Producing Skolem Expansion Trees with the CERES^ω method: A Case Study

Martin Riener, joint work with Alexander Leitsch

2017-05-26

Introduction

- Skolem Expansion Proofs:
  - Proof representation for Classical Higher-Order Logic (ETT)
  - Stores instantiations, no propositional reasoning
  - Compact alternative to LK
- LK_{sk}:
  - Cut-free higher-order sequent calculus with Skolem terms
  - Produced in the CERES^{ω} method:
    1. Proof formalization in LK
    2. Conversion to LK_{sk} + Cut (LK_{skc})
    3. Creation of Characteristic Sequent Set + Proof Projections
    4. Assembling of LK_{skc} proof in passive-cut normal-form
    5. Transformation to LK_{sk} by reductive method
    6. Repair locally unsound inferences and transfer to LK
- Goal: replace steps 5 and 6 by expansion proof extraction.
  This talk: step 6.

Overview

- Skolem Expansion Proofs
- LK_{sk}
- From LK_{sk} to Skolem Expansion Proofs
- Case Study: Infinite Pigeon Principle

What are Skolem Expansion Proofs?

- Generalization of Herbrand Disjunctions to Higher Order Logic
- Tree mirrors logical structure of the formula
- Weak Quantifier nodes store instantiations
- Strong Quantifier nodes store Skolem terms
  - Shallow formula: formula with quantifiers
  - Deep formula: formula with instantiations

Example

- Shallow formula:
  $\forall X \exists Y \forall i (X(i) \rightarrow Y(i+1)) \rightarrow \exists i \exists Z Z(((i+1)+1)+1)$
- Deep formula:

$$
\begin{aligned}
&(C(i) \rightarrow s_0(C, i+1)) \\
&\wedge (s_0(C, i+1) \rightarrow s_0(s_0(C), (i+1)+1)) \\
&\wedge (s_0(s_0(C), (i+1)+1) \rightarrow s_0(s_0(s_0(C)), ((i+1)+1)+1)) \\
&\rightarrow (C(i) \rightarrow s_0(s_0(s_0(C)), ((i+1)+1)+1))
\end{aligned}
$$

- Expansion Tree (blackboard)

Definition: Expansion Tree

- Atom Node $A(F)$: HOL Atom or weakly quantified formula
  $\mathrm{Sh}(A) = \mathrm{Dp}(A) = F$
- Logical Operator Node: $\neg T$, $T_1 \circ T_2$ with $\circ \in \{\wedge, \vee, \rightarrow\}$:
  $\mathrm{Sh}(\neg T) = \neg\mathrm{Sh}(T)$, $\mathrm{Dp}(\neg T) = \neg\mathrm{Dp}(T)$
  $\mathrm{Sh}(T_1 \circ T_2) = \mathrm{Sh}(T_1) \circ \mathrm{Sh}(T_2)$, $\mathrm{Dp}(T_1 \circ T_2) = \mathrm{Dp}(T_1) \circ \mathrm{Dp}(T_2)$
- Strong (Skolem) Quantifier Node: $QT +^s S$ with $Q \in \{\forall, \exists\}$:
  $\mathrm{Sh}(QT +^{s(t_1,\ldots,t_n)})S) = QT$,
  $\mathrm{Dp}(QT +^{s(t_1,\ldots,t_n)} S) = \mathrm{Dp}(S)$
- Weak Quantifier Node: $QT +^{t_1} T_1 + \ldots +^{t_n} T_n$:
  $\mathrm{Sh}(QT +^{t_1} T_1 + \ldots +^{t_n} T_n) = QT$
  $\mathrm{Dp}(\forall T +^{t_1} T_1 + \ldots +^{t_n} T_n) = \bigwedge_{i=1}^{n} \mathrm{Dp}(T_i)$
  $\mathrm{Dp}(\exists T +^{t_1} T_1 + \ldots +^{t_n} T_n) = \bigvee_{i=1}^{n} \mathrm{Dp}(T_i)$

Definition: Expansion Proof

- Each Skolem quantifier node introduces a unique Skolem function s.
- The path from the root to a Skolem quantifier node contains exactly p weak quantifier nodes with expansion terms t_1 to t_p (in that order).
- Groundedness: “no dangling weak quantifier leaves“
- Expansion Proof: Deep formula is valid

Relationships

- Theorem: Grounded EP convertible to grounded Skolem EP
- Theorem: Grounded Skolem EP convertible to grounded EP
- Theorem: EP convertible to LK
- Theorem: LK convertible to EP

What is LK_{sk}?

