Università della Svizzera italiana

Facoltà di scienze informatiche

Visualizing SMT-Based Parallel Constraint Solving

Jelena Budakovic, Amedeo Zucchetti, Matteo Marescotti,
Antti E. J. Hyvärinen, and Natasha Sharygina

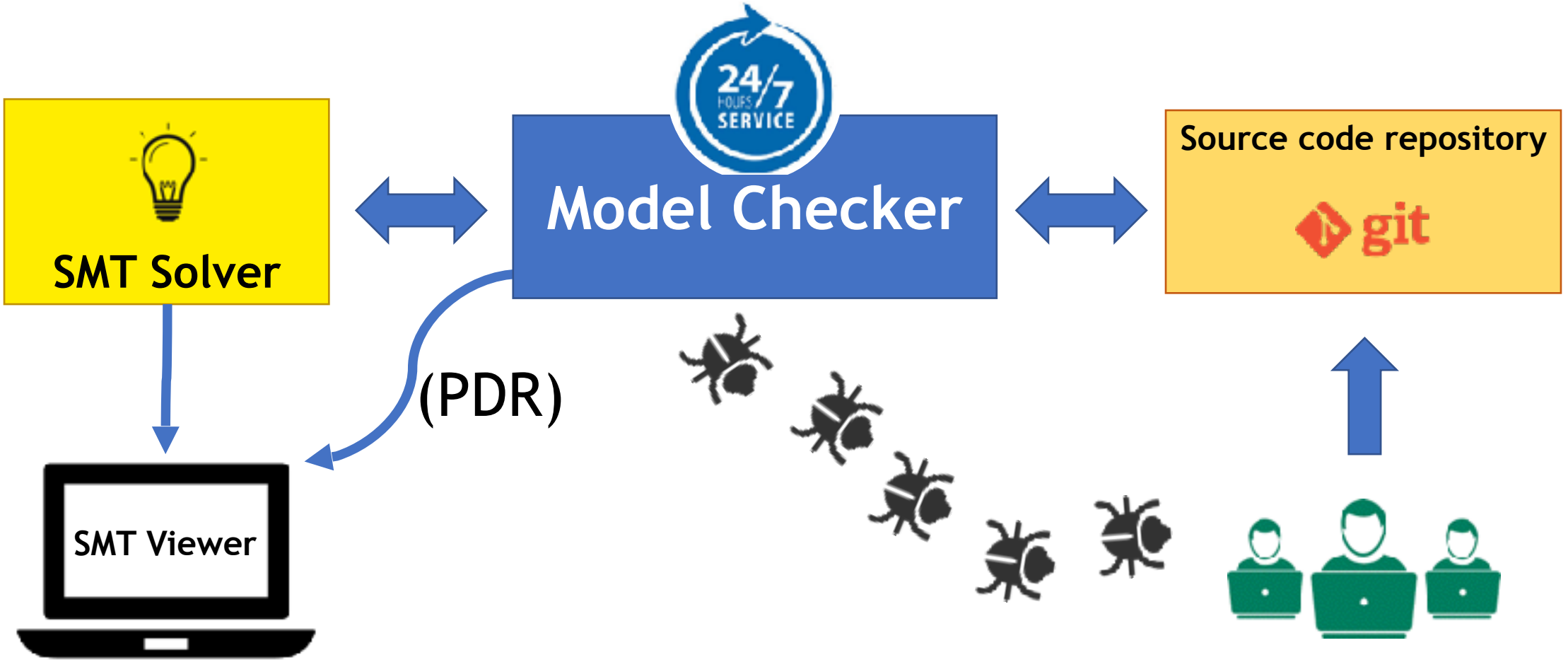
Università della Svizzera italiana
Switzerland

SMT@CAV2017

Motivations

- SMT
  - Very expressive language
  - Problem's intrinsic high complexity (SAT + decision procedures)
- SMT solvers:
  - Widely used for modelling
  - Highly optimized sequentially
- Parallel SMT:
  - Aims to solve more and faster
  - Parallel computing is *difficult!*

Verification Use Case



Parallelize the work

- Portfolio:
  - Many search-independent processes
- Partitioning:
  - The **problem** is partitioned to several *sub-problems* such that:
    - **problem** is SAT: exists a *sub-problem* SAT
    - **problem** is UNSAT: all *sub-problems are* UNSAT
- **Combination of the two above**

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)

Parallelization Tree Framework

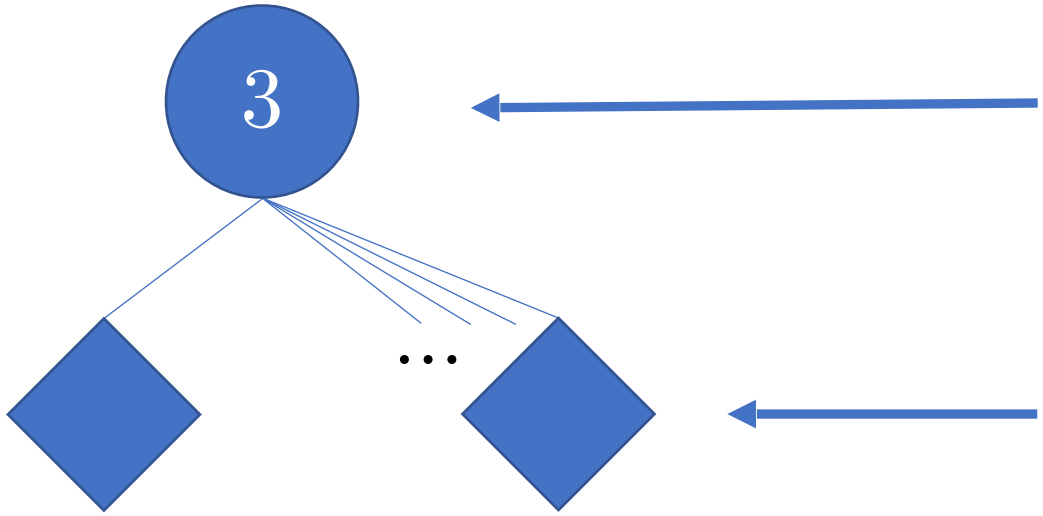
SAT 2015, ATVA 2016, FMCAD 2017 (P3)

Circle: original instance

3 solvers **portfolio**

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

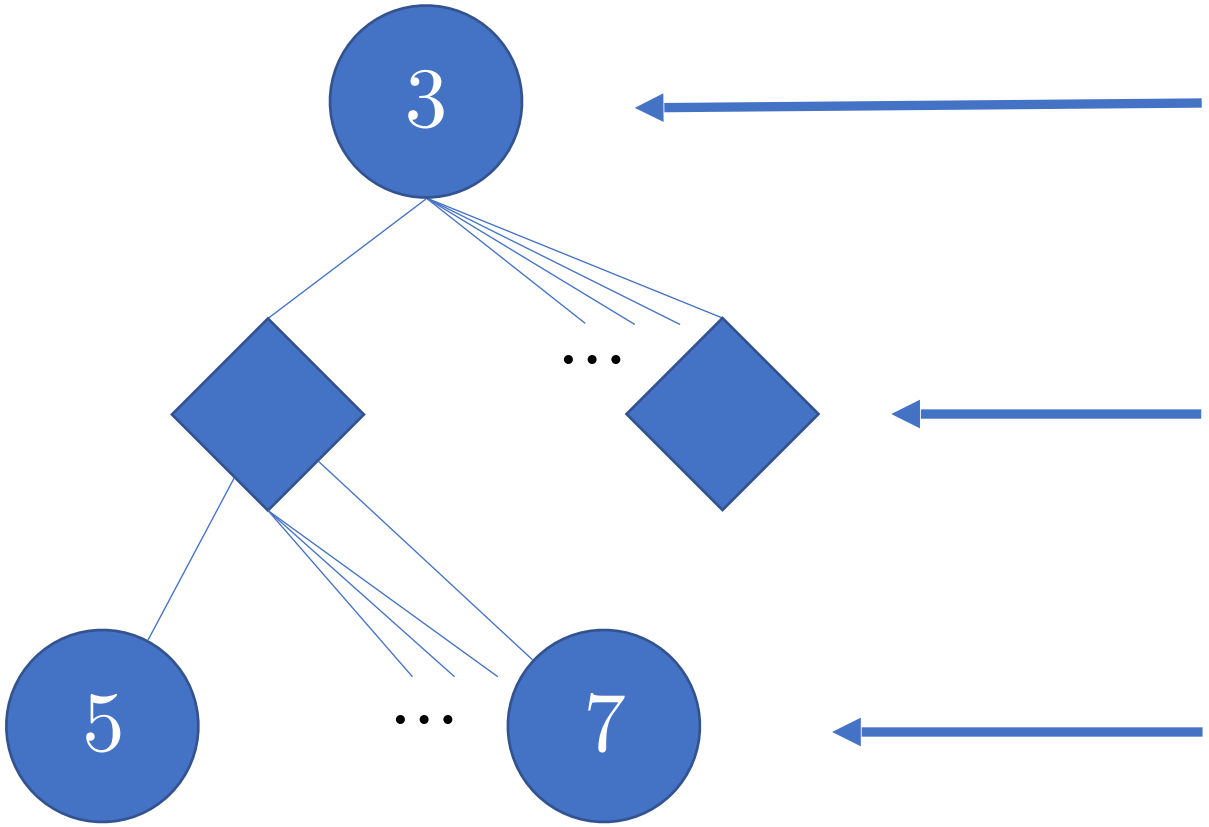
3 solvers **portfolio**

Diamond: **partitioning**

Each diamond represents a way to partition the parent's instance.

