



R.E.C.A.R



Recursive Explore and Check Abstraction Refinement

Jean-Marie Lagniez, Daniel Le Berre,
Tiago de Lima and Valentin Montmirail

CRIL, Université d'Artois and CNRS, France

{lagniez, leberre, delima, montmirail}@cril.fr



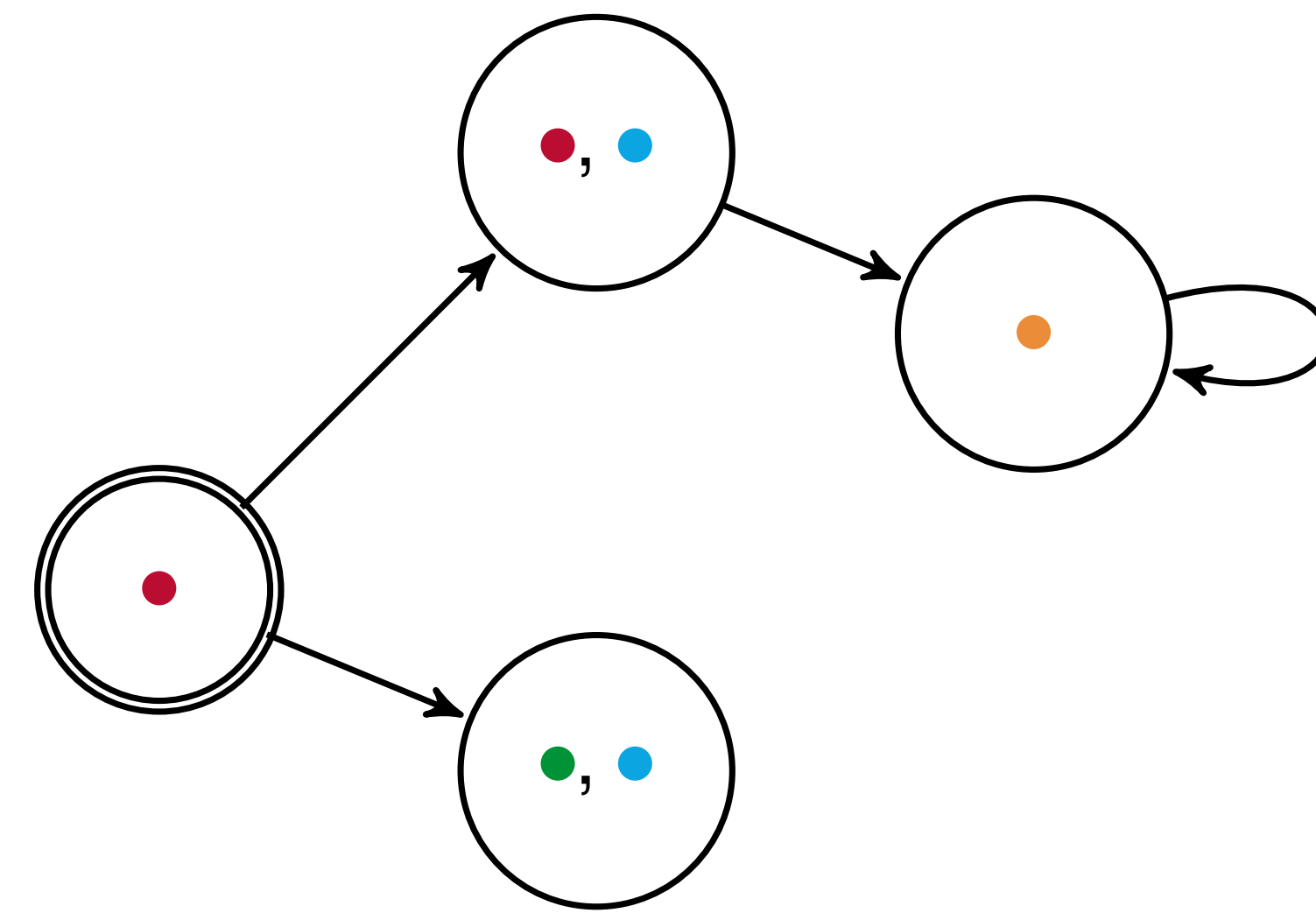
UNIVERSITÉ D'ARTOIS

MOTIVATION AND CONTRIBUTIONS

