

TYPE-DIRECTED COMPONENT-BASED SYNTHESIS

{ YU FENG, YUEPENG WANG, RUBEN MARTINS, ARATI KAUSHIK, ISIL DILLIG }@UT AUSTIN



OBJECTIVES

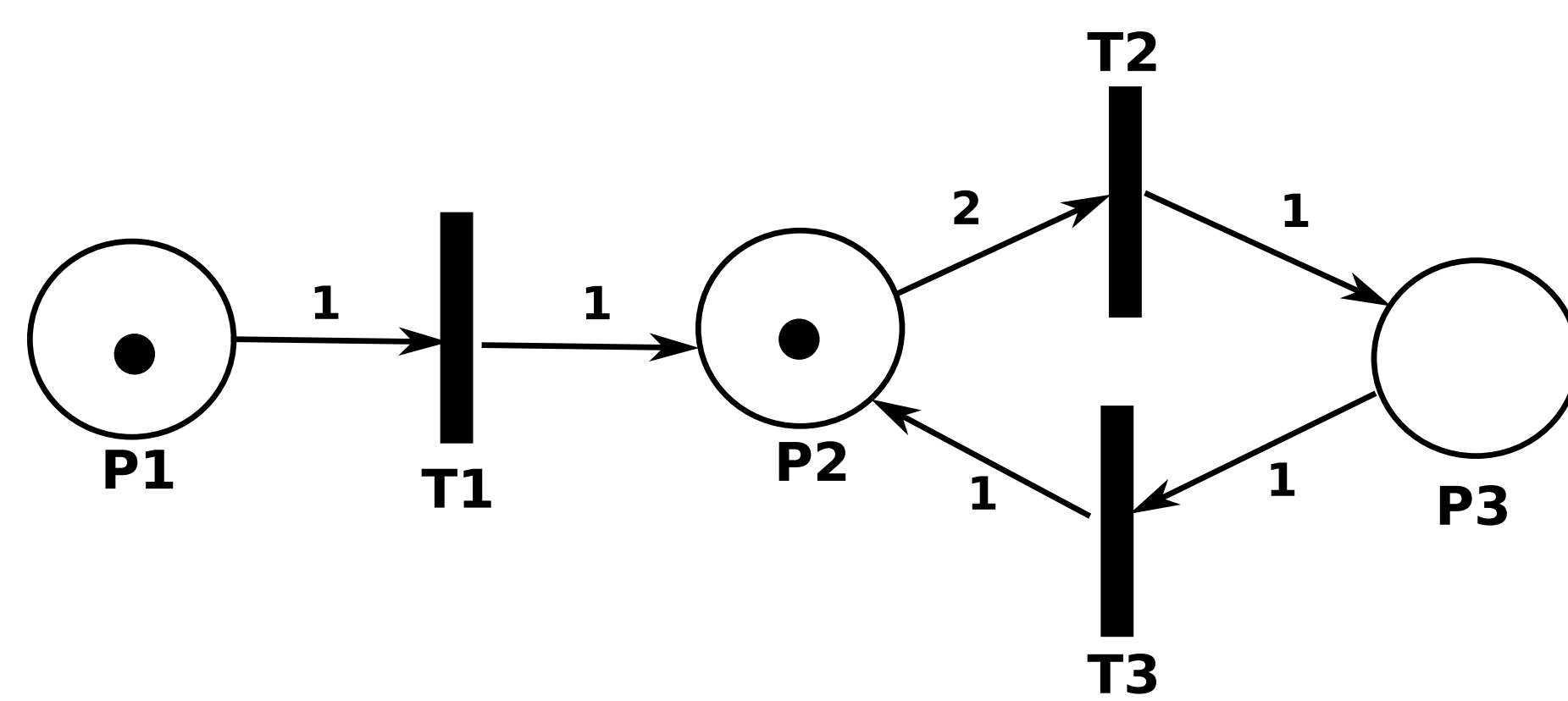


SYPET is a component-based synthesizer for large libraries that automatically synthesizes executable programs by composing API calls.

Key components:

1. **Synthesis of program sketches:** SYPET uses Petri nets to generate programs sketches from signatures of the desired method and underlying library components.
2. **Completion of program sketches:** SYPET generates constraints on the synthesized program with holes and uses a SAT solver to find a candidate method.