- Variant of Higher-Order Sequent Calculus LK
- Universal quantifiers inferred from Skolem terms
- Labels trace Skolem context
- No eigenvariable condition
- Not every derivation tree sound
- Quantifier inferences of weakly regular trees can be shifted into place

Rules

- Label: set of lambda terms
- Introduction Rule: $\langle F \rangle^{\ell_1} \vdash \langle F \rangle^{\ell_2}$
- Strong Quantifier:

$$\frac{\Gamma \vdash \Delta, \langle F(f S_1 \ldots S_n) \rangle^{\ell}}{\Gamma \vdash \Delta, \langle \forall_\alpha F \rangle^{\ell}} \forall^{sk}: r$$

  with $\ell = S_1, \ldots, S_n$ and f a Skolem symbol of appropriate type.
- Weak Quantifier:

$$\frac{\Gamma \vdash \Delta, \langle F T \rangle^{\ell, T}}{\Gamma \vdash \Delta, \langle \exists_\alpha F \rangle^{\ell}} \exists^{sk}: r$$

- Other Rules: Like LK, labels of auxiliary and primary formulas must agree

Global Conditions

- Properness: End-sequent labels are empty
- Weak regularity: If two strong quantifier rules have the same Skolem term, the inferences are homomorphic (“eventually contracted”)

Example

$$\dfrac{s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(C),((k+1)+1)) \qquad s_0(s_0(s_0(C)),(((k+1)+1)+1)) \vdash s_0(s_0(s_0(C)),(((k+1)+1)+1))}{s_0(s_0(C),((k+1)+1)) \to s_0(s_0(s_0(C)),(((k+1)+1)+1)), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \rightarrow: l$$

$$\dfrac{s_0(s_0(C),((k+1)+1)) \to s_0(s_0(s_0(C)),(((k+1)+1)+1)), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall z(s_0(s_0(C),z) \to s_0(s_0(s_0(C)),(z+1))), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \forall: l$$

$$\dfrac{\forall z(s_0(s_0(C),z) \to s_0(s_0(s_0(C)),(z+1))), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\exists Y \forall z(s_0(s_0(C),z) \to Y((z+1))), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \exists^{sk}: l$$

$$\dfrac{\exists Y \forall z(s_0(s_0(C),z) \to Y((z+1))), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall X \exists Y \forall z(X(z) \to Y((z+1))), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \forall: l$$

$$\dfrac{s_0(C,(k+1)) \vdash s_0(C,(k+1)) \qquad \forall X \exists Y \forall z(X(z) \to Y((z+1))), s_0(s_0(C),((k+1)+1)) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{s_0(C,(k+1)) \to s_0(s_0(C),((k+1)+1)), s_0(C,(k+1)), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \rightarrow: l$$

$$\dfrac{s_0(C,(k+1)) \to s_0(s_0(C),((k+1)+1)), s_0(C,(k+1)), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall z(s_0(C,z) \to s_0(s_0(C),(z+1))), s_0(C,(k+1)), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \forall: l$$

$$\dfrac{\forall z(s_0(C,z) \to s_0(s_0(C),(z+1))), s_0(C,(k+1)), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\exists Y \forall z(s_0(C,z) \to Y((z+1))), s_0(C,(k+1)), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \exists^{sk}: l$$

$$\dfrac{C(k) \vdash C(k) \qquad \exists Y \forall z(s_0(C,z) \to Y((z+1))), s_0(C,(k+1)), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{C(k) \to s_0(C,(k+1)), C(k), \exists Y \forall z(s_0(C,z) \to Y((z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \rightarrow: l$$

$$\dfrac{C(k) \to s_0(C,(k+1)), C(k), \exists Y \forall z(s_0(C,z) \to Y((z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \to s_0(C,(k+1)), C(k), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \forall: l$$

$$\dfrac{\forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \to s_0(C,(k+1)), C(k), \forall X \exists Y \forall z(X(z) \to Y((z+1))) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \to s_0(C,(k+1)), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} c: l$$

