Beyond

Alessio Guglielmi

University of Bath and INRIA Nancy-Grand Est

6 July 2009

This talk is available at

http://cs.bath.ac.uk/ag/t/oslo-gsb.pdf

Outline

(Proof) System SKS [Brünnler & Tiu(2001)]

- Atomic rules:

$$
\mathsf{ai}\!\downarrow \frac{\mathsf{t}}{a \vee \bar{a}} \quad \mathsf{aw}\!\downarrow \frac{\mathsf{f}}{a} \quad \mathsf{ac}\!\downarrow \frac{a \vee a}{a}
$$

identity   *weakening*   *contraction*

$$
\mathsf{ai}\!\uparrow \frac{a \wedge \bar{a}}{\mathsf{f}} \quad \mathsf{aw}\!\uparrow \frac{a}{\mathsf{t}} \quad \mathsf{ac}\!\uparrow \frac{a}{a \wedge a}
$$

cut   *coweakening*   *cocontraction*

- Linear rules:

$$
\mathsf{s}\frac{\alpha \wedge [\beta \vee \gamma]}{(\alpha \wedge \beta) \vee \gamma} \qquad \mathsf{m}\frac{(\alpha \wedge \beta) \vee (\gamma \wedge \delta)}{[\alpha \vee \gamma] \wedge [\beta \vee \delta]}
$$

switch   *medial*

- Plus an '$=$' linear rule (associativity, commutativity, units).
- Rules are applied anywhere inside formulae.
- Negation on atoms only.
- Cut is atomic.
- SKS is complete and implicationally complete for propositional logic.

Example 1

- In the calculus of structures (CoS):

$$\begin{array}{r c}
 & [a \vee b] \wedge a \\ \hline
\mathsf{ac}\uparrow & [(a \wedge a) \vee b] \wedge a \\ \hline
\mathsf{ac}\uparrow & [(a \wedge a) \vee (b \wedge b)] \wedge a \\ \hline
\mathsf{ac}\uparrow & [(a \wedge a) \vee (b \wedge b)] \wedge (a \wedge a) \\ \hline
\mathsf{m} & ([a \vee b] \wedge [a \vee b]) \wedge (a \wedge a) \\ \hline
= & ([a \vee b] \wedge a) \wedge ([a \vee b] \wedge a)
\end{array}$$

- In 'Formalism A':

$$\mathsf{m}\,\dfrac{\dfrac{a}{a \wedge a} \vee \dfrac{b}{b \wedge b}}{[a \vee b] \wedge [a \vee b]} \quad \wedge \quad \dfrac{a}{a \wedge a}$$

Top-down symmetry: so inference steps can be made atomic (the medial rule, m, is impossible in the sequent calculus).

Example 2

- In CoS:

$$
=\dfrac{\mathsf{ai}\!\uparrow\dfrac{=\dfrac{\mathsf{s}\dfrac{=\dfrac{\mathsf{s}\dfrac{=\dfrac{\mathsf{m}\dfrac{=\dfrac{\mathsf{ai}\!\downarrow\dfrac{\mathsf{t}}{a \vee \bar{a}}}{(a \wedge \mathsf{t}) \vee (\mathsf{t} \wedge \bar{a})}}{[a \vee \mathsf{t}] \wedge [\mathsf{t} \vee \bar{a}]}}{[a \vee \mathsf{t}] \wedge [\bar{a} \vee \mathsf{t}]}}{([a \vee \mathsf{t}] \wedge \bar{a}) \vee \mathsf{t}}}{(\bar{a} \wedge [a \vee \mathsf{t}]) \vee \mathsf{t}}}{[(\bar{a} \wedge a) \vee \mathsf{t}] \vee \mathsf{t}}}{(a \wedge \bar{a}) \vee \mathsf{t}}}{\mathsf{f} \vee \mathsf{t}}}{\mathsf{t}}
$$

- In 'Formalism A':

$$
\mathsf{s}\dfrac{\mathsf{m}\dfrac{\dfrac{\mathsf{t}}{a \vee \bar{a}}}{[a \vee \mathsf{t}] \wedge [\mathsf{t} \vee \bar{a}]}}{\left[\mathsf{s}\dfrac{[a \vee \mathsf{t}] \wedge \bar{a}}{\dfrac{a \wedge \bar{a}}{\mathsf{f}} \vee \mathsf{t}} \quad \vee \quad \mathsf{t}\right]}
$$

Locality

- Deep inference allows locality,
- *i.e.*, inference steps can be checked in constant time (so, inference steps are small).

Example, atomic cocontraction:

$$\mathsf{m}\,\frac{\dfrac{a}{a\wedge a}\vee\dfrac{b}{b\wedge b}}{[a\vee b]\wedge[a\vee b]} \quad\wedge\quad \frac{a}{a\wedge a}$$

Note: the sequent calculus

- does not allow locality in contraction (counterexample in [Brünnler(2004)]), and
- does not allow local reduction of cut into atomic form.

Goal of This Talk

To illustrate the slogans:

- Deep inference (*i.e.*, a 'beyond') $=$ locality ($+$ symmetry).
- Locality $=$ linearity $+$ atomicity.
- geometry $=$ syntax independence ($\approx$ elimination of syntactic bureaucracy).
- Locality $\rightarrow$ geometry $\rightarrow$ semantics of proofs (Lamarche *dixit*).

This is a path towards solving the problem of proof identity, *i.e.*, determining when two proofs are the same (Hilbert's '24th problem').

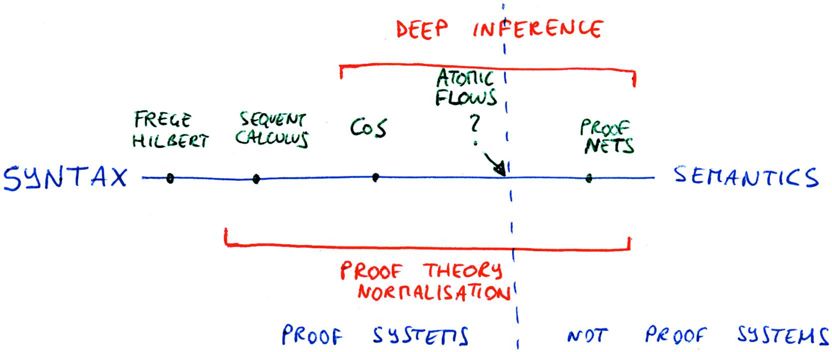
What Do We Need to Solve the Proof Identity Problem?

A finer representation of proofs, achieving locality.

This yields:

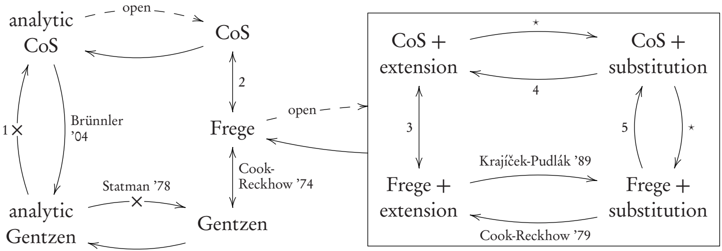
- more proofs to choose representatives from, and especially
- bureaucracy-free proofs;
- more manipulation possibilities, *viz.*, for normalisation;
- nice geometric models [Guiraud(2006)];
- smaller proofs, but
- not as small as proof nets [Lamarche & Straßburger(2005)].

Elimination of Bureaucracy



- Propositional logic.
- Proof system $\approx$ proofs can be checked in polytime.
- CoS = calculus of structures (fully developed deep inference).
- Normalisation = mainly, but not only!, cut elimination.
- Objective: eliminate bureaucracy, *i.e.*, find 'something' at the boundary.

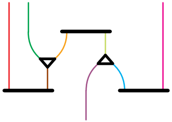
What About Proof Complexity?



Deep inference has as small proofs as the best proof systems
and
it has a normalisation theory
and
its analytic proof systems are more powerful than Gentzen ones
and
cut elimination is quasipolynomial (instead of exponential).
(See [Jeřábek(2009), Bruscoli & Guglielmi(2009),
Bruscoli et al.(2009)Bruscoli, Guglielmi, Gundersen, & Parigot]).

(Atomic) Flows

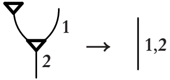
$$\mathsf{s}\,\dfrac{\mathsf{m}\,\dfrac{\dfrac{\mathsf{t}}{a \vee \bar a}}{[a \vee \mathsf{t}] \wedge [\mathsf{t} \vee \bar a]}}{\left[\mathsf{s}\,\dfrac{[a\vee \mathsf{t}]\wedge \bar a}{\dfrac{a\wedge\bar a}{\mathsf{f}} \vee \mathsf{t}} \vee \mathsf{t}\right]} = \dfrac{\left(\mathsf{s}\,\dfrac{a \wedge \left[\bar a \vee \dfrac{\mathsf{t}}{\bar a \vee a}\right]}{\dfrac{a\wedge \dfrac{\bar a \vee \bar a}{\bar a}}{\mathsf{f}} \vee \dfrac{a}{a\wedge a}} \wedge \bar a \right)}{a \wedge \dfrac{a\wedge \bar a}{\mathsf{f}}} \qquad \mathsf{m}\,\dfrac{\dfrac{a}{a\wedge a} \vee \dfrac{b}{b\wedge b}}{[a\vee b]\wedge[a\vee b]} \wedge \dfrac{a}{a \wedge a}$$

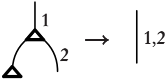


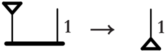
- Below derivations, their (atomic) flows are shown.
- Only structural information is retained in flows.
- Logical information is lost.
- Flow size is polynomially related to derivation size.

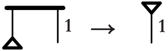
Flow Reductions: (Co)Weakening (1)

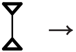
Consider these flow reductions:

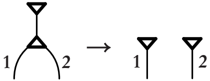
aw↓-ac↓: 

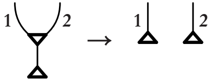
ac↑-aw↑: 

aw↓-ai↑: 

ai↓-aw↑: 

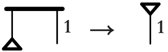
aw↓-aw↑: 

aw↓-ac↑: 

ac↓-aw↑: 

Each of them corresponds to a correct derivation reduction.

Flow Reductions: (Co)Weakening (2)

For example, ai↓-aw↑:  specifies that

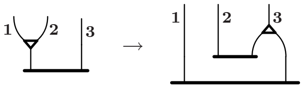
$$\begin{array}{c} \Pi'' \Vert \\ \xi\left\{\dfrac{\mathsf{t}}{a^{\epsilon} \vee \bar{a}}\right\} \\ \Phi \Vert \\ \zeta\left\{\dfrac{a^{\epsilon}}{\mathsf{t}}\right\} \\ \Psi \Vert \\ \alpha \end{array} \qquad \text{becomes} \qquad \begin{array}{c} \Pi'' \Vert \\ \xi\left[\mathsf{t} \vee \dfrac{\mathsf{f}}{\bar{a}}\right] \\ \Phi\{a^{\epsilon}/\mathsf{t}\} \Vert \\ \zeta\{\mathsf{t}\} \\ \Psi \Vert \\ \alpha \end{array}$$

We can operate on flow reductions instead than on derivations: it is much easier and we get natural, syntax-independent induction measures.

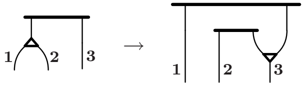
Flow Reductions: (Co)Contraction

Consider these flow reductions:

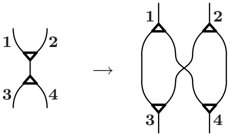
c↓-i↑:



i↓-c↑:



c↓-c↑:



- They conserve the number and length of paths.
- Note that they can blow up a derivation exponentially.
- It's a good thing: cocontraction is a new compression mechanism (sharing?).
- Open problem: does cocontraction provide exponential compression? Conjecture: yes.

Normalisation Overview

SYMMETRIC GENERALISATION

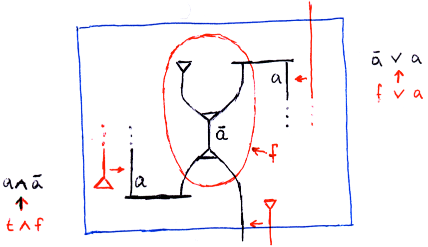
	CUT ELIMINATION	STREAMLINING
EXPONENTIAL	- SIMPLE 'EXPERIMENTS'	- 'OPTIMISABLE' PROCEDURE (1) - BY THE 'NORMALISER' (2)
QUASI POLYNOMIAL (I.E., $n^{O(\log n)}$)	- BY 'THRESHOLD FUNCTIONS' (3)	? (WORK IN PROGRESS)

- None of these methods existed before atomic flows, none of them requires permutations or other syntactic devices.
- Quasipolynomial procedures are surprising.
- Conjecture: polynomial normalisation is possible.

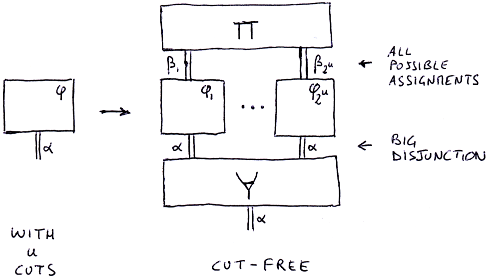
(1) [Guglielmi & Gundersen(2008)]; (2) work in progress; (3) [Bruscoli et al.(2009)Bruscoli, Guglielmi, Gundersen, & Parigot].

Cut Elimination (on Proofs) by 'Experiments'

Experiment:

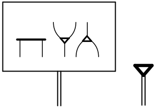
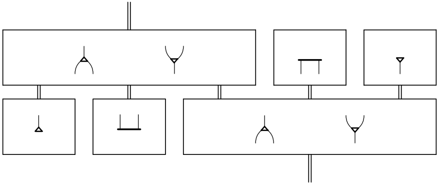


We do:



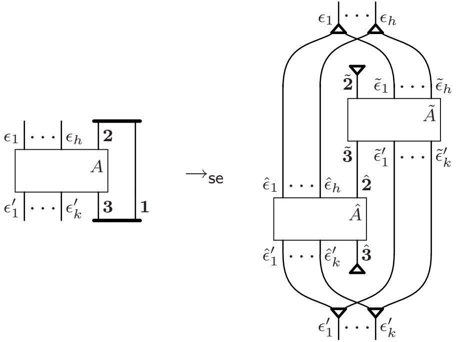
Simple, exponential cut elimination; proof generates 2^n experiments.

Generalising the Cut-Free Form

- Normalised proof: 
- Normalised derivation: 
- The symmetric form is called streamlined.
- Cut elimination is a corollary of streamlining.

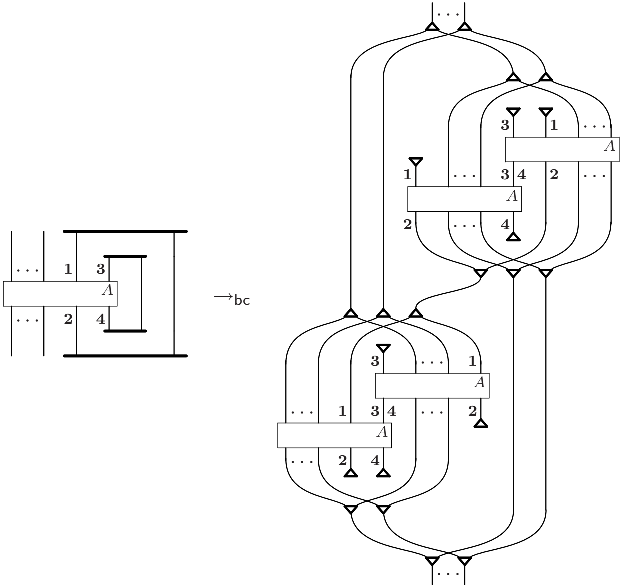
Removal of a 'Simple Edge'

Remove identity and cut:



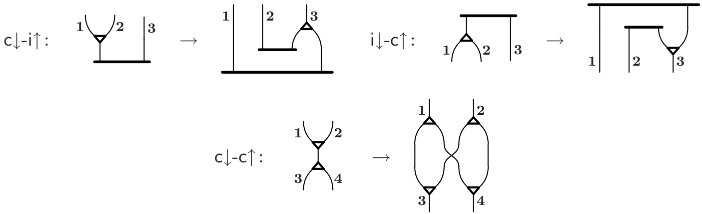
- We can do so on simple edges, like $\mathbf{1}$ above.
- The procedure requires a strategy, not to loop.
- The chunks to be copied can be small.
- Open: computational interpretation?

Composition of Simple Edge Removal

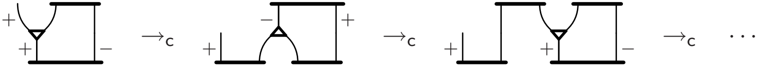


How to Obtain a Simple Edge?

- By moving away (co)contractions by way of their reductions:

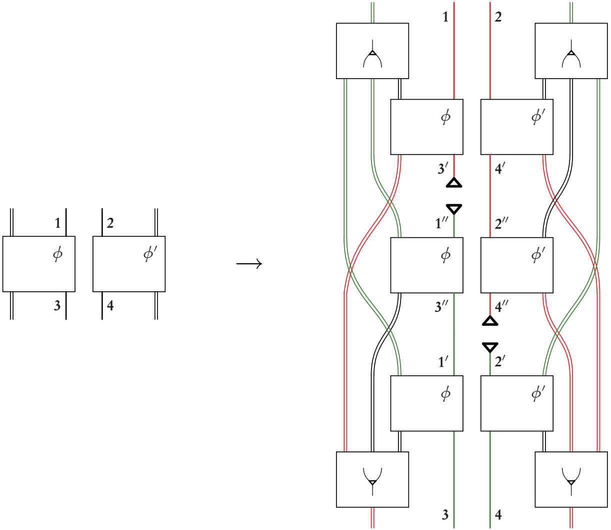


- But beware of loops:



- This and more is in [Guglielmi & Gundersen(2008)].

How Do We Break Paths Without ‘Preprocessing’?



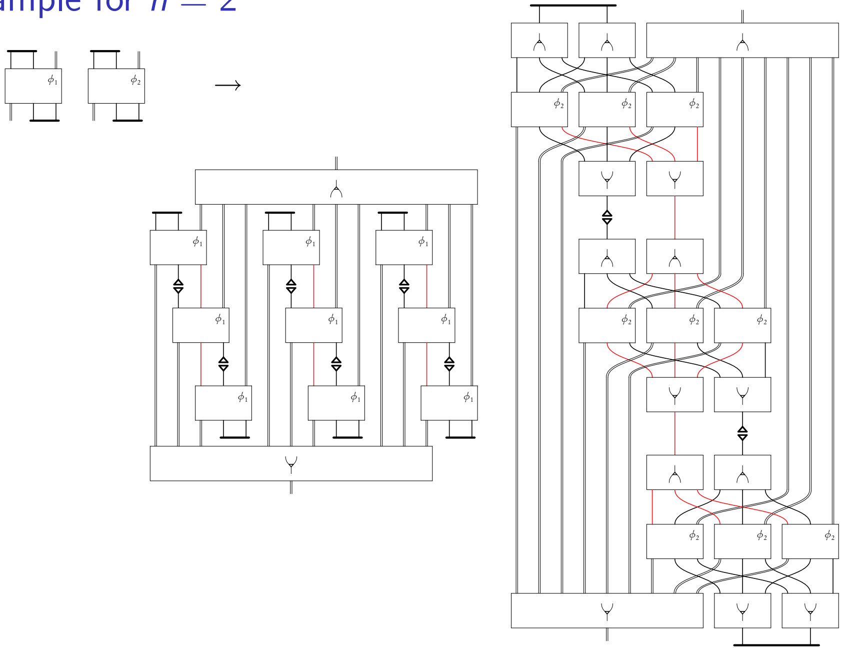
Even if there is a path between **1** and **3** on the left, there is none on the right (and the same for **2** and **4**).

We Can Do This on Derivations, of Course

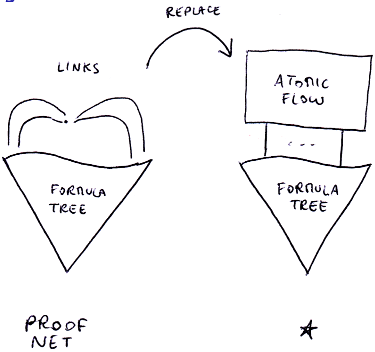
$$
\Phi = \begin{array}{c}[a \vee \bar{a}] \wedge \alpha \\ \Vert \\ \beta \vee (a \wedge \bar{a})\end{array}
\qquad \longrightarrow \qquad
\mathsf{Break}\,\Phi =
{=}\dfrac{[a \vee \bar{a}] \wedge \dfrac{\alpha}{\alpha \wedge \alpha \wedge \alpha}}
{{\mathsf{s}}\dfrac{\left(\begin{array}{c}[a \vee \bar{a}] \wedge \alpha \\ \Phi \Vert \\ \left[\beta \vee \left(\dfrac{a}{\mathsf{t}} \wedge \bar{a}\right)\right]\end{array} \wedge \alpha \wedge \alpha\right)}
{{\mathsf{s}}\dfrac{\left(\left[\beta \vee \begin{array}{c}\left(\left[\dfrac{\mathsf{f}}{a} \vee \bar{a}\right] \wedge \alpha\right) \\ \Phi \Vert \\ \left[\beta \vee \left(a \wedge \dfrac{\bar{a}}{\mathsf{t}}\right)\right]\end{array}\right] \wedge \alpha\right)}
{{=}\dfrac{\left[\beta \vee \beta \vee \begin{array}{c}\left(\left[a \vee \dfrac{\mathsf{f}}{\bar{a}}\right] \wedge \alpha\right) \\ \Phi \Vert \\ \beta \vee (a \wedge \bar{a})\end{array}\right]}
{\dfrac{\beta \vee \beta \vee \beta}{\beta} \vee (a \wedge \bar{a})}}}}
$$