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

3 solvers **portfolio**

Diamond: **partitioning**

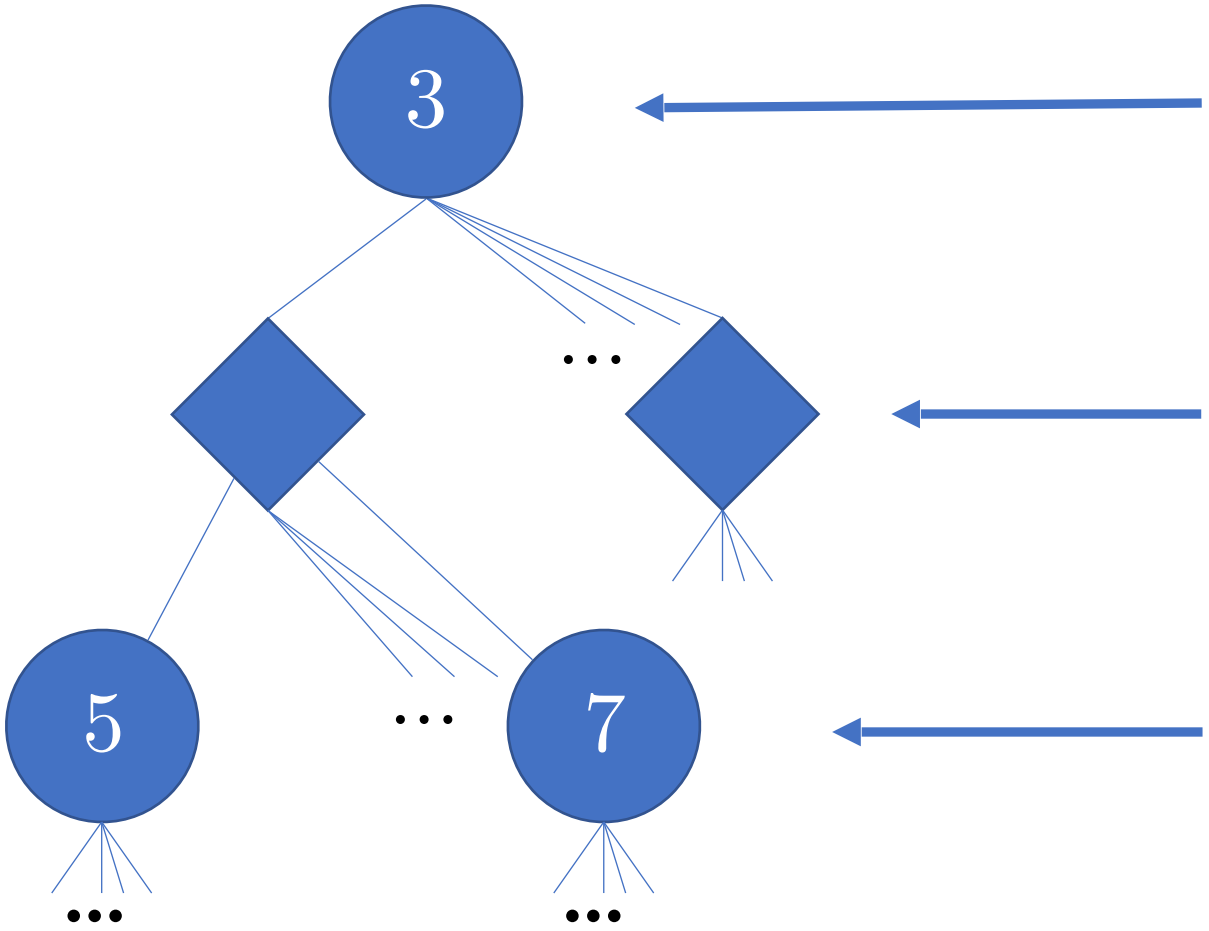
Each diamond represents a way to partition the parent's instance.

Circle: a partition of the parent's instance

Each node could possibly be the root of a new tree.

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

3 solvers **portfolio**

Diamond: **partitioning**

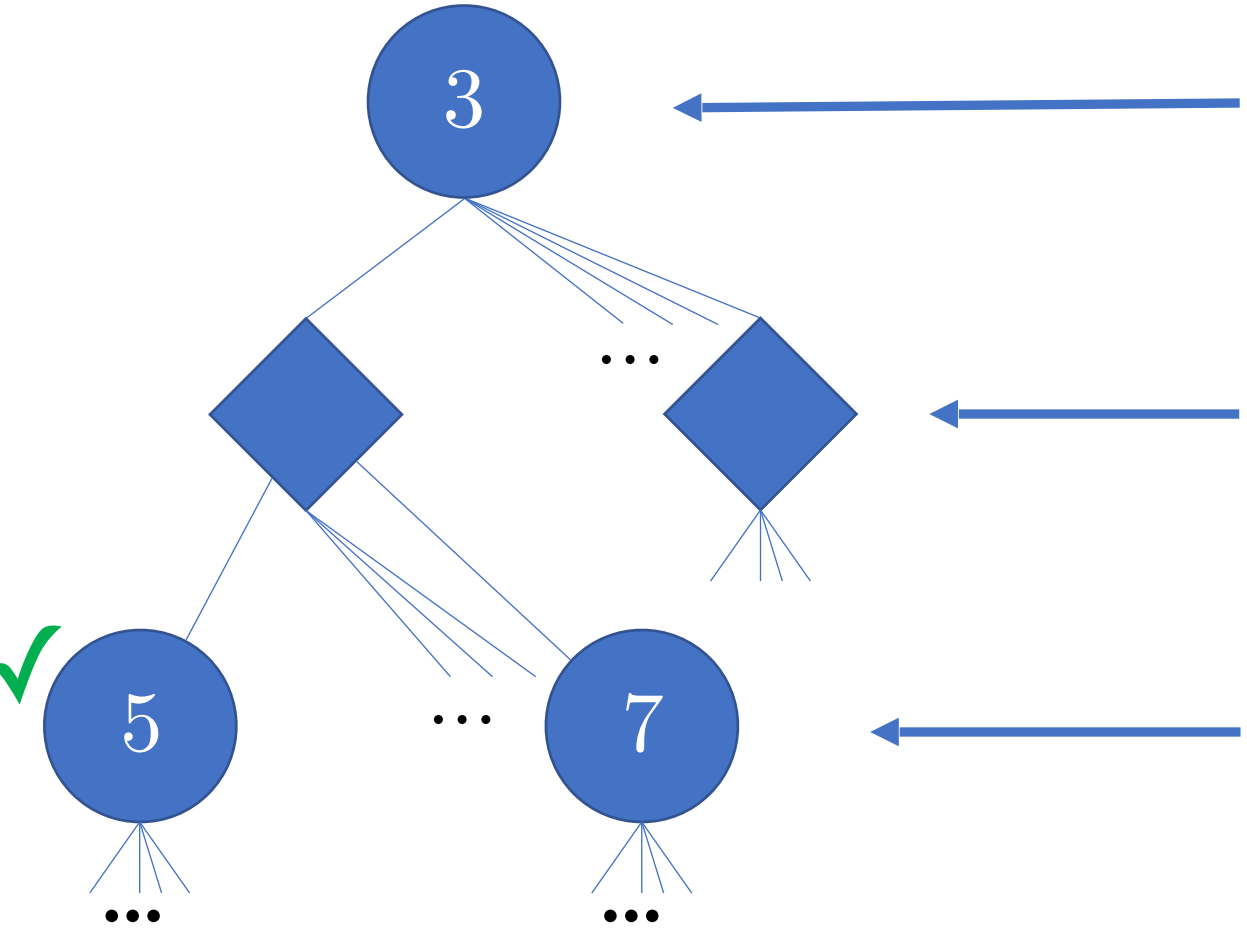
Each diamond represents a way to partition the parent's instance.

Circle: a partition of the parent's instance

Each node could possibly be the root of a new tree.

