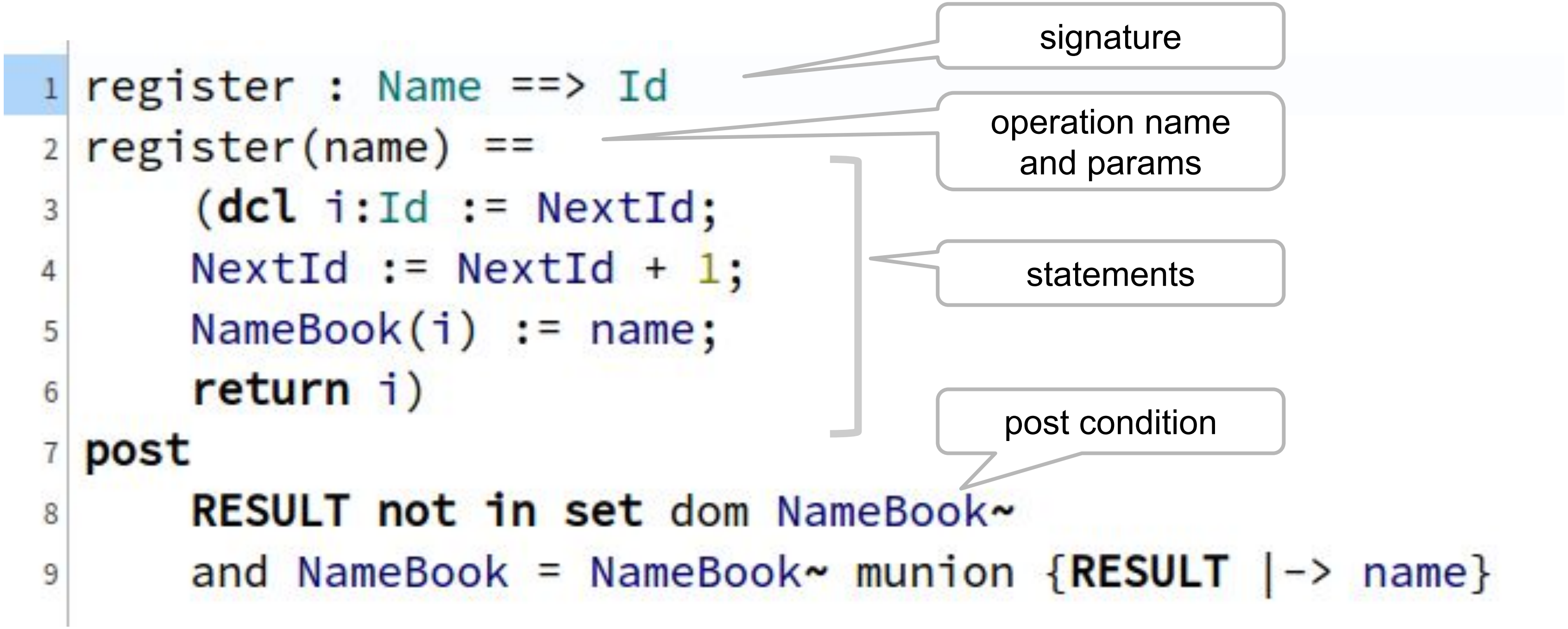
Specification Slicing for VDM-SL

Tomohiro Oda
Software Research Associates, Inc.
Han-Myung Chang
Nanzan University

Agenda

- background 1: explicit operation definitions
- background 2: program slicing
- slice extraction for VDM-SL
- slicing in ViennaTalk
- demo
- summary and future work

