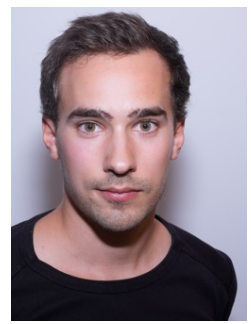
Part 3: Cooperative Verification by Reducer-Based Construction of Conditional Verifiers

Dirk Beyer

Joint work with Marie-Christine Jakobs, Thomas Lemberger, and Heike Wehrheim

LMU Munich, Germany





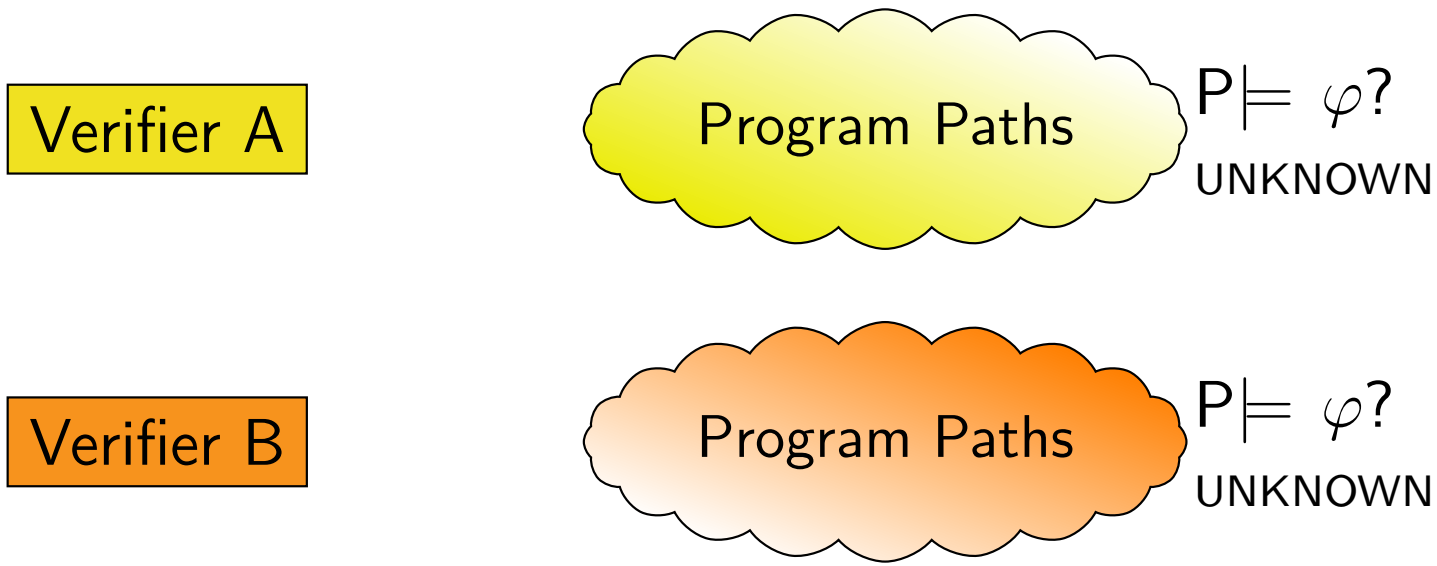

Vision

I have a dream ...

- ... that one day, all tools for formal methods work together to solve hard verification problems and make our world safer and more secure.
- ... that one day, model checkers and theorem provers can be integrated into the software-development process as seamless as unit testing today.
- ... that one day, model checkers, theorem provers, SMT solvers, and testers use common interfaces for interaction and composition.

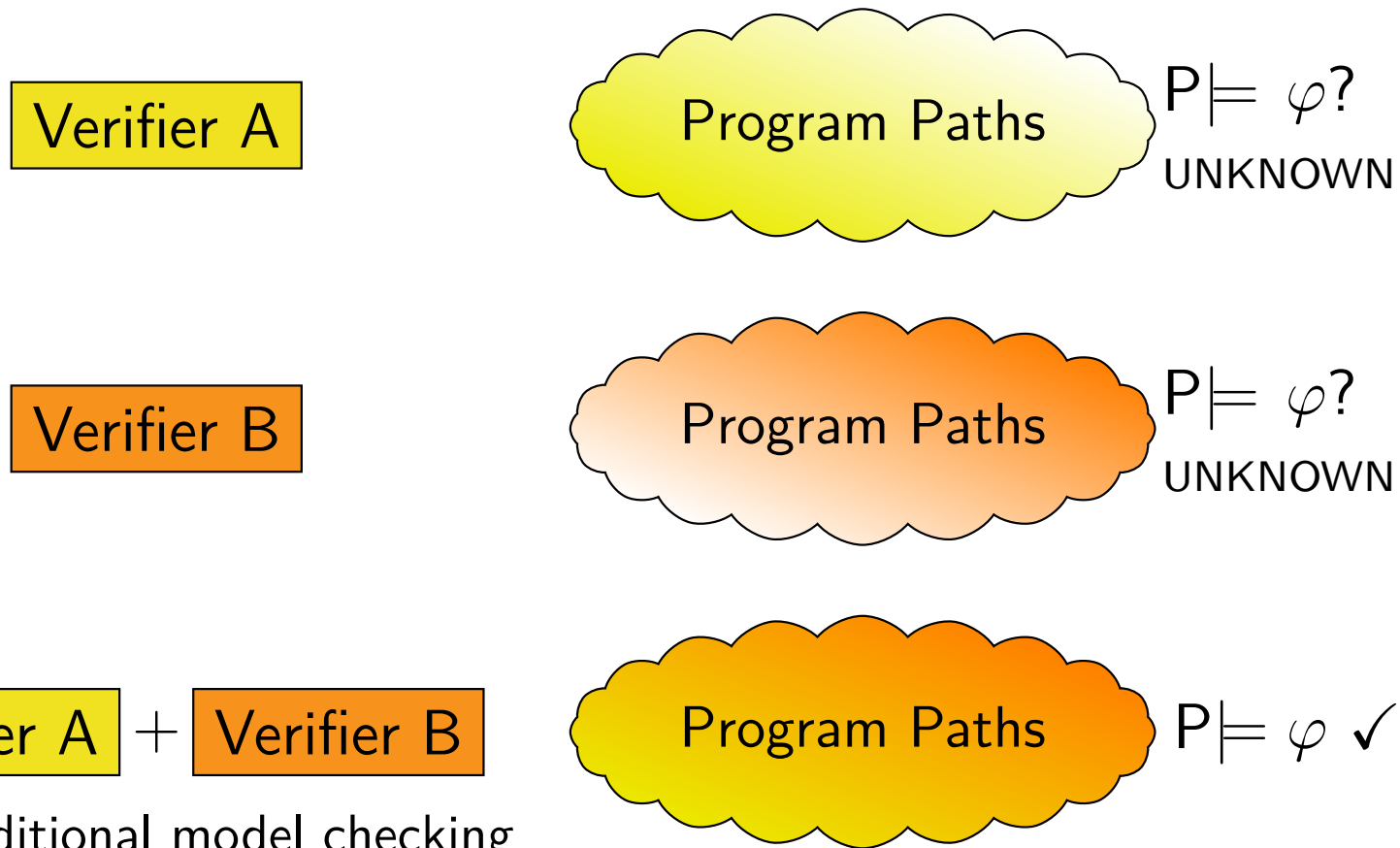
Facing Hard Verification Tasks

Given: Program P$\models \varphi$?



Facing Hard Verification Tasks

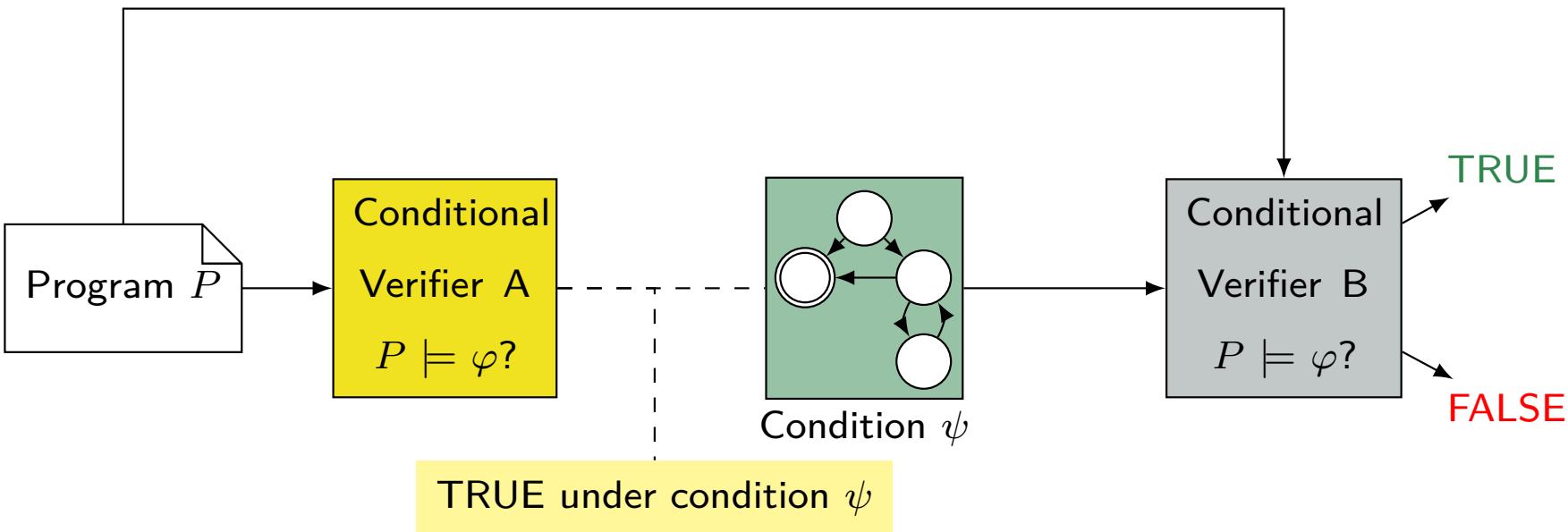
Given: Program P$\models \varphi$?



