On the Role of Proof Theory in Automated Deduction

Frank Pfenning

Department of Computer Science
Carnegie Mellon University

25th International Conference on Automated Deduction
(CADE-25)
Special Session on the Past, Present and
Future of Automated Deduction
Berlin, Germany
August 3, 2015

Model-Theoretic Semantics

- Theorem proving by refutation of negation
- Can be understood as attempt to construct a model
  - Highly successful for classical logic (Herbrand models)
  - Applied also to nonclassical logics (Kripke models)
- Requires deep understanding of model theory
- Proof extraction may be difficult

Proof-Theoretic Semantics

- Theorem proving by direct proof construction
  - Immediately applicable to nonclassical logics
- Requires deep understanding of proof theory
- Proof is primary artifact

Applications of Nonclassical Logics in CS

- A personal and biased sampling
- Propositions as types, proofs as programs
  - Intuitionistic logic and type theory [Martin-Löf'80]
  - Staged computation, run-time code generation (JS4) [Davies & Pf'96]
  - Monadic encapsulation (lax logic) [Fairtlough & Mendler'97]
  - Partial evaluation (temporal logic) [Davies'96]
  - Message-passing concurrency (linear logic) [Caires & Pf'10] [Toninho'15]
- Reasoning about programs
  - Dynamic logic [Pratt'74]
  - Temporal logics [Pnueli'77] [Clarke & Emerson'80]
  - Separation logic [O'Hearn & Pym'99] [Reynolds'02]
- Security
  - Authorization logics [Garg et al.'06]
  - Protocol logics [Datta et al.'03]

Constructing a Theorem Prover

1. Present a logic as a deductive system amenable to search
2. Iterate
   - Devise an equivalent system with less nondeterminism
     - Don't-know nondeterminism: reduce backtracking
     - Don't-care nondeterminism: reduce redundancy
3. Exploit techniques for efficient implementation

Outline

- Past: How to define a logic
  - Sequent calculus [Gentzen'35]
  - Harmony [Dummett'76] [Martin-Löf'83]
- Present: How to reduce nondeterminism in search
  - Focusing and polarization [Andreoli'92] [Laurent'99]
- Future: How to combine logics
  - Adjunctions [Benton'94] [Reed'09]

Running Example: Linear Logic

- Logical hypotheses as resources [Girard'87]
- Exemplify techniques in a deceptively simple setting
- Model-theoretic techniques not easily available
- Many applications in computer science
  - Planning as (linear) theorem proving [Bibel'85]
  - Close cousin to separation logic [Reynolds'02]
  - Quantum computation [van Tonder'03]
  - Message-passing concurrency [Caires & Pf'10] [Toninho'15]

Logic Definition: Proof-Theoretic Semantics

- Gerhard Gentzen [1935]

  > *The introduction [rules of natural deduction] represent, as it were, the 'definitions' of the symbols concerned, and the eliminations are no more, in the final analysis, than the consequences of these definitions.*

- In a slight departure, we use his sequent calculus
- Antecedents $\Delta ::= \bullet \mid \Delta, A$ (modulo exchange)
- Linear hypothetical judgment

$$\Delta \vdash A$$

Using antecedents in Δ exactly once,
we can prove that A is true

Global Harmony Requirement

- Use justifies truth (id)
- Truth justifies use (cut)

$$\frac{}{A \vdash A}\ \mathrm{id}_A \qquad \frac{\Delta_1 \vdash A \quad \Delta_2, A \vdash C}{\Delta_1, \Delta_2 \vdash C}\ \mathrm{cut}_A$$

- Cut elimination: Any deduction can be transformed into one not using the rule of cut
- Identity elimination: Any deduction can be transformed into one using identity id_a only for atomic propositions a

Local Harmony Requirement

- Right rules for connectives define how to prove them
- Left rules for connectives define how to use them
- Example: linear implication $A \multimap B$

$$\frac{\Delta, A \vdash B}{\Delta \vdash A \multimap B} \multimap R \qquad \frac{\Delta_1 \vdash A \quad \Delta_2, B \vdash C}{\Delta_1, \Delta_2, A \multimap B \vdash C} \multimap L$$

- They must be in local harmony
  - Truth justifies use (cut reduction)
  - Use justifies truth (identity expansion)

Cut Reduction

$$\dfrac{\dfrac{\Delta, A \vdash B}{\Delta \vdash A \multimap B}\ {\multimap}R \qquad \dfrac{\Delta_1 \vdash A \quad \Delta_2, B \vdash C}{\Delta_1, \Delta_2, A \multimap B \vdash C}\ {\multimap}L}{\Delta, \Delta_1, \Delta_2 \vdash C}\ \mathsf{cut}_{A \multimap B}$$

$$\Longrightarrow_R \quad \dfrac{\dfrac{\Delta_1 \vdash A \quad \Delta, A \vdash B}{\Delta, \Delta_1 \vdash B}\ \mathsf{cut}_A \qquad \Delta_2, B \vdash C}{\Delta, \Delta_1, \Delta_2 \vdash C}\ \mathsf{cut}_B$$

Identity Expansion

$$\dfrac{}{A \multimap B \vdash A \multimap B}\ \mathsf{id}_{A \multimap B} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\dfrac{}{A \vdash A}\ \mathsf{id}_A \quad \dfrac{}{B \vdash B}\ \mathsf{id}_B}{A \multimap B, A \vdash B}\ \multimap L}{A \multimap B \vdash A \multimap B}\ \multimap R$$

External Choice

- Second sample linear connective: external choice $A \mathbin{\&} B$

$$\frac{\Delta \vdash A \quad \Delta \vdash B}{\Delta \vdash A \mathbin{\&} B} \mathbin{\&}R$$

$$\frac{\Delta, A \vdash C}{\Delta, A \mathbin{\&} B \vdash C} \mathbin{\&}L_1 \qquad \frac{\Delta, B \vdash C}{\Delta, A \mathbin{\&} B \vdash C} \mathbin{\&}L_2$$

- Locally, satisfies cut reduction and identity expansion
- Globally, satisfies cut and identity elimination