background 1: explicit operation definitions



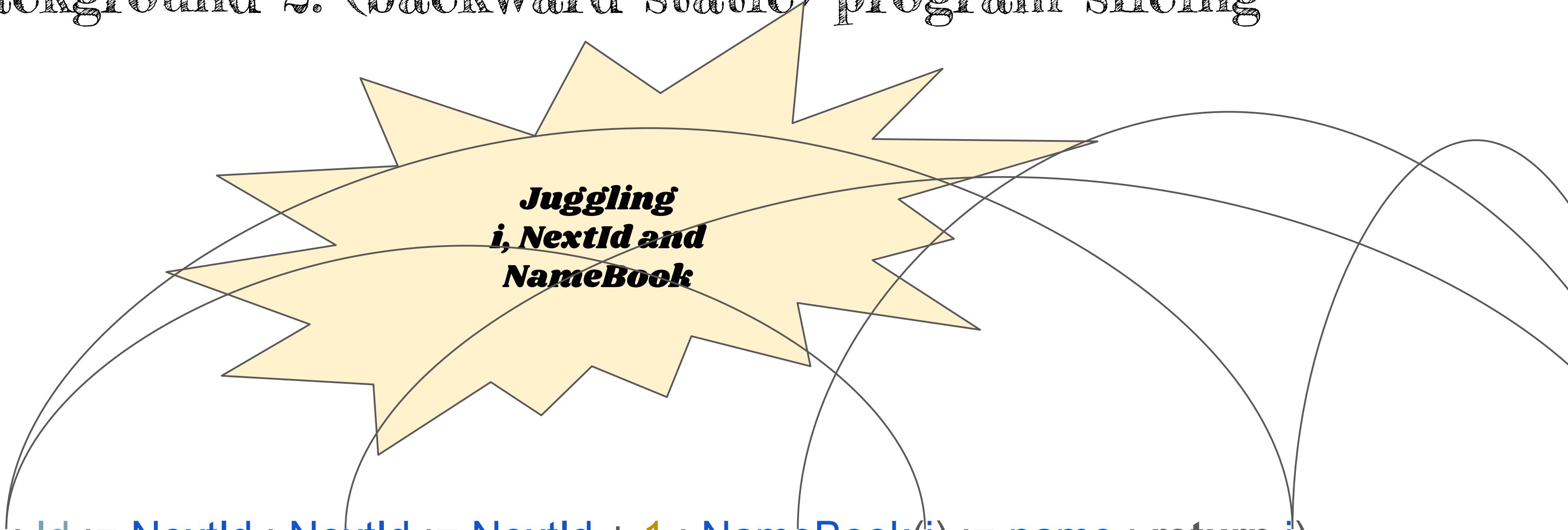
```
register : Name ==> Id
register(name) ==
      (dcl i:Id := NextId;
      NextId := NextId + 1;
      NameBook(i) := name;
      return i)
post
      RESULT not in set dom NameBook~
      and NameBook = NameBook~ munion {RESULT |-> name}
```

background 2: (backward static) program slicing

backward static slice = subset of the original source that produce the same result with regard to the value of the particular variable (slicing criterion).

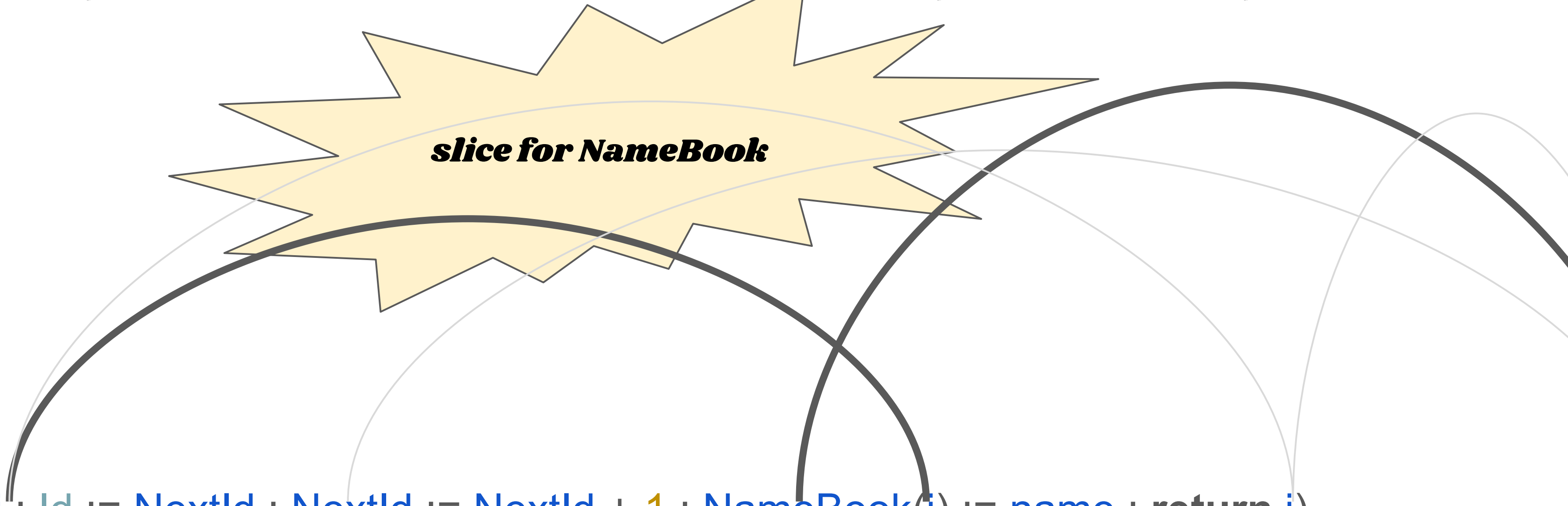
```
(dcl i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; return i)
```

