Inductive theorem proving based on tree grammars

Stefan Hetzl

Institute of Discrete Mathematics and Geometry
Vienna University of Technology

PUMA Graduiertenkolleg
TU München

April 6, 2016

Introduction

Why is inductive theorem proving difficult?

- Only *restricted* cut-elimination in arithmetic
  $\Longrightarrow$ No subformula property
- Induction invariants necessarily non-analytic
  - Theoretical results: $\mathrm{Con}_{I\Sigma_k} \in \Pi_1$, but $I\Sigma_k \nvdash \mathrm{Con}_{I\Sigma_k}$
  - Practical experience: loop invariants, math. proofs, ...
- How to find **non-analytic** key lemma / generalization?

Our Approach

- Consider proofs of instances $A(n)$ of $\forall x\, A(x)$
  E.g. bounded model checking, constructive ω-rule, ...

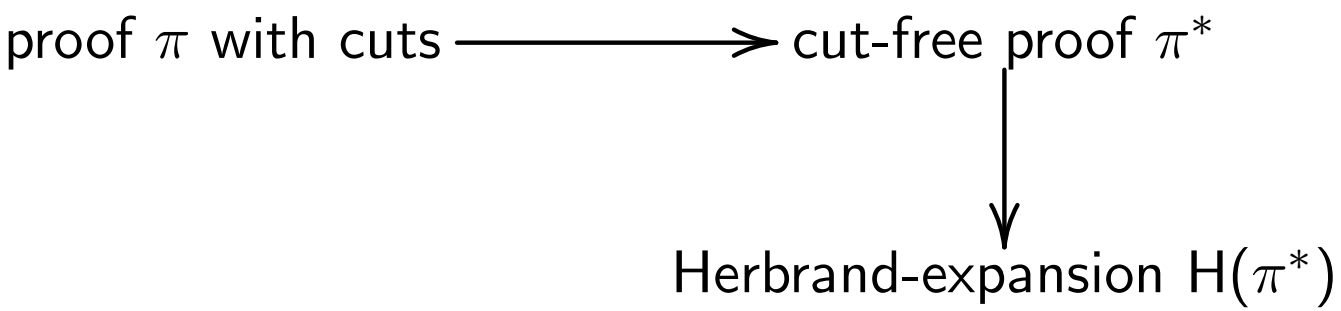
- Proof-theoretic analysis of structure of instance-proofs
  - Based on Herbrand's theorem and tree grammars
  - Extending method for cut-introduction to induction

Outline

- Cut-elimination and cut-introduction
- Inductive theorem proving based on tree grammars

Cut-elimination and Herbrand's theorem

- **Herbrand's theorem** (1930). A first-order formula φ is valid iff a finite expansion of φ is a tautology.
  E.g. $\exists x\, A_{\text{qf}}(x)$ expands to $A_{\text{qf}}(t_1) \vee \ldots \vee A_{\text{qf}}(t_n)$
- In total:

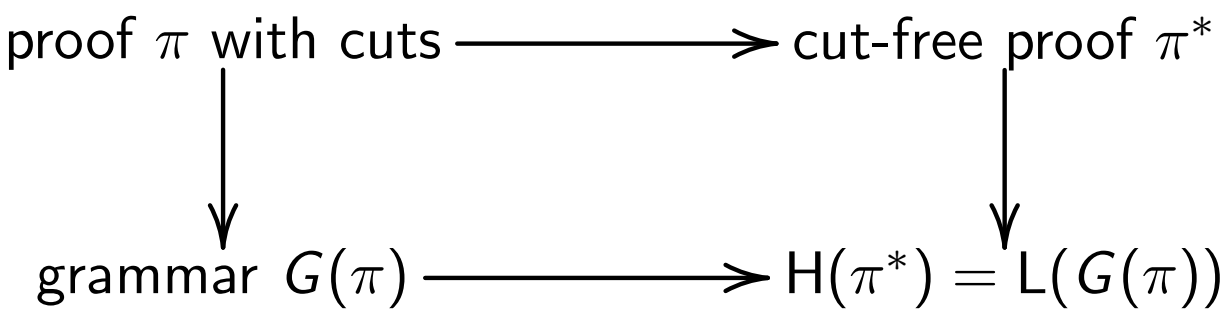


Cut-elimination and tree grammars

- Information of Herbrand-expansion $A_{\text{qf}}(t_1) \vee \ldots \vee A_{\text{qf}}(t_n)$ is $T = \{t_1, \ldots, t_n\}$ – a (finite) tree language.

