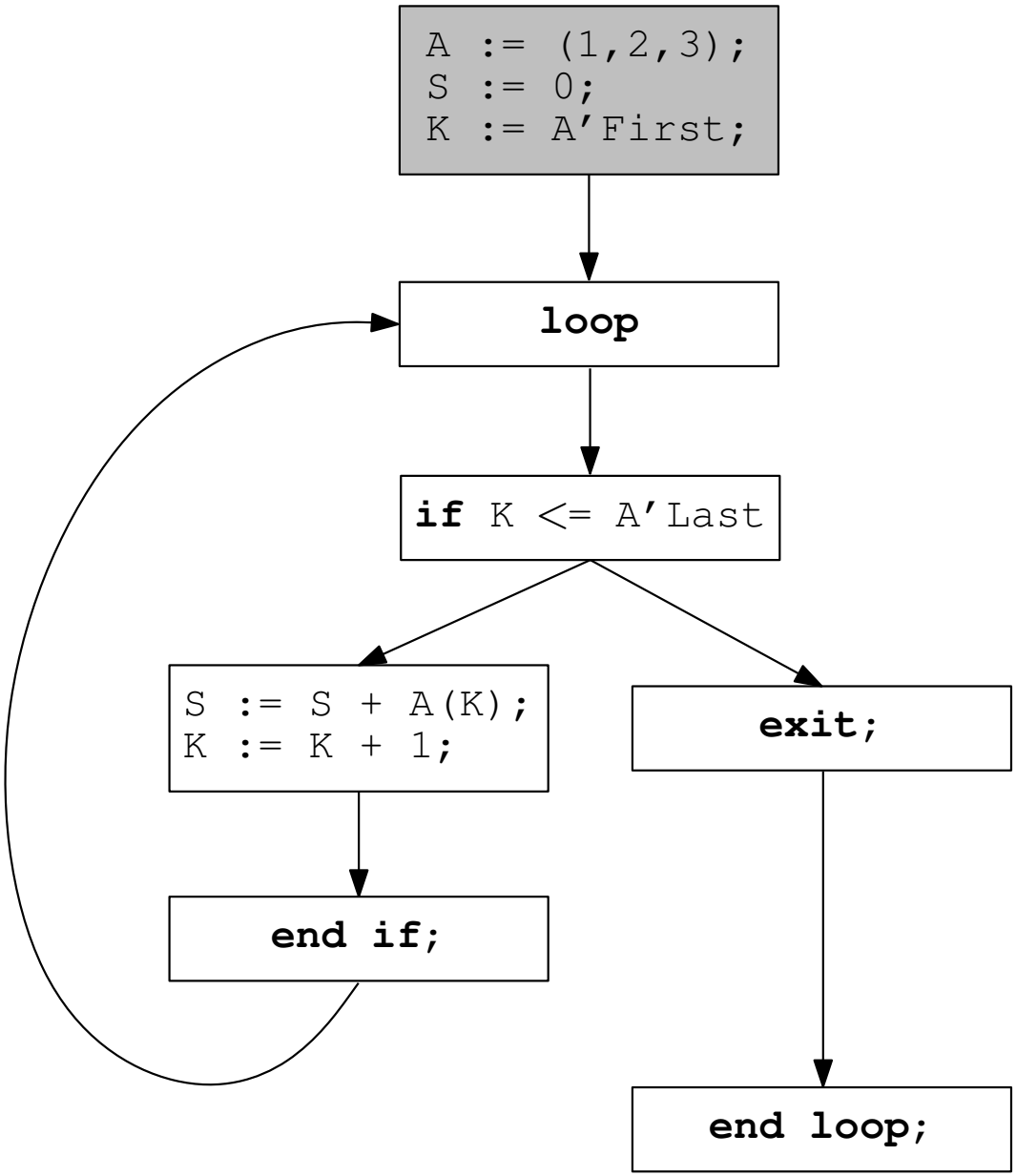
Control Flow Graph (CFG)

- directed graph
- nodes represent statements
- There is an edge from node u to node v if v can follow u in some execution sequence.
- unique start node called *root* or *initial* node

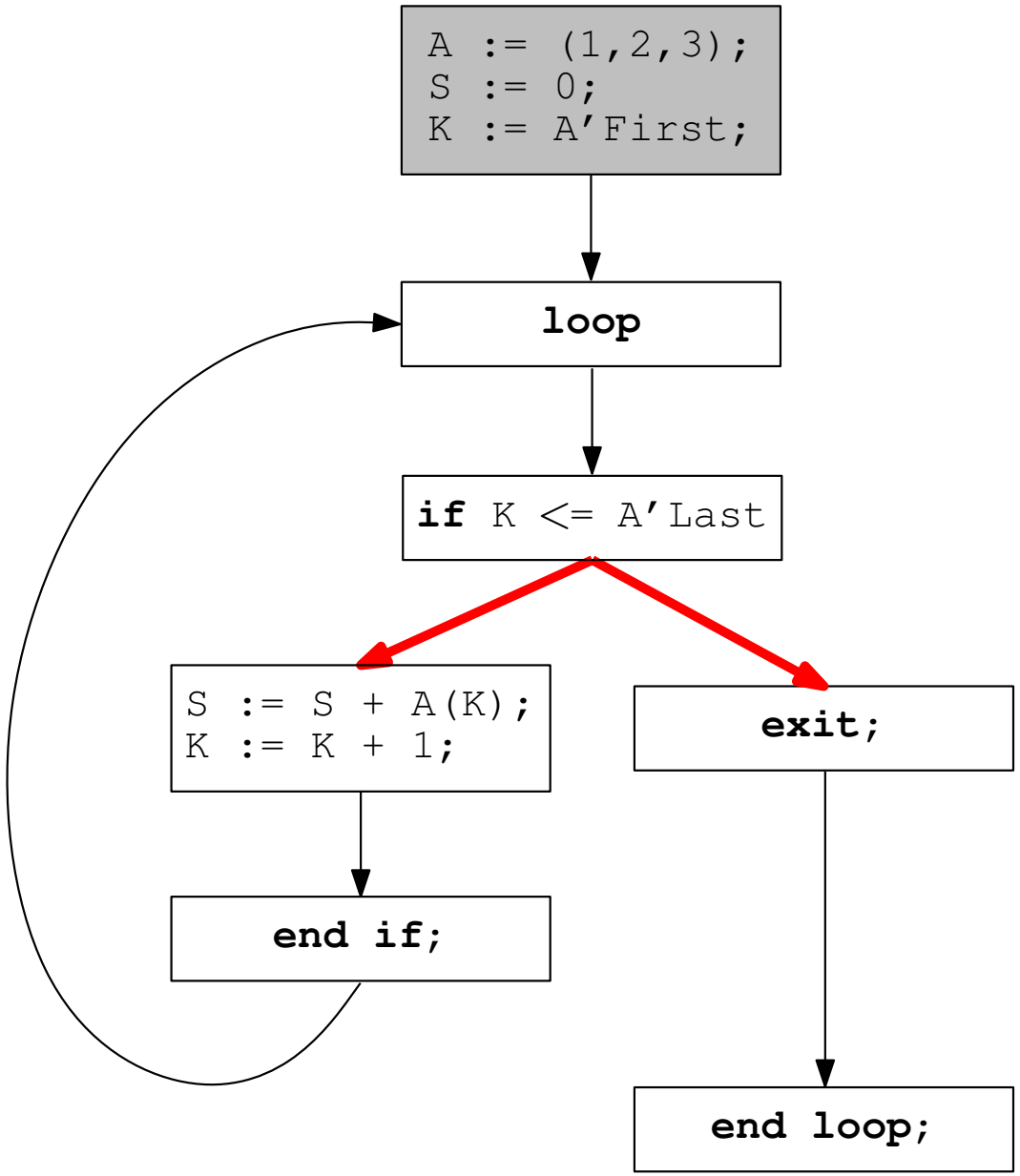
Example: The Source Code

```
procedure Test is
   A: array (1 .. 3) of Natural;
   S: Natural;
   K: Natural;
begin
   A := (1,2,3);
   S := 0;
   K := A'First;

   loop
      if K <= A'Last then
         S := S + A(K);
         K := K + 1;
      else
         exit;
      end if;
   end loop;
end Test;
```

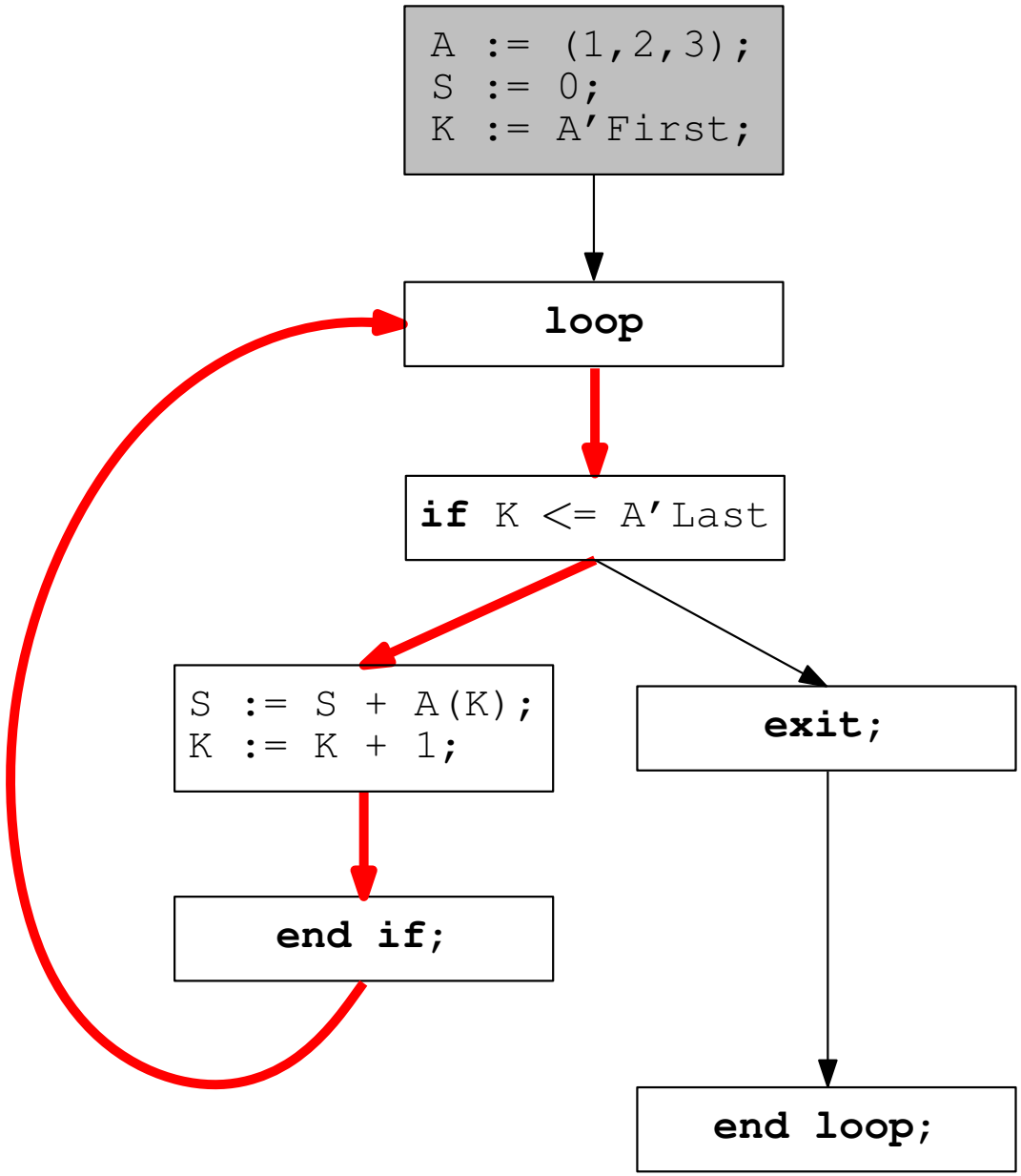
Example: The Control Flow Graph (CFG)



