Conjecturing over large corpora

Thibault Gauthier Cezary Kaliszyk Josef Urban

April 6, 2016

Goal

Automatically discover conjectures in formalized libraries.

Which formalized libraries ?

	theorems	constants	types	theories
Mizar	51086	6462	2710	1230
Coq	23320	3981	860	390
• HOL4	16476	2188	59	126
HOL Light	16191	790	30	68
Isabelle/HOL	14814	1046	30	77
Matita	1712	339	290	101

Why formalized libraries ?

- Easier to learn from.
- Sufficiently large number of theorems ?

What for ?

- Improve proof automation, by discovering important intermediate lemmas.

Challenges

How do we conjecture interesting lemmas ?

- Generation: large numbers of possible conjectures
- Learning: large amount of data
- Pruning: how to remove false conjectures fast, and select interesting ones

How to integrate these mechanism in a goal-oriented proof ?

Our approach

Conjecturing:

	Current solution	Limitation	Available improvement
Generation	analogies	small space	probabilistic grammar
Learning	pattern-matching		genetic algorithm
Pruning	proof	too slow	model-based guidance

Proof strategy including intermediate conjectured lemmas.

- Copy human reasoning.
- Make high-level inference steps: premise selection + ATPs.

Finding analogies

Theorems (first-order, higher-order or type theory):

$$\forall x : \mathit{num}.\ x + 0 = x \qquad \forall x : \mathit{real}.\ x = x \times s(0)$$

Normalization + Conceptualization + Abstraction $\rightarrow$
Properties:

$$\lambda \mathit{num}, +, 0.\ \forall x : \mathit{num}\ x = x + 0 \qquad \lambda \mathit{real}, \times, 1.\ \forall x : \mathit{real}.\ x = x \times 1$$

Derived constant pairs:

$$\mathit{num} \leftrightarrow \mathit{real},\ +\ \leftrightarrow\ \times,\ 0 \leftrightarrow 1$$

Scoring analogies

- Number of common properties.
- TF-IDF to advantage rarer properties.
- Dynamical process (similarity of 0 1 $\rightarrow$ similarity of $+$ *).
- Not greedy. Concepts can have multiple analogues.

3881 analogies in HOL4. 5842 if we include subterms.

Analogy		Score
*BIT*2	*BIT*1	0.97
real	*int*	0.96
int_of_num	*real_of_num*	0.95
real	*extreal*	0.94
semi_ring	*ring*	0.94
$\leq$	$<$	0.93

Creating conjectures from analogies

Normalized theorems	Properties	Concept pairs
$x * (y - z) = x * y - x * z$	$Dist(*, -, i)$	$\{- \leftrightarrow +\}$
$x * (y + z) = x * y + x * z$	$Dist(*, +, i)$	$\{* \leftrightarrow \cup, + \leftrightarrow \cap, i \leftrightarrow s\}$
$x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$	$Dist(\cup, \cap, s)$	$\{* \leftrightarrow \cup, - \leftrightarrow \cap, i \leftrightarrow s\}$
$x + 0 = x$	$Neut(+, 0, i)$	$\{- \leftrightarrow +\}$
$x - 0 = x$	$Neut(-, 0, i)$	
$exp(a + b) = exp(a) * exp(b)$	$P(exp, +, *, i, r)$	

Original theorem:

- $exp(a + b) = exp(a) * exp(b)$

Analogies:

- $+ \rightarrow -$
- $+ \rightarrow \cap,\ * \rightarrow \cup$

Conjectures:

- $exp(a - b) = exp(a) * exp(b)$
- $exp(a \cap b) = exp(a) \cup exp(b)$

Untargeted conjecture generation

Procedure:

- Generation of "best" 73535 conjectures from the Mizar library.
- Premise selection + Vampire prove 10% in 10 s.
- 4464 are not tautologies or consequences of single lemmas.

