Task Planning with OMT: an Application to Production Logistics

Francesco Leofante[1,2] Erika Ábrahám[1] Armando Tacchella[2]

[1] RWTH Aachen University, Germany

[2] University of Genoa, Italy

IFM 2018

Maynooth, Ireland

September 6, 2018

What is this presentation about?

Planning problem

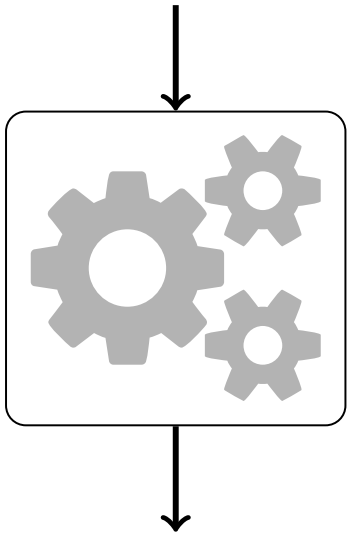
What is task planning in logistics?

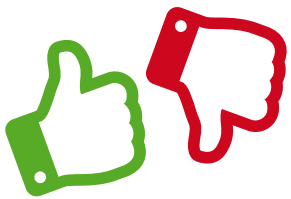
Plan

What is this presentation about?

$\neg a \wedge b \vee c$

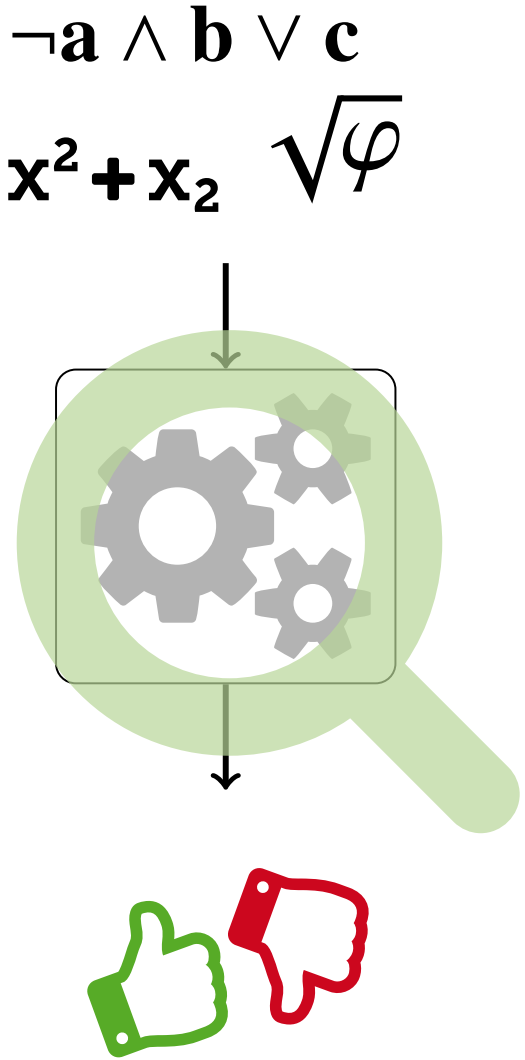
$x^2 + x_2 \quad \sqrt{\varphi}$





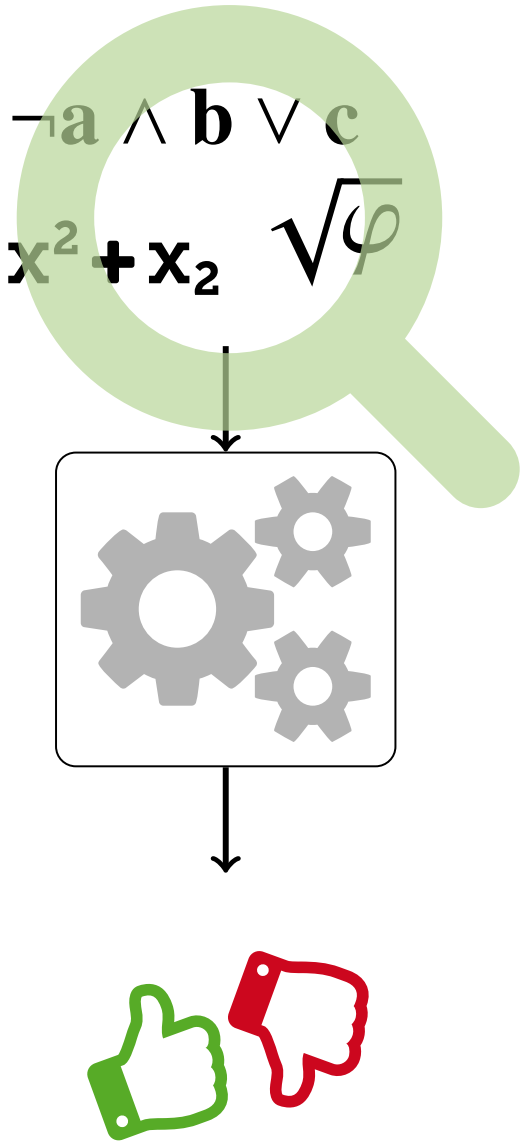
What is this presentation about?

What is OMT?



What is this presentation about?

How can we use OMT for planning?

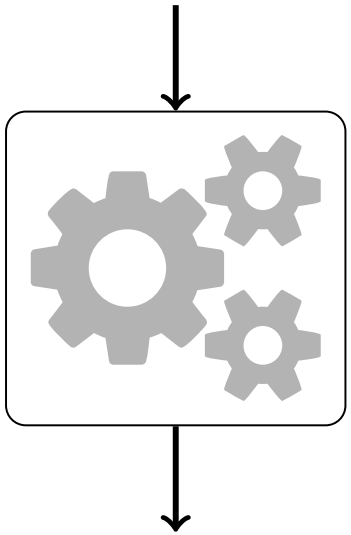


What is this presentation about?

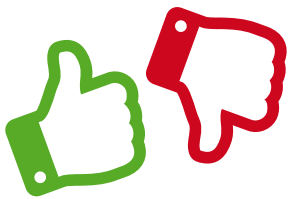
Planning problem

$\neg a \wedge b \vee c$

$x^2 + x_2 \quad \sqrt{\varphi}$



Can we use OMT for planning in logistics?

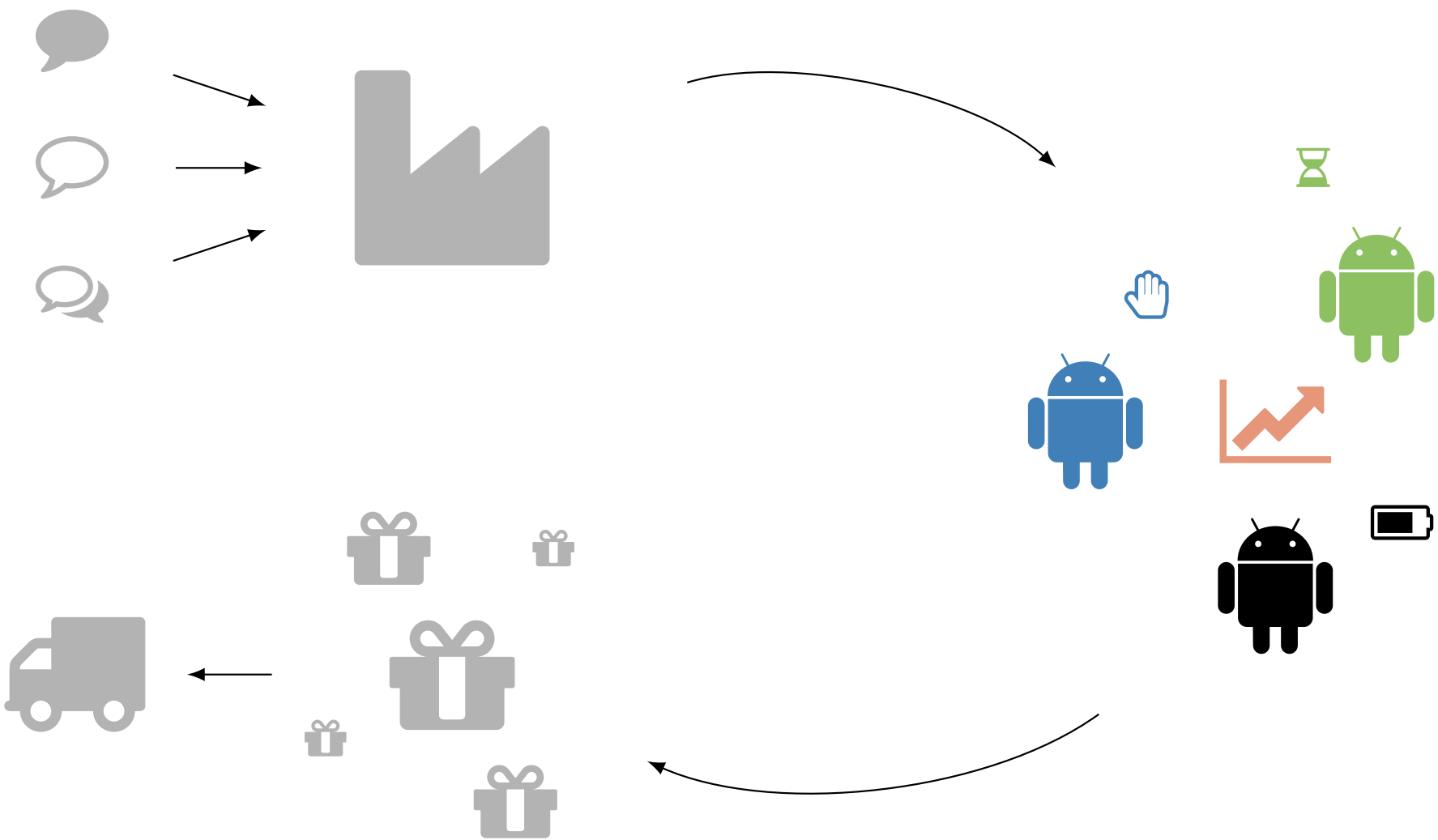


Plan

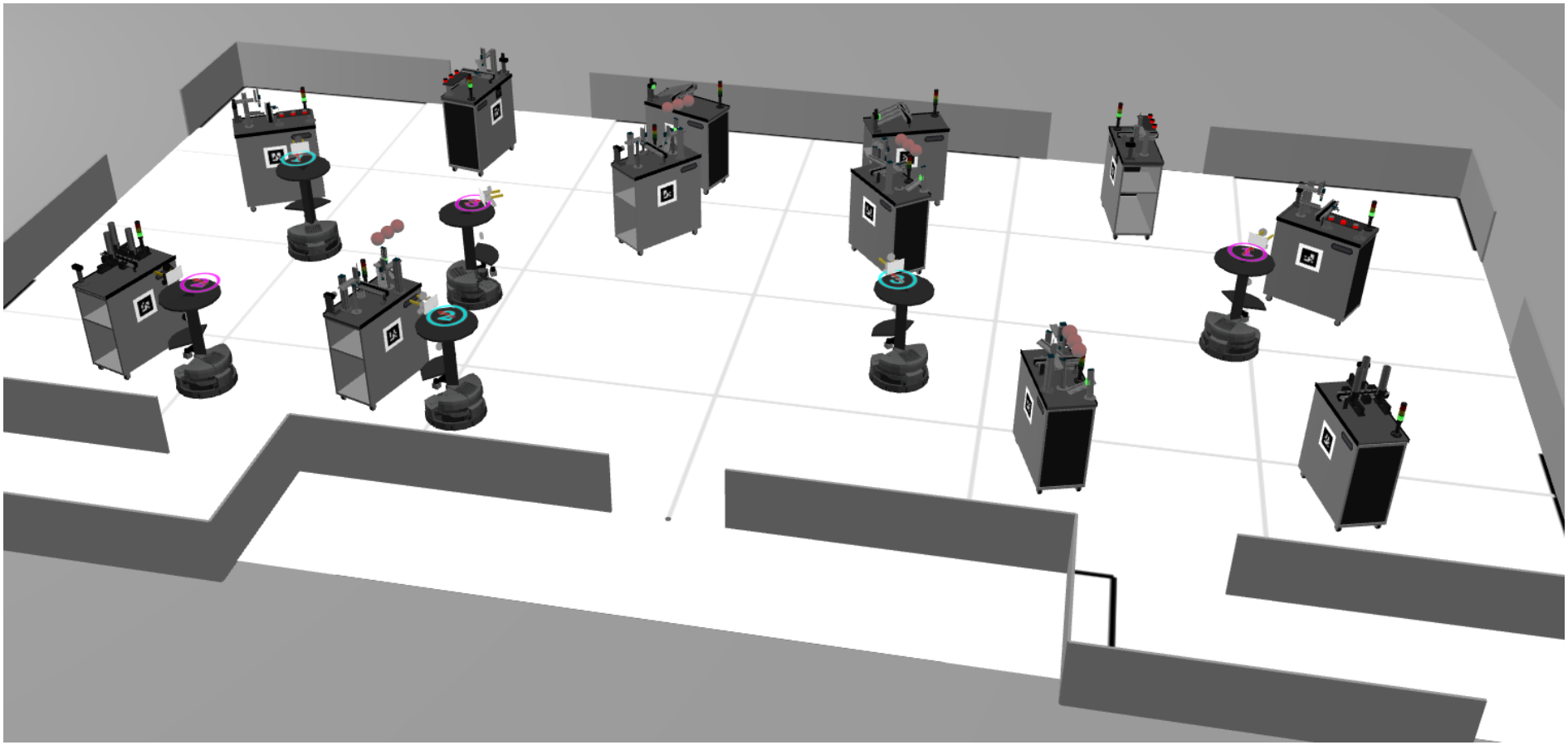
Before we go into the details. . .

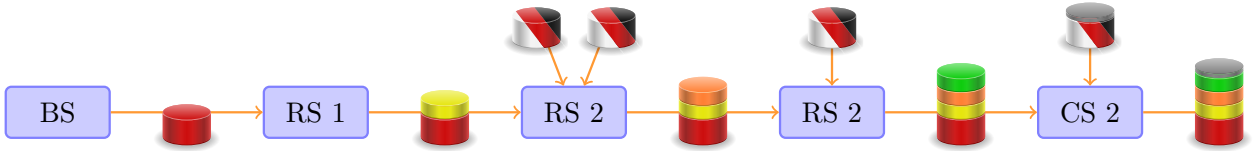
a little teaser

Planning in the era of *Smart Factories*



Planning & Execution Competition for Logistics Robots in Simulation





Planning & Execution Competition for Logistics Robots in Simulation

Planning and excution for the RCLL. What's hard?

- time windows
- domain representation: over 250 configurations possible!
- combinatorics
- scalability
- execution on robots/machines that can break
- performance guarantees
- ...

What is OMT?

Satisfiability Modulo Theories

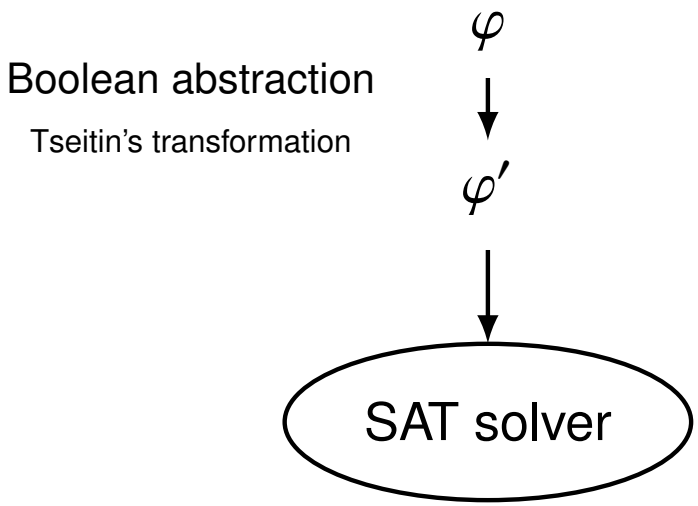
φ

Satisfiability Modulo Theories

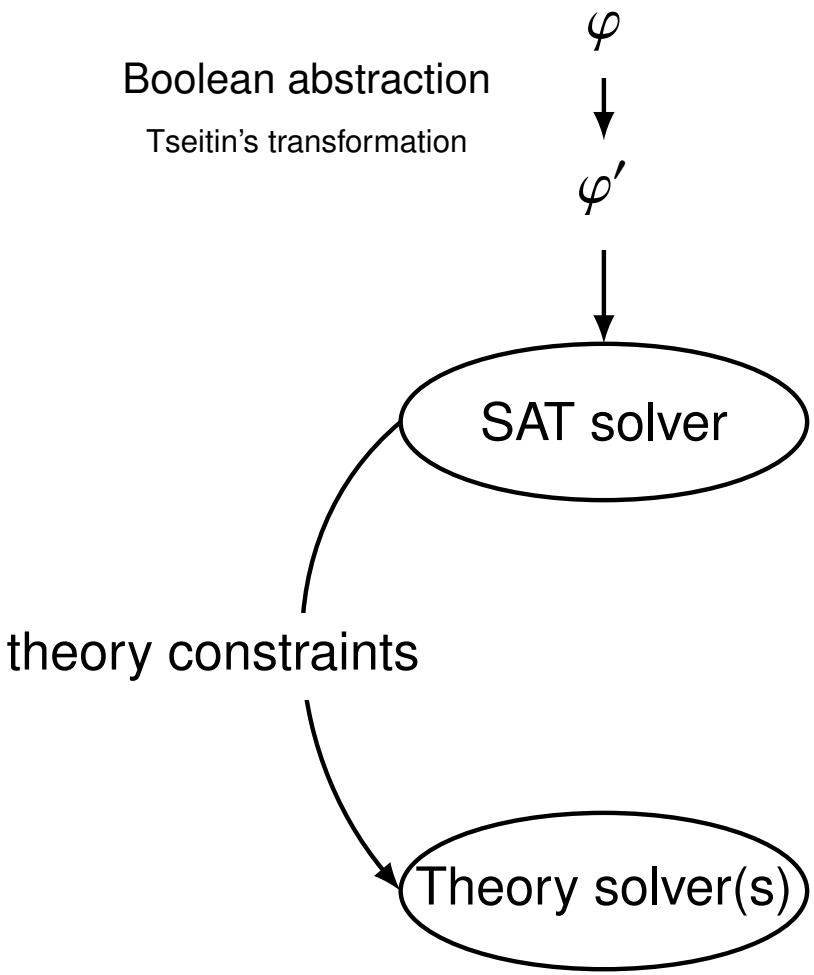
Boolean abstraction

Tseitin's transformation

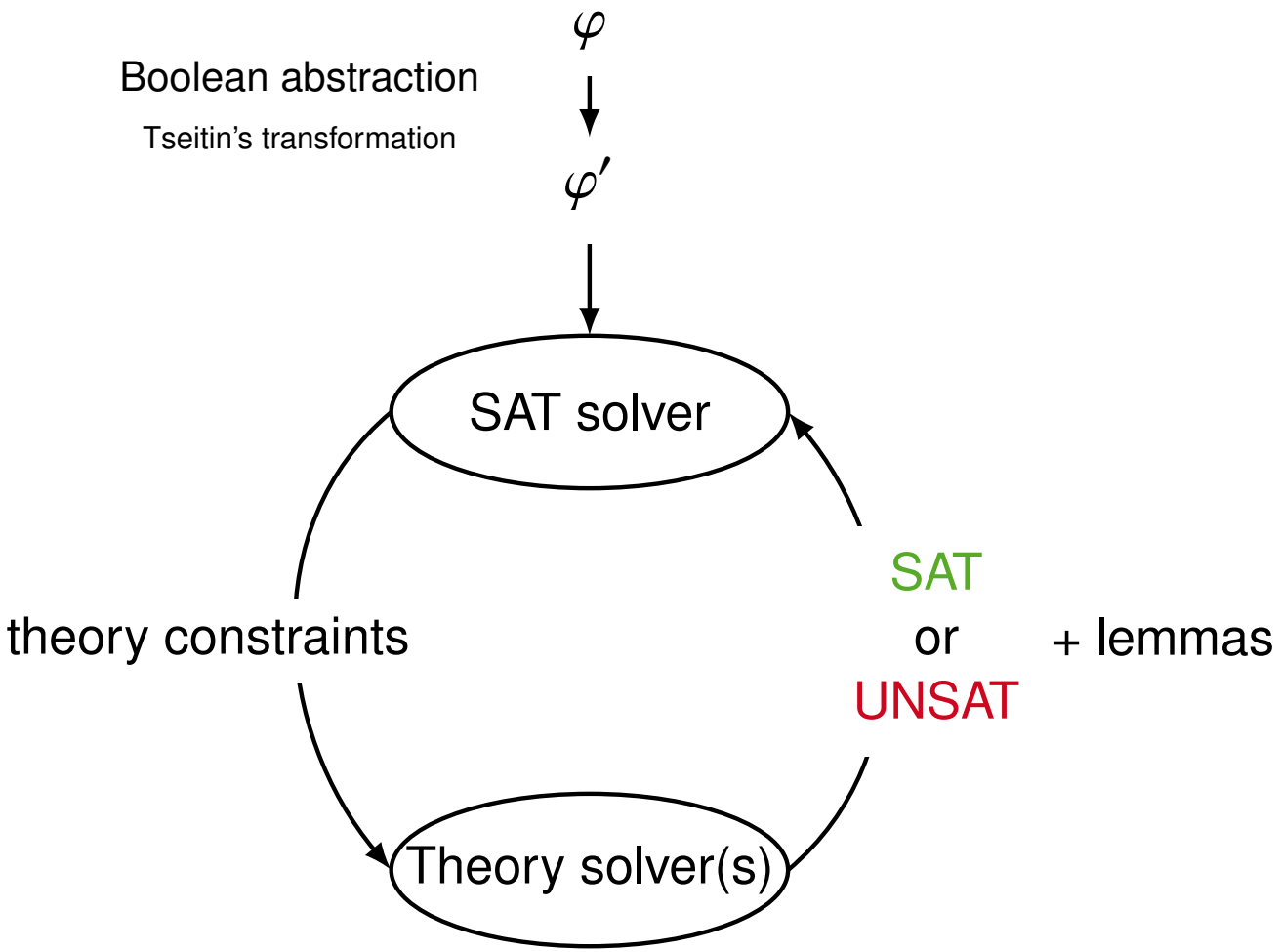
Satisfiability Modulo Theories



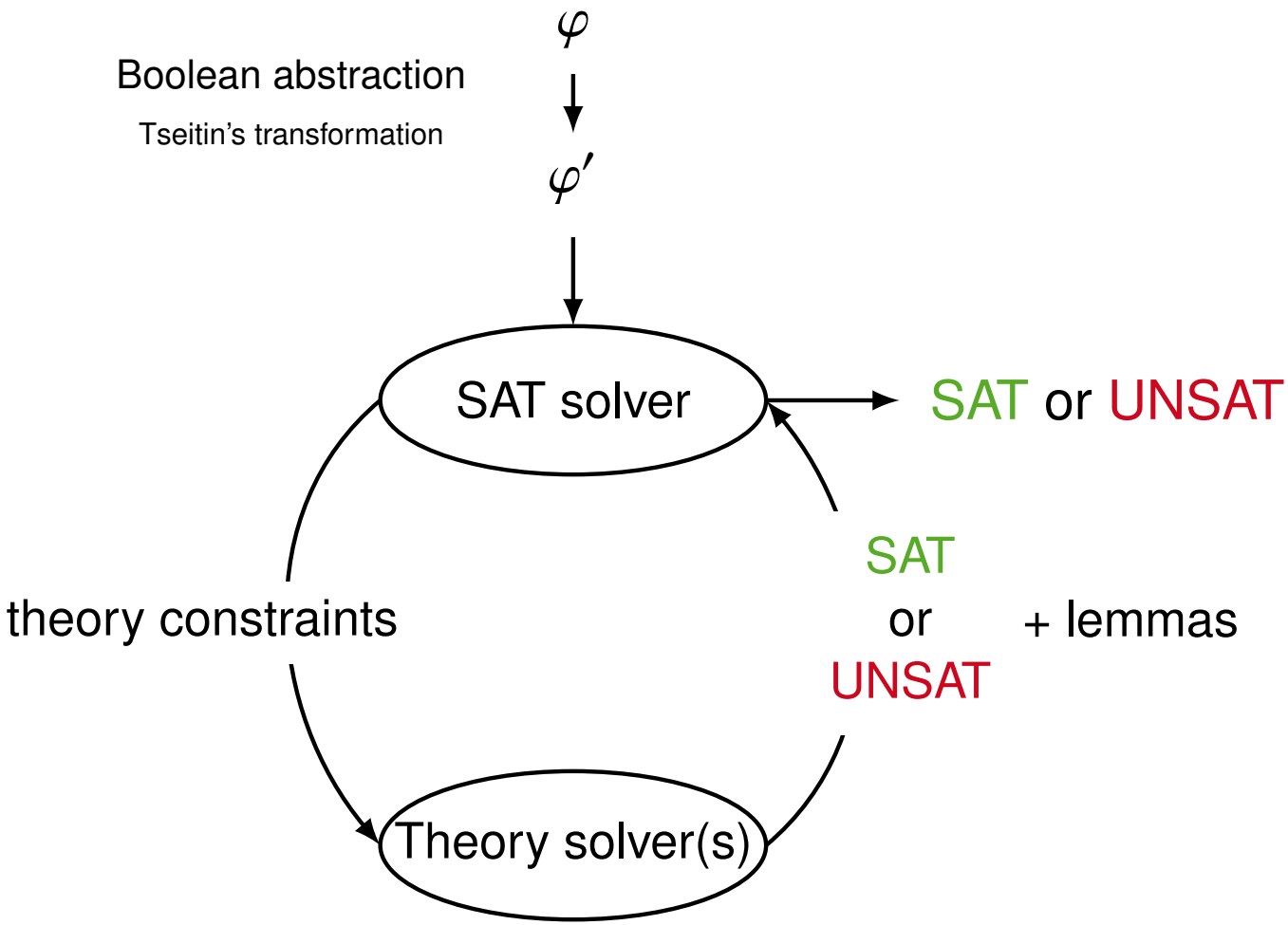
Satisfiability Modulo Theories



Satisfiability Modulo Theories



Satisfiability Modulo Theories



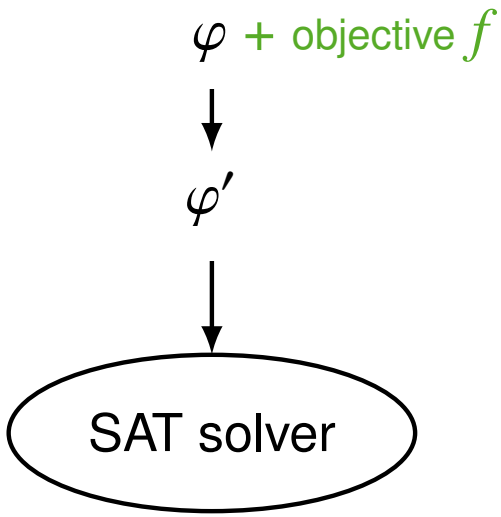
Optimization Modulo Theories

φ

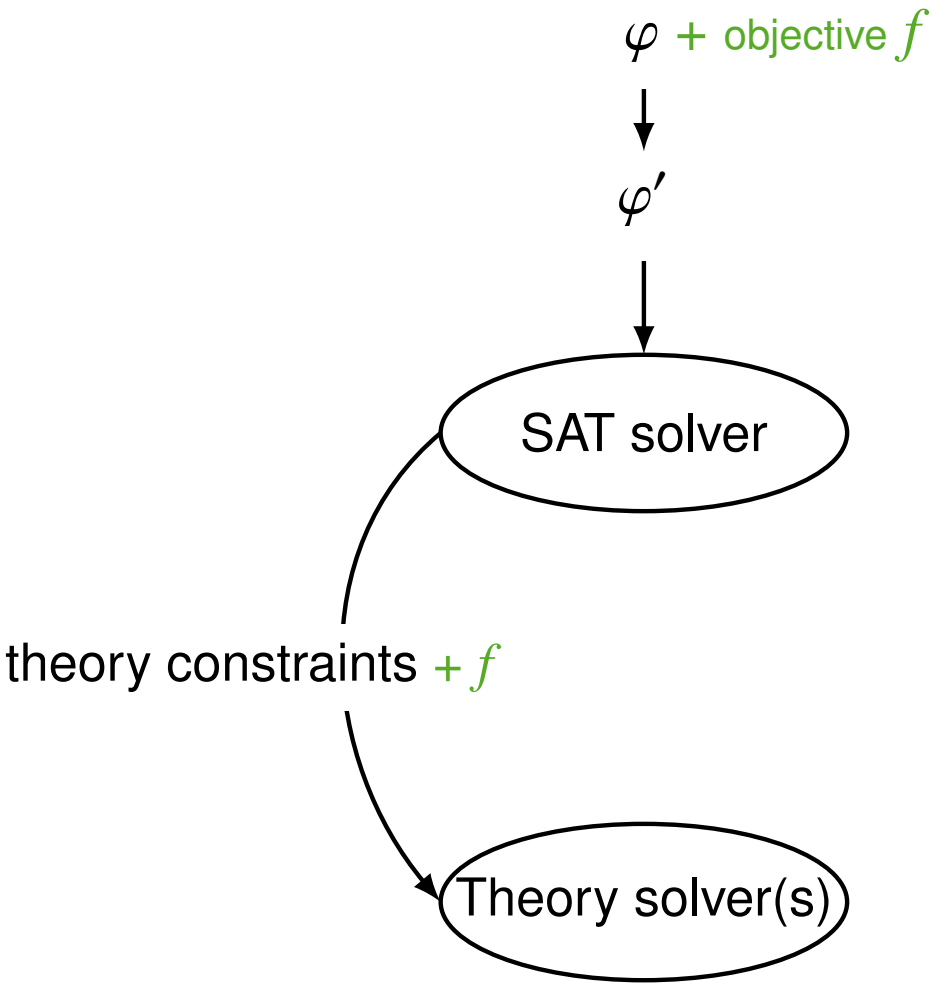
Optimization Modulo Theories

φ + objective f

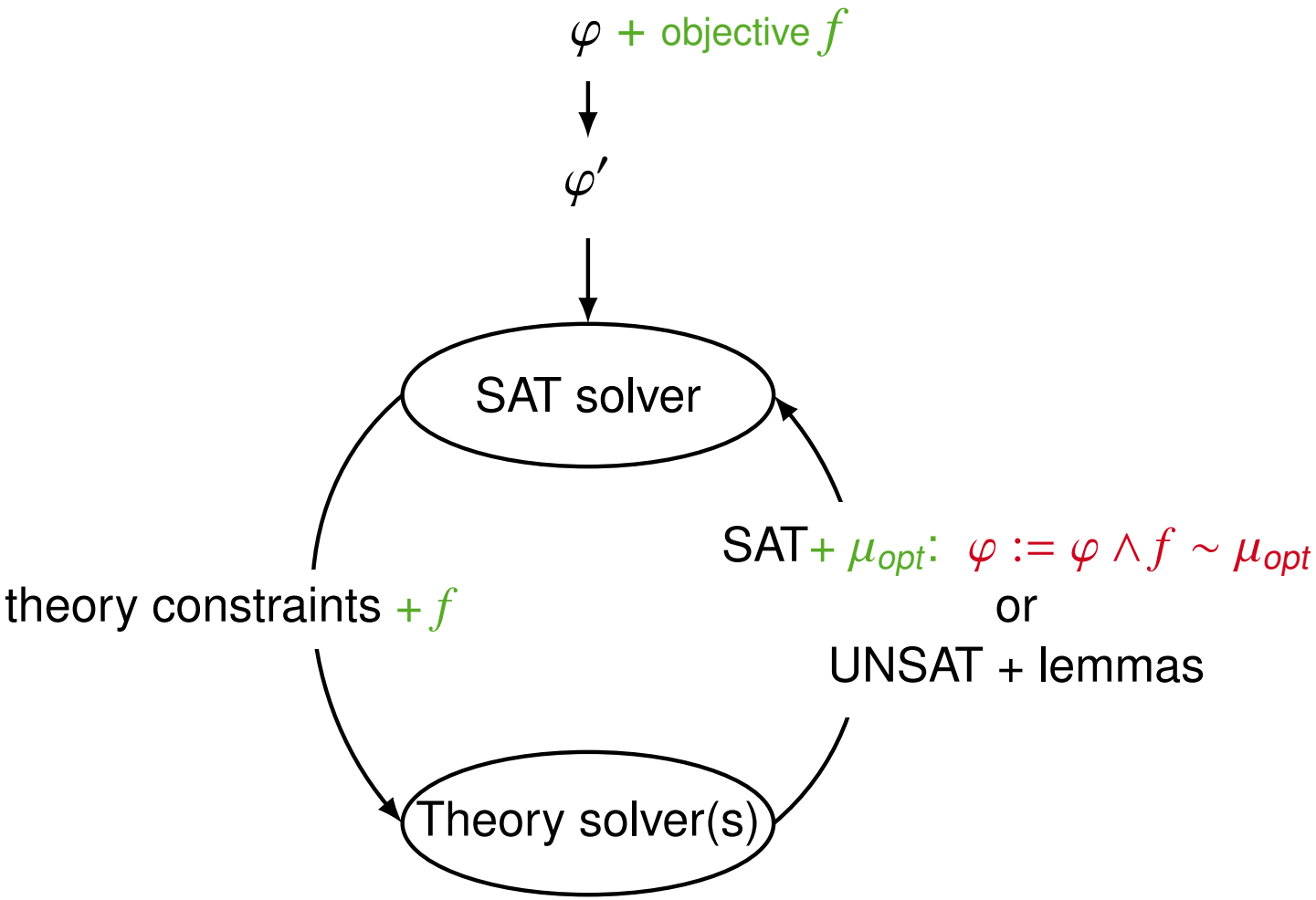
Optimization Modulo Theories



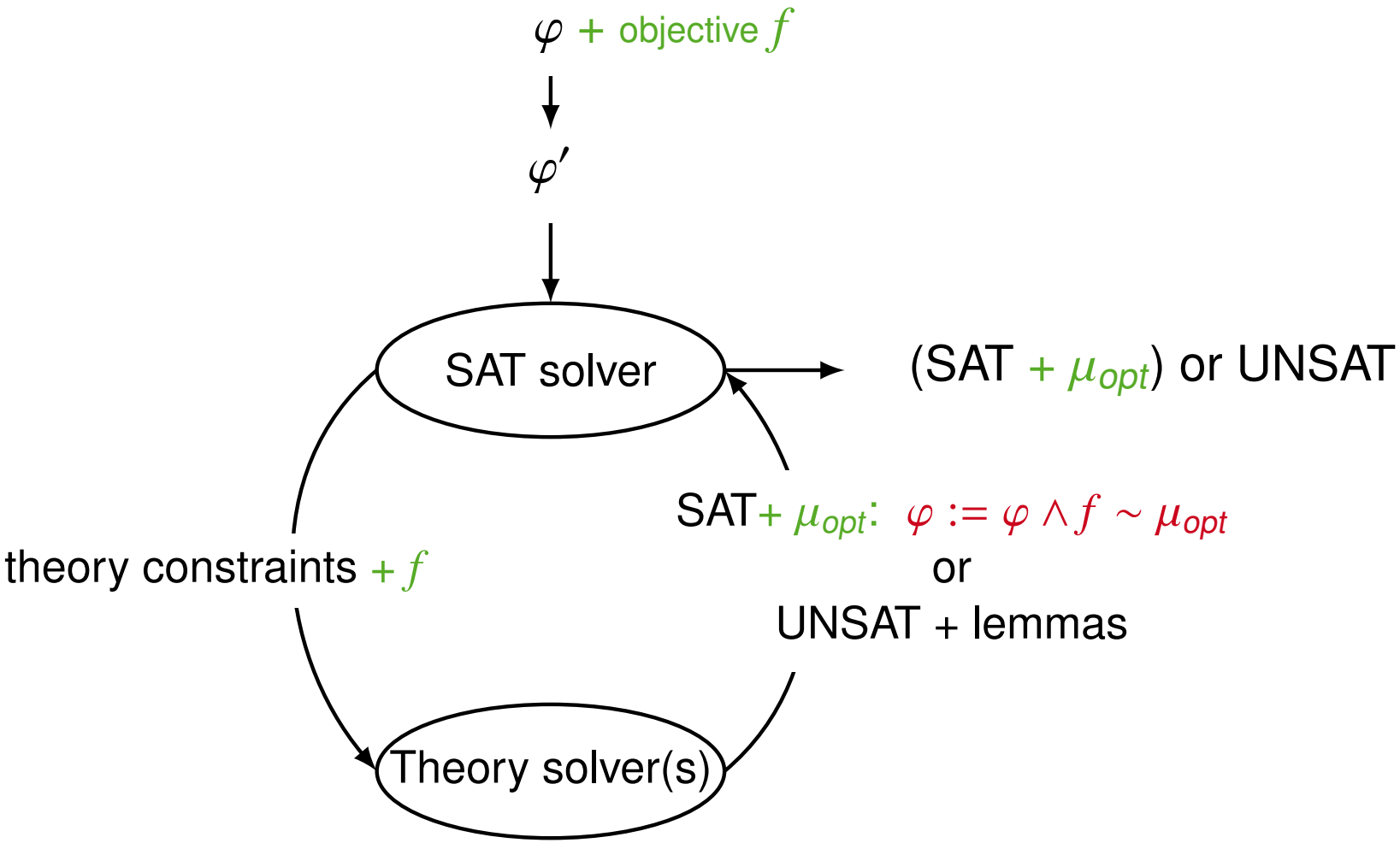
Optimization Modulo Theories



Optimization Modulo Theories

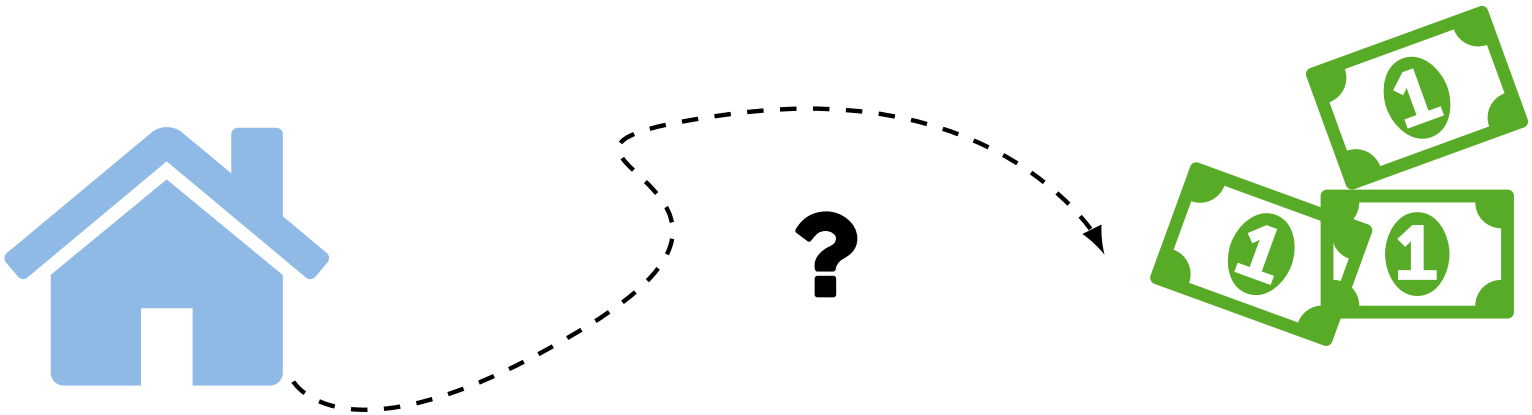


Optimization Modulo Theories

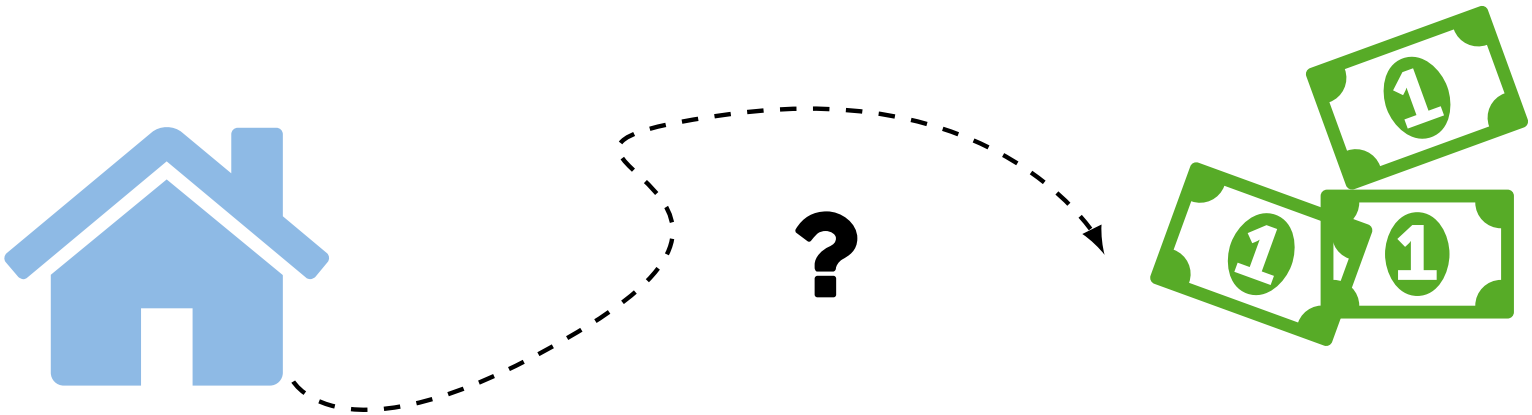


How can we use OMT for planning?

Planning as Satisfiability

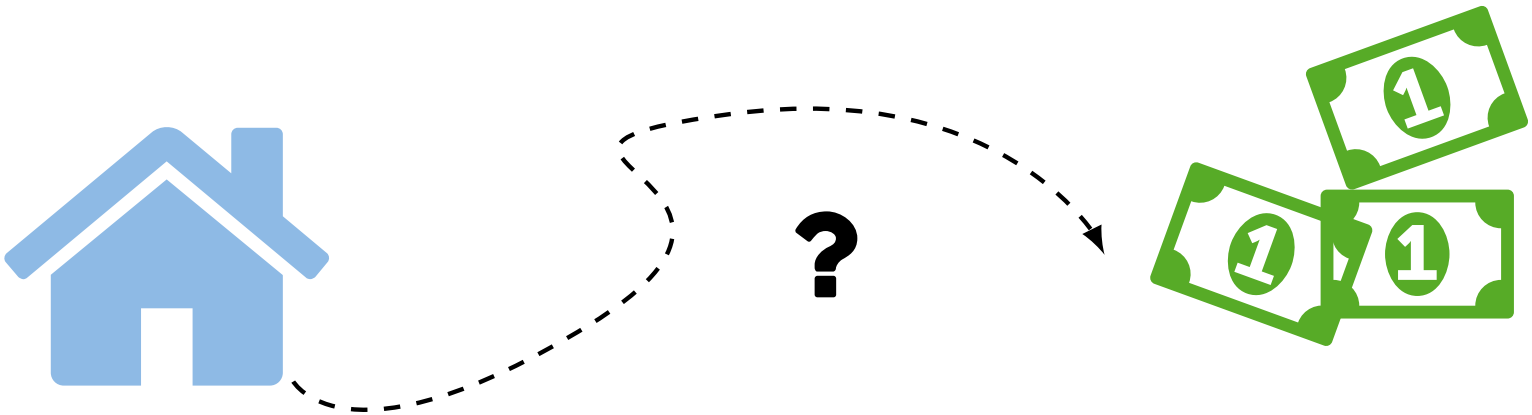


Planning as Satisfiability



- restrict search for a plan to paths with (predetermined) bound

Planning as Satisfiability



restrict search for a plan to paths with (predetermined) bound

Reductions of planning to SAT

- linear encodings[Kautz and Selman, 1992]
- later extended, *e.g.*, concurrency, theories...

Planning as Satisfiability

Planning problem

Let $\mathcal{F}$ and $\mathcal{A}$ be the sets of *fluents* and *actions*.

Let $\mathcal{X} = \mathcal{F} \cup \mathcal{A}$ and $\mathcal{X}' = \{x' : x \in \mathcal{X}\}$ be its *next state* copy.

A planning problem is a triple of boolean formulae $\Pi = \langle I, T, G \rangle$ where

- $I(\mathcal{F})$ represents the set of *initial* states
- $T(\mathcal{X}, \mathcal{X}')$ describes how actions *affect* states
- $G(\mathcal{F})$ represents the set of *goal* states

Planning as Satisfiability

Planning problem

Let $\mathcal{F}$ and $\mathcal{A}$ be the sets of *fluents* and *actions*.

Let $\mathcal{X} = \mathcal{F} \cup \mathcal{A}$ and $\mathcal{X}' = \{x' : x \in \mathcal{X}\}$ be its *next state* copy.

A planning problem is a triple of boolean formulae $\Pi = \langle I, T, G \rangle$ where

- $I(\mathcal{F})$ represents the set of *initial* states
- $T(\mathcal{X}, \mathcal{X}')$ describes how actions *affect* states
- $G(\mathcal{F})$ represents the set of *goal* states

Encoding Π in SAT - the formula

The planning problem Π with makespan k is the formula

$$\varphi_{(\Pi,k)} := I(\mathcal{X}_0) \wedge \bigwedge_{i=0}^{k-1} T(\mathcal{X}_i, \mathcal{X}_{i+1}) \wedge G(\mathcal{X}_k)$$

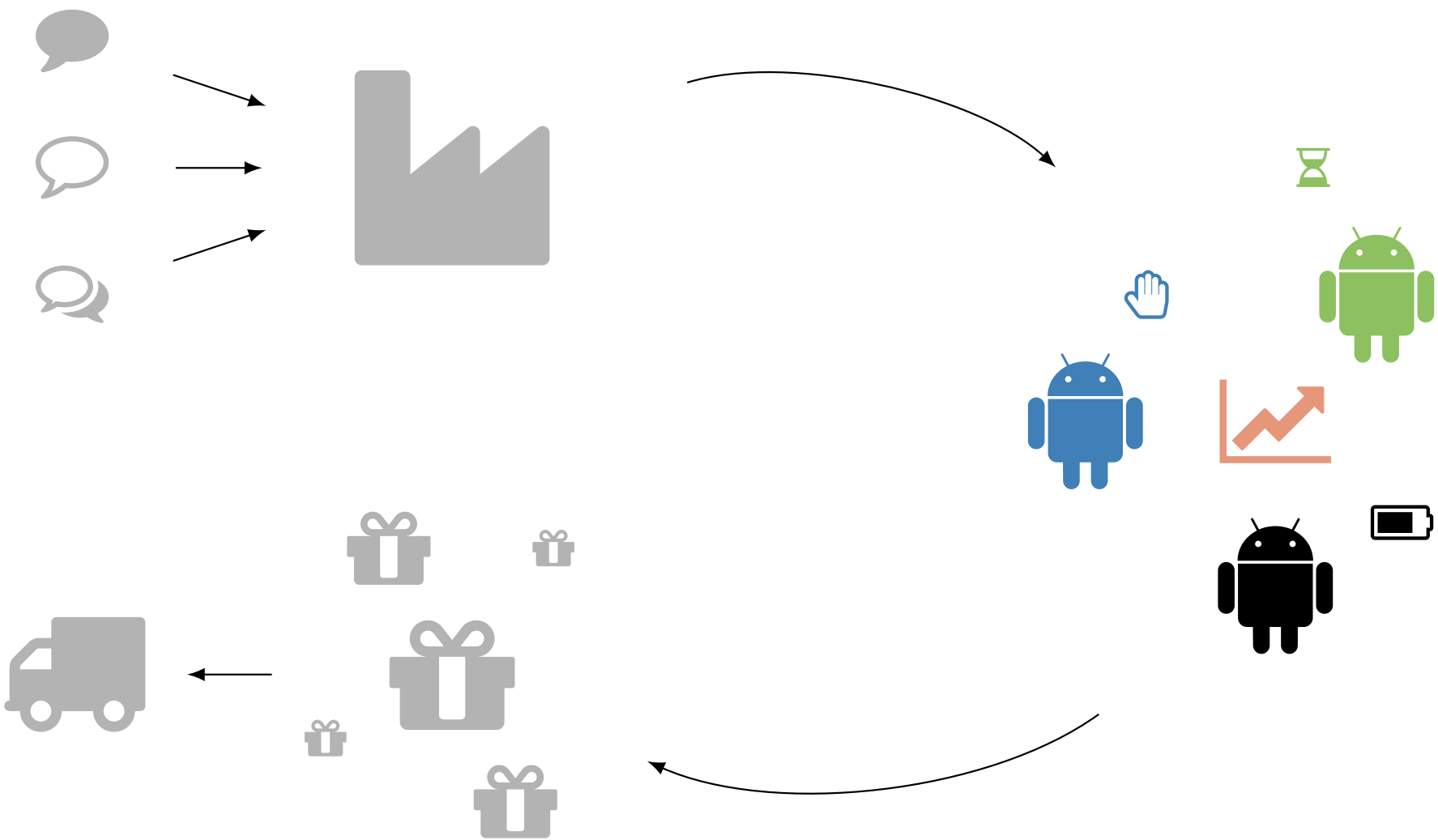
Planning as Satisfiability

Encoding Π in SAT

- $\varphi_{(\Pi,k)}$ is sat iff there exists a plan with length k
  - ✔ in that case, a plan can be extracted from the satisfying assignment
- in parallel encodings, two actions can be executed in parallel if they are non-mutex
- optimal plans minimize the number of steps:
  - ▶ start with $k = 1$
  - 🔁 increase until $\varphi_{(\Pi,k)}$ becomes sat or upper bound on k is reached.

Can we use OMT for planning in logistics?

Planning in the era of *Smart Factories*



Planning in the era of *Smart Factories*

Planning in production logistics is tough!

Many factors must be considered while planning

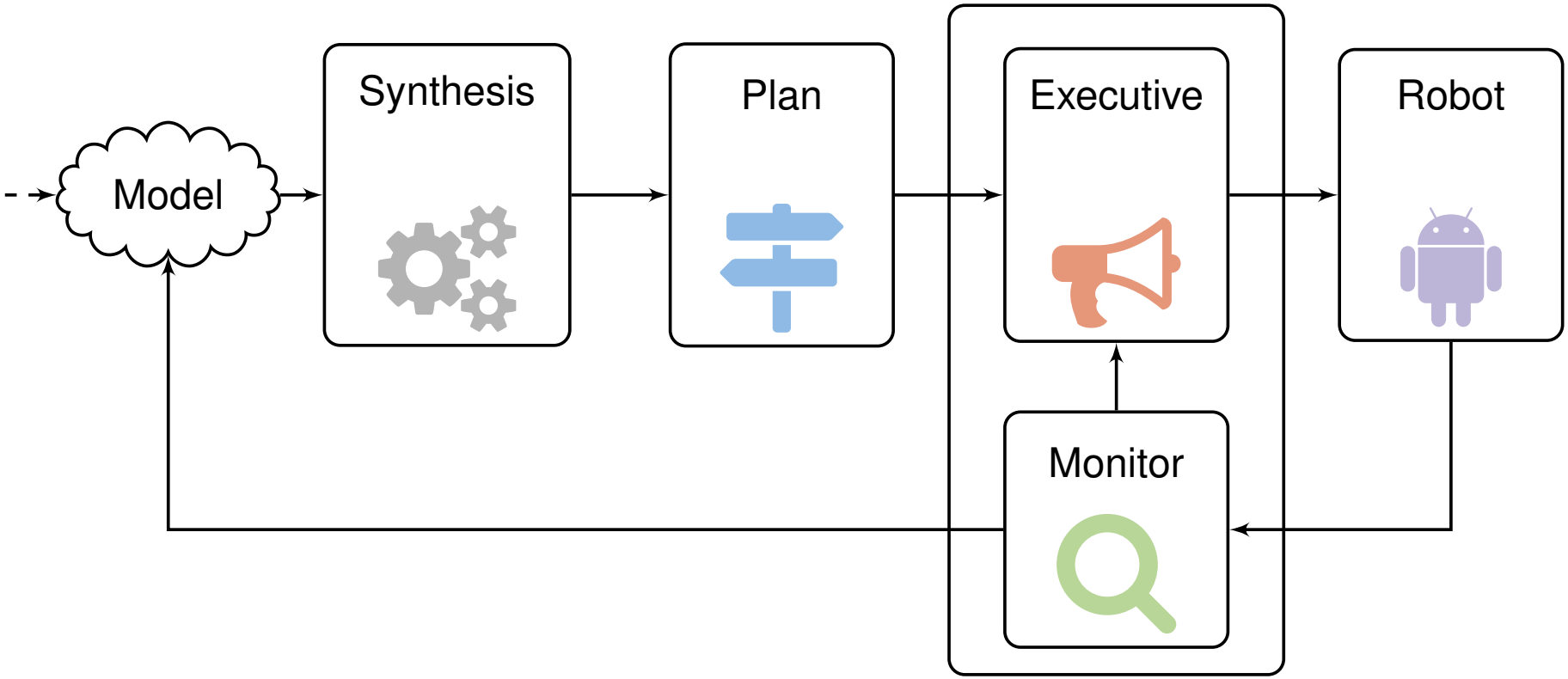
- state space of possible configurations
- efficiency of production processes
- costs/rewards
- ...

Trade-off: fast heuristic vs optimal solutions

Objective


Study the feasibility of OMT planning in a non-trivial logistic domain

Integrated synthesis and execution



1st place in the Planning and Execution Competition for Logistics Robots

PExC @ ICAPS 2018

Performance: a quick look

	C0 – R1		C0 – R2		C0 – R3	
Planner	Solved	Runtime (s)	Solved	Runtime (s)	Solved	Runtime (s)
RCLLPlan -Sat	**20**	0.98	**20**	2.52	**20**	2.73
RCLLPlan -Opt	**20**	15.22	17	42.18	4	54.26
POPF	**20**	20.73	4	25.66	0	-
Optic	13	24.39	2	57.60	0	–
SMTPlan+	0	–	0	–	0	–

Checkpoint

Achieved so far

- implemented a domain-specific planner for the RCLL
- planner is based on satisfiability checking and optimization
- integrated OMT planning with online execution and monitoring

Checkpoint

Achieved so far

- implemented a domain-specific planner for the RCLL
- planner is based on satisfiability checking and optimization
- integrated OMT planning with online execution and monitoring

Are we done?

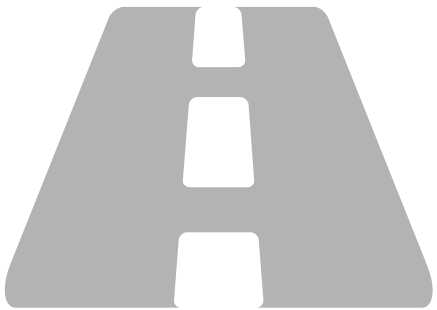
Checkpoint

Achieved so far

- implemented a domain-specific planner for the RCLL
- planner is based on satisfiability checking and optimization
- integrated OMT planning with online execution and monitoring

Are we done?

There's still a lot to do to ease applicability of OMT technology in planning!



Open challenges

Ongoing

Need a general approach for planning with OMT

- currently developing a domain-independent OMT planner
- use it as a platform to evaluate state of the art in SMT/OMT

Open challenges

Ongoing

Need a general approach for planning with OMT

- currently developing a domain-independent OMT planner
- use it as a platform to evaluate state of the art in SMT/OMT

This is the starting point, sevaral major challenges await:

- develop new reductions that are “solver friendly”
- exploit parallel solving, hybrid architectures
- how to explain decisions made by a solver?
- ... more in the paper ...

More ideas?