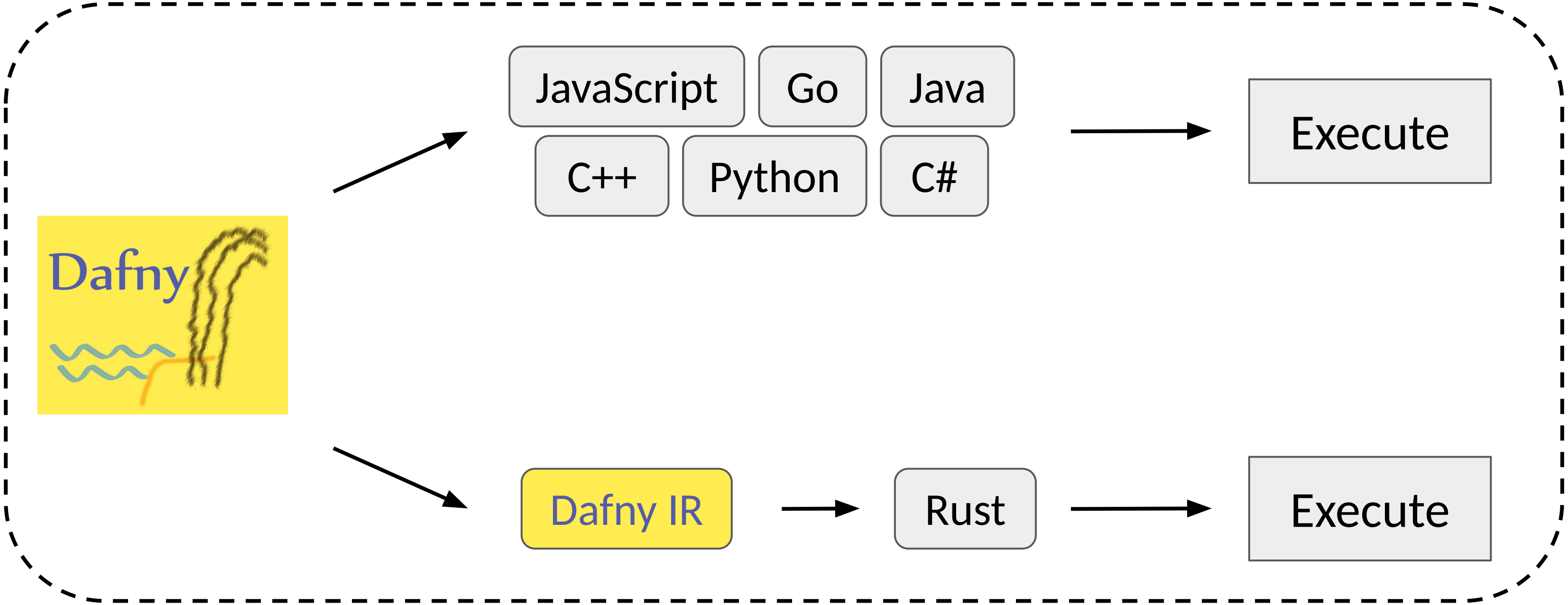
Baking for Dafny:
A CakeML Backend for Dafny

Daniel Nezamabadi, Magnus Myreen
Chalmers University of Technology and University of Gothenburg

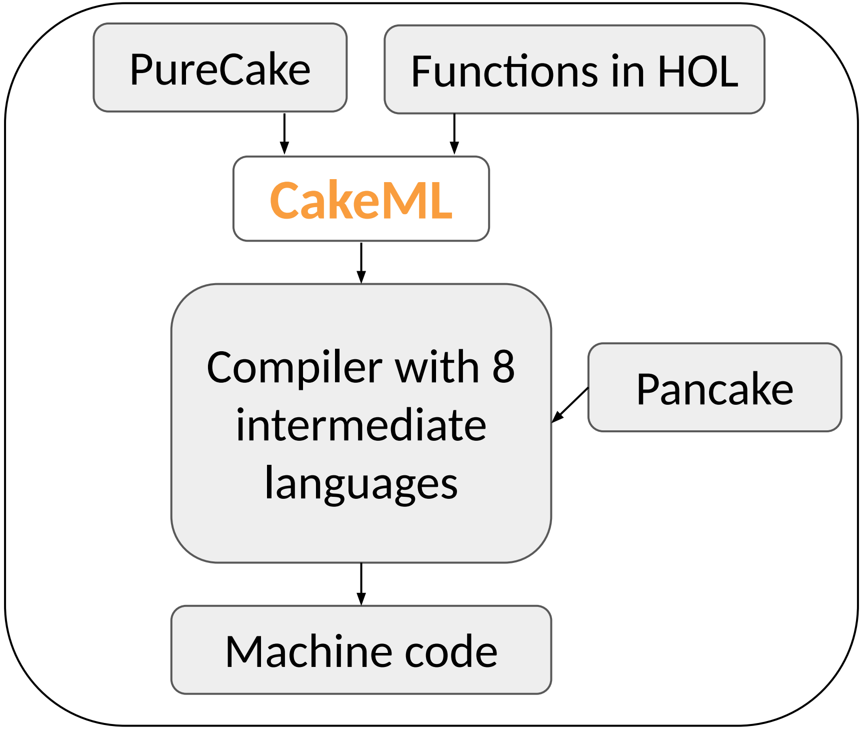
Dafny Workshop @ POPL 2025
January 19, 2025

