Lecture 7: Component-based Synthesis

Yu Feng
Spring 2021

Summary of previous lecture

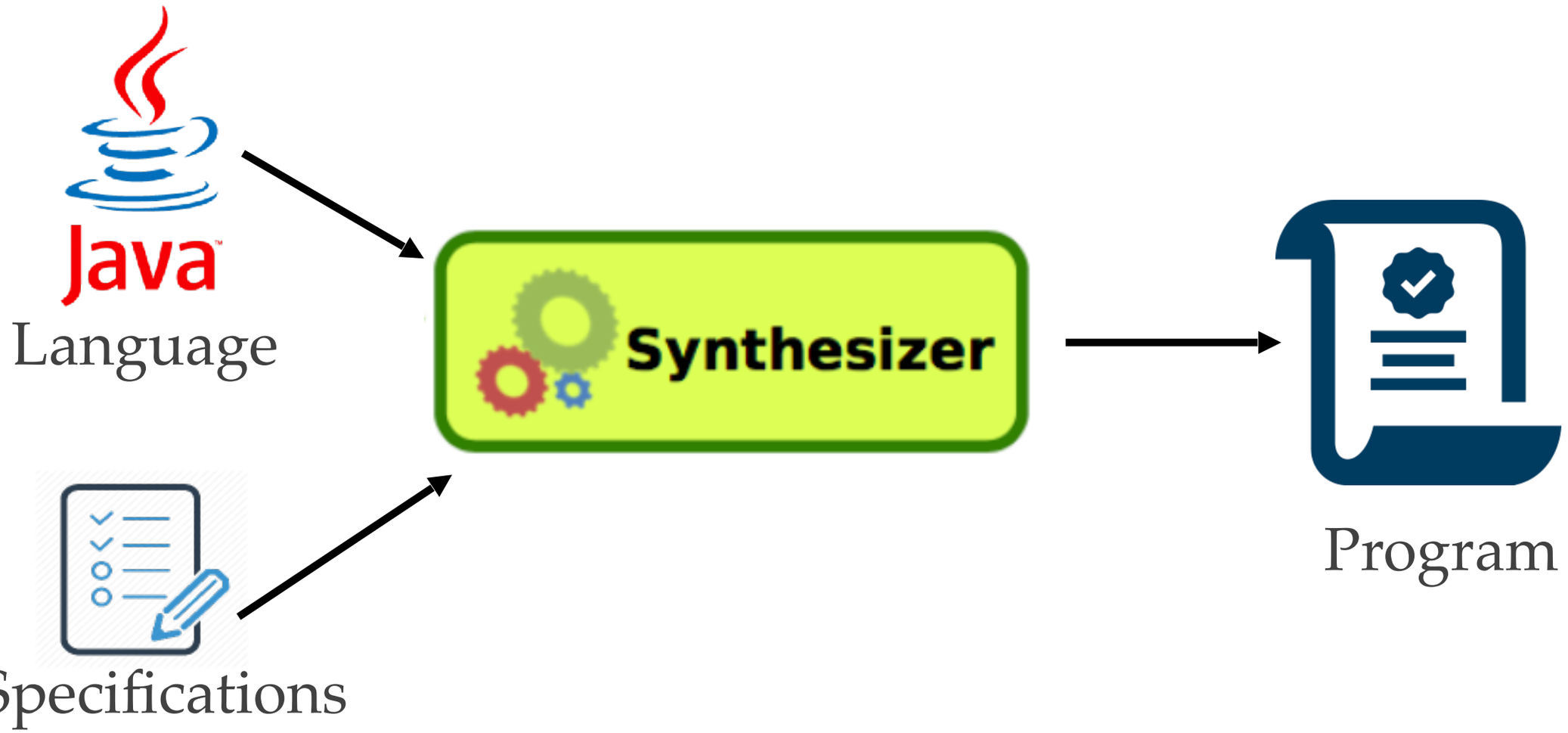
- HW1 is due today
- HW2 is out
- Inductive program synthesis
  - Enumerative search
  - Symbolic search
  - Neural-guide search

Outline of this lecture

- Component-based synthesis

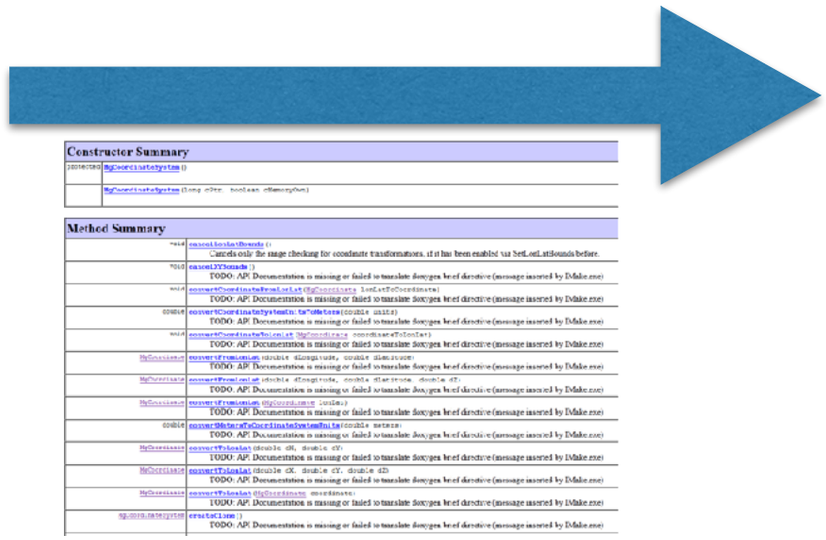
Component-Based Synthesis for Complex APIs, POPL'17

What is program synthesis



Automatically generate programs satisfying high-level specs

Programming with API in reality



Constructor Summary

20050580	MyCoordinateSystem ()
	MyCoordinateSystem (long cPtr, boolean cMemoryOwn)

Method Summary

void	cancelBoundsCheck () Cancels only the range checking for coordinate transformations, if it has been enabled via SetLonLatBounds before.
TOLO	cancelXYBounds () TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
void	convertCoordinateFromLatLon (MyCoordinate lonLatToCoordinate) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
double	convertCoordinateSystemIn: toMeters (double units) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
void	convertCoordinateToLonLat (MyCoordinate coordinateToLonLat) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
MyCoordinate	convertFromLonLat (double dLongitude, double dLatitude) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
MyCurrentLatLon	convertFromLonLat (double dLongitude, double dLatitude, double dT) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
MyCoordinate	convertFromLonLat (MyCoordinate lonLat) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
double	convertMetersToCoordinateSystemUnits (double meters) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
MyCoordinate	convertToLonLat (double dN, double dV) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
MyCoordinate	convertToLonLat (double dX, double dY, double dZ) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
MyCoordinate	convertToLonLat (MyCoordinate coordinate) TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)
egismo.mycrypto	createClone () TODO: API Documentation is missing or failed to translate doxygen brief directive (message inserted by Doxlate.exe)

Send HTTP request
Compute GCD
Rotate an image

Programmers are spending tons of effort learning APIs!

