Proof theoretic approaches to rewriting

Thomas Powell

Institut des Hautes Études Scientifiques

Workshop on Two Faces of Complexity
part of the Vienna Summer of Logic
12 July 2014

Outline of talk

Part I. Looking to the past

- The computational content of proofs - classical proof theoretic work of the 20th century.
- Link to rewriting: Recursive path orders and fragments of arithmetic.

Part II. Looking towards the future

- Modern proof theory as a tool for establishing complexity bounds on programs.
- Some concrete examples of how this technique could be applied

Theorem. For all integers n there exists some prime $p \geq n$.

Theorem. For all integers n there exists some prime $p \geq n$.

Implicit bound in Euclid's proof: $p(n) \leq n! + 1$.

Theorem. For all integers n there exists some prime $p \geq n$.

Implicit bound in Euclid's proof: $p(n) \leq n! + 1$.

Theorem. For all terms t in $\mathcal{R}$ there exists some N such that all rewrite sequence starting from t have length $\leq N$.

Theorem. For all integers n there exists some prime $p \geq n$.

Implicit bound in Euclid's proof: $p(n) \leq n! + 1$.

Theorem. For all terms t in $\mathcal{R}$ there exists some N such that all rewrite sequence starting from t have length $\leq N$.

Can we find a concrete bound in termination proof: $N(t) \leq f(|t|)$?

Theorem. For all integers n there exists some prime $p \geq n$.

Implicit bound in Euclid's proof: $p(n) \leq n! + 1$.

Theorem. For all terms t in $\mathcal{R}$ there exists some N such that all rewrite sequence starting from t have length $\leq N$.

Can we find a concrete bound in termination proof: $N(t) \leq f(|t|)$?

multiset path ordering $\Rightarrow$ primitive recursive d. c.

lexicographic path ordering $\Rightarrow$ multiply recursive d. c.

General challenge:

$$\boxed{\text{termination proof technique}} \Rightarrow \boxed{\text{upper bound on complexity}}$$

General challenge:

$$\boxed{\text{termination proof technique}} \Rightarrow \boxed{\text{upper bound on complexity}}$$

G. Kreisel "What more do we know if we have a proved a theorem by restricted means than if we merely know that it is true?"

General challenge:

termination proof technique $\Rightarrow$ upper bound on complexity

G. Kreisel "What more do we know if we have a proved a theorem by restricted means than if we merely know that it is true?"

Techniques such as proof interpretations - which analyse the logical structure of proofs - provide us with a powerful tool for tackling this problem, and have great potential for both:

- Obtaining new complexity results;
- Understanding *why* termination techniques yield the complexity bounds that they do.

In one line, a proof interpretation is a formal translation acting on the logical structure of proofs. E.g. Gödel's Dialectica interpretation.

Of particular interest are (combinations of) interpretations that directly realize $\forall\exists$-formulas.

In one line, a proof interpretation is a formal translation acting on the logical structure of proofs. E.g. Gödel's Dialectica interpretation.

Of particular interest are (combinations of) interpretations that directly realize $\forall\exists$-formulas.

$$A \to \forall n \exists m B_0(n, m)$$

In one line, a proof interpretation is a formal translation acting on the logical structure of proofs. E.g. Gödel's Dialectica interpretation.

Of particular interest are (combinations of) interpretations that directly realize $\forall\exists$-formulas.

$$A \to \forall n \exists m B_0(n,m)$$
$$\Rightarrow \forall n (A \to \exists m B_0(n,m))$$

In one line, a proof interpretation is a formal translation acting on the logical structure of proofs. E.g. Gödel's Dialectica interpretation.

Of particular interest are (combinations of) interpretations that directly realize $\forall\exists$-formulas.

$$\begin{aligned} & A \to \forall n \exists m B_0(n,m) \\ & \Rightarrow \forall n (A \to \exists m B_0(n,m)) \\ & \Rightarrow \forall n (\exists \varepsilon \ \mathrm{app}(A_\varepsilon) \to \exists m B_0(n,m)) \end{aligned}$$

where $A \leftrightarrow \exists \varepsilon \ \mathrm{app}(A_\varepsilon)$.

In one line, a proof interpretation is a formal translation acting on the logical structure of proofs. E.g. Gödel's Dialectica interpretation.

