Skolemization, Cut-free proofs and Complexity

Daniel Weller

TU Vienna

September 11, 2011
Workshop ,,Epsilon Calculus and Constructivity"

The power of functions

- Setting of this talk: classical first-order logic.
- It is well-known that ,,quantifiers can be eliminated by introduction of fresh functions".
- Known as Skolemization, Herbrandization.

The power of functions

Note: for simplicity, consider only formulas in NNF.

Proposition

For every formula φ there exists a formula ψ that does not contain $\forall$, such that φ is valid iff ψ is.

- $\psi := \mathrm{sk}(\varphi)$ is obtained from φ by removing $\forall$ quantifiers and introducing fresh function symbols (Herbrand/Skolem functions).

The power of functions

Note: for simplicity, consider only formulas in NNF.

Proposition

For every formula φ there exists a formula ψ that does not contain $\forall$, such that φ is valid iff ψ is.

- $\psi := \mathrm{sk}(\varphi)$ is obtained from φ by removing $\forall$ quantifiers and introducing fresh function symbols (Herbrand/Skolem functions).

Definition

$\mathrm{sk}(L, V) = L$ for literals L
$\mathrm{sk}(\varphi \circ \psi, V) = \mathrm{sk}(\varphi, V) \circ \mathrm{sk}(\psi, V)$ for $\circ \in \{\wedge, \vee\}$
$\mathrm{sk}(\exists x \varphi, V) = \exists x\ \mathrm{sk}(\varphi, V, x)$
$\mathrm{sk}(\forall x \varphi, x_1, \ldots, x_n) = \mathrm{sk}(\varphi\{x \leftarrow f(x_1, \ldots, x_n)\}, x_1, \ldots, x_n)$

The power of functions

Note: for simplicity, consider only formulas in NNF.

Proposition

For every formula φ there exists a formula ψ that does not contain $\forall$, such that φ is valid iff ψ is.

- $\psi := \mathrm{sk}(\varphi)$ is obtained from φ by removing $\forall$ quantifiers and introducing fresh function symbols (Herbrand/Skolem functions).

Example

$$\mathrm{sk}(\exists x \forall y\ y \geqslant x) = \exists x\ f(x) \geqslant x$$

In general, $\varphi \rightarrow \mathrm{sk}(\varphi)$ but not vice-versa.

The power of functions

Note: for simplicity, consider only formulas in NNF.

Proposition

For every formula φ there exists a formula ψ that does not contain $\forall$, such that φ is valid iff ψ is.

- $\psi := \mathrm{sk}(\varphi)$ is obtained from φ by removing $\forall$ quantifiers and introducing fresh function symbols (Herbrand/Skolem functions).
- Useful when working with (cut-free) proof systems: only have to consider one type of quantifier.

The power of functions

Proposition

The theory $T \cup \{\forall\vec{x}.\exists y\varphi(\vec{x},y) \rightarrow \varphi(\vec{x},f(\vec{x}))\}$ *is a conservative extension of* T *(where the language of* T *does not contain* f*).*

Note: Here, the $\exists$ quantifier is removed since we operate on an *assumption*.

The power of functions

- In a sense, Skolem functions have no power:
  - $\mathrm{sk}(\varphi)$ is valid iff φ is valid, and
  - adding Skolem axioms yields a conservative extension.
- In another sense, they may have power: How expensive is it to go from a proof with Skolem functions to a proof without?

The power of functions

Question (Pudlák)

Assume that $\forall x \exists y \phi(x, y)$ is provable in predicate logic. Introduce a new function symbol f and an axiom A_ϕ which states

$$\forall x \phi(x, f(x)).$$

Does there exist a formula ϕ such that the extended system gives a superexponential speed-up over predicate calculus, with respect to number of symbols in proofs?[a]

[a] From P. Clote and J. Krajíček. *Open problems*, Arithmetic, proof theory and computational complexity, 1993.

The power of functions

- In its most general form, the problem is still wide open.
- More generally, in this talk we will discuss

How can Skolem functions be removed from proofs? How does this affect the length of proofs?

The rest of this talk

1. A first approach

2. Further results

The first approach

- Based on the topic of this workshop, it would be convenient if there was an algorithm based on the ε-calculus.