Cut-Free System

- Compositional meaning explanation of $A \multimap B$, $A \mathbin{\&} B$
- Subformula property; independence of connectives
- Basis for simple proof construction algorithm
- Complete by cut and identity elimination

$$\frac{}{a \vdash a}\ \mathrm{id}_a$$

$$\frac{\Delta, A \vdash B}{\Delta \vdash A \multimap B}\ {\multimap}R \qquad \frac{\Delta_1 \vdash A \quad \Delta_2, B \vdash C}{\Delta_1, \Delta_2, A \multimap B \vdash C}\ {\multimap}L$$

$$\frac{\Delta \vdash A \quad \Delta \vdash B}{\Delta \vdash A \mathbin{\&} B}\ \&R$$

$$\frac{\Delta, A \vdash C}{\Delta, A \mathbin{\&} B \vdash C}\ \&L_1 \qquad \frac{\Delta, B \vdash C}{\Delta, A \mathbin{\&} B \vdash C}\ \&L_2$$

Backward Proof Construction

$$\frac{}{a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c}\ ?$$

Backward Proof Construction

$$\dfrac{\dfrac{}{a, a \multimap (b \mathbin{\&} c), b \multimap c \vdash c}\ ?}{a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c}\ \multimap R$$

Backward Proof Construction

$$\dfrac{\dfrac{}{a \vdash a}\,? \qquad \dfrac{}{b \mathbin{\&} c, b \multimap c \vdash c}\,?}{\dfrac{a, a \multimap (b \mathbin{\&} c), b \multimap c \vdash c}{a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c}\,{\multimap}R}\,{\multimap}L$$

Backward Proof Construction

$$\dfrac{\dfrac{}{a \vdash a}\ \mathrm{id}_a \qquad \dfrac{}{b \mathbin{\&} c, b \multimap c \vdash c}\ ?}{\dfrac{a, a \multimap (b \mathbin{\&} c), b \multimap c \vdash c}{a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c}\ {\multimap}R}\ {\multimap}L$$

Backward Proof Construction

$$\dfrac{\dfrac{\dfrac{}{a \vdash a}\,\mathrm{id}_a \qquad \dfrac{\dfrac{}{b, b \multimap c \vdash c}\,?}{b \mathbin{\&} c, b \multimap c \vdash c}\,\&L_1}{a, a \multimap (b \mathbin{\&} c), b \multimap c \vdash c}\,{\multimap}L}{a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c}\,{\multimap}R$$

Backward Proof Construction

$$
\dfrac{\dfrac{}{a \vdash a}\,\mathrm{id}_a \qquad \dfrac{\dfrac{\dfrac{}{b \vdash b}\,? \qquad \dfrac{}{c \vdash c}\,?}{b, b \multimap c \vdash c}\,{\multimap}L}{b \mathbin{\&} c, b \multimap c \vdash c}\,{\&}L_1}{\dfrac{a, a \multimap (b \mathbin{\&} c), b \multimap c \vdash c}{a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c}\,{\multimap}R}\,{\multimap}L
$$

Backward Proof Construction

$$
\dfrac{
  \dfrac{}{a \vdash a}\;\mathrm{id}_a
  \qquad
  \dfrac{
    \dfrac{
      \dfrac{}{b \vdash b}\;\mathrm{id}_b
      \qquad
      \dfrac{}{c \vdash c}\;?
    }{b, b \multimap c \vdash c}\;\multimap L
  }{b \mathbin{\&} c, b \multimap c \vdash c}\;\mathbin{\&} L_1
}{
  \dfrac{a, a \multimap (b \mathbin{\&} c), b \multimap c \vdash c}{a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c}\;\multimap R
}\;\multimap L
$$

Backward Proof Construction

$$
\dfrac{\dfrac{}{a \vdash a}\,\mathrm{id}_a \qquad \dfrac{\dfrac{\dfrac{}{b \vdash b}\,\mathrm{id}_b \quad \dfrac{}{c \vdash c}\,\mathrm{id}_c}{b, b \multimap c \vdash c}\,{\multimap}L}{b \,\&\, c, b \multimap c \vdash c}\,\&L_1}{\dfrac{a, a \multimap (b \,\&\, c), b \multimap c \vdash c}{a \multimap (b \,\&\, c), b \multimap c \vdash a \multimap c}\,{\multimap}R}\,{\multimap}L
$$

Forward Proof Construction

- Inverse method [Gentzen'35] [Maslov'64] [Mints'81]
- Step 1: specialize rules to subformulas of end sequent

$$a \multimap (b \mathbin{\&} c), b \multimap c \vdash a \multimap c$$

$$\frac{}{a \vdash a}\ \mathrm{id}_a \qquad \frac{}{b \vdash b}\ \mathrm{id}_b \qquad \frac{}{c \vdash c}\ \mathrm{id}_c$$

$$\frac{\Delta, b \vdash \gamma}{\Delta, b \mathbin{\&} c \vdash \gamma}\ \& L_1 \qquad \frac{\Delta, c \vdash \gamma}{\Delta, b \mathbin{\&} c \vdash \gamma}\ \& L_2$$

$$\frac{\Delta_1 \vdash a \quad \Delta_2, b \mathbin{\&} c \vdash \gamma}{\Delta_1, \Delta_2, a \multimap b \mathbin{\&} c \vdash \gamma}\ {\multimap}L$$

$$\frac{\Delta_1 \vdash b \quad \Delta_2, c \vdash \gamma}{\Delta_1, \Delta_2, b \multimap c \vdash \gamma}\ {\multimap}L \qquad \frac{\Delta, a \vdash c}{\Delta \vdash a \multimap c}\ {\multimap}R$$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \,\&\, c$, $a \multimap (b \,\&\, c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

$$
\begin{array}{ccccc}
a & \vdash & a & & 1 = \mathrm{id}_a \\
b & \vdash & b & & 2 = \mathrm{id}_b \\
c & \vdash & c & & 3 = \mathrm{id}_c \\
\hline
\end{array}
$$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \mathbin{\&} c$, $a \multimap (b \mathbin{\&} c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

$$
\begin{array}{rcll}
a & \vdash & a & 1 = \mathrm{id}_a \\
b & \vdash & b & 2 = \mathrm{id}_b \\
c & \vdash & c & 3 = \mathrm{id}_c \\
\hline
b \mathbin{\&} c & \vdash & b & 4 = \mathbin{\&}L_1(2)
\end{array}
$$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \mathbin{\&} c$, $a \multimap (b \mathbin{\&} c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