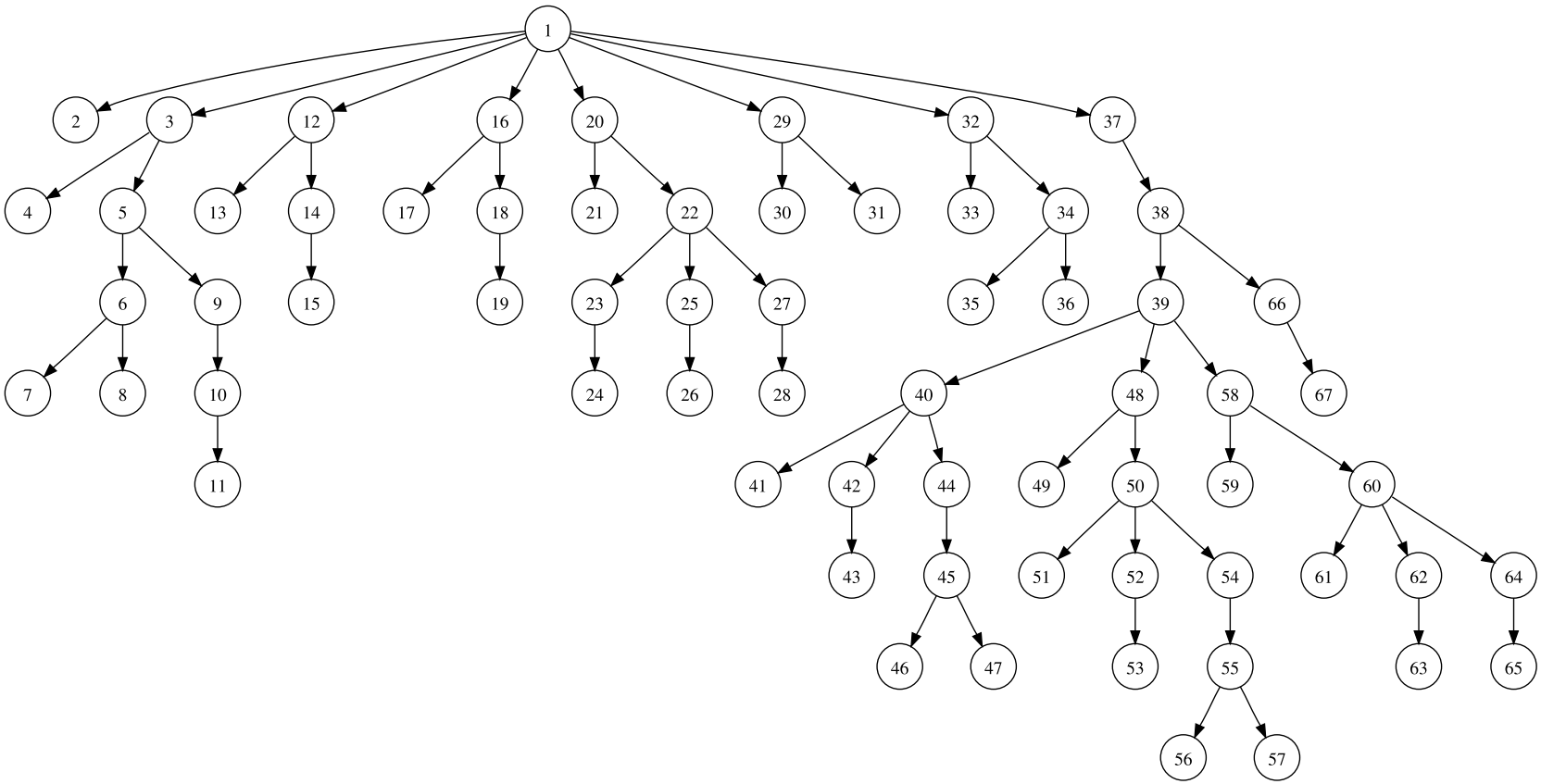
Example: The Control Flow Graph (CFG)



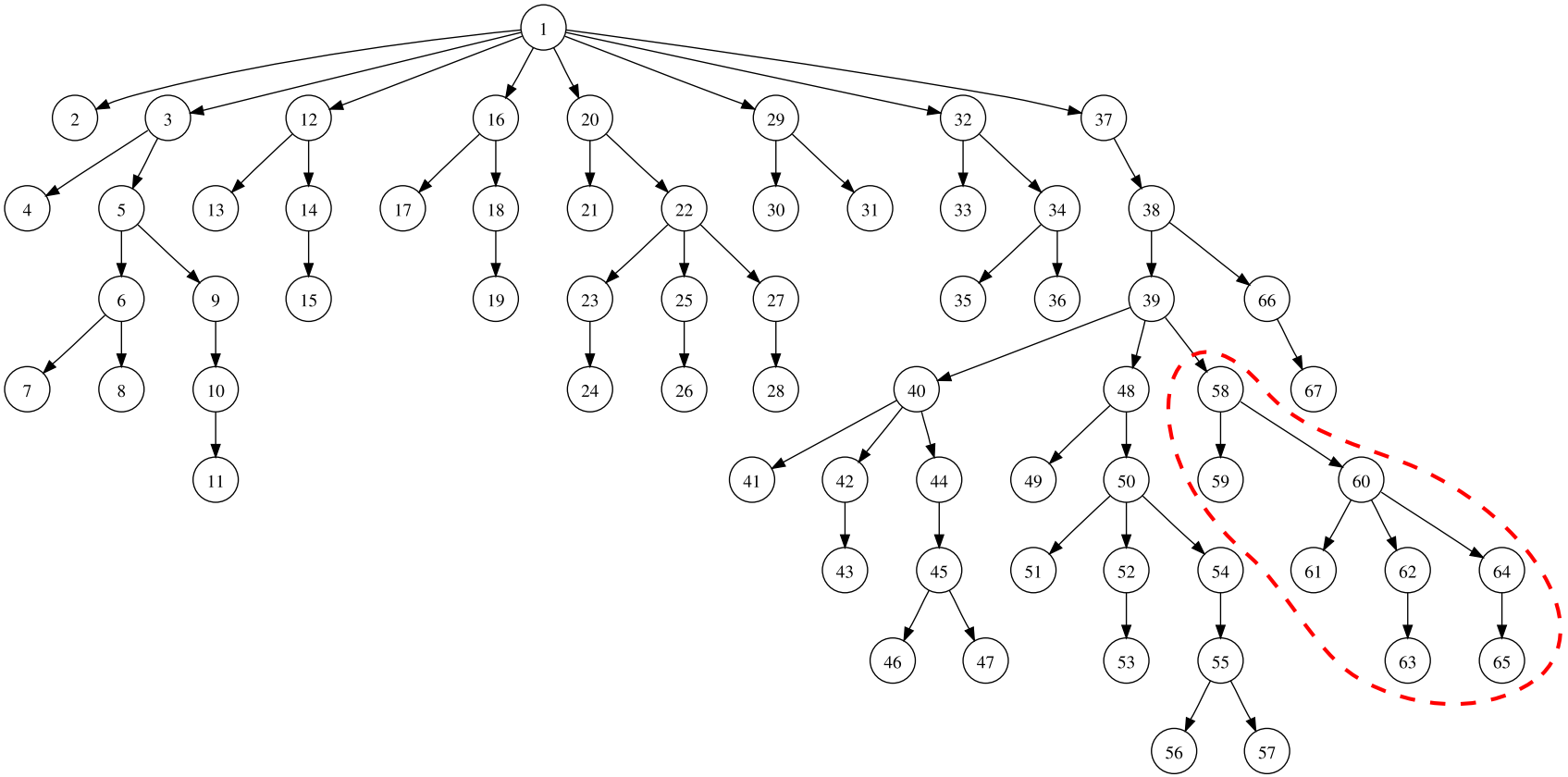
Example: The Control Flow Graph (CFG)