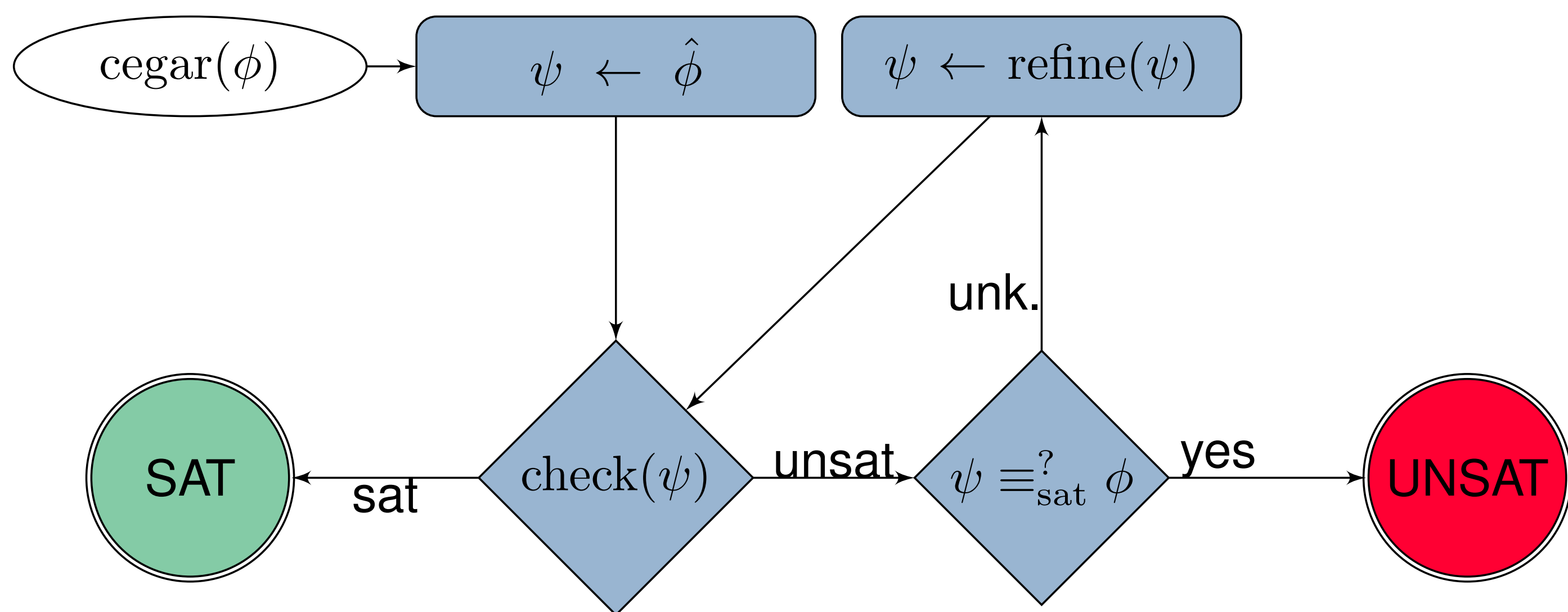
- Modal Logic K Satisfiability Problem is **PSPACE complete**.
- How to solve a PSPACE problem with a SAT solver? By using the latest features of SAT solvers: unsatisfiable cores and incremental SAT.
- Contribution: A CEGAR-like framework called RECAR and an instantiation for modal logic K within the solver MoSaiC