Of particular interest are (combinations of) interpretations that directly realize $\forall\exists$-formulas.

$$\begin{aligned} & A \to \forall n \exists m B_0(n,m) \\ & \Rightarrow \forall n (A \to \exists m B_0(n,m)) \\ & \Rightarrow \forall n (\exists \varepsilon \ \mathrm{app}(A_\varepsilon) \to \exists m B_0(n,m)) \\ & \Rightarrow \forall \varepsilon, n \exists m (\mathrm{app}(A_\varepsilon) \to B_0(n,m)) \end{aligned}$$

where $A \leftrightarrow \exists \varepsilon \ \mathrm{app}(A_\varepsilon)$.

In one line, a proof interpretation is a formal translation acting on the logical structure of proofs. E.g. Gödel's Dialectica interpretation.

Of particular interest are (combinations of) interpretations that directly realize $\forall\exists$-formulas.

$$\begin{aligned}
&A \to \forall n \exists m B_0(n,m) \\
&\Rightarrow \forall n (A \to \exists m B_0(n,m)) \\
&\Rightarrow \forall n (\exists \varepsilon \ \mathrm{app}(A_\varepsilon) \to \exists m B_0(n,m)) \\
&\Rightarrow \forall \varepsilon, n \exists m (\mathrm{app}(A_\varepsilon) \to B_0(n,m)) \\
&\Rightarrow \exists f \forall \varepsilon, n (\mathrm{app}(A_\varepsilon) \to B_0(n, f_\varepsilon(n)))
\end{aligned}$$

where $A \leftrightarrow \exists \varepsilon \ \mathrm{app}(A_\varepsilon)$.

In one line, a proof interpretation is a formal translation acting on the logical structure of proofs. E.g. Gödel's Dialectica interpretation.

Of particular interest are (combinations of) interpretations that directly realize $\forall\exists$-formulas.

$$\begin{aligned}
&A \to \forall n \exists m B_0(n,m)\\
&\Rightarrow \forall n(A \to \exists m B_0(n,m))\\
&\Rightarrow \forall n(\exists \varepsilon\ \mathrm{app}(A_\varepsilon) \to \exists m B_0(n,m))\\
&\Rightarrow \forall \varepsilon, n \exists m(\mathrm{app}(A_\varepsilon) \to B_0(n,m))\\
&\Rightarrow \exists f \forall \varepsilon, n(\mathrm{app}(A_\varepsilon) \to B_0(n, f_\varepsilon(n)))
\end{aligned}$$

where $A \leftrightarrow \exists \varepsilon\ \mathrm{app}(A_\varepsilon)$.

The computational complexity of f_ε is determined by the proof-theoretic complexity of A.

Classic results of 20th century proof theory (Gödel 1958, Parsons 1972):

$$\begin{aligned}
\mathsf{PA} \vdash \forall n \exists m A_0(n,m) \Rightarrow & \exists f \in \mathcal{F}_{\varepsilon_0} \forall n A_0(n, f(n)) \\
& f \textit{ in system } \mathsf{T} \\
& \vdots \\
\mathsf{I}\Sigma_2 \vdash \forall n \exists m A_0(n,m) \Rightarrow & \exists f \in \mathcal{F}_{\omega^\omega} \forall n A_0(n, f(n)) \\
& f \textit{ multiply recursive} \\
\mathsf{I}\Sigma_1 \vdash \forall n \exists m A_0(n,m) \Rightarrow & \exists f \in \mathcal{F}_{\omega} \forall n A_0(n, f(n)) \\
& f \textit{ primitive recursive}
\end{aligned}$$

Application to program complexity (Buchholz 1994):

Step 1. Any finite rewrite system $\mathcal{R}$ reducing under $\prec_{\text{mpo}}$ or $\prec_{\text{lpo}}$ uses only a (finitely branching) approximation $\prec^k_{\text{mpo}}$, $\prec^k_{\text{lpo}}$ of these orderings, where k depends on $\mathcal{R}$.

Step 2. $\prec^k_{\text{mpo}}$ is provably well-founded in $\mathsf{I}\Sigma_1$, while $\prec^k_{\text{lpo}}$ is provably well-founded in $\mathsf{I}\Sigma_2$.

Application to program complexity (Buchholz 1994):

Step 1. Any finite rewrite system $\mathcal{R}$ reducing under $\prec_{\text{mpo}}$ or $\prec_{\text{lpo}}$ uses only a (finitely branching) approximation $\prec^k_{\text{mpo}}$, $\prec^k_{\text{lpo}}$ of these orderings, where k depends on $\mathcal{R}$.

