Adjoint Logic and Its Concurrent Semantics

Frank Pfenning

ABCD Meeting, Edinburgh, December 18-19, 2017

Joint work with Klaas Pruiksma and William Chargin

Outline

- Proofs as programs
- Linear sequent proofs as concurrent programs
  - Take 1: standard sequent calculus = synchronous communication
  - Take 2: unary $\otimes$R and $\multimap$R
  - Take 3: positive axiomatic form = asynchronous communication
- Adjoint logic
  - Structural properties and modes of truth
  - Declaration of Independence
- Concurrent operational semantics
  - Contraction, weakening, multicut, and multi-identity
  - Garbage collection

Proofs as Programs

- Codesign of language and its reasoning principles
- Three levels of correspondence
  - Propositions as types
  - Proofs as programs
  - Proof reduction as computation
- Style of proof system is critical to characterize computation
  - Axiomatic style ⟺ combinators and combinatory reduction [Curry'35]
  - Natural deduction ⟺ λ-calculus and substitution [Howard'69]
  - Sequent calculus ⟺ explicit substitutions [Herbelin'94]
- All intuitionistic and structural (admitting weakening & contraction)

Linear Proofs as Session-Typed Programs

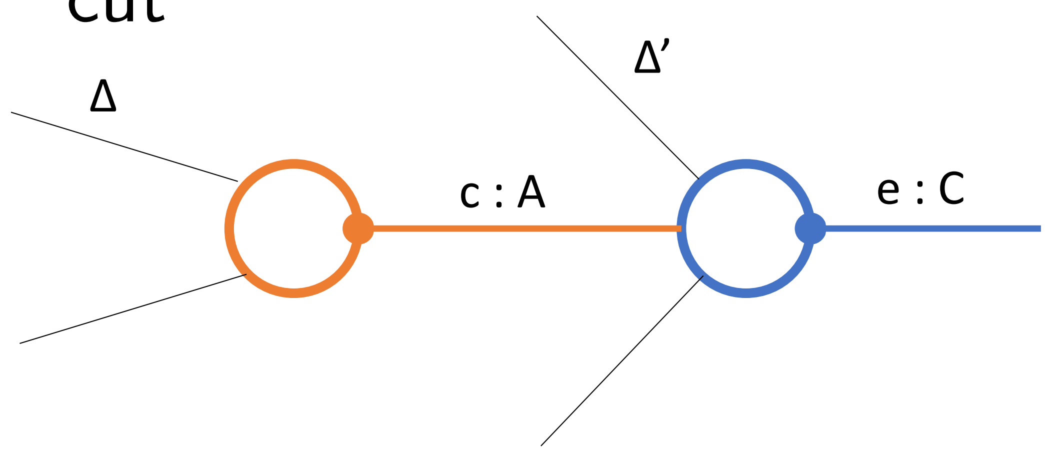
- Three levels of correspondence
  - Linear propositions as session types
  - Sequent proofs as concurrent programs
  - Cut reduction as communication
- Intuitionistic variant: provider/client model [Caires & Pf’10]
  - No need to dualize types
  - Dependent types [Toninho et al.’11]
  - Monadic integration with functional language (SILL) [Toninho’15] [Griffith’15]
  - Integration in imperative language (CC0) [Willsey et al.16]
  - Polymorphism and logical relations [Perez et al.’13]
  - Exploiting categorical view (this talk)
- Classical variant: symmetric communication
  - [Wadler’12] [Caires et al.’16]

Outline

- Proofs as programs
- Linear sequent proofs as concurrent programs
  - Take 1: standard sequent calculus = synchronous communication
  - Take 2: unary ⊗R and ⊸R
  - Take 3: positive axiomatic form = asynchronous communication
- Adjoint logic
  - Structural properties and modes of truth
  - Declaration of Independence
- Concurrent operational semantics
  - Contraction, weakening, multicut, and multi-identity
  - Garbage collection