The first approach

- Based on the topic of this workshop, it would be convenient if there was an algorithm based on the ε-calculus.
- Luckily, there is!
- It is introduced already in D. Hilbert and P. Bernays, *Grundlagen der Mathematik II*, Springer, 1939.

Hilbert's ε-calculus

Predicate calculus + ε-symbol + ε-formulas

Example

ε-term: $\varepsilon_x \, \forall y \; x \neq s(y)$.
ε-formula: $\exists x \forall y \; x \neq s(y) \rightarrow \forall z . \varepsilon_x((\forall y) x \neq s(y)) \neq s(z)$.

Hilbert's ε-calculus

Predicate calculus + ε-symbol + ε-formulas

Example

ε-term: $\varepsilon_x \, \forall y \; x \neq s(y)$.
ε-formula: $\exists x \forall y \; x \neq s(y) \rightarrow \forall z . \varepsilon_x((\forall y) x \neq s(y)) \neq s(z)$.

In general:

$$\exists x \varphi(x) \rightarrow \varphi(\varepsilon_x \varphi(x))$$

The problem

We will look for an algorithm solving the following

Problem

Given a proof of φ using Skolem axioms, find a proof of φ that does not use Skolem axioms.

Skolem axioms: $\forall \vec{x}.\exists y \psi(\vec{x}, y) \rightarrow \psi(\vec{x}, f(\vec{x}))$ where f does not occur in ψ.
Proof: Some proof system with cut (Hilbert-style, sequent calculus, ...)

The problem

To solve

Problem

Given a proof of φ using Skolem axioms, find a proof of φ that does not use Skolem axioms.

it is sufficient to solve

Problem

Let φ be an ε-free formula. Given a proof of φ in the ε-calculus, find a proof of φ in the predicate calculus.

Going ε

From an ε-formula

$$\exists y \psi(\vec{x}, y) \rightarrow \psi(\vec{x}, \varepsilon_y \psi(\vec{x}, y))$$

and the explicit definition $f(\vec{x}) = \varepsilon_y \psi(\vec{x}, y)$ we can deduce the Skolem axiom

$$\forall \vec{x}. \exists y \psi(\vec{x}, y) \rightarrow \psi(\vec{x}, f(\vec{x})).$$

Since explicit definitions can be eliminated (by replacing definiendum by definiens), we can obtain a proof in the ε-calculus.

Removing ε

Theorem (Second ε-Theorem)

If an ε-free formula φ is derivable in the ε-calculus, then φ can be derived in predicate logic (without ε).

Theorem (Second ε-Theorem)

If an ε-free formula φ is derivable in the ε-calculus, then φ can be derived in predicate logic (without ε).

2. Ist $\mathfrak{E}$ eine in F ableitbare Formel, welche kein ε-Symbol enthält, so kann diese aus den Axiomen $\mathfrak{P}_1, \dots, \mathfrak{P}_{\mathfrak{t}}$ ohne Benutzung des ε-Symbols allein mittels des Prädikatenkalkuls abgeleitet werden („Zweites ε-Theorem").

Figure: Second ε-theorem in „Grundlagen der Mathematik".

Proving the second ε-Theorem

Proof sketch.

1. It suffices to consider validity-equivalent *Skolem normal forms* $\varphi = \exists\vec{x}\forall\vec{y}\psi(\vec{x},\vec{y})$, with ψ quantifier-free.
2. From proof of φ get proof of $\exists\vec{x}\psi(\vec{x}, f_1(\vec{x}), \ldots, f_n(\vec{x}))$, with $f_1, \ldots, f_n$ fresh.
3. To this proof, apply the *extended first ε-Theorem*. Obtain a proof of a *Herbrand disjunction* $\bigvee_{1 \leqslant i \leqslant \ell} \psi(\vec{t_i}, f_1(\vec{t_i}), \ldots, f_n(\vec{t_i}))$.
4. This proof *does not use* ε-formulas.
5. By replacing terms $f_j(\vec{t_i})$ by fresh variables $\alpha_{i,j}$ in the correct order, obtain a proof of $\bigvee_{1 \leqslant i \leqslant \ell} \psi(\vec{t_i}, \vec{\alpha_i})$.
6. Introduce quantifiers to obtain the desired proof of φ.