CONTEXT

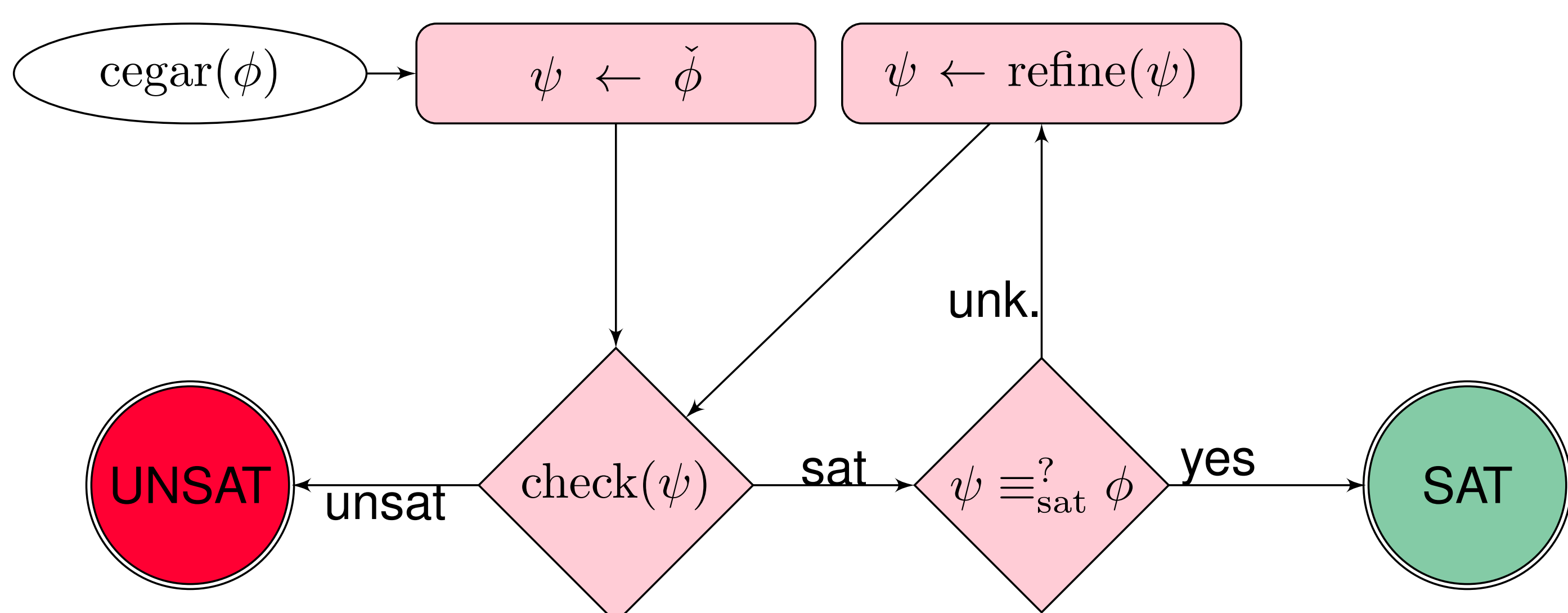
- A lot of AI problems are solved using propositional logic
- But representing a problem in this logic is a tedious task.
- Modal Logics are **more expressive**. They allow to reason about:
 - What is possible; What somebody thinks; ... (\Diamond)
 - What is necessary; What somebody knows; ... (\Box)
- They represent in a compact way those problems



