

Taking Satallax by the hand

Michael Färber

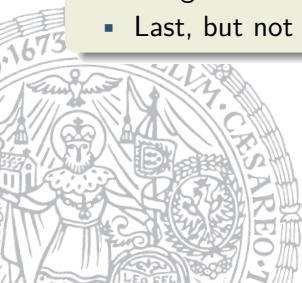
9 November, 2015



My secret mission in Praha

Goals

- Suck out knowledge of Chad and Stephan (“Vampire”-style)
- Get to know Satallax
- Integrate machine learning into Satallax
- Last, but not least . . . learn Czech :)



Satallax has a priority queue, where commands are put in with a certain priority (or delay)

Commands

- ProcessProp1 (~99% of all commands!)
- Mating
- Confrontation
- ...

Delays

- FORALL_DELAY
- CHOICE_EMPTY_DELAY
- POST_MATING_DELAY
- ...

Satallax (cont'd)

ProcessProp1(m)

Pick the formula m from the unprocessed formulas, make it react with the processed formulas and possibly put new formulas into unprocessed formulas



Machine Learning

Learning

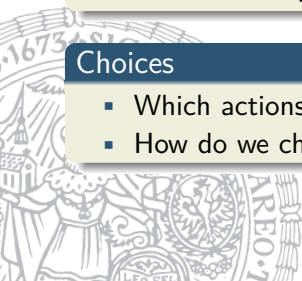
Record actions which led to success in certain situations

Classification

When encountering new situation, find actions most likely to lead to success, based on previous knowledge

Choices

- Which actions are hard to choose?
- How do we characterise situations?



Machine learning in Satallax

- Critical decision: Which priority do commands get, i.e. in which order are formulas processed?
- Prover state: Which propositions have already been processed?



What is a prover state?

Intuition

Depends on formulas we have already worked on + original problem

First try

- “Lean” approach: state = all constants of all processed terms
- Problems:
 - Does not work when constants are different across problems!
 - Gives Chad stomach ache.

Second try

Abstract over terms (generality vs precision), i.e. don't save $b = f(a, a) \vee a \neq b$, but $f_{20}(f_{00}, f_{00}) = f_{01} \vee f_{00} \neq f_{01}$ (see Stephan's paper “Learning Search Control Knowledge for Equational Theorem Proving” from 2001)

Implementation

Recording proof data

1. When a proof is found, we store every inference contributing to the proof.
2. Rerun Satallax and store the prover state for every previously recorded inference.

Processing proof data

We convert recorded data to a representation that allows fast prediction of formulas -> see FEMaLeCoP.

Using proof data

When inserting a ProcessProp1 command into the priority queue, we run the ML algorithm on the prover state. If the proposition matches a predicted proposition, we influence its delay.

TODO

- Complete implementation
- Integrate simpler heuristics into Satallax, such as formulas size or number of common symbols with processed formulas
- Experiment with different features (prover state characterisations)
 - Compare abstract vs. concrete terms
 - Evaluate ML performance on suitable HOL test sets
 - ... profit?