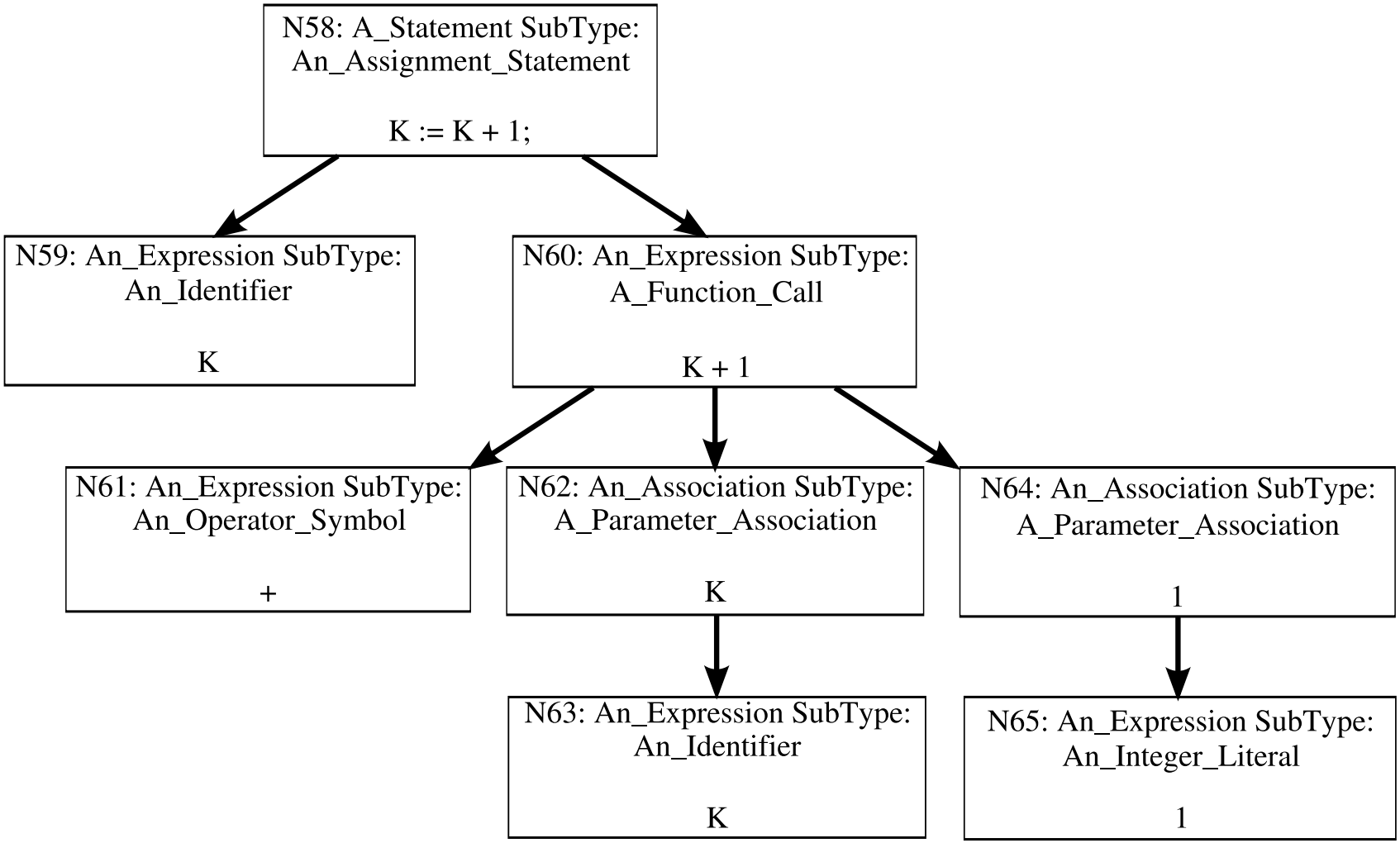
Example: The ASIS Abstract Syntax Tree (AST) - 1



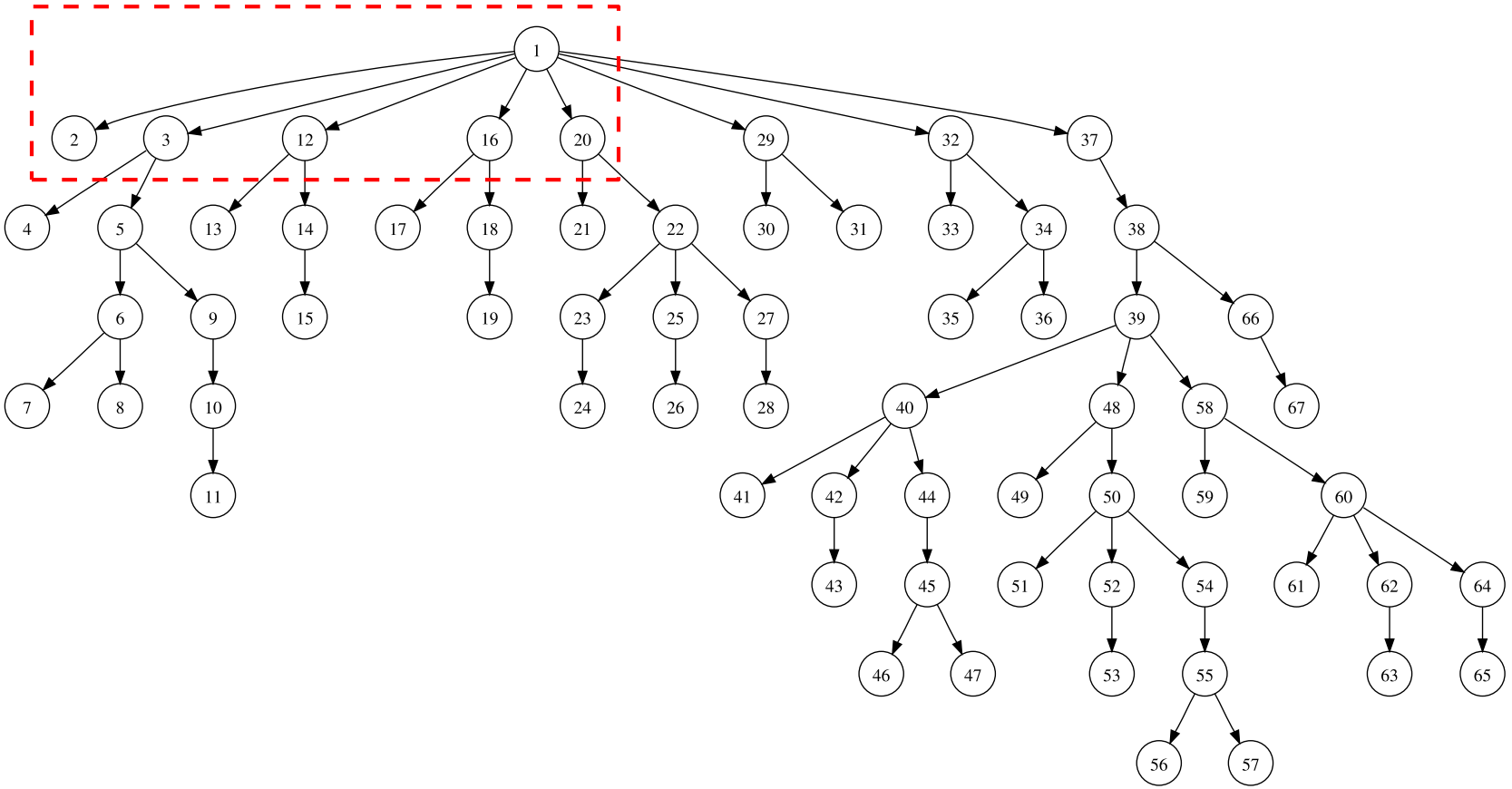
Example: The ASIS Abstract Syntax Tree (AST) - 1



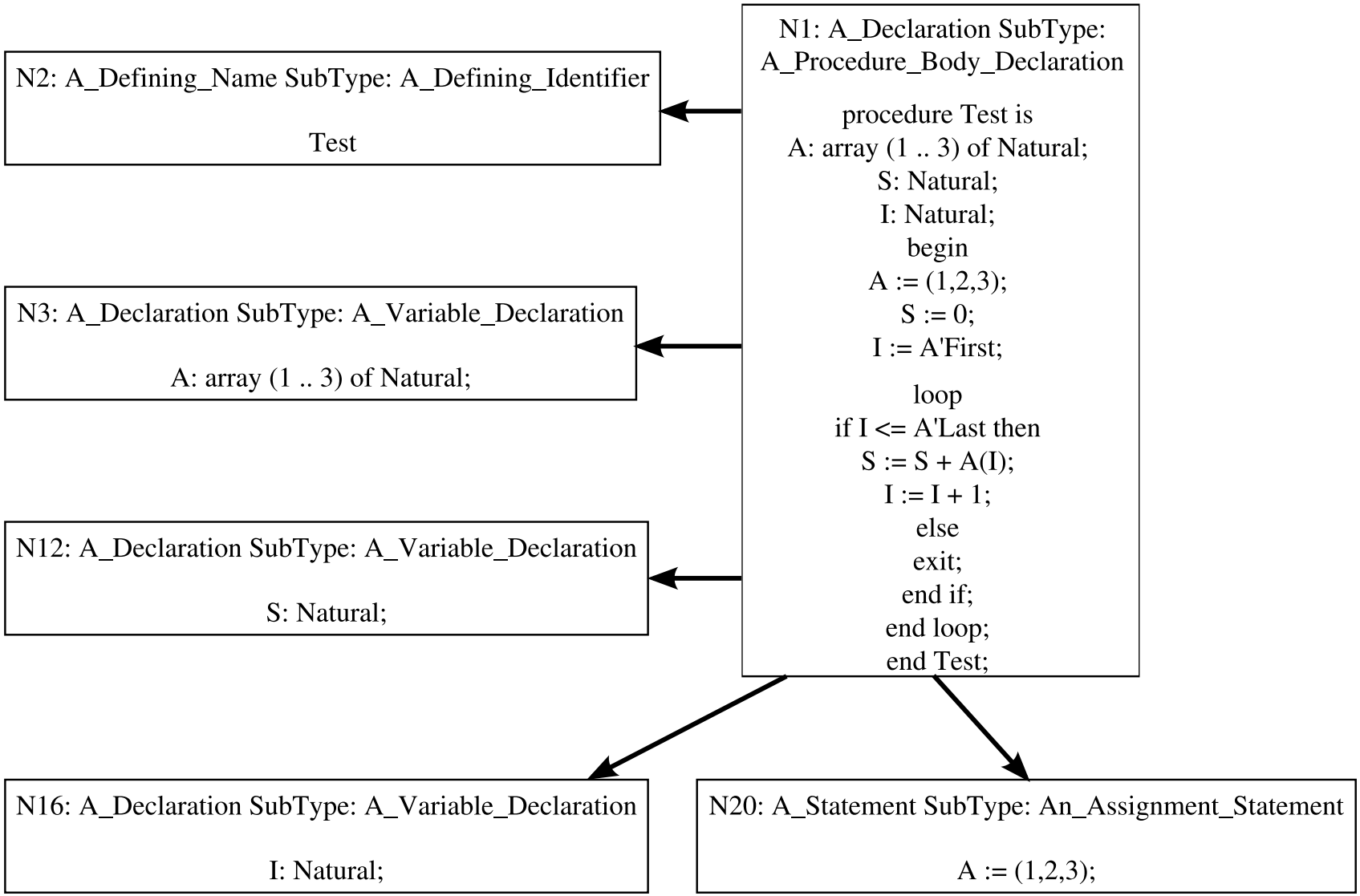
Example: The ASIS Abstract Syntax Tree (ASIS) - 2