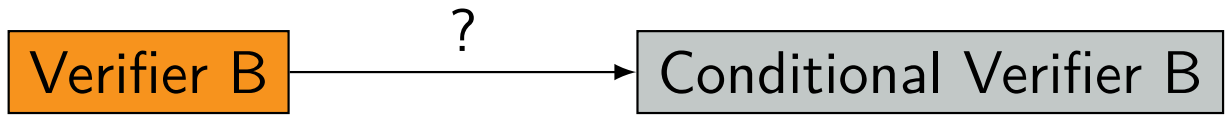
e.g., conditional model checking

Conditional Model Checking

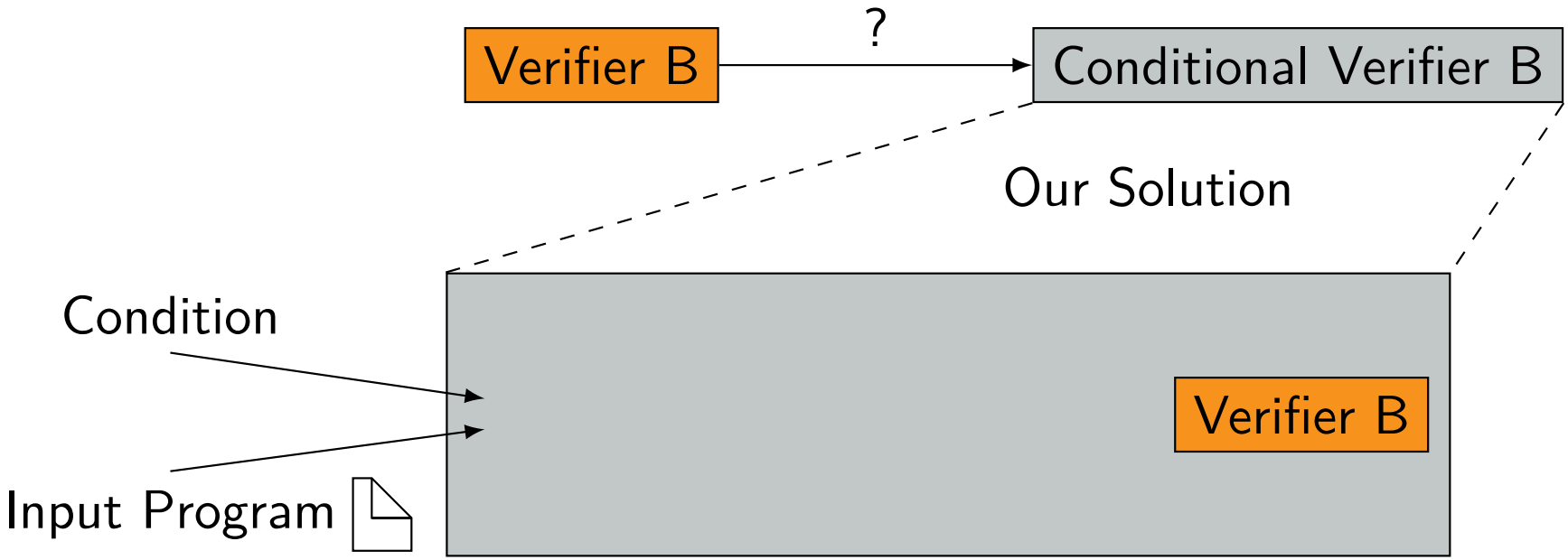
[Beyer/Henzinger/Keremoglu/Wendler FSE'12]



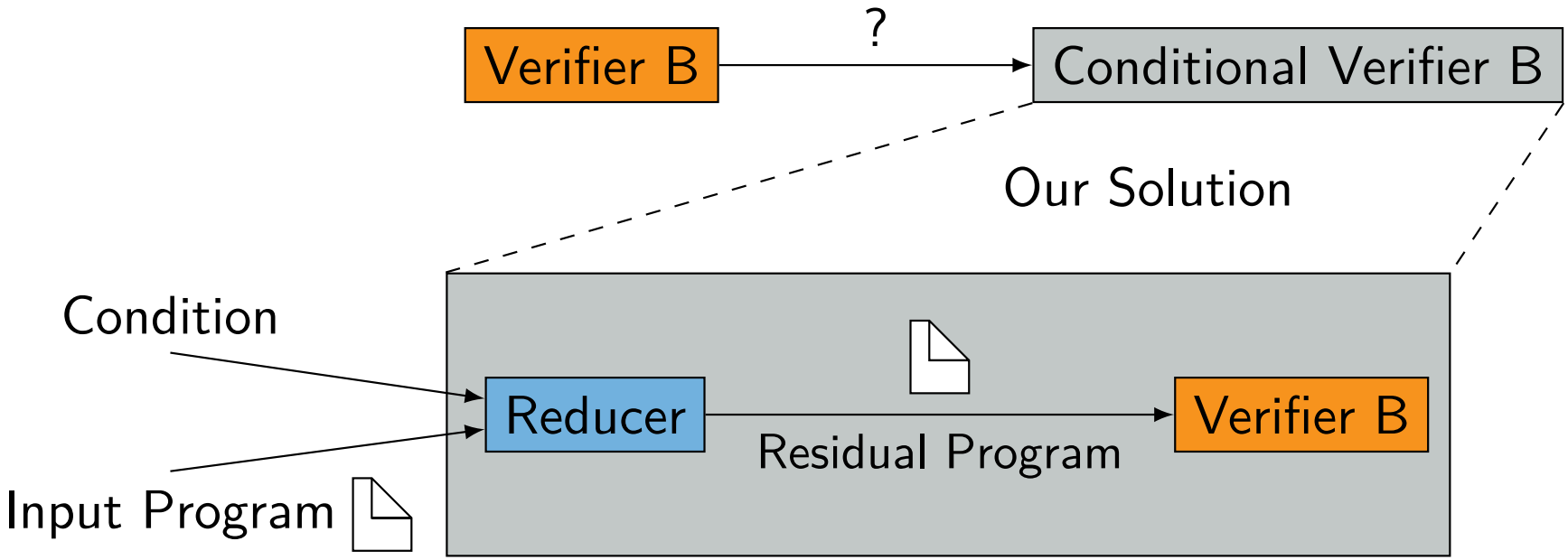
Reducer-Based Conditional Verifier Construction



Reducer-Based Conditional Verifier Construction



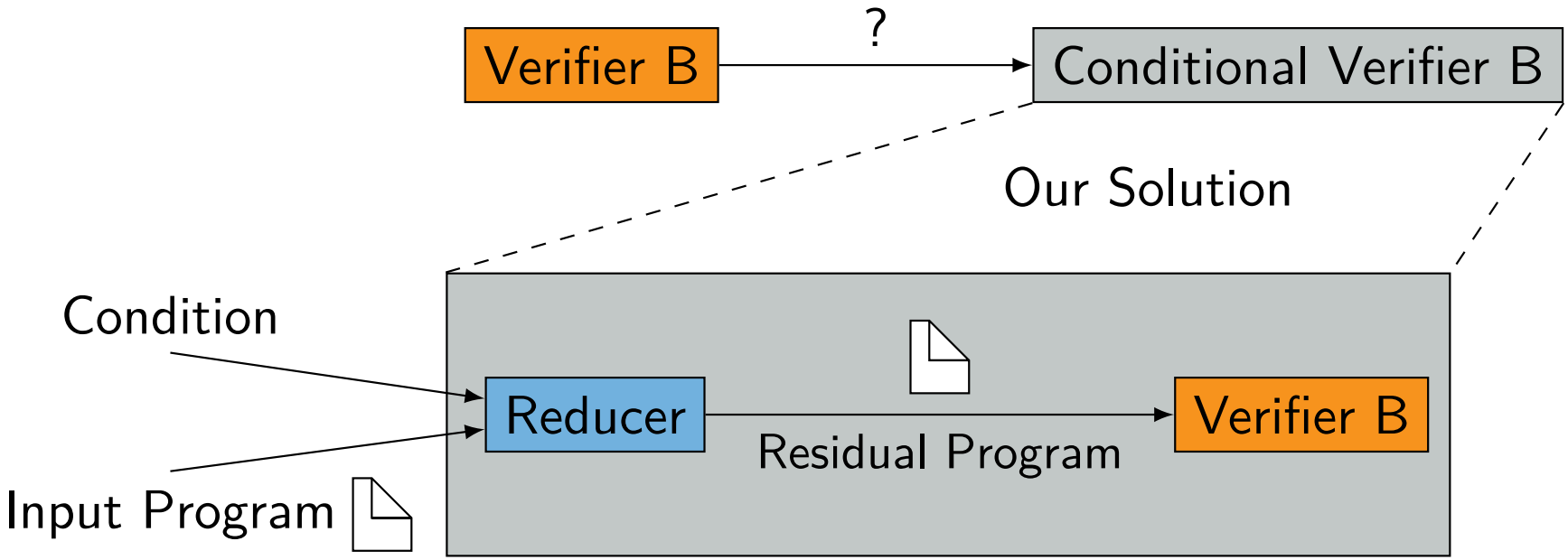
Reducer-Based Conditional Verifier Construction



Reducer (preprocessor)

- Builds standard input (C program)
- Representing a subset of paths
- Contains at least all non-verified paths

Reducer-Based Conditional Verifier Construction



Reducer (preprocessor)

- Builds standard input (C program)
- Representing a subset of paths
- Contains at least all non-verified paths