$$
\begin{array}{rcll}
a & \vdash & a & 1 = \mathrm{id}_a \\
b & \vdash & b & 2 = \mathrm{id}_b \\
c & \vdash & c & 3 = \mathrm{id}_c \\
\hline
b \mathbin{\&} c & \vdash & b & 4 = \mathbin{\&}L_1(2) \\
b \mathbin{\&} c & \vdash & c & 5 = \mathbin{\&}L_2(3)
\end{array}
$$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \mathbin{\&} c$, $a \multimap (b \mathbin{\&} c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

a	$\vdash$	a	$1 = \mathrm{id}_a$
b	$\vdash$	b	$2 = \mathrm{id}_b$
c	$\vdash$	c	$3 = \mathrm{id}_c$
$b \mathbin{\&} c$	$\vdash$	b	$4 = \mathbin{\&}L_1(2)$
$b \mathbin{\&} c$	$\vdash$	c	$5 = \mathbin{\&}L_2(3)$
$b, b \multimap c$	$\vdash$	c	$6 = \multimap L(2, 3)$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \mathbin{\&} c$, $a \multimap (b \mathbin{\&} c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

$$
\begin{array}{rcll}
a & \vdash & a & 1 = \mathrm{id}_a \\
b & \vdash & b & 2 = \mathrm{id}_b \\
c & \vdash & c & 3 = \mathrm{id}_c \\
\hline
b \mathbin{\&} c & \vdash & b & 4 = \mathbin{\&}L_1(2) \\
b \mathbin{\&} c & \vdash & c & 5 = \mathbin{\&}L_2(3) \\
b, b \multimap c & \vdash & c & 6 = \multimap L(2,3) \\
\hline
a, a \multimap (b \mathbin{\&} c) & \vdash & b & 7 = \multimap L(1,4)
\end{array}
$$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \mathbin{\&} c$, $a \multimap (b \mathbin{\&} c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

a	$\vdash$	a	$1 = \mathrm{id}_a$
b	$\vdash$	b	$2 = \mathrm{id}_b$
c	$\vdash$	c	$3 = \mathrm{id}_c$
$b \mathbin{\&} c$	$\vdash$	b	$4 = \& L_1(2)$
$b \mathbin{\&} c$	$\vdash$	c	$5 = \& L_2(3)$
$b, b \multimap c$	$\vdash$	c	$6 = \multimap L(2, 3)$
$a, a \multimap (b \mathbin{\&} c)$	$\vdash$	b	$7 = \multimap L(1, 4)$
$a, a \multimap (b \mathbin{\&} c)$	$\vdash$	c	$8 = \multimap L(1, 5)$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \,\&\, c$, $a \multimap (b \,\&\, c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

$$
\begin{array}{rcll}
a & \vdash & a & 1 = \mathrm{id}_a \\
b & \vdash & b & 2 = \mathrm{id}_b \\
c & \vdash & c & 3 = \mathrm{id}_c \\
\hline
b \,\&\, c & \vdash & b & 4 = \& L_1(2) \\
b \,\&\, c & \vdash & c & 5 = \& L_2(3) \\
b, b \multimap c & \vdash & c & 6 = \multimap L(2,3) \\
\hline
a, a \multimap (b \,\&\, c) & \vdash & b & 7 = \multimap L(1,4) \\
a, a \multimap (b \,\&\, c) & \vdash & c & 8 = \multimap L(1,5) \\
\hline
a \multimap (b \,\&\, c) & \vdash & a \multimap c & 9 = \multimap R(7)
\end{array}
$$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \mathbin{\&} c$, $a \multimap (b \mathbin{\&} c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

a	$\vdash$	a	$1 = \mathrm{id}_a$
b	$\vdash$	b	$2 = \mathrm{id}_b$
c	$\vdash$	c	$3 = \mathrm{id}_c$
$b \mathbin{\&} c$	$\vdash$	b	$4 = \mathbin{\&}L_1(2)$
$b \mathbin{\&} c$	$\vdash$	c	$5 = \mathbin{\&}L_2(3)$
$b, b \multimap c$	$\vdash$	c	$6 = \multimap L(2,3)$
$a, a \multimap (b \mathbin{\&} c)$	$\vdash$	b	$7 = \multimap L(1,4)$
$a, a \multimap (b \mathbin{\&} c)$	$\vdash$	c	$8 = \multimap L(1,5)$
$a \multimap (b \mathbin{\&} c)$	$\vdash$	$a \multimap c$	$9 = \multimap R(7)$
$a, a \multimap (b \mathbin{\&} c), b \multimap c$	$\vdash$	c	$10 = \multimap L(7,3)$

Forward Deduction

- Step 2: Apply just these rule instances, arbitrarily
- Subformulas, left: $b \mathbin{\&} c$, $a \multimap (b \mathbin{\&} c)$, $b \multimap c$
- Subformulas, right: $a \multimap c$

$$
\begin{array}{rcll}
a & \vdash & a & 1 = \mathrm{id}_a \\
b & \vdash & b & 2 = \mathrm{id}_b \\
c & \vdash & c & 3 = \mathrm{id}_c \\
\hline
b \mathbin{\&} c & \vdash & b & 4 = \mathbin{\&}L_1(2) \\
b \mathbin{\&} c & \vdash & c & 5 = \mathbin{\&}L_2(3) \\
b, b \multimap c & \vdash & c & 6 = \multimap L(2,3) \\
\hline
a, a \multimap (b \mathbin{\&} c) & \vdash & b & 7 = \multimap L(1,4) \\
a, a \multimap (b \mathbin{\&} c) & \vdash & c & 8 = \multimap L(1,5) \\
\hline
a \multimap (b \mathbin{\&} c) & \vdash & a \multimap c & 9 = \multimap R(7) \\
a, a \multimap (b \mathbin{\&} c), b \multimap c & \vdash & c & 10 = \multimap L(7,3) \\
\hline
a \multimap (b \mathbin{\&} c), b \multimap c & \vdash & a \multimap c & 11 = \multimap R(10)
\end{array}
$$

