Parsifal[1]
"To Correctness through Proof"

Dale Miller (Team Leader) and
Kaustuv Chaudhuri, Joëlle Despeyroux, Stéphane Lengrand, Lutz Straßburger
plus 5 PhD students and a postdoc

INRIA-Saclay & LIX/École Polytechnique
Palaiseau, France

INRIA Evaluation Seminar, Paris, 23 March 2011

[1] Preuves Automatiques et Raisonnement sur des SpécIFicAtions Logiques

Outline

Vision

Peter Andrews selected the subtitle

"To Truth through Proof"

to his textbook

Vision

Peter Andrews selected the subtitle

"To Truth through Proof"

to his textbook because
"in mathematics the primary and ultimate tool for establishing truth is logic."

$$\vdash A \qquad \text{implies} \qquad \models A$$

Vision

Peter Andrews selected the subtitle

"To Truth through Proof"

to his textbook because
"in mathematics the primary and ultimate tool for establishing truth is logic."

$$\vdash A \qquad \text{implies} \qquad \models A$$

For the Information Age, we have fashioned the slogan

"To Correctness through Proof"

Vision

Peter Andrews selected the subtitle

"To Truth through Proof"

to his textbook because
"in mathematics the primary and ultimate tool for establishing truth is logic."

$$\vdash A \qquad \text{implies} \qquad \models A$$

For the Information Age, we have fashioned the slogan

"To Correctness through Proof"

- Various *artifacts* (*i.e.*, programming languages, type systems, programs, computation traces, protocols, *etc.*) are our focus.
- Proofs relate in various ways to their *correctness*.

$$\vdash P \ : \ A$$

Vision

Peter Andrews selected the subtitle

"To Truth through Proof"

to his textbook because
"in mathematics the primary and ultimate tool for establishing truth is logic."

$$\vdash A \qquad \text{implies} \qquad \models A$$

For the Information Age, we have fashioned the slogan

"To Correctness through Proof"

- Various *artifacts* (*i.e.*, programming languages, type systems, programs, computation traces, protocols, *etc.*) are our focus.
- Proofs relate in various ways to their *correctness*.

$$\vdash P \ : \ A$$

We exploit and develop *structural proof theory* (*a la* Gentzen, Girard, . . .) to provide rich properties of syntactic systems.

Outline

From the 2007 Parsifal proposal

From the 2007 Parsifal proposal

“

The Parsifal project will exploit recent developments in

proof search, logic programming, and type theory

From the 2007 Parsifal proposal

“

The Parsifal project will exploit recent developments in

proof search, logic programming, and type theory

to make the specification of

operational semantics more expressive and declarative

From the 2007 Parsifal proposal

“

The Parsifal project will exploit recent developments in

proof search, logic programming, and type theory

to make the specification of

operational semantics more expressive and declarative

and will develop techniques and tools for

animating and reasoning directly on logic-based specifications.

”

The two-level logic approach to reasoning

computational artifacts

e.g. λ-calculus, π-calculus, PCF, ...

The two-level logic approach to reasoning

Example:

$$\text{PAR}: \quad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \quad \mathrm{bn}(\alpha) \cap \mathrm{fn}(Q) = \emptyset$$

$$\text{COM}: \quad \frac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'\{y/z\}} \qquad \text{CLOSE}: \quad \frac{P \xrightarrow{\overline{x}(w)} P' \quad Q \xrightarrow{x(w)} Q'}{P \mid Q \xrightarrow{\tau} (w)(P' \mid Q')}$$

$$\text{RES}: \quad \frac{P \xrightarrow{\alpha} P'}{(y)P \xrightarrow{\alpha} (y)P'} \quad y \notin \mathrm{n}(\alpha) \qquad \text{OPEN}: \quad \frac{P \xrightarrow{\overline{x}y} P'}{(y)P \xrightarrow{\overline{x}(w)} P'\{w/y\}} \quad \begin{array}{l} y \neq x \\ w \notin \mathrm{fn}((y)P') \end{array}$$

A few operational semantic rules taken from Milner, Parrow & Walker, “A Calculus of Mobile Processes, Part II” (1989)

computational artifacts

e.g. λ-calculus, π-calculus, PCF, . . .

