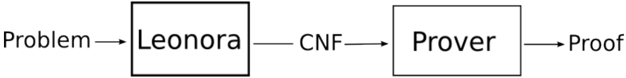
Effective Normalization Techniques for HOL[1]

Max Wisniewski Alexander Steen Kim Kern Christoph Benzmüller
Freie Universität Berlin

International Joint Conference on Automated Reasoning

[1] This work has been supported by the DFG under grant BE 2501/11-1 (Leo-III).

Leonora



- Transforms problem into a clause normal form.
- Can be used as a library
- and as a stand-alone tool.

Difference between HOL and FOL

1. Quantification over arbitrary types

$$\forall F. F \text{ monotonic} \longleftrightarrow (\forall x. \forall y. x < y \rightarrow F\,x \leq F\,y)$$

2. Embedded formulae

$$\forall x. \forall y. p(x \vee y)$$

Difference between HOL and FOL

1. Quantification over arbitrary types

$$\forall F.\, F \text{ monotonic} \longleftrightarrow (\forall x. \forall y.\, x < y \rightarrow F\,x \leq F\,y)$$

2. Embedded formulae

$$\forall x. \forall y. p(x \vee y)$$

Difference between HOL and FOL

1. Quantification over arbitrary types

$$\forall F.\, F \text{ monotonic} \longleftrightarrow (\forall x. \forall y.\, x < y \rightarrow F\,x \leq F\,y)$$

2. Embedded formulae

$$\forall x. \forall y. p(x \vee y)$$

Difference between HOL and FOL

1. Quantification over arbitrary types

$$\forall F.\, F \text{ monotonic} \longleftrightarrow (\forall x. \forall y.\, x < y \rightarrow F\,x \leq F\,y)$$

2. **Embedded formulae**

$$\forall x. \forall y. p(x \vee y)$$

Freie Universität Berlin

Clause Normal Form

A clause is in *clause normal form* if it has the following form:

$$\forall x_1 \ldots \forall x_n . (l_{1,1} \vee \ldots l_{1,m_1}) \wedge \ldots \wedge (l_{k,1} \vee \ldots \vee l_{k,m_k})$$

where $l_{i,j}$ are literals.

Clause Normal Form

A clause is in *clause normal form* if it has the following form:

$$\forall x_1 \dots \forall x_n . (l_{1,1} \vee \dots l_{1,m_1}) \wedge \dots \wedge (l_{k,1} \vee \dots \vee l_{k,m_k})$$

where $l_{i,j}$ are literals.

- simplification
- skolemization
- prenex normal form
- clausification

Normalization

Clause Normal Form

A clause is in *clause normal form* if it has the following form:

$$\forall x_1 \ldots \forall x_n.(l_{1,1} \vee \ldots l_{1,m_1}) \wedge \ldots \wedge (l_{k,1} \vee \ldots \vee l_{k,m_k})$$

where $l_{i,j}$ are literals.

- simplification
- **skolemization**
- **prenex normal form**
- **clausification**

Freie Universität Berlin

Elimination of Existential Quantifiers

$$\forall A\,.\,\exists B\,.\,\forall C\,.\,\exists D\,.\,(A \vee B) \wedge (C \vee D)$$

Freie Universität Berlin

Elimination of Existential Quantifiers

$$\forall A\,.\,\exists B\,.\,\forall C\,.\,\exists D\,.\,(A \vee B) \wedge (C \vee D)$$

can be transformed to

$$\forall A\,.\,\forall C\,.\,(A \vee sk_B(A)) \wedge (C \vee sk_D(A, C))$$

Freie Universität Berlin

With embedded formulas

$$\exists A.\, p(A) \lor q(\exists B.p(B))$$

Skolemization

With embedded formulas

$$\exists A\,.\,p(A) \vee q(\exists B.p(B))$$

can be transformed to

$$p(sk_A) \vee q(\exists B.p(B))$$

With embedded formulas

$$\exists A\,.\,p(A) \lor q(\exists B.p(B))$$

can be transformed to

$$p(sk_A) \lor q(\exists B.p(B))$$

Is $\exists B.p(B)$ an existential or universal quantified formula?

Moving Quantifiers to the front

$$(\forall A.\, A \lor \forall B.\, B) \land (\forall C.\forall D.C \lor \neg D)$$

Moving Quantifiers to the front

$$(\forall A. A \vee \forall B. B) \wedge (\forall C. \forall D. C \vee \neg D)$$

can be transformed to

$$\forall A, B, C, D. (A \vee B) \wedge (C \vee \neg D)$$

Problems in Higher-Order

With embedded formulas

$$(\forall X.a(X)) \land \forall Y.p((\forall X.b(Y)) \to a(X))$$

Freie Universität Berlin

With embedded formulas

$$(\forall X. a(X)) \land \forall Y. p((\forall X. b(Y)) \to a(X))$$

normalizes to

$$\forall X. \forall Y. (a(X) \land p(\forall X. (b(Y) \to a(X))).$$

Problems in Higher-Order

With embedded formulas

$$(\forall X.a(X)) \land \forall Y.p((\forall X.b(Y)) \to a(X))$$

normalizes to

$$\forall X.\forall Y.(a(X) \land p(\forall X.(b(Y) \to a(X))).$$

We cannot move the quantifiers to the front!