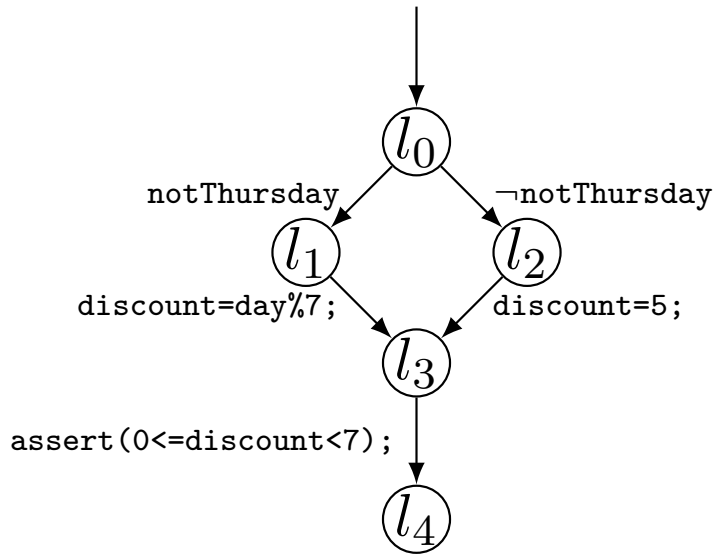
\+ Verifier-unspecific approach

\+ Many conditional verifiers possible

Example Program and Condition

Program

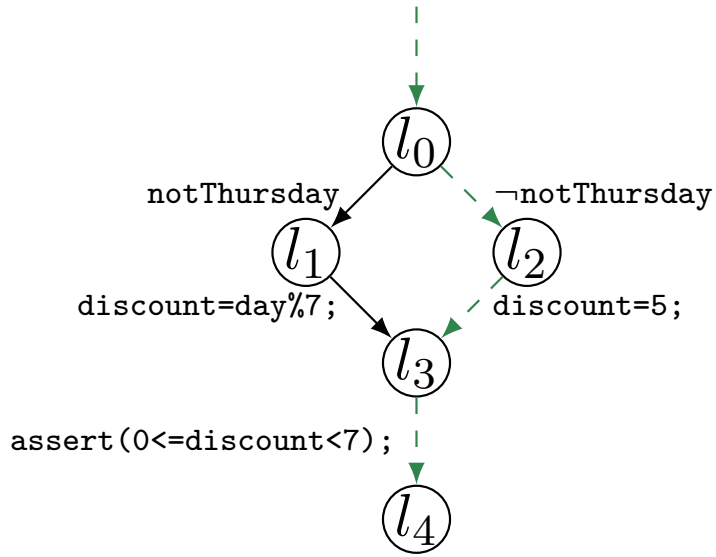
```
0: if(notThursday)
1:  discount=day%7;
    else
2:  discount=5;
3: assert(0<=discount<7);
4:
```



Example Program and Condition

Program

```
0: if(notThursday)
1:  discount=day%7;
    else
2:  discount=5;
3: assert(0<=discount<7);
4:
```

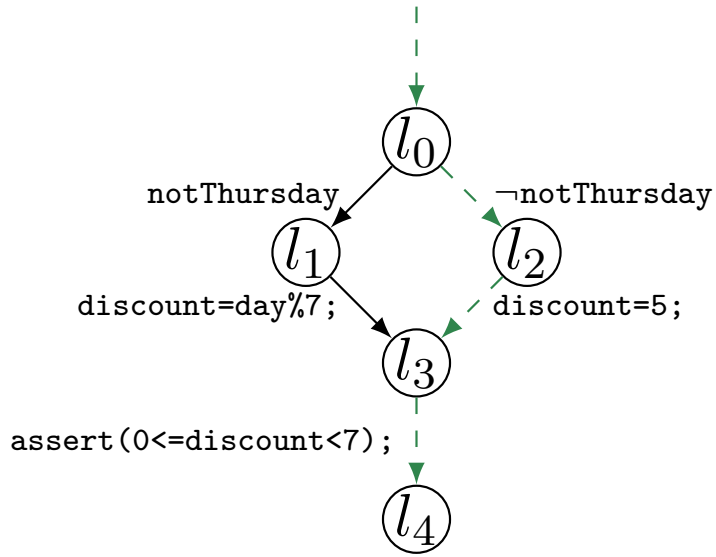


Verifier A only proofs else branch

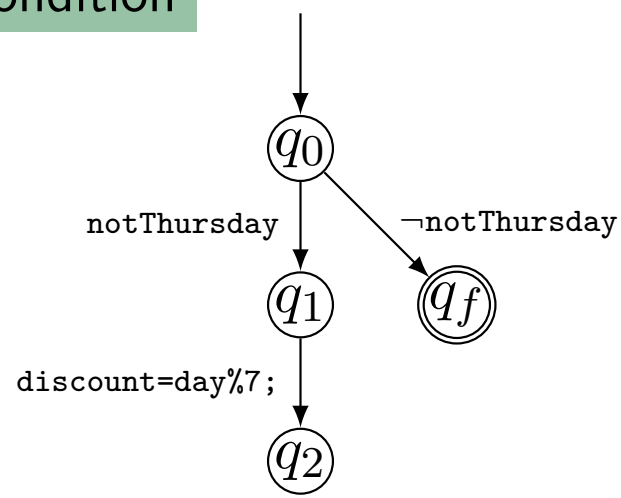
Example Program and Condition

Program

```
0: if(notThursday)
1:  discount=day%7;
    else
2:  discount=5;
3: assert(0<=discount<7);
4:
```