background 2: (backward static) program slicing



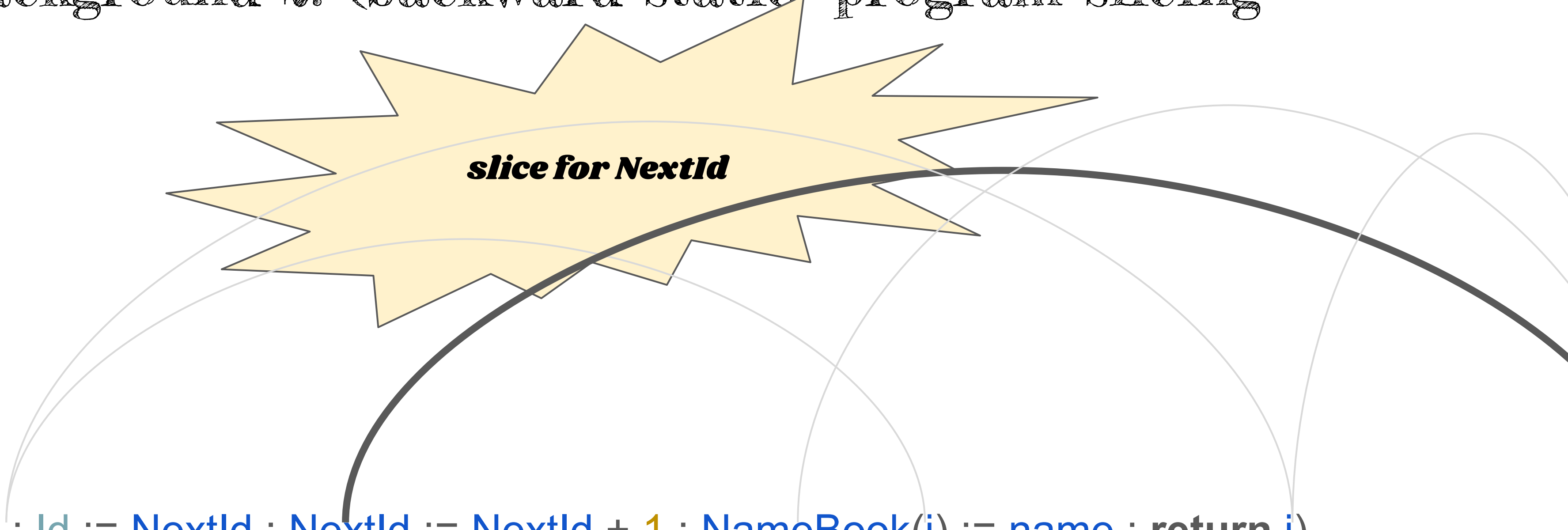
```
(dcl i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; return i)
```