Examples:

- convex - circled

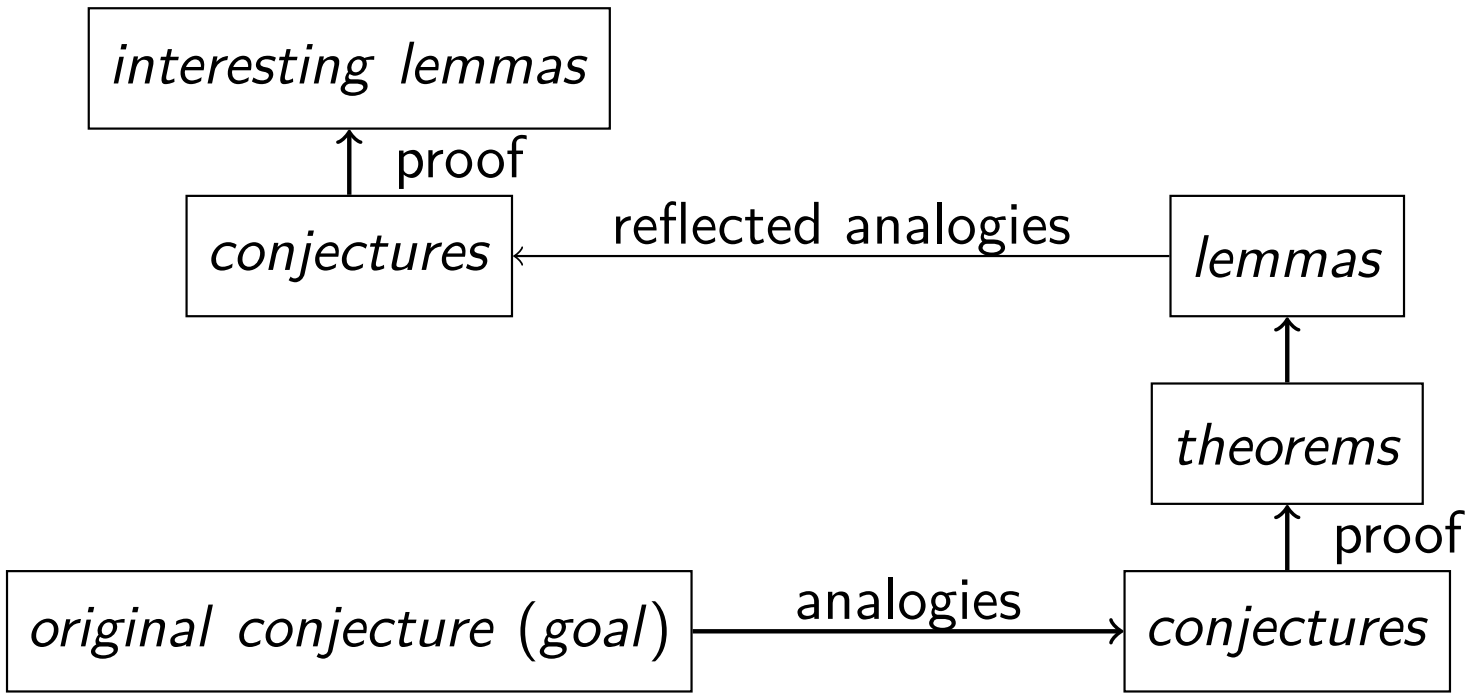
Problem:

- Unlikely to find something useful for a specific goal.
- How to adapt this method in a goal-oriented setting?

Targeted conjecture generation: evaluation settings

	First experiment	Second experiments
Library	Mizar	HOL4
Evaluated theorems	hardest (22069)	all
Accessible library	past theorems	past theorems
Concepts	ground subterms	only constants
Pair creation	pre-computed	fair
Type checking	no	yes
Analogies per theorem	20	20
Premise selection	k-NN 128	-kNN 128
ATP	Vampire 8s	E-prover 8s
Basic strategy	no conjectures	no conjectures
Premise selection	k-NN 128	k-NN 128
ATP	Vampire 3600s	E-prover 16s

First experiment: proof strategy



First experiment: results

	Number	Non-trivial and proven
Hard goals	22069	
Analogous conjectures	441242	3414
Back-translated conjectures	26770	2170
Affected hard goals	500	7
New proven hard goals		1

- Non-trivial theorem: consequences of at least two theorems.
- Affected goal: From the goal, the procedure proves at least one back-translated conjecture.
- Time: 14 hours on a 64-CPU server (proofs)

First experiment: example

```
theorem :: MATHMORP:25
for T being non empty right_complementable Abelian
            add-associative right_zeroed RLSStruct
for X, Y, Z being Subset of T
  holds X (+) (Y (-) Z) c= (X (+) Y) (-) Z
```

Proven using:

- Analogy between + and - in additive structures.
- A conjectured lemma which happens to be MATHMORP:26.