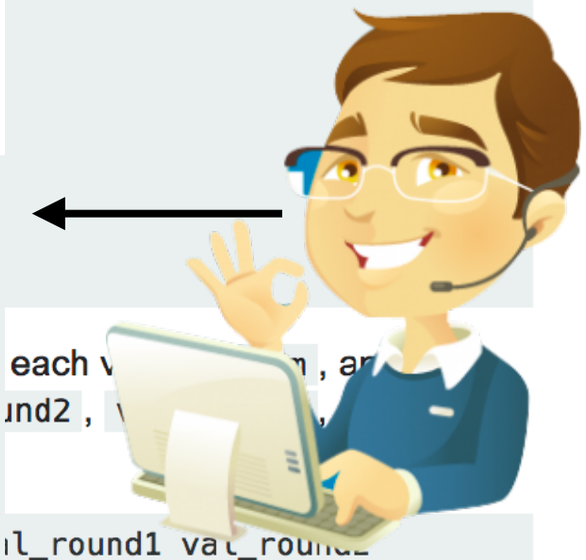
What do they do? Ask for help

I have a data frame which is structured like this one:

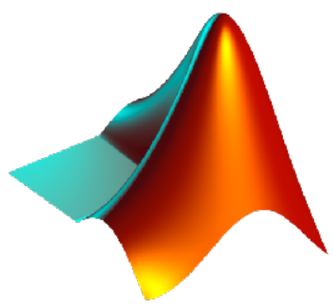
```
dd <- data.frame(round = c("round1", "round2", "round1", "round2"),
                 var1 = c(22, 11, 22, 11),
                 var2 = c(33, 44, 33, 44),
```

We can use `tidyr/dplyr` to do this. We `gather` the dataset to 'long' format, `unite` the 'variable' and 'round' to create 'var' and then `spread` to 'wide' format.

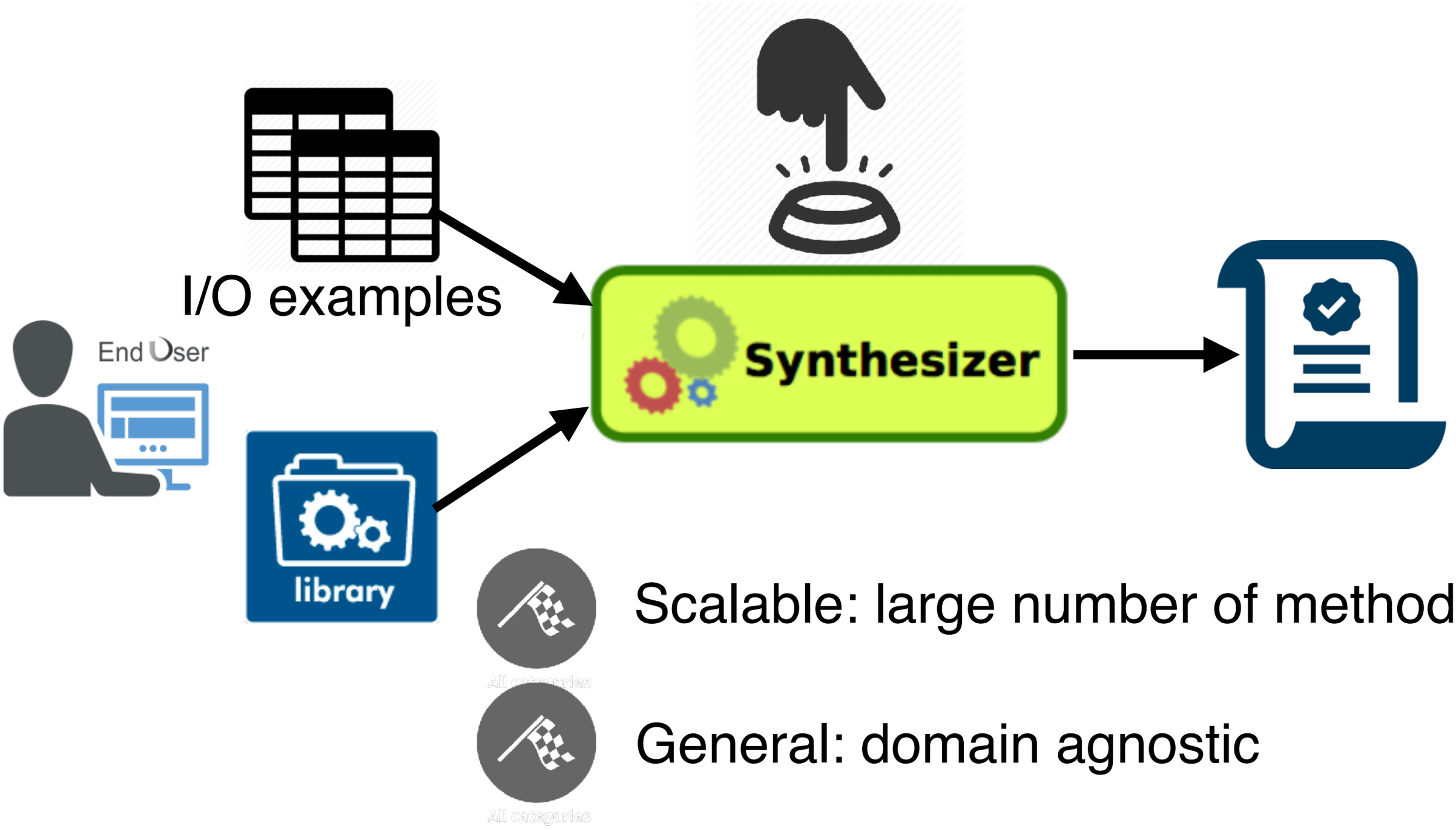
```
library(dplyr)
library(tidyr)
gather(dd, variable, value, var1, var2, val) %>%
        unite(var, variable, round) %>%
        spread(var, value)
#  nam val_round1 val_round2 var1_round1 var1_round2 var2_round1 var2_round2
#1 bar  0.7187271  0.6022287          22          11          33          44
#2 foo  0.2672339  0.7199101          22          11          33          44
```



```
1 foo          22          11          33          44 0.32995729  0.8921504
2 bar          22          11          33          44 0.09213526  0.8264472
```



