Extracting expansion trees from resolution proofs with splitting and definitions

Gabriel Ebner

2018-01-17

TU Wien

Herbrand disjunctions

Expansion proofs

Resolution proofs

Extraction

Splitting

Subformula definitions

Empirical evaluation

Herbrand's theorem

- Classical first-order logic

Theorem (special case of Herbrand 1930)
Let $\varphi(x)$ be a quantifier-free first-order formula.

Then $\exists x\, \varphi(x)$ is valid iff there exist terms $t_1, \dots, t_n$ such that $\varphi(t_1) \vee \cdots \vee \varphi(t_n)$ is a tautology.

- reduces first-order validity (undecidable) to propositional validity (coNP-complete)

Herbrand's theorem

- Classical first-order logic with equality

Theorem (special case of Herbrand 1930)
Let $\varphi(x)$ be a quantifier-free first-order formula.

Then $\exists x\, \varphi(x)$ is valid iff there exist terms $t_1, \ldots, t_n$ such that $\varphi(t_1) \vee \cdots \vee \varphi(t_n)$ is a quasi-tautology.

- reduces first-order validity (undecidable) to propositional validity modulo equality (coNP-complete)
  - (QF_UF in SMT solvers)

Running example

Drinker's formula: $\exists x \forall y\, (Px \rightarrow Py)$

After Skolemization: $\exists x (Px \rightarrow P(sx))$

Herbrand disjunction: $(Px \rightarrow P(sx)) \vee (P(sx) \rightarrow P(s(sx)))$
$(t_1 = x, t_2 = s(x))$

Herbrand's theorem—why it is awesome

- fundamental result
- bounds on proof complexity
- answer substitutions
    - logic programming
    - typeclass inference
    - synthesis
- computational content of classical logic
- cut-introduction
- automated inductive theorem proving

Implementation

- Essential for cut-introduction and inductive proving
- Also used in expansion-proof-based CERES (Lolic)

- Reliable
- Fast
- Supports every prover we can get our hands on

- Default algorithm since GAPT 2.2 (July 2016)
    - "General Architecture for Proof Theory"
    - `https://logic.at/gapt`
    - `https://github.com/gapt/gapt`

Herbrand disjunctions

Expansion proofs

Resolution proofs

Extraction

Splitting

Subformula definitions

Empirical evaluation

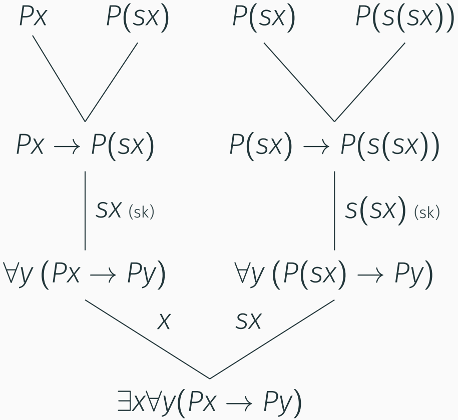
Expansion trees

- “Herbrand-disjunctions for non-prenex formulas”
- higher-order as well
- Miller 1987

- reduces first-order validity (undecidable) to propositional validity modulo equality (coNP-complete)

Expansion trees

- deep formula, here $(Px \to P(sx)) \vee (P(sx) \to P(s(sx)))$
- deep formula tautological $\Rightarrow$ shallow formula valid
- shallow formula $\longrightarrow$



Expansion proofs

- Expansion proof of φ =
  expansion tree with shallow formula φ
  - deep formula quasi-tautological
  - dependency relation acyclic

Expansion proofs

- Expansion proof of $\underline{\Gamma \vdash \Delta}$ =
  expansion sequent with shallow sequent $\Gamma \vdash \Delta$
  - deep sequent quasi-tautological
  - dependency relation acyclic

Herbrand disjunctions

Expansion proofs

Resolution proofs

Extraction

Splitting

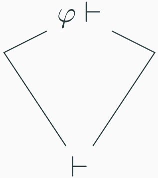
Subformula definitions

Empirical evaluation

Resolution calculus

- Integrated clausification inferences
- Like higher-order calculi (Andrews 1971)

- Refutational proofs:



- Upper part clausification, lower part clausal refutation

- Clausification rules unary
- Resolution (essentially) only binary rule

Resolution inferences

Some rules:

$$\frac{\Gamma \vdash \Delta, \forall x\, \varphi(x, \bar{y})}{\Gamma \vdash \Delta, \varphi(x, \bar{y})} \forall_r \qquad \frac{\forall x\, \varphi(x, \bar{y}), \Gamma \vdash \Delta}{\varphi(s_\varphi(\bar{y}), \bar{y}), \Gamma \vdash \Delta} \forall_l$$

$$\frac{\Gamma \vdash \Delta, \varphi \wedge \psi}{\Gamma \vdash \Delta, \varphi} \wedge_r^1 \qquad \frac{\Gamma \vdash \Delta, \varphi \wedge \psi}{\Gamma \vdash \Delta, \psi} \wedge_r^2 \qquad \frac{\varphi \wedge \psi, \Gamma \vdash \Delta}{\varphi, \psi, \Gamma \vdash \Delta} \wedge_l$$

$$\frac{\Gamma \vdash \Delta}{\Gamma\sigma \vdash \Delta\sigma} \text{subst} \qquad \frac{\Gamma \vdash \Delta, \varphi \qquad \varphi, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda} \text{res} \quad (\varphi \text{ atom})$$

Running example

$$\dfrac{\dfrac{\dfrac{\exists x \forall y (Px \to Py) \vdash}{\forall y (Px \to Py) \vdash}\ \exists_l}{Px \to P(sx) \vdash}\ \forall_l}{\vdash Px}\ \to_l^1 \qquad \dfrac{\dfrac{\dfrac{\exists x \forall y (Px \to Py) \vdash}{\forall y (Px \to Py) \vdash}\ \exists_l}{Px \to P(sx) \vdash}\ \forall_l}{P(sx) \vdash}\ \to_l^2$$

Running example

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{\exists x \forall y (Px \to Py) \vdash}{\forall y (Px \to Py) \vdash}\ \exists_l
      }{Px \to P(sx) \vdash}\ \forall_l
    }{\vdash Px}\ \to_l^1
  }{\vdash P(sx)}\ \text{subst}
  \qquad
  \dfrac{
    \dfrac{
      \dfrac{\exists x \forall y (Px \to Py) \vdash}{\forall y (Px \to Py) \vdash}\ \exists_l
    }{Px \to P(sx) \vdash}\ \forall_l
  }{P(sx) \vdash}\ \to_l^2
}{\vdash}\ \text{res}
$$

Herbrand disjunctions

Expansion proofs

Resolution proofs

Extraction

Splitting

Subformula definitions

Empirical evaluation

Extraction

- Expansion trees propagate *upwards*
  from the empty clause to the input formula.

- At every moment, each subproof has
  a finite set of expansion sequents
- Move up one inference at a time

- No DAG$\to$tree conversion
    - (still exponential worst case)

Extraction

- *Extraction state* $\mathcal{P}$ is set of pairs $(\pi'; \mathcal{E})$
  - subproof π' ending in $\mathcal{T}$
  - expansion sequent $\mathcal{E}$
  - $\exists\sigma\ (\mathcal{T}\sigma = \mathrm{sh}(\mathcal{E}))$
- *Initial state* is $\{(\pi; \vdash)\}$
- $\{(\pi; \vdash)\} \quad \rightsquigarrow^* \quad \{(\pi_0; E \vdash)\}$
  - π_0 is the $\varphi \vdash$ leaf
  - E is the resulting expansion proof

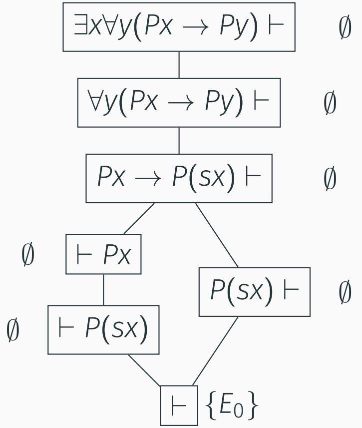
Extraction steps

One case:

$$\mathcal{P} \cup \{(\dfrac{\begin{matrix}(\pi)\\ \Gamma \vdash \Delta, \forall x\, \varphi(x)\end{matrix}}{\Gamma \vdash \Delta, \varphi(x)}; E_\Gamma \vdash E_\Delta, E_\varphi)\} \rightsquigarrow \mathcal{P} \cup \{(\pi; E_\Gamma \vdash E_\Delta, \begin{matrix} E_\varphi \\ \mid x\sigma \\ \forall x\, \varphi(x)\end{matrix})\}$$