Summary: Past

- Sequent calculus [Gentzen'35]
- Define connectives by right rules (how to prove) and left rules (how to use)
- Should satisfy harmony [Dummett'76]
  - Global harmony: cut and identity elimination
  - Local harmony: cut reduction and identity expansion
- Per Martin-Löf [1983]

  > *The meaning of a proposition is determined by what it is to verify it, or what counts as a verification of it.*

- Cut-free sequent calculus as basis for proof search
  - Backwards, with backtracking proof search
  - Forwards, based on specialized inference rules

Outline

- Past: How to define a logic
  - Sequent calculus [Gentzen'35]
  - Harmony [Dummett'76] [Martin-Löf'83]
- Present: How to reduce nondeterminism in search
  - Focusing and polarization [Andreoli'92] [Laurent'99]
- Future: How to combine logics
  - Adjunctions [Benton'94] [Reed'09]

Nondeterminism

- Much nondeterminism remains
- Backward search (don't-know)
  - At each step: which rule do we try?
  - Backtrack upon failure
- Forward search (don't-care)
  - At each step: which (specialized) rule do we apply?
  - Generate useless and redundant sequents

Observations: Inversion and Focusing

- We can always decompose some connectives without losing provability (inversion)

$$\frac{\Delta, A \vdash B}{\Delta \vdash A \multimap B} \multimap R \qquad \left[\ \dfrac{\Delta \vdash A \multimap B \qquad \dfrac{\dfrac{}{A \vdash A}\,\mathrm{id}_A \quad \dfrac{}{B \vdash B}\,\mathrm{id}_B}{A, A \multimap B \vdash B} \multimap L}{\Delta, A \vdash B}\,\mathrm{cut}_A\ \right]$$

- Other connectives require a choice, but we can combine successive choices on the same formula (focusing)

$$\frac{\Delta, A \vdash D}{\Delta, A \,\&\, (B \,\&\, C) \vdash D} \,\&L_1$$

$$\frac{\Delta, B \vdash D}{\Delta, A \,\&\, (B \,\&\, C) \vdash D} \,\&L_{21} \qquad \frac{\Delta, C \vdash D}{\Delta, A \,\&\, (B \,\&\, C) \vdash D} \,\&L_{22}$$

Negative and Positive Connectives

- Negative connectives have invertible right rule
  - Involve some choice on the left
- Positive connectives have invertible left rule
  - Involve some choice on the right
- Segregate in syntax to exploit inversion and focusing

$$\begin{array}{llll} \text{Negative} & A^- & ::= & A_1^+ \multimap A_2^- \mid A_1^- \mathbin{\&} A_2^- \mid \top \mid a^- \mid {\uparrow}A^+ \\ \text{Positive} & A^+ & ::= & A_1^+ \otimes A_2^+ \mid \mathbf{1} \mid A_1^+ \oplus A_2^+ \mid \mathbf{0} \mid a^+ \mid {\downarrow}A^- \end{array}$$

- Glimpse at other linear connectives $(\top, \otimes, \mathbf{1}, \oplus, \mathbf{0})$
- Shifts ${\uparrow}A^+$ and ${\downarrow}A^-$ ensure every formula can be polarized
  - Minimal polarization exists
  - Assign atoms arbitrary consistent polarity

Re-engineering Deduction, Inversion Phase

- $\Delta \Vdash A$ for polarized focused deduction
- Inversion phase: break down negatives on right and positives on left (confluent)

$$\frac{\Delta, A^+ \Vdash B^-}{\Delta \Vdash A^+ \multimap B^-} \multimap R \qquad \frac{\Delta \Vdash A^- \quad \Delta \Vdash B^-}{\Delta \Vdash A^- \mathbin{\&} B^-} \mathbin{\&} R$$

$$\frac{\Delta \Vdash A^+}{\Delta \Vdash {\uparrow} A^+} {\uparrow} R \qquad \frac{\Delta, A^- \Vdash C}{\Delta, {\downarrow} A^- \Vdash C} {\downarrow} L$$

- Suspend atoms during inversion

$$\frac{\Delta \Vdash \langle a^- \rangle}{\Delta \Vdash a^-} \langle\,\rangle^- \qquad \frac{\Delta, \langle a^+ \rangle \Vdash C}{\Delta, a^+ \Vdash C} \langle\,\rangle^+$$

- Stable antecedents $\Delta^- ::= \bullet \mid \Delta^-, A^- \mid \Delta^-, \langle a^+ \rangle$
- Stable succedents $\gamma^+ ::= A^+ \mid \langle a^- \rangle$

Re-engineering Deduction, Inversion Phase

- $\Delta \Vdash A$ for polarized focused deduction
- Inversion phase: break down negatives on right and positives on left (confluent)

$$\frac{\Delta, A^+ \Vdash B^-}{\Delta \Vdash A^+ \multimap B^-} \multimap R \qquad \frac{\Delta \Vdash A^- \quad \Delta \Vdash B^-}{\Delta \Vdash A^- \mathbin{\&} B^-} \mathbin{\&} R$$

$$\frac{\Delta \Vdash A^+}{\Delta \Vdash \uparrow A^+} \uparrow R \qquad \frac{\Delta, A^- \Vdash \gamma}{\Delta, \downarrow A^- \Vdash \gamma} \downarrow L$$

- Suspend atoms during inversion

$$\frac{\Delta \Vdash \langle a^- \rangle}{\Delta \Vdash a^-} \langle\,\rangle^- \qquad \frac{\Delta, \langle a^+ \rangle \Vdash \gamma}{\Delta, a^+ \Vdash \gamma} \langle\,\rangle^+$$

- Stable antecedents $\Delta^- ::= \bullet \mid \Delta^-, A^- \mid \Delta^-, \langle a^+ \rangle$
- Stable succedents $\gamma^+ ::= A^+ \mid \langle a^- \rangle$

Re-engineering Deduction, Transition

- Focus on positive on right or negative on left
- Only one formula may be in focus in a sequent

$$\frac{\Delta^- \Vdash [A^+]}{\Delta^- \Vdash A^+}\,[\,]^+ \qquad \frac{\Delta^-, [A^-] \Vdash \gamma^+}{\Delta^-, A^- \Vdash \gamma^+}\,[\,]^-$$

