TicTacToe in HOL4

Thibault Gauthier

University of Innsbruck

Motivation

How can we improve automation in ITP?

- connect to strong external provers: E-prover and select relevant lemmas: HolyHammer SledgeHammer MizAR.
- improve current tactics in our ITP.

Ultimate meta-tactic: select relevant tactics and apply them.

Plan

Related works.

Recording tactics.

Learning.

Tactic-based search.

Evaluating.

Tactician: HOL Light

Author: Mark Adams

Goal:

$$(\forall x.\ x = x) \wedge (\forall y.\ y = y)$$

```
Tactic: REPEAT CONJ_TAC THEN GEN_TAC THEN REFL_TAC
```

```
Interactive proof:
e (REPEAT CONJ_TAC)
(*** Branch 1 ***)
e (GEN_TAC)
e (REFL_TAC);;
(*** Branch 2 ***)
e (GEN_TAC);;
e (REFL_TAC);;
```

ML4PG: Recycling proof patterns in Coq

Authors: E.Komendantskaya and J.Heras

Goal: refractoring libraries by discovering similarity between proofs

Features (captured at each proof step):

1. Names and number of tactics used in one command line.
2. Types of the tactic arguments.
3. Relation of the tactic arguments to the hypotheses or lemmas.
4. Three top symbols in the term-tree of the current subgoal.
5. Number of subgoals each tactic command-line generates.

Unsupervised learning: clustering algorithm.

Pre-recording

Modifying tactical proofs so that they record their tactics.

Tactic: REPEAT CONJ_TAC THEN GEN_TAC THEN REFL_TAC

Wrapping: record (REPEAT CONJ_TAC, “REPEAT CONJ_TAC”) THEN record (GEN_TAC, “GEN_TAC”) THEN record (REFL_TAC, “REFL_TAC”)

Generalizing: record (REPEAT CONJ_TAC, “Tactical.REPEAT Tactic.CONJ_TAC”) THEN record (GEN_TAC, “Tactic.GEN_TAC”) THEN record (REFL_TAC, “Tactic.REFL_TAC”)

Pre-recording: example

The tactic ARW is generalized as:

```
( let open simpLib in ( let val ARW = BasicProvers.RW_TAC ( let
val arith_ss = boolSimps.bool_ss ++ numSimps.ARITH_ss in
arith_ss end ) in ARW end ) [ ] end )
```

to be readable from its string representation anywhere in the program.

Recording and learning

Features (captured at each tactical step):

1. top goal: subterms, types.
2. current subgoal: subterms, types, variables, logical structure.
3. Time taken by the tactic.
4. To do!: relation of the tactic arguments to the current goal.

Predictions

Before the search:

- Pre-selection of 500 tactics: k-NN with top-goal features.

During the search:

- Re-ordering by k-NN of the feature vectors based on the current subgoal features.

Searching

New goals:

- goal$_0$: $(\forall x.\ x = x) \wedge (\forall y.\ y = y)$

Proof state:

- goal$_0$: [(CONJ_TAC,0.9),(STRIP_TAC,0.5),...]

New goals: CONJ_TAC

- goal$_1$: $(\forall x.\ x = x)$
- goal$_2$: $(\forall y.\ y = y)$

Proof state:

- goal$_0$: [(STRIP_TAC,0.5),...]
- goal$_1$: [(REFL_TAC,0.6),...], pending: goal$_2$

Early results

Settings:

- Tactic time out: 0.02 seconds
- Search time out: 5 seconds

Results:

- 1974 proofs of 7951 theorem.
- a proof was constituted of 37 tactics
- 100 percent reconstruction rate.