$\square$

Summary of the approach

- We want to eliminate Skolem functions from proofs.
- This reduces to eliminating ε-terms from proofs (by setting $f(\vec{x}) = \varepsilon_y \varphi(\vec{x}, y)$).
- This can be done, but the approach uses the extended first ε-theorem.
- What change in proof length does this induce?

The first ε-Theorem

Theorem (Extended first ε-Theorem)

If a formula $\exists\vec{x}\varphi(\vec{x})$, with φ quantifier-free, is derivable in the ε-calculus, then a formula

$$\bigvee_{1 \leqslant i \leqslant n} \varphi(\vec{t_i})$$

is derivable in predicate calculus without the use of bound variables, for some sequences of terms $\vec{t_1}, \ldots, \vec{t_n}$ not containing the ε-symbol.

Complexity of the approach

- We will see: Application of the extended first ε-Theorem may cause a large increase in proof length.
- Therefore, so does application of the second ε-Theorem, and hence this approach to elimination of Skolem functions.

Complexity of the first ε-Theorem

- We will show that the extended first ε-Theorem can be used to do cut-elimination.
- We then apply the following:

Theorem (Orevkov, Statman)

There exists a family of formulas $(\varphi_i)_{i \in \mathbb{N}}$ (of elementary size) such that

1. *φ_i have proofs with cut of elementary length, but*
2. *all cut-free proofs of φ_i have non-elementary length.*

Non-elementary: $\left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} i$

Cut-elimination using the first ε-Theorem

1. Sequent calculus with cut can be translated into predicate calculus.
2. Using the extended first ε-Theorem, we get a proof in the predicate calculus which *does not use bound variables*.
3. This proof can be translated into sequent calculus with *quantifier-free cuts*.
4. Quantifier-free cuts have (only) exponential elimination.[1]

[1](For a more direct proof, see (Moser, Zach 2006)).

Summary of the approach

- We want to eliminate Skolem functions from proofs.
- This reduces to eliminating ε-terms from proofs (by setting $f(\vec{x}) = \varepsilon_y \varphi(\vec{x}, y)$).
- This can be done, but the approach uses the extended first ε-theorem, which has non-elementary worst-case complexity.
- Can we do better?

Summary of the approach

- Complexity of algorithm due to the fact that an "essentially cut-free" proof is produced.
- Can cut-elimination be avoided?
- What happens if we consider cut-free proofs right away?

The rest of this talk

1. A first approach

2. Further results

Further results

- An algorithm due to Maehara (1955), based on cut-elimination.
- An algorithm due to Shoenfield (2001), based on Herbrand's theorem.
- A better algorithm for a subproblem due to Avigad (2003).
- An algorithm and a lower bound for a problem on cut-free proofs due to Baaz, Hetzl, W (2010).

Further results

- An algorithm due to Maehara (1955), based on cut-elimination.
- An algorithm due to Shoenfield (2001), based on Herbrand's theorem.
- A better algorithm for a subproblem due to Avigad (2003).
- An algorithm and a lower bound for a problem on cut-free proofs due to Baaz, Hetzl, W (2010).

Avigad's result

Theorem (Avigad 2003)

Suppose Γ codes finite functions. Then Γ has an efficient (i.e. polynomial-time) elimination of Skolem functions.

Coding finite functions

A set of sentences Γ *codes finite functions* if for each n there are

1. a definable element, "$\emptyset_n$";
2. a definable relation, "$x_0, \ldots, x_{n-1} \in dom_n(p)$";
3. a definable function, "$eval_n(p, x_0, \ldots, x_{n-1})$"; and
4. a definable function, "$p \oplus_n (x_0, \ldots, x_{n-1} \mapsto y)$".

such that, for each n, Γ proves

$$\vec{x} \notin dom_n(\emptyset_n)$$

and such that all definitions and proofs can be constructed in time polynomial in n.

Coding finite functions

A set of sentences Γ *codes finite functions* if for each n there are

1. a definable element, "$\varnothing_n$";
2. a definable relation, "$x_0, \dots, x_{n-1} \in dom_n(p)$";
3. a definable function, "$eval_n(p, x_0, \dots, x_{n-1})$"; and
4. a definable function, "$p \oplus_n (x_0, \dots, x_{n-1} \mapsto y)$".

such that, for each n, Γ proves

$$\vec{w} \in dom_n(p \oplus_n (\vec{x} \mapsto y)) \leftrightarrow (\vec{w} \in dom_n(p) \vee \vec{w} = \vec{x})$$

and such that all definitions and proofs can be constructed in time polynomial in n.