Re-engineering Deduction, Focusing Phase

- Combine successive choices in focus

$$\frac{\Delta_1^- \Vdash [A^+] \quad \Delta_2^-, [B^-] \Vdash \gamma^+}{\Delta_1^-, \Delta_2^-, [A^+ \multimap B^-] \Vdash \gamma^+} \multimap L$$

$$\frac{\Delta^-, [A^-] \Vdash \gamma^+}{\Delta^-, [A^- \,\&\, B^-] \Vdash \gamma^+} \,\&L_1 \qquad \frac{\Delta^-, [B^-] \Vdash \gamma^+}{\Delta^-, [A^- \,\&\, B^-] \Vdash \gamma^+} \,\&L_2$$

$$\frac{\Delta^-, A^+ \Vdash \gamma^+}{\Delta^-, [\uparrow A^+] \Vdash \gamma^+} \uparrow L \qquad \frac{\Delta^- \Vdash A^-}{\Delta^- \Vdash [\downarrow A^-]} \downarrow R$$

$$\frac{}{[a^-] \Vdash \langle a^- \rangle} \mathrm{id}_a^- \qquad \frac{}{\langle a^+ \rangle \Vdash [a^+]} \mathrm{id}_a^+$$

Soundness and Completeness

- Judgments now A true, $[A]$ in focus, $\langle A \rangle$ suspended
- Collectively, the system is called focusing
- Focusing is sound
  - Restricts inferences
  - Depolarize and induct over derivation
- Focusing is complete. Key properties [Simmons'13]
  - Cuts on A are admissible by nested induction, first on A
  - Identities on A are admissible by induction on A
- Use as basis to improve both backward and forward search [Andreoli'01] [Chaudhuri et al.'06] [McLaughlin & Pf'09]

Example Revisited

- Choose all atoms to be negative (a^-, b^-, c^-)

$$\downarrow a \multimap (b \mathbin{\&} c), \downarrow b \multimap c \Vdash \downarrow a \multimap c$$

- Step 1: Invert to stable sequents

$$a, \downarrow a \multimap (b \mathbin{\&} c), \downarrow b \multimap c \Vdash \langle c \rangle$$

- Step 2: Construct derived rules between stable sequents $\Delta^- \Vdash \gamma^+$

Constructing Derived Rules

- Correspondence between formulas and rules of inference [Andreoli'01]
- Example: Antecedent $\downarrow a \multimap (b \mathbin{\&} c)$

$$\frac{}{\Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \gamma^+} \; []^-$$

Constructing Derived Rules

- Correspondence between formulas and rules of inference [Andreoli'01]
- Example: Antecedent $\downarrow a \multimap (b \,\&\, c)$

$$\frac{\overline{\Delta^-, [\downarrow a \multimap (b \,\&\, c)] \Vdash \gamma^+}}{\Delta^-, \downarrow a \multimap (b \,\&\, c) \Vdash \gamma^+} \; []^-$$

Constructing Derived Rules

- Correspondence between formulas and rules of inference [Andreoli'01]
- Example: Antecedent $\downarrow a \multimap (b \mathbin{\&} c)$

$$\dfrac{\dfrac{\Delta^- = (\Delta_1^-, \Delta_2^-) \quad \dfrac{}{\Delta_1^- \Vdash [\downarrow a]} \quad \dfrac{}{\Delta_2^-, [b \mathbin{\&} c] \Vdash \gamma^+}}{\Delta^-, [\downarrow a \multimap (b \mathbin{\&} c)] \Vdash \gamma^+}\ {\multimap}L}{\Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \gamma^+}\ [\,]^-$$

Constructing Derived Rules

- Correspondence between formulas and rules of inference [Andreoli'01]
- Example: Antecedent $\downarrow a \multimap (b \mathbin{\&} c)$

$$
\dfrac{
\dfrac{
\Delta^- = (\Delta_1^-, \Delta_2^-)
\quad
\dfrac{\dfrac{\Delta_1^- \Vdash \langle a \rangle}{\Delta_1^- \Vdash a}\ \langle\,\rangle^-}{\Delta_1^- \Vdash [\downarrow a]}\ \downarrow R
\quad
\dfrac{}{\Delta_2^-, [b \mathbin{\&} c] \Vdash \gamma^+}
}{
\Delta^-, [\downarrow a \multimap (b \mathbin{\&} c)] \Vdash \gamma^+
}\ \multimap L
}{
\Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \gamma^+
}\ [\,]^-
$$

Constructing Derived Rules

- Correspondence between formulas and rules of inference [Andreoli'01]
- Example: Antecedent $\downarrow a \multimap (b \,\&\, c)$

$$
\dfrac{
\dfrac{
\Delta^- = (\Delta_1^-, \Delta_2^-)
\qquad
\dfrac{\dfrac{\Delta_1^- \Vdash \langle a \rangle}{\Delta_1^- \Vdash a}\,\langle\,\rangle^-}{\Delta_1^- \Vdash [\downarrow a]}\,{\downarrow}R
\qquad
\dfrac{\dfrac{}{\Delta_2^-, [b] \Vdash \gamma^+}}{\Delta_2^-, [b \,\&\, c] \Vdash \gamma^+}\,\&L_1
}{
\Delta^-, [\downarrow a \multimap (b \,\&\, c)] \Vdash \gamma^+
}\,{\multimap}L
}{
\Delta^-, \downarrow a \multimap (b \,\&\, c) \Vdash \gamma^+
}\,[\,]^-
$$

Constructing Derived Rules

- Correspondence between formulas and rules of inference [Andreoli'01]
- Example: Antecedent $\downarrow a \multimap (b \mathbin{\&} c)$

$$
\dfrac{
  \dfrac{
    \Delta^- = (\Delta_1^-, \Delta_2^-)
    \quad
    \dfrac{\dfrac{\Delta_1^- \Vdash \langle a \rangle}{\Delta_1^- \Vdash a}\ \langle\,\rangle^-}{\Delta_1^- \Vdash [\downarrow a]}\ \downarrow R
    \quad
    \dfrac{\dfrac{\Delta_2^- = \bullet \quad \gamma^+ = \langle b \rangle}{\Delta_2^-, [b] \Vdash \gamma^+}\ \mathrm{id}_b^-}{\Delta_2^-, [b \mathbin{\&} c] \Vdash \gamma^+}\ \mathbin{\&} L_1
  }{
    \Delta^-, [\downarrow a \multimap (b \mathbin{\&} c)] \Vdash \gamma^+
  }\ \multimap L
}{
  \Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \gamma^+
}\ [\,]^-
$$