Example: The ASIS Abstract Syntax Tree (ASIS) - 3



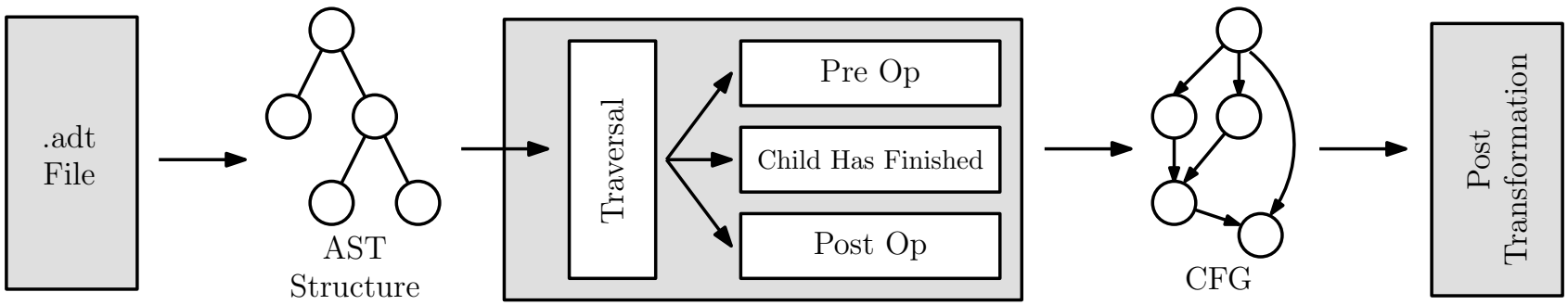
Example: The ASIS Abstract Syntax Tree (AST) - 4



The Software

- Ast2Cfg - the framework, designed as a library
- Cfg2Dot - uses Ast2Cfg, outputs the CFG structure in *dot* format
- Ast2Dot - uses ASIS directly to output the AST in *dot* format
- available under GPLv2 (Ast2Dot) or GPLv3 (Ast2Cfg, Cfg2Dot)

The Structure of the Transformation Process



Flow World: The Main Data Structure

- `World_Object`

Flow World: The Main Data Structure

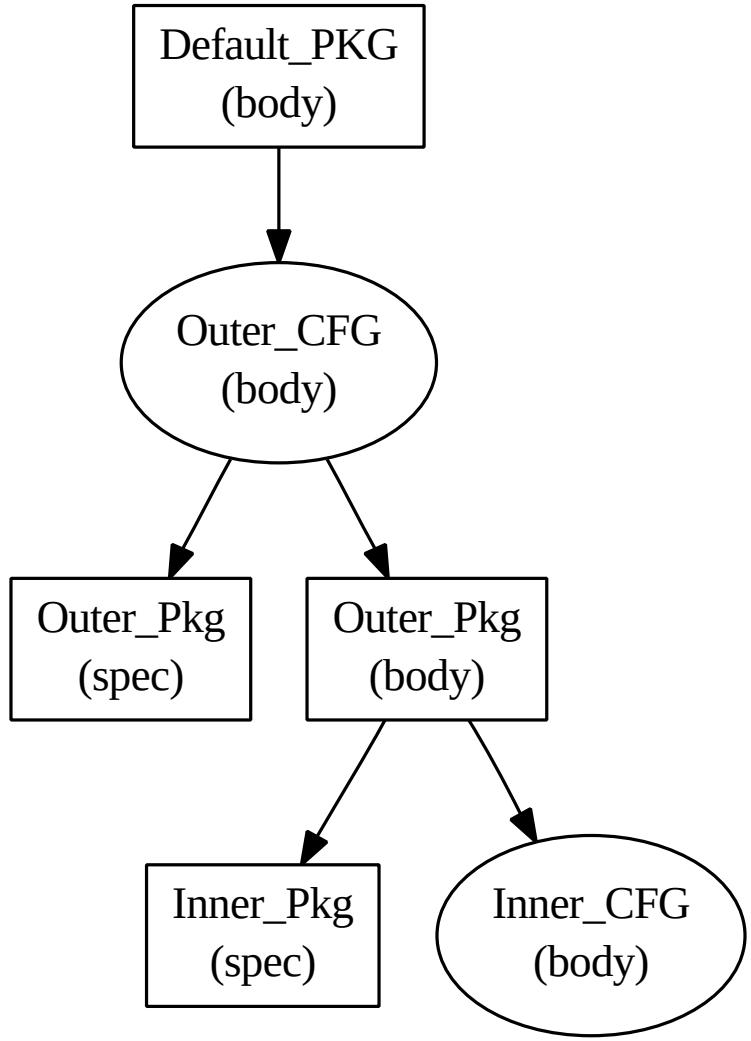
- `World_Object`
- `Flow_Object`
  - `Pkg_Object`
  - `CFG_Object`
  - `Node_Object`

Flow World: The Main Data Structure

- `World_Object`
- `Flow_Object`
  - `Pkg_Object`
  - `CFG_Object`
  - `Node_Object`
- Pkg/CFG Tree (implicit structure)

Example: Pkg/CFG Tree

```
procedure Outer_CFG is

  package Outer_Pkg is
  end Outer_Pkg;

  package body Outer_Pkg is

    package Inner_Pkg is
    end Inner_Pkg;

    procedure Inner_CFG is
    begin
    end Inner_CFG;

  end Outer_Pkg;

begin
end Outer_CFG;
```

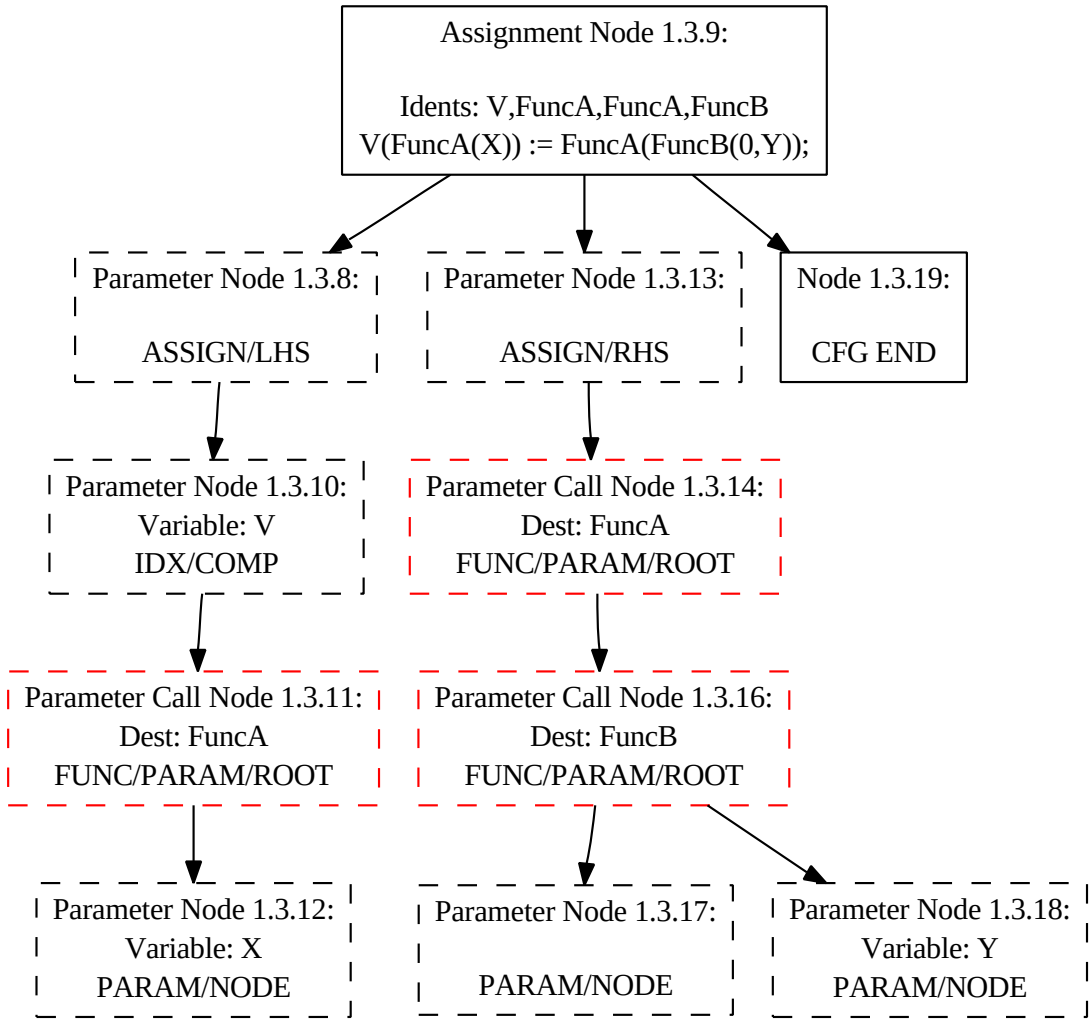


Flow World: The Main Data Structure (contd.)

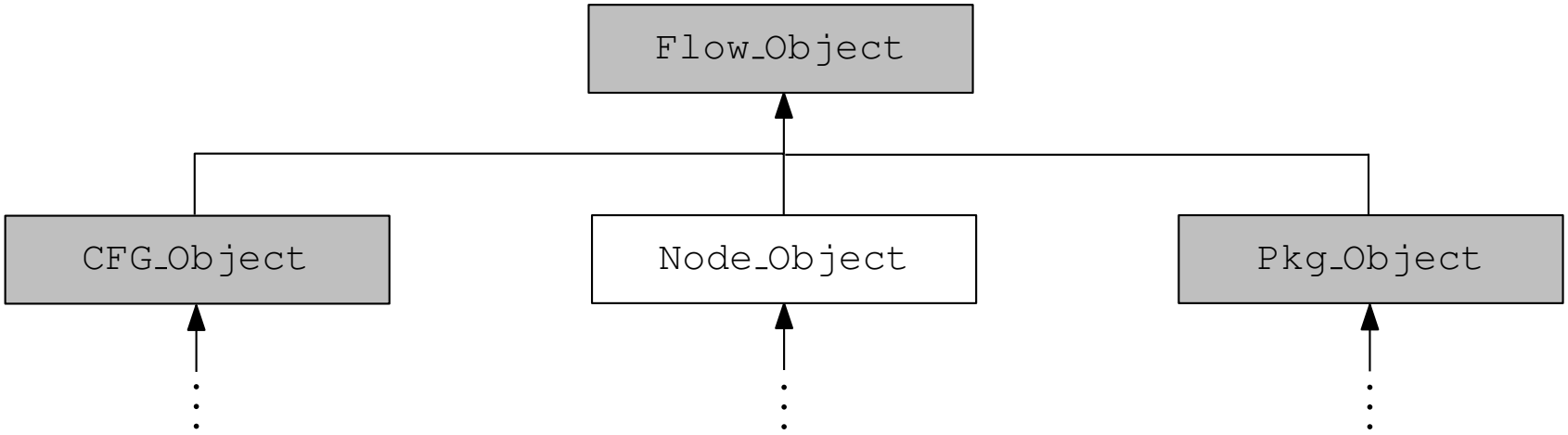
- `World_Object`
- `Flow_Object`
  - `Pkg_Object`
  - `CFG_Object`
  - `Node_Object`
- Pkg/CFG Tree (implicit structure)
- Parameter Tree (saved in nodes)