The two-level logic approach to reasoning

Example:

$$\text{PAR}: \quad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \quad \mathrm{bn}(\alpha) \cap \mathrm{fn}(Q) = \emptyset$$

$$\text{COM}: \quad \frac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'\{y/z\}} \qquad \text{CLOSE}: \quad \frac{P \xrightarrow{\overline{x}(w)} P' \quad Q \xrightarrow{x(w)} Q'}{P \mid Q \xrightarrow{\tau} (w)(P' \mid Q')}$$

$$\text{RES}: \quad \frac{P \xrightarrow{\alpha} P'}{(y)P \xrightarrow{\alpha} (y)P'} \quad y \notin \mathrm{n}(\alpha) \qquad \text{OPEN}: \quad \frac{P \xrightarrow{\overline{x}y} P'}{(y)P \xrightarrow{\overline{x}(w)} P'\{w/y\}} \quad \begin{array}{l} y \neq x \\ w \notin \mathrm{fn}((y)P') \end{array}$$

A few operational semantic rules taken from Milner, Parrow & Walker, "A Calculus of Mobile Processes, Part II" (1989)

We wish to formalize and prove strong properties:

- reachability, model-checking
- subject-reduction (type preservation)
- bisimulation is a congruence

computational artifacts

e.g. λ-calculus, π-calculus, PCF, ...

The two-level logic approach to reasoning

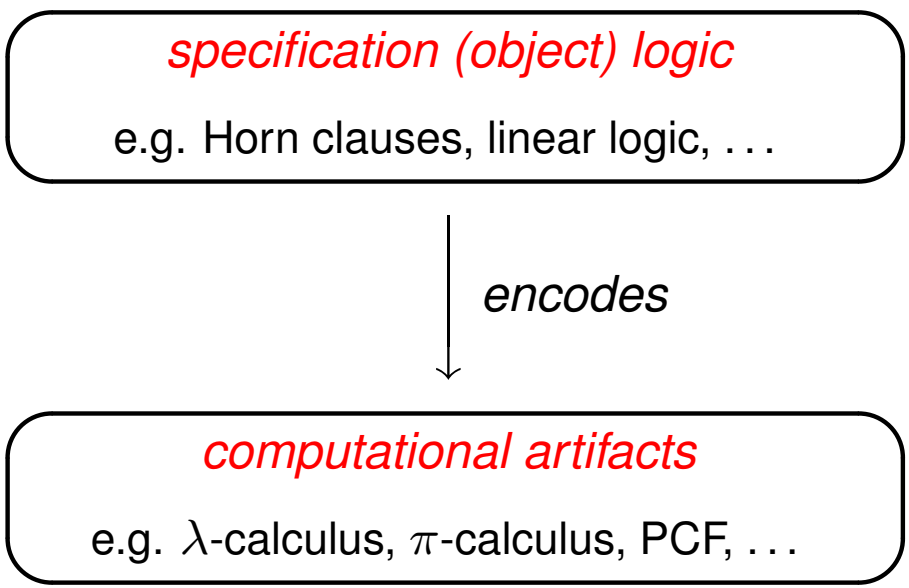
We wish to formalize and prove strong properties:

- reachability, model-checking
- subject-reduction (type preservation)
- bisimulation is a congruence

computational artifacts

e.g. λ-calculus, π-calculus, PCF, ...

