High-Performance Automated Theorem Proving

Bruno Woltzenlogel Paleo
Theory and Logic Group
Institut für Computersprachen
Vienna University of Technology

Inspiration from History (1965)

A Machine-Oriented Logic Based on the Resolution Principle

J. A. Robinson

Argonne National Laboratory and Rice University†*

Abstract. Theorem-proving on the computer, using procedures based on the fundamental theorem of Herbrand concerning the first-order predicate calculus, is examined with a view towards improving the efficiency and widening the range of practical applicability of these procedures. A close analysis of the process of substitution (of terms for variables), and the process of truth-functional analysis of the results of such substitutions, reveals that both processes can be combined into a single new process (called *resolution*), iterating which is vastly more efficient than the older cyclic procedures consisting of substitution stages alternating with truth-functional analysis stages.

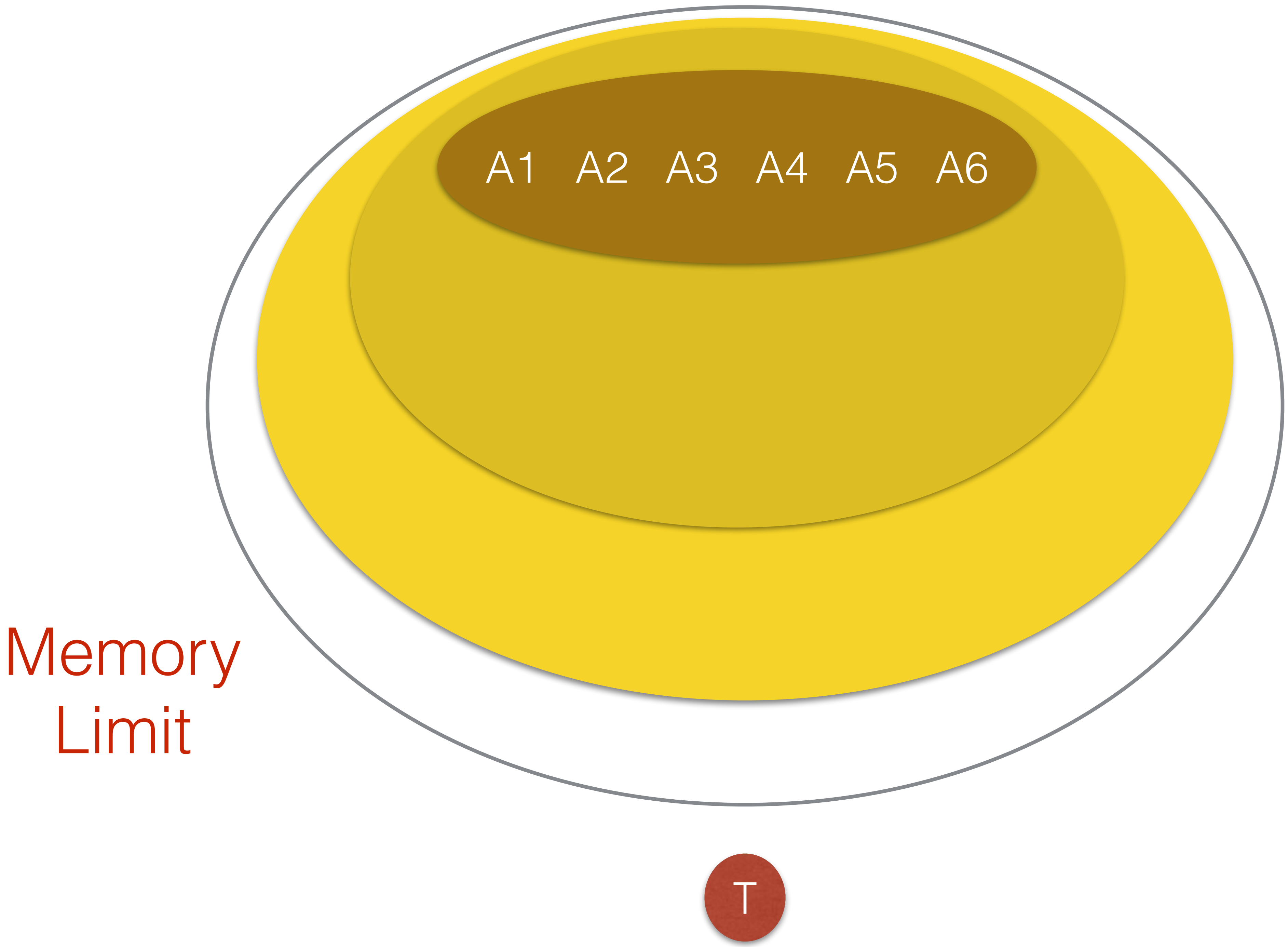
The theory of the resolution process is presented in the form of a system of first-order logic with just one inference principle (the resolution principle). The completeness of the system is proved; the simplest proof-procedure based on the system is then the direct implementation of the proof of completeness. However, this procedure is quite inefficient, and the paper concludes with a discussion of several principles (called search principles) which are applicable to the design of efficient proof-procedures employing resolution as the basic logical process.