Cut as Parallel Composition

$$\frac{\Delta \vdash c : A \quad \Delta', c : A \vdash e : C}{\Delta, \Delta' \vdash e : C}\ \text{cut}$$

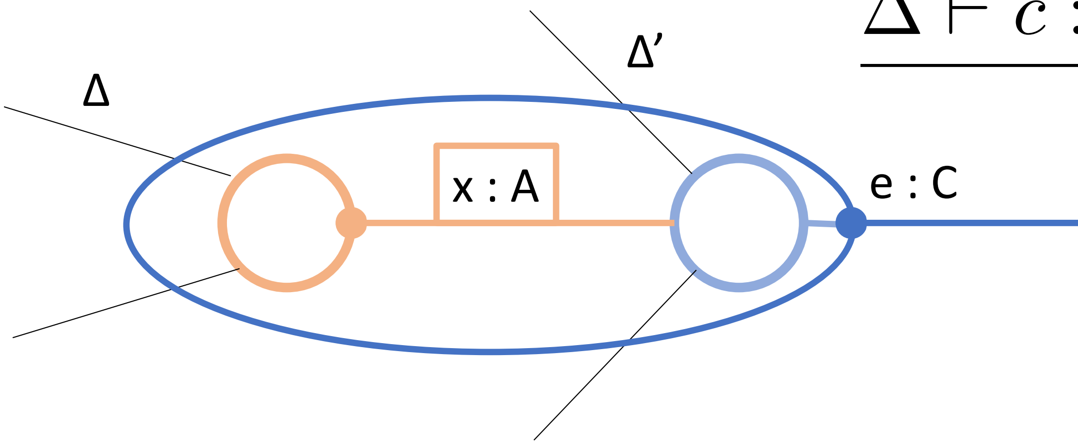


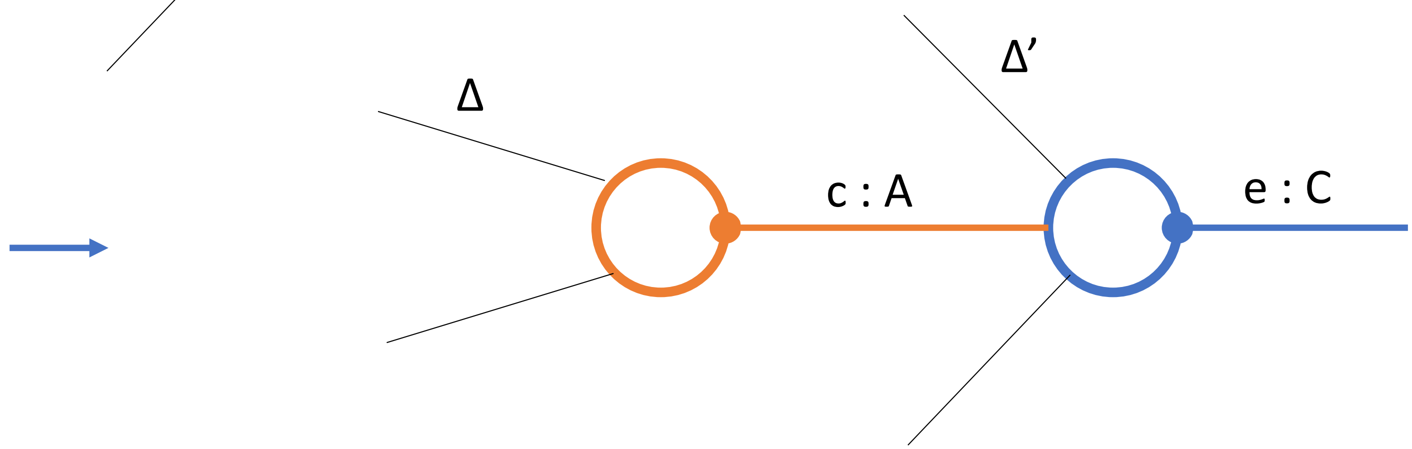
process configuration always forms a tree

process P
provides channel c : A
used by
process Q
which provides e : C

Cut as Process Spawn

$$\frac{\Delta \vdash c : A \quad \Delta', c : A \vdash e : C}{\Delta, \Delta' \vdash e : C}\ \text{cut}$$