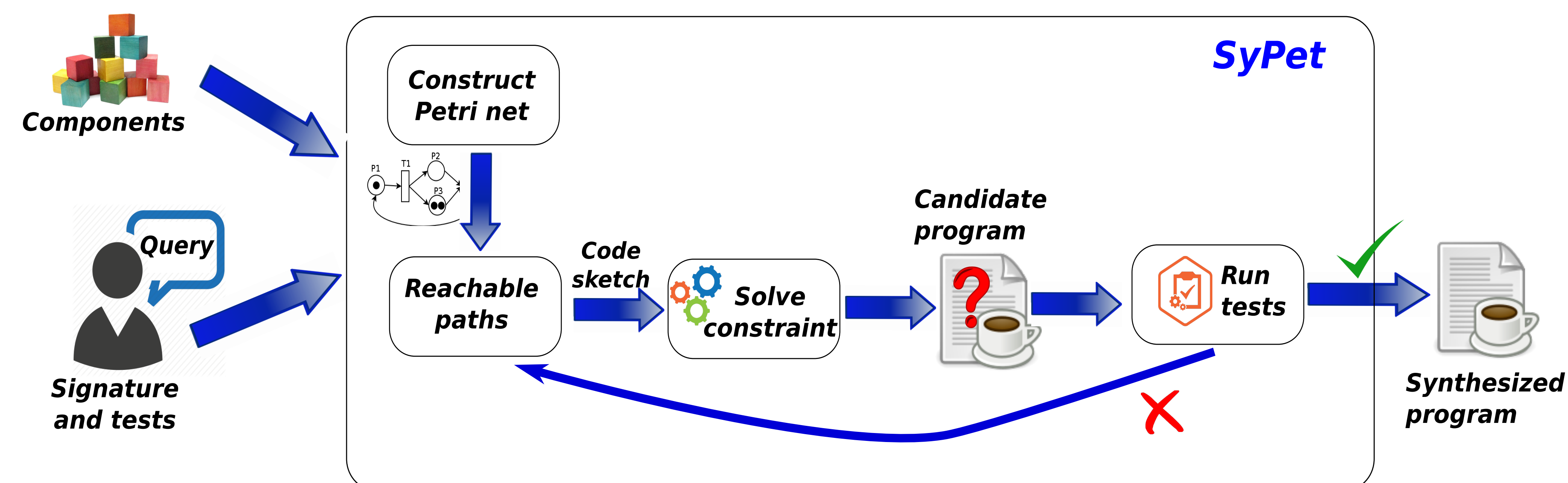
PETRI NET IN A NUTSHELL



A Petri net is a bipartite graph with two types of nodes:

- **Place (in circle):** it can contain a number of *tokens*, which are drawn as dots and typically represent resources.
- **Marking:** a mapping from each place p to the number of tokens at p .
- **Transition (in solid bar):** corresponds to events that change the marking.

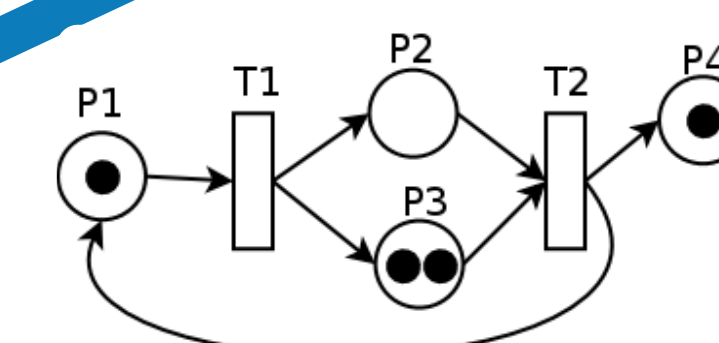
OVERVIEW



OUR APPROACH BY EXAMPLE

Example: Determine the roots of a polynomial equation

```
Complex[] findRoots(PolynomialFunction arg0,
                    double arg1) {
}
```



Synthesizing sketches

```
Complex[] findRoots(PolynomialFunction arg0,
                    double arg1) {
    LaguerreSolver v1 = new LaguerreSolver();
    double[] v2 = ?.getCoefficients();
    Complex[] v3 = ?.solveAllComplex(?, ?);
    return ?;
}
```