state-of-the art machines —> great theory

Today

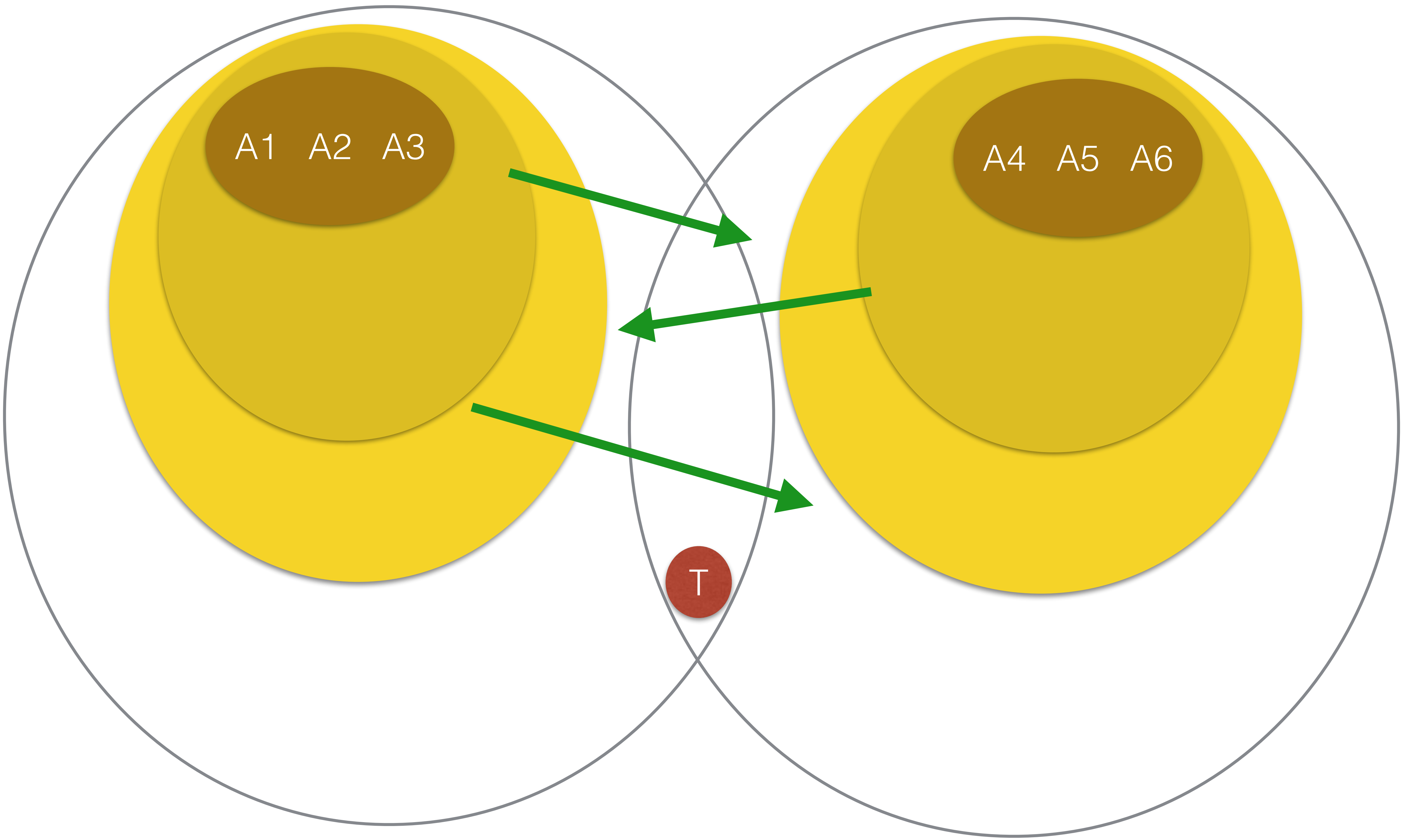
- What are the great machines of our time? e.g. VSC !
- Are we using them for automated deduction? No…
- Should we use them? Yes! Why?
  - To discover new elegant automated reasoning principles
  - To solve practical problems that:
    - can be encoded as logical conjectures (not only math)
    - cannot be solved by current theorem provers
      (1539 problems out of 20306 in the TPTP library)
      (Example: Collatz Conjecture)

Current Theorem Provers



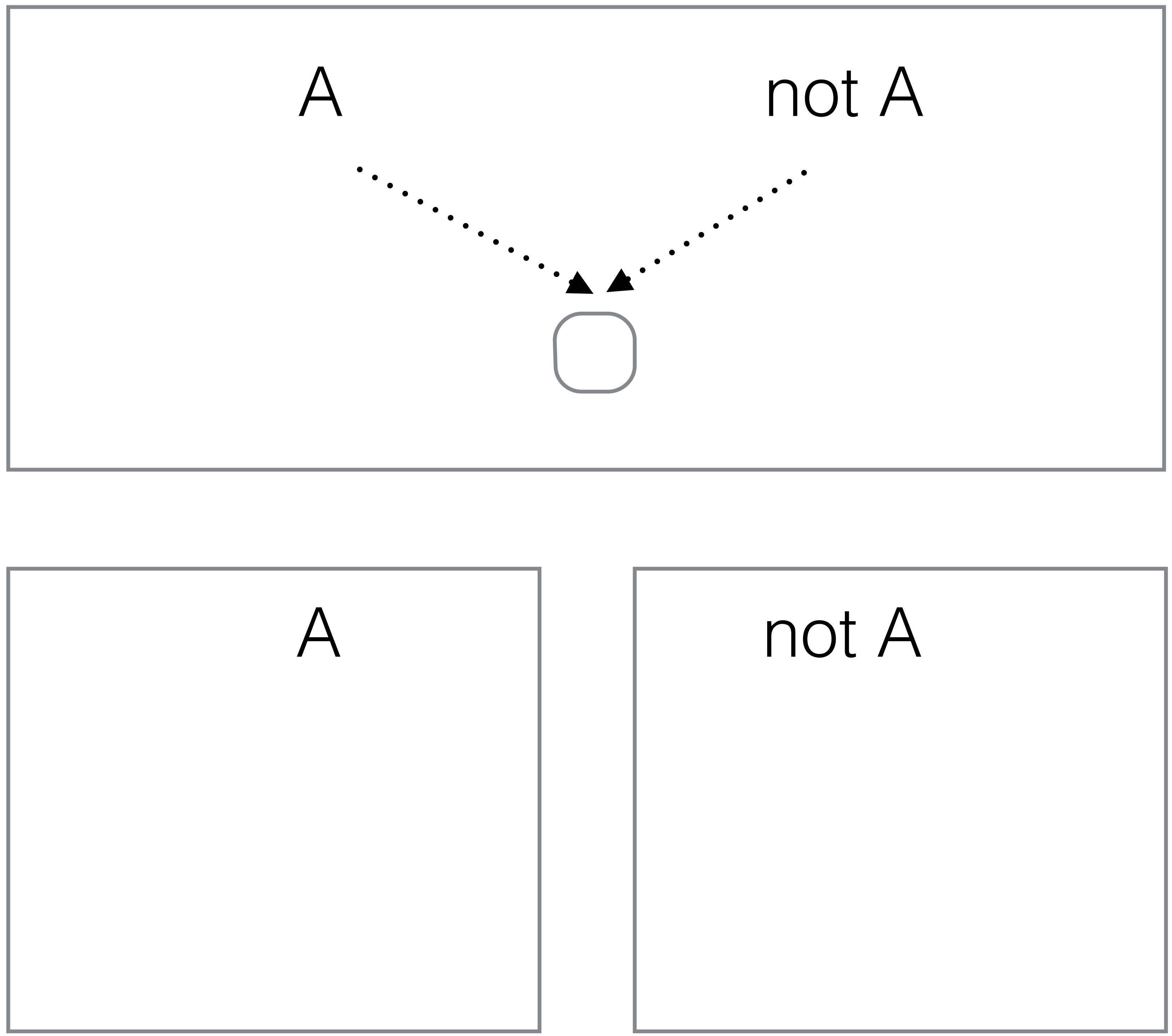
resolution refinements, orderings, conflict-driven learning, reasoning modulo theories (SMT),…

The idea of this project



A Completeness Issue

illustrated on a trivial example



Needs

- Ideally: a PhD student for 3 years,
  with previous experience in HPC or ATP

Concrete Tasks and Goals

- Modify an open-source prover (e.g. Lean)
  to use many cores and nodes

- Develop a sound and complete ATP architecture
  such that a problem is divided among nodes
  and nodes collaborate to solve the problem

- Try the 1539 unsolved problems of the TPTP library