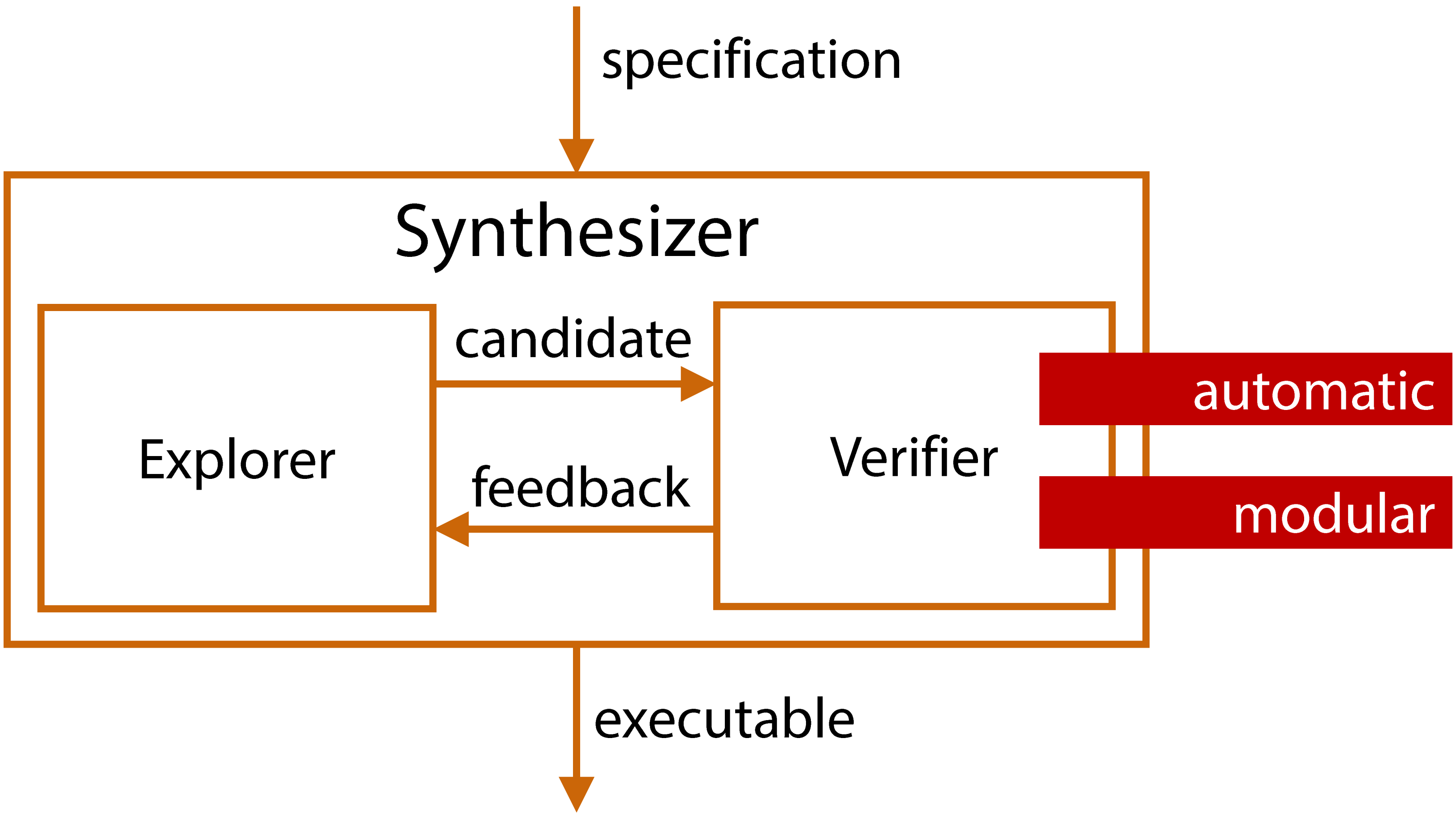
Programs Synthesis from Refinement Types

Nadia Polikarpova
Ivan Kuraj
Armando Solar-Lezama
(MIT CSAIL)