background 2: (backward static) program slicing



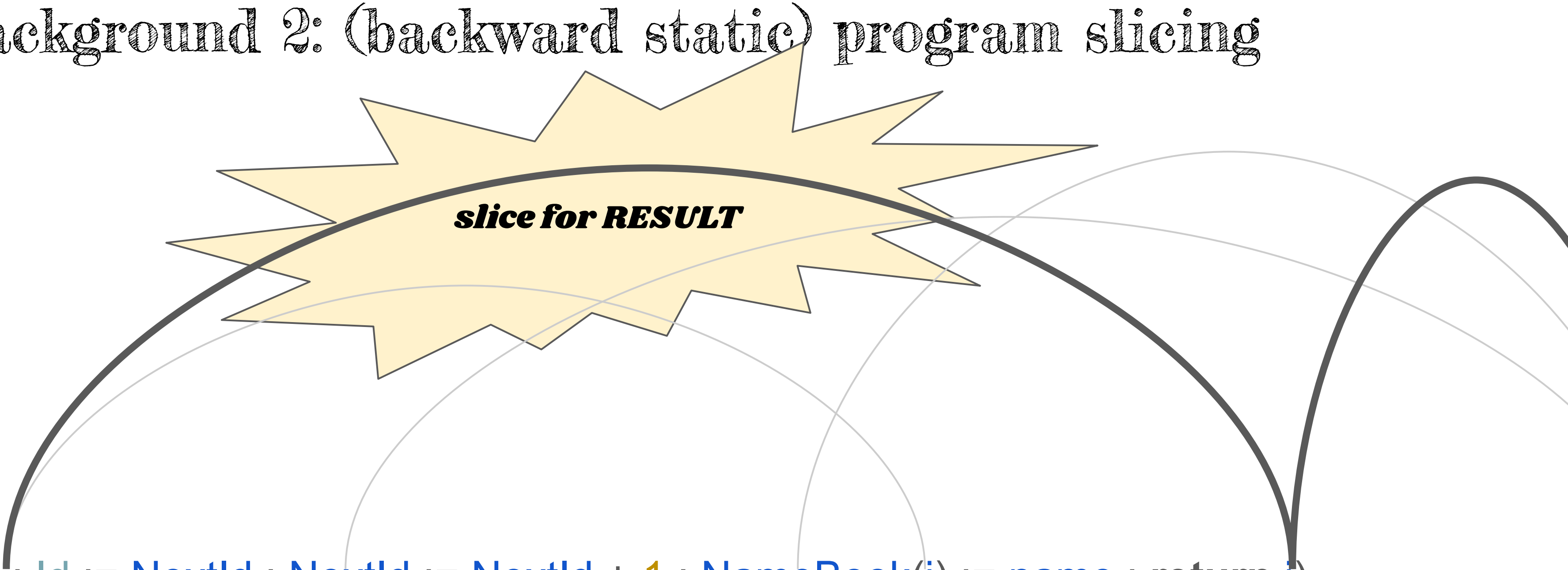
```
(dcl i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; return i)
```

background 2: (backward static) program slicing



```
(dcl i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; return i)
```

background 2: (backward static) program slicing

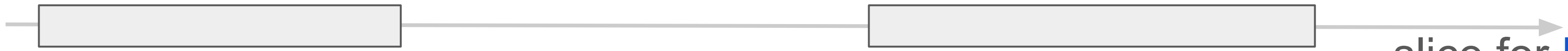


```
(dcl i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; return i)
```

background 2: (backward static) program slicing

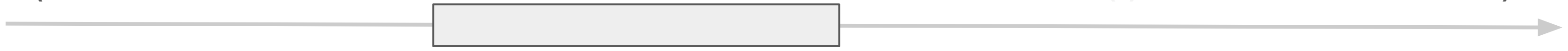
(**dcl** i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; **return** i)

(**dcl** i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; return i)