Step 2. $\prec^k_{\text{mpo}}$ is provably well-founded in $\mathsf{I}\Sigma_1$, while $\prec^k_{\text{lpo}}$ is provably well-founded in $\mathsf{I}\Sigma_2$.

Result.

$$\text{multiset path ordering } (\mathsf{I}\Sigma_1) \Rightarrow N(t) \leq f(|t|) \text{ for } f \in \mathcal{F}_\omega$$
$$\text{lexicographic path ordering } (\mathsf{I}\Sigma_2) \Rightarrow N(t) \leq f(|t|) \text{ for } f \in \mathcal{F}_{\omega^\omega}$$

Very rough explanation...

Want to prove $t_1, \ldots, t_n$ well-founded w.r.t. $\prec$ then $(t_1, \ldots, t_n)$ well-founded w.r.t. $\prec^{\text{mul}}/\prec^{\text{lex}}$.

$$(t_1, \ldots, t_{j-1}, t'_1, \ldots, t'_m, t_{j+1}, \ldots, t_n) \prec^{\text{mul}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

$$(t_1, \ldots, t_{j-1}, t'_j, \underbrace{s_{j+1}, \ldots, s_n}_{\text{'uncontrolled'}}) \prec^{\text{lex}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

Very rough explanation...

Want to prove $t_1, \ldots, t_n$ well-founded w.r.t. $\prec$ then $(t_1, \ldots, t_n)$ well-founded w.r.t. $\prec^{\text{mul}}/\prec^{\text{lex}}$.

$$(t_1, \ldots, t_{j-1}, t'_1, \ldots, t'_m, t_{j+1}, \ldots, t_n) \prec^{\text{mul}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

$$(t_1, \ldots, t_{j-1}, t'_j, \underbrace{s_{j+1}, \ldots, s_n}_{\text{'uncontrolled'}}) \prec^{\text{lex}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

Need a universal quantification to deal with uncontrolled terms in $\prec^{\text{lex}}$:

$$B(t_{j-1}) :\equiv \forall s_{j+1}, \ldots, s_n[(t_1, \ldots, t_{j-1}, t'_j, s_{j+1}, \ldots, s_n) \text{ w.f.}]$$

Causes a one-step shift up the arithmetic hierarchy!

Method bears close connections with alternative approaches that deal explicitly with ordinal analysis or monotone assingments.

Nevertheless, focusing on the precise formalization of termination arguments and appealing to proof-theoretic meta-theorems has several advantages.

Method bears close connections with alternative approaches that deal explicitly with ordinal analysis or monotone assingments.

Nevertheless, focusing on the precise formalization of termination arguments and appealing to proof-theoretic meta-theorems has several advantages.

- Reveals on a deep, structural level *why* a termination proofs produces a certain complexity bound, and encourages a uniform and general way of thinking about these proofs.

Method bears close connections with alternative approaches that deal explicitly with ordinal analysis or monotone assingments.

Nevertheless, focusing on the precise formalization of termination arguments and appealing to proof-theoretic meta-theorems has several advantages.

- Reveals on a deep, structural level *why* a termination proofs produces a certain complexity bound, and encourages a uniform and general way of thinking about these proofs.
- Great potential for new applications in the analysis of a wider range of termination methods, building on substantial amount of work done in last 20 years.

Method bears close connections with alternative approaches that deal explicitly with ordinal analysis or monotone assingments.

Nevertheless, focusing on the precise formalization of termination arguments and appealing to proof-theoretic meta-theorems has several advantages.

- Reveals on a deep, structural level *why* a termination proofs produces a certain complexity bound, and encourages a uniform and general way of thinking about these proofs.
- Great potential for new applications in the analysis of a wider range of termination methods, building on substantial amount of work done in last 20 years.
- Can simultaneously work towards new complexity bounds while developing broadly applicable results in proof theory.

The last few decades has seen great advances in proof theoretic techniques and their application in mathematics and computer science.

In particular, meta-theorems of the form

$$\mathcal{T} \vdash A \Rightarrow \exists t \in \mathcal{F}(t \text{ interprets } A)$$

have become increasingly refined and sophisticated, and oriented towards practical as opposed to foundational results.

The last few decades has seen great advances in proof theoretic techniques and their application in mathematics and computer science.

In particular, meta-theorems of the form

$$\mathcal{T} \vdash A \Rightarrow \exists t \in \mathcal{F}(t \text{ interprets } A)$$

have become increasingly refined and sophisticated, and oriented towards practical as opposed to foundational results.