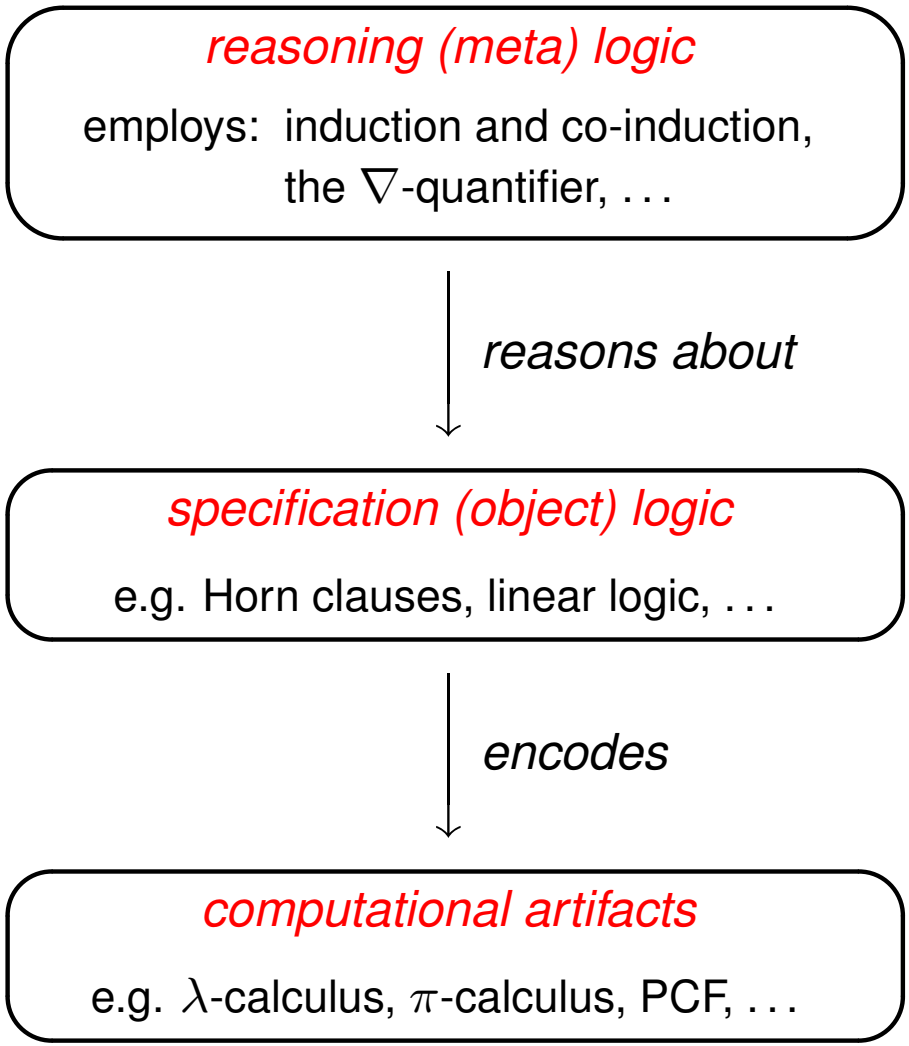
The two-level logic approach to reasoning



We wish to formalize and prove strong properties:

- reachability, model-checking
- subject-reduction (type preservation)
- bisimulation is a congruence

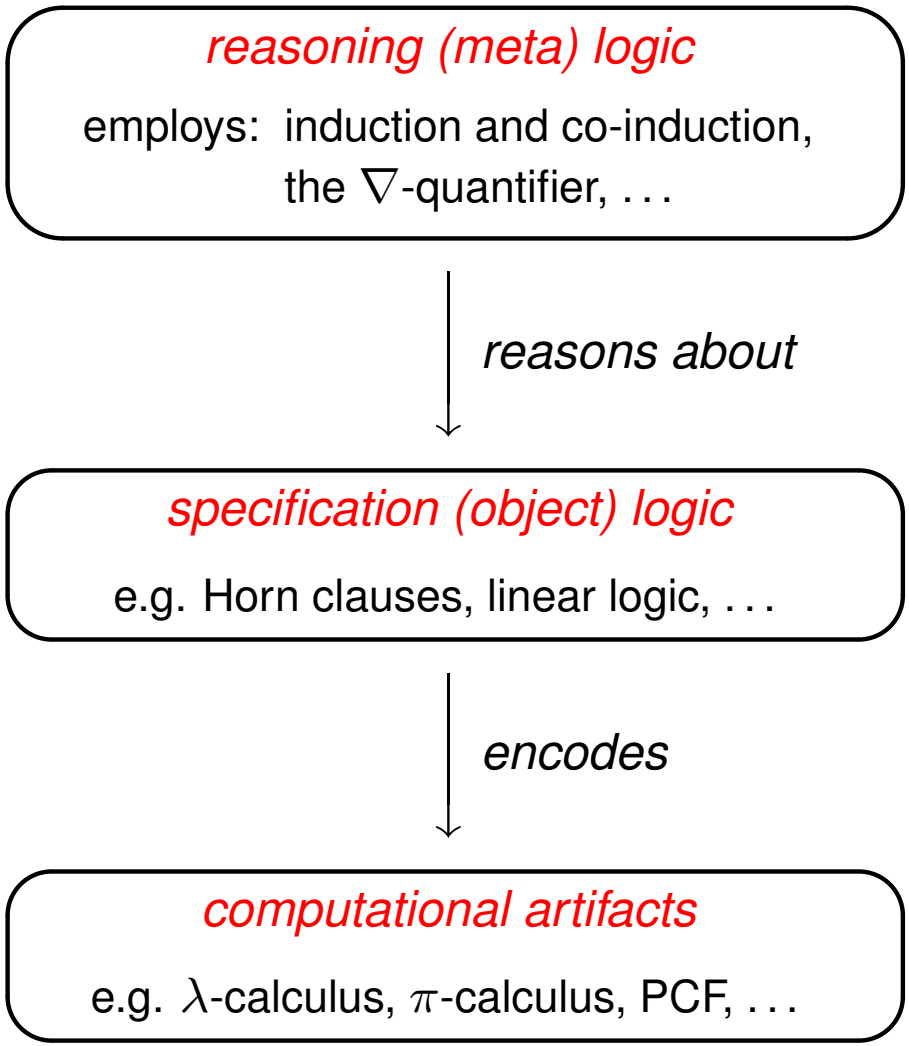
The two-level logic approach to reasoning