Formula Renaming

$$(a \wedge b \wedge c) \vee (d \wedge e \wedge f)$$

standard clausification

$$(a \vee d), (a \vee e), \ldots, (c \vee f) \quad \text{(9 cases)}$$

Formula Renaming

$$(a \wedge b \wedge c) \vee (d \wedge e \wedge f)$$

standard clausification

$$(a \vee d), (a \vee e), \ldots, (c \vee f) \quad \text{(9 cases)}$$

$$(a \wedge b \wedge c) \vee r \quad , \quad \neg r \vee (d \wedge e \wedge f)$$

Formula Renaming

$$(a \wedge b \wedge c) \vee (d \wedge e \wedge f)$$

standard clausification

$$(a \vee d), (a \vee e), \ldots, (c \vee f) \quad \text{(9 cases)}$$

$$(a \vee r), (b \vee r), (c \vee r), (\neg r \vee d), (\neg r \vee e), (\neg r \vee f) \quad \text{(6 cases)}$$

Argument Extraction

Idea: Replace all complex formula arguments.

Idea: Replace all complex formula arguments.

Argument Extraction

Let Φ be a formula with $f(p)$ occurring as subterm. If

$$\begin{array}{l} p \text{ headsymbol is a Boolean connective} \\ \{X_1, \ldots, X_n\} = free(p). \end{array}$$

we introduce a new function symbol s and return

$$\begin{array}{l} \Phi[p \setminus s(X_1, \ldots, X_n)] \\ \forall X_1 \ldots X_n \, . \, p = s(X_1, \ldots, X_n) \end{array}$$

Argument Extraction

Returning to our example:

$$\forall X.\forall Y.(a(X) \wedge p(\forall X.b(Y) \rightarrow a(X))).$$

Returning to our example:

$$\forall X. \forall Y. (a(X) \wedge p(\forall X. b(Y) \rightarrow a(X))).$$

normalizes to

$$\begin{array}{ll} & \forall Y \,.\, s_1(Y) \leftrightarrow \forall X \,.\, (b(Y) \rightarrow a(X)) \\ \wedge & \forall X \,.\, \forall Y \,.\, ((a(X) \wedge p(s_1(Y)))). \end{array}$$

Extended Prenex Normal Form

With extensionality

$$\forall X.\forall Y.((a(X) \wedge p(\forall X.b(Y) \to a(X))).$$

normalizes to

$$\begin{array}{ll} & \forall Y.\forall X.\neg s_1(Y) \vee (\neg b(Y) \vee a(X)) \\ \wedge & \forall Y.s_1(Y) \vee (b(Y) \wedge \neg a(sk_1(Y))) \\ \wedge & \forall X.\forall Y.((a(X) \wedge p(s_1(Y))). \end{array}$$

Results

Does this really help?

- Tested 537 Problems from 8 TPTP domains.
- Overall best improvment in CSR (common sense reasoning $6\times$).
- Worst domain LCL still proved three additional theorems.
- Solved the 1.0-rated problem PUZ145ˆ1 (previously not solved).

Overall

	AGT (23 Prob.)			CSR (123 Prob.)			LCL (139 Prob.)		
	Org	N_1	N_2	Org	N_1	N_2	Org	N_1	N_2
LEO-II									
Solved	23	23	23	61	69	45	104	93	104
–THM	23	23	23	**57**	**65**	41	**99**	**88**	99
Σ [s]	4.9	4.8	4.6	417	98	200	22.2	13.1	22.2
Avg. [s]	0.2	0.2	0.2	**6.8**	**1.4**	4.4	0.2	0.1	0.2
Satallax									
Solved	18	19	19	58	78	51	113	114	112
–THM	**18**	**19**	19	**54**	**74**	47	104	105	103
Σ [s]	97	157	157	290	52	252	267	361	276
Avg. [s]	**5.4**	**8.3**	8.3	**5**	**0.7**	4.9	2.4	3.2	2.5

Best

Problem	Time [s]		
	Orig.	N_1	N_2
CSR153^2	38.254	0.054	†
CSR138^1	9.858	0.029	9.875
CSR153^1	5.492	0.038	5.493
CSR126^2	31.425	0.676	31.451
CSR139^1	10.022	0.266	10.027
CSR137^2	1.351	0.039	4.270
CSR134^1	9.713	0.338	0.359
CSR122^2	19.307	0.683	19.266
CSR143^2	2.714	0.238	†
CSR153^3	7.743	0.988	†
CSR119^3	26.588	3.672	†
CSR120^3	26.609	3.678	†
CSR137^1	0.242	0.042	0.246
CSR152^3	14.960	3.671	†
CSR151^3	14.939	3.668	27.075

(a) Satallax

Problem	Time [s]		
	Orig.	N_1	N_2
CSR139^2	5.331	0.146	5.633
CSR132^2	5.331	0.283	†
CSR139^1	1.196	0.055	1.142
CSR150^1	1.633	0.091	1.604
CSR141^2	1.229	0.202	†
CSR148^1	0.295	0.057	†
CSR149^2	1.002	0.194	1.479
CSR123^2	0.930	0.192	†
CSR124^2	0.731	0.190	†
CSR122^2	0.725	0.193	†
CSR125^2	0.757	0.327	†
CSR119^2	0.355	0.173	†
CSR138^1	0.104	0.054	0.122
CSR120^2	0.388	0.202	†
CSR127^2	0.342	0.190	0.529

(b) LEO-II

Summary

Further Work

- Replacement of defined Equalities (Leibniz, Andrews).
- Rule Detection:
  - Induction.
  - Choice Functions.
- More tests.

Summary

- Adaptation of FOL Normalization to HOL
- Argument Extraction for full clause normalforms
- Implementation shows promising results
- Many open topics to improve on

Summary

Further Work

- Replacement of defined Equalities (Leibniz, Andrews).
- Rule Detection:
  - Induction.
  - Choice Functions.
- More tests.

Summary

- Adaptation of FOL Normalization to HOL
- Argument Extraction for full clause normalforms
- Implementation shows promising results
- Many open topics to improve on

Thank you