First experiment: limits

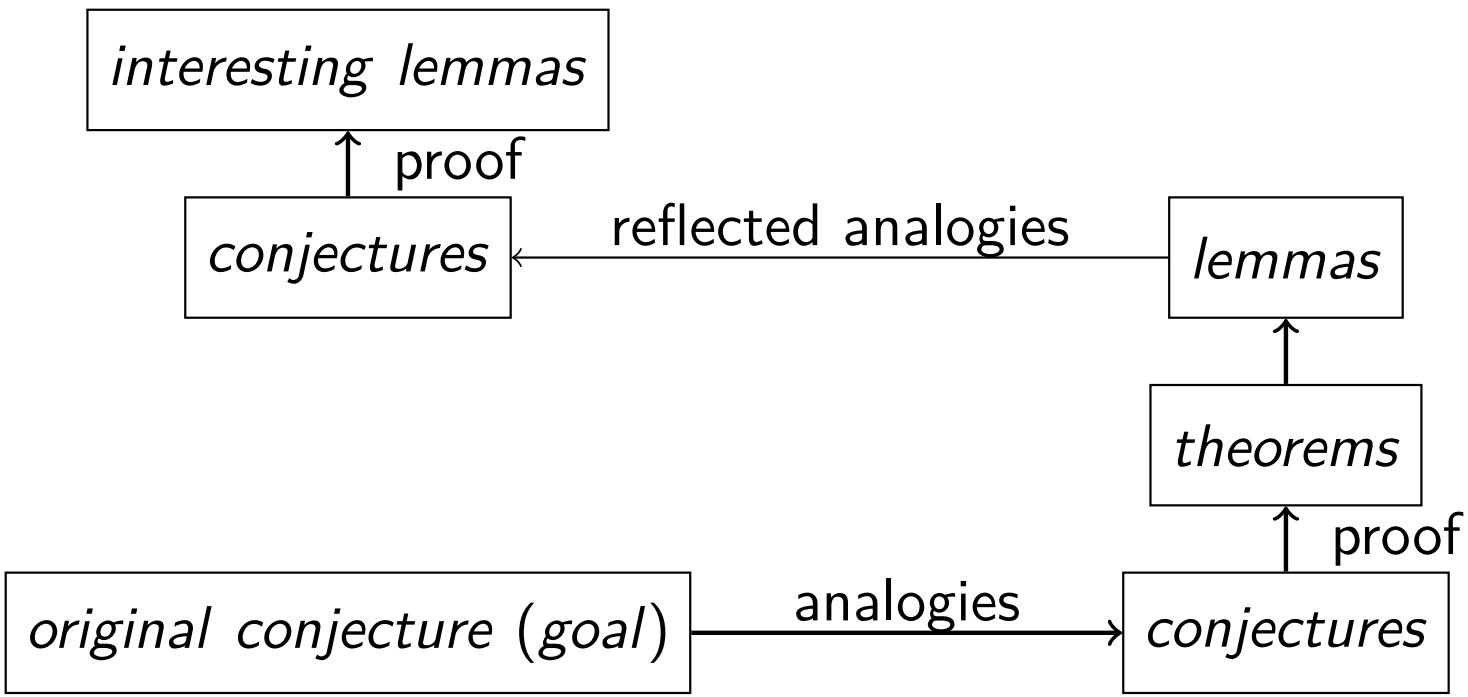
Issues:

- Huge number of proofs.
- Few affected theorems (500).
- Few conjectured lemmas (in average 4 per affected theorems).
- Do not help in proving the goal.