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

3 solvers **portfolio**

Diamond: **partitioning**

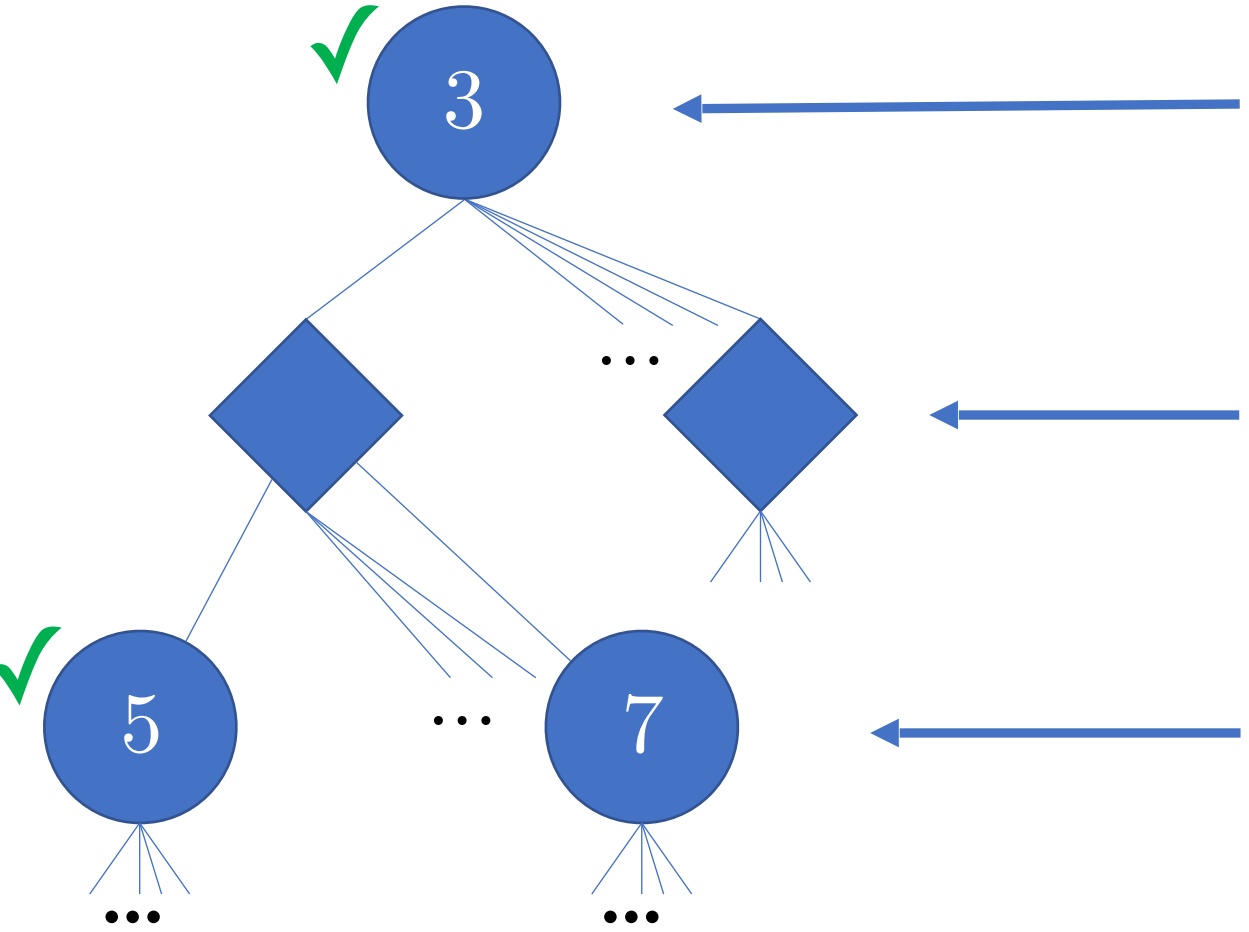
Each diamond represents a way to partition the parent's instance.

Circle: a partition of the parent's instance

Each node could possibly be the root of a new tree.

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

3 solvers **portfolio**

Diamond: **partitioning**

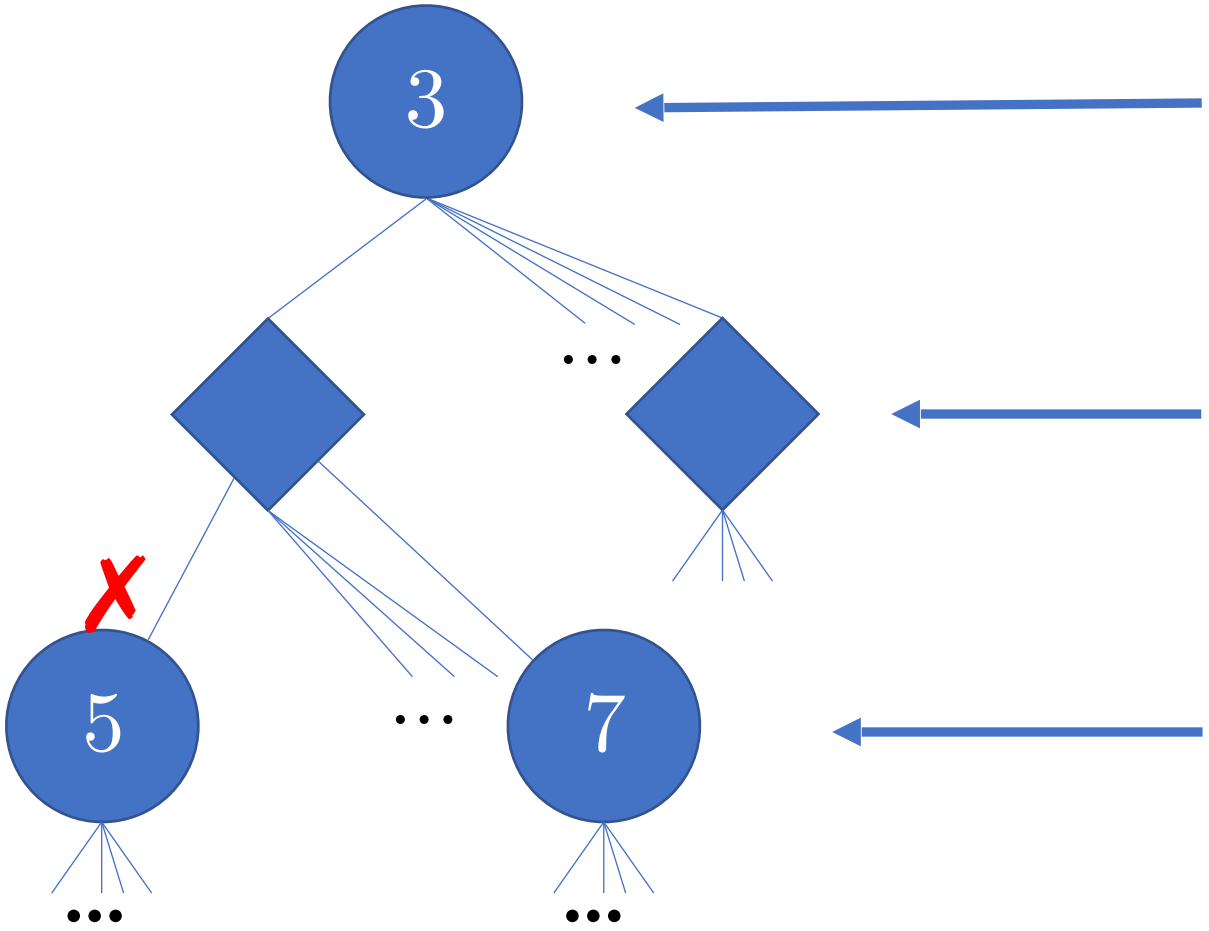
Each diamond represents a way to partition the parent's instance.

Circle: a partition of the parent's instance

Each node could possibly be the root of a new tree.

