Vampire & Machine Learning

The saturation context for ATP learning

First-Order Logic

FOL - Background

- Expressive
- Flexible
- Arguably intuitive
- *Undecidable proof search*

FOL – terms

- Constants *c*
- Functions *f(t1, t2, …)*
- Variables *x*

FOL – atomic propositions

- Truth values $\top$, $\bot$
- Propositions P, Q, …
- Predicates P(t1, t2, …)
- Equalities t1 = t2

FOL – connectives

- ¬P – “negation”, “not P”, “P is not the case”, “P is absurd”
- P ∧ Q – “conjunction”, “P and Q”, “both P and Q are the case”
- P ∨ Q – “disjunction”, “P or Q”, “one of P or Q holds”
- P ⇒ Q – “implication”, “P implies Q”, “if P then Q”
- P ⇔ Q – “equivalence”, “P is equivalent to Q”, “P and Q are the same”

FOL - Quantifiers

- $\forall x.\ P$ – “for all x P”, “P holds whatever x is”
- $\exists x.\ P$ – “exists x P”, “there is at least one x for which P holds”

Applications

- Mathematics (mostly)
- AI
- Natural language semantics
- Software/hardware systems
- Embedding other logics
- …

Vampire

Automating FOL

Setting

- Axioms A
- Conjecture C
- Prove C, possibly with the help of (some of) A.
- How?

Refutation theorem proving

- Assume axioms
- Assume *negated* conjecture
- Make (valid) deductions
- Deduce $\perp$
- Done!

Simplifications

- Replace equivalences with implications
- Replace implications with disjunctions
- Push negations “all the way in” (NNF)
- Remove trivial sub-formulae

Quantifier elimination

- Drop universal quantifiers, leaving free variables
- Introduce fresh constants for existential quantifiers - *skolemisation*

CNF

- Still quite a lot of structure
- Push bits around until you arrive at an “AND of ORs” – CNF
- A set of “clauses”

Resolution

- Only one inference (*well*, *actually*…)
- Take two clauses and produce another one
- “unify” the variables involved
- Empty clause means a proof has been found

Vampire's Algorithm

- Convert input to set of clauses.
- Perform all possible resolutions, but in a *fair* way.
- If you find the empty clause, you found a proof.
- If you ran out of inferences, the statement isn't true.
- If you timed out…you timed out!
- Plus *lots* of optimisations and options

Demo

- TPTP input syntax
- Default Vampire options – many more to choose from

Vampire: much more

- Reasoning with theories
- Reasoning with higher-order logics (!)
- Different proof calculi/algorithms
- “Limited resource” modes
- “Disprove” modes
- Strategy scheduling
- Many, *many* options

Machine Learning for Vampire

Arguments

- Vampire is *good* because it is *fast* (but not parallel)
- “Internal” heuristics inside the algorithm are likely to be *slow*
- “External” heuristics are better and (can) have more effect

Axiom Selection

- Some domains are very large, with millions of axioms
- Vampire is not happy about this
- Fewer axioms are better!
- …provided the ones you need are still present
- Manual heuristics exist (SInE), but work also progressing on machine-learned axiom selection policies

Strategy Selection

- Vampire has “strategies” (combinations of options)
- Some strategies are likely to solve a problem, others not so.
- Pick a strategy!
- Quite hard, but rewarding if you get it right.

Strategy Scheduling

- Often better to run lots of strategies quickly than one longer one.
- Vampire has “strategy schedules” to do this
- You could be waiting a while!
- Reshuffle the schedules so that you wait less.

Parameter tweaks*

- Vampire has many internal parameters which bias the algorithm.
- Imagine a system “learning to drive” Vampire by changing these.
- For example: age/weight ratio.
- Possibly a reinforcement learning topic: unclear what to optimise for.

???

- Vampire is an established theorem prover, with top performance
- Currently it does no learning whatsoever.
- Large space for machine learning research and application.