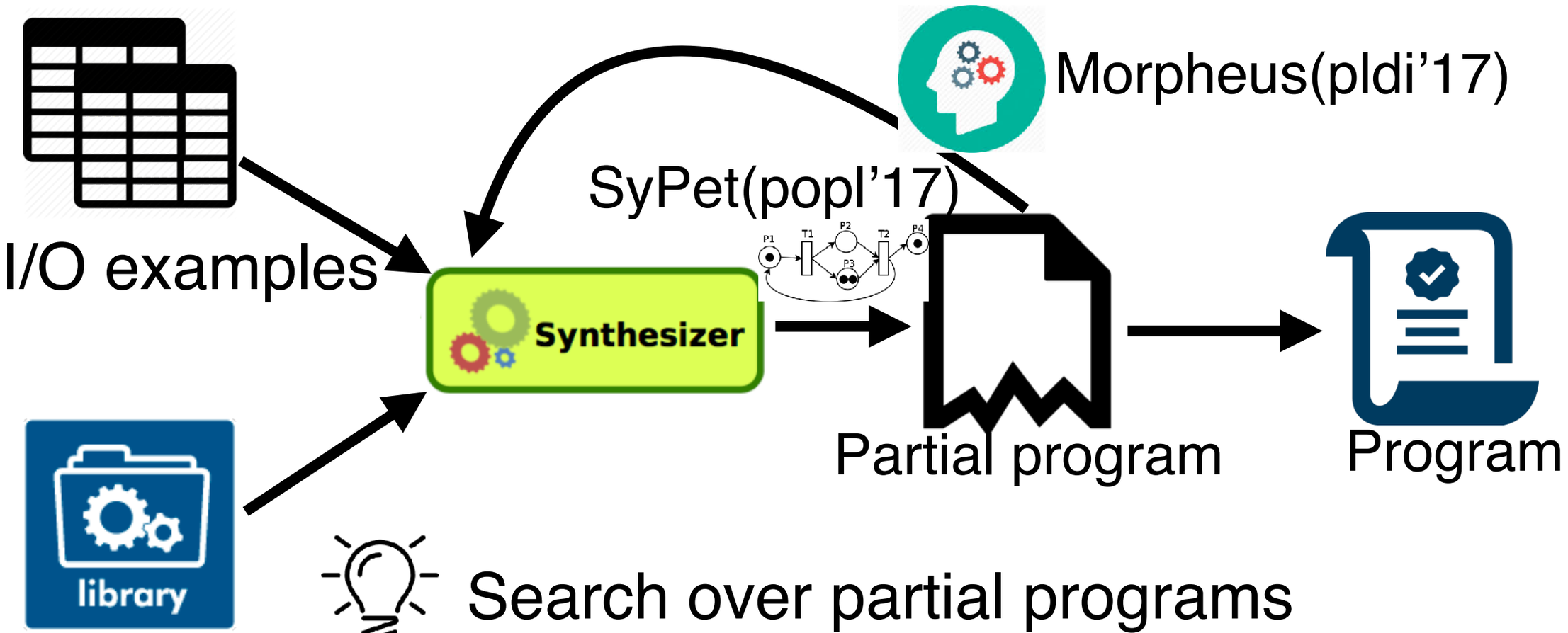
A general synthesizer



Scalable: large number of methods

General: domain agnostic

Key insight



- Search over partial programs
- Prune via types
- Prune via logical deduction

A motivating example

- Consider rotating some object using a Java API

```
Area rotate(Area obj, Point 2D pt, double angle)
```

- Possible to do this using java.awt.geom, but not trivial:

```
AffineTransform at = new AffineTransform();

double x = pt.getX();

double y = pt.getY();

at.setToRotation(angle, x, y);

Area tr = obj.createTransformedArea(at);

return tr;
```

Must know this class

Deconstruct Point object

Call impure method

Finally get the object

Challenge

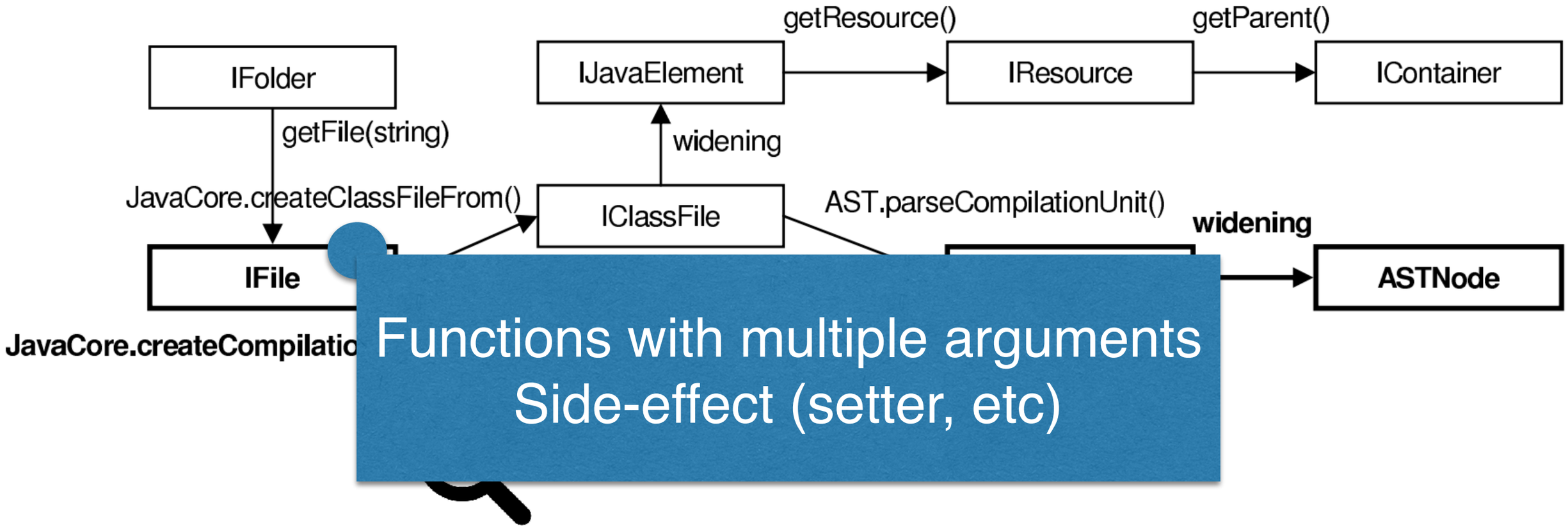
Large search space: for java.awt.geom library that has 700+ methods, > 5m candidates of size 3