Running Dafny Code



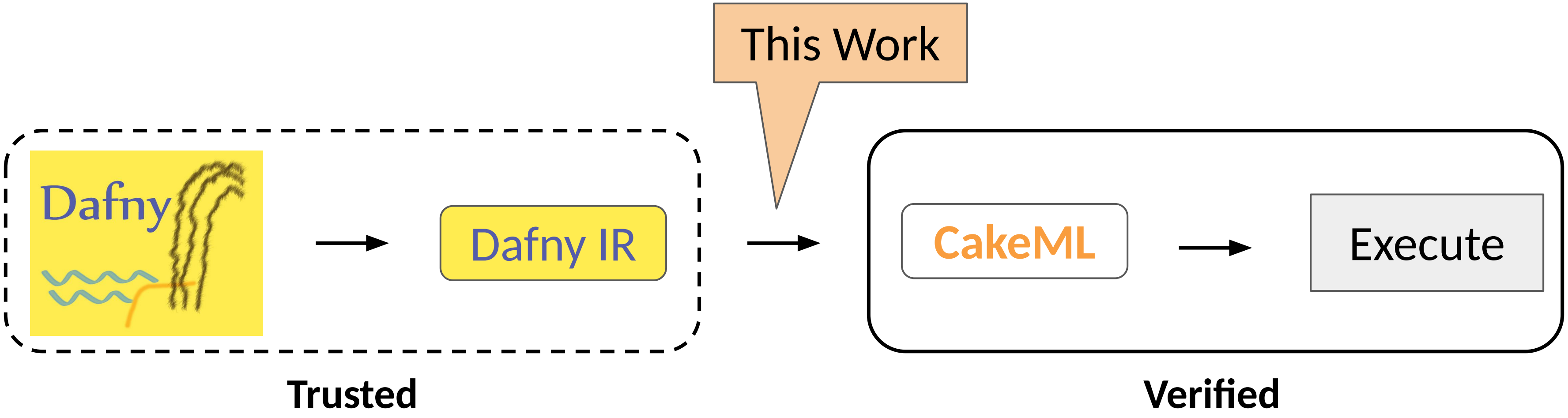
Trusted

CakeML Project

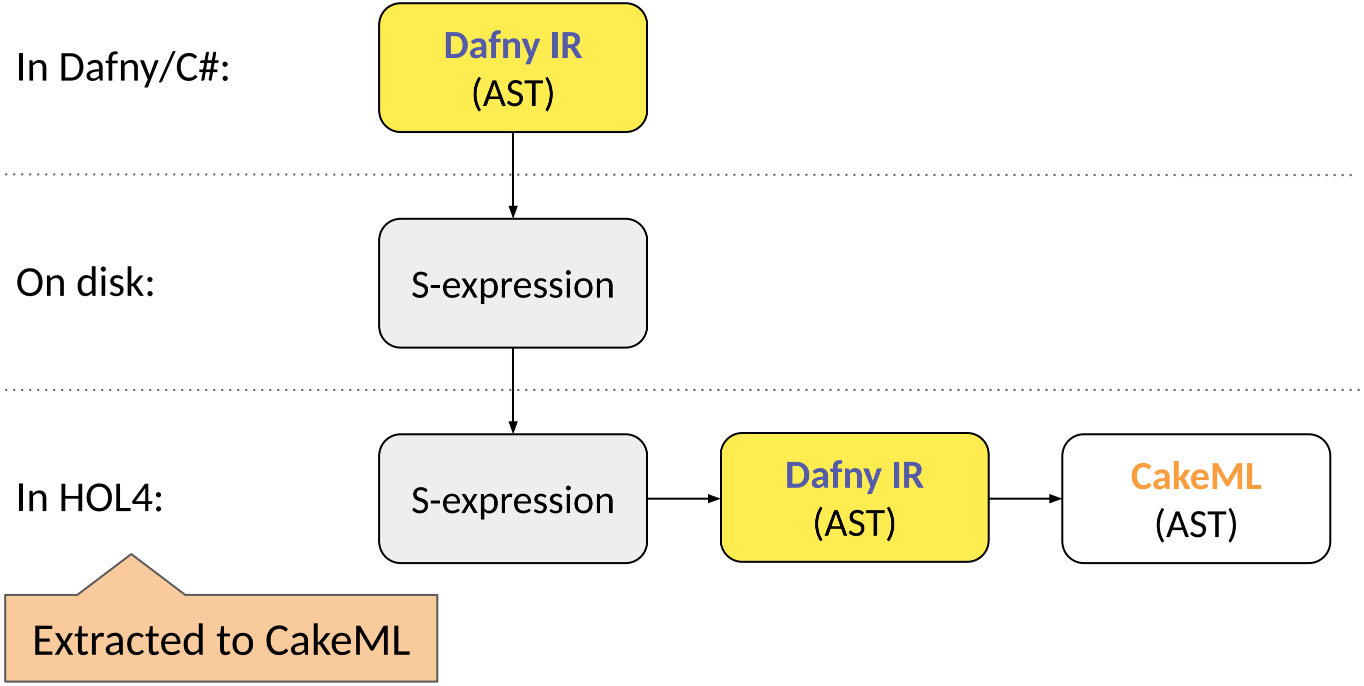


Verified in HOL4

Dafny to CakeML Compiler



Dafny to CakeML Compiler (Zoomed In)



Compiling “Hello, Cake”

```
method Main() {
  print "Hello, Cake\n";
}
```

Dafny

Return is modeled using exceptions

```
structure  _module = struct
 fun  Main  =
   (print (String.implode ("Hello, Cake\n");
    raise  Dafny.Return)