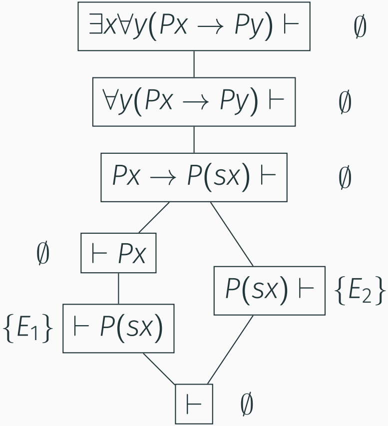
- Depends only on last inference of subproof
- Expansion tree is not traversed
- Deep formula stays quasi-tautology

Extraction example



$E_0 = \vdash$

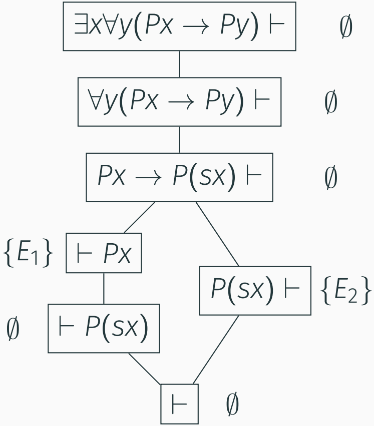
Extraction example



$$E_1 = \vdash P(sx)$$

$$E_2 = P(sx) \vdash$$

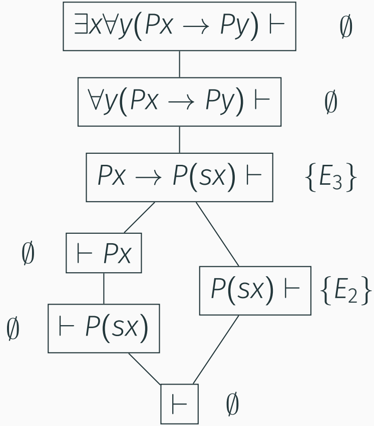
Extraction example



$$E_1 = \vdash P(sx)$$

$$E_2 = P(sx) \vdash$$

Extraction example

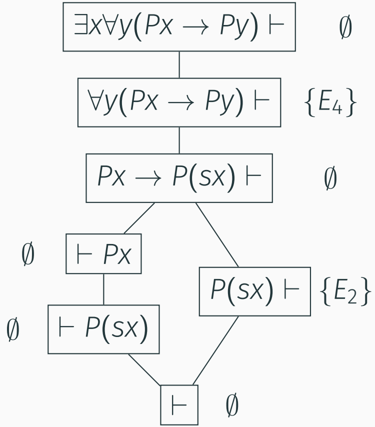


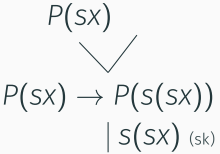
$P(sx)$

$$E_3 = \; P(sx) \to P(s(sx)) \;\; \vdash$$

$$E_2 = P(sx) \vdash$$

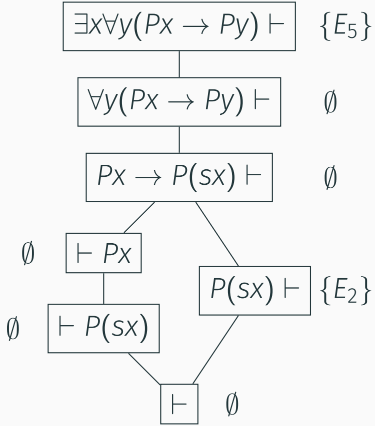
Extraction example

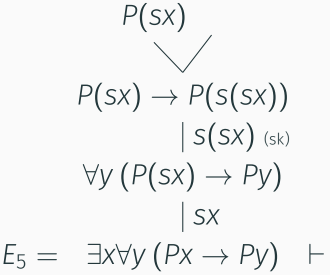




$$E_4 = \quad \forall y\,(P(sx) \to Py) \quad \vdash$$

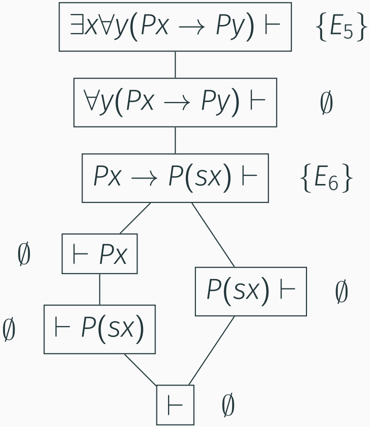
$$E_2 = P(sx) \vdash$$

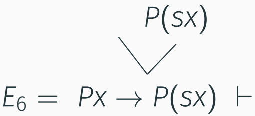

$\exists x \forall y (Px \to Py) \vdash$
$\{E_5\}$
$\forall y (Px \to Py) \vdash$
$\emptyset$
$Px \to P(sx) \vdash$
$\emptyset$
$\emptyset$
$\vdash Px$
$P(sx) \vdash$
$\{E_2\}$
$\emptyset$
$\vdash P(sx)$
$\vdash$
$\emptyset$