- We can compose Break as many times as there are paths between identities and cut.
- We obtain a family of normalisers that only depends on n.
- The construction is exponential.
- Note: finding something like this is *unthinkable* without flows.

Example for $n = 2$



Conjecture



- We think that (*) might make for a proof system (see also recent work by Straßburger).
- This means that there should exist a polynomial algorithm to check the correctness of (*).
- If this is true, we have an excellent bureaucracy-free formalism.
- Note: if such a thing existed for proof nets, then coNP $=$ NP.

Conclusion

- (Exponential) normalisation does not depend on logical rules.
- It only depends on structural information, *i.e.*, geometry.
- Normalisation is extremely robust.
- Deep inference's locality is key.
- Complexity-wise, deep inference is as powerful as the best formalisms,
- and more powerful if analiticity is requested.
- Deep inference is the continuation of Girard politics with other means.

In my opinion, much of the future of structural proof theory is in geometric methods: we have to free ourselves from the tyranny of syntax (so, war to bureaucracy!).

This talk is available at
`http://cs.bath.ac.uk/ag/t/oslo-gsb.pdf`

References

Brünnler, K. (2004).
Deep Inference and Symmetry in Classical Proofs.
Berlin: Logos Verlag.
http://www.iam.unibe.ch/~kai/Papers/phd.pdf.

Brünnler, K., & Tiu, A. F. (2001).
A local system for classical logic.
In R. Nieuwenhuis, & A. Voronkov (Eds.) *LPAR 2001*, vol. 2250 of *Lecture Notes in Computer Science*, (pp. 347–361).
Springer-Verlag.
http://www.iam.unibe.ch/~kai/Papers/lcl-lpar.pdf.

Bruscoli, P., & Guglielmi, A. (2009).
On the proof complexity of deep inference.
ACM Transactions on Computational Logic, *10*(2), 1–34.
Article 14. http://cs.bath.ac.uk/ag/p/PrComplDI.pdf.

Bruscoli, P., Guglielmi, A., Gundersen, T., & Parigot, M. (2009).
Quasipolynomial normalisation in deep inference via atomic flows and threshold formulae.
Submitted. http://cs.bath.ac.uk/ag/p/QuasiPolNormDI.pdf.

Guglielmi, A., & Gundersen, T. (2008).
Normalisation control in deep inference via atomic flows.
Logical Methods in Computer Science, *4*(1:9), 1–36.
http://www.lmcs-online.org/ojs/viewarticle.php?id=341.

Guiraud, Y. (2006).
The three dimensions of proofs.
Annals of Pure and Applied Logic, *141*(1-2), 266–295.
http://www.loria.fr/~guiraudy/recherche/cos1.pdf.

Jeřábek, E. (2009).
Proof complexity of the cut-free calculus of structures.
Journal of Logic and Computation, *19*(2), 323–339.
http://www.math.cas.cz/~jerabek/papers/cos.pdf.

Lamarche, F., & Straßburger, L. (2005).
Naming proofs in classical propositional logic.
In P. Urzyczyn (Ed.) *Typed Lambda Calculi and Applications*, vol. 3461 of *Lecture Notes in Computer Science*, (pp. 246–261).
Springer-Verlag.
http://www.lix.polytechnique.fr/~lutz/papers/namingproofsCL.pdf.