# MaLeCoP
# Machine Learning Connection Prover

Josef Urban, Jiří Vyskočil, Petr Štěpánek

July 7, 2011

# Reasoning in complex mathematical theories

- Large formal theories recently built with proof assistants
- Many advanced proofs containing many ideas
- We should be able to re-use them!
- Mizar Mathematical Library (set theory) - 100k facts/proofs
- Isabelle/HOL - translated to first-order logic - 20k facts
- We want strong automated reasoning support for such systems
- "Sledgehammer" already popular among Isabelle/HOL users

# The MPTP Challenge benchmark

- 2006/7: Bring large-theory problems to ATP developers
- Predecessor and initial design of the CASC LTB category (2008)
- 252 Mizar problems leading to Bolzano-Weierstrass theorem
- All previous theorems/definitions available for each problem
- More than thousand of formulas in some problems

## Successful large-theory techniques

- Symbol-based premise selection heuristics (SInE, Sledgehammer, MaLARea)
- Goal-directed calculi (LeanCoP, some E prover's strategies)
- Model-based (semantic) premise selection (Vyskocil, Pudlak, MaLARea, SRASS)
- Learning premise relevance from proofs (MaLARea)

# Our goal here:

- Improve "standard" ATPs by applying the successful large-theory techniques
- Plan: guide ATPs' internal inference analogously to guiding premise selection
- This should allow ATPs to solve harder problems

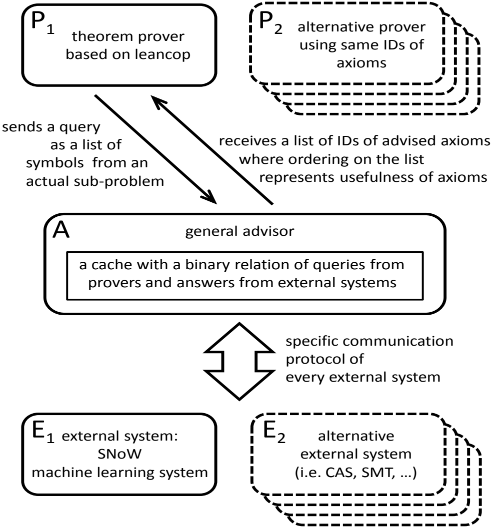
# Machine Learner for Automated Reasoning (MaLARea)

- Solve some unsolved problems
- Use solutions to learn relevance of premises for conjectures
- Use only a fragment of the most relevant premises for the unsolved problems
- Loop (combining learning with ATPs and model finders)

# LeanCoP: Lean Connection Prover

- Connected tableau calculus - goal oriented
- Very good performance on the MPTP Challenge (better than SPASS)
- Tableau seems more suitable than resolution for guiding clause selection
- Compact Prolog implementation - easy to modify
- This already allowed quite advanced combinations of strategies (and so we hope for more)

# General Advising Design



# LeanCoP modifications

- Consistent clausification across many problems needed for consistent learning/advice
- Options like definition introduction need to be fixed
- Providing training data for external advising systems
- Mechanisms for taking advice from external system(s)
- Profiling mechanisms
- External advice is quite slow: number of strategies defined trading advice for speed

# External advice

- Now used for clause selection when extending the tableau
- Needs to be reasonably fast and general
- The SNoW system (naive Bayes, C++) used
- Library of functions implementing fast caching, communication, data translation

# Strategies for Guidance

- advising every clause selection step is very slow
- dozen parametrized strategies and meta-strategies
- trade fast blind raw inference for slower guided inference
- trade caching the guidance in Prolog for memory
- trade (theoretical) completeness for speed by focusing only on relevant clauses
- Example: ask for advice only when there are many matching clauses
- Example: ask for advice only for a small number of initial steps
- Example: use one global relevance order instead of local ones for each path

# Experiments: Proof Search Shortening

- First solve as many of the 252 MPTP Challenge problems as possible with LeanCoP
- 73 problems solved, providing 703 training examples for the advisor
- Then SNoW trained and run as a daemon, advising following MaLeCoP runs on the problems
- The number of extension steps drops 20 times on average
- All 73 problems solved again

# Experiments: Solving New Problems

- Seven strategies tested on the unsolved problems
- The most successful solves eight more problems
- All seven solve fifteen more problems
- Not directly comparable to MaLARea (no looping yet, etc.)
- Finer guidance seems to generally help more than just course initial relevance-ordering

# Experiments: Solving New Problems

1. leancop_ala_malarea
2. limited_smart_with_first_advice(3)
3. scalable_with_first_advice(40,limited_smart_and_complete_with_fa(3))
4. scalable_with_first_advice(3,20,original_leancop)
5. limited_smart_with_first_advice(4)
6. scalable_with_first_advice(3,20,limited_smart_with_first_advice(5))
7. limited_smart_and_complete_with_first_advice(3)

# Solving New Problems

| problem | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| t26_finset_1 | | | | | | 4033 | |
| t72_funct_1 | 1310 | | | | | | |
| t143_relat_1 | | 28458 | 59302 | | | | 61660 |
| t166_relat_1 | | 17586 | | | 4067 | 5263 | |
| t65_relat_1 | | | | | 79756 | 36217 | |
| t43_subset_1 | | 82610 | | | | | |
| t16_wellord1 | | | | | 37148 | | |
| t18_wellord1 | | | 3517 | 2689 | | | 2524 |
| t33_xboole_1 | 3659 | 16456 | | | | 16902 | 17925 |
| t40_xboole_1 | | 16488 | | | 15702 | | 28404 |
| t30_yellow_0 | 24277 | | | | | | |
| l2_zfmisc_1 | | | | | | | 85086 |
| l3_zfmisc_1 | | | | | | | 79786 |
| l4_zfmisc_1 | | 17074 | | | 9584 | 14299 | 30273 |
| t9_zfmisc_1 | | | 80684 | | | | 77532 |

# Future Work and Conclusion

- ATP system that can use dynamic external low-level guidance and provides low-level training information to external AI systems
- ATP/learning architecture that is more deeply integrated than other systems
- This poses new interesting problems, but it already seems to practically work

Future work:

- Better strategies, better profiling, faster advice
- Investigate the trade-off between completeness and relevance
- Integrate more techniques used in MaLARea: semantic selection, looping
- Learn even on unsuccessful proofs how to avoid bad choices
- Integrate other kinds of systems (CASes like LeanCoP-Omega?)
- Try detecting terminating classes of inputs/outputs
- ...