Coding finite functions

A set of sentences Γ *codes finite functions* if for each n there are

1. a definable element, "$\varnothing_n$";
2. a definable relation, "$x_0, \ldots, x_{n-1} \in dom_n(p)$";
3. a definable function, "$eval_n(p, x_0, \ldots, x_{n-1})$"; and
4. a definable function, "$p \oplus_n (x_0, \ldots, x_{n-1} \mapsto y)$".

such that, for each n, Γ proves

$$eval_n(p \oplus_n (\vec{x} \mapsto y), \vec{x}) = y$$

and such that all definitions and proofs can be constructed in time polynomial in n.

Coding finite functions

A set of sentences Γ *codes finite functions* if for each n there are

1. a definable element, "$\emptyset_n$";
2. a definable relation, "$x_0, \ldots, x_{n-1} \in dom_n(p)$";
3. a definable function, "$eval_n(p, x_0, \ldots, x_{n-1})$"; and
4. a definable function, "$p \oplus_n (x_0, \ldots, x_{n-1} \mapsto y)$".

such that, for each n, Γ proves

$$\vec{w} \neq \vec{x} \rightarrow eval_n(p \oplus_n (\vec{x} \mapsto y), \vec{w}) = eval_n(p, \vec{w})$$

and such that all definitions and proofs can be constructed in time polynomial in n.

Avigad's result

Theorem (Avigad 2003)

Suppose Γ codes finite functions. Then Γ has an efficient (i.e. polynomial-time) elimination of Skolem functions.

Proof idea.

1. For a single Skolem function f:
2. Define a translation t^p that replaces $f(t_1, \dots, t_n)$ by $eval_n(p, t_1^p, \dots, t_n^p)$.
3. Define a relation $p \Vdash \varphi$ that replaces terms t in φ by t^p.
4. Transform a proof of φ to a proof of $\forall p(Cond(p) \to p \Vdash \varphi)$, where $Cond(p) =$"p is an approximation of f".
5. Use proofs of $(p \Vdash \varphi) \leftrightarrow \varphi$ and $\Vdash \forall \vec{x}, y(\psi(\vec{x}, y) \to \psi(\vec{x}, f(\vec{x})))$ to obtain a proof of φ (in the original language).

□

Avigad's result

Theorem (Avigad 2003)

Suppose Γ codes finite functions. Then Γ has an efficient (i.e. polynomial-time) elimination of Skolem functions.

Proof idea.

1. For more than one Skolem function:
2. Show that $\Gamma \supseteq \{\exists x, y (x \neq y)\}$ has an efficient elimination of *definitions*.
3. Trivially, $\Gamma \supseteq \{\forall x, y (x = y)\}$ has efficient elimination of Skolem functions.
4. Use definitions to handle the iteration of the translation efficiently, then apply elimination of definitions.

$\square$

Further results

- An algorithm due to Maehara (1955), based on cut-elimination.
- An algorithm due to Shoenfield (2001), based on Herbrand's theorem.
- A better algorithm for a subproblem due to Avigad (2003).
- An algorithm and a lower bound for a related problem due to Baaz, Hetzl, W (2010).

Further results

- An algorithm due to Maehara (1955), based on cut-elimination.
- An algorithm due to Shoenfield (2001), based on Herbrand's theorem.
- A better algorithm for a subproblem due to Avigad (2003).
- An algorithm and a lower bound for a related problem due to Baaz, Hetzl, W (2010).

Cut-free proofs

- We are interested in the effect of Skolem functions on *cut-free proofs*.
- Cut-free proofs are interesting:
  - Usually generated by automated theorem provers.
  - Efficient extraction of data: Interpolants, Herbrand sequents.
- Here, we look at cut-free *tree-like* proofs.

Cut-free proofs

One can formulate a cut-free version of the problem in Pudlák's question.

Problem

Input: *Proof of* $(\forall x)\phi(x, f(x)) \vdash \psi$.
Output: *Proof of* $(\forall x)(\exists y)\phi(x, y) \vdash \psi$.