QUESTION:

Can we develop new and interesting meta-theorems (along the lines of Buchholz '94, but not necessarily restricted to path orders!) which capture families of termination proofs and can be used to derive corresponding complexity bounds in a uniform way?

Example

$$(t_1, \ldots, t_{j-1}, t'_1, \ldots, t'_m, t_{j+1}, \ldots, t_n) \prec^{\text{mul}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

$$(t_1, \ldots, t_{j-1}, t'_j, \underbrace{s_{j+1}, \ldots, s_n}_{\text{'uncontrolled'}}) \prec^{\text{lex}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

Can we characterise a broader class of 'bounded' path-orderings for which well-foundedness is provable in $\mathsf{I}\Sigma_1$?

Example

$$(t_1, \ldots, t_{j-1}, t'_1, \ldots, t'_m, t_{j+1}, \ldots, t_n) \prec^{\text{mul}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

$$(t_1, \ldots, t_{j-1}, t'_j, \underbrace{s_{j+1}, \ldots, s_n}_{\text{'uncontrolled'}}) \prec^{\text{lex}} (t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$$

Can we characterise a broader class of 'bounded' path-orderings for which well-foundedness is provable in $\mathsf{I}\Sigma_1$?

Could be used to derive primitive recursive closure properties e.g. unnested multiple recursion:

$$f(x+1, y+1) = h(x, y, f(x, p(x, y)), f(x+1, y))$$

Typically, path orders developed to incorporate such schemata resort to a complex ordinal analysis to verify complexity bound.

We now have a better understanding of the constructive meaning of key principles used to prove termination, such as Ramsey's theorem, Higman's lemma and Kruskal's theorem. Recent work utilising proof interpretations includes

We now have a better understanding of the constructive meaning of key principles used to prove termination, such as Ramsey's theorem, Higman's lemma and Kruskal's theorem. Recent work utilising proof interpretations includes

- **Ramsey's theorem** New quantitative results (Kreuzer/Kohlenbach 2012), constructive game theoretic interpretation (Oliva/P. 2013).

We now have a better understanding of the constructive meaning of key principles used to prove termination, such as Ramsey's theorem, Higman's lemma and Kruskal's theorem. Recent work utilising proof interpretations includes

- **Ramsey's theorem** New quantitative results (Kreuzer/Kohlenbach 2012), constructive game theoretic interpretation (Oliva/P. 2013).
- **Higman and Kruskal** Quite general computational interpretations using realizability (Berger 2004, Seisenberger 2003) and Dialectica (P. 2013).

We now have a better understanding of the constructive meaning of key principles used to prove termination, such as Ramsey's theorem, Higman's lemma and Kruskal's theorem. Recent work utilising proof interpretations includes

- **Ramsey's theorem** New quantitative results (Kreuzer/Kohlenbach 2012), constructive game theoretic interpretation (Oliva/P. 2013).
- **Higman and Kruskal** Quite general computational interpretations using realizability (Berger 2004, Seisenberger 2003) and Dialectica (P. 2013).

QUESTION:

By focusing on restricted forms of these principles can we extract useful numerical information from termination proofs which use them?

Example

A form of Ramsey's theorem for pairs has been used to prove program termination via *disjuctively well-founded transition invariants* (Podelski/Rybalchenko 2004): If a program P has a transition invariant of the form

$$T = T_1 \cup \ldots \cup T_n$$

where the T_i are well-founded, then P terminates.

Example

A form of Ramsey's theorem for pairs has been used to prove program termination via *disjuctively well-founded transition invariants* (Podelski/Rybalchenko 2004): If a program P has a transition invariant of the form

$$T = T_1 \cup \ldots \cup T_n$$

where the T_i are well-founded, then P terminates.

Current work by Berardi, Oliva and Steila is focusing on the formal analysis of this result and aims towards producing upper bounds on the complexity of programs shown to terminate using this method.

Proof theoretic machinery designed to give computational interpretations to proofs often finds applications in other areas, some quite unexpected!

$$\boxed{\text{intended application}} \Leftarrow \boxed{\text{theoretical work}} \rightsquigarrow \boxed{\text{new insights}}$$

Proof theoretic machinery designed to give computational interpretations to proofs often finds applications in other areas, some quite unexpected!

$$\boxed{\text{intended application}} \Leftarrow \boxed{\text{theoretical work}} \rightsquigarrow \boxed{\text{new insights}}$$

- Monotone Dialectica interpretation $\rightsquigarrow$ structural understanding of aspects of ergodic theory.