Example: Parameter Tree

```
V: array(a..b)
  of Integer;

V(FuncA(X)) :=
  FuncA(FuncB(0,Y));
```



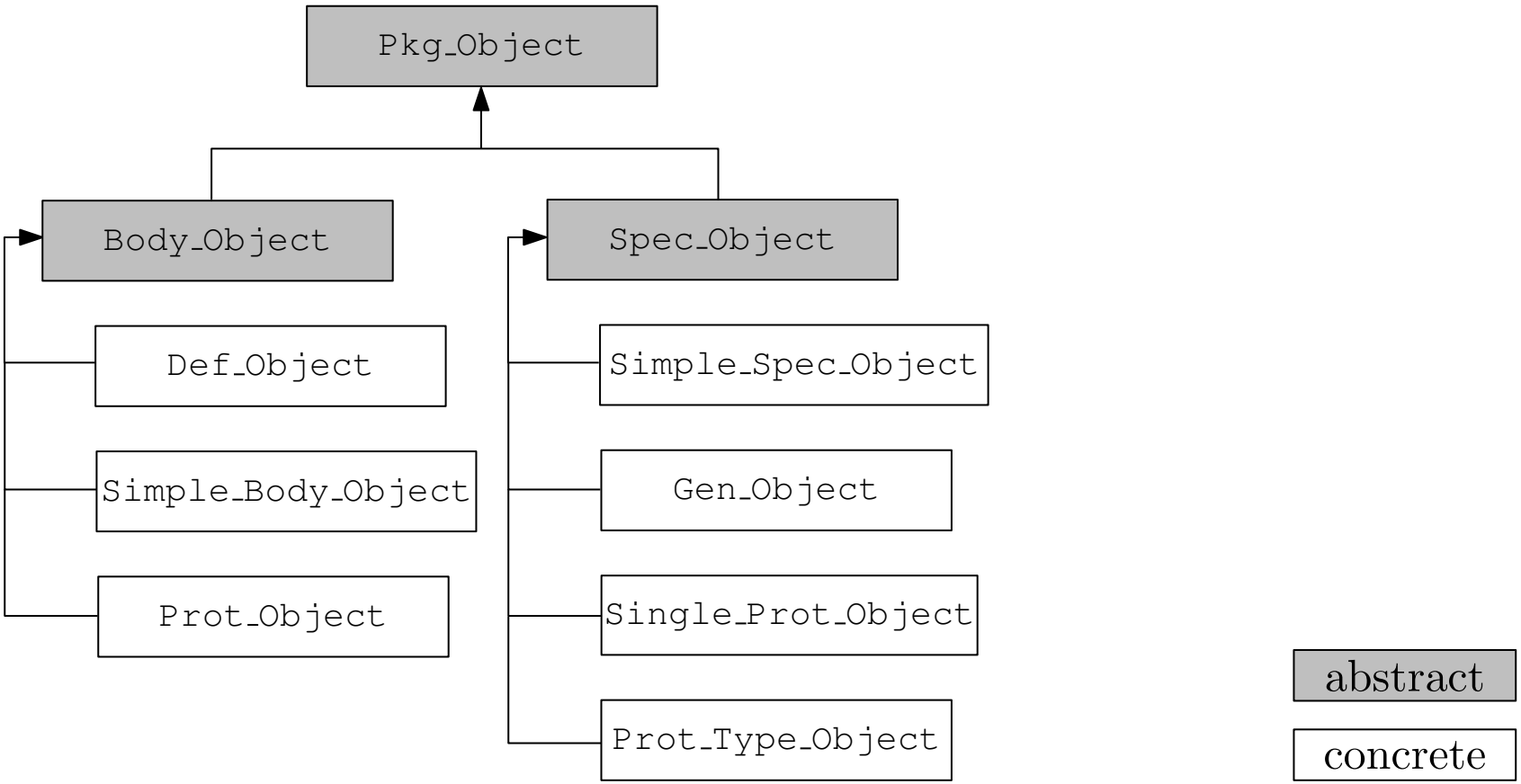
Flow Types: An Overview



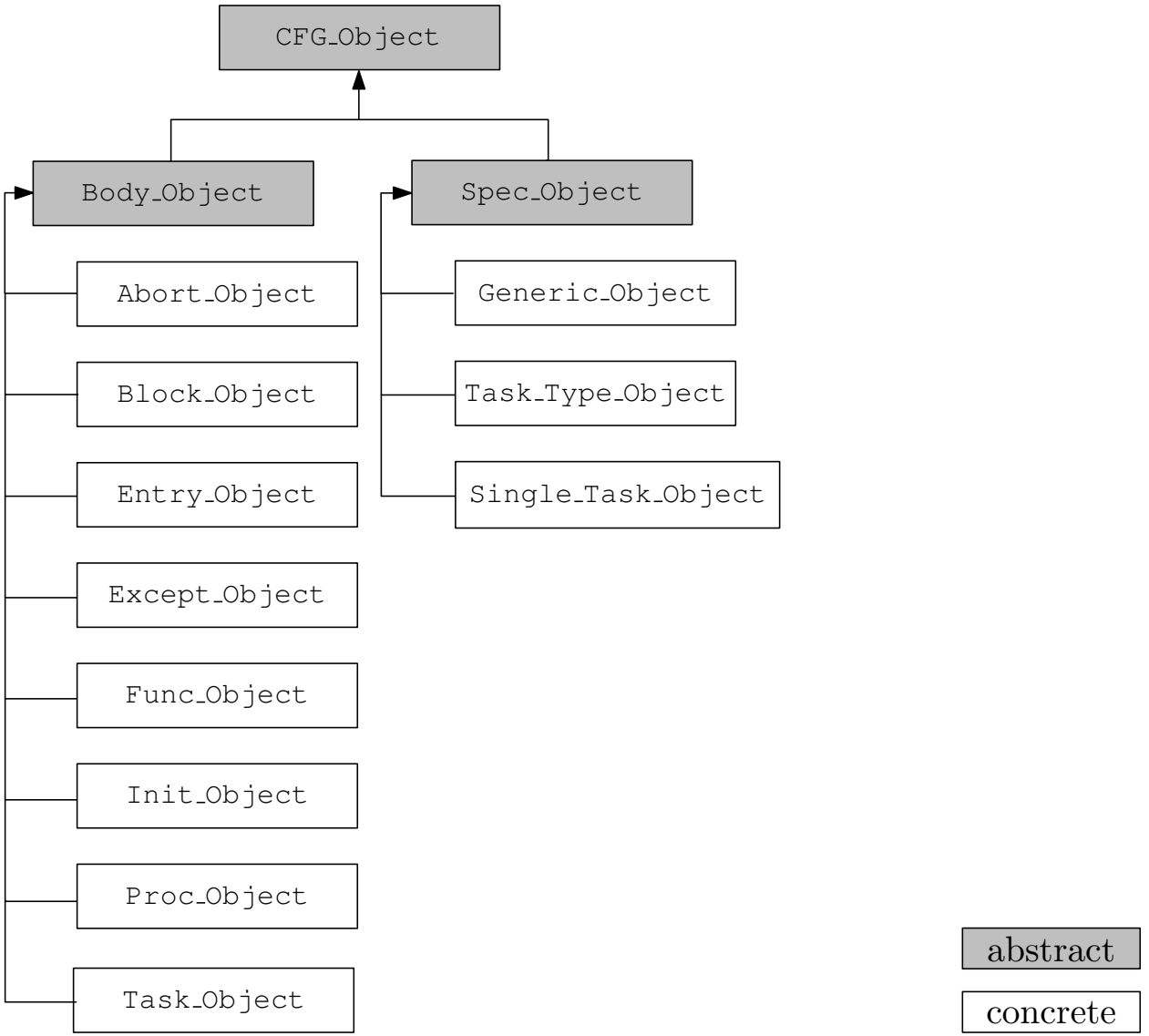
abstract

concrete

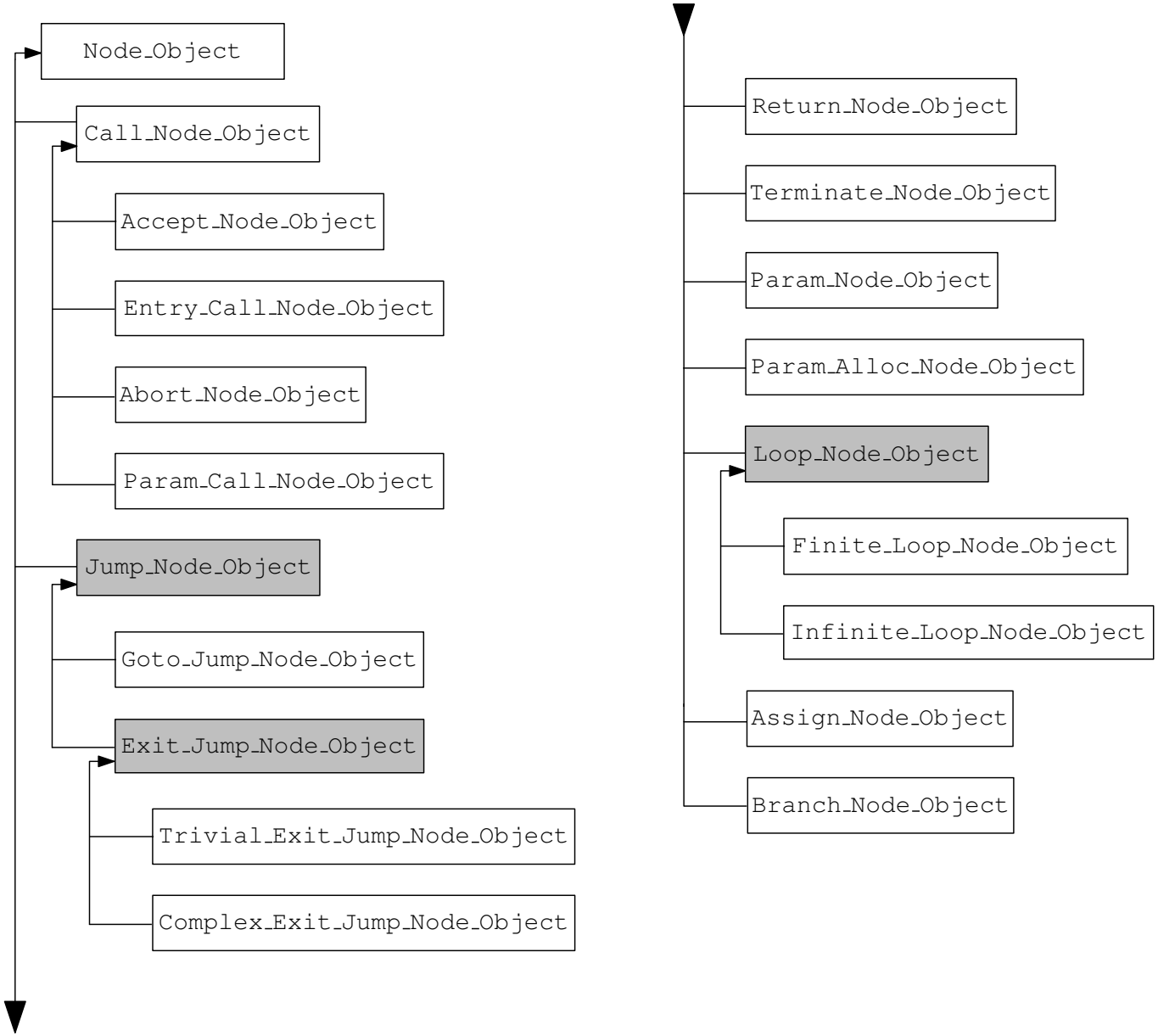
Flow Types: Package



Flow Types: CFG



Flow Types: Node



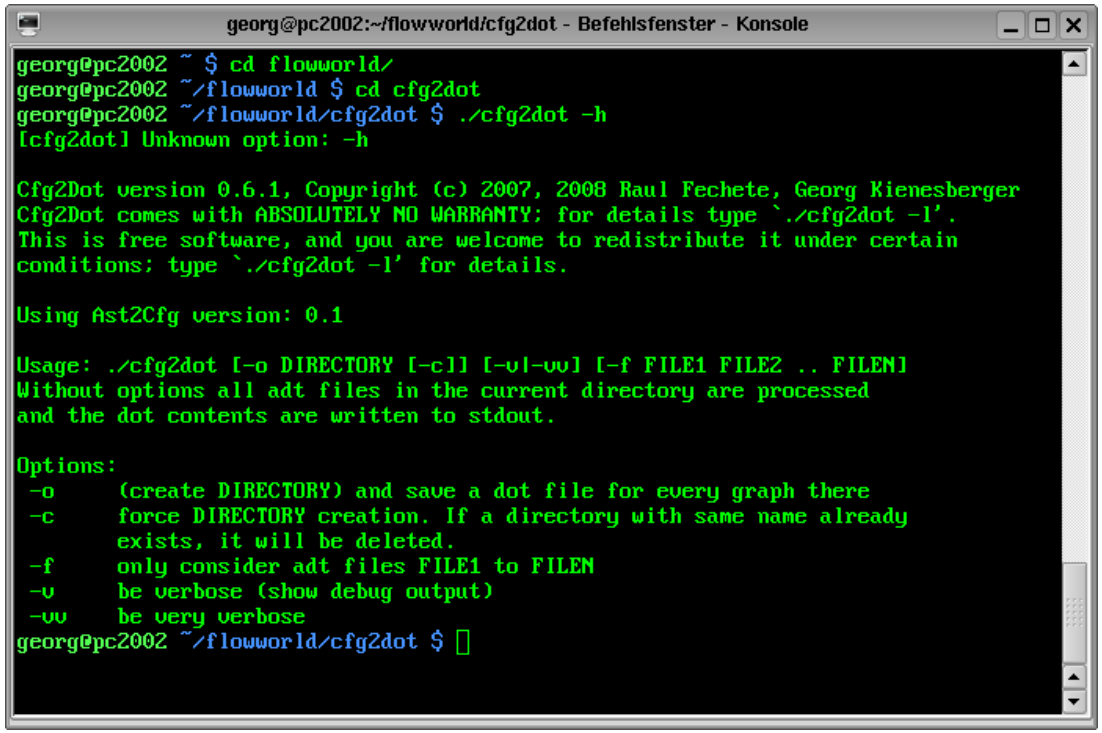
abstract

concrete

Usage

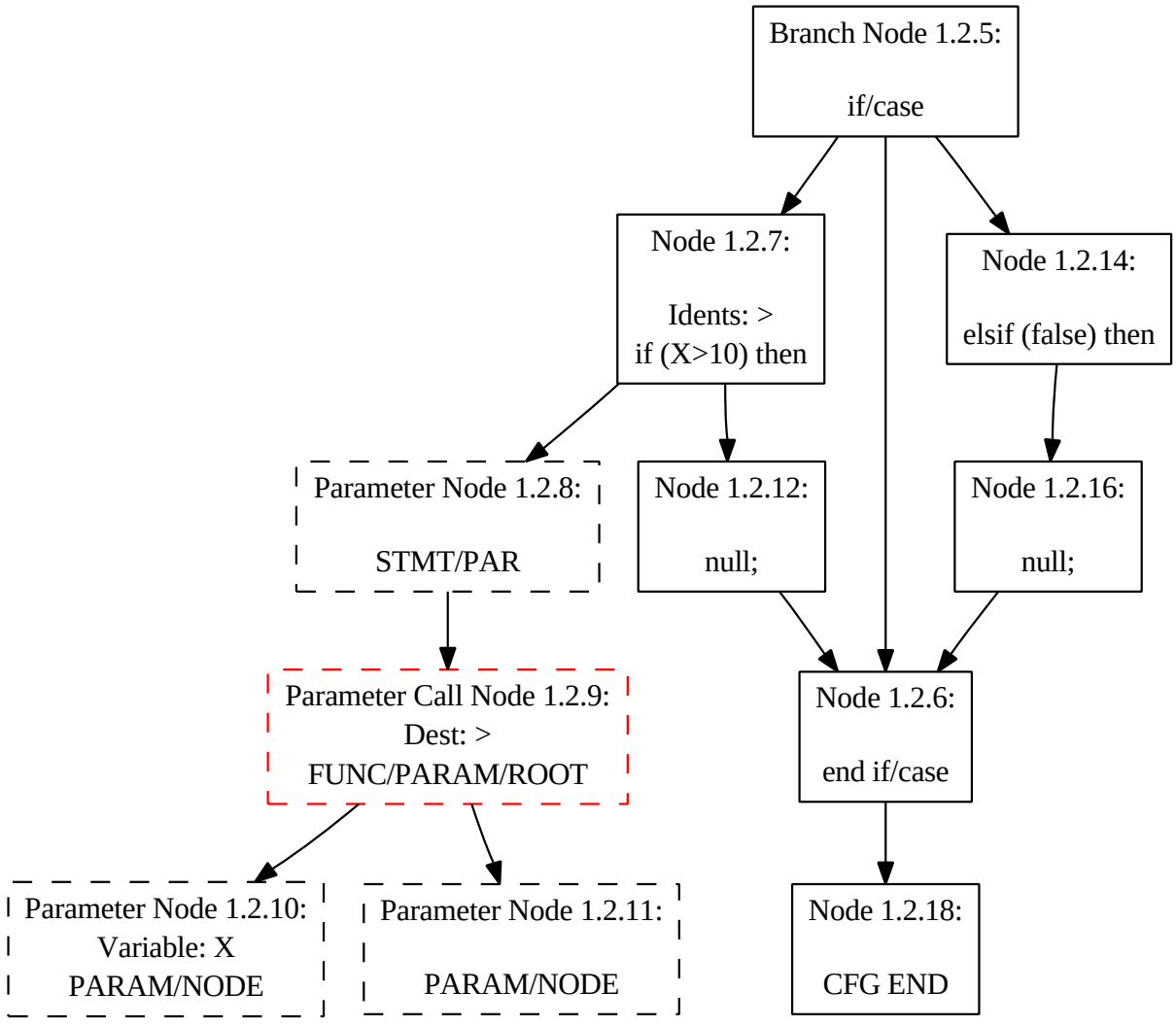
```
with Ada.Text_IO; use Ada.Text_IO; with Ast2Cfg.Pkgs; use Ast2Cfg.Pkgs;
with Ast2Cfg.Control; with Ast2Cfg.Flow_World; with Ast2Cfg.Output;

procedure Run is
        World: Ast2Cfg.Flow_World.World_Object_Ptr;
        Pkgs: Pkg_Class_Ptr_List.Object;
        Pkg:  Pkg_Class_Ptr := null;
begin
        -- Initialisations
        Ast2Cfg.Output.Set_Level(Ast2Cfg.Output.Warning);
        Ast2Cfg.Control.Init("–CN foo.adt bar.adt");

        -- Fill the World with flow data
        World := Ast2Cfg.Control.Generate;

        -- Output the name of all top-level packages
        Pkgs := Ast2Cfg.Flow_World.Get_Pkgs(World.all);
        Pkg_Class_Ptr_List.Reset(Pkgs);
        while Pkg_Class_Ptr_List.Has_Next(Pkgs) loop
                Pkg_Class_Ptr_List.Get_Next(Pkgs, Pkg);
                Put_Line(Get_Name(Pkg.all));
        end loop;

        -- Finalisation
        Ast2Cfg.Control.Final;
end Run;
```