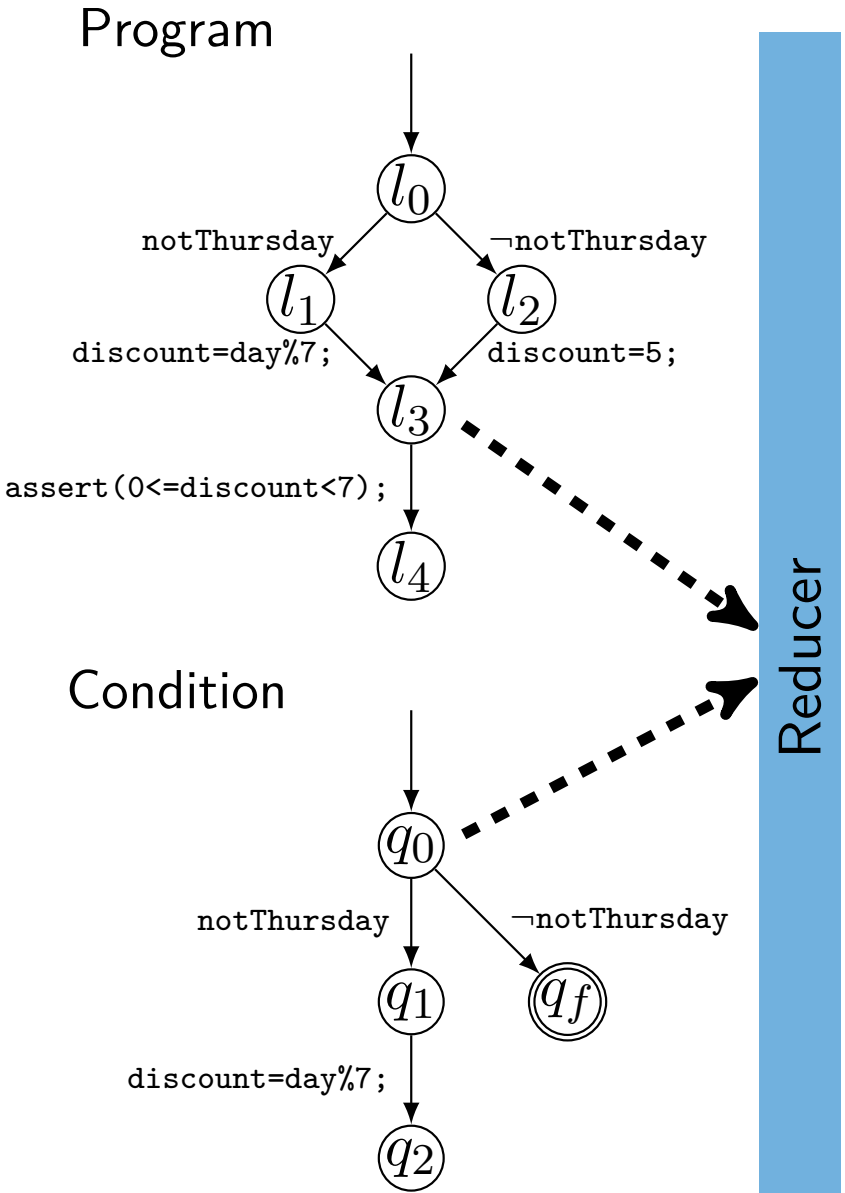


Condition



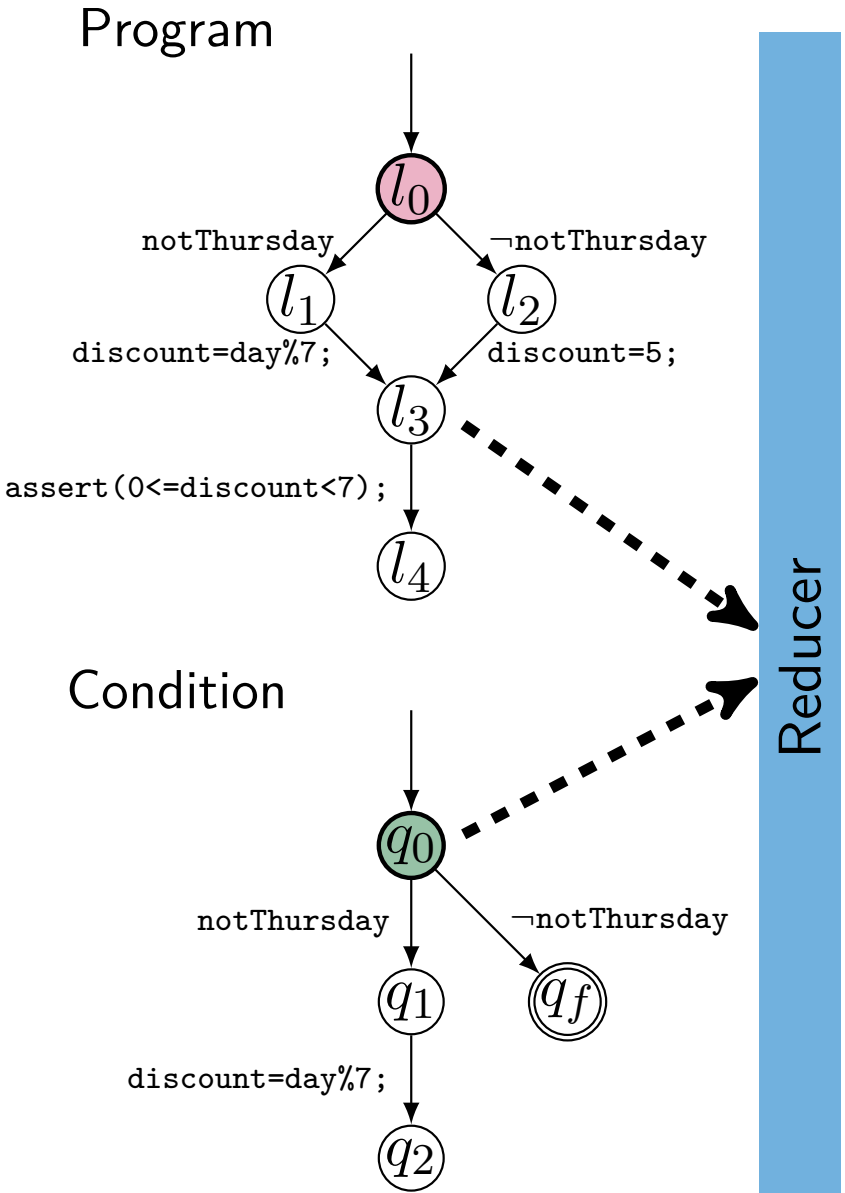
Verifier A only proofs else branch

Reducer: Residual Program Construction



Residual Program

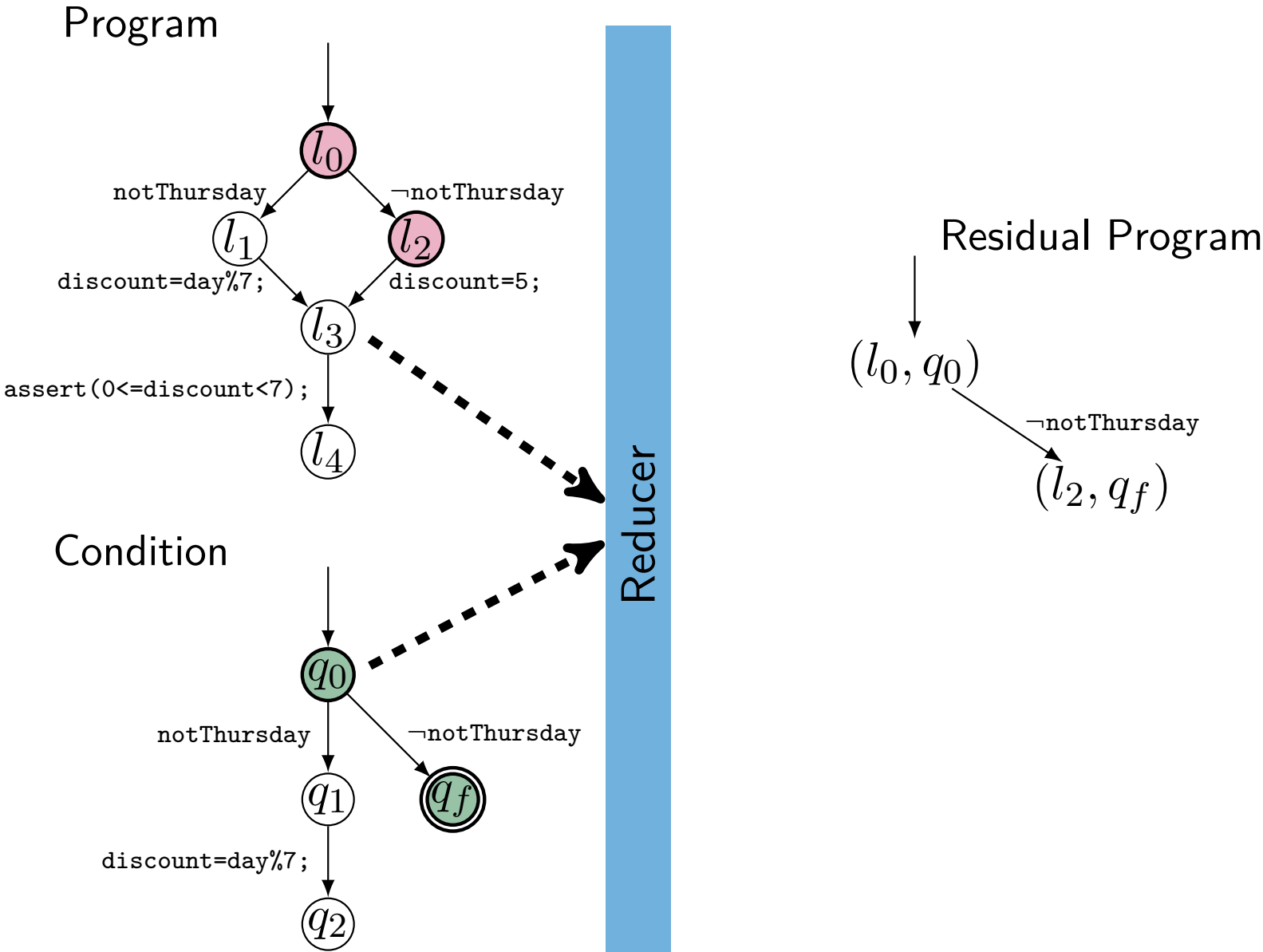
Reducer: Residual Program Construction



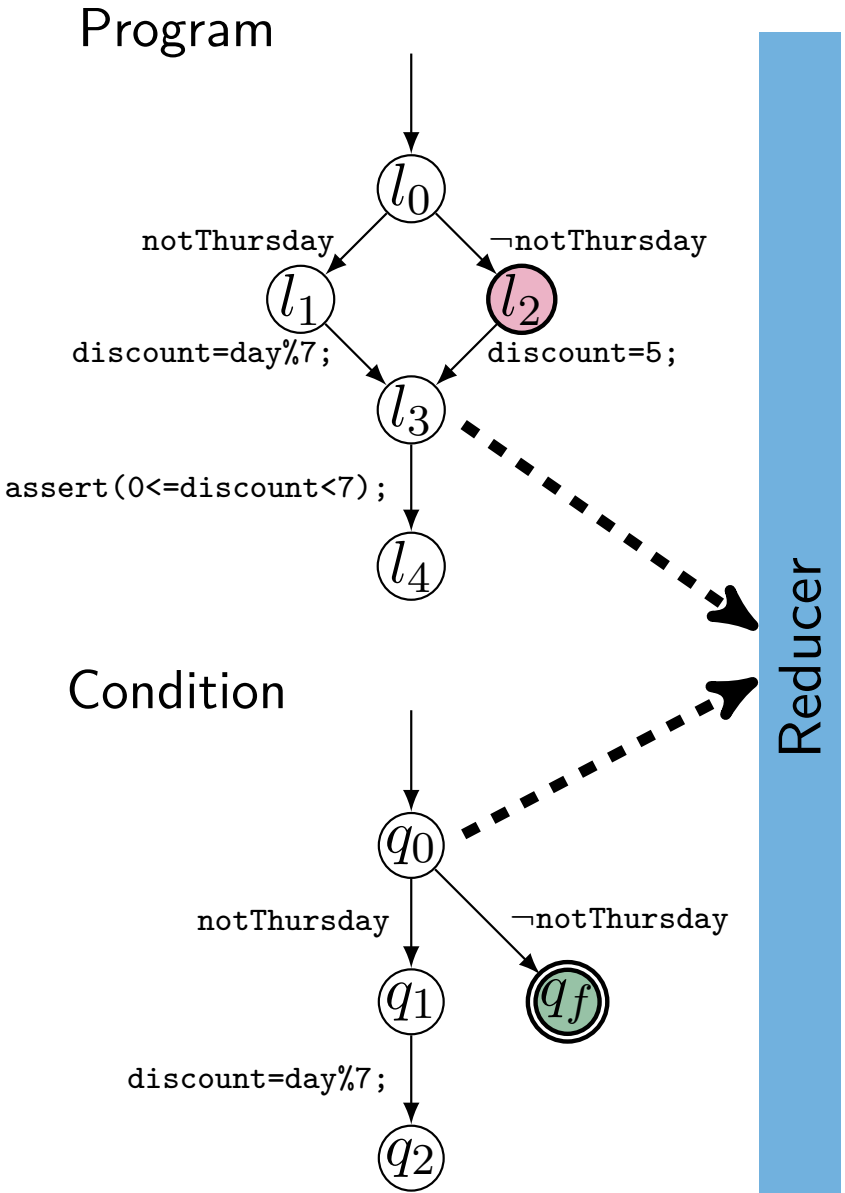
Residual Program

(l_0, q_0)