Prune invalid programs: using types of APIs as the coarse-grained specs

```
T2 foo(T1 x) {
    …
}
```

How to enumerate well-typed programs

Synthesis as graph reachability




```
IFile file = ...;
ICompilationUnit cu =
  JavaCore.createCompilationUnitFrom(file);
ASTNode ast = AST.parseCompilationUnit(cu, false);
```

Need a more general graph representation!

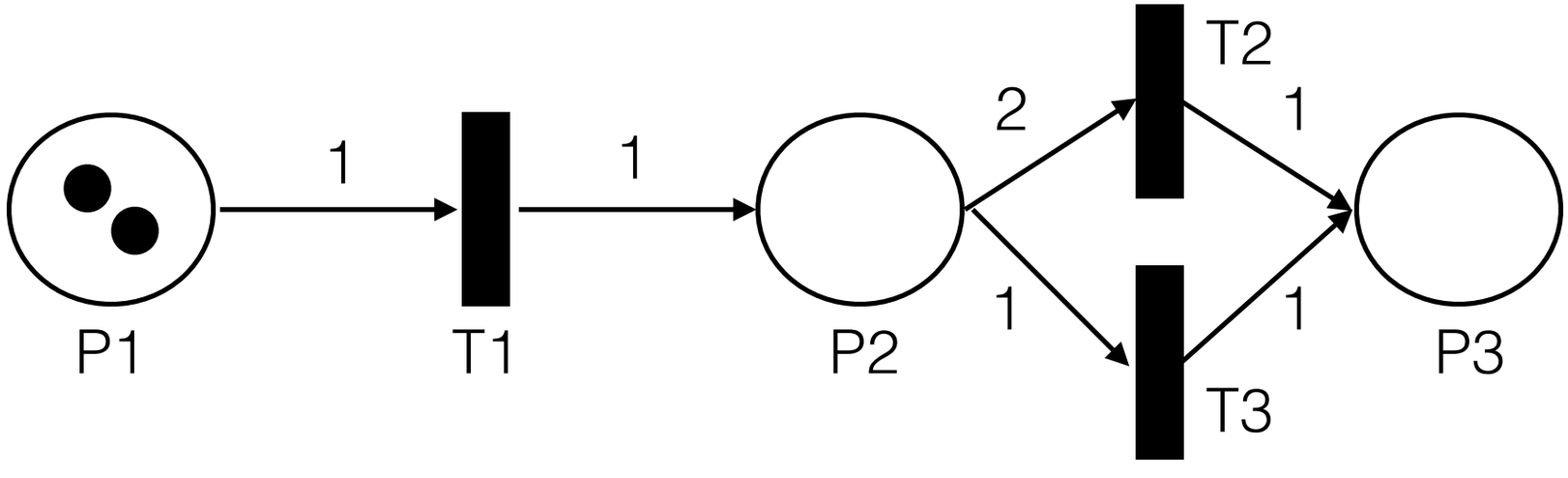
Mandelin, David, et al. "Jungloid mining: helping to navigate the API jungle." PLDI'05

Our solution

Use Petri net reachability analysis to look for well-typed programs of the desired type

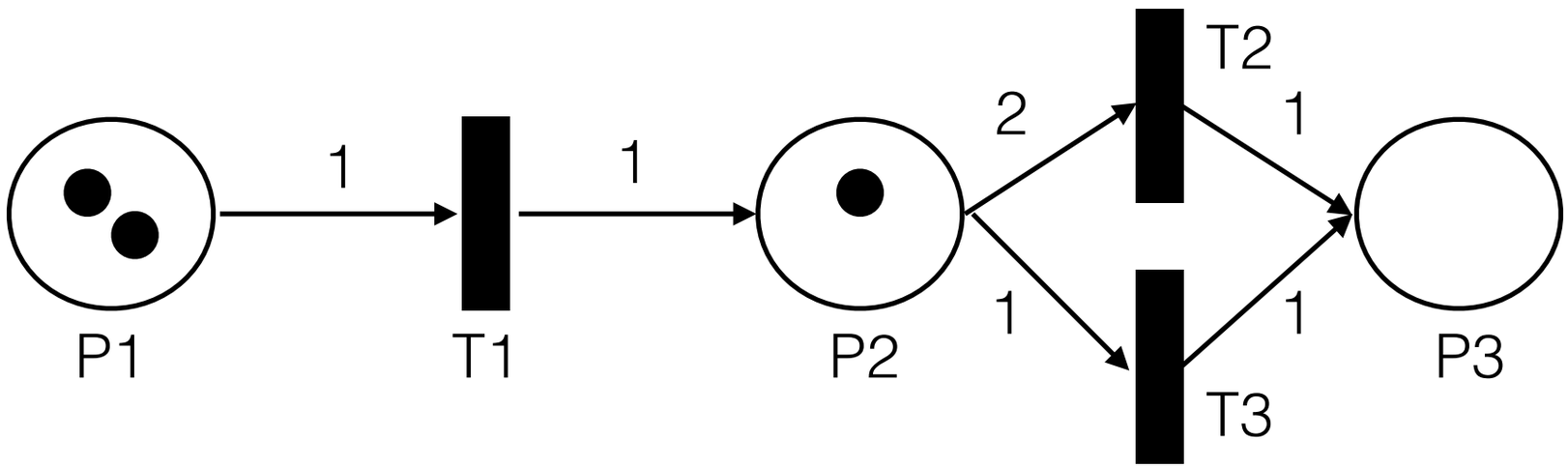
- Model relationships between components using Petri net
- Use type signature of desired method to mark initial and target configurations
- Perform reachability analysis to find valid sequences of method calls

Primer on Petri nets



- Petri net is a generalized graph with two kinds of nodes: places and transitions
- Each place contains zero or more tokens; edges are labeled with a number (of tokens)
- A transition T can fire if, for each edge (p,T) with label n, place p contains at least n tokens
- Firing a transition T consumes (resp. produces) the indicated number of tokens at the source (resp. target) nodes

Reachability problem in Petri nets



- Reachability problem: Given a Petri net with initial marking M and a target marking M′, is it possible to obtain M’ by firing a sequencing of transitions?
- Example: Consider marking M′ : [P1→0,P2→0,P3→1].
- This marking is reachable, and accepting run is T1,T1,T2.