   handle  Dafny.Return  =>  ();
end
```

Module for helpers

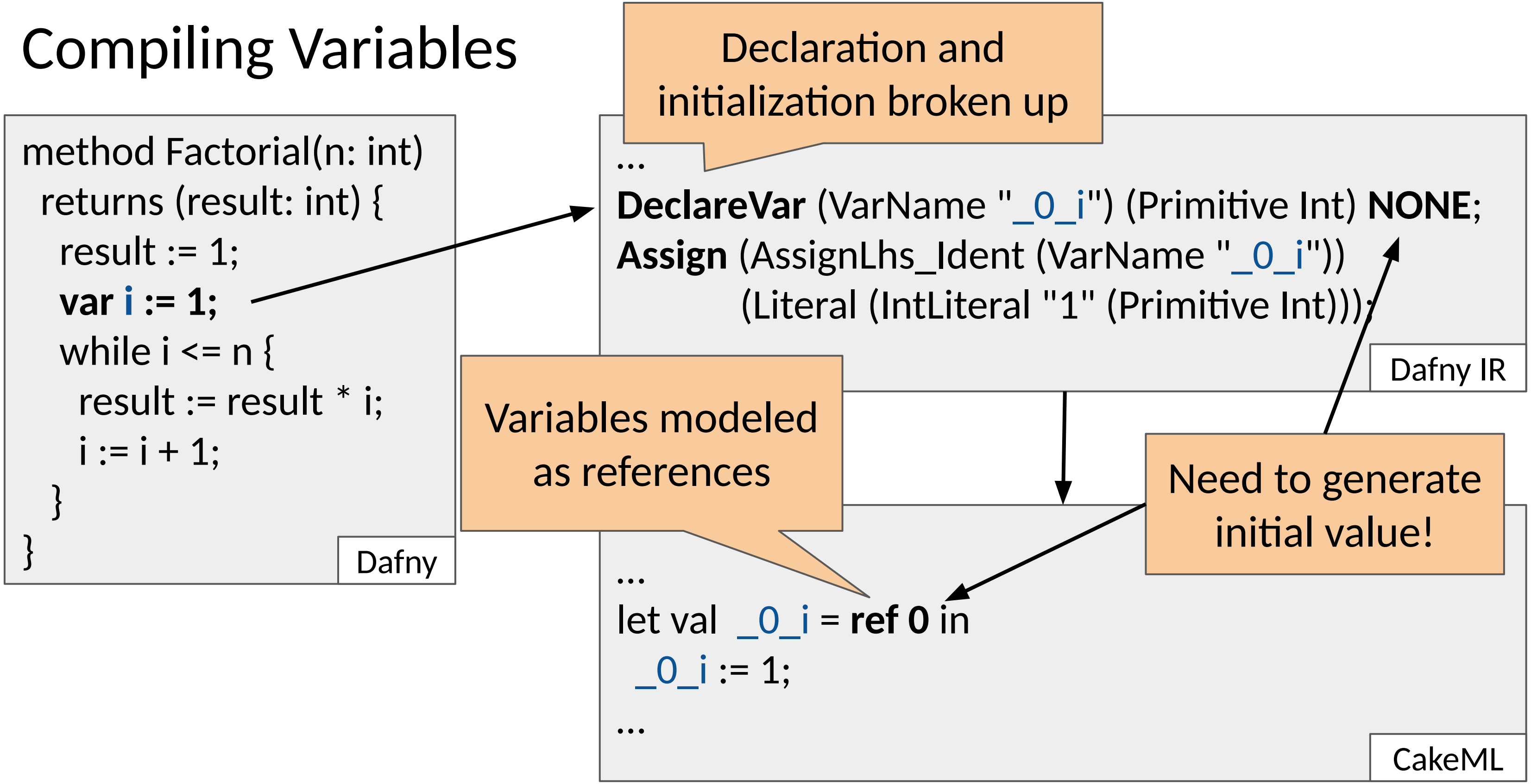
CakeML

```
[Module (Name "_module") [] F
   (SOME
     [ModuleItem_Class
        (Class (Name "__default") (Ident (Name "_module")) [] [] []
           [ClassItem_Method
              (Method T T F F NONE (Name "Main") [] []
                [Print (Literal (StringLiteral "Hello, Cake\\n" F));
                 EarlyReturn]
                [] (SOME []))] [])])]
```