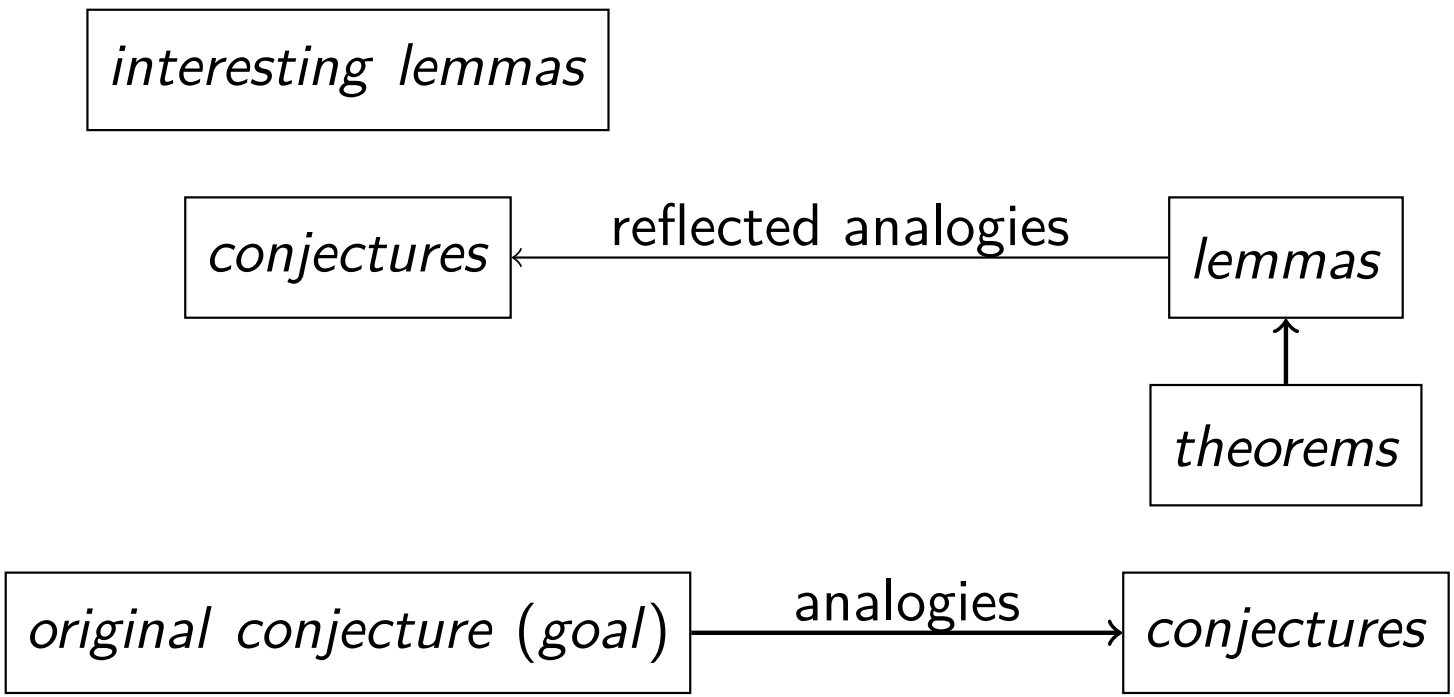
Reasons:

- Design of the strategy.
- Problem set is hard.
- Proof selection is too restrictive.
- Analogies may be too strict.
- No type checking (set theory).
- No understanding of the type hierarchy.