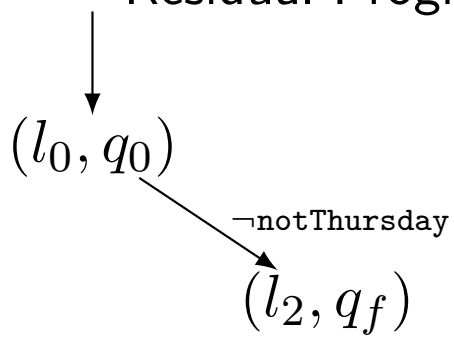
Reducer: Residual Program Construction



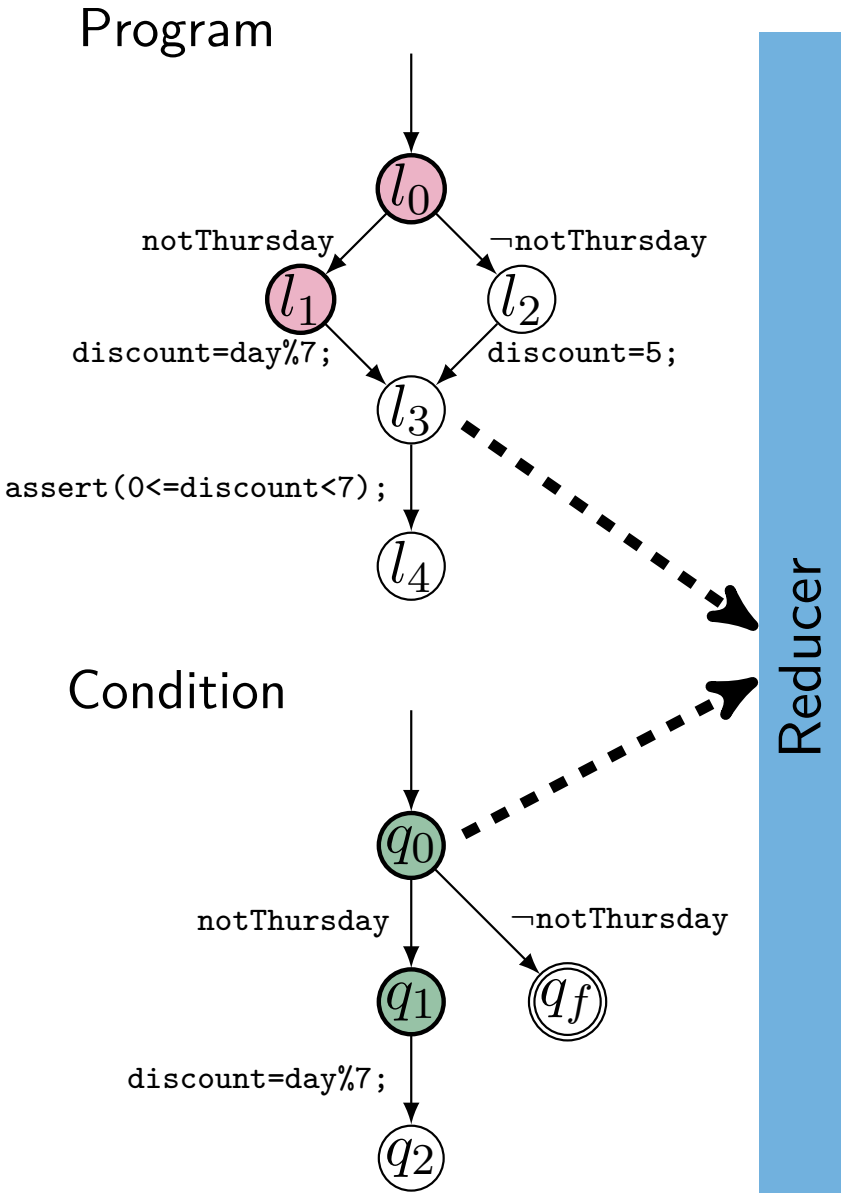
Reducer: Residual Program Construction



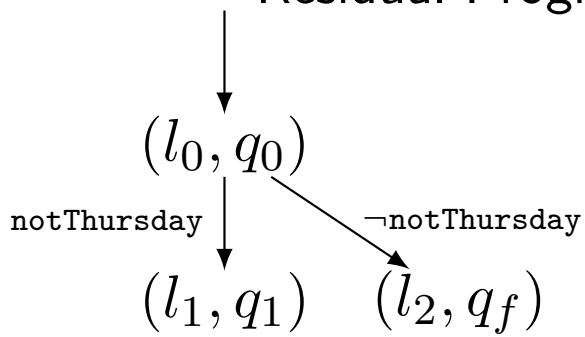
Residual Program



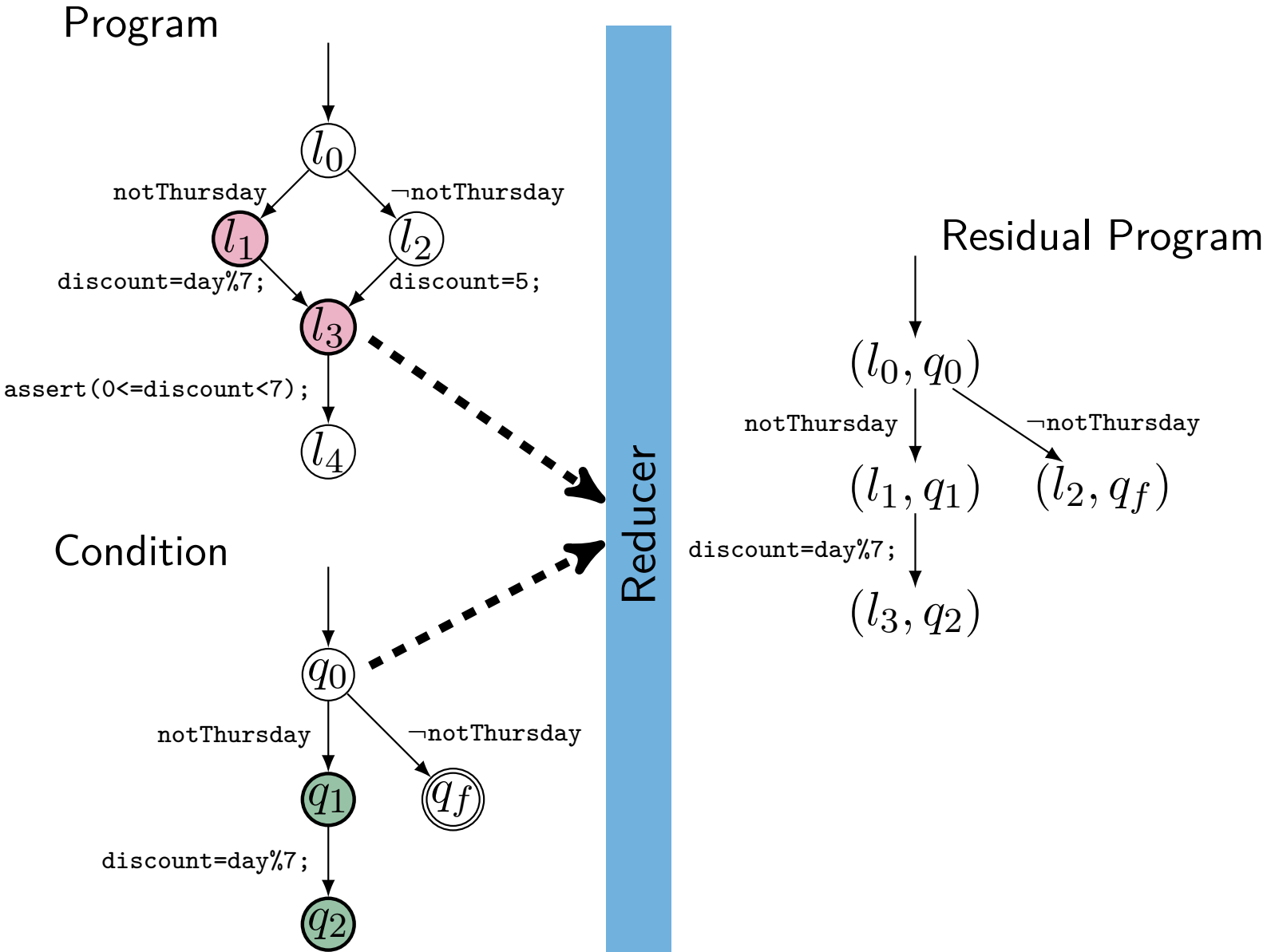
Reducer: Residual Program Construction



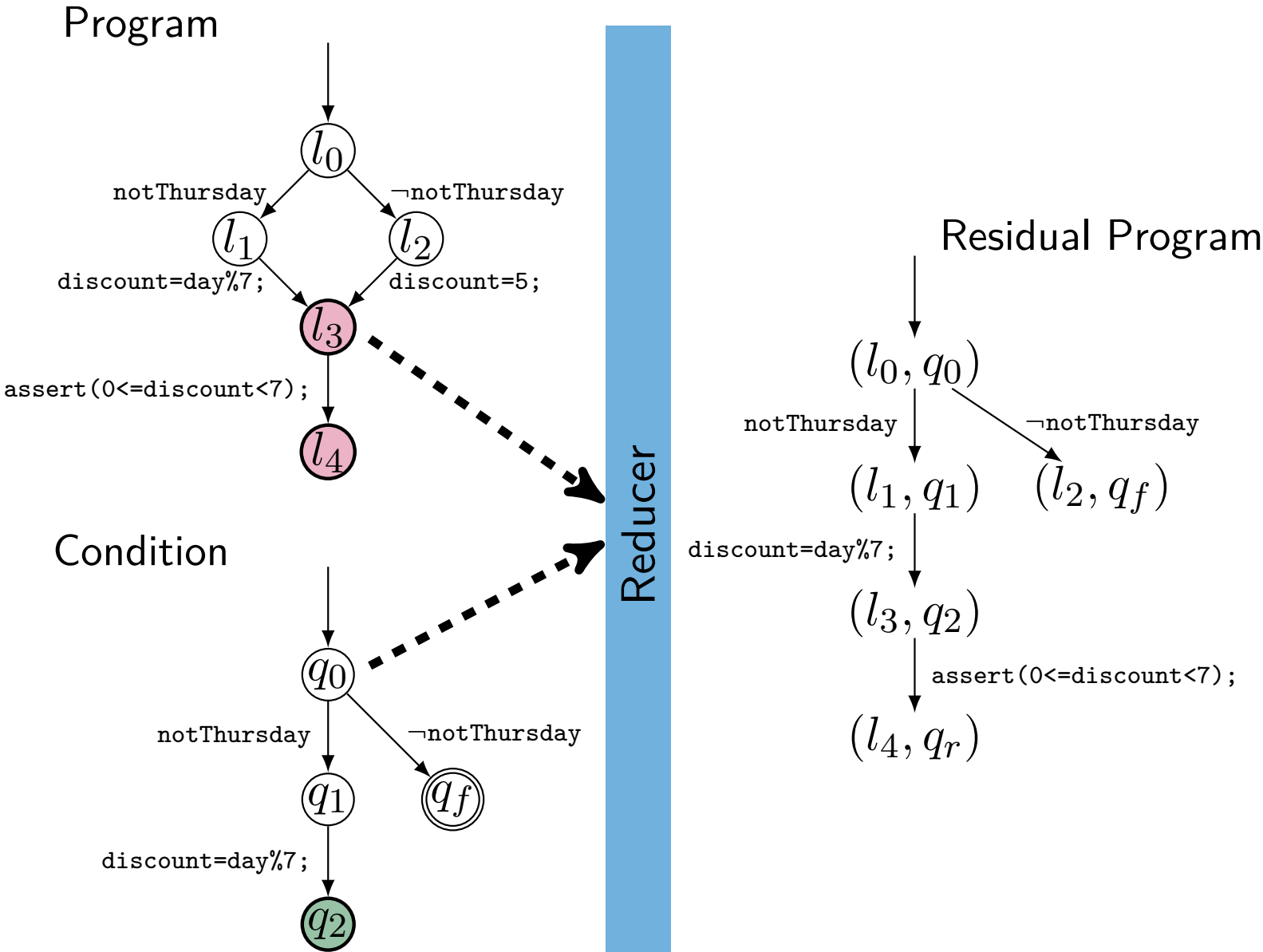