Constructing Derived Rules

- Correspondence between formulas and rules of inference [Andreoli'01]
- Example: Antecedent $\downarrow a \multimap (b \mathbin{\&} c)$

$$\dfrac{\Delta^- = (\Delta_1^-, \Delta_2^-) \quad \dfrac{\dfrac{\Delta_1^- \Vdash \langle a \rangle}{\Delta_1^- \Vdash a}\ \langle\,\rangle^-}{\Delta_1^- \Vdash [\downarrow a]}\ \downarrow R \quad \dfrac{\dfrac{\Delta_2^- = \bullet \quad \gamma^+ = \langle b \rangle}{\Delta_2^-, [b] \Vdash \gamma^+}\ \mathrm{id}_b^-}{\Delta_2^-, [b \mathbin{\&} c] \Vdash \gamma^+}\ \mathbin{\&} L_1}{\dfrac{\Delta^-, [\downarrow a \multimap (b \mathbin{\&} c)] \Vdash \gamma^+}{\Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \gamma^+}\ [\,]^-}\ \multimap L$$

- Yields derived rule

$$\dfrac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3]$$

Backward Search

- Derived rules

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, \downarrow b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, \downarrow a \multimap (b \,\&\, c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, \downarrow a \multimap (b \,\&\, c) \Vdash \langle c \rangle}\ [4]$$

- Step 3a: Backward search using only derived rules

Backward Search

- Derived rules

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \,\&\, c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \,\&\, c) \Vdash \langle c \rangle}\ [4]$$

- Step 3a: Backward search using only derived rules

$$\frac{}{a, {\downarrow} a \multimap (b \,\&\, c), {\downarrow} b \multimap c \Vdash \langle c \rangle}\ ?$$

Backward Search

- Derived rules

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \,\&\, c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \,\&\, c) \Vdash \langle c \rangle}\ [4]$$

- Step 3a: Backward search using only derived rules

$$\frac{a, {\downarrow} b \multimap c \Vdash \langle a \rangle}{a, {\downarrow} a \multimap (b \,\&\, c), {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [4] \qquad \frac{}{a, {\downarrow} a \multimap (b \,\&\, c), {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

Backward Search

- Derived rules

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, \downarrow b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, \downarrow a \multimap (b \,\&\, c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, \downarrow a \multimap (b \,\&\, c) \Vdash \langle c \rangle}\ [4]$$

- Step 3a: Backward search using only derived rules

failure: no rule applies

$$\frac{a, \downarrow b \multimap c \Vdash \langle a \rangle}{a, \downarrow a \multimap (b \,\&\, c), \downarrow b \multimap c \Vdash \langle c \rangle}\ [4] \qquad \frac{}{a, \downarrow a \multimap (b \,\&\, c), \downarrow b \multimap c \Vdash \langle c \rangle}\ [2]$$

Backward Search

- Derived rules

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, \downarrow b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\ [4]$$

- Step 3a: Backward search using only derived rules

failure: no rule applies

$$\frac{a, \downarrow b \multimap c \Vdash \langle a \rangle}{a, \downarrow a \multimap (b \mathbin{\&} c), \downarrow b \multimap c \Vdash \langle c \rangle}\ [4] \qquad \frac{\dfrac{}{a, \downarrow a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ ?}{a, \downarrow a \multimap (b \mathbin{\&} c), \downarrow b \multimap c \Vdash \langle c \rangle}\ [2]$$

Backward Search

- Derived rules

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\ [4]$$

- Step 3a: Backward search using only derived rules

$$\frac{\begin{array}{c}\text{failure: no rule applies}\\ a, {\downarrow} b \multimap c \Vdash \langle a \rangle\end{array}}{a, {\downarrow} a \multimap (b \mathbin{\&} c), {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [4] \qquad \frac{\dfrac{\dfrac{}{a \Vdash \langle a \rangle}\ ?}{a, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3]}{a, {\downarrow} a \multimap (b \mathbin{\&} c), {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

Backward Search

- Derived rules

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow}b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow}a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow}a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\ [4]$$

- Step 3a: Backward search using only derived rules
  - Only two possible attempts!

failure: no rule applies

$$\frac{a, {\downarrow}b \multimap c \Vdash \langle a \rangle}{a, {\downarrow}a \multimap (b \mathbin{\&} c), {\downarrow}b \multimap c \Vdash \langle c \rangle}\ [4] \qquad \frac{\dfrac{\overline{a \Vdash \langle a \rangle}\ [1]}{a, {\downarrow}a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3]}{a, {\downarrow}a \multimap (b \mathbin{\&} c), {\downarrow}b \multimap c \Vdash \langle c \rangle}\ [2]$$

Forward Search

- Recall derived rules: use only these!

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\ [4]$$

- Step 3b: Focused inverse method [McLaughlin & Pf'09]

Forward Search

- Recall derived rules: use only these!

$$\frac{}{a \Vdash \langle a \rangle}\,[1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow}b \multimap c \Vdash \langle c \rangle}\,[2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow}a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\,[3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow}a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\,[4]$$

- Step 3b: Focused inverse method [McLaughlin & Pf'09]

$$\underline{\quad a \quad \Vdash \quad \langle a \rangle \quad 1 = [1] \quad}$$

Forward Search

- Recall derived rules: use only these!

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\ [4]$$

- Step 3b: Focused inverse method [McLaughlin & Pf'09]

$$\begin{array}{rccl} a & \Vdash & \langle a \rangle & 1 = [1] \\ \hline a, {\downarrow} a \multimap (b \mathbin{\&} c) & \Vdash & \langle b \rangle & 2 = [3](1) \end{array}$$

Forward Search

- Recall derived rules: use only these!