Proof theoretic machinery designed to give computational interpretations to proofs often finds applications in other areas, some quite unexpected!

$$\boxed{\text{intended application}} \Leftarrow \boxed{\text{theoretical work}} \rightsquigarrow \boxed{\text{new insights}}$$

- Monotone Dialectica interpretation $\rightsquigarrow$ structural understanding of aspects of ergodic theory.
- Spector's bar recursive interpretation of analysis $\rightsquigarrow$ higher-type generalisations of Nash Equilibrium.

Proof theoretic machinery designed to give computational interpretations to proofs often finds applications in other areas, some quite unexpected!

$$\boxed{\text{intended application}} \Leftarrow \boxed{\text{theoretical work}} \rightsquigarrow \boxed{\text{new insights}}$$

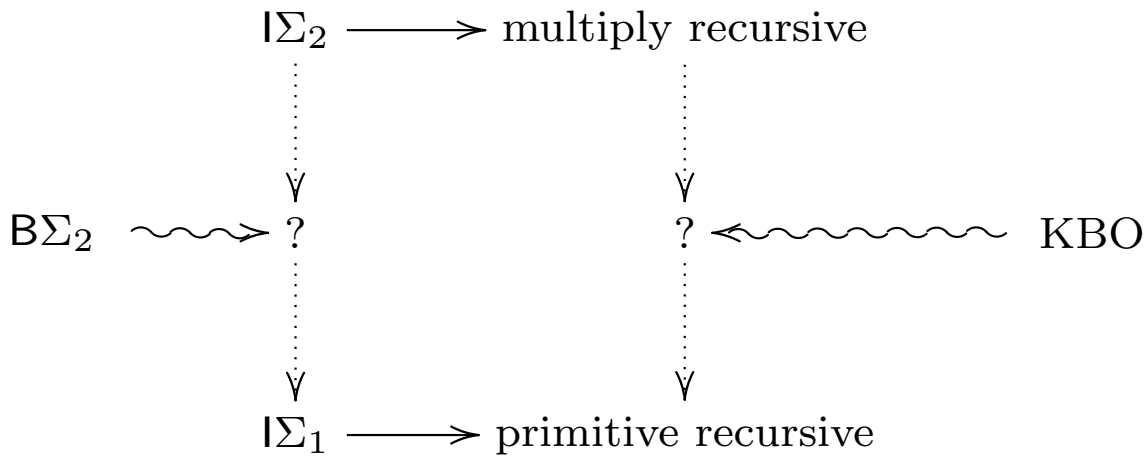
- Monotone Dialectica interpretation $\rightsquigarrow$ structural understanding of aspects of ergodic theory.
- Spector's bar recursive interpretation of analysis $\rightsquigarrow$ higher-type generalisations of Nash Equilibrium.

QUESTION

Will developing new theoretical results tailored to analysing termination proofs have an impact in other areas of logic or computer science?

Example

Most existing applications of proof theory focus on obtaining primitive recursive bounds. On the other hand, many termination methods induce multiply recursive complexity.



Can Parsons' result be extended to systems between $\mathsf{I}\Sigma_1$ and $\mathsf{I}\Sigma_2$?

Concluding remarks

Concluding remarks

- The formal analysis of proofs has led to many important applications in mathematics and computer science.

Concluding remarks

- The formal analysis of proofs has led to many important applications in mathematics and computer science.
- In particular this approach has been used to obtain bounds on the derivational complexity of path orderings, but arguably has not been used to anywhere near its full potential.

Concluding remarks

- The formal analysis of proofs has led to many important applications in mathematics and computer science.
- In particular this approach has been used to obtain bounds on the derivational complexity of path orderings, but arguably has not been used to anywhere near its full potential.
- Given the amount of progress made in applied proof theory in recent years, there is a great scope for new applications in complexity analysis

Concluding remarks

- The formal analysis of proofs has led to many important applications in mathematics and computer science.
- In particular this approach has been used to obtain bounds on the derivational complexity of path orderings, but arguably has not been used to anywhere near its full potential.
- Given the amount of progress made in applied proof theory in recent years, there is a great scope for new applications in complexity analysis
- Can be used not only to directly obtain new results, but to provide a proof-theoretic framework with which to better understand existing work