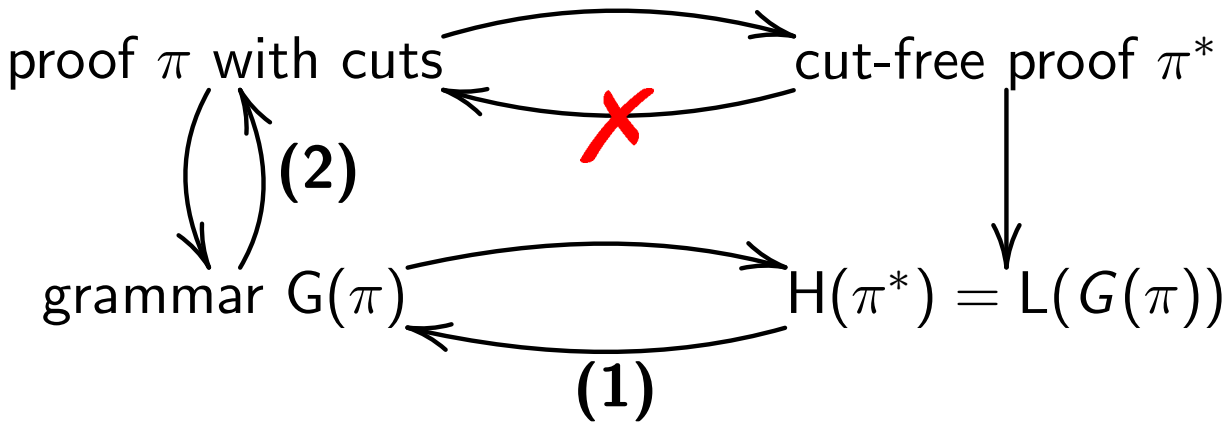
Cut-elimination and tree grammars

- Information of Herbrand-expansion $A_{\text{qf}}(t_1) \vee \ldots \vee A_{\text{qf}}(t_n)$ is $T = \{t_1, \ldots, t_n\}$ – a (finite) tree language.
- **Theorem** (H 2012). If π is a proof with Π_1-cuts then there is a totally rigid acyclic tree grammar $G(\pi)$ s.t. $L(G(\pi))$ is a Herbrand-expansion and $|G(\pi)| \leq |\pi|$.
- In total:



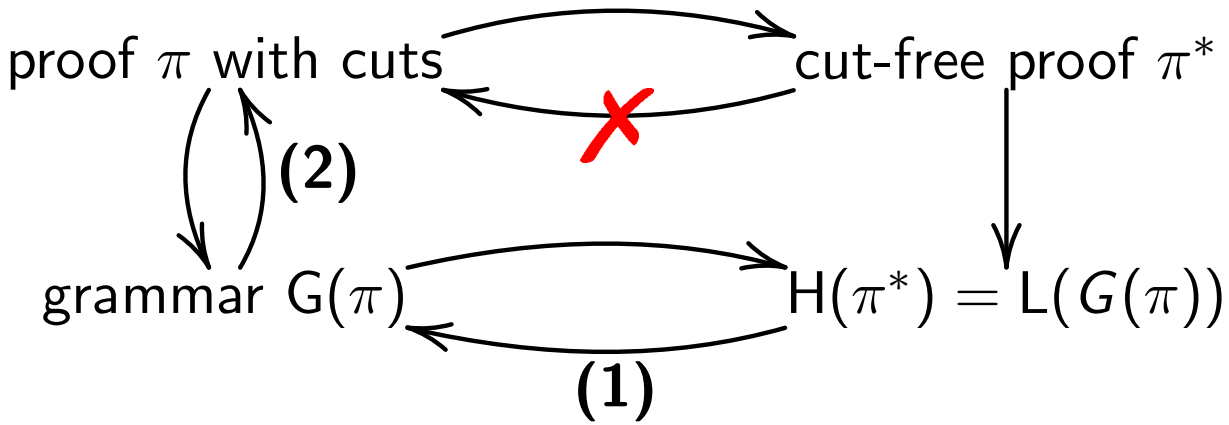
Cut-introduction based on tree grammars

- Cut-introduction method [H, Leitsch, Reis, Weller '12, '14]