Filling sketches

```
Complex[] findRoots(PolynomialFunction arg0,
                    double arg1) {
    LaguerreSolver v1 = new LaguerreSolver();
    double[] v2 = arg0.getCoefficients();
    Complex[] v3 = v1.solveAllComplex(v2, arg1);
    return v3;
}
```

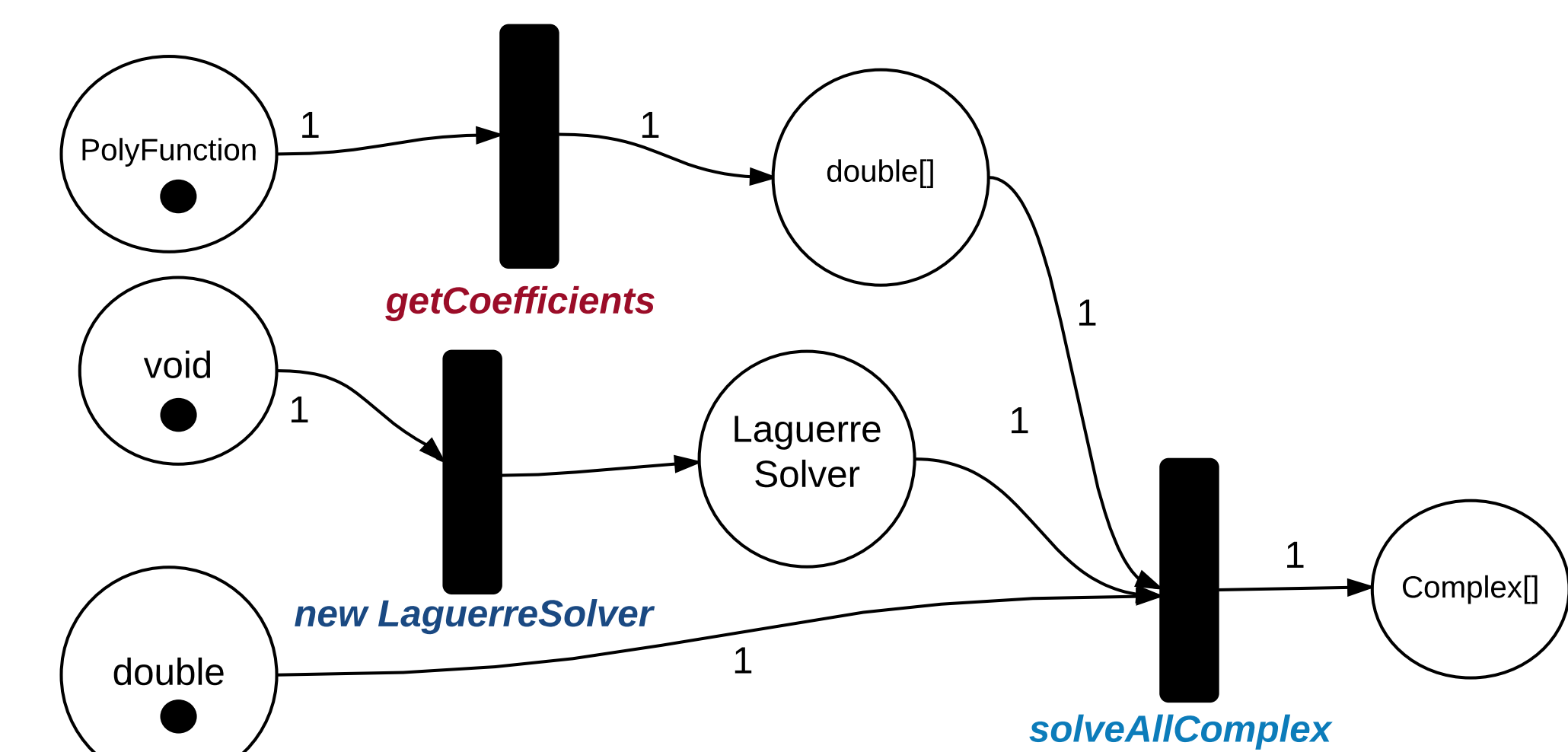
SYNTHESIS VIA PETRI NET

Places:

- Types

Transitions:

- API methods



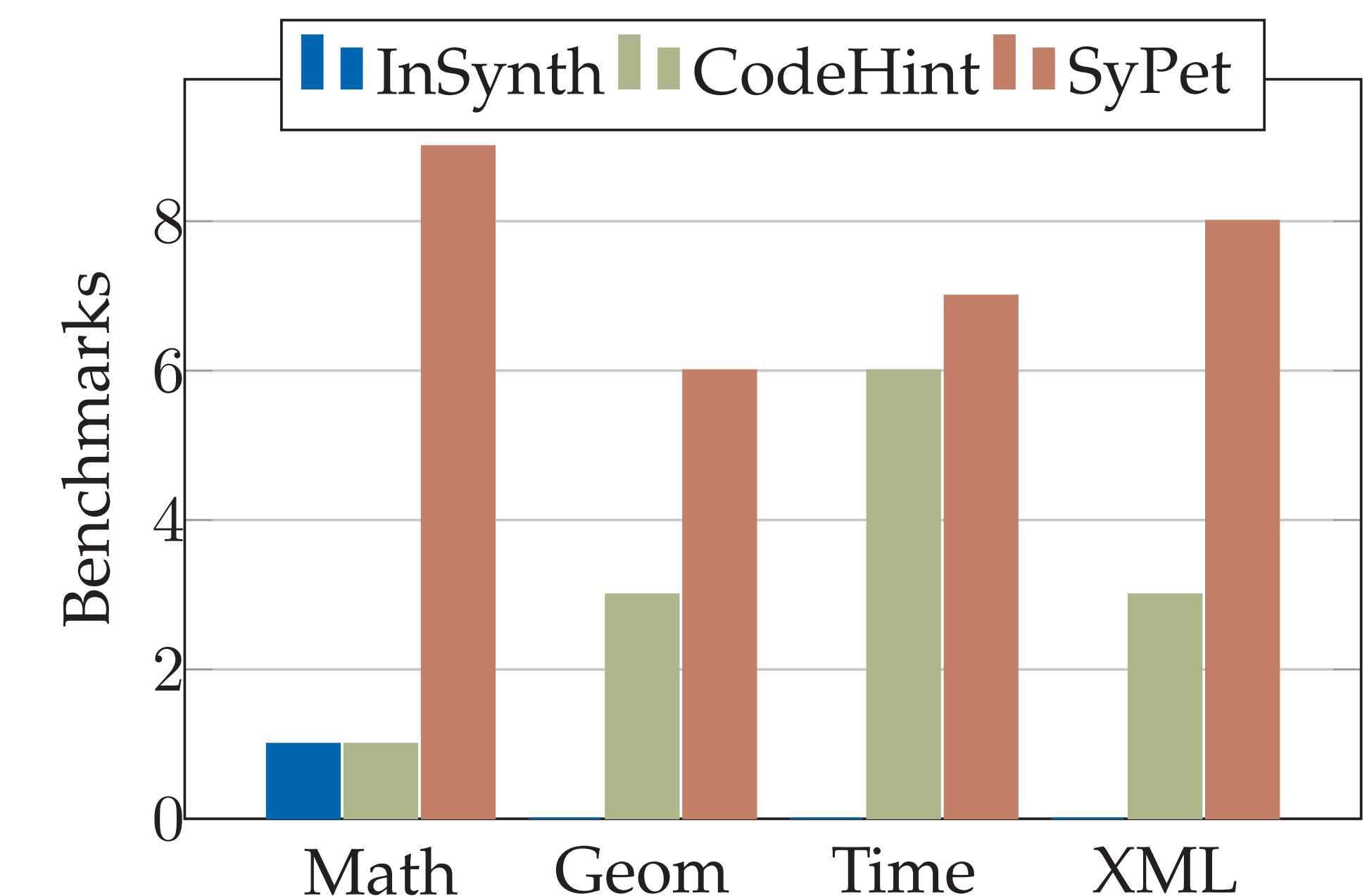
A reachable path in the Petri net corresponds to a program sketch. For example:

- new LaguerreSolver
- getCoefficients
- solveAllComplex

RESULTS

Java libraries:

- Math: apache.commons.math
- Geometry: awt.geom
- Time: joda
- XML: jsoup, wc3.dom, javax.xml



SyPet can handle *large* libraries:

- 49-246 classes (218 average)
- 725-4105 methods (2080 average)

ACKNOWLEDGMENTS

This material is based on research sponsored by the Air Force Research Laboratory, under agreement number FA8750-14-2-0270.