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

3 solvers **portfolio**

Diamond: **partitioning**

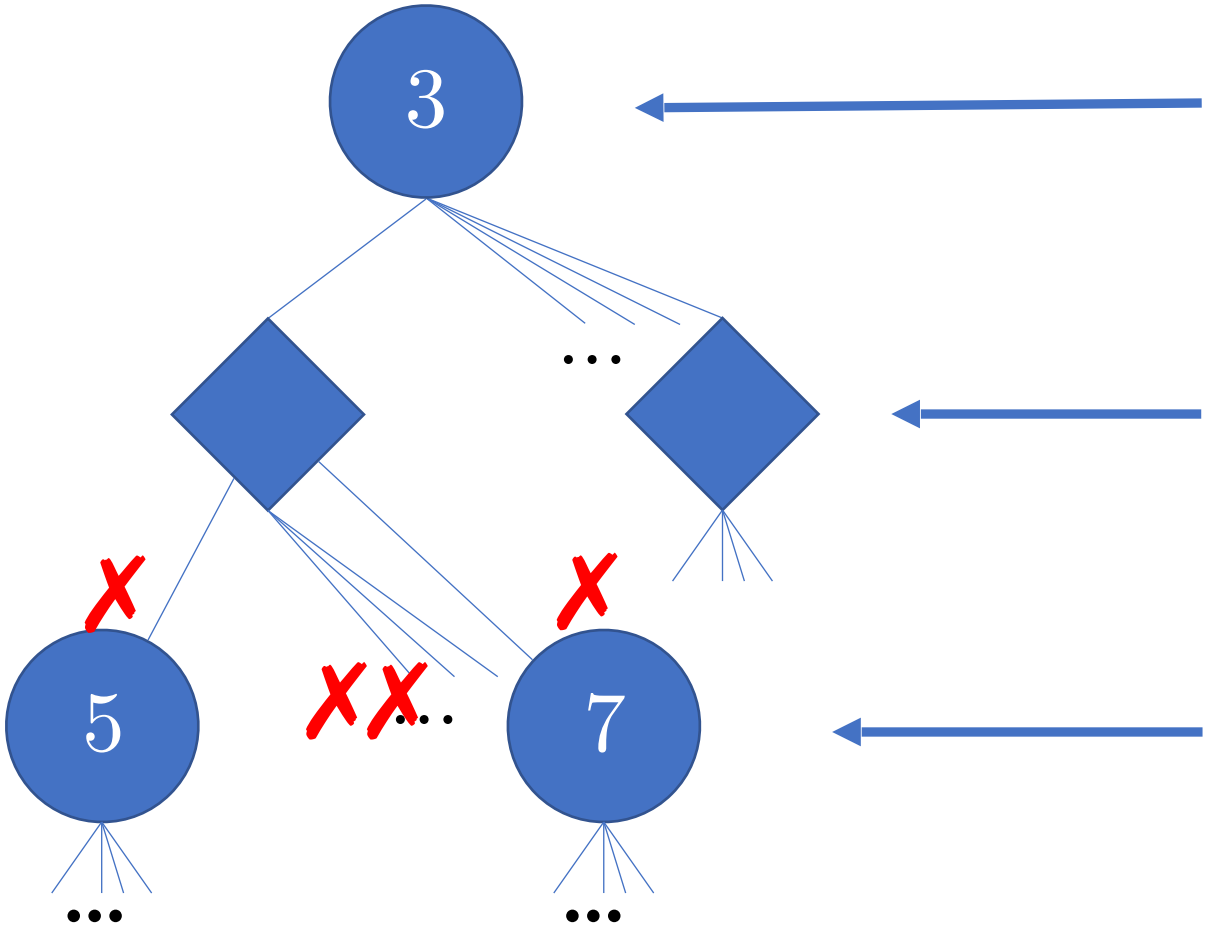
Each diamond represents a way to partition the parent's instance.

Circle: a partition of the parent's instance

Each node could possibly be the root of a new tree.

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

3 solvers **portfolio**

Diamond: **partitioning**

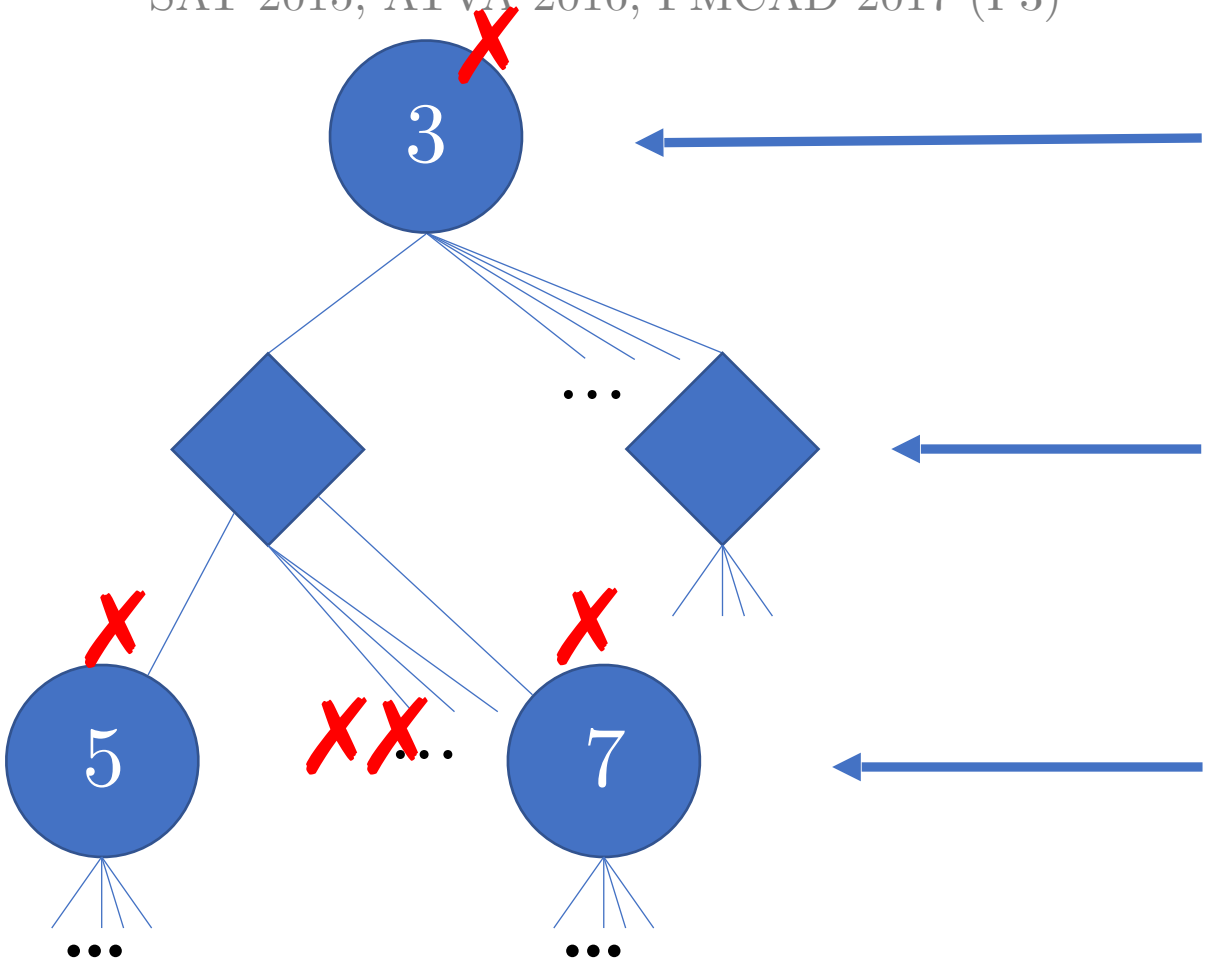
Each diamond represents a way to partition the parent's instance.

Circle: a partition of the parent's instance

Each node could possibly be the root of a new tree.

Parallelization Tree Framework

SAT 2015, ATVA 2016, FMCAD 2017 (P3)



Circle: original instance

3 solvers **portfolio**

Diamond: **partitioning**

Each diamond represents a way to partition the parent's instance.

Circle: a partition of the parent's instance

Each node could possibly be the root of a new tree.

SMT Portfolio & Partitioning

- Portfolio:
  - Solver random seed

- Partitioning:
  - Search space partitioning (SAT)
  - Jointly exhaustive assumptions *(disjunction is a tautology)*

PDR Portfolio & Partitioning

- Portfolio:
  - **PDR strategy**
  - Underlying SMT solver's random seed

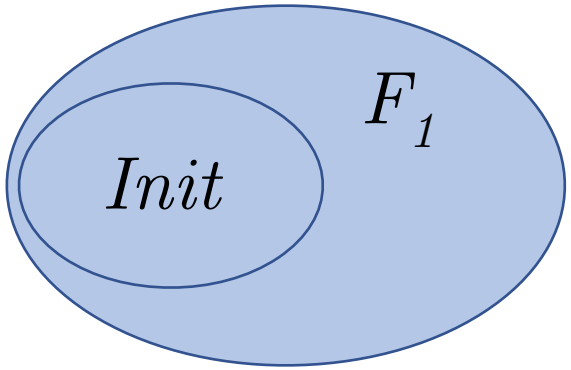
- Partitioning:
  - **PDR partitioning**

PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$ $\qquad$ $Init \implies \neg Bad$

PDR

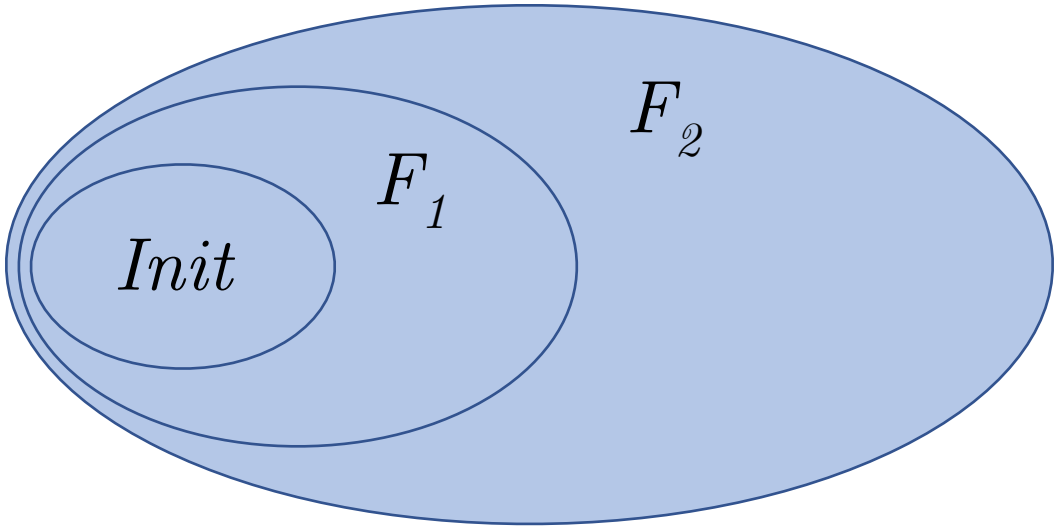
$\langle Init(X), Tr(X, X'), Bad(X) \rangle$ $\qquad$ $Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

