

Automated conjecturing for Mizar

July 14, 2017

Goal

Automatically discover conjectures in formalized libraries.

Why formalized libraries ?

- Large number of theorems to be learned from.

What for ?

- To improve proof automation, by discovering important lemmas.

Challenges

How do we find interesting conjectures?

- Generation: large numbers of possible conjectures
- Learning: large amount of data
- Pruning: how to remove false conjectures fast

How to do we filter interesting proved conjectures?

Our approach

- Start with a large library of theorems (about 150.000).
- Find analogies (about 100.000).
- Apply analogies to create new conjectures.
 - ▶ In theory: about 1 million conjectures
 - ▶ In our experiment: about 8000 conjectures.
- Try to prove these conjectures using E-prover. (about 300 proven conjectures → 100 non trivial ones)

Creating conjectures from analogies

Theorems (CNF,AC)

$$x * (y - z) = x * y - x * z$$

$$x * (y + z) = x * y + x * z$$

$$x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$$

$$x + 0 = x$$

$$x - 0 = x$$

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

Creating conjectures from analogies

Theorems (CNF,AC)

$$x * (y - z) = x * y - x * z$$

$$x * (y + z) = x * y + x * z$$

$$x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$$

$$x + 0 = x$$

$$x - 0 = x$$

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

Properties

$$Dist(*, -, i)$$

$$Dist(*, +, i)$$

$$Dist(\cup, \cap, s)$$

$$Neut(+, 0, i)$$

$$Neut(-, 0, i)$$

$$P(\exp, +, *, i, r)$$

Creating conjectures from analogies

Theorems (CNF,AC)	Properties	Substitutions
$x * (y - z) = x * y - x * z$	$Dist(*, -, i)$	$\{- \leftrightarrow +\}$
$x * (y + z) = x * y + x * z$	$Dist(*, +, i)$	$\{* \leftrightarrow \cup, + \leftrightarrow \cap, i \leftrightarrow s\}$
$x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$	$Dist(\cup, \cap, s)$	$\{* \leftrightarrow \cup, - \leftrightarrow \cap, i \leftrightarrow s\}$
$x + 0 = x$	$Neut(+, 0, i)$	$\{- \leftrightarrow +\}$
$x - 0 = x$	$Neut(-, 0, i)$	
$exp(a + b) = exp(a) * exp(b)$	$P(exp, +, *, i, r)$	

Creating conjectures from analogies

Theorems (CNF,AC)	Properties	Substitutions	Score
$x * (y - z) = x * y - x * z$	$Dist(*, -, i)$	$\{- \leftrightarrow +\}$	1/3
$x * (y + z) = x * y + x * z$	$Dist(*, +, i)$	$\{ * \leftrightarrow \cup, + \leftrightarrow \cap, i \leftrightarrow s \}$	1/3
$x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$	$Dist(\cup, \cap, s)$	$\{ * \leftrightarrow \cup, - \leftrightarrow \cap, i \leftrightarrow s \}$	1/3
$x + 0 = x$	$Neut(+, 0, i)$	$\{- \leftrightarrow +\}$	1/2
$x - 0 = x$	$Neut(-, 0, i)$		
$exp(a + b) = exp(a) * exp(b)$	$P(exp, +, *, i, r)$		

Creating conjectures from analogies

Theorems (CNF,AC)	Properties	Substitutions	Score
$x * (y - z) = x * y - x * z$	$Dist(*, -, i)$	$\{- \leftrightarrow +\}$	1/3
$x * (y + z) = x * y + x * z$	$Dist(*, +, i)$	$\{ * \leftrightarrow \cup, + \leftrightarrow \cap, i \leftrightarrow s \}$	1/3
$x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$	$Dist(\cup, \cap, s)$	$\{ * \leftrightarrow \cup, - \leftrightarrow \cap, i \leftrightarrow s \}$	1/3
$x + 0 = x$	$Neut(+, 0, i)$	$\{- \leftrightarrow +\}$	1/2
$x - 0 = x$	$Neut(-, 0, i)$		
$exp(a + b) = exp(a) * exp(b)$	$P(exp, +, *, i, r)$		

Conjectures:

- $exp(a - b) = exp(a) * exp(b)$ (1/2 + 1/3).
- $exp(a \cap b) = exp(a) \cup exp(b)$ (1/3).

Example of a proved conjecture

- Analogy: *continuous* \rightarrow *lipschitzian*.
- Number of conjectures: 75
- Number of non-trivial theorems and lemmas: 6

Challenges and limitations

How do we find interesting conjectures?

	Our solution	Limitation	Improvement?
Generation	substitutions	search space	probabilistic grammar
Learning	analogies	specific relation	genetic algorithm
Pruning	scoring function	only syntactic	model-based guidance

How do we filter interesting proved conjectures?

- true : running first-order prover Eprover
- non-trivial: number of dependencies
- interesting: helping to reprove the library

One more choice?

- theory exploration vs goal-oriented lemmas generation