$$\dfrac{\forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \to s_0(C,(k+1)), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall z(C(z) \to s_0(C,(z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \forall: l$$

$$\dfrac{\forall z(C(z) \to s_0(C,(z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\exists Y \forall z(C(z) \to Y((z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \exists^{sk}: l$$

$$\dfrac{\exists Y \forall z(C(z) \to Y((z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall X \exists Y \forall z(X(z) \to Y((z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} \forall: l$$

$$\dfrac{\forall X \exists Y \forall z(X(z) \to Y((z+1))), \forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)}{\forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)} c: l$$

(π)

$$\dfrac{\dfrac{\begin{matrix}(\pi)\\ \forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash s_0(s_0(s_0(C)),((k+1)+1)+1)\end{matrix}}{\forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash \exists z s_0(s_0(s_0(C)),((z+1)+1)+1)} \exists^{sk}: r}{\forall X \exists Y \forall z(X(z) \to Y((z+1))), C(k) \vdash \exists X \exists z X(((z+1)+1)+1)} \exists^{sk}: r$$

Homomorphic pruning, Independent rule shifting

- ρ independent from σ: primary formula of σ not ancestor of auxiliary formulas of ρ
- Rewrite systems for permuting independent inferences:
  - Contractions: $\rhd_c$
  - Unary Rules: $\rhd_u$
  - Binary Rules: $\rhd_b$
- Idea: permute strong quantifiers downwards until eigenvariable condition would be fulfilled

Problematic rule: shift binary over contraction

$$\dfrac{\dfrac{\dfrac{\overset{(\pi_1)}{\Pi, \Gamma_1, F_1, G_1 \vdash \Delta_1, \Lambda} \qquad \overset{(\pi_2)}{\Pi, \Gamma_2, F_2, G_1 \vdash \Delta_2, \Lambda}}{\Pi, \Pi, \Gamma_1, \Gamma_2, F_1 \vee F_2, G_1, G_1 \vdash \Delta_1, \Delta_2, \Lambda, \Lambda}\rho}{\Pi, \Gamma_1, \Gamma_2, F_1 \vee F_2, G_1 \vdash \Delta_1, \Delta_2, \Lambda}c:* \qquad \overset{(\pi_3)}{G_2, \Gamma_3 \vdash \Delta_3}}{\Pi, \Gamma_1, \Gamma_2, \Gamma_3, F_1 \vee F_2, G_1 \vee G_2 \vdash \Delta_1, \Delta_2, \Delta_3, \Lambda}\sigma$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\overset{(\pi_1)}{\Pi, \Gamma_1, F_1, G_1 \vdash \Delta_1, \Lambda} \qquad \overset{(\pi_3)}{G_2, \Gamma_3 \vdash \Delta_3}}{\Pi, \Gamma_1, \Gamma_3, F_1, G_1 \vee G_2 \vdash \Delta_1, \Delta_3, \Lambda}\sigma \qquad \dfrac{\overset{(\pi_2)}{\Pi, \Gamma_2, F_2, G_1 \vdash \Delta_2, \Lambda} \qquad \overset{(\pi_3)}{G_2, \Gamma_3 \vdash \Delta_3}}{\Pi, \Gamma_1, \Gamma_3, F_2, G_1 \vee G_2 \vdash \Delta_1, \Delta_3, \Lambda}\sigma}{\Pi, \Pi, \Gamma_1, \Gamma_2, \Gamma_3, F_1 \vee F_2, G_1 \vee G_2 \vdash \Delta_1, \Delta_2, \Delta_3, \Lambda, \Lambda}\rho}{\Pi, \Gamma_1, \Gamma_2, \Gamma_3, F_1 \vee F_2, G_1 \vdash \Delta_1, \Delta_2, \Delta_3, \Lambda}c:*$$

- Solution Sequential Pruning: permute contractions down, “zip up”

Expansion Proof Extraction from LK_{sk}

- Axiom: translated to Atom Node
- Logical Rules: translated to corresponding operator nodes
- Contraction: $merge(C_1, C_2)$.