We wish to formalize and prove strong properties:

- reachability, model-checking
- subject-reduction (type preservation)
- bisimulation is a congruence

The two-level logic approach to reasoning



We **can** formalize and prove strong properties:

- reachability, model-checking
- subject-reduction (type preservation)
- bisimulation is a congruence

Bedwyr: a model checker

Bedwyr is a completely automatic implementation of a fragment of the "reasoning logic."

- It implements the ∇-quantifier and proof search via the unfolding of fixed points.
- It can be used as a model checker for linguistic expressions, possibly containing bound variables.
- Implemented by Baelde (Parsifal PhD student) and Gacek (Parsifal intern).

Bedwyr provides an entirely declarative model checker for the (finite) π-calculus.

Collaborators: Gacek & Nadathur (U. Minnesota), Tiu (Australian National University)
Funding: INRIA Associate Team Slimmer, NSF.
Pubs: CADE07, CSL07, LFMTP08, Tableaux09

Abella: an interactive, two-level logic prover

Abella is an *interactive* theorem prover for the *full reasoning logic* and for one specific specification logic.

Implemented by Gacek (PhD, U. Minnesota; postdoc, Parsifal).

Examples (many contributed by users):

- POPLmark challenge: Part 1a and Part 2a
- Church-Rosser theorem
- weak and strong normalization of the simply-typed λ-calculus
- strong normalization for a variant of the $\lambda\sigma$-calculus
- some of the π-calculus meta-theory
- correctness of a compiler from an Esterel-like language to C

Collaborators: Abel (LMU Munich), Pollack (Edinburgh), Schack-Nielsen (ITU, Copenhagen), Tiu (Australian National University), Wilson (California State University)
Funding: INRIA Associate Team Slimmer, NSF.
Pubs: LICS08, LFMTP08, PPDP10, APLAS10, JAR 2010, I&C 2011

Outline

Focusing: the chemistry behind inference

Complete (focused) proof search involves alternating between two phases.

- In *logic programming*: “goal-reduction” and “backchaining” (1987).
- In *linear logic*: “invertible” and “non-invertible” phases (Andreoli, 1991).

Focusing provides a “chemistry” for inference.

- Gentzen’s introduction rules are the *atoms of inference*.
- Focusing provides the *rules of chemistry*: some atoms can stick together; others cannot go together.
- The result yields new *molecules of inference* (sometimes big phases).
- This chemistry is *flexible* and allows a range of *engineering* possibilities.

Focusing: new systems

The team has embraced “focused proof systems” in a strong way.

- focused proofs systems for classical (LKF) and intuitionistic (LJF) logics: these account for all previous focusing systems (LJT, LJQ, λRCC, *etc.*)
- *maximal multi-focusing:* capturing parallelism in proofs: *e.g.*, abstracting sequent calculus to obtain proof nets
- Focused proof system fixed points: a new approach to *mixing computation with deduction*.

Collaborators: Liang (Hofstra University, NY),
Funding: FP6 Mobius; INRIA Associate Team Slimmer.
Pubs: CSL07/10, LICS08/09, JAR 2008/2010, IJCAR08, PPDP09, TCS 2009, LPAR10

Focusing: rethinking unbounded behavior in logic

MALL is the core of linear logic, but it is decidable.