$P(sx)$
$P(sx) \to P(s(sx))$
$s(sx)$ (sk)
$\forall y\,(P(sx) \to Py)$
sx
$E_5 = \quad \exists x \forall y\,(Px \to Py) \quad \vdash$

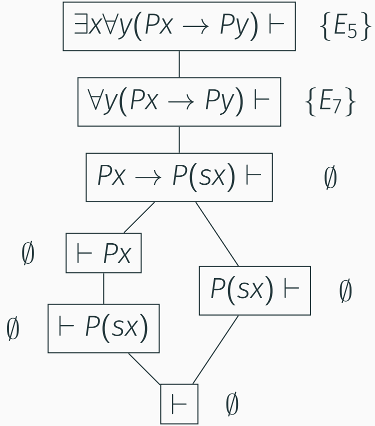
$$E_2 = P(sx) \vdash$$

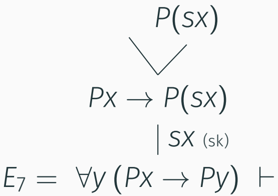
Extraction example



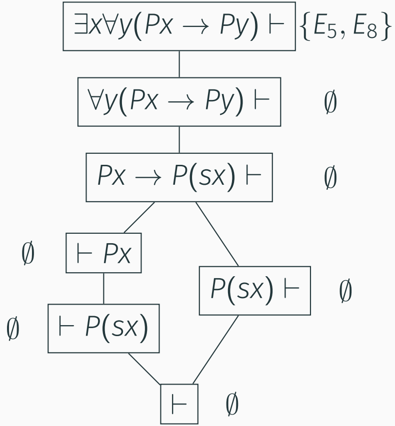


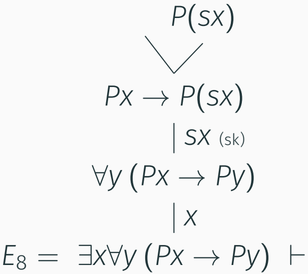
Extraction example



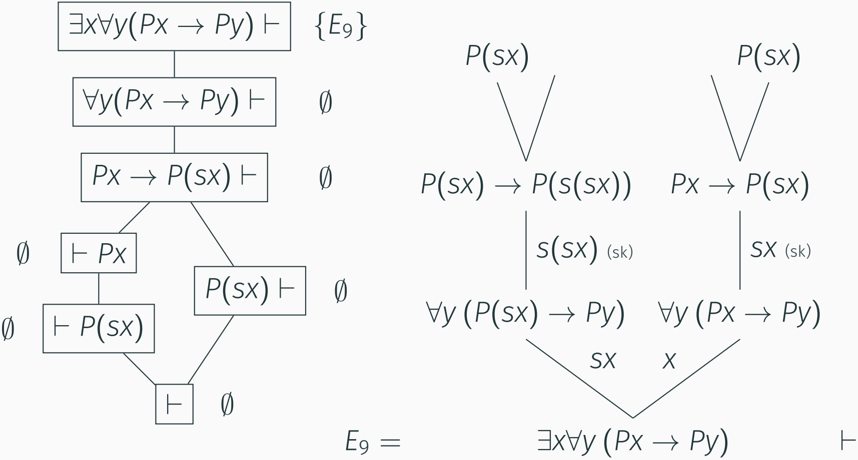


Extraction example





Extraction example



Extensions

1. New inferences in resolution calculus
2. New types of nodes in expansion trees
3. Expansion trees in antecedent of expansion proof
4. Post-processing steps for elimination

Herbrand disjunctions

Expansion proofs

Resolution proofs

Extraction

Splitting

Subformula definitions

Empirical evaluation

Splitting in provers

- Very effective inference rule in first-order provers (Vampire, SPASS, Escargot, …)
- $\vdash Px, Qy$ is equivalent to $(\forall x\, Px) \vee (\forall y\, Qy)$
    - free variables are disjoint
- do case distinction
    - in one branch: $\vdash Px$
    - in the other: $\vdash Qy$