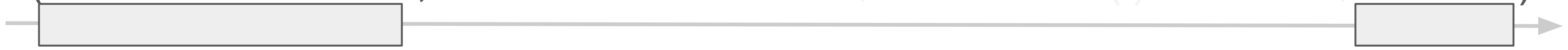
slice for NameBook

(dcl i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; return i)



slice for NextId

(**dcl** i : Id := NextId ; NextId := NextId + 1 ; NameBook(i) := name ; **return** i)



slice for **RESULT**

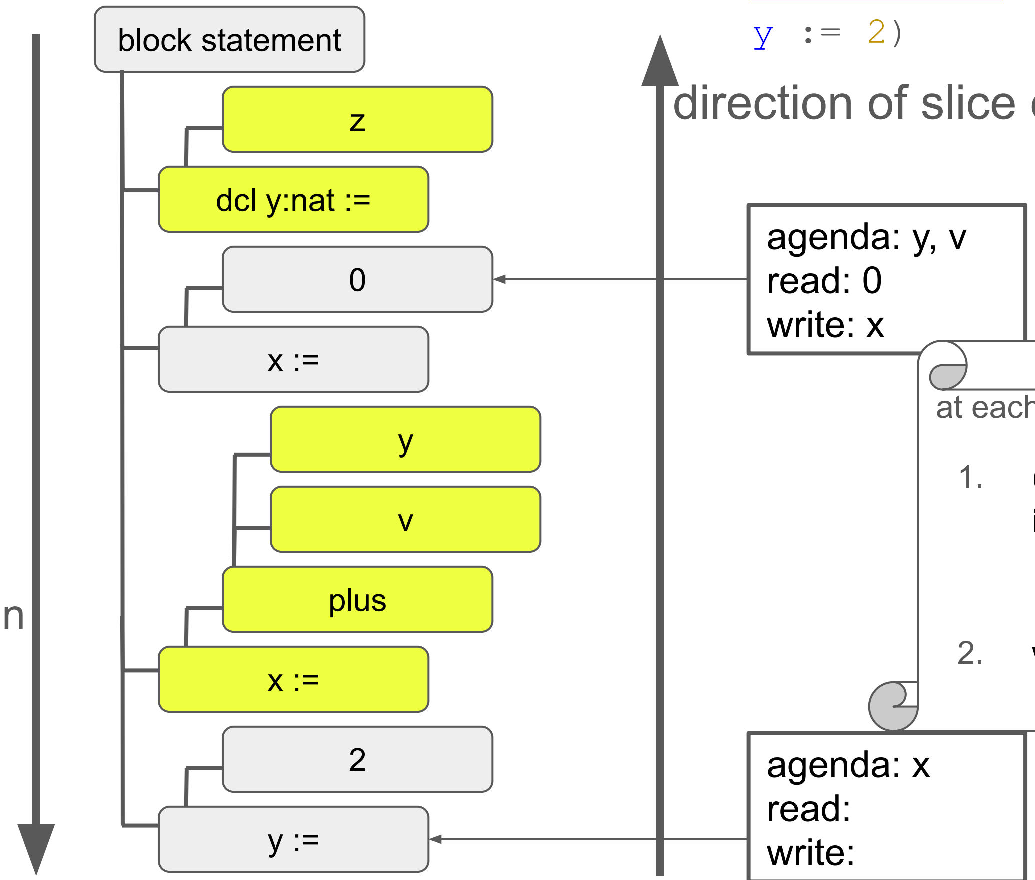
background 2: (backward static) program slicing

to extract a slice, you trace

- data dependency : write-read relationship between AST nodes
    - assignment, return, apply expression, ...
- control dependency : conditions of execution
    - if, cases, for, while, ...