Dafny IR*

* details may have changed

Compiling Variables



```
method Factorial(n: int)
  returns (result: int) {
    result := 1;
    var i := 1;
    while i <= n {
      result := result * i;
      i := i + 1;
    }
}
```

Dafny

Declaration and initialization broken up

```
...
DeclareVar (VarName "_0_i") (Primitive Int) NONE;
Assign (AssignLhs_Ident (VarName "_0_i"))
         (Literal (IntLiteral "1" (Primitive Int)));
```

Dafny IR

Variables modeled as references

Need to generate initial value!

```
...
let val  _0_i = ref 0 in
  _0_i := 1;
...
```

CakeML

Compiling Loops

```
method Factorial(n: int)
 returns (result: int) {
  result := 1;
  var i := 1;
  while i <= n {
   result := result * i;
   i := i + 1;
  }
}
```

Dafny

```
While cond body
```

Dafny IR

Encode as recursive function

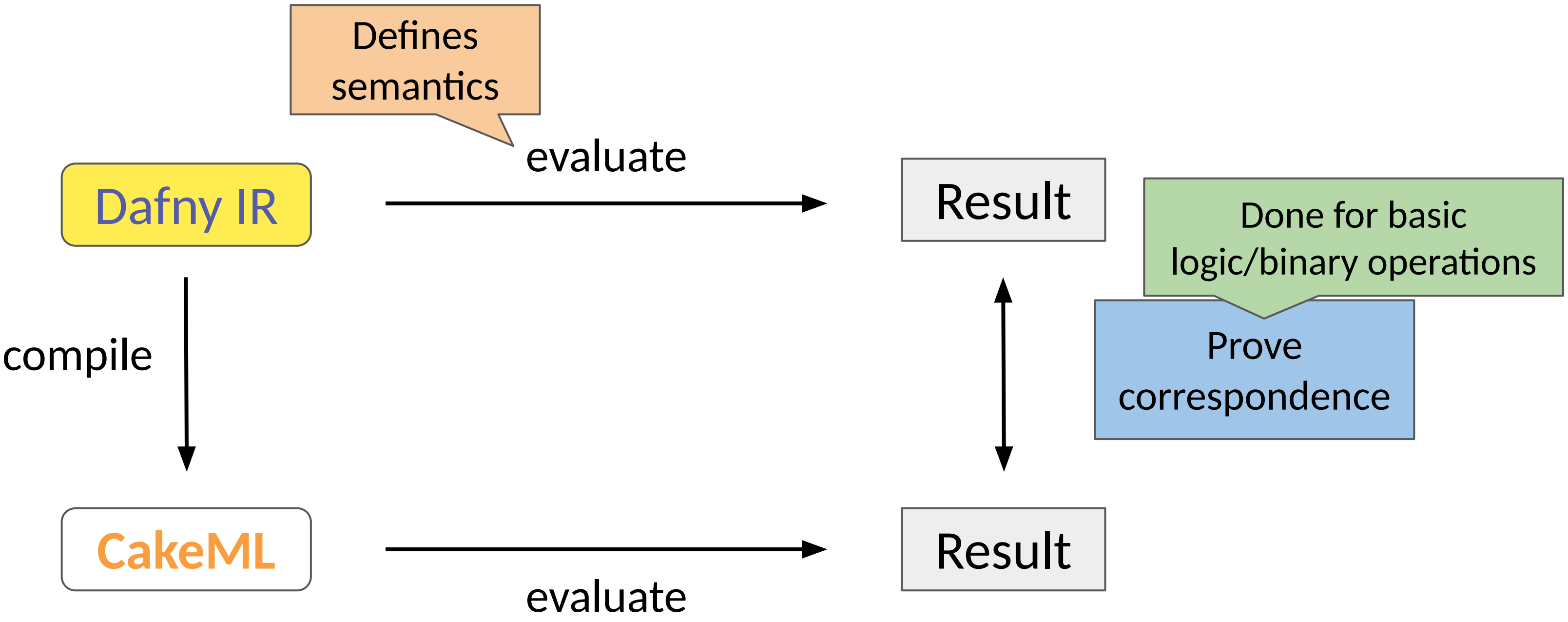
```
...
let  fun  CML_while_0  =
 if not ((! n) < (! _0_i)) then
  (result := (! result) * (! _0_i);
   _0_i := !( _0_i) + 1;
   CML_while_0 ())
 else  ()
in
 (CML_while_0 ()) handle Dafny.Break  =>  ()
...
```