Excursus on the Command Line



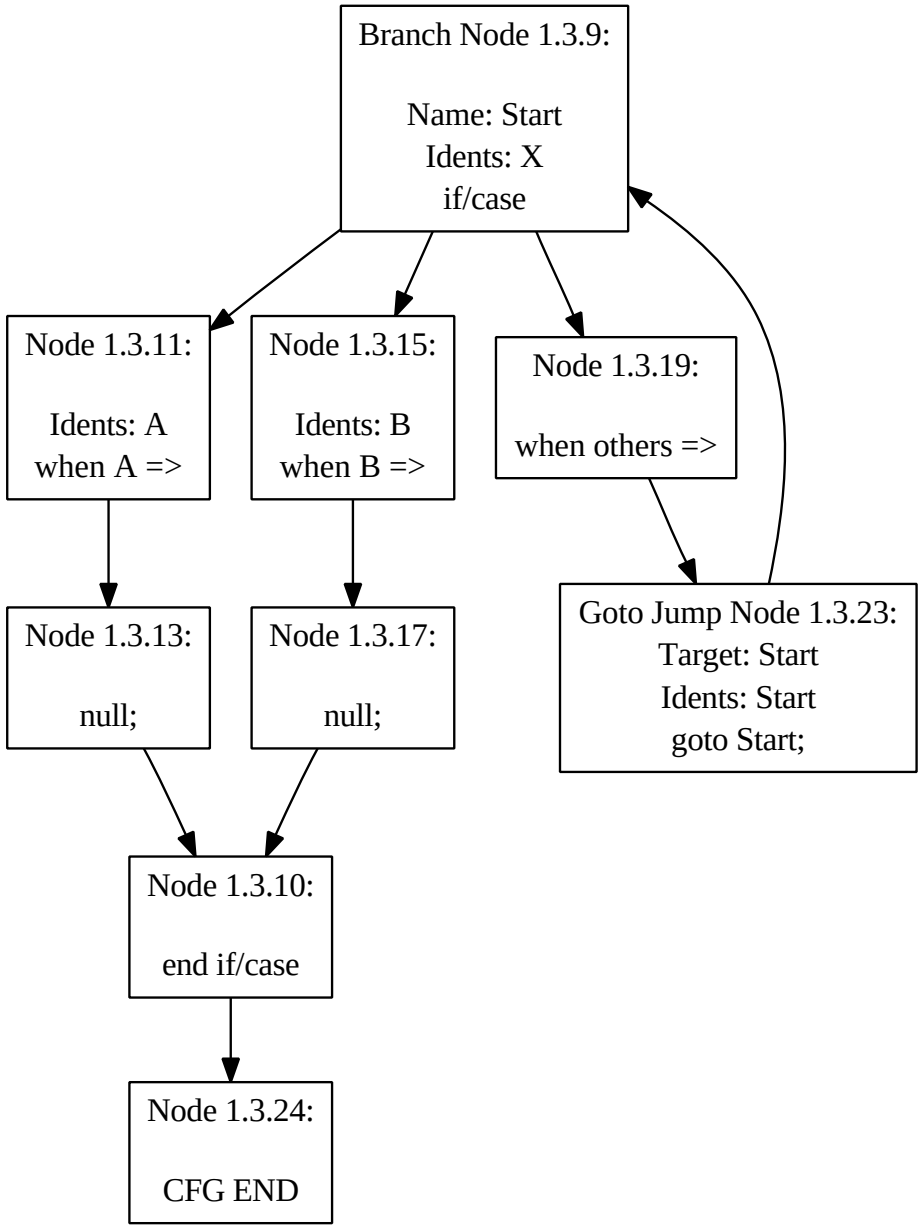
Examples: If Statement

```
if (X>10) then
  null;
elsif (false) then
  null;
end if;
```



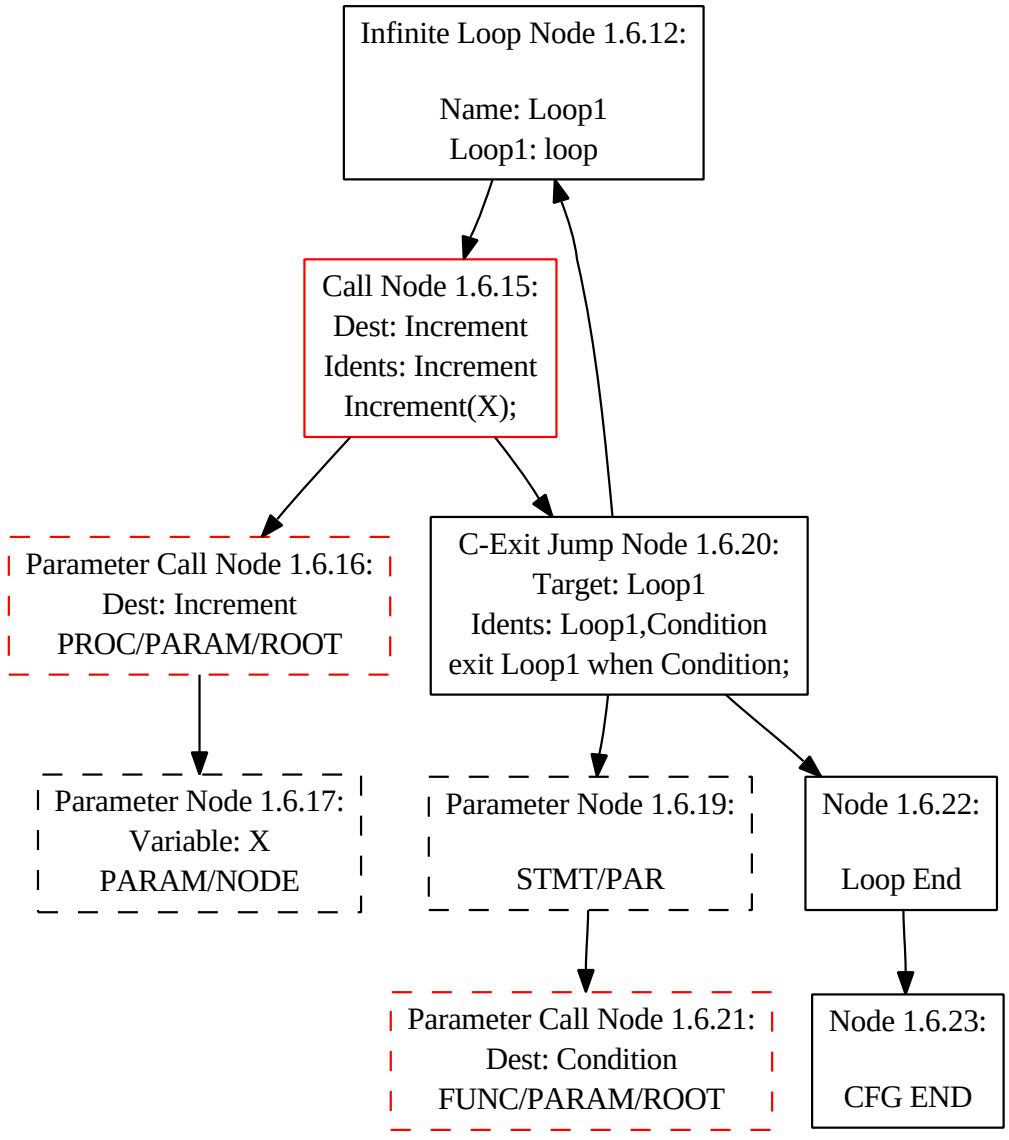
Examples: Case Statement

```
<<Start>> case X is
   when A =>
      null;
   when B =>
      null;
   when others =>
      goto Start;
end case;
```



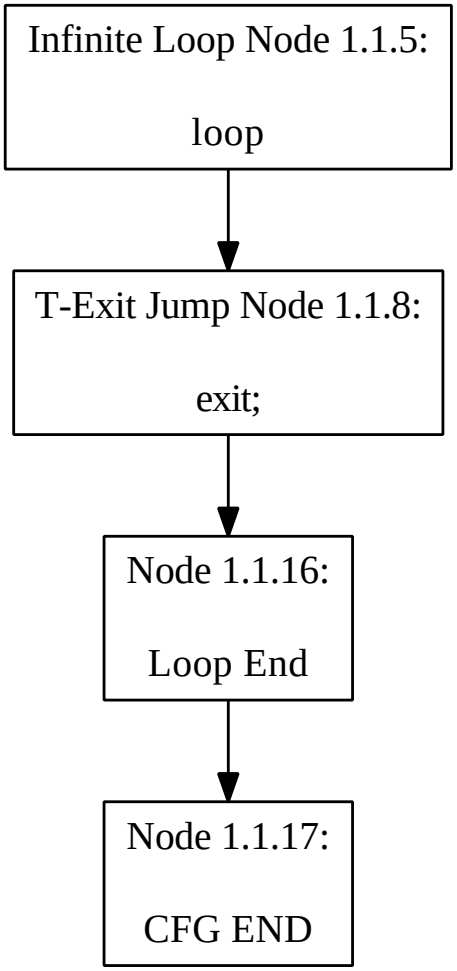
Examples: Loop Statement

```
Loop1: loop
  Increment(X);
  exit Loop1
    when Condition;
end loop Loop1;
```



Examples: A few words on the word *static*

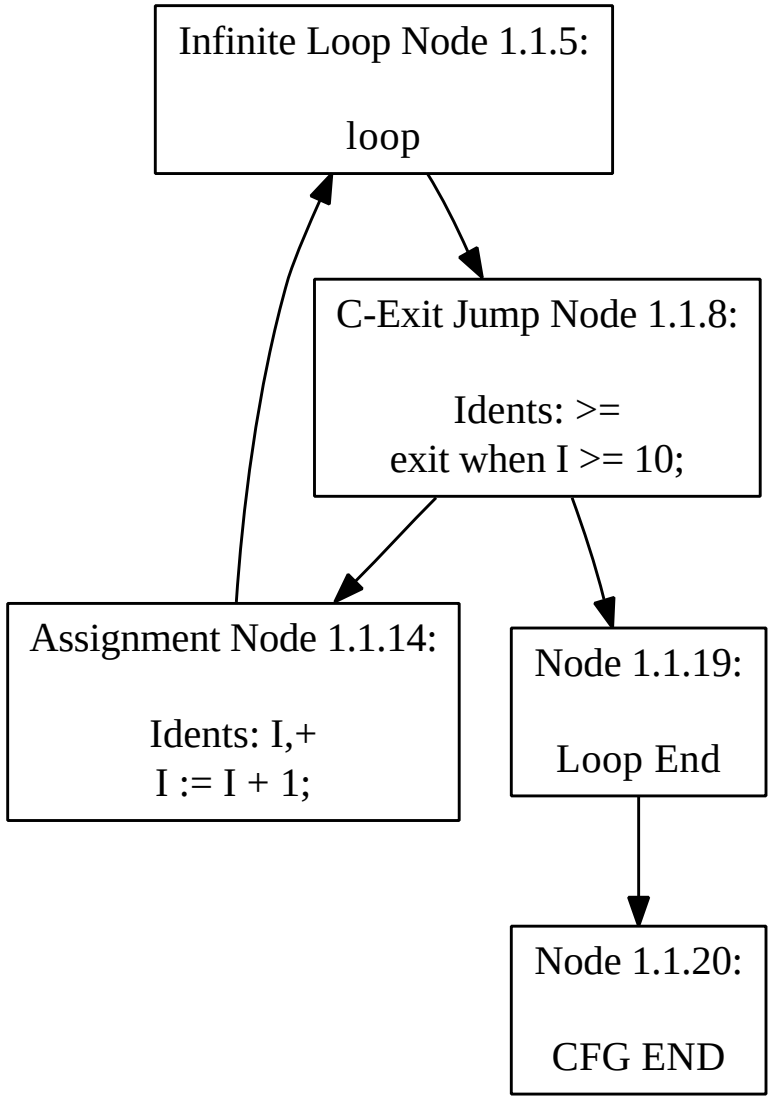
```
procedure Not_A_Loop is
   I: Integer := 0;
begin

   loop
      exit;
      I := I + 1;
   end loop;

end Not_A_Loop;
```




```
not_a_loop.adb:7:09: warning: unreachable code
```