$\rightarrow$ much smaller clauses

Splitting in resolution calculus

$$\frac{\vdash Px, Qy}{\vdash \mathrm{spl}_1, \mathrm{spl}_2} \qquad \frac{}{\mathrm{spl}_1 \vdash Px} \qquad \frac{}{\mathrm{spl}_2 \vdash Qy}$$

$$\mathrm{spl}_1 =_{\mathrm{def}} \forall x\, Px \qquad \mathrm{spl}_2 =_{\mathrm{def}} \forall y\, Qy$$

- Vampire sends $\mathrm{spl}_1 \vee \mathrm{spl}_2$ to a SAT-solver (“Avatar”)
- SPASS does a manual case distinction

Expansion proofs with cut

- Cut node: $\dfrac{E_1 \quad E_2}{\mathrm{Cut}_\psi}$ (actually: $\dfrac{\dfrac{E_1 \quad E_2}{\psi \to \psi}}{\forall X\,(X \to X)}\ \psi$)
    - where $\mathrm{sh}(E_1) = \mathrm{sh}(E_2) = \psi$
    - just like cuts in LK
- Expansion proofs with cuts = expansion proofs of

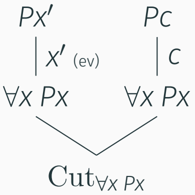
$$\mathrm{Cut}_{\psi_1}, \ldots, \mathrm{Cut}_{\psi_n} \vdash \varphi$$

- Cut-elimination (Hetzl, Weller 2013)
    - like ε-calculus

Splitting in expansion proofs

$$\frac{\vdash Px', Qy}{\vdash \mathrm{spl}_1, \mathrm{spl}_2} \qquad \frac{\overline{\mathrm{spl}_1 \vdash Px}}{\mathrm{spl}_1 \vdash Pc}$$

$\rightarrow$ results in the cut



Herbrand disjunctions

Expansion proofs

Resolution proofs

Extraction

Splitting

Subformula definitions

Empirical evaluation

Subformula definitions

- Exponential number of clauses:

$$\vdash P_1 \wedge Q_1, \ldots, P_n \wedge Q_n$$

- Solution: introduce definitions

$$\dfrac{\dfrac{\dfrac{\vdash P_1 \wedge Q_1, P_2 \wedge Q_2, \ldots, P_n \wedge Q_n}{\vdash D_1, P_2 \wedge Q_2, \ldots, P_n \wedge Q_n}\,\text{def}}{\vdots}\,\text{def}}{\vdash D_1, D_2, \ldots, D_n}\,\text{def} \qquad \dfrac{}{D_i \vdash P_i \wedge Q_i}$$

- only $D_i \vdash P_i \wedge Q_i$, *not* $P_i \wedge Q_i \vdash D_i$

Definitions in “expansion” proofs

- Definition nodes: $\begin{array}{c} D_\psi(\overline{x}) \\ \Big| \scriptstyle\text{(def)} \\ \psi(\overline{x}) \end{array}$ $\quad (D_\psi(\overline{x})$ atom)

- Expansion proof with definition(s) =
  expansion proof of $\forall \overline{x}\, (D_\psi(\overline{x}) \to \psi(\overline{x})) \vdash \varphi$

Definition elimination

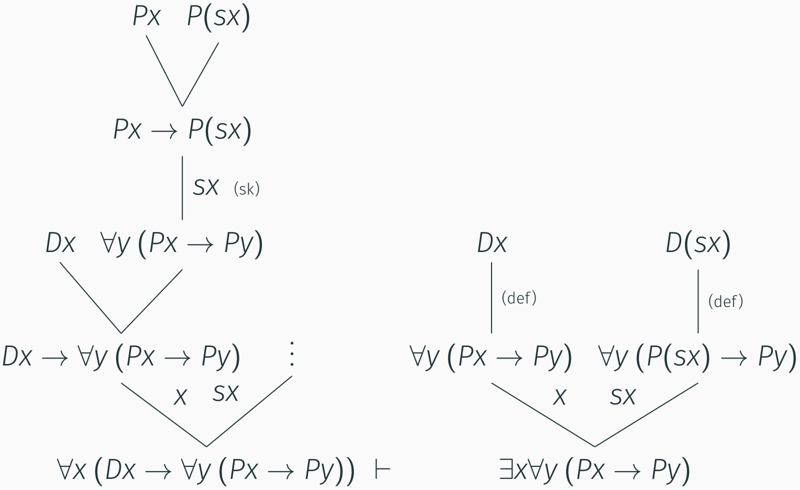
Input: expansion proof with definition

Output: expansion proof of $\vdash \varphi$

(without definition nodes for D_ψ)