Typical search:

- infstep : 60-200
- nodes : 25-40
- maxdepth: 4-7

Bugs (or special parameters): subterms features

Example in gcdTheory

GCD_ADD_L:

$$\forall a\ b.\ gcd\ (a + b)\ a = gcd\ a\ b$$

Human proof:

PROVE_TAC[GCD_SYM,GCD_ADD_R]

TicTacToe proof:

- ARW_TAC
- THEN MATCH_MP_TAC (SPECL [a, a + b] IS_GCD_UNIQUE)
- THEN ARW [. . .] IS_GCD_MINUS_R
- THEN PROVE_TAC [GCD_IS_GCD, IS_GCD_UNIQUE, IS_GCD_SYM]

HolyHammer: GCD_SYM GCD_ADD_R

Example in rich_listTheory

EXISTS_TAKE:

$$\forall m\ l.\ m \leq LENGTH\ l \Rightarrow \forall P.\ EXISTS\ P\ (TAKE\ m\ l) \Rightarrow$$
$$EXISTS\ P\ l$$

Human proof:

- REPEAT GEN_TAC THEN DISCH_TAC
- THEN IMP_RES_THEN SUBST1_TAC TAKE_SEG
- THEN MATCH_MP_TAC EXISTS_SEG
- THEN ASM_REWRITE_TAC [ADD_0]

TicTacToe proof:

- REPEAT numLib.INDUCT_TAC
- THEN Cases_on `l` THEN ...
- THEN REPEAT STRIP_TAC THEN ...

HolyHammer: EXISTS_TAKE_IMP

Example in rich_listTheory

FCOMM_FOLDL_FLAT:

$$\forall f\ g.\ FCOMM\ f\ g \Rightarrow \forall e.\ RIGHT\_ID\ g\ e \Rightarrow$$

$$\forall\ l.\ FOLDL\ f\ e\ (FLAT\ l) = FOLDL\ g\ e\ (MAP\ (FOLDL\ f\ e)\ l)`$$

TicTacToe proof:

- GEN_TAC THEN GEN_TAC THEN . . . THEN DISCH_TAC THEN
- SNOC_INDUCT_TAC THENL

goal$_1$:

- ASM_REWRITE_TAC [FLAT_SNOC, MAP_SNOC, MAP, FLAT, FOLDL_SNOC]
- THEN IMP_RES_TAC FCOMM_FOLDL_APPEND
- THEN ASM_REWRITE_TAC [APPEND_NIL, APPEND_SNOC, FOLDL_SNOC, FOLDL],

goal$_2$: same as goal$_1$ except last step is ASM_REWRITE_TAC [ ]

Holyhammer: No proof found with Eprover (BliStr) in 30s.

Improvements: Is 5 seconds enough?

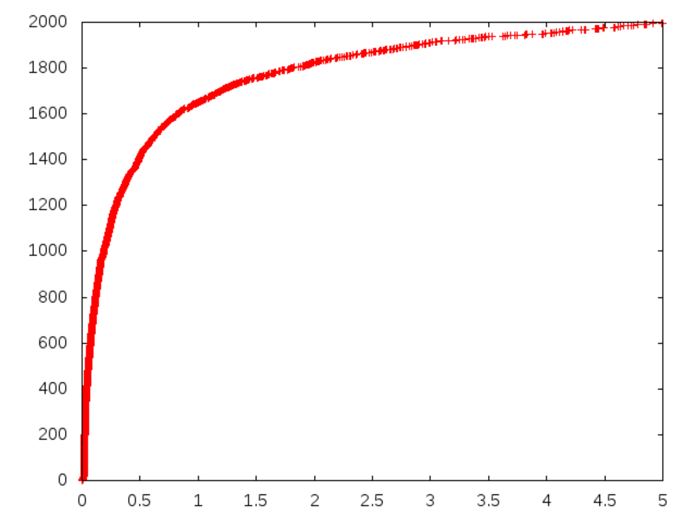

Figure 1: Number of proofs found (y axis) in a fixed time (x axis)

Improvements: Proof search (implemented)

- Caching previous predictions and tactic applications
- Penalty on the depth or/and width of the subgoals.

Improvements: Learning (implemented)

- Learning from previously discovered proofs.
- Relabelization of feature vectors. Clustering ?.
    - Improve orthogonalities of the tactics.

Improvements: Predictions (not implemented)

Theorems and terms (arguments of tactics) have also features.

Example 1: REWRITE_TAC *theorem_list*

- REWRITE_TAC [FLAT_SNOC, MAP_SNOC]
- FOLDL_SNOC is close to the current goal.
- REWRITE_TAC [FLAT_SNOC, MAP_SNOC, FOLDL_SNOC]

Example 2: Cases_on *variable_name*

- Cases_on `y`.
- y doesn't appear in the goal but `x`.
- Cases_on `x`.