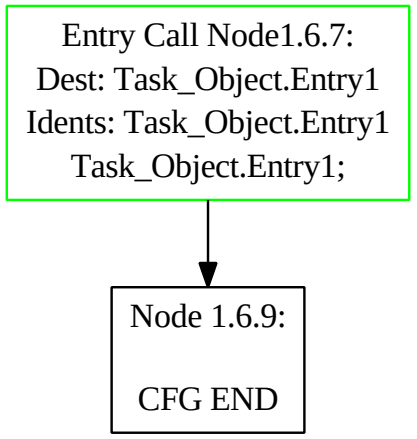
Examples: A few words on the word *static* (2)

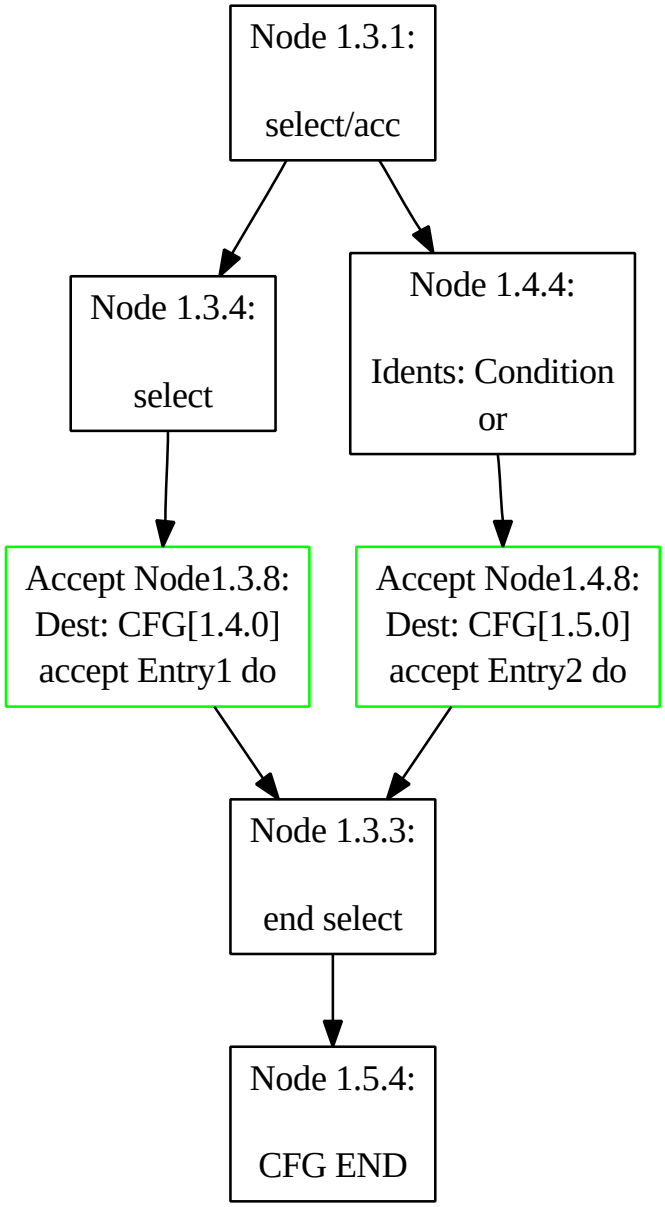
```
procedure Not_A_Loop_2 is
   I: Integer := 10;
begin

   loop
      exit when I >= 10;
      I := I + 1;
   end loop;

end Not_A_Loop_2;
```



Examples: Tasks

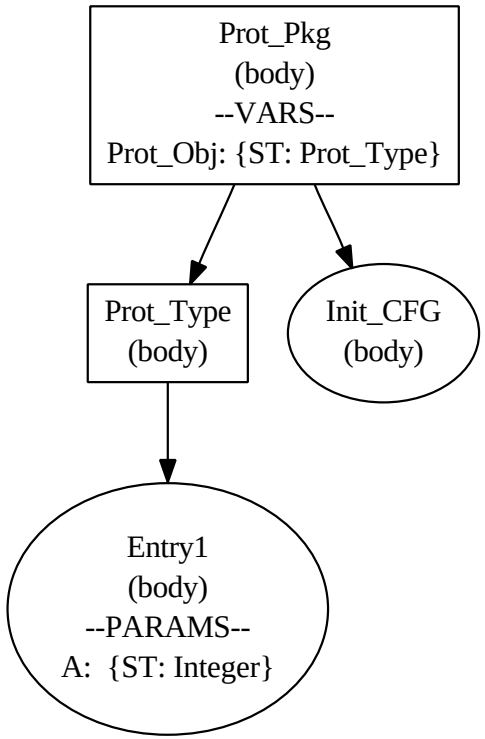
```
-- Caller Object
Task_Object.Entry1;

-- Called Object
select
  accept Entry1 do
    null;
  end Entry1;
or
  when Condition =>
    accept Entry2 do
      null;
    end Entry2;
end select;
```

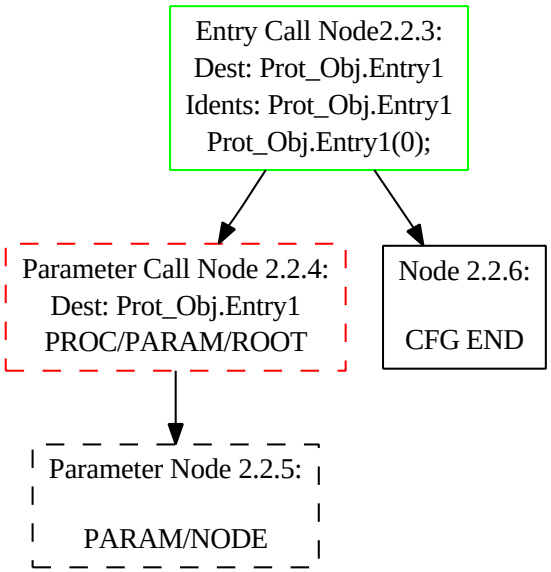




Examples: Protected Objects

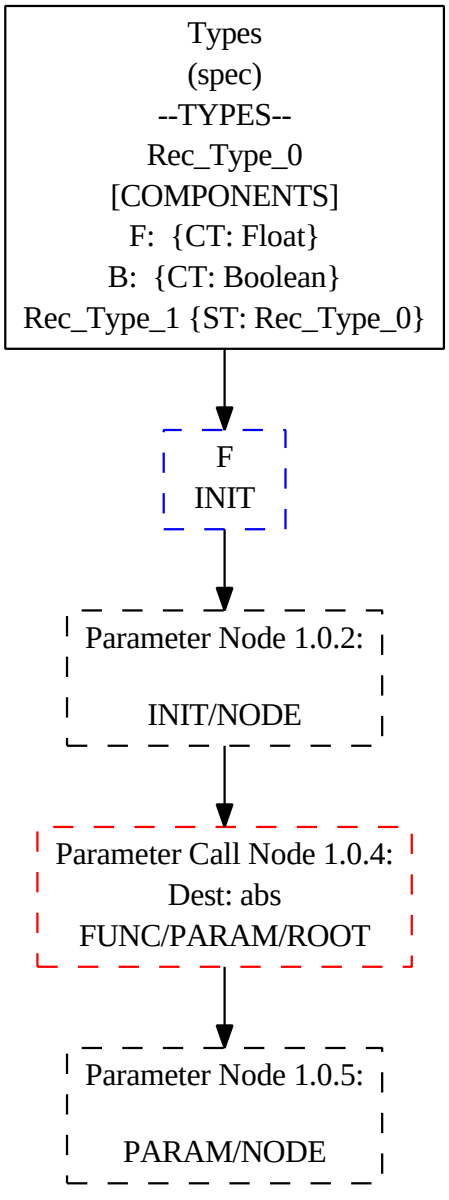
```
package body Prot_Pkg is

  protected body
    Prot_Type is

    entry Entry1
      (A: in Integer)
      when True is
      begin
        null;
      end Entry1;

   end Prot_Type;

  Prot_Obj: Prot_Type;

begin
  Prot_Obj.Entry1(0);
end Prot_Pkg;
```





Examples: Types

```
package Types is

   type Rec_Type_0 is
     tagged
      record
        F: Float := abs(3.14);
        B: Boolean;
      end record;

   type Rec_Type_1 is
     new Rec_Type_0
     with null record;

end Types;
```



Further Information & Downloads

- http://cfg.w3x.org
- For comments, bug reports and feature requests please contact us:
  - cfg@w3x.org

Further Information & Downloads

- http://cfg.w3x.org
- For comments, bug reports and feature requests please contact us:
  - cfg@w3x.org
- Raul Fechete and Georg Kienesberger. **Generating control flow graphs for Ada programs. Technical Report** 183/1-139, Department of Automation, TU Vienna, September 2007.

Further Information & Downloads

- http://cfg.w3x.org
- For comments, bug reports and feature requests please contact us:
  - cfg@w3x.org
- Raul Fechete and Georg Kienesberger. **Generating control flow graphs for Ada programs. Technical Report** 183/1-139, Department of Automation, TU Vienna, September 2007.
- Raul Fechete, Georg Kienesberger, and Johann Blieberger. **A Framework for CFG-based Static Program Analysis of Ada Programs.** In **Ada-Europe'2008** International Conference on Reliable Software Technologies, pages 130-143, Venice, Italy, June 2008.

Further Information & Downloads

- http://cfg.w3x.org
- For comments, bug reports and feature requests please contact us:
  - cfg@w3x.org
- Raul Fechete and Georg Kienesberger. **Generating control flow graphs for Ada programs. Technical Report** 183/1-139, Department of Automation, TU Vienna, September 2007.
- Raul Fechete, Georg Kienesberger, and Johann Blieberger. **A Framework for CFG-based Static Program Analysis of Ada Programs.** In **Ada-Europe'2008** International Conference on Reliable Software Technologies, pages 130-143, Venice, Italy, June 2008.
- updated documentation in the next few months

Thank you very much!

Any questions?

http://cfg.w3x.org