slice extraction for VDM-SL

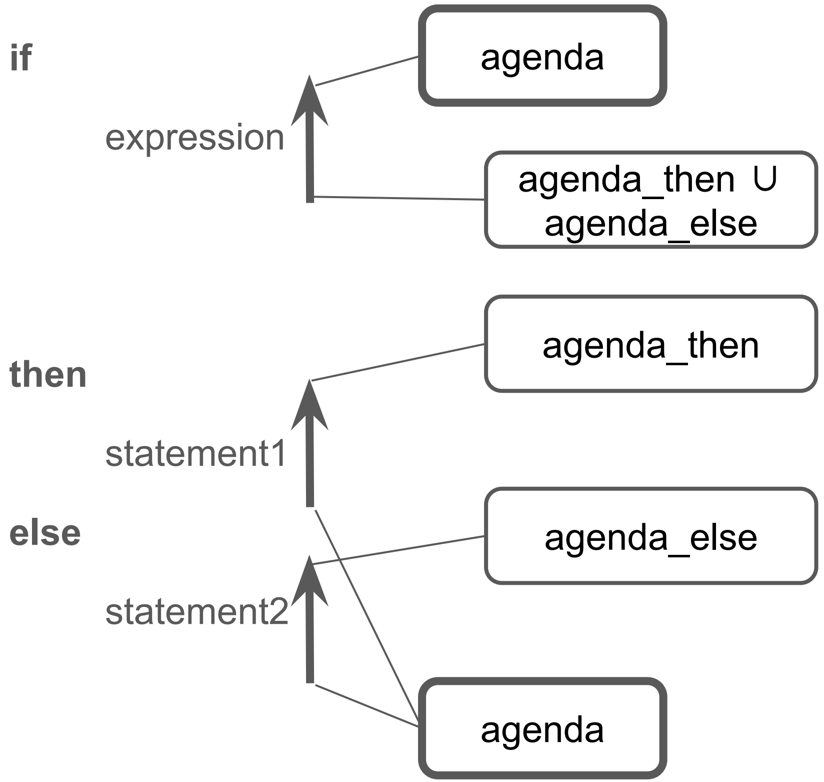
```
(dcl y:nat := z;
 x := 0;
 x := y + v;
 y := 2)
```