Girard: Logic is MALL plus exponentials (!,?): yields linear logic.

- But exponentials keep molecules from being large.

Focusing: rethinking unbounded behavior in logic

MALL is the core of linear logic, but it is decidable.

Girard: Logic is MALL plus exponentials (!,?): yields linear logic.

- But exponentials keep molecules from being large.

Parsifal: Logic is MALL plus fixed points (μ, ν): yields μ*MALL*.

- molecules in μ*MALL* can become arbitrarily large.

Focusing: rethinking unbounded behavior in logic

MALL is the core of linear logic, but it is decidable.

Girard: Logic is MALL plus exponentials (!,?): yields linear logic.

- But exponentials keep molecules from being large.

Parsifal: Logic is MALL plus fixed points (μ, ν): yields μ*MALL*.

- molecules in μ*MALL* can become arbitrarily large.
- Restricting μ*MALL* yields an intuitionistic logic: μ*LJ*.
- μ*LJ* captures many aspects of model checking.
- μ*LJ* is the foundation for **Bedwyr**.

Pubs: LPAR07, LICS08, Tableaux09, APAL 2010, ToCL 2011

Outline

From Hilbert to Gentzen

Many (substructural) logics are only given as Hilbert systems

- not suitable for proof search

Obtaining an equivalent Gentzen system suitable for proof search is difficult.

Question: Can we automatize this process?

Collaborators: Ciabattoni (Vienna), Terui (Kyoto)
Funding: PHC Amadeus
Pubs: CSL09

From Hilbert to Gentzen

Many (substructural) logics are only given as Hilbert systems

- not suitable for proof search

Obtaining an equivalent Gentzen system suitable for proof search is difficult.

Question: Can we automatize this process?

Answer: **Yes.**
A certain class of Hilbert axioms can be transformed into structural rules preserving cut elimination.

Collaborators: Ciabattoni (Vienna), Terui (Kyoto)
Funding: PHC Amadeus
Pubs: CSL09

Deep inference

Deep inference provides a different approach to the *atoms of inference* with different chemistry rules: interactions can occur deep inside a formula.

This framework provides

- new approaches to non-commutative logic
- a modular treatment of various modals logics
- a new understanding of parallelism in proofs
- a uniform treatment of methods of proof compression

Collaborators: Brünnler (Bern), Guglielmi (Bath), Bruscoli (Bath), Gundersen (PPS), Hetzl (PPS)
Funding: ANR blanc “INFER”, ARC “REDO”, PHC Germaine de Staël
Pubs: RTA07, Tableaux09, TLCA09, JLC 2009, MSCS 2010, ToCL 2010

Proof Nets and Atomic Flows

Find canonical representations of proofs that

- reduce bureaucracy (no rule permutation)
- capture the “essence” of proof
- allow new proof transformations and normal forms

Collaborators: Lamarche (Nancy), Guglielmi (Bath), Gundersen (PPS)
Funding: ANR blanc “INFER”, ARC “REDO”
Pubs: TAC07, LICS10, JLC 2009.

Proof Nets and Atomic Flows

Find canonical representations of proofs that

- reduce bureaucracy (no rule permutation)
- capture the “essence” of proof
- allow new proof transformations and normal forms

Example:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\bar{b} \vee a}{\bar{b} \vee ((\bar{b} \vee b) \wedge a)}\,\mathsf{ai}{\downarrow}}{\bar{b} \vee \bar{b} \vee (b \wedge a)}\,\mathsf{s}}{\bar{b} \vee (b \wedge a)}\,\mathsf{ac}{\downarrow}}{(\bar{b} \wedge \bar{b}) \vee (b \wedge a)}\,\mathsf{ac}{\uparrow}}{(\bar{b} \wedge (a \vee \bar{a}) \wedge \bar{b}) \vee (b \wedge a)}\,\mathsf{ai}{\downarrow}}{(\bar{b} \wedge (a \vee (\bar{a} \wedge \bar{b}))) \vee (b \wedge a)}\,\mathsf{s}}{((\bar{b} \wedge a) \vee (\bar{a} \wedge \bar{b})) \vee (b \wedge a)}\,\mathsf{s}$$