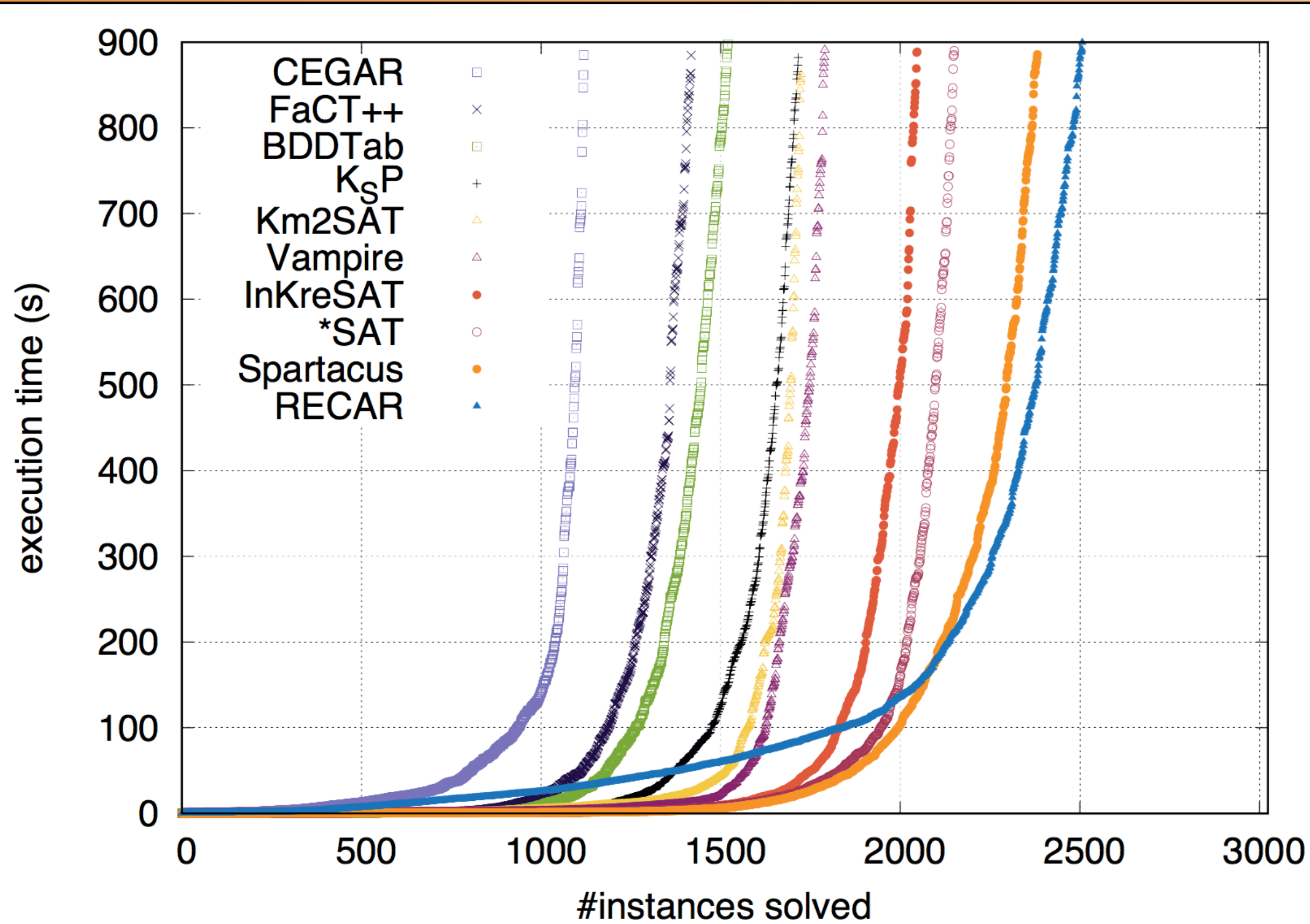
CEGAR WITH OVER-APPROXIMATIONS



CEGAR WITH UNDER-APPROXIMATIONS

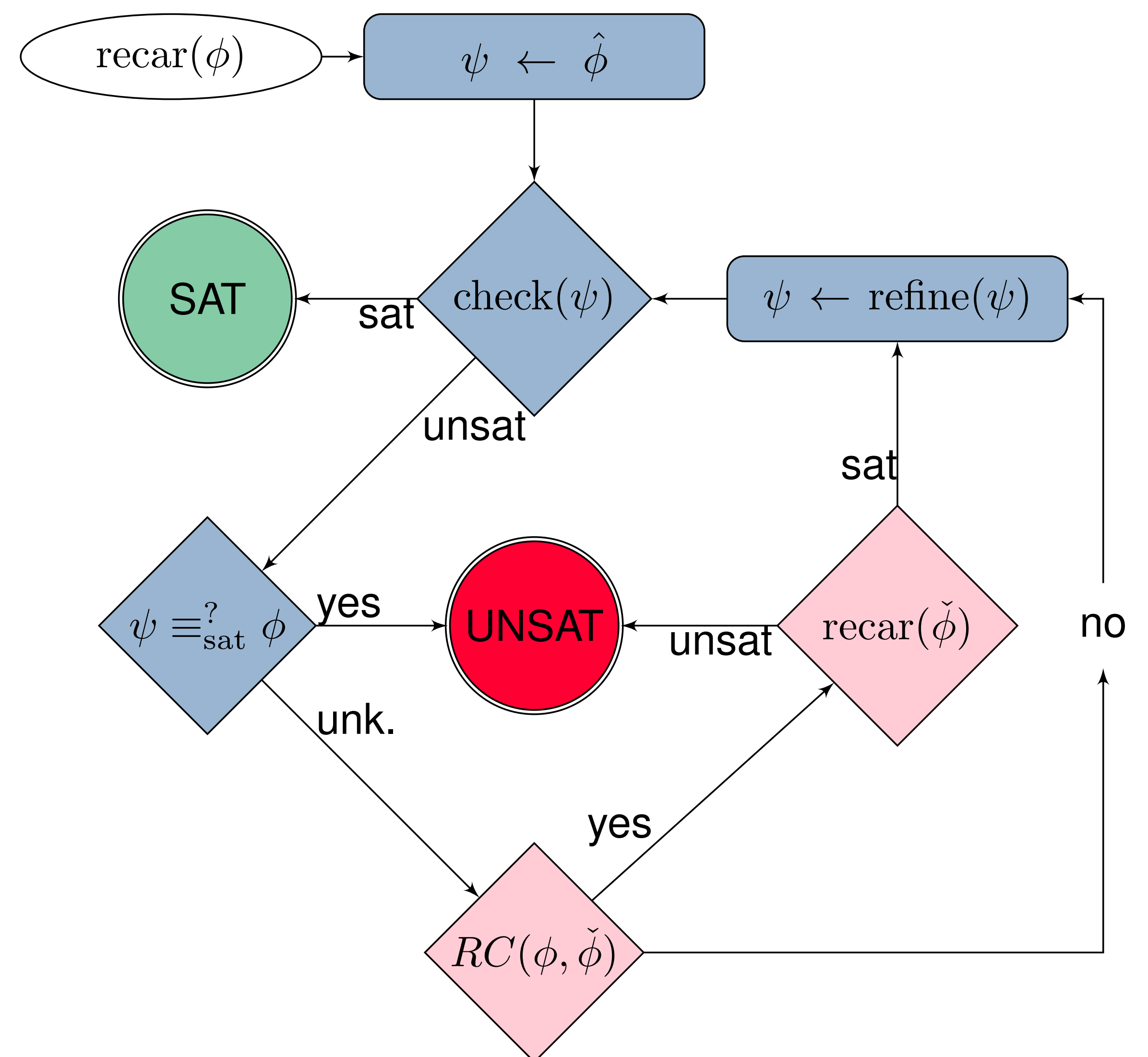


EXPERIMENTAL RESULTS: MoSaiC AGAINST SOTA SOLVERS



MoSaiC solves more benchmarks than the other solvers on the benchmarks considered.

RECAR: RECURSIVE EXPLORE AND CHECK ABSTRACTION REFINEMENT



EXPERIMENTAL SETTINGS

- 2 configurations of MoSaiC: CEGAR vs RECAR
- Many state-of-the-art solvers for modal logic K satisfiability problem
- CentOS 6.0, bi-proc. XEON, 4 cores, 3.3 GHz, 32GB, 900 seconds.

CONCLUSION AND FUTURE WORK

- RECAR is an efficient merge between CEGAR-over and CEGAR-under
- RECAR for other PSPACE problems
- Extend MoSaiC for others modal logics