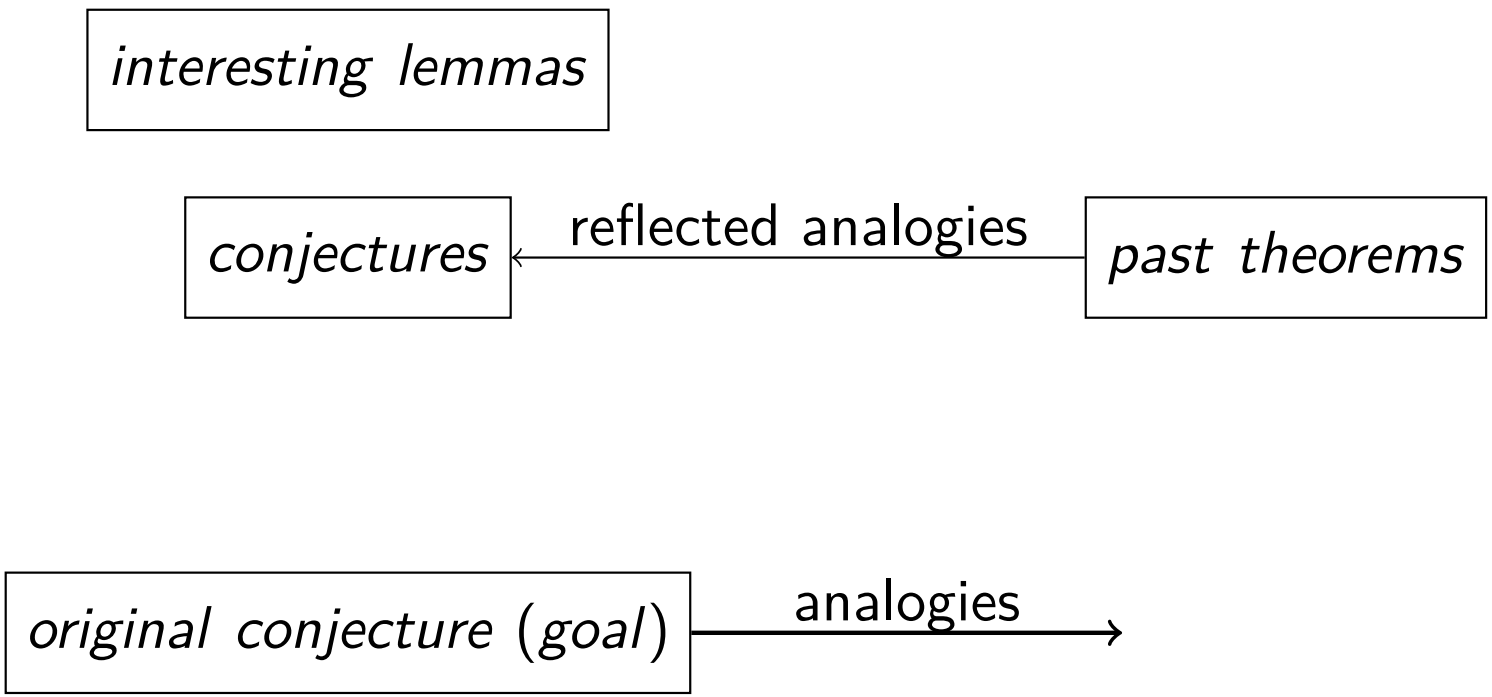
Second experiment: proof strategy



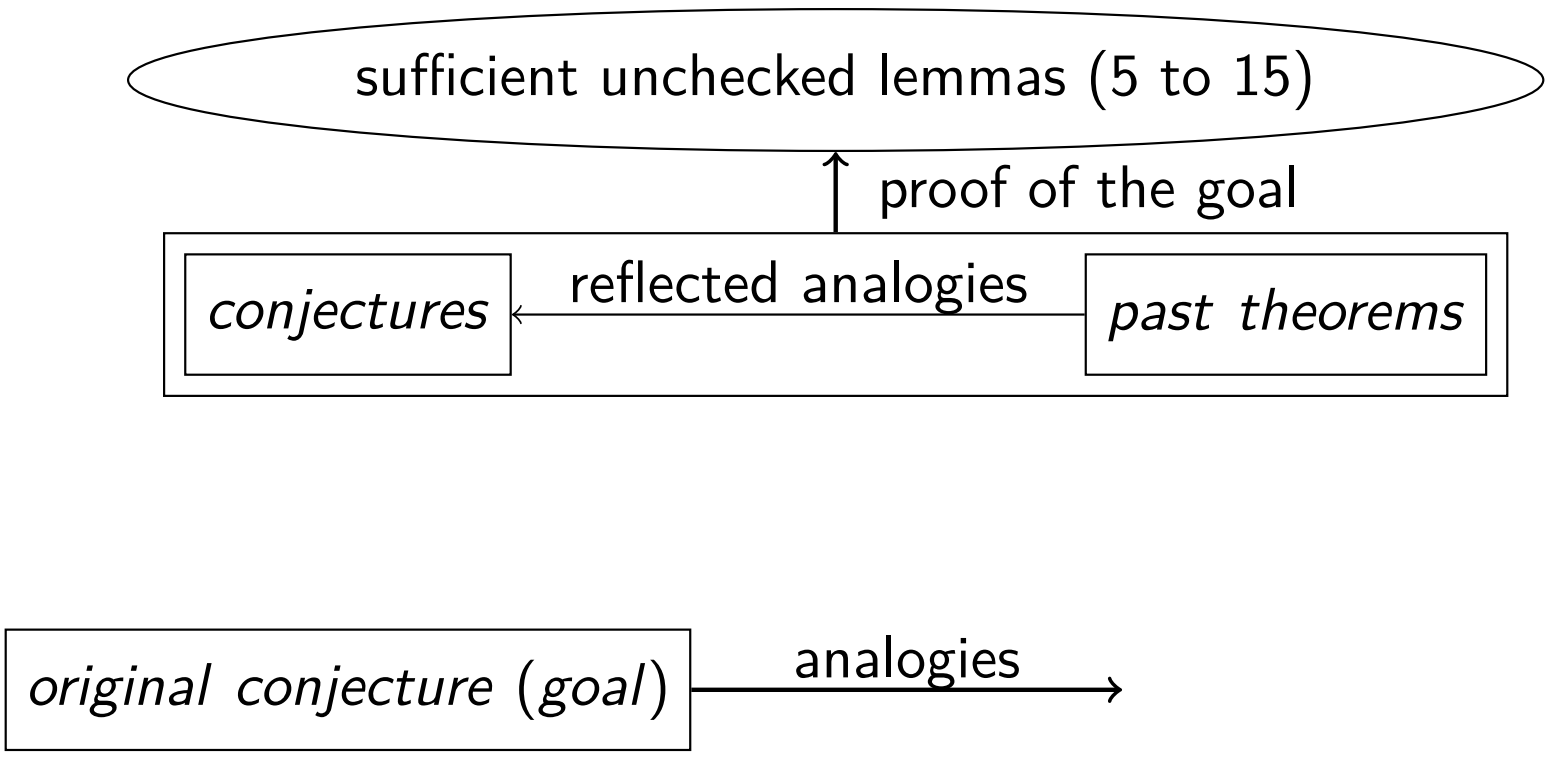
Second experiment: proof strategy



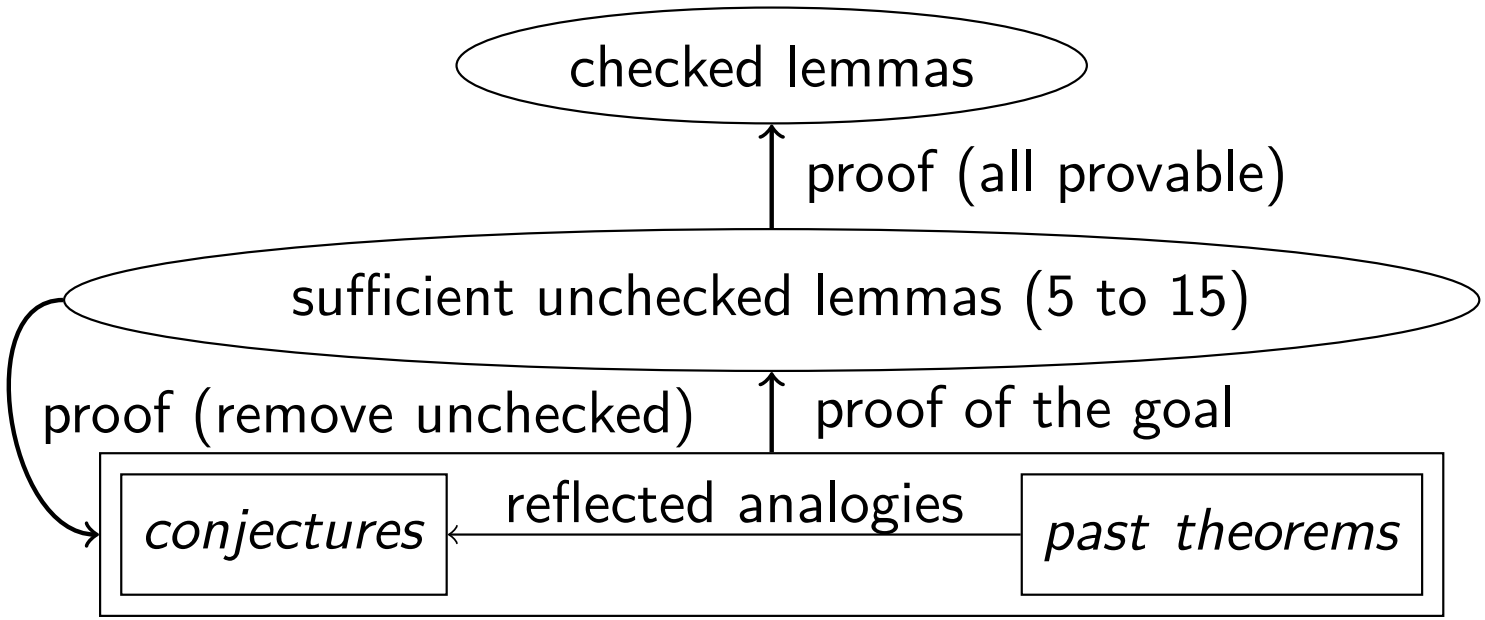
Second experiment: proof strategy

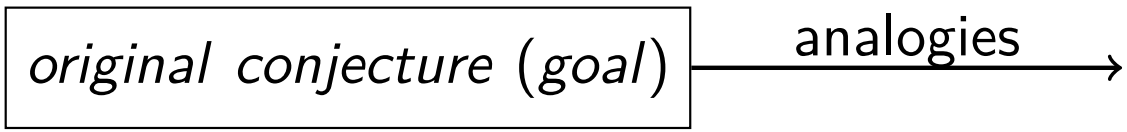


Second experiment: proof strategy



Second experiment: proof strategy





Second experiment: results

Goals	10163
Proven conjectures	8246
Proven goals	2700
Proven goals using one conjecture	724
New proven goals	7

Number of tries	0	1	2	3	4	5	6	7
Proven goals	444	100	58	45	35	21	13	8

Time: 10 hours on a 40-CPU server
(analogies + premise selection + translation + proof)

Reason to be hopeful:
2787 goals were "half-proven".

Second experiment: examples

Theorem	From analogues of
extreal.sub_rdistrib	extreal.sub_ldistrib
pred_set.inter_countable	pred_set.FINITE_DIFF
real.pow_rat_2 (7 tries)	real.POW_2_LT (21 lemmas)
numpair.tri_le	arithmetic.LESS_EQ_SUC_REFL
ratRing.tLRLRRRRRRR	integerRing.tLRLRRRRRRR
words.word_L2_MULT_e3	words.WORD_NEG_L
real.REAL_EQ_LMUL	intExtension.INT_NO_ZERODIV
	integer.INT_EQ_LMUL2

Conclusion

We designed two conjecture-based proving methods.

- Support many ITP libraries.
- Generate conjectures using analogies.
- Learn analogies by pattern-matching and dynamical scoring.
- Integrated in a proof strategy:
  Combine analogies and standard hammering techniques
  (premise selections and translations to ATPs).

We evaluated them.

- 10% of conjectures from best analogies are provable.
- $+1$ hard Mizar problem.
- $+7$ hard HOL4 problem.

Coming sooner or later

- Conjecture generation:
  - more complex concepts
  - probabilistic grammar
  - generalization/specification, weakening/strengthening
- Learning:
  - faster pattern-matching,
  - genetic algorithm + model evaluation.
  - from proofs?
- Pruning or/and guidance:
  - better scoring mechanism for substitutions,
  - model-based guidance.
  - Truth intuition using machine learning (?).
- Improving proof strategies:
  - Recursion
  - Tree search (Monte-Carlo)

Let's have fun !!!