Algorithm overview

APIs

Construct Petri net

Init/target markings

Signature

$\tau_1 \to \tau_2$

Reachability analysis

Candidate Sketch

Sketch Completion

Candidate Program

JUnit

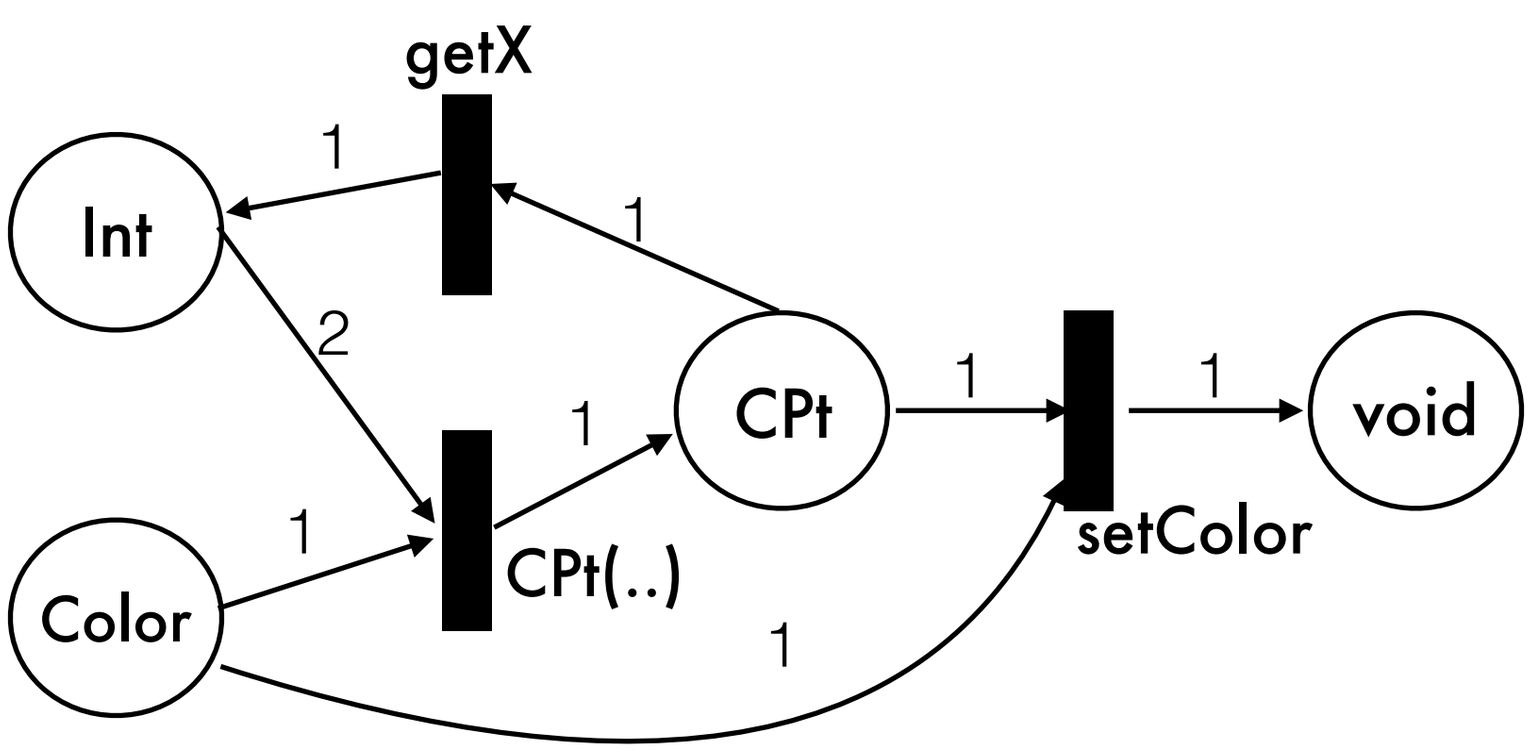
Check Candidate

Java

Backtrack

Petri net construction

```
class CPt {
  CPt(Int x, Int y, Color c);
  Int getX();
  void setColor(Color c);
  ...
}
```