Cut-introduction based on tree grammars

- Cut-introduction method [H, Leitsch, Reis, Weller '12, '14]



1. Given L, compute G s.t. $L(G) = L$ (or $L(G) \supseteq L$).
2. Solve Boolean unification problem with predicates, e.g.

$$S(X) \quad \equiv \quad X(\alpha) \to (X(t_1) \wedge X(t_2)) \vdash A(u_1), A(u_2), A(u_3)$$

Find F s.t. $S(F)$ is a tautology

- Quantifier-free
- Every solution induces proof with cuts
- For $S(X)$ induced by grammar for Π_1-cuts: canonical solution

Outline

- ✓ Cut-elimination and cut-introduction
- ▶ Inductive theorem proving based on tree grammars

Cut and induction

$$\dfrac{\begin{array}{cc}(\pi_b) & (\pi_s(\alpha)) \\ \Gamma \vdash A(0) & \Gamma, A(\alpha) \vdash A(s\alpha)\end{array}}{\Gamma \vdash A(t)}\ \text{ind}$$

If t is variable-free, there is $n \in \mathbb{N}$ s.t. $|t| = n$

$$\dfrac{\dfrac{\begin{array}{cc}(\pi_b) & (\pi_s(0)) \\ \Gamma \vdash A(0) & A(0), \Gamma \vdash A(s0)\end{array}}{\Gamma \vdash A(s0)}\ \text{cut}}{\begin{array}{c}\vdots \\ \Gamma \vdash A(s^n 0)\end{array} \qquad A(s^n 0) \vdash A(t)}\ \text{cut}$$

$\Rightarrow$ Induction is infinitary cut

Simple induction proofs

No nested induction, Π_1-invariant

$$\dfrac{\dfrac{\Gamma \vdash \forall y\, F(0,y) \quad \Gamma, \forall y\, F(\nu,y) \vdash \forall y\, F(s\nu,y)}{\Gamma \vdash \forall y\, F(\alpha,y)} \qquad \Gamma, \forall y\, F(\alpha,y) \vdash B(\alpha)}{\dfrac{\Gamma \vdash B(\alpha)}{\Gamma \vdash \forall x\, B(x)}}\ \text{cut}$$

Simple induction proofs: example

- Background theory Γ:

$$
\begin{aligned}
&f(0) = 1\\
&f(sx) = sx \cdot f(x)\\
&g(x,0) = x\\
&g(x,sy) = g(x \cdot sy, y)\\
&x \cdot 1 = x\\
&1 \cdot x = x\\
&(x \cdot y) \cdot z = x \cdot (y \cdot z)
\end{aligned}
$$

- Want to prove: $g(1,\alpha) = f(\alpha)$.