Residual Program



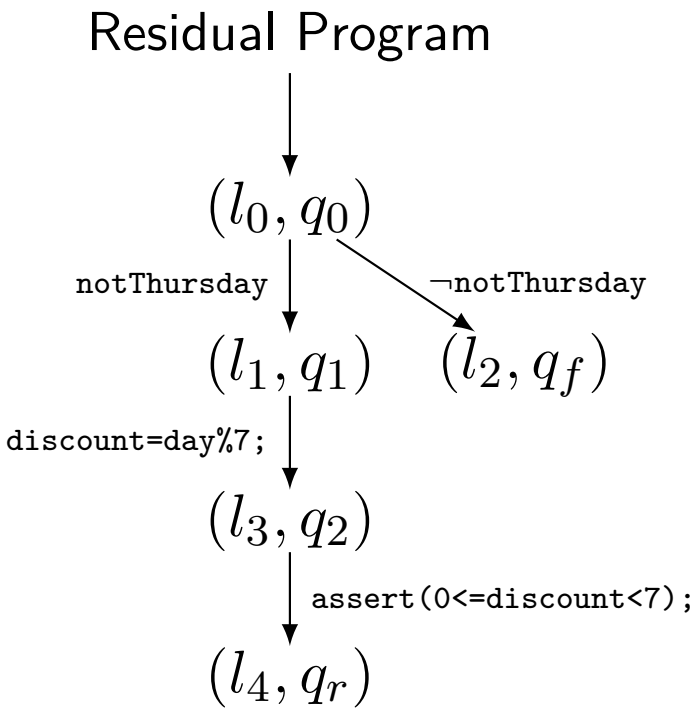
Reducer: Residual Program Construction



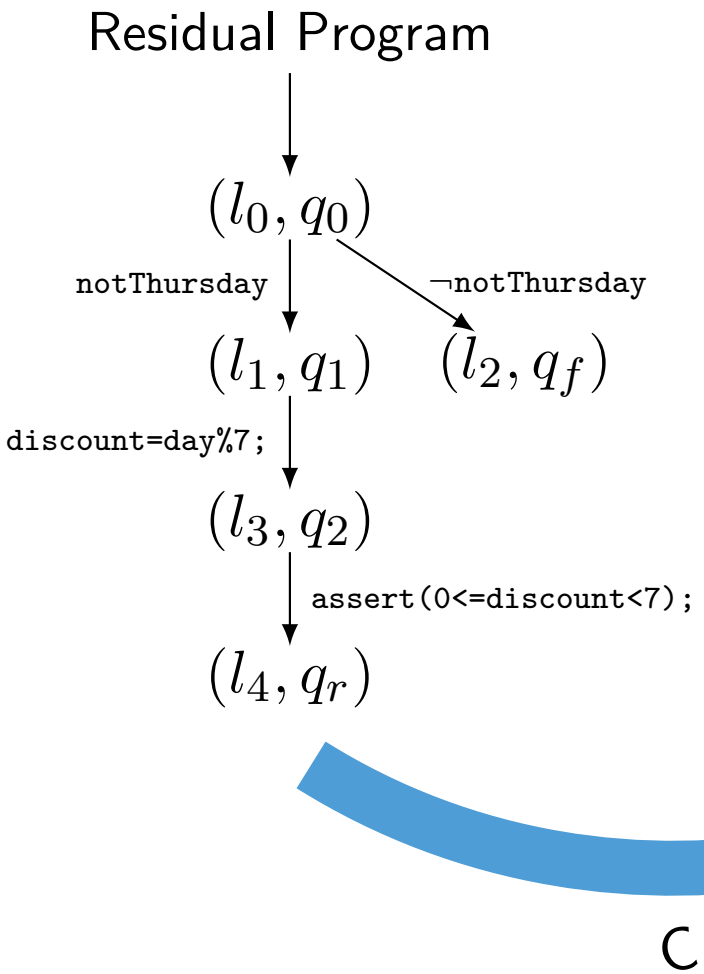
Reducer: Residual Program Construction



Reducer: C Transformation



Reducer: C Transformation

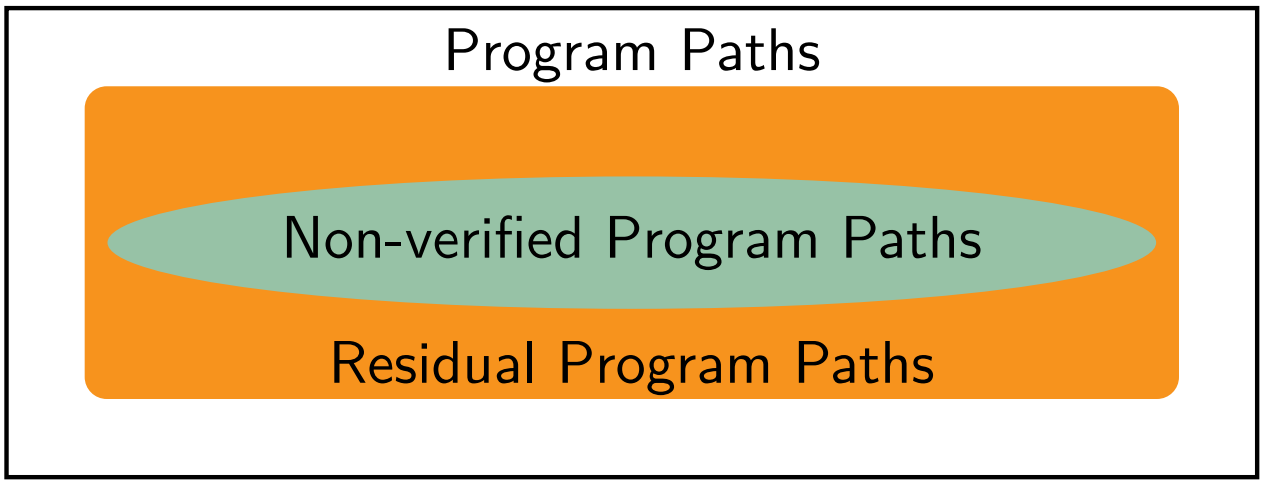



```
if(notThursday)
{
  discount=day%7;
  assert(0<=discount<7);
}
```

