



THE UNIVERSITY OF
MELBOURNE

SWEN90010 High Integrity System

Workshop 9 Spark Pointer



Chuang Wang





THE UNIVERSITY OF
MELBOURNE

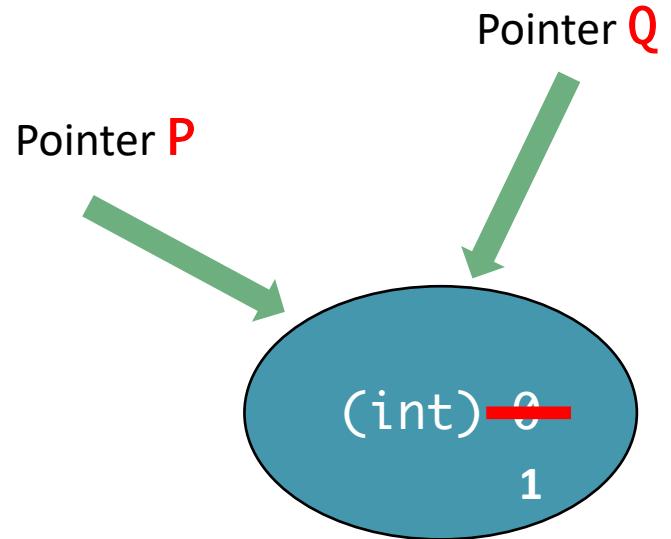
1. Recap

Problem of Alias

```
int x = P;
```

```
Q = P;
```

```
Q <- 1;
```



- The value of x???

x is no longer equal to the content referenced by pointer P !

- Problem of Alias

Changing one things ends up changing something else, which makes reasoning your program harder!



SPARK Prover

- This is where Spark prover comes in as it can detect illegal alias.
- You can use multiple pointers to point same piece of memory.
- You cannot use the old reference to look up the memory.



Pointer in Spark

```
procedure Pass_By_Reference is
    type Rec is record
        X : Integer;
    end record;
    type RecPtr is access all Rec; pointer type
    R : aliased Rec; allows constructing pointers to R

    procedure Print_R is
    begin
        Put("R.X: "); Put(R.X); New_Line;
    end Print_R;      takes a pointer as its argument

    procedure CheckR(givenR : in RecPtr) is
    begin
        Put("In CheckR"); New_Line;
        givenR.X := R.X + 1;
        Print_R;
    end CheckR;

    begin          constructing a pointer to R
        R.X := 5;
        CheckR(R'Access);
        Put("CheckR done"); New_Line;
        Print_R;

    end Pass_By_Reference;
```

The code illustrates pointer usage in Spark. It defines a record type 'Rec' with a single integer field 'X'. A pointer type 'RecPtr' is defined as 'access all Rec', which is highlighted with a yellow oval. The variable 'R' is declared as an 'aliased Rec', allowing the construction of pointers to it, also highlighted with a yellow oval. The 'CheckR' procedure takes a parameter of type 'in RecPtr' (also highlighted with a yellow oval), demonstrating how a pointer can be passed as an argument. Inside 'CheckR', the value of 'givenR.X' is modified, and then 'Print_R' is called to show the updated value. Finally, a check is performed using 'R' directly and another call to 'Print_R' is made. The entire block of code is enclosed in a 'Pass_By_Reference' procedure.



THE UNIVERSITY OF
MELBOURNE

2. Exercise



Thank you !





COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

Warning

This material has been reproduced and communicated to you by or on behalf of the University of Melbourne pursuant to Part VB of the *Copyright Act 1968* (*the Act*).

The material in this communication may be subject to copyright under the Act. Any further copying or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice