Approaches to platform-based system synthesis with answer set programming

based on works by
B. Andres, A. Biewer, M. Eberl, M. Gebser, J. Gladigau, M. Glaß, C. Haubelt, B. Kaufmann, R. Otero, F. Reimann, J. Romero, T. Schaub, J. Teich, P. Wanko

Davide Liessi

Department of Mathematics, Computer Science and Physics
University of Udine, Italy

Udine, 14 December 2016

satisfiability (SAT)

$$P, Q, \ldots \quad \bot \quad \neg, \wedge, \vee, \rightarrow, \equiv$$

$$\nu \colon \{\bot, P, Q, \ldots\} \rightarrow \{\mathbb{T}, \mathbb{F}\} \qquad \nu(\bot) = \mathbb{F}$$

F satisfiable if there exists ν such that $\nu(F) = \mathbb{T}$

satisfiability modulo theories (SMT)

$$x, y, \ldots \quad f, g, \ldots \quad p, q, \ldots \quad = \quad \bot \quad \neg, \wedge, \vee, \rightarrow, \equiv, \forall, \exists$$

interpretation I: D^I, f^I, p^I (p of arity $\geqslant 1$), $=^I$ $\quad (\rightsquigarrow t^I)$

state s: $\begin{cases} s(x) \in D^I \\ s(p) \in \{\mathbb{T}, \mathbb{F}\} \quad (p \text{ of arity } 0) \end{cases}$ $\quad (\rightsquigarrow s(t))$

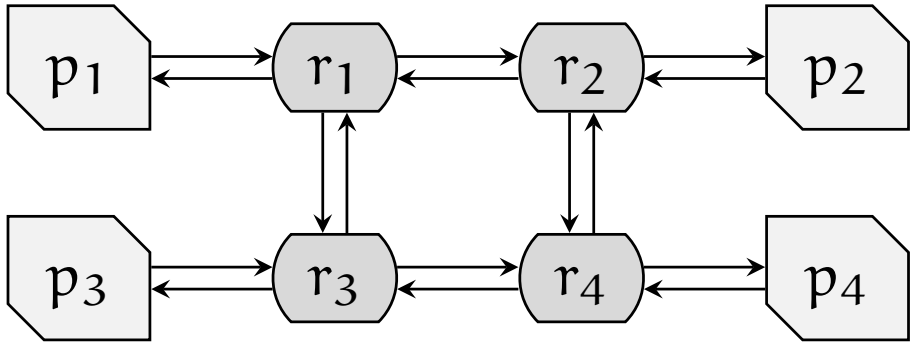
$$\nu_{I,s} : \{\text{formulas}\} \rightarrow \{\mathbb{T}, \mathbb{F}\} \qquad \nu_{I,s}(\bot) = \mathbb{F}$$

F satisfiable if there exist I and s such that $\nu_{I,s}(F) = \mathbb{T}$

F satisfiable modulo T if $T \cup \{F\}$ satisfiable

platform-based symbolic system synthesis

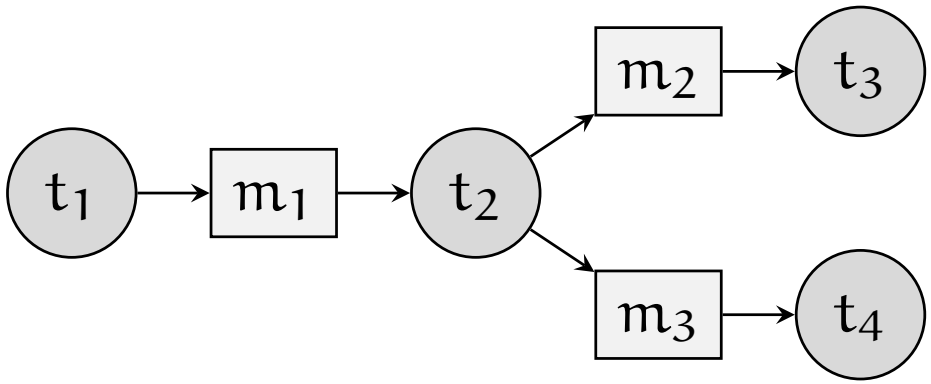
platform $g_P = (R, E_R)$



applications

$\mathbb{A} \ni A_i = (g_{A_i}, P_i, D_i)$

$g_{A_i} = (T_i \cup M_i, E_{A_i})$



implementation $I = (B, (R_m)_m, S)$

B binding: total function from tasks to processors

R_m routing for each message m

S schedule: start times for tasks and message transfers

SMT-based system synthesis (3S)

$$\text{functional constraints} \rightsquigarrow \Psi_F \rightsquigarrow \text{SAT}$$

$$\text{nonfunctional constraints} \rightsquigarrow \Psi_N \rightsquigarrow \text{SMT (QF\_IDL)}$$

$$\Psi_f = \Psi_F \wedge \Psi_N = \Psi_F \wedge \neg\Psi_{\overline{N}}$$

3S: simple learning

$$\Psi_N^0 = \mathbb{F}, \quad \Psi_f^0 = \Psi_F \wedge \neg\Psi_N^0 = \Psi_F$$

i-th step:

1. find implementation I satisfying Ψ_f^i (SAT solver)
2. check nonfunctional constraints (theory solver)
3. if feasible $\rightsquigarrow$ problem solved

   else $\rightsquigarrow$
   $$\begin{cases} \Psi_N^{i+1} = \Psi_N^i \vee \Theta \\ \Psi_f^{i+1} = \Psi_F \wedge \neg\Psi_N^{i+1} \end{cases}$$
   Θ encoding of I

3S: early learning

$$\widetilde{\Theta} \sqsubseteq \Theta \quad \text{if} \quad \widetilde{\Theta} \vee \Theta \equiv \widetilde{\Theta}$$

$\widetilde{I}$ partial implementation of I if $\widetilde{\Theta} \sqsubseteq \Theta$ (Θ, $\widetilde{\Theta}$ encodings)

$g\colon \{\text{implementations}\} \to \{\mathbb{F}, \mathbb{T}\}$ theory evaluation

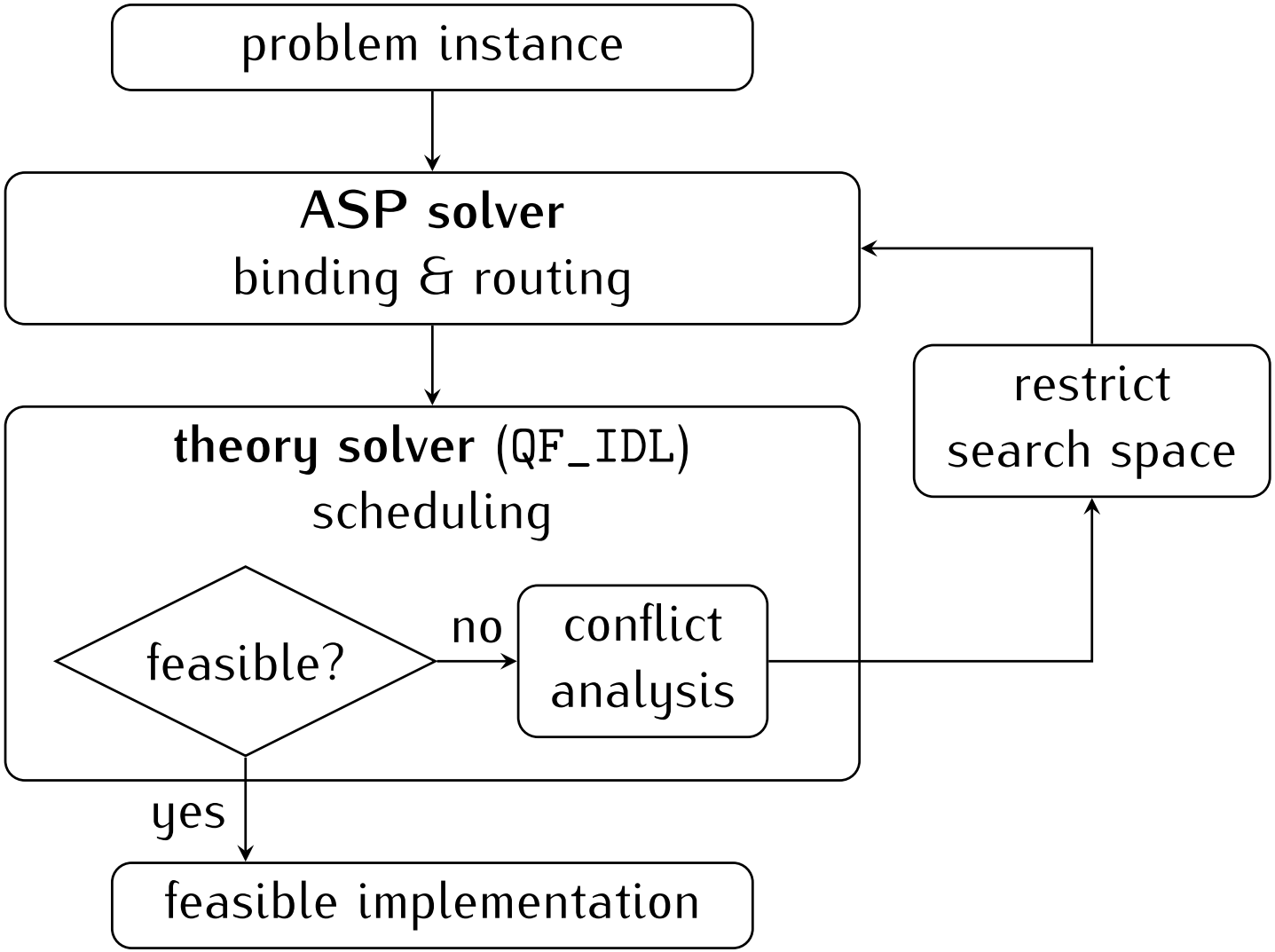
$$\Psi_N^{i+1} = \Psi_N^i \vee (\neg g(\widetilde{I}) \wedge \widetilde{\Theta})$$

3S: SAT $\rightsquigarrow$ ASP

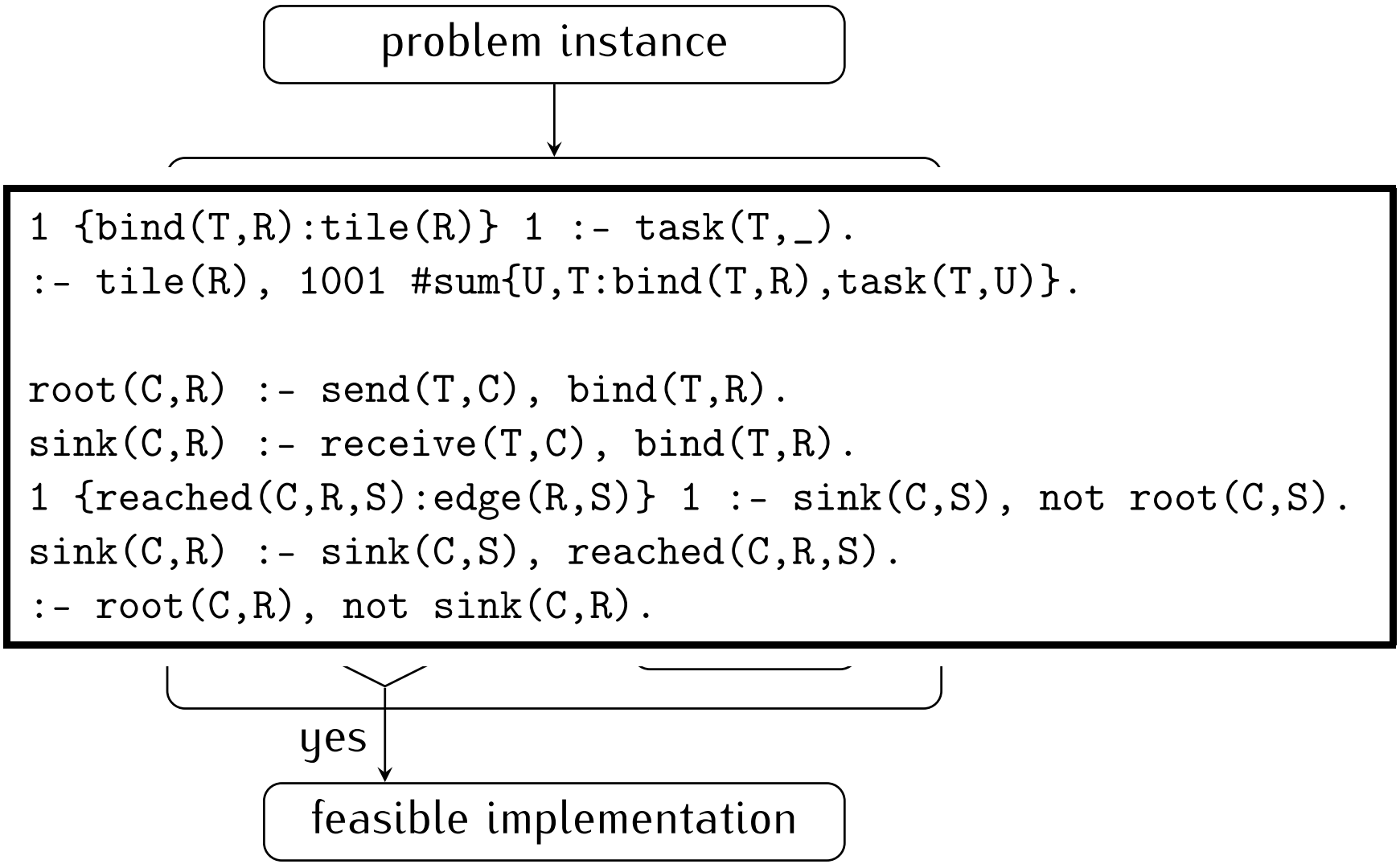
SAT encoding of "R_m is routing" scales with $|E_R| \cdot |R|$

inductive strategy scales with $|E_R| \rightsquigarrow$ ASP

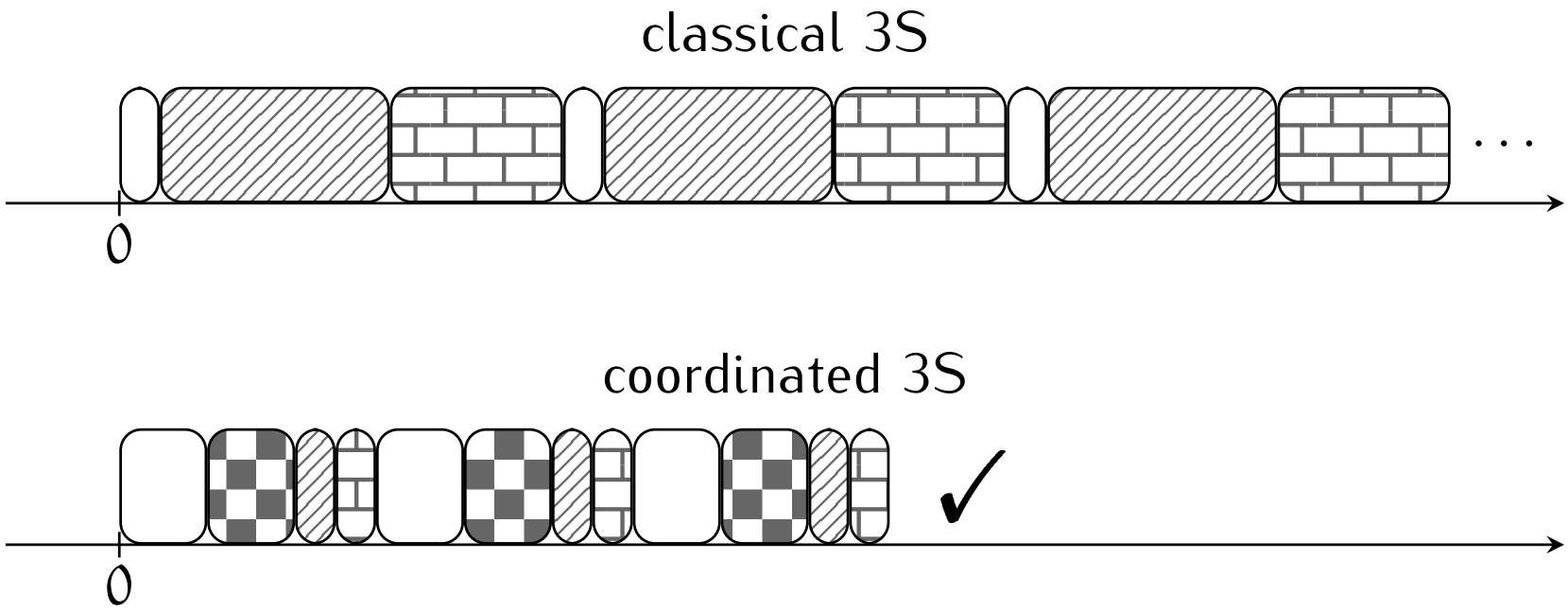
3S: overview


problem instance
ASP solver
binding & routing
theory solver (QF_IDL)
scheduling
feasible?
no
conflict
analysis
yes
feasible implementation
restrict
search space

3S: overview


problem instance
1 {bind(T,R):tile(R)} 1 :- task(T,_).
:- tile(R), 1001 #sum{U,T:bind(T,R),task(T,U)}.
root(C,R) :- send(T,C), bind(T,R).
sink(C,R) :- receive(T,C), bind(T,R).
1 {reached(C,R,S):edge(R,S)} 1 :- sink(C,S), not root(C,S).
sink(C,R) :- sink(C,S), reached(C,R,S).
:- root(C,R), not sink(C,R).
yes
feasible implementation

coordinated 3S



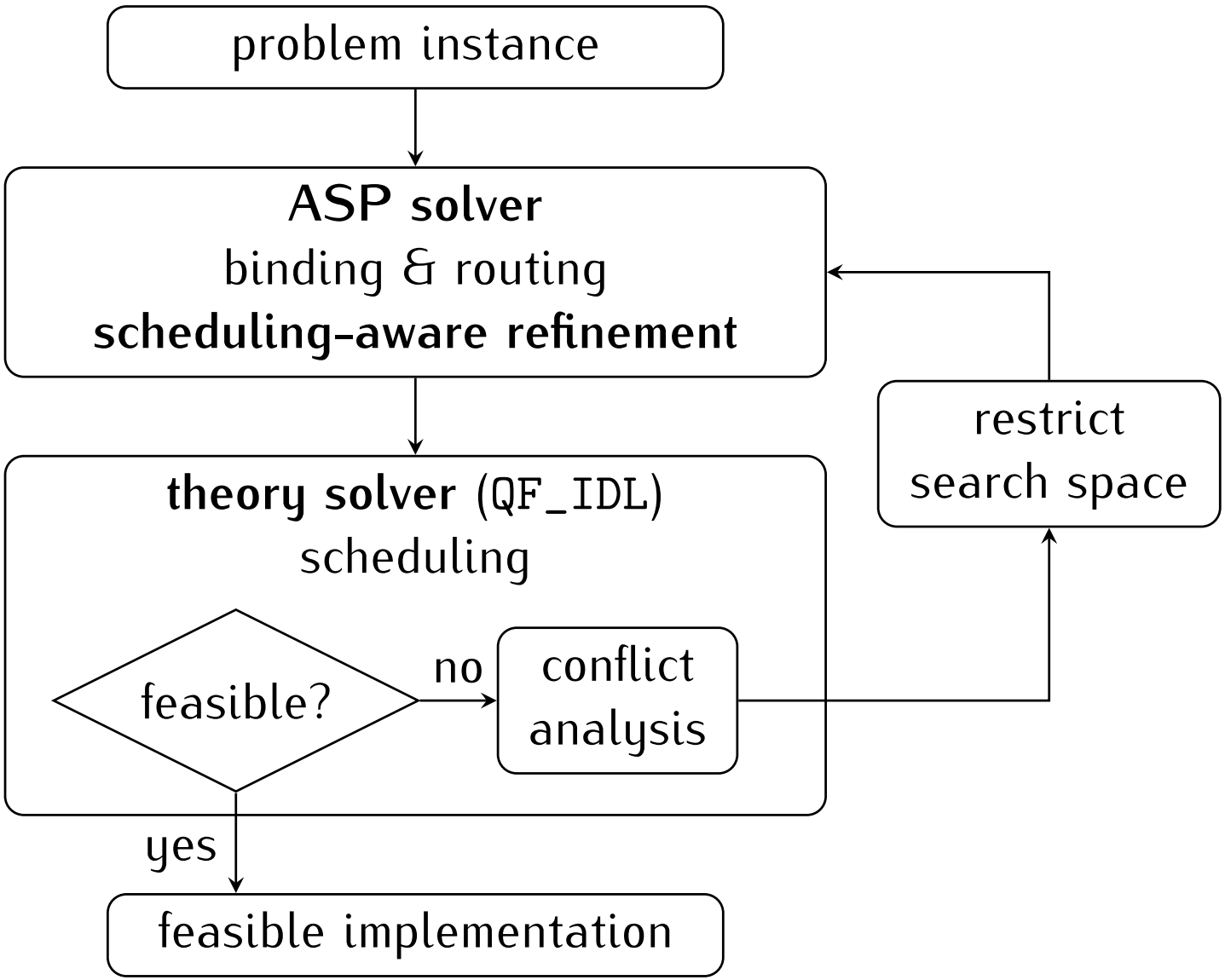
binding & routing

refinement

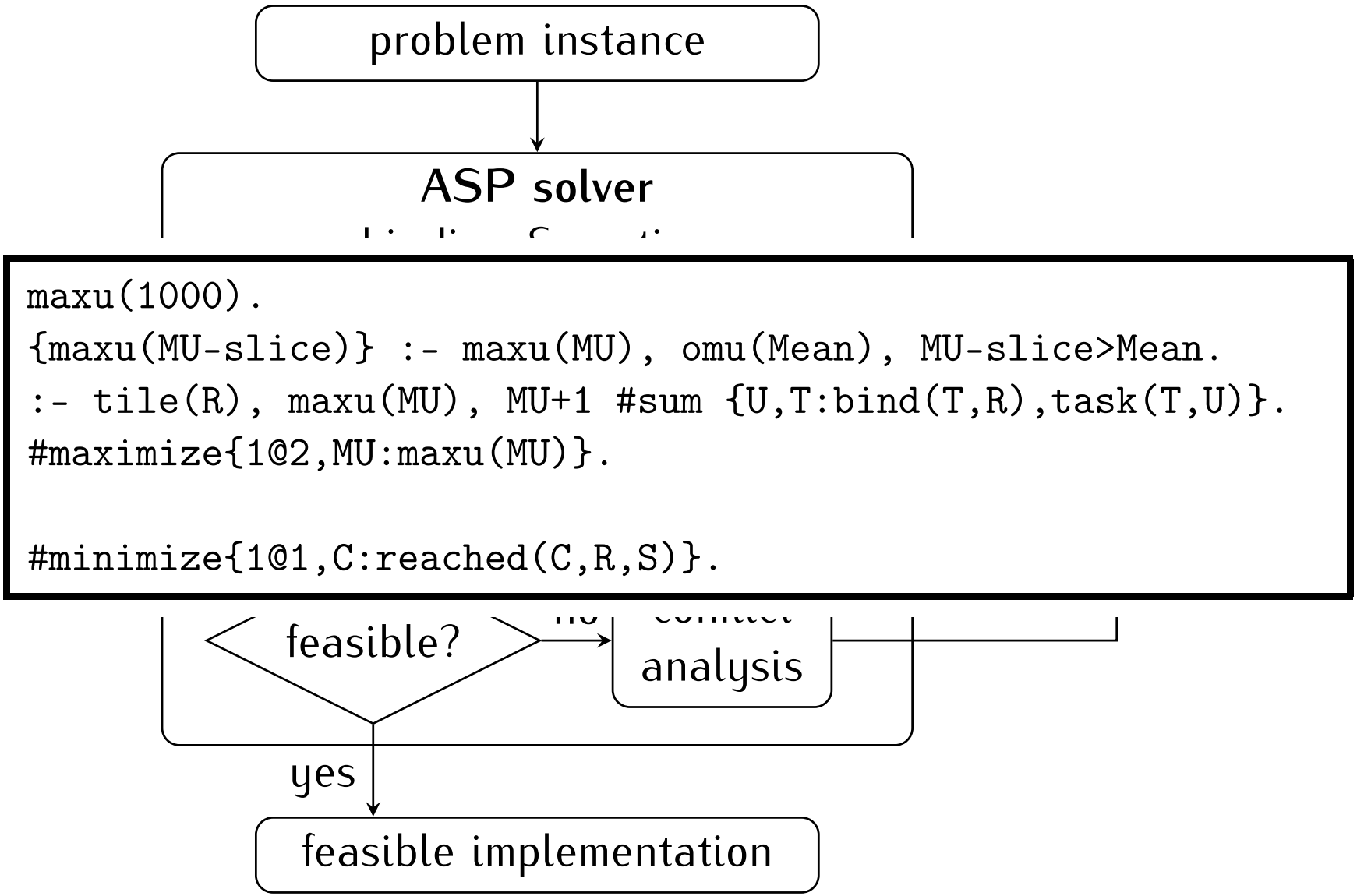
scheduling

conflict analysis

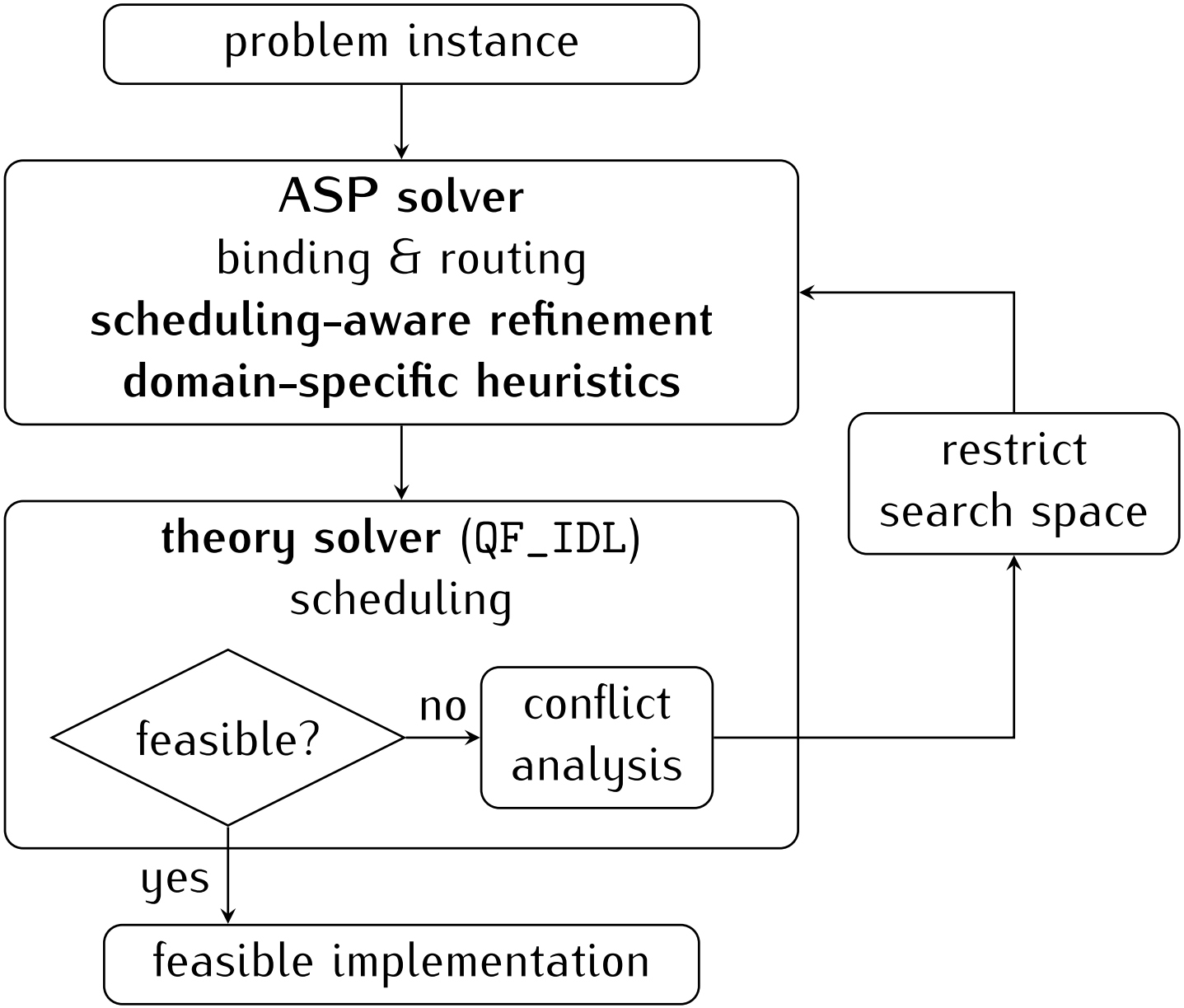
coordinated 3S: overview



coordinated 3S: overview



coordinated 3S with domain-specific heuristics



coordinated 3S with domain-specific heuristics

problem instance

H1: bindings first, then routing

```
_h(bind(T,R),level,2) :- task(T,_), tile(R).
_h(reached(C,R,S),level,1) :- edge(R,S), send(_,C).
```

H2: discourage routing

```
_h(reached(C,R,S),sign,-1) :- edge(R,S), send(_,C).
```

H3: bind sender and receiver together

```
_h(bind(T',R),sign,1) :- bind(T,R), send(T,C), receive(T',C).
```

H4: clustering computational tasks

```
_h(bind(T,R),true,A+2) :- bind(T',R), send(T',M), receive(T,M),
                          belongs(A,T).
_h(bind(T,R),true,A+1) :- bind(T',R'), send(T',M), receive(T,M),
                          neighbor(R',R), belongs(A,T).
_h(bind(T,R),level,A) :- belongs(A,T), task(T,_), tile(R).
```

feasible implementation

references

C. Barrett, R. Sebastiani, S. A. Seshia, C. Tinelli, *Satisfiability modulo theories*, in Handbook of satisfiability, A. Biere, M. Heule, H. Van Maaren, T. Walsh, eds., IOS Press, 2009, pp. 825–885.

M. Gebser, B. Kaufmann, T. Schaub, *Conflict-driven answer set solving: From theory to practice*, Artificial Intelligence, 187–188 (2012), pp. 52–89. DOI:10.1016/j.artint.2012.04.001

F. Reimann, M. Glass, C. Haubelt, M. Eberl, J. Teich, *Improving platform-based system synthesis by satisfiability modulo theories solving*, in 2010 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2010, pp. 135–144. DOI:10.1145/1878961.1878986

B. Andres, M. Gebser, T. Schaub, C. Haubelt, F. Reimann, M. Glass, *Symbolic system synthesis using answer set programming*, in Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning, Corunna, Spain, 2013, pp. 79–91. DOI:10.1007/978-3-642-40564-8

A. Biewer, B. Andres, J. Gladigau, T. Schaub, C. Haubelt, *A symbolic system synthesis approach for hard real-time systems based on coordinated SMT-solving*, in 2015 Design, Automation Test in Europe Conference Exhibition (DATE), 2015, pp. 357–362. DOI:10.7873/DATE.2015.0606

M. Gebser, B. Kaufmann, J. Romero, R. Otero, T. Schaub, P. Wanko, *Domain-specific heuristics in answer set programming*, in AAAI Conference on Artificial Intelligence, 2013.
http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6278

B. Andres, A. Biewer, J. Romero, C. Haubelt, T. Schaub, *Improving coordinated SMT-based system synthesis by utilizing domain-specific heuristics*, in Proceedings of the 13th International Conference on Logic Programming and Nonmonotonic Reasoning, 2015, pp. 55–68. DOI:10.1007/978-3-319-23264-5_6