  Precondition: $\mathrm{Sh}(C_1) = \mathrm{Sh}(C_2)$

  - $\mathrm{mg}(A, A) := A$
  - $\mathrm{mg}(\neg A, \neg B) := \neg \mathrm{mg}(A, B)$
  - $\mathrm{mg}(A \circ B, C \circ D) := \mathrm{mg}(A, C) \circ \mathrm{mg}(B, D)$
  - $\mathrm{mg}(QF +^{s} A, F +^{s} B) := F +^{s} \mathrm{mg}(A, B)$
  - $\mathrm{mg}(QF +^{s_1} S_1 \ldots +^{s_n} S_n, QF +^{t_1} T_1 \ldots +^{t_m} T_m) :=$
    $QF +^{s_1} S_1 \ldots +^{s_n} S_n +^{t_1} T_1 \ldots +^{t_m} T_m$

Some Properties of Expansion Trees / merge

- $\mathrm{Dp}(A) \leftrightarrow \mathrm{Dp}(\mathrm{mg}(A, A))$
- $\mathrm{Dp}(A) \rightarrow \mathrm{Dp}(\mathrm{mg}(A, B))$
- $\mathrm{Dp}(\mathrm{mg}(A, B)) \rightarrow \mathrm{Dp}(A \vee B)$
- but not: $\mathrm{Dp}(A \vee B) \rightarrow \mathrm{Dp}(\mathrm{mg}(A, B))$

Relationship LK_{sk} and Skolem Expansion Proof

- Theorem: Skolem ET extracted from proper, weakly regular LK_{sk} proof always fulfills global soundness conditions
- Theorem: deep formula is valid
  - Idea: during ET extraction, validity of deep formula preserved under $\rhd_c, \rhd_u, \rhd_b$ and sequential pruning
  - Main Ingredients: independence and ET properties

Case Study: Infinite Pigeon Hole Principle

- Theory: SO Arithmetic + first order equality
- Given: function $f : Nat \mapsto \{0, 1\}$
  $\forall x.f(x) = 0 \vee f(x) = 1$
- Lemma: set $\{x|f(x) = s\}$ is unbounded
  $\forall x \exists y.x < y \wedge f(y) = s$
- Statement: there exists a monotonic function h enumerating n occurrences on f
  $\forall n \exists h.MON(h, n) \wedge \exists s.NOCC(h, n, s)$
- Monotonicity:
  $\forall i \forall j.i < j \wedge j < n + 1 \rightarrow h(i) < h(j)$
- n Occurrences:
  $\forall i : i < n + 1 \rightarrow f(h(i)) = s$

Expansion Proof

- too big to show completely, extracted instances
- witness terms for h in expansion of induction axiom, not conclusion

Source		*Term*
$BASE(0)$	$h(x) =$	$(s_{25}(q_1, s_{26}(q_1)) + 1) + s_9(q_2, s_{10}(q_2))$
$BASE(1)$	$h(x) =$	$(s_9(q_2, s_{10}(q_2)) + 1) + s_{25}(q_1, s_{26}(q_1))$
$STEP(0)$	$h(x) =$	if $\llcorner x < (s_{10}(q_2) + 1) \lrcorner$
		then $s_9(q_2, x)$
		else $(s_{25}(q_1, s_{26}(q_1)) + 1) + s_9(q_2, s_{10}(q_2))$
$STEP(1)$	$h(x) =$	if $\llcorner x < (s_{26}(q_1) + 1) \lrcorner$
		then $s_{25}(q_1, x)$
		else $(s_9(q_2, s_{10}(q_2)) + 1) + s_{25}(q_1, s_{26}(q_1))$

q_1, q_2: labels (goal instance of induction for $s = 1$ and $s = 0$)

- variation of input proof leads to same terms but different label q: $q(1) = q_1$ and $q(0) = q_2$

Lessons Learned, Future Work

- Automated higher-order provers (Leo II, Satallax) fail to reprove deep formula
  - Primary reason: treatment of first-order equality
  - Secondary reason: labels are huge, prover stuck in parsing without lambda-lifting
- Isabelle manages q_1, q_2 version (via encoding to SMT)
- Need better conditional: either improve encodings or built-in if-then-else (e.g. in zipperposition)
- Work on better integration of first-order equality in HO-ATPs (Matryoshka)

The End

Thanks for listening!