But in the cut-free context, the requirement that the quantifier-to-be-skolemized is in prefix position can be bad.

Cut-free proofs

Theorem (Baaz, Leitsch 1994)

There exists a family of formulas $(\varphi_i)_{i \in \mathbb{N}}$ (of elementary size) such that

1. $\mathrm{sk}(\varphi_i)$ *have proofs of elementary length, but*
2. *there exist prefix forms ψ_i of φ_i such that all cut-free proofs of ψ_i have non-elementary length.*

Cut-free proofs

Instead, we consider

Problem (Proof deskolemization)

Input: *φ, proof of* $\mathrm{sk}(\varphi)$.
Output: *Proof of φ.*

Cut-free proofs

- In the ε-calculus based method for elimination of Skolem functions, we saw how to:
- Given a proof with only quantifier-free cuts of $\exists\vec{x}\psi(\vec{x}, f_1(\vec{x}), \ldots, f_n(\vec{x}))$, with $f_1, \ldots, f_n$ fresh,
- obtain a proof of $\exists\vec{x}\forall\vec{y}\psi(\vec{x}, \vec{y})$,

Cut-free proofs

- In the ε-calculus based method for elimination of Skolem functions, we saw how to:
- Given a proof with only quantifier-free cuts of $\exists\vec{x}\psi(\vec{x}, f_1(\vec{x}), \ldots, f_n(\vec{x}))$, with $f_1, \ldots, f_n$ fresh,
- obtain a proof of $\exists\vec{x}\forall\vec{y}\psi(\vec{x}, \vec{y})$, by replacing terms $f_j(\vec{t_i})$ by fresh variables $\alpha_{i,j}$, and introducing quantifiers.

Cut-free proofs

- In the ε-calculus based method for elimination of Skolem functions, we saw how to:
- Given a proof with only quantifier-free cuts of $\exists \vec{x} \psi(\vec{x}, f_1(\vec{x}), \ldots, f_n(\vec{x}))$, with $f_1, \ldots, f_n$ fresh,
- obtain a proof of $\exists \vec{x} \forall \vec{y} \psi(\vec{x}, \vec{y})$, by replacing terms $f_j(\vec{t_i})$ by fresh variables $\alpha_{i,j}$, and introducing quantifiers.
- Method can be easily extended to obtain a polynomial algorithm for the *prefix case* $(Q_1 x_1) \cdots (Q_n y_n) \psi$.

Cut-free proofs

- For the infix case, we necessarily have to rearrange the proof:

Proposition

There exists a family of formulas $(\varphi_i)_{i\in\mathbb{N}}$ (of polynomial-size) such that

1. *there exist polynomial-length*[a] *proofs of* $\mathrm{sk}(\varphi_i)$ *but*
2. *all proofs of* φ_i *have exponential length.*

[a] Here, length = number of sequents. For a more efficient version of sk, it also holds for number of symbols.

- This is essentially due to the eigenvariable condition forcing application of a binary inference.

Cut-free proofs

- But this is the worst that can happen.

Theorem

Let π *be a proof of* $\mathrm{sk}(\varphi)$. *Then there exists a proof* λ *of* φ *such that* $|\lambda| \leqslant 2^{p(|\pi|)}$ *for some polynomial* p.

An extension

- This result can be lifted to some proofs with cut:

Theorem

Let π be a proof of $\mathrm{sk}(\varphi)$ such that for all Skolem terms $f(t_1, \ldots, t_n)$ occurring in cut-formulas, no t_i contains a bound variable. Then there exists a proof λ of φ such that $|\lambda| \leqslant 2^{p(|\pi|)}$.

- Any cut-free deskolemization algorithm can be lifted to this class of proofs.
- One is reminded of the restriction imposed by (Miller 1983) to obtain *soundness* of Skolemization in higher-order logic.

Conclusion

- The problem of removing Skolem functions from proofs *efficiently* is still open.
- The general algorithms are of non-elementary complexity.
- There exists a polynomial algorithm for a restricted case.
- Concrete open problems:
  - Pudlák's question for theories that *do not* code finite functions.
  - Deskolemization problem, cut-free case: DAG-like proofs.
- Further settings: Non-classical, higher-order, equality.