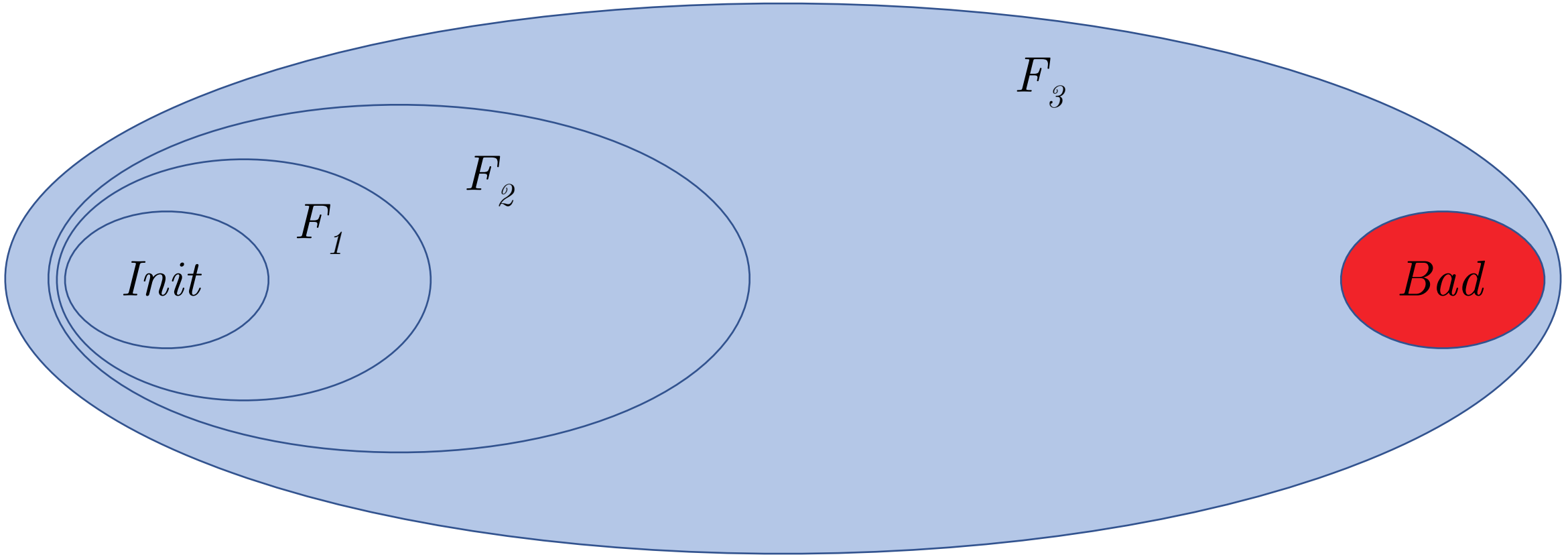
$Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

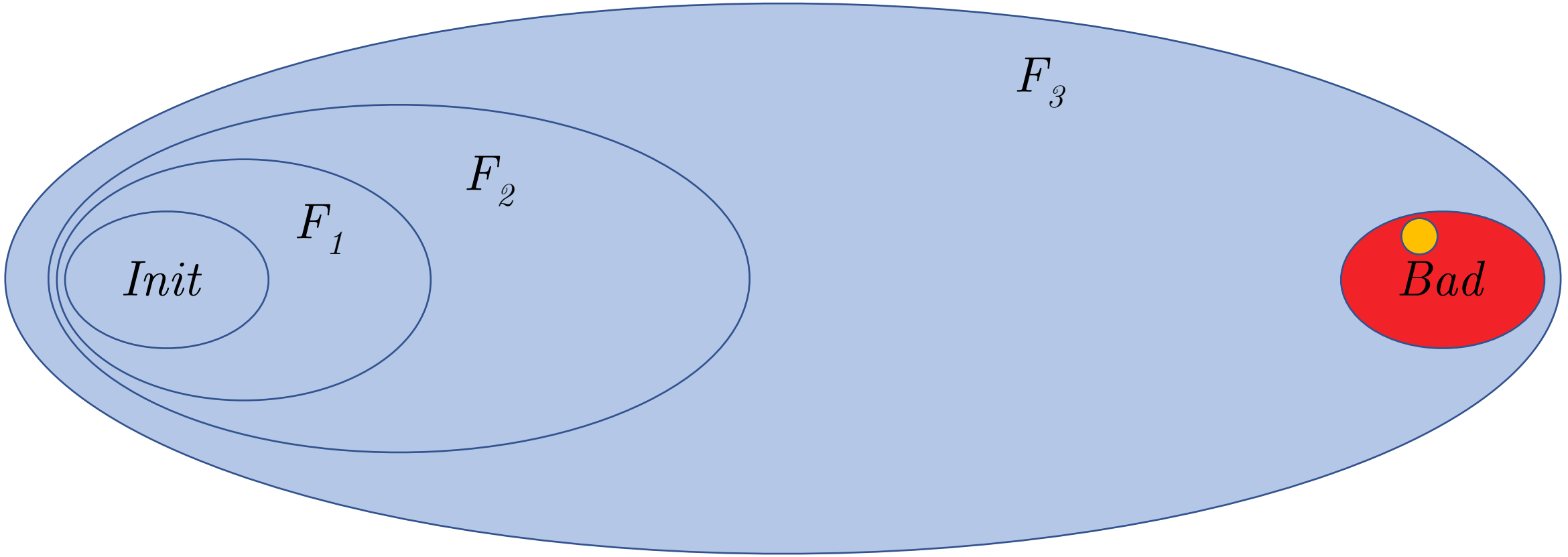
$Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

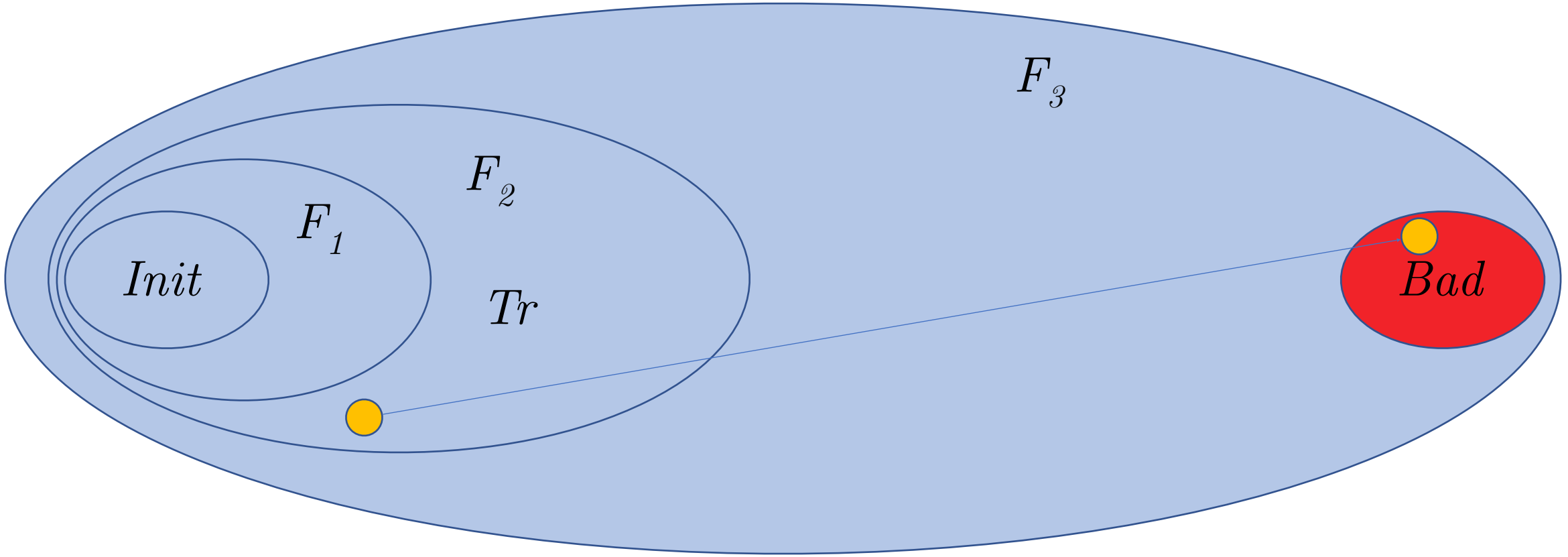
$Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

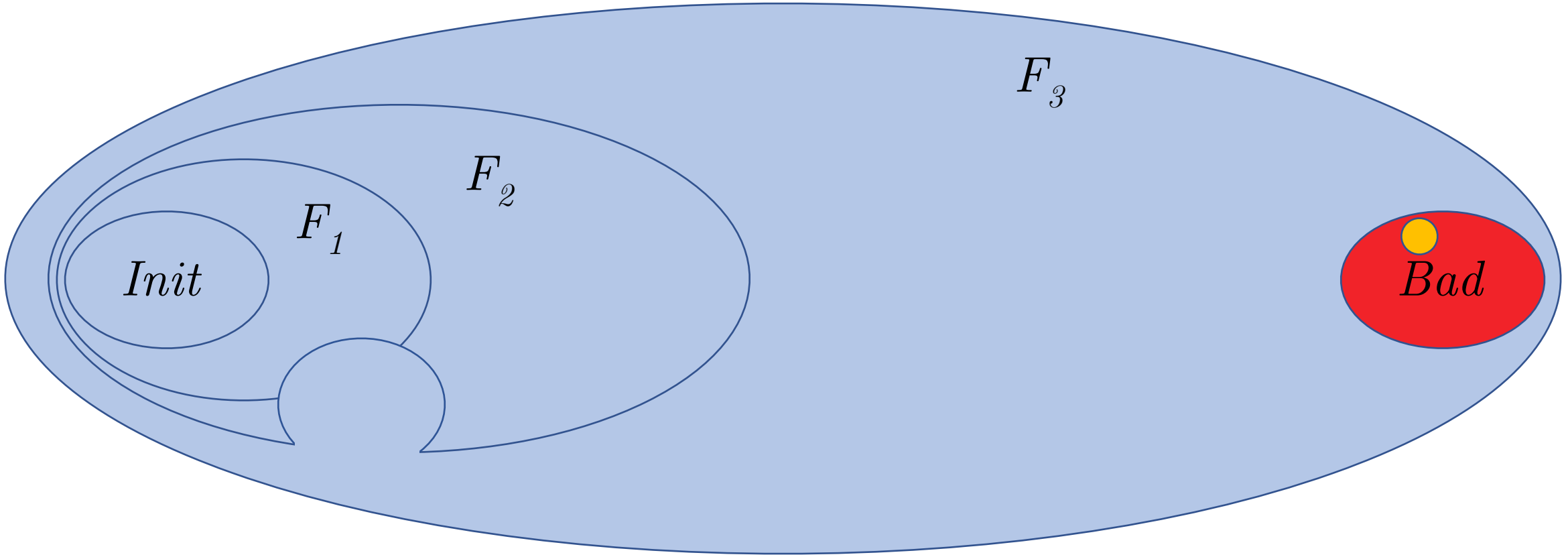
$Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

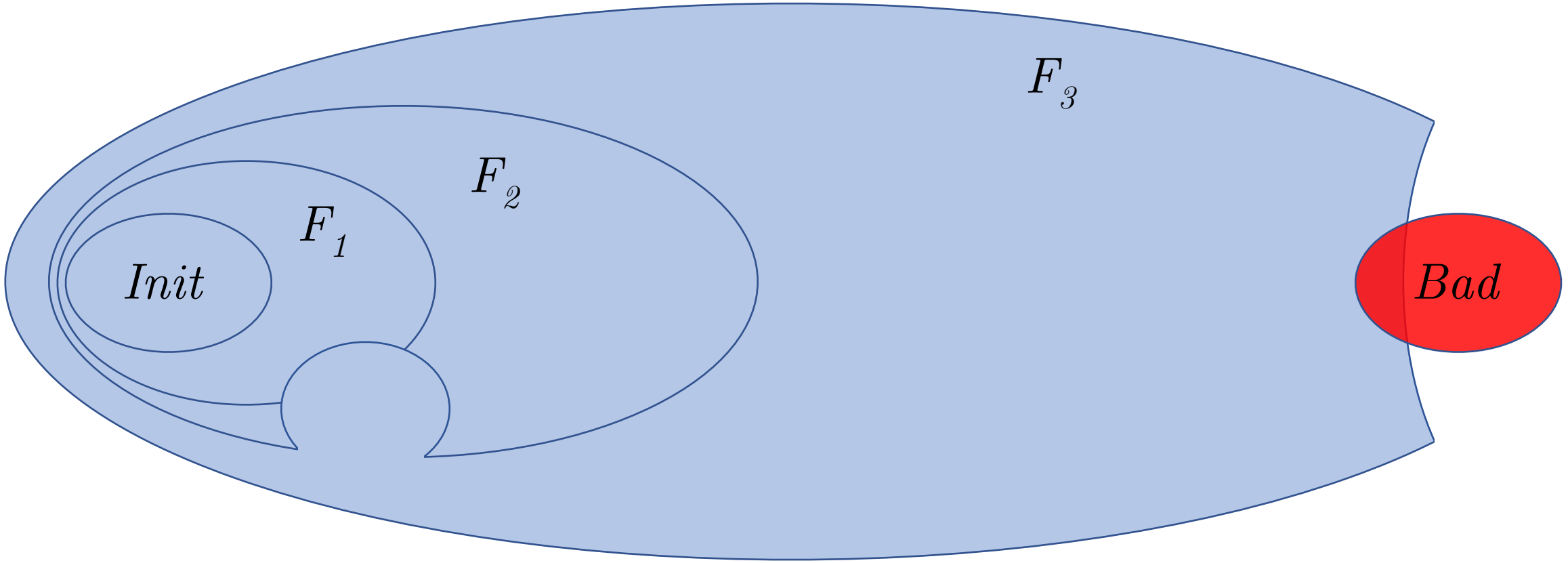
$Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

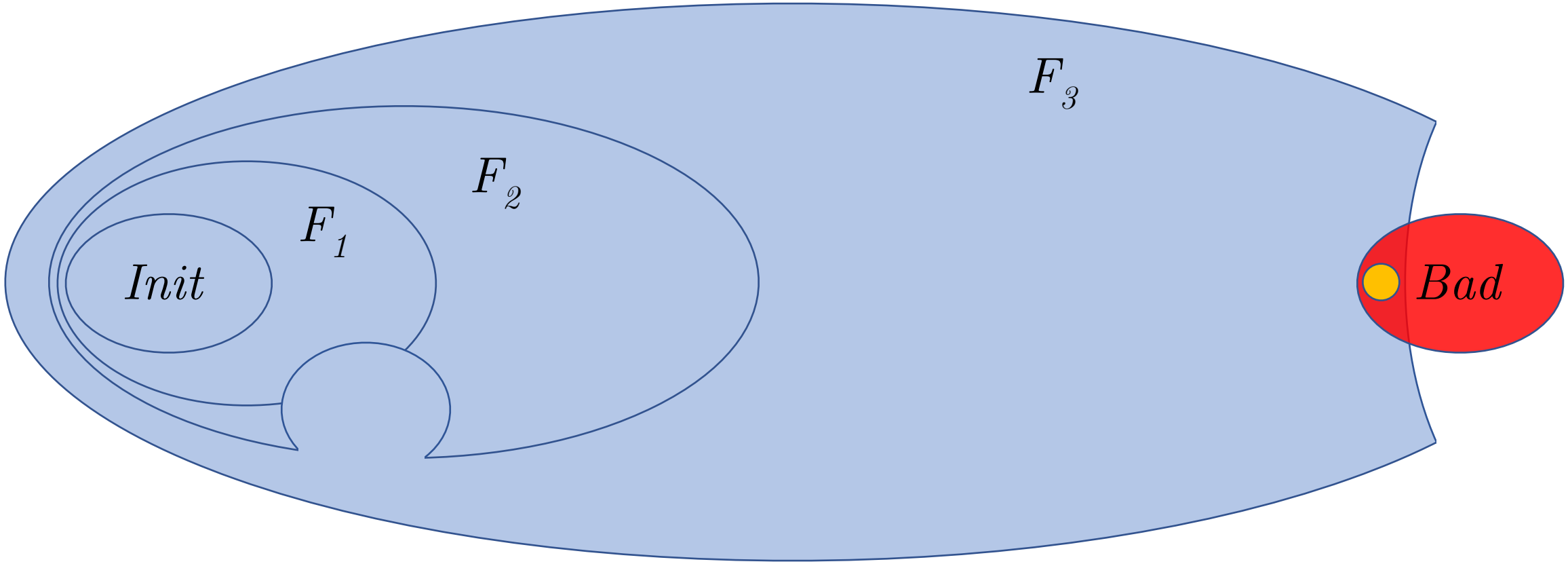
$Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

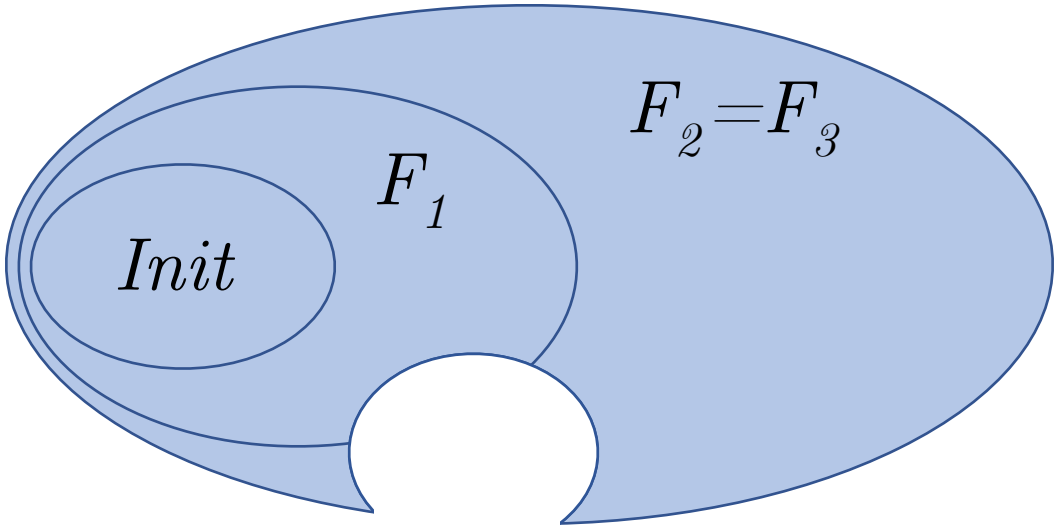
$Init \implies \neg Bad$



PDR

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

$Init \implies \neg Bad$



PDR Partitioning

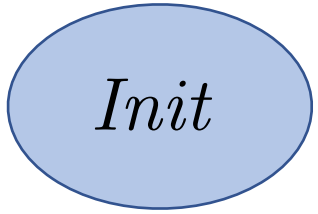
FMCAD 2017

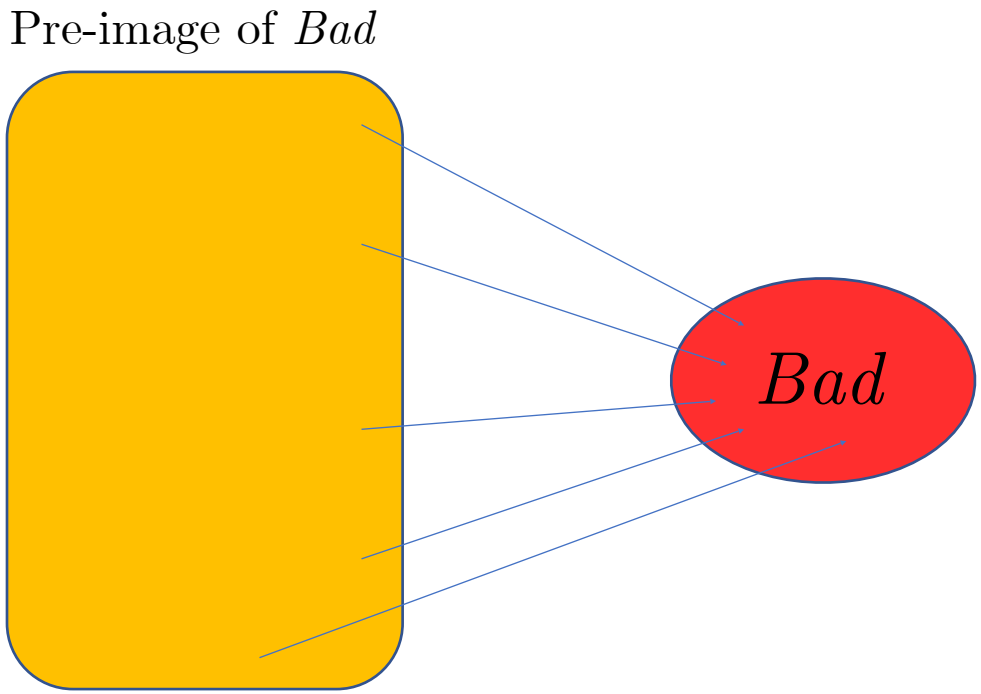
$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

PDR Partitioning

FMCAD 2017

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

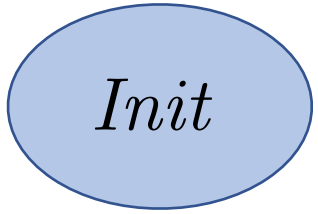


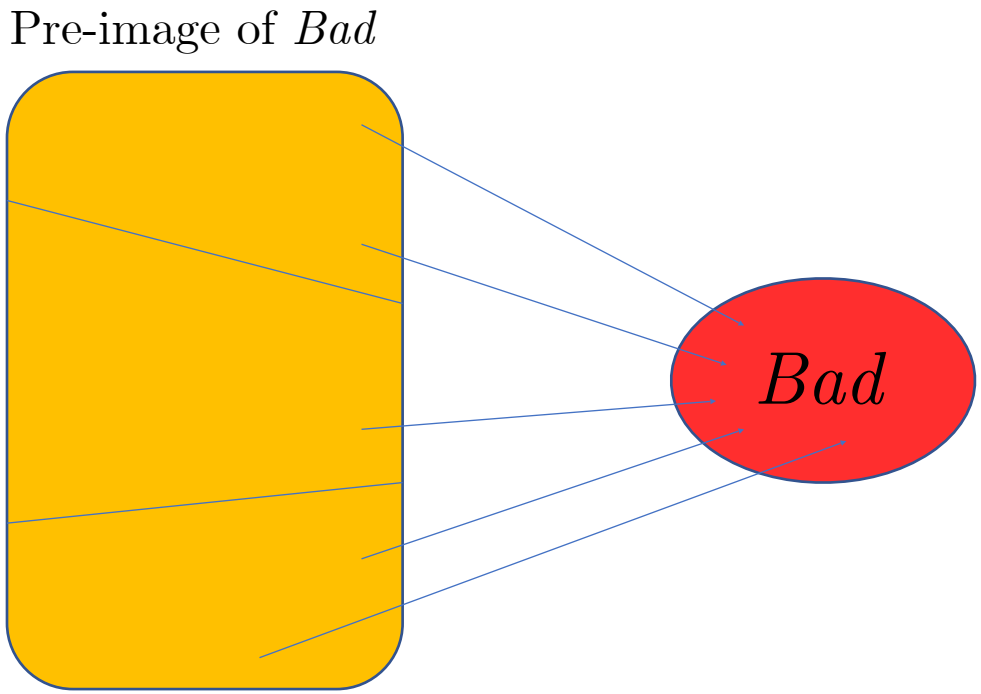


PDR Partitioning

FMCAD 2017

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

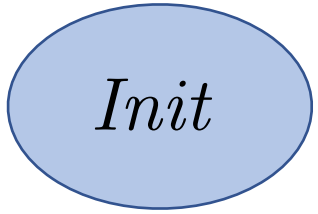


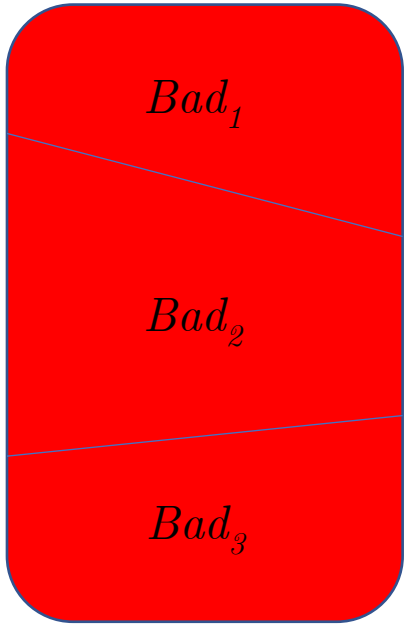


PDR Partitioning

FMCAD 2017

$\langle Init(X), Tr(X, X'), Bad(X) \rangle$

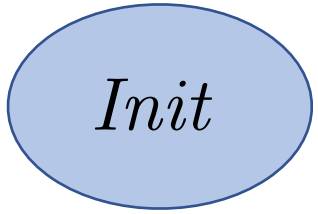


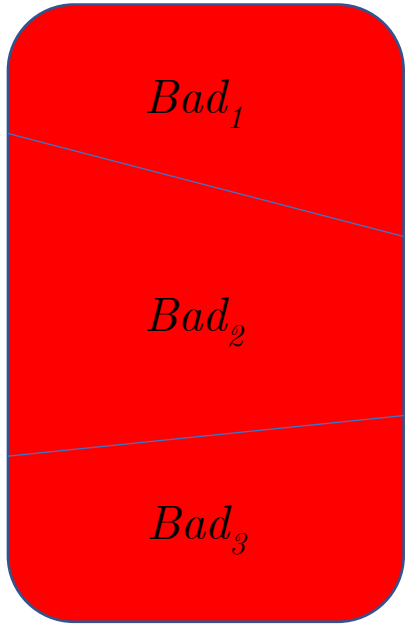


PDR Partitioning

FMCAD 2017

$$\langle Init(X), Tr(X, X'), Bad(X) \rangle \equiv \begin{array}{l} \langle Init(X), Tr(X, X'), Bad_1(X) \rangle \\ \langle Init(X), Tr(X, X'), Bad_2(X) \rangle \\ \langle Init(X), Tr(X, X'), Bad_3(X) \rangle \end{array}$$