Initial and target markings

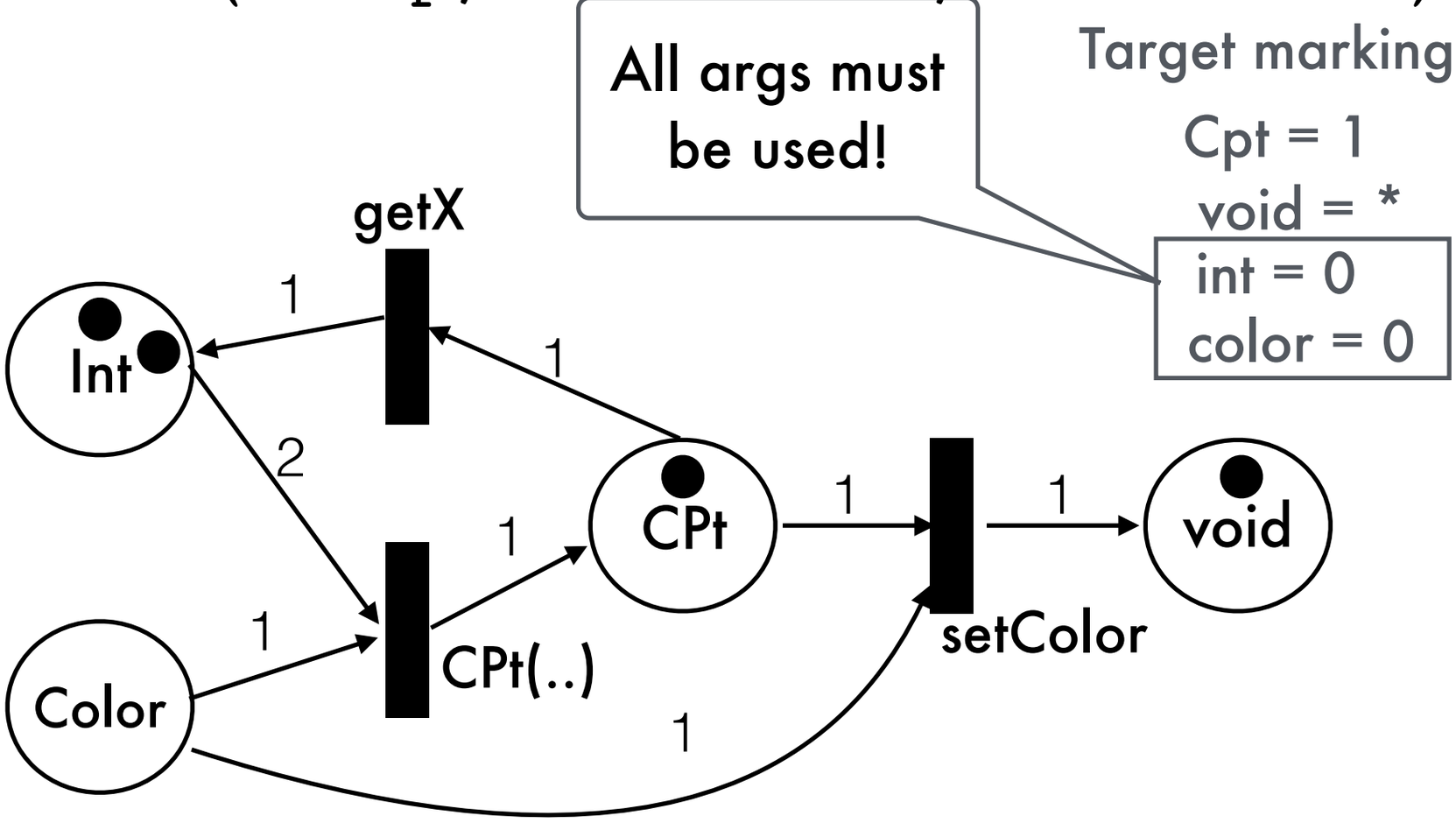
Use signature to determine initial and target markings of Petri net

```
CPt shift (CPt p, Int shiftX, Int shiftY)
```

All args must be used!

Target marking:

Cpt = 1
void = *
int = 0
color = 0



Next step

APIs

Construct Petri net

Candidate Sketch

Candidate Program

JUnit

Reachability analysis

Sketch Completion

Check Candidate

Java

Init/target markings

Backtrack

Signature

$\tau_1 \to \tau_2$

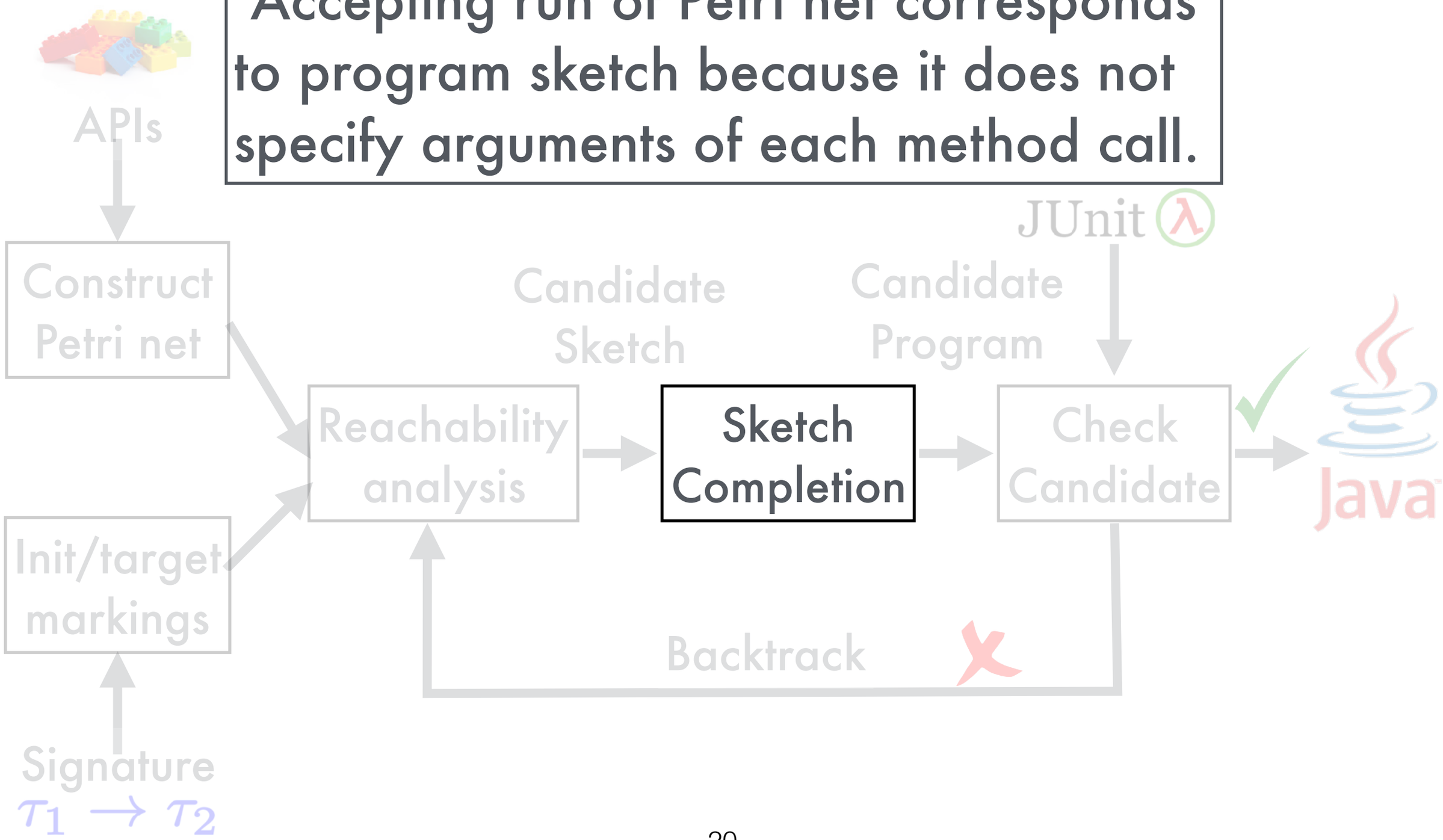
Reachability analysis

All accepting runs of Petri net correspond to method call sequences with desired type signature!

