An Instantiation-Based Theorem Prover for First–Order Programming

- 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

First-Order Programming

- Combines the expensiveness of a first-order representation with the strength of MILP-like syntax
- \[ F = \bigwedge_{i=1}^{n} C_i \lor \ldots \lor C_n \]
  - Min Quantification
  - Max-Clause
- \[ C_i = \Sigma_i \lor \ldots \lor \Sigma_{im} \]
  - Sum-Clause
- \[ \Sigma_{ij} = \kappa_{ij} \cdot P_{ij} + \ldots + \kappa_{ijk} \cdot P_{ijk} \]
  - 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
  - Reflexive: \((i = i) \geq 1\)
  - Symmetric: \((i = j) \leq (j = i) \leq 2\)
  - Transitive: \((i = j) + (j = k) + (i = k) \leq 1\)

Inference Example

\[ \begin{align*}
\forall x, \ P(x) & \lor Q(x) \\
\neg P(a) & \lor \neg P(b) \\
\forall x, \neg Q(x) & \\
P(a) & \lor Q(a)
\end{align*} \]

Propositional MILP

\[
\begin{align*}
\text{max} & \quad \mathcal{V} \\
\text{s.t.} & \quad \left( \sum_{k \in \mathcal{L}_{ij}} \kappa_{ijk} \cdot p_{ijk} \right) + \mathcal{U}(1 - d_{ij}) \geq \mathcal{V} \\
& \quad \sum_{j \in \mathcal{L}_{ij}} d_{ij} \geq 1 \\
& \quad p_{ijk} \in \text{Range}_{ijk} \\
& \quad d_{ij} \in \{0, 1\}
\end{align*}
\]

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!

- 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?