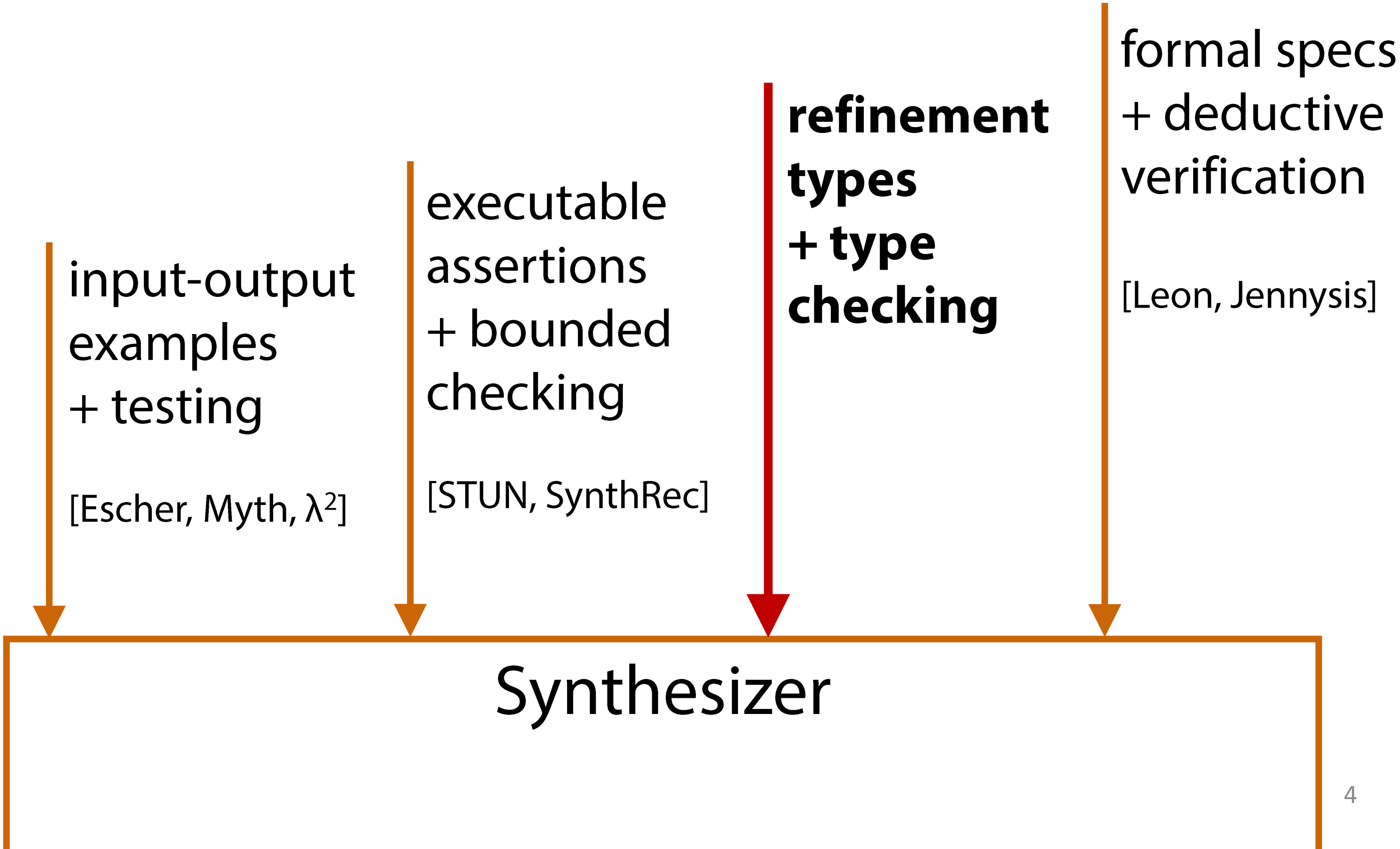
CSAIL

Program Synthesis



Modular Verification for Synthesis

Synthesis of Functional Programs



Refinement Types

Nat

base types

max :: x: Nat → y: Nat → { ν: Nat | x ≤ ν ∧ y ≤ ν }

dependent function types

Cons x xs :: { ν: List Nat }

polymorphic datatypes

data List α **where**
  Nil :: ν: { List α | *len* ν = 0 }
  Cons :: x: α → xs: List α
      → ν: { List α | *len* ν = *len* xs + 1 }

measure *len* :: List α → Int
  len Nil = 0
  len (Cons _ xs) = *len* xs + 1

[Rondon et al.'08, Kawaguchi et al.'09]

Example 1: replicate

measure len :: List α -> Int
data List α **where**
  Nil :: {ν: List α | len ν = 0}
  Cons :: x: α → xs: List α → {ν: List α | len ν = len xs + 1}

zero :: {ν: Int | ν = 0}
inc :: x: Int → {ν: Int | ν = x + 1}
dec :: x: Int → {ν: Int | ν = x - 1}

replicate :: n: Nat → x: α → {ν: List α | len ν = n}
replicate = ??