- Need to perform reachability analysis to identify accepting runs of the Petri Net
- Furthermore, need to do this lazily because there may be many accepting runs
- Our solution reduces reachability analysis to integer linear programming (ILP) ⇒ solution corresponds to shortest sequence of method calls

Accepting run as program sketch

Accepting run of Petri net corresponds to program sketch because it does not specify arguments of each method call.



Sketch completion

```
x = #1.getX(); y = #2.getY();
#3.setToRotation(#4, #5, #6);
a = #7.createTransformedArea(#8);
return #9;
```

- Given a program sketch with holes, need to instantiate each hole with a program variable such that program type checks
- Encode this as a boolean satisfiability problem:

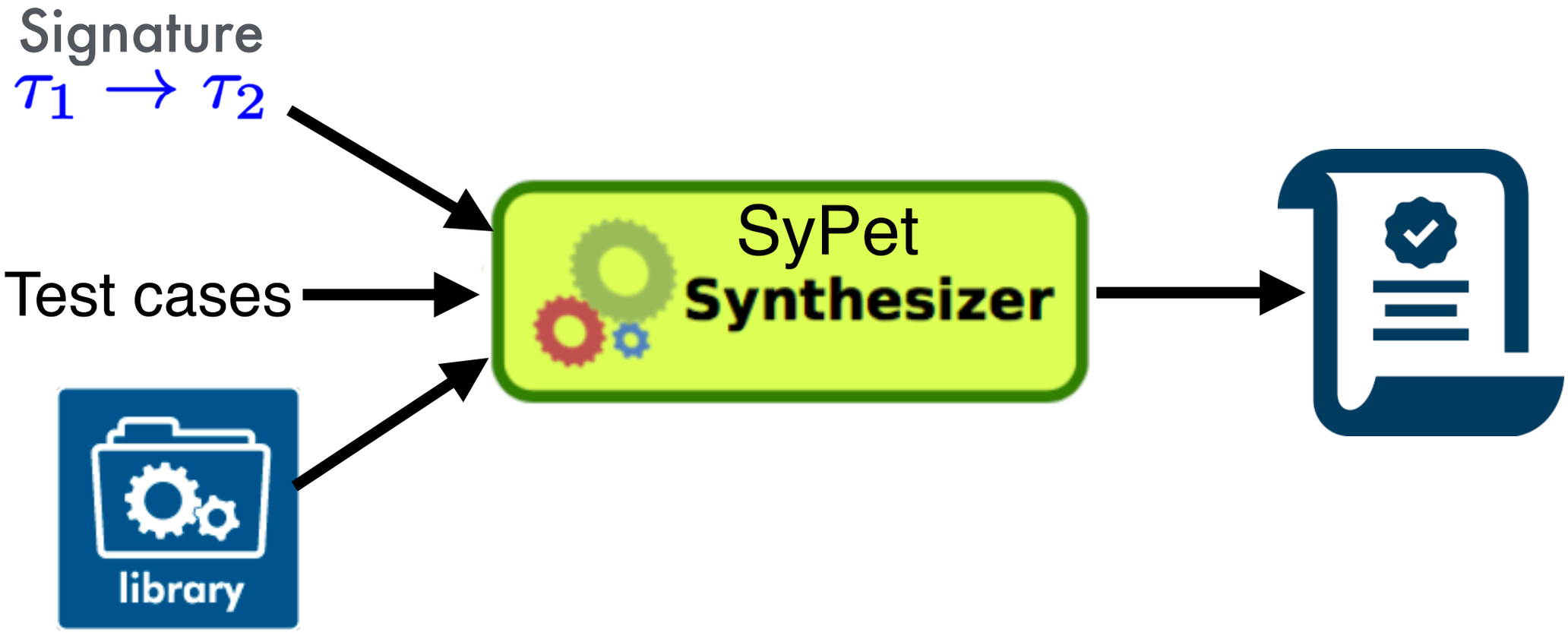
$$\forall_{\#i \in H} . \forall_{v \in getV(V,\#i)} . \; \Sigma h_v^{\#i} = 1$$