CakeML

Support for break statements*

* this one should not be generated or optimized away

Proving Correctness (Ongoing Work)


Defines semantics

evaluate

Dafny IR

Result

Done for basic logic/binary operations

compile

Prove correspondence

CakeML

Result

evaluate

Limitations, Challenges and Future Work

Missing support for

- Synonym types
- Data types
- Type arguments
- Traits/Classes
- (Multi)Sets, Maps
- Tail recursion
- Foreach/Range
- Bitwise operations
- Quantifiers
- Extern
- Unicode

…

How to initialize variables?

May need something like vtables

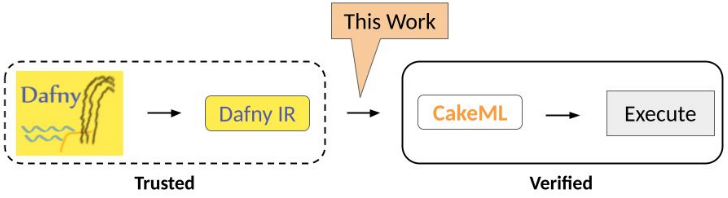
Future Work

Improve feature support

Prove correctness of compiler

Verify translation to Dafny IR

Dafny to CakeML Compiler



Compiling “Hello, Cake”

```
method Main() {
  print "Hello, Cake\n";
}
```

Dafny

```
structure _module = struct
 fun Main =
  (print (String.implode ("Hello, Cake\n");
   raise Dafny.Return)
  handle Dafny.Return => ();
end
```

CakeML

Return is modeled using exceptions

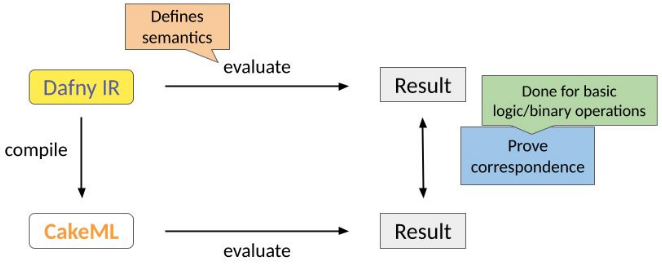
Module for helpers

```
[Module (Name "_module") [] F
  (SOME
    [ModuleItem_Class
      (Class (Name "__default") (Ident (Name "_module")) [] [] []
        [ClassItem_Method
          (Method T T F F NONE (Name "Main") [] []
            [Print (Literal (StringLiteral "Hello, Cake\\n" F));
             EarlyReturn]
            [] (SOME []))] [])]))]
```

Dafny IR*

* details may have changed

Proving Correctness



Limitations, Challenges and Future Work

Missing support for

- Synonym types
- Data types
- Type arguments
- Traits/Classes
- (Multi)Sets, Maps
- Tail recursion
- Foreach/Range
- Bitwise operations
- Quantifiers
- Extern
- Unicode

...

How to initialize variables?

May need something like vtables

Future Work

Improve feature support

Prove correctness of compiler

Verify translation to Dafny IR

Code: https://github.com/CakeML/cakeml/tree/master/compiler/dafny