C Translation

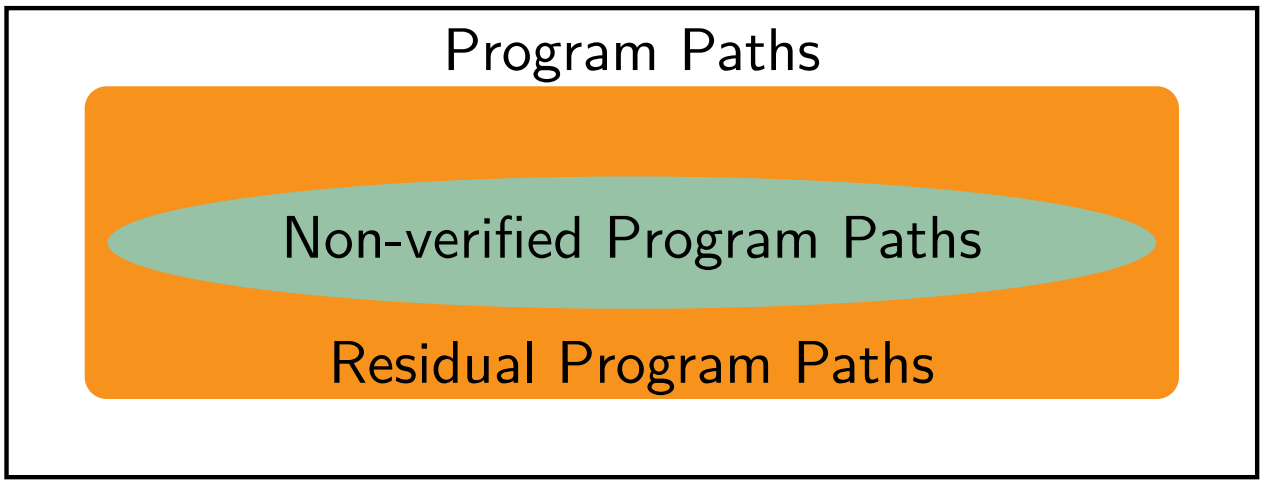
Reducer: Soundness

Residual Condition



Reducer: Soundness

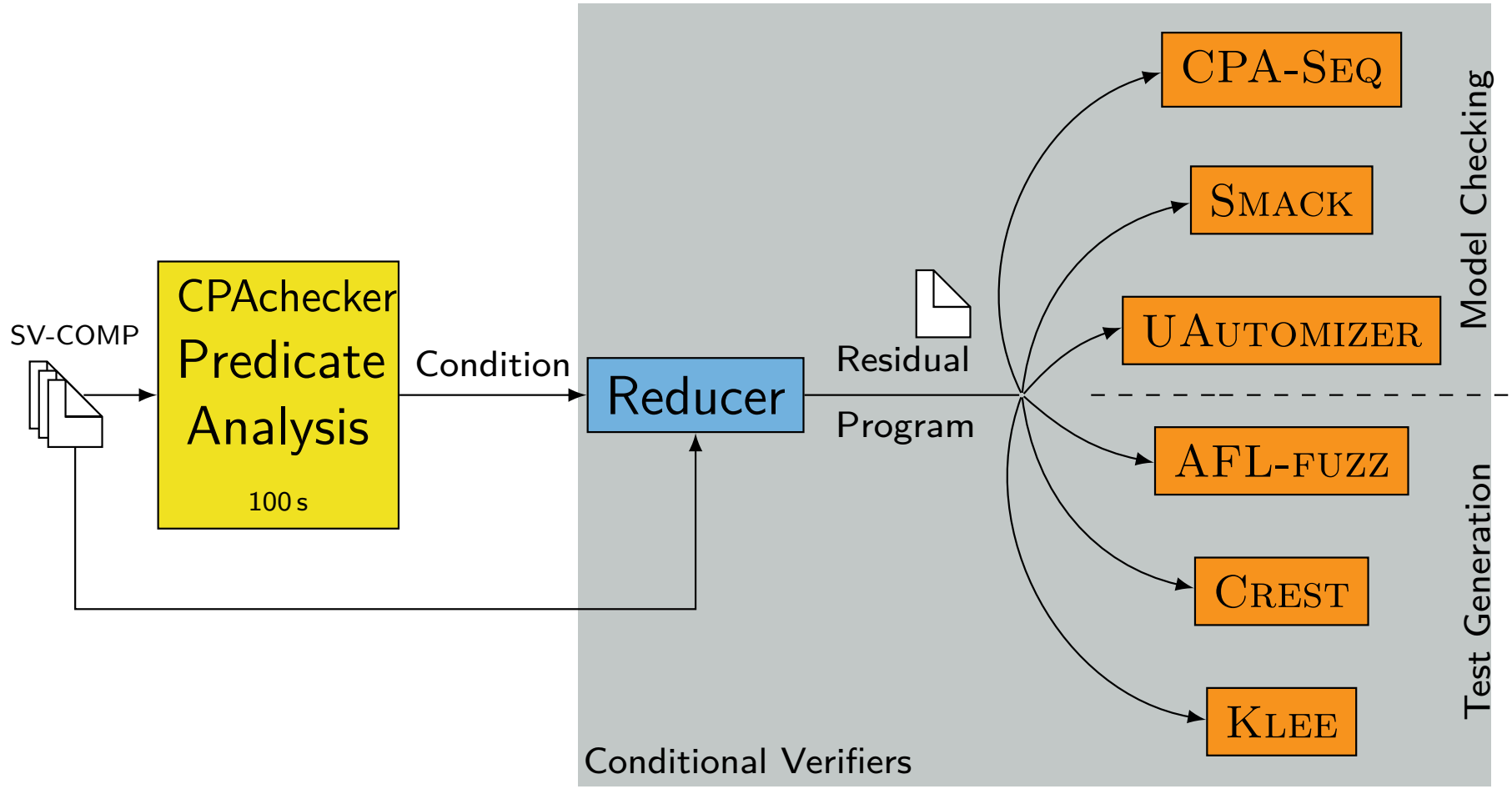
Residual Condition



Theorem

Presented reducer fulfills residual condition.

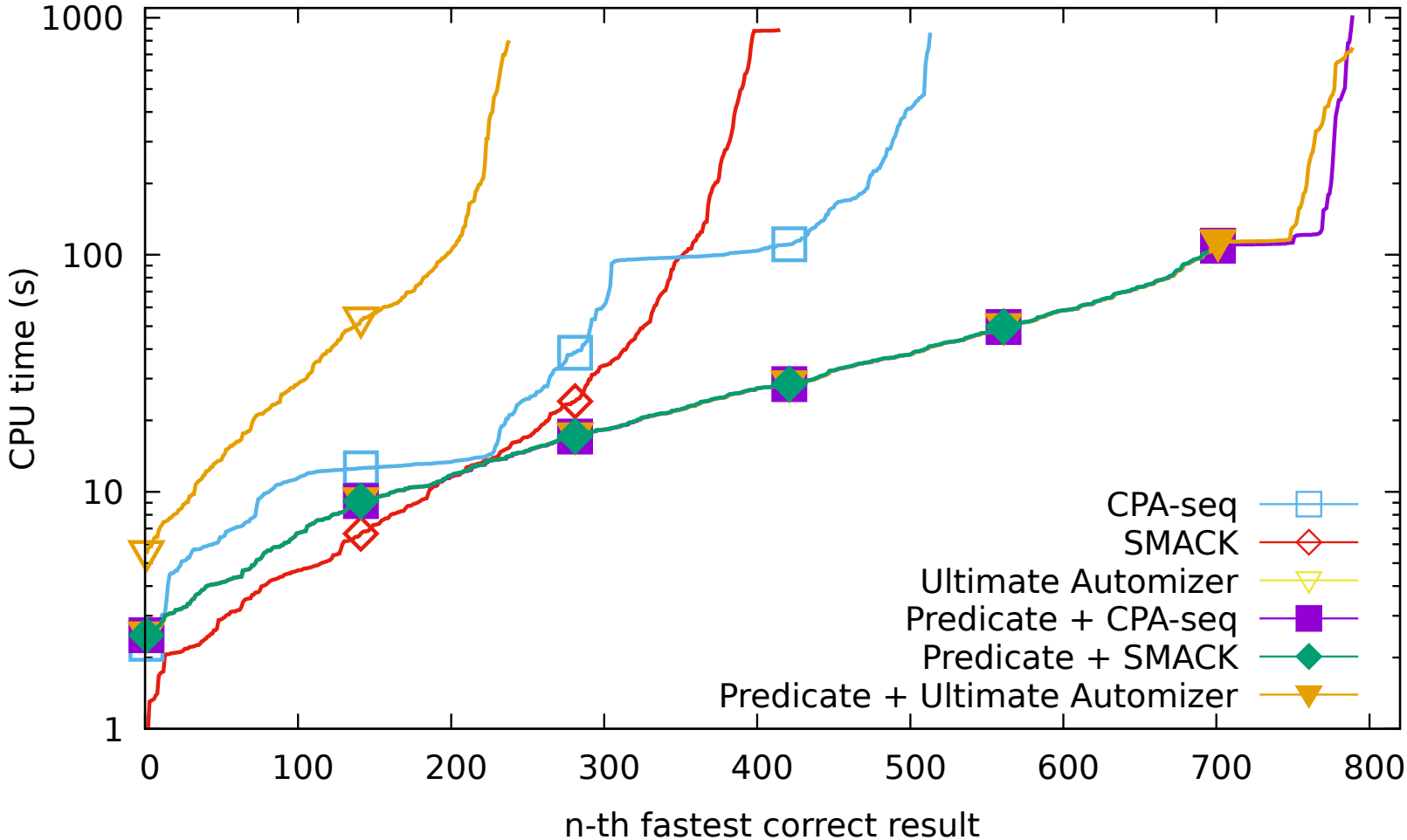
Evaluation Setup



Small Extract of Results

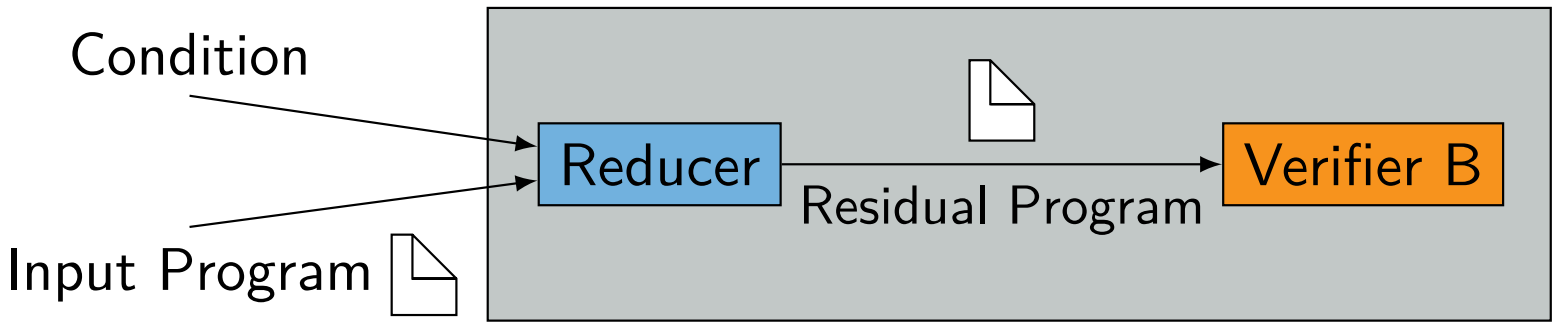
Task	R	CPA-Seq S	CPA-Seq t(s)	UAutomizer S	UAutomizer t(s)	Predicate +Reducer +CPA-Seq S	Predicate +Reducer +CPA-Seq t(s)	Predicate +Reducer +UAutomizer S	Predicate +Reducer +UAutomizer t(s)
P15I01	T	✗	910	✗	900	✓	120	✓	130
flood4	T	✗	910	✗	910	✓	450	✗	1100
newt3_6	F	✗	950	✗	490	✗	910	✓	260
P07I38	T	✗	950	✗	910	✗	1100	✓	470

Effectiveness on Hard Tasks



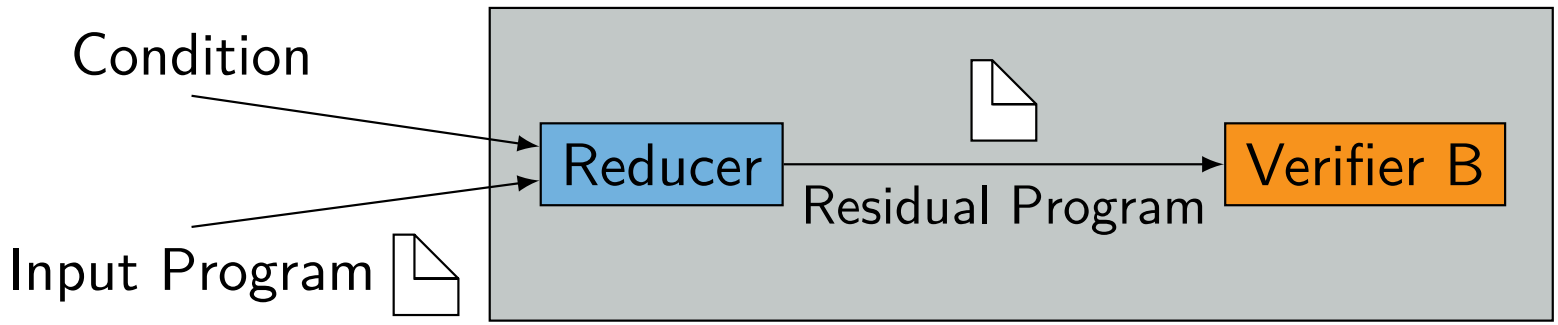
Conclusion

- Template-based conditional verifier construction



Conclusion

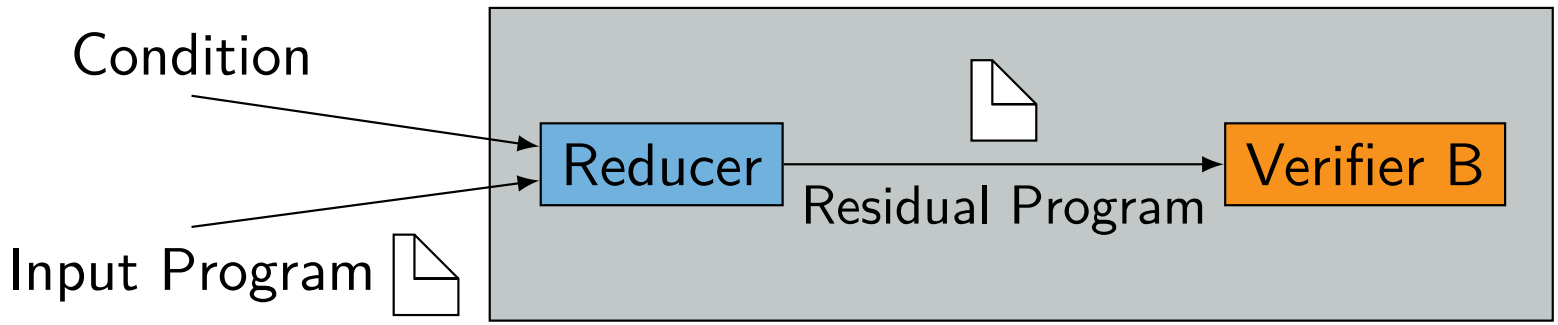
- Template-based conditional verifier construction



- One Reducer
  - Proven sound
  - Used in many conditional verifiers

Conclusion

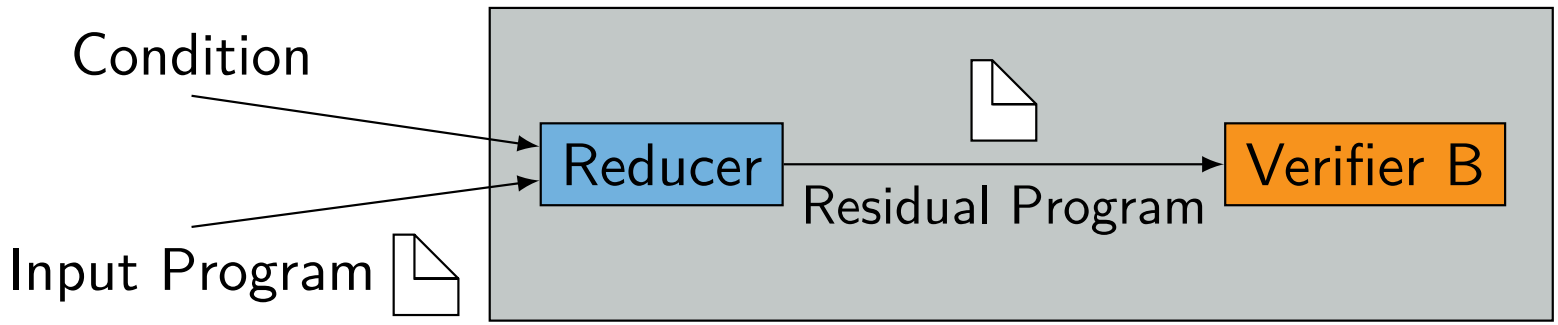
- Template-based conditional verifier construction



- One Reducer
  - Proven sound
  - Used in many conditional verifiers
- Effective on hard tasks for verifiers and test tools

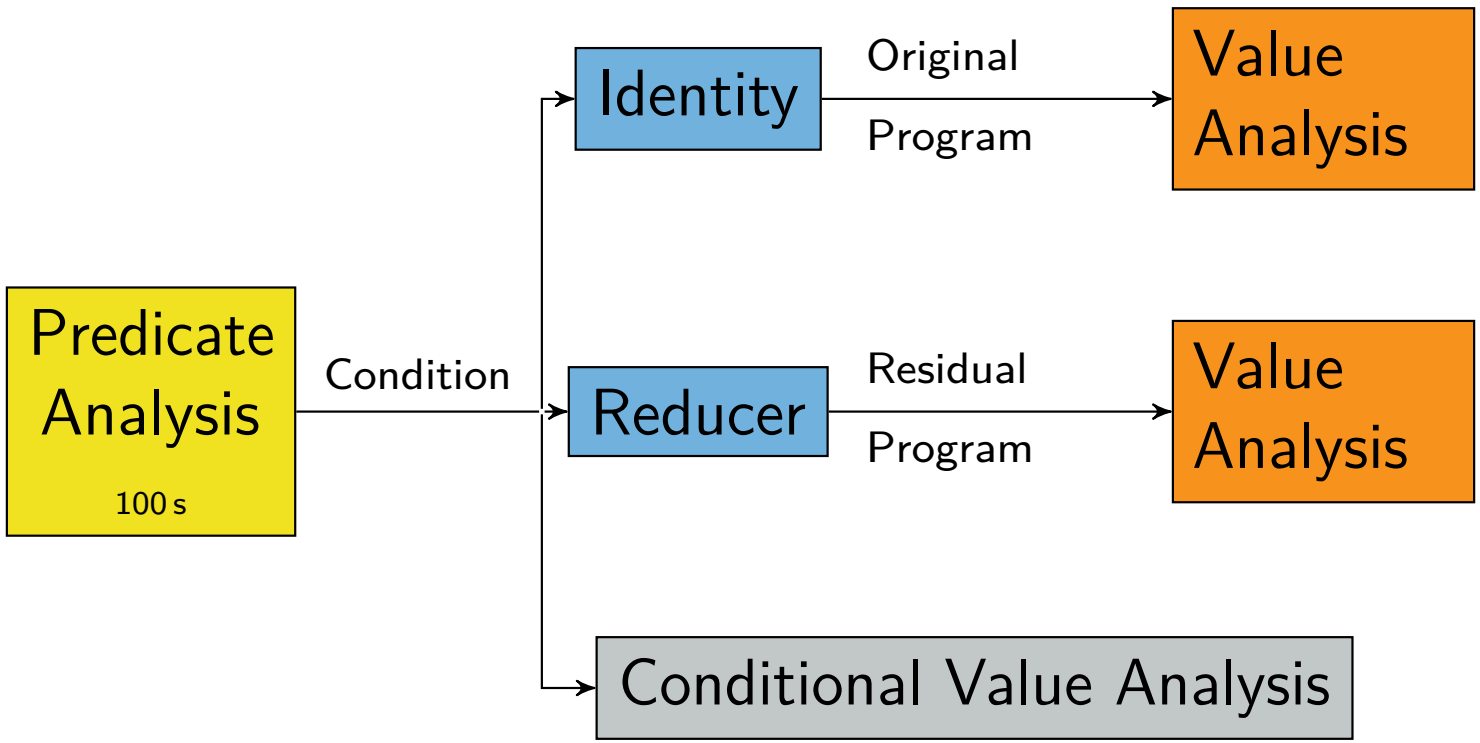
Conclusion

- Template-based conditional verifier construction

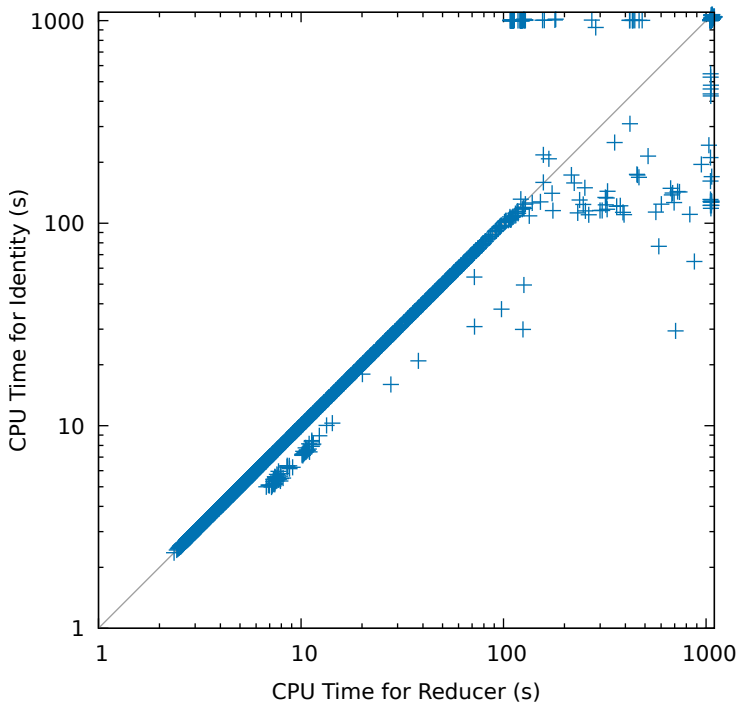


- One Reducer
  - Proven sound
  - Used in many conditional verifiers
- Effective on hard tasks for verifiers and test tools
- Future Work
  - More reducers
  - Using conditions from other tools

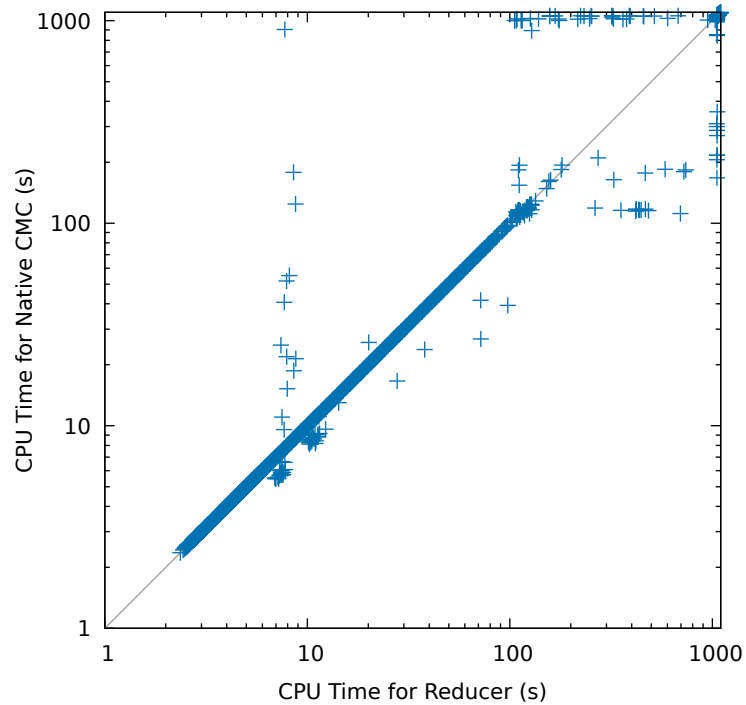
Comparison Setup



Comparison Results



(a) Identity vs.reducer



(b) Native vs. reducer-based

References I


D. Beyer, T. A. Henzinger, M. E. Keremoglu, and P. Wendler. Conditional Model Checking: A Technique to Pass Information Between Verifiers. In *Proc. FSE*. ACM, 2012.