each hole filled with one variable

$$\forall_{v \in V} . \forall_{\#i \in getH(H,v)} . \; \Sigma h_v^{\#i} \geq 1$$

each variable used at least once

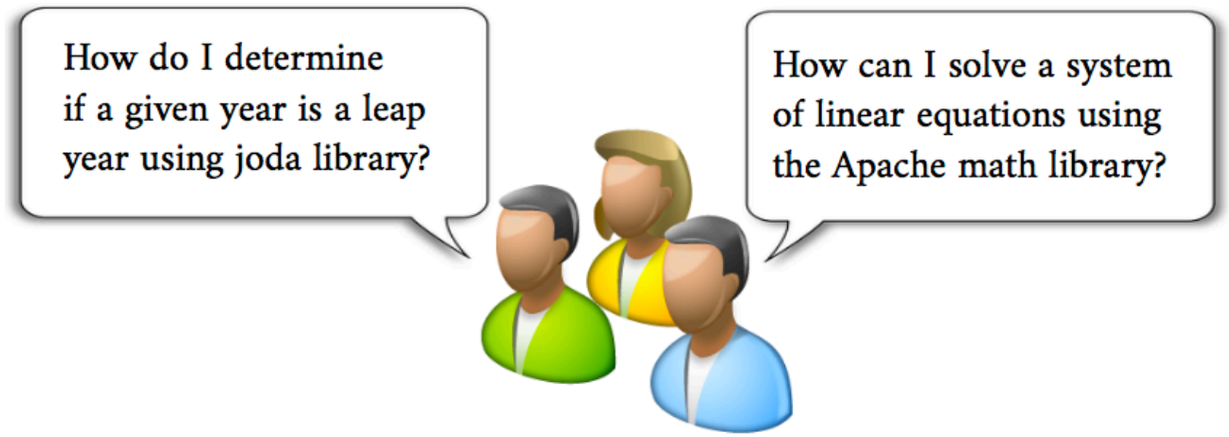
Implementation



https://github.com/utopia-group/sypet

Evaluation

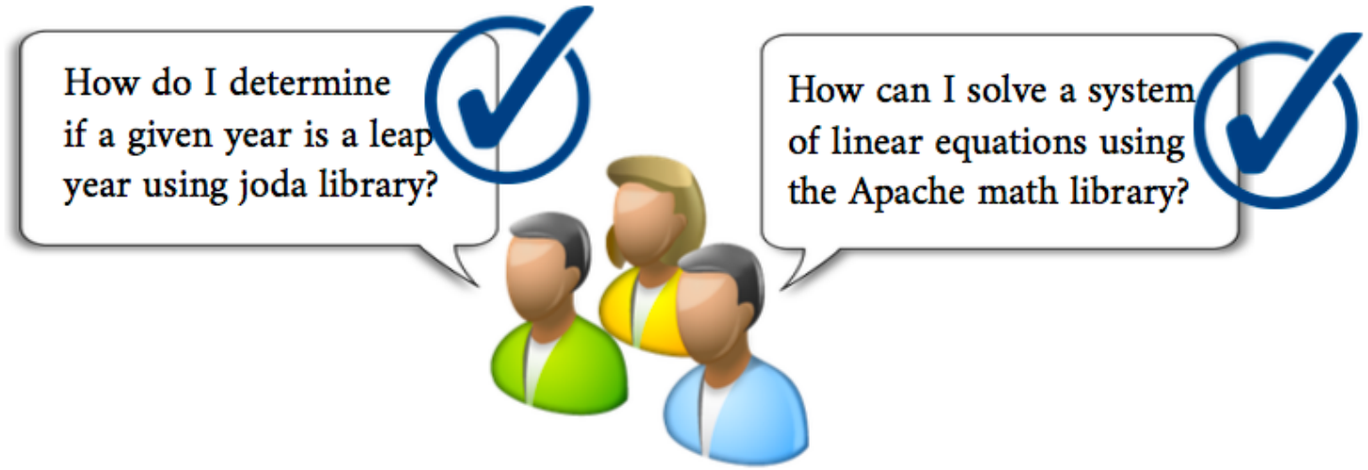
- Collected 30 API-usage questions from Stackoverflow involving six different libraries with 751-9578 methods:



- Extracted signature and test case from post if available, otherwise wrote it ourselves

- Used SyPet to automatically synthesize the implementation

Results



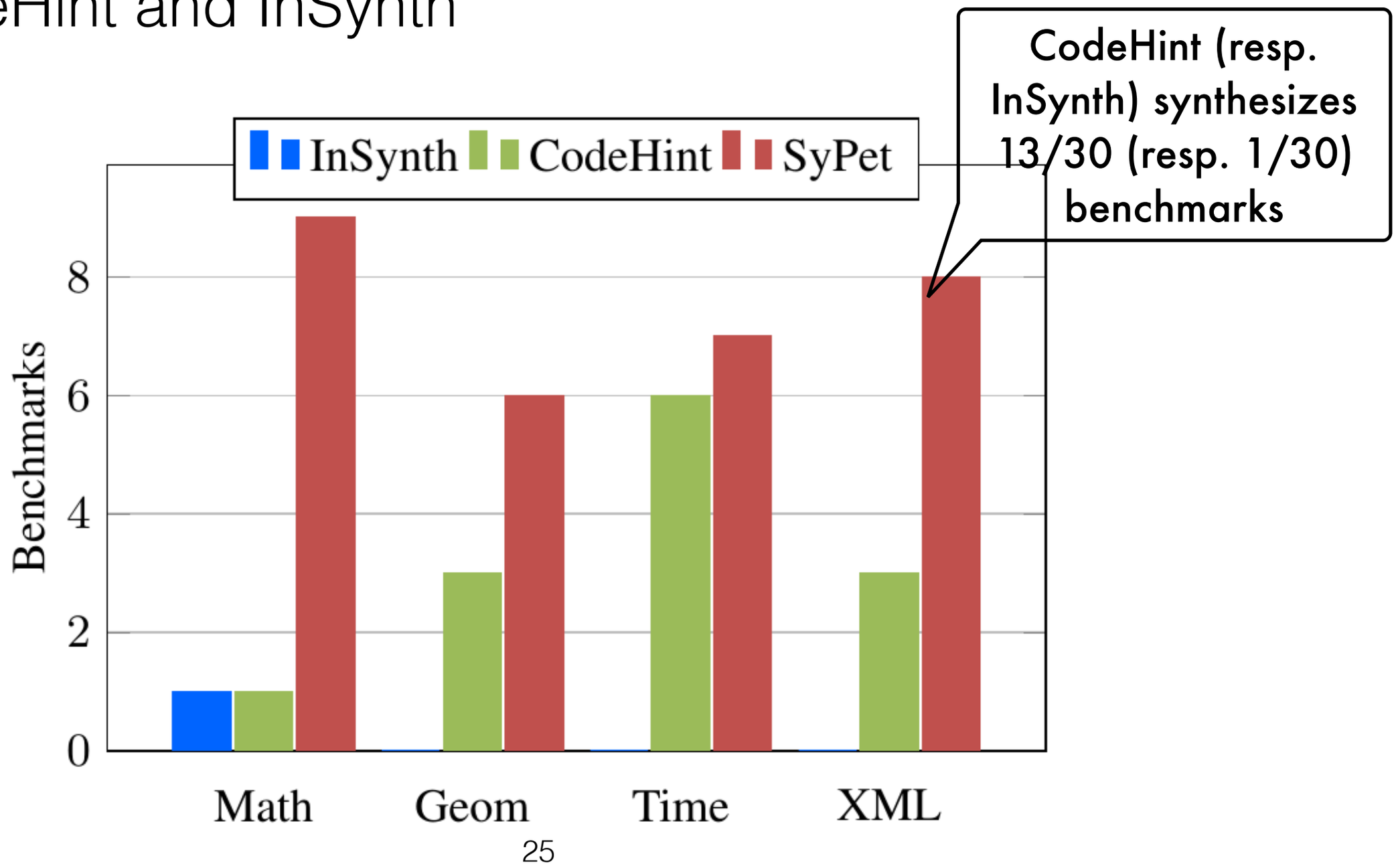
- Our technique was able to successfully synthesize the correct implementation of all 30 benchmarks

- Median synthesis time 1.57 seconds

Our synthesis technique is useful to programmers

Comparison with other tools

- Also compared SyPet with two other synthesis tools, CodeHint and InSynth



When type is not helpful anymore

String manipulations
Numerical transformation
Table transformations
...

Most of methods have the same input&output types

TODOs by next lecture

- Start to work on HW2
- Start to work on the proposal for your final project