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\ [4]$$

- Step 3b: Focused inverse method [McLaughlin & Pf'09]

$$\begin{array}{rcll} a & \Vdash & \langle a \rangle & 1 = [1] \\ \hline a, {\downarrow} a \multimap (b \mathbin{\&} c) & \Vdash & \langle b \rangle & 2 = [3](1) \\ a, {\downarrow} a \multimap (b \mathbin{\&} c) & \Vdash & \langle c \rangle & 3 = [4](1) \\ \hline \end{array}$$

Forward Search

- Recall derived rules: use only these!

$$\frac{}{a \Vdash \langle a \rangle}\ [1] \qquad \frac{\Delta^- \Vdash \langle b \rangle}{\Delta^-, {\downarrow} b \multimap c \Vdash \langle c \rangle}\ [2]$$

$$\frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle b \rangle}\ [3] \qquad \frac{\Delta^- \Vdash \langle a \rangle}{\Delta^-, {\downarrow} a \multimap (b \mathbin{\&} c) \Vdash \langle c \rangle}\ [4]$$

- Step 3b: Focused inverse method [McLaughlin & Pf'09]
  - Only one unused sequent!

$$\begin{array}{rcll}
a & \Vdash & \langle a \rangle & 1 = [1] \\
\hline
a, {\downarrow} a \multimap (b \mathbin{\&} c) & \Vdash & \langle b \rangle & 2 = [3](1) \\
a, {\downarrow} a \multimap (b \mathbin{\&} c) & \Vdash & \langle c \rangle & 3 = [4](1) \\
\hline
a, {\downarrow} a \multimap (b \mathbin{\&} c), {\downarrow} b \multimap c & \Vdash & \langle c \rangle & 4 = [2](2)
\end{array}$$

Summary: Present

- Polarize the logic into negative and positive propositions
  - Negatives are invertible on the right
  - Positives are invertible on the left
- Focused deduction
  - Decompose all invertible connectives
  - Focus on one noninvertible one
  - Continue to focus until invertibles are uncovered
- Sound and complete, key is cut elimination for polarized focused logic [Simmons'13]
- Use for big-step inferences (backwards and forwards)
- Drastically reduces search space
- So far, focusing applies for many interesting logics (linear, intuitionistic, classical) [Liang & Miller'09]

Outline

- Past: How to define a logic
  - Sequent calculus [Gentzen'35]
  - Harmony [Dummett'76] [Martin-Löf'83]
- Present: How to reduce nondeterminism in search
  - Focusing and polarization [Andreoli'92] [Laurent'99]
- Future: How to combine logics
  - Adjunctions [Benton'94] [Reed'09]

Example: Recovering Intuitionistic Logic

- $A \to B \simeq\, !A \multimap B$ [Girard'87]
  - $!A$ internalizes categorical judgment $\bullet \vdash A$
  - $!A$ satisfies weakening and contraction
- Alternative: combine intuitionistic and linear logic via an adjunction [Benton'94]
  - Two functors F and G, F left adjoint to G
  - Syntax as modal operators $G\,A$ and $F\,X$
  - Decompose $!A \simeq F\,(G\,A)$
- Generalized to multi-modal logics [Reed'09]

Adjoint Logic

- Two-level system [Benton'94]

$$\begin{array}{llll} \text{Unrestricted} & A_{\mathsf{U}} & ::= & A_{\mathsf{U}} \to A_{\mathsf{U}} \mid A_{\mathsf{U}} \wedge B_{\mathsf{U}} \mid a_{\mathsf{U}} \mid G\, A_{\mathsf{L}} \\ \text{Linear} & A_{\mathsf{L}} & ::= & A_{\mathsf{L}} \multimap A_{\mathsf{L}} \mid A_{\mathsf{L}} \,\&\, B_{\mathsf{L}} \mid a_{\mathsf{L}} \mid F\, A_{\mathsf{U}} \end{array}$$

- Represent $!A_{\mathsf{L}} \simeq F\; G\; A_{\mathsf{L}}$

Adjoint Logic

- Two-level system [Benton'94]

Unrestricted	A_{U}	$::=$	$A_{\mathsf{U}} \rightarrow A_{\mathsf{U}} \mid A_{\mathsf{U}} \wedge B_{\mathsf{U}} \mid a_{\mathsf{U}} \mid G\, A_{\mathsf{L}}$
Linear	A_{L}	$::=$	$A_{\mathsf{L}} \multimap A_{\mathsf{L}} \mid A_{\mathsf{L}} \,\&\, B_{\mathsf{L}} \mid a_{\mathsf{L}} \mid F\, A_{\mathsf{U}}$

- Represent $!A_{\mathsf{L}} \simeq F\, G\, A_{\mathsf{L}}$
- Observation: $\uparrow A^{+}$ and $\downarrow A^{-}$ of polarized linear logic also combine two separate language levels!

Adjoint Logic

- Two-level system [Benton'94]

$$\begin{array}{llcl} \text{Unrestricted} & A_{\mathsf{U}} & ::= & A_{\mathsf{U}} \to A_{\mathsf{U}} \mid A_{\mathsf{U}} \wedge B_{\mathsf{U}} \mid a_{\mathsf{U}} \mid G\, A_{\mathsf{L}} \\ \text{Linear} & A_{\mathsf{L}} & ::= & A_{\mathsf{L}} \multimap A_{\mathsf{L}} \mid A_{\mathsf{L}} \,\&\, B_{\mathsf{L}} \mid a_{\mathsf{L}} \mid F\, A_{\mathsf{U}} \end{array}$$

- Represent $!A_{\mathsf{L}} \simeq F\, G\, A_{\mathsf{L}}$
- Observation: $\uparrow A^{+}$ and $\downarrow A^{-}$ of polarized linear logic also combine two separate language levels!
- Observation: They follow the same rule structure!