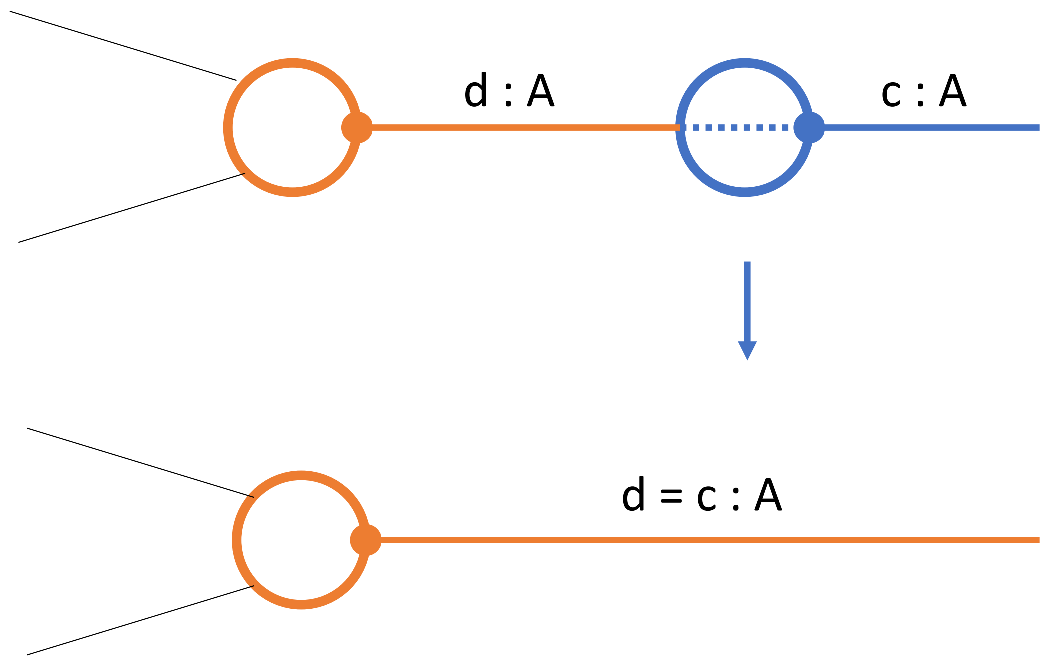




Identity as Identification

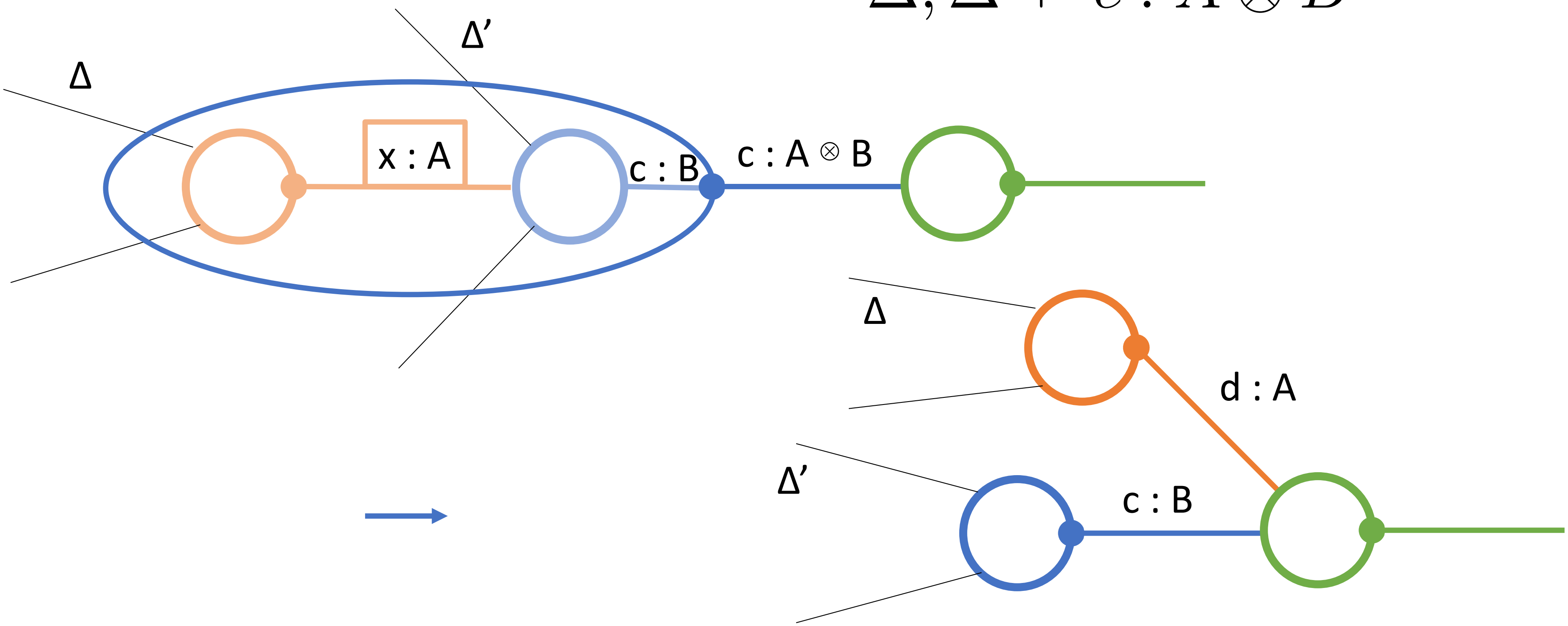
$$\frac{}{d : A \vdash c : A}\ \text{id}$$

transition of configuration models
cut of any proof with identity



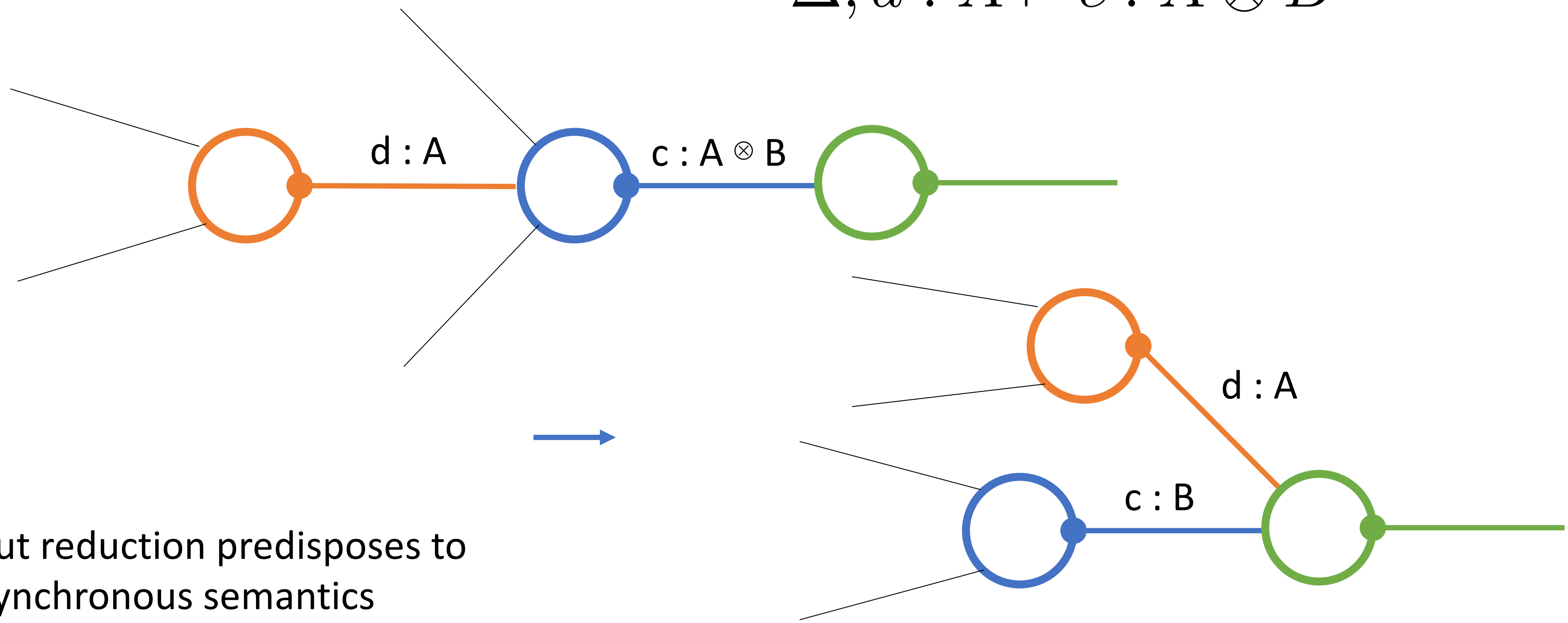
Tensor, Original

$$\frac{\Delta \vdash x : A \quad \Delta' \vdash c : B}{\Delta, \Delta' \vdash c : A \otimes B} \otimes R$$



Tensor, Simplified

$$\frac{\Delta \vdash c : B}{\Delta, d : A \vdash c : A \otimes B} \otimes R^*$$



cut reduction predisposes to synchronous semantics