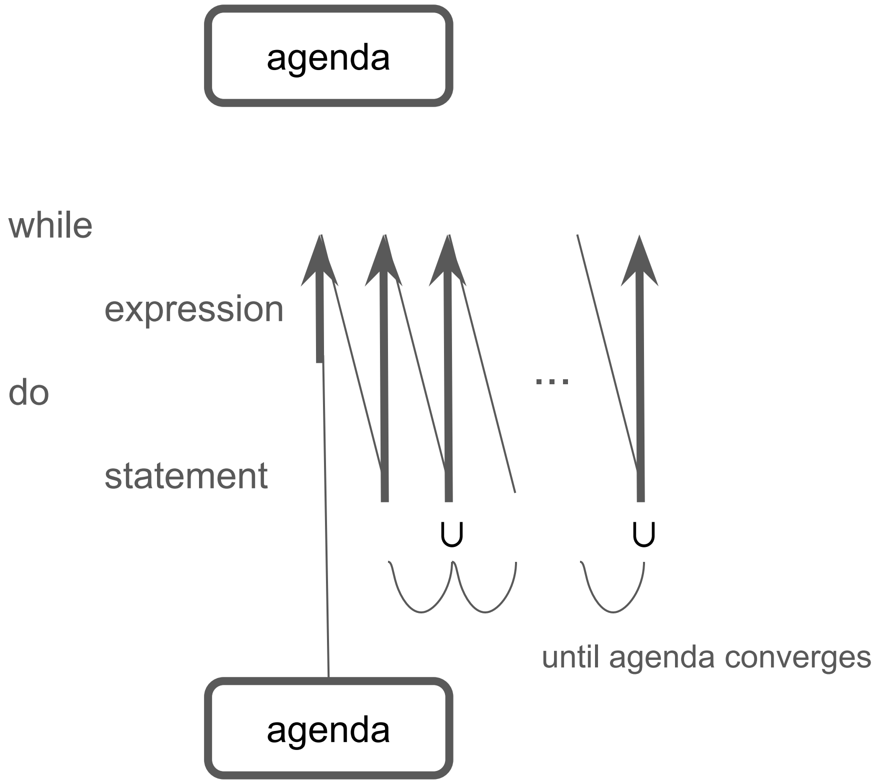


at each assignment statement,

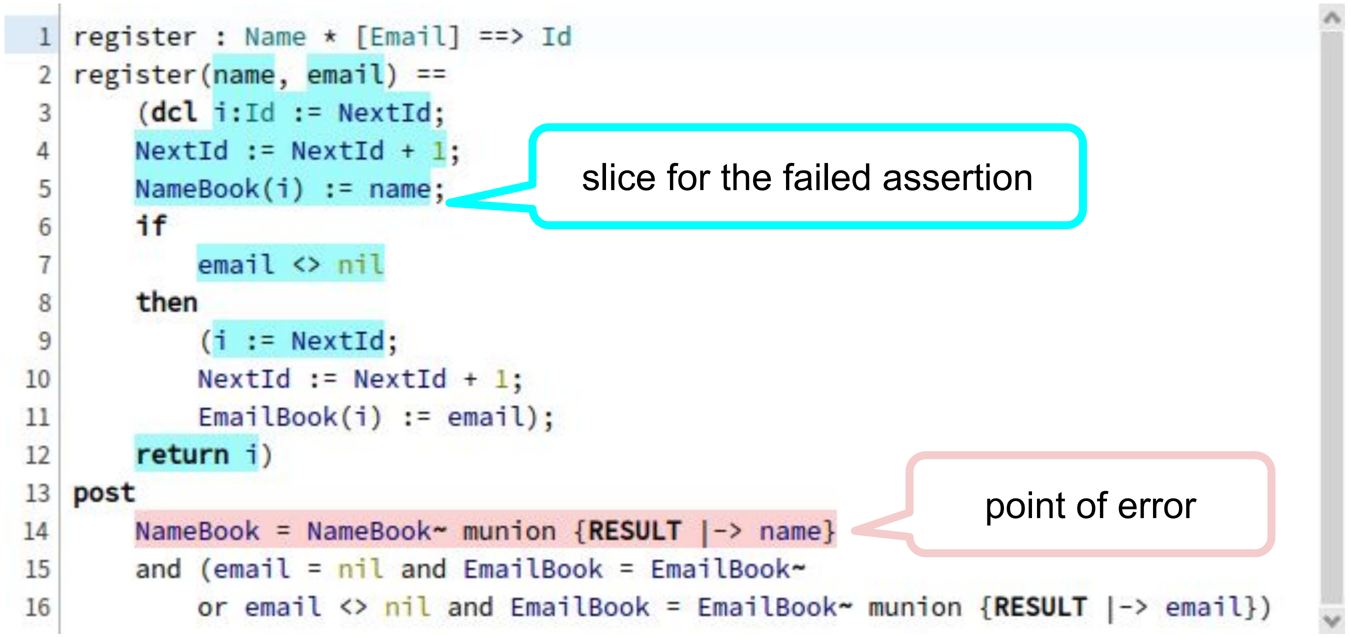
```
1.  (let c = agenda inter writes in
    if common <> {} then
        (agenda := (agenda \ c) union read;
         slice := slice union {node});
2.  writes := {};  reads := {})
```

conditionals and loops





debugging using slicing



```
register : Name * [Email] ==> Id
register(name, email) ==
    (dcl i:Id := NextId;
    NextId := NextId + 1;
    NameBook(i) := name;
    if
        email <> nil
    then
        (i := NextId;
        NextId := NextId + 1;
        EmailBook(i) := email);
    return i)
post
    NameBook = NameBook~ munion {RESULT |-> name}
    and (email = nil and EmailBook = EmailBook~
        or email <> nil and EmailBook = EmailBook~ munion {RESULT |-> email})
```

Advantage of VDM-SL in slice extraction

- the value semantics of VDM-SL makes slicing easier

  Example:

```
(dcl xs:seq of nat := [1,2,3], ys: seq of nat;
ys := xs;
ys(1) := 0;
return xs(1))
```

⇨ **RESULT = 1**
but in the most PLs
RESULT = 0

  - no aliasing ⇒ a state variable can be updated only by assignments.
  - no hidden states in lower layers or 3rd-party binary modules
  - not applicable to VDM++ and VDM-RT because of objects can be aliased.

demo

Summary and Future Work

- Applied slicing technique to explicit definitions in VDM-SL.
- ViennaTalk provides specification slicing in Browser and Debugger.

Future Work

- More applications
  - to filter testcases/traces
  - version control
- More specifications
  - implicit operation definitions
  - implicit function definitions

ViennaTalk repository: https://github.com/tomoooda/ViennaTalk

A part of this research was supported by JSPS KAKENHI Grant Number JP 23K11058 and Nanzan University Pache Research Subsidy I-A-2 for the 2024 academic year.