Polarized Adjoint Logic

- Unify the two concepts [Pf & Griffith'15]
- Every proposition has a polarity $(^+, ^-)$ and mode (U, L)

$$
\begin{array}{llll}
\text{Modes} & m, k & ::= & \mathsf{U} \mid \mathsf{L} \quad \text{where } U \geq L \\
\text{Neg.} & A_m^- & ::= & A_m^+ \multimap B_m^- \mid A_m^- \mathbin{\&} B_m^- \mid a_m^- \mid {\uparrow}_k^m A_k^+ \quad (m \geq k) \\
\text{Pos.} & A_k^+ & ::= & A_k^+ \otimes B_k^+ \mid A_k^+ \oplus B_k^+ \mid a_k^+ \mid {\downarrow}_k^m A_m^- \quad (m \geq k)
\end{array}
$$

- Define $F\, A_{\mathsf{U}} = {\downarrow}_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{U}}$, $G\, A_{\mathsf{L}} = {\uparrow}_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{L}}$
  - So $!A \simeq F\,(G\, A) \simeq {\downarrow}_{\mathsf{L}}^{\mathsf{U}} {\uparrow}_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{L}}$
- Define $A^+ \rightarrow B^+ \simeq A_{\mathsf{U}}^+ \multimap B_{\mathsf{U}}^-$
- Define $A^+ \wedge B^+ \simeq A_{\mathsf{U}}^+ \otimes B_{\mathsf{U}}^+$
- Define $A^- \wedge B^- \simeq A_{\mathsf{U}}^- \mathbin{\&} B_{\mathsf{U}}^-$
- Earlier modalities ${\uparrow} A = {\uparrow}_{\mathsf{L}}^{\mathsf{L}} A_L$, ${\downarrow} A = {\downarrow}_{\mathsf{L}}^{\mathsf{L}} A_L$

Polarized Adjoint Sequent Calculus

- Mixed antecedents $\Psi ::= \bullet \mid \Psi, A_m$
- Mixed-level judgment $\Psi \vdash A_k$
- Independence and inclusion
  - $\Psi \geq k$ means $m \geq k$ for every A_m in Ψ
  - $\Psi \vdash A_k$ presupposes $\Psi \geq k$

Polarized Adjoint Logic, Inversion Phase

$$\frac{\Psi, A_m^+ \Vdash B_m^-}{\Psi \Vdash A_m^+ \multimap B_m^-} \multimap R \qquad \frac{\Psi \Vdash A_m^- \quad \Psi \Vdash B_m^-}{\Psi \Vdash A_m^- \,\&\, B_m^-} \,\&R$$

$$\frac{\Psi \Vdash A_k^+}{\Psi \Vdash {\uparrow_k^m} A_k^+} \uparrow R \qquad \frac{\Psi, A_m^- \Vdash \gamma_r}{\Psi, {\downarrow_k^m} A_m^- \Vdash \gamma_r} \downarrow L$$

$$\frac{\Psi \Vdash \langle a_m^- \rangle}{\Psi \Vdash a_m^-} \langle\,\rangle^- \qquad \frac{\Psi, \langle a_m^+ \rangle \Vdash \gamma_r}{\Psi, a_m^+ \Vdash \gamma_r} \langle\,\rangle^+$$

Polarized Adjoint Logic, Transition

$$\frac{\Psi^- \Vdash [A_m^+]}{\Psi^- \Vdash A_m^+}\ []^+$$

$$\frac{\Psi^-, [A_{\mathsf{L}}^-] \Vdash \gamma^+}{\Psi^-, A_{\mathsf{L}}^- \Vdash \gamma^+}\ []_{\mathsf{L}}^- \qquad \frac{\Psi^-, A_{\mathsf{U}}^-, [A_{\mathsf{U}}^-] \Vdash \gamma^+}{\Psi^-, A_{\mathsf{U}}^- \Vdash \gamma^+}\ []_{\mathsf{U}}^-$$

Polarized Adjoint Logic, Focusing Phase

- Ψ, Ψ' admits contraction for A_{U} in Ψ and Ψ'

$$\frac{\Psi_1^- \geq m \quad \Psi_1^- \Vdash [A_m^+] \quad \Psi_2^-, [B_m^-] \Vdash \gamma^+}{\Psi_1^-, \Psi_2^-, [A_m^+ \multimap B_m^-] \Vdash \gamma^+} \multimap L$$

$$\frac{\Psi^-, [A_m^-] \Vdash \gamma^+}{\Psi^-, [A_m^- \,\&\, B_m^-] \Vdash \gamma^+} \& L_1 \qquad \frac{\Psi^-, [B_m^-] \Vdash \gamma^+}{\Psi^-, [A_m^- \,\&\, B_m^-] \Vdash \gamma^+} \& L_2$$

$$\frac{k \geq r \quad \Psi^-, A_k^+ \Vdash \gamma_r^+}{\Psi^-, [\uparrow_k^m A_k^+] \Vdash \gamma_r^+} \uparrow L \qquad \frac{\Psi^- \geq m \quad \Psi^- \Vdash A_m^-}{\Psi^- \Vdash [\downarrow_k^m A_m^-]} \downarrow R$$

$$\frac{\Psi \geq \mathsf{U}}{\Psi, [a_m^-] \Vdash \langle a_m^- \rangle} \mathrm{id}_a^- \qquad \frac{\Psi \geq \mathsf{U}}{\Psi, \langle a_m^+ \rangle \Vdash [a_m^+]} \mathrm{id}_a^+$$

Polarized Adjoint Logic

- Different from Andreoli's system
- Polarized (unfocused) adjoint logic satisfies structural cut and identity elimination [Pf & Griffith'15]
- Conjectures:
  - Polarized focused adjoint logic satisfies structural cut and identity elimination
  - Polarized focused adjoint logic is sound and complete
  - Polarized focused adjoint logic is conservative over focused intuitionistic and focused intuitionistic linear logic for proof construction

Further Conjectures

- Adjunction and polarization are generally compatible
- Adjunctions provide a flexible way to combine logics
  - Conservative over both levels
  - Preserves both search spaces under focusing
  - Affine logics are compatible [Pf & Griffith'15]
  - Extends to preorders of logics, under some conditions [Nigam & Miller'09] [Reed'09]
- Combining logics conservatively is important
  - Embeddings lose structure
  - Nonconservative combinations are difficult

Summary: Future

- Study adjunctions as a flexible way to combine logics conservatively
- Examples: intuitionistic, affine, linear, modal logics
- Compatibility with focusing
- Preserving search spaces

Conclusion

- Proof theory is a critical tool in automated deduction
  - Especially in nonclassical logics
  - Which have many applications in computer science
- Complements model-theoretic techniques
- Past: How to define a logic
  - Sequent calculus and harmony
- Present: How to reduce nondeterminism in search
  - Focusing and polarization
- Future: How to combine logics
  - Adjunctions (?)