Interderivable using Identity and Cut

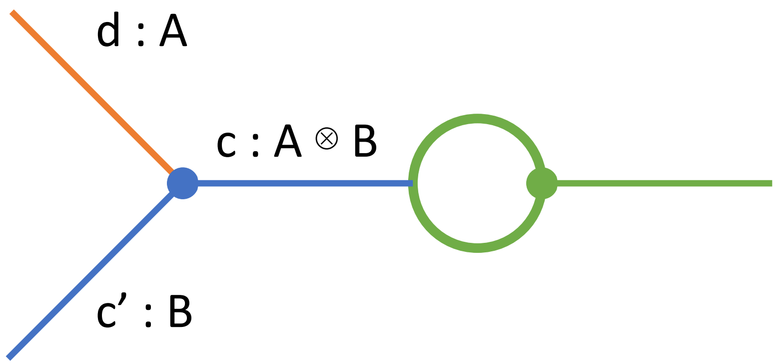
$$\dfrac{\dfrac{}{A \vdash A}\,\text{id} \qquad \Delta \vdash B}{\Delta, A \vdash A \otimes B}\,\otimes R$$

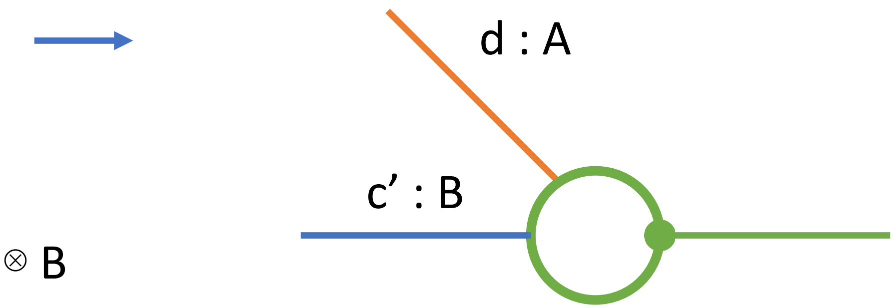
$$\dfrac{\Delta \vdash A \qquad \dfrac{\Delta' \vdash B}{\Delta', A \vdash A \otimes B}\,\otimes R^*}{\Delta, \Delta' \vdash A \otimes B}\,\text{cut}$$