SMTViewer API

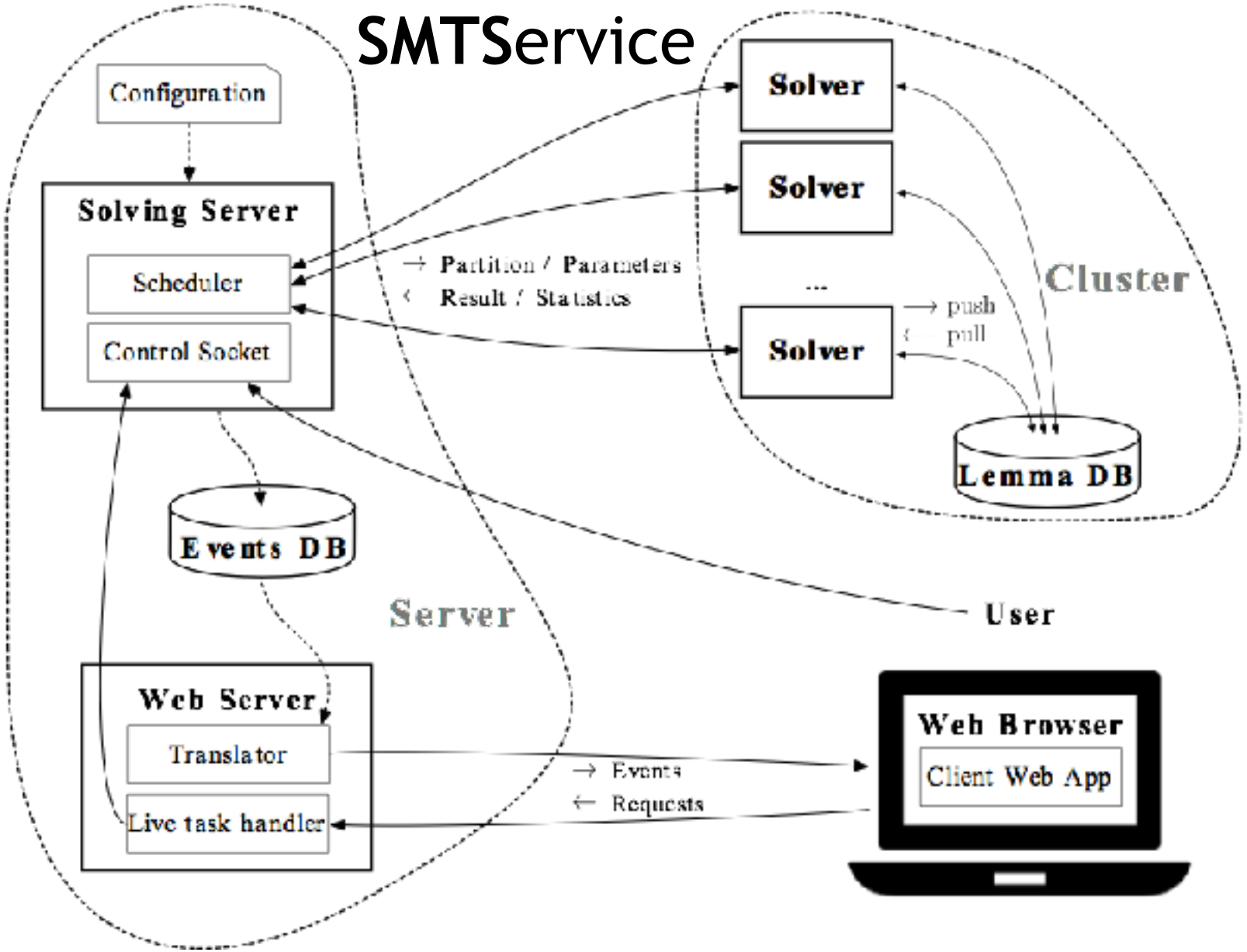
- SQLite database of events happened during parallel solving:

Column	Type	Description
`id`	Integer	Unique event id number.
`ts`	Integer	The Unix timestamp of the event.
`name`	String	Instance name.
`node`	JsonArray<Integer>	Path from the root to the node.
`event`	String	Type of the event.
`solver`	String	Solver identifier.
`data`	JsonObject<String, String>	Data associated with the event.

Analysis Demo

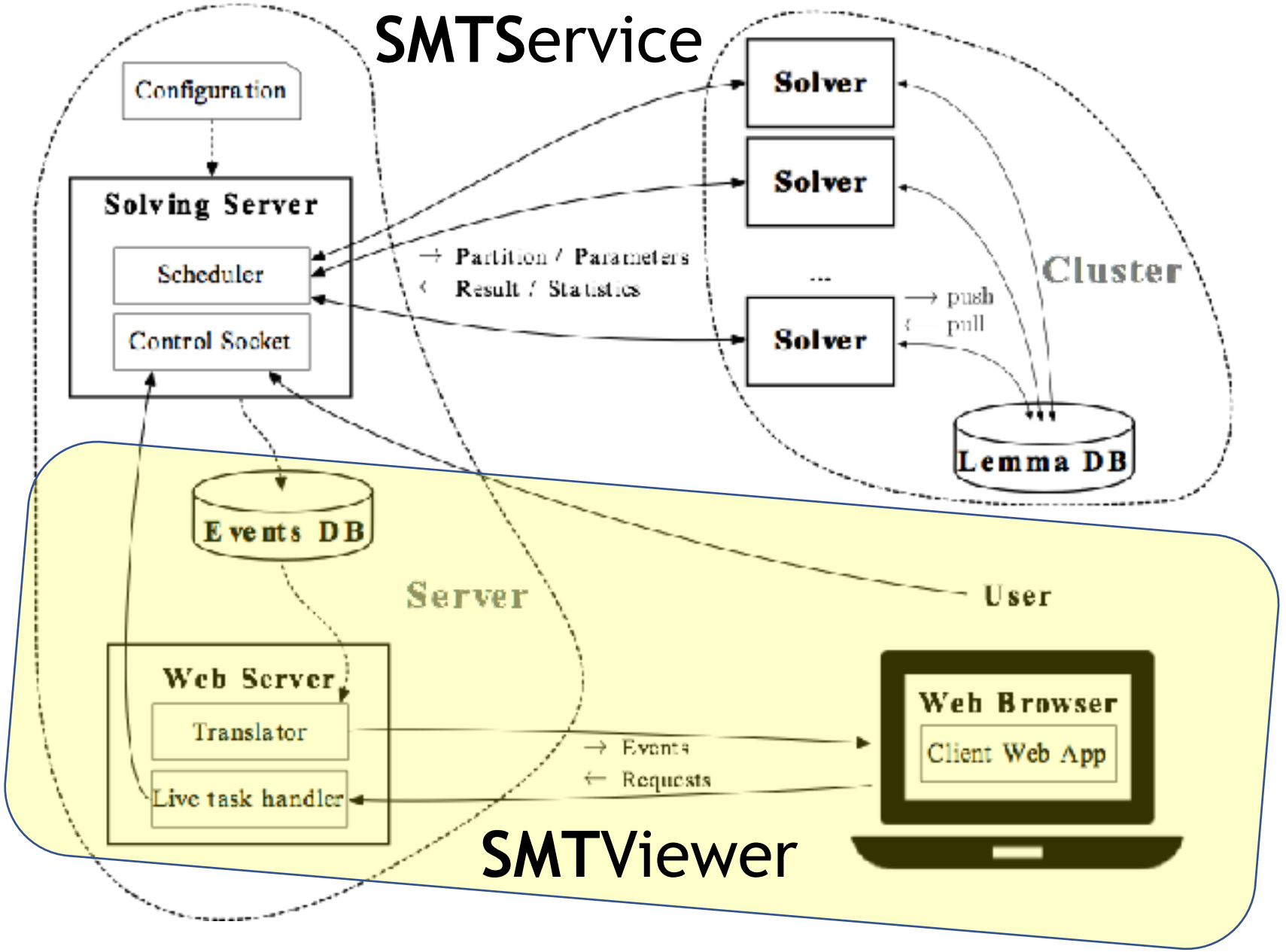
Architecture

- **SMTS**ervice:
  - SMT: OpenSMT2
  - PDR: Z3 Spacer



Architecture

- **SMTS**ervice:
  - SMT: OpenSMT2
  - PDR: Z3 Spacer
- **SMTV**iewer:
  - Real time mode
  - Analysis mode



Live Demo

Future work

- SMT formula visualization
- SMT learned clauses analysis

- PDR frame lemmas analysis

- SMTS: support for more solvers, reproducibility

- Any suggestions?

Thank you

SMTS public repository:

```
$ git clone https://scm.ti-edu.ch/repogit/smts.git
```