Example 2: insert

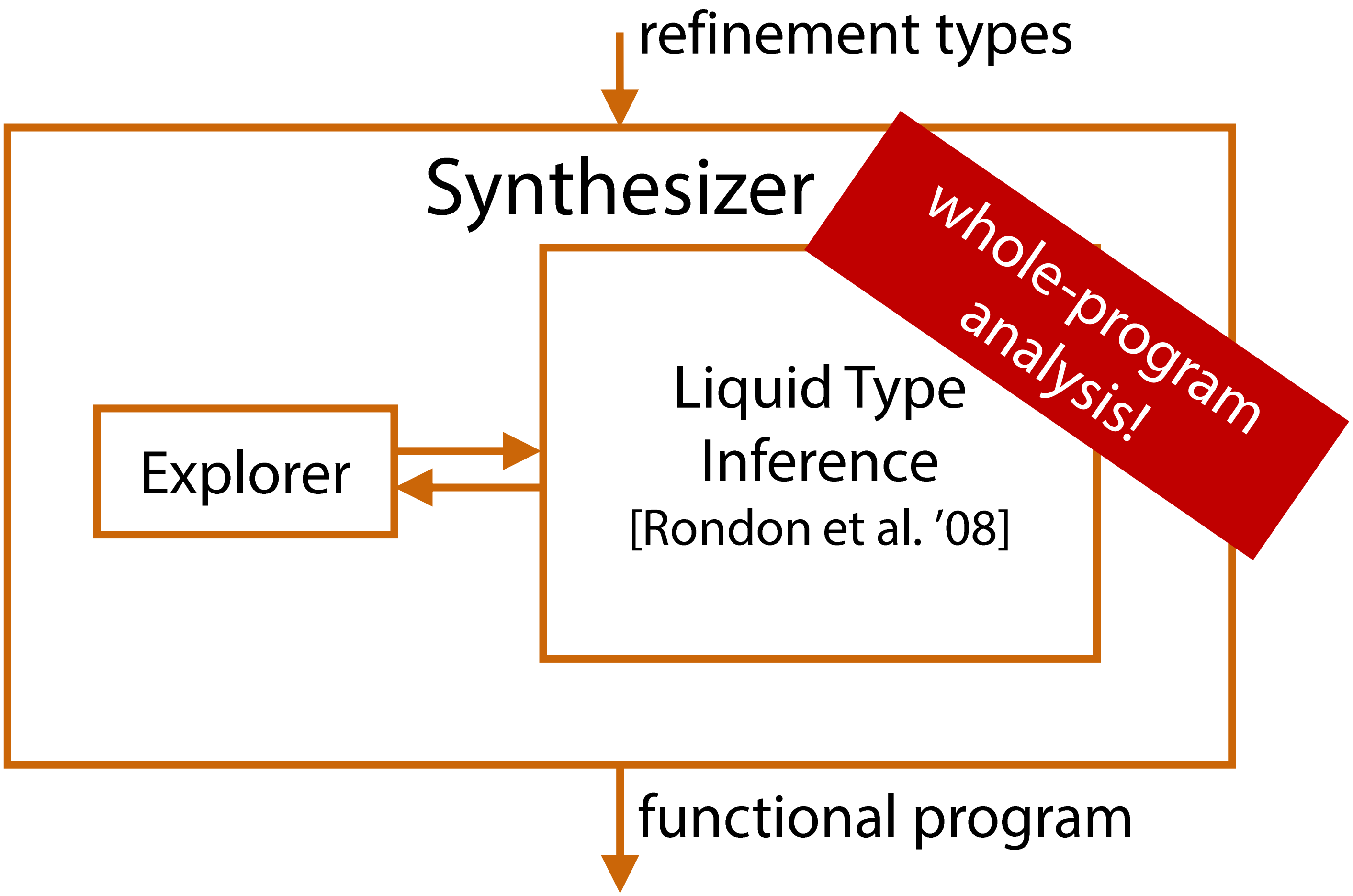
```
measure elems :: BST α -> Set α
data BST α where
  Empty :: {ν: BST α | elems ν = []}
  Node :: x: α ->
    l: BST {α | ν < x} ->
    r: BST {α | x < ν} ->
    {ν: BST α | elems ν = elems l + elems r + [x]}

insert :: x: α -> t: BST α -> {ν: BST α | elems ν = elems t + [x]}
insert = ??
```