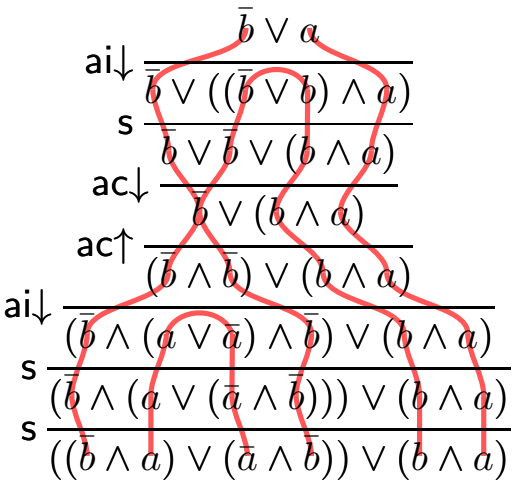
Collaborators: Lamarche (Nancy), Guglielmi (Bath), Gundersen (PPS)
Funding: ANR blanc “INFER”, ARC “REDO”
Pubs: TAC07, LICS10, JLC 2009.

Proof Nets and Atomic Flows

Find canonical representations of proofs that

- reduce bureaucracy (no rule permutation)
- capture the “essence” of proof
- allow new proof transformations and normal forms

Example:



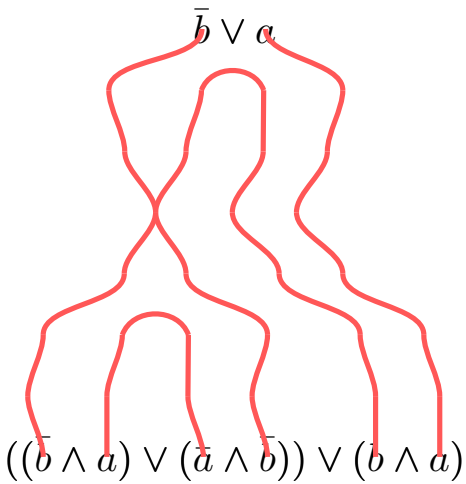
Collaborators: Lamarche (Nancy), Guglielmi (Bath), Gundersen (PPS)
Funding: ANR blanc “INFER”, ARC “REDO”
Pubs: TAC07, LICS10, JLC 2009.

Proof Nets and Atomic Flows

Find canonical representations of proofs that

- reduce bureaucracy (no rule permutation)
- capture the “essence” of proof
- allow new proof transformations and normal forms

Example:



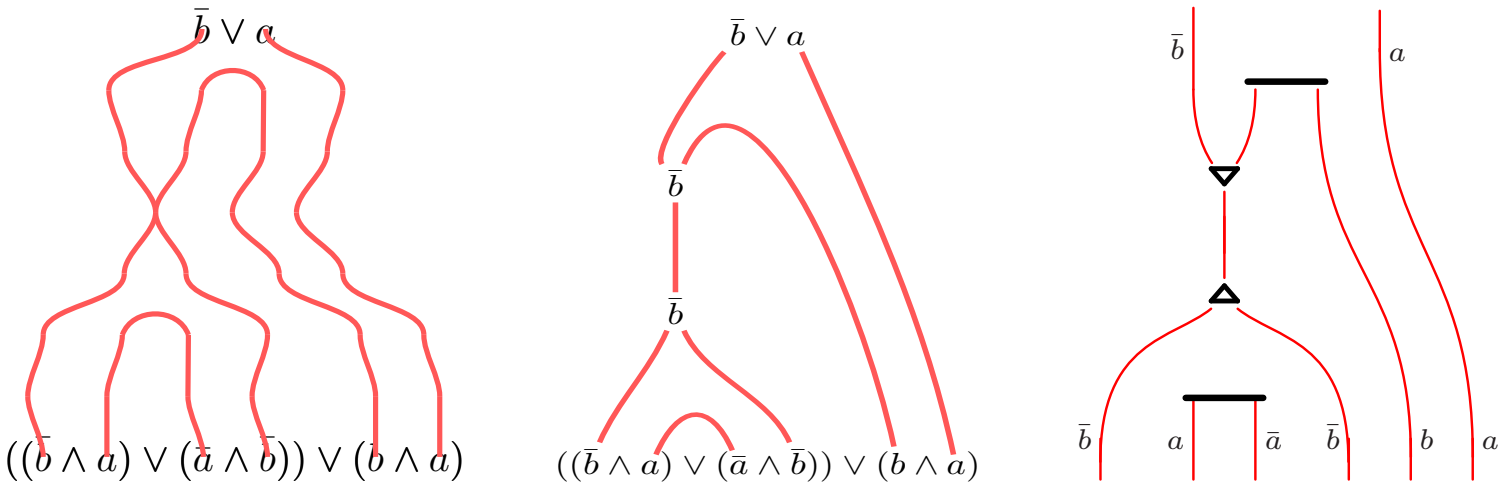
Collaborators: Lamarche (Nancy), Guglielmi (Bath), Gundersen (PPS)
Funding: ANR blanc “INFER”, ARC “REDO”
Pubs: TAC07, LICS10, JLC 2009.

Proof Nets and Atomic Flows

Find canonical representations of proofs that

- reduce bureaucracy (no rule permutation)
- capture the “essence” of proof
- allow new proof transformations and normal forms

Example:



Collaborators: Lamarche (Nancy), Guglielmi (Bath), Gundersen (PPS)
Funding: ANR blanc “INFER”, ARC “REDO”
Pubs: TAC07, LICS10, JLC 2009.

Outline

Improve theorem proving capabilities

The team is involved with four different theorem provers.

- λProlog: automated, logic programming
- Bedwyr: automated, model checking
- Abella: interactive
- Tac (prototype): automatic inductive theorem proving

Our theorem proving ambitions include:

- merging the implementations of Bedwyr, Abella, and Tac since they implement roughly the same logic, and
- improve the integration and control of SMT (satisfiability modulo theories) within theorem provers.

Communicating and trusting proofs

We live with many programming languages.

Must we live with many different proof structures?

One theorem prover's proofs are unusable to another prover (even a later version of the same prover).

There are numerous efforts addressing the exchange of proofs between various pairs of provers.

Communicating and trusting proofs

We live with many programming languages.

Must we live with many different proof structures?

One theorem prover's proofs are unusable to another prover (even a later version of the same prover).

There are numerous efforts addressing the exchange of proofs between various pairs of provers.

Focused proof systems provide an exciting and *foundational* approach to a broad spectrum of *proof certificates*.

- A universal proof certificate checker needs to know the "atoms of inference" and the "rules of chemistry." These are few and fixed.
- The certificate describes the needed molecules and then sends only the high-level molecular description of proof.

Pubs: ACM-BCS Vision 2010

Continued research into proof theory

Computational complexity trade-offs between proof size and proof checking.

Balancing the split between computation and deduction within proofs.

New techniques for proof compression and for proof reconstruction (*e.g.*, unification).

Expand our understanding and uses of focused proof systems.

Positioning

International

Systems implementation: Australian National University, Carnegie Mellon, University of McGill, University of Minnesota

Proof theory: Hofstra University (NY, USA), RIMS Kyoto University, Technical University of Vienna, University of Bath, University of Bern, University of Bologna

National

PPS (Paris VII) various proof theory topics

TypiCaL (INRIA Saclay) Proof certificates, computation vs deduction, SMT integration

Calligramme, Pareo (INRIA, Nancy) Deduction modulo, proof theory

Self assessment

- We consider the research into two-level logic and its tools to be highly successful: we covered theory, design, implementation, and applications.
- Our research efforts into the foundations of proof theory provide us with novel designs and implemented systems: *e.g.,* focused proof systems and the ∇-quantifier.
- Our implemented systems remain about the size of one PhD: we need to move to multiple year implementation efforts.

Highlights

- *PhD award:* Alexis Saurin's thesis won the "Prix de thèse de l'Ecole Polytechnique" and the "Prix de thèse ASTI 2009."
- *Fellowship:* Vivek Nigam (PhD 9/2009) was awarded an Alexander von Humboldt scholarship for LMU (Munich, Germany) 2010/2012.
- *Invited talks:* Logic, Methodology, and Philosophy of Science 2011, APLAS 2010 (Shanghai), FICS 2010 (Brno), SOS 2008 (Reykjavik), plus 9 others.
- *Invited tutorials:* International School on Computational Logic, Italy (April 2011); 8th Panhellenic Logic Symposium, Greece (July 2011).

Questions ?