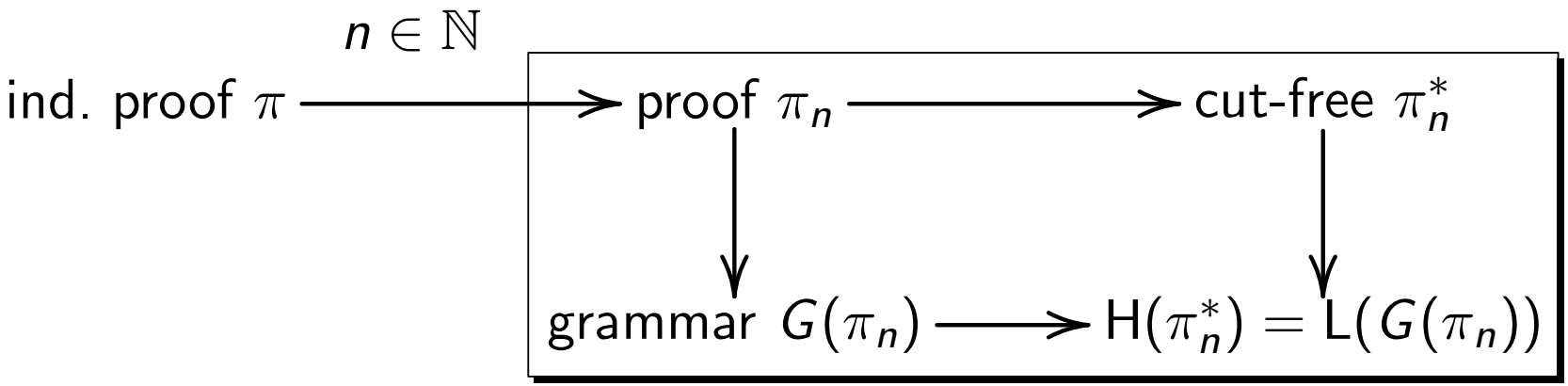
Simple induction proofs: example

- Background theory Γ:

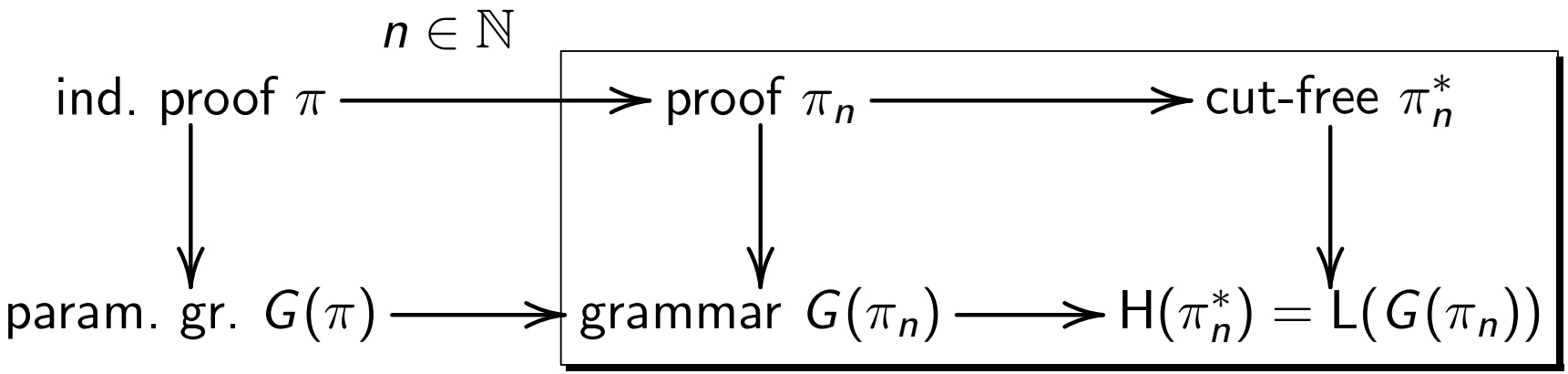
$$
\begin{aligned}
&f(0) = 1\\
&f(sx) = sx \cdot f(x)\\
&g(x,0) = x\\
&g(x,sy) = g(x \cdot sy, y)\\
&x \cdot 1 = x\\
&1 \cdot x = x\\
&(x \cdot y) \cdot z = x \cdot (y \cdot z)
\end{aligned}
$$

- Want to prove: $g(1,\alpha) = f(\alpha)$.
- Needs generalisation to: $\forall y\, g(y,\alpha) = y \cdot f(\alpha)$