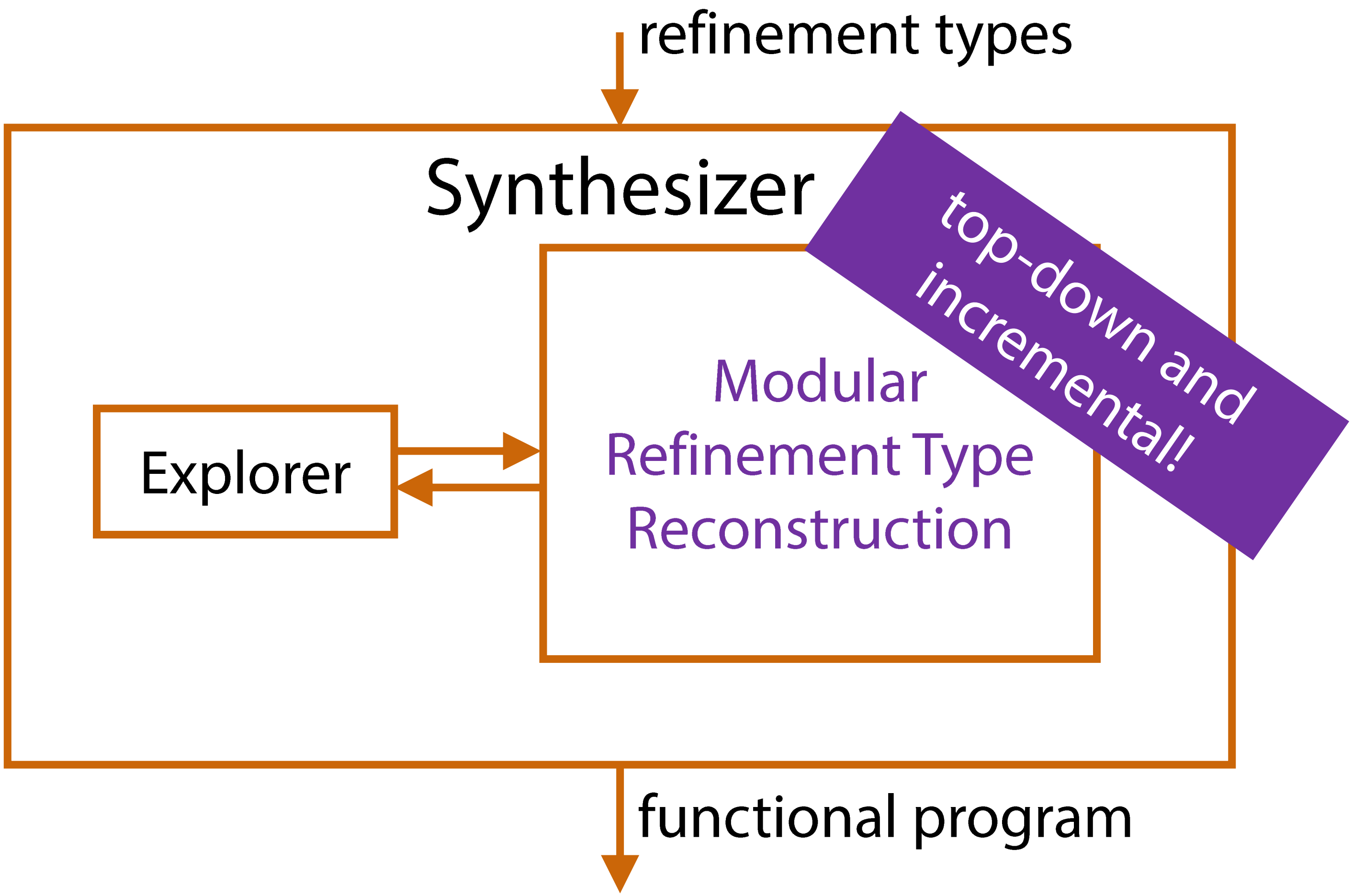
Example 2: insert (solution)

insert x t = **match** t **with**
  Empty ⟶ Node x Empty Empty
  Node y l r ⟶ **if** y = x
    then t
    else if y ≤ x
      then Node y l (insert x r) :: BST { ν: α | ν > y }
      else Node y (insert x l) r

Synthesis from Refinement Types: Take One



Synthesis from Refinement Types: Our Solution



Modular Verification for Synthesis

Evaluation

Lists

take, drop, delete, zip with function, reverse, de-duplicate, fold, length/append with fold, ...

Sorting

insertion sort, merge sort, quick sort

Binary Search Trees

member, insert, delete

User datatypes

AST desugaring

Red-black trees

rotation

50 benchmarks

< 5 s

19 s

Synquid

bitbucket.org/nadiapolikarpova/synquid