$\rightsquigarrow$ just replace definition nodes by expansion of $\psi(\bar{x})$

Definition elimination



Definition elimination

$$Px \quad P(sx) \qquad P(sx) \quad P(s(sx))$$

$$Px \to P(sx) \qquad P(sx) \to P(s(sx))$$

$$sx \ \text{(sk)} \qquad s(sx) \ \text{(sk)}$$

$$\forall y\,(Px \to Py) \qquad \forall y\,(P(sx) \to Py)$$

$$x \qquad sx$$

$$\vdash \qquad \exists x \forall y\,(Px \to Py)$$

Herbrand disjunctions

Expansion proofs

Resolution proofs

Extraction

Splitting

Subformula definitions

Empirical evaluation

Clausification quality

- Resolution calculus induces clausification
- or rather class of possible clausifications
  - naive exponential with outer Skolemization
  - with subformula definitions and inner Skolemization
- Some techniques are not supported
  - $(\forall y\,(Py \lor Px)) \;\mapsto\; ((\forall y\,Py) \lor Px)$
  - sharing Skolem functions for propositionally equivalent formulas
- How good is the clausification allowed by the calculus?

Clausification quality

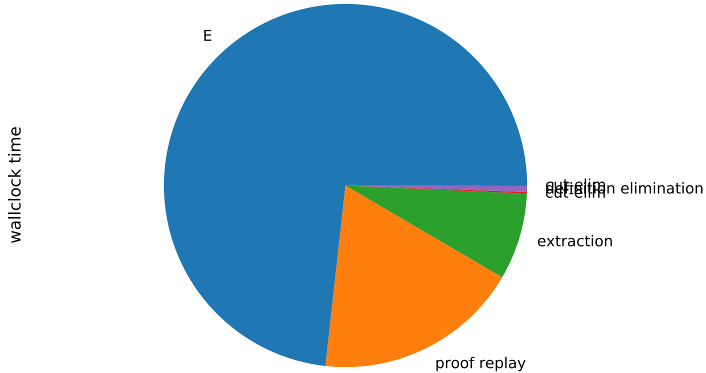
Evaluated as preprocessing for E

Proofs found (CASC-26 competition problems):

clausifier:	GAPT	E
default	89	81
`--auto-schedule`	285	308

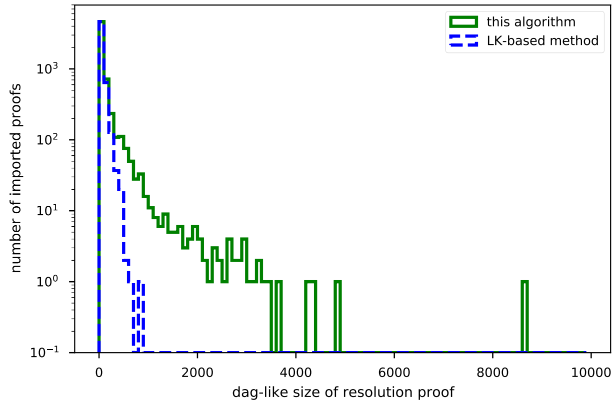
$\rightarrow$ competitive with state-of-the-art clausification

Relative runtime



(7 proofs could not be imported due to excessive runtime)

Extraction performance (compared to Hetzl et al. 2013)



- We can now extract expansion proofs from 96.6% of the Prover9 proofs in the TSTP (prev. 86.4%)

Conclusion

- Faster
- More features
  - Definitions for subformulas
  - Splitting

$\rightarrow$ supports all in-processing techniques (that we know of)

Future work:

- CERES with characteristic formula
- Complexity bounds
- Size of proofs with Avatar inferences?
- Higher-order logic

A note on Skolemization

Drinker's formula (alt.): $\exists x(Px \to \forall y\, Py)$

Inner Skolemization: $\exists x(Px \to Ps)$

Outer Skolemization: $\exists x(Px \to P(sx))$

Theorem (corollary of Aguilera, Baaz 2016)
In general, inner Skolemization has non-elementarily smaller cut-free proofs than outer Skolemization.

- Our extraction supports both.