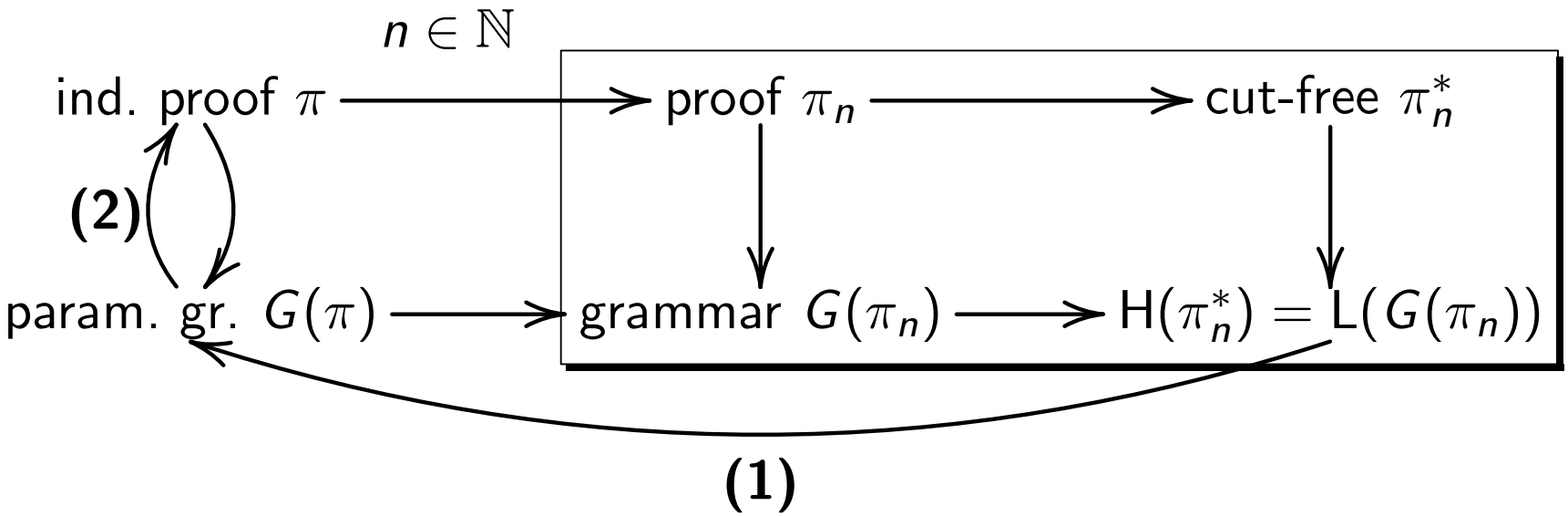
Induction and tree grammars (1/3)



Induction and tree grammars (2/3)



Induction and tree grammars (3/3)



The algorithm

Input: background theory Γ, statement $\forall x\, A(x)$

1. Compute instance proofs $\{\pi_i^* : A(i) \mid i \in M \subseteq_{\text{fin}} \mathbb{N}\}$
2. Compute parametric grammar G s.t. $L(G_i) \supseteq H(\pi_i^*)$ for $i \in M$
3. $S(X) :=$ unification problem induced by G
4. (Try to) find an F s.t. $S(F)$ is a tautology
5. $\pi :=$ proof induced by $S(F)$

Output: inductive proof π of $\Gamma \vdash \forall x\, A(x)$

Computation of grammar

- **Problem.** Given $\{H(\pi_i^*) \mid i \in M \subseteq_{\text{fin}} \mathbb{N}\}$ compute parametric grammar G s.t. $\mathrm{L}(G_i) \supseteq H(\pi_i^*)$ for all $i \in M$.
- Find a smallest such G by PTIME-translation to MaxSAT
- No coverage-guarantee for all $i \in \mathbb{N}$

Solving the unification problem

- **Problem.** Solve unification problem of the form:
  $\Gamma_0 \vdash X(0, \beta)$
  $\Gamma_1, \bigwedge_{1 \leq i \leq n} X(\nu, t_i[\alpha, \nu, \gamma]) \vdash X(s\nu, \gamma)$
  $\Gamma_2, \bigwedge_{1 \leq i \leq m} X(\alpha, u_i[\alpha]) \vdash B(\alpha)$
- Quantifier-free
- Every solution gives inductive proof
- In general: undecidable, but ...
- Complete algorithm
- Fast heuristics
- Decidable classes, e.g. parametric grammar with affine productions based on Karr's program analysis

Summary

Gist of this approach:

- Reduction of problem with quantifiers to one without
  - Compute parametric grammar based on ground terms only
  - Parametric grammar fixes many aspects of induction proof
- Rudimentary implementation finds non-analytic invariants

Towards an inductive theorem prover:

- Decidable subclasses of Boolean unification with predicates
- TIP library: many-sorted, algebraic data types
- Generalize proof shape
- Work modulo background theory

Further future work:

- Completeness
- Π_n-cuts and -induction