An Instantiation-Based Theorem Prover for First –Order Programming

Carnegie Mellon
Erik Zawadzki
Geoffrey Gordon
André Platzer

- **Mixed integer linear programs (MILP)** are an incredibly useful language for problems in operations research, planning, VSLI, and many other AI problems.
- But they are:
  - Purely **propositional**!
  - **Unable to reason about classes of objects** and relations except by a **tedious and expensive process of unrolling**
  - Unrolling **destroys structure** and prevents **lifted reasoning**
- **First-order programming (FOP)** is a language that generalizes both first-order logic (FOL) and MILP—it has the benefits of both
- Our solver **InstGen iteratively refines a propositional relaxation** by instantiating
  - Refined propositional formula eventually become infeasible iff the first-order formula is infeasible
  - **Sound and refutationally complete**

Propositional MILP

$$\begin{aligned} \max \quad & \mathcal{V} \\ \text{s.t.} \quad & \left(\sum_{k \in \mathbf{I}_{\Sigma_{ij}}} \kappa_{ijk} \cdot p_{ijk} \right) + \mathcal{U}(1 - d_{ij}) \geq \mathcal{V} \\ & \sum_{j \in \mathbf{I}_{C_i}} d_{ij} \geq 1 \\ & p_{ijk} \in \text{Range}_{ijk} \\ & d_{ij} \in \{0, 1\} \end{aligned}$$

Algorithm

1. Propositionalize
2. Try to solve propositional MILP and get feasible solution
3. If propositionally infeasible, then first-order infeasible!
4. Else, if there is a discordant pair there is a new clause to instantiate and add. Goto 1.
5. Else, we are feasible!

First-Order Programming

- Combines the expensiveness of a first-order representation with the strength of MILP-like syntax

$$\mathbf{F} = \bigwedge_{\text{Var}} C_1 \wedge \ldots \wedge C_n$$ — **Min Quantification** ($\bigwedge_{\text{Var}}$), **Max-Clause** (C_n)

$$C_i = \Sigma_{i1} \vee \ldots \vee \Sigma_{im}$$ — **Sum-Clause**

$$\Sigma_{ij} = \kappa_{ij1} \cdot P_{ij1} + \ldots + \kappa_{ijk} \cdot P_{ijk}$$ — **Literal**

- Can **compactly represent** any statement of FOL
- Some statements are **exponentially smaller** in FOP than FOL
- Semantics are similar to FOL, but predicates can range upon compact intervals of reals or integers rather than just *true* and *false*
- We are interested in the **value of the formula**, especially whether it is positive (**feasible**)
- **Example**: the axioms of equality

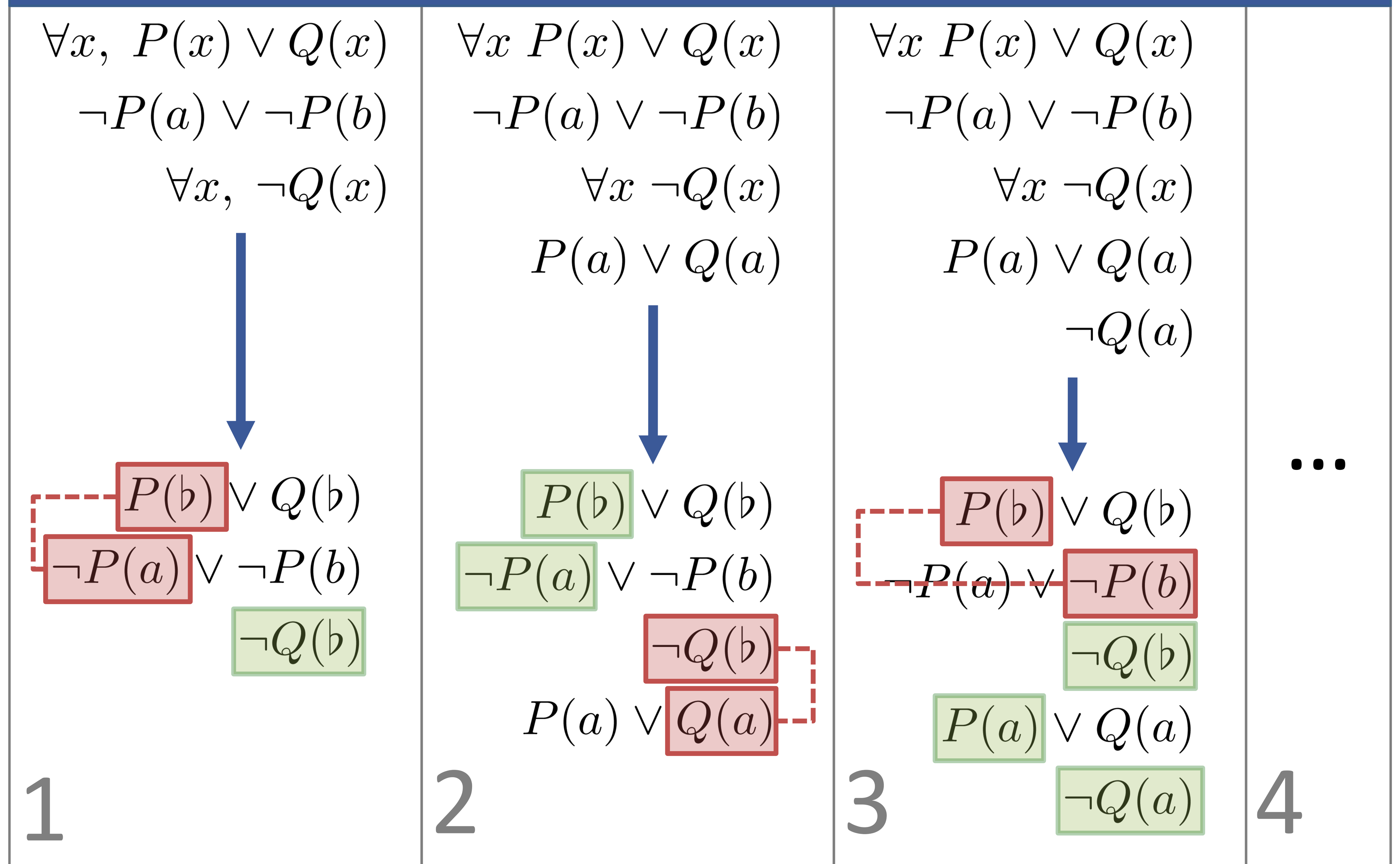
$(i = i) \geq 1$ — **Reflexive**

$(i = j) + (j = i) \leq 0 \vee (i = j) + (j = i) \geq 2$ — **Symmetric**

$(i = j) + (j = k) + (i = k) \leq 1$ — **Transitive**

$\vee\, (i = j) + (j = k) + (i = k) \geq 3.$

Inference Example



- An example of the first 4 steps of an InstGen proof
- So far in this proof, every propositional MILP is feasible
- Each **discordant pair** in the model suggest a new instantiation
- This particular proof takes 6 iterations before discovering a propositional MILP that demonstrates the infeasibility of the first-order program

---

- **InstGen** is a sound and refutationally complete inference procedure based on **parsimonious instantiation**
- The first implemented inference procedure for FOP
- Directions for the future include
  - **Improved automation:** developing good heuristics for picking discordant pairs and deleting implied clauses that are no longer interesting
  - **Applications:** finding good problems with first-order structure and a large or unknown number of objects
  - **Probabilistic First-Order:** we want to use lifted reasoning on problems that exhibit uncertainty. How can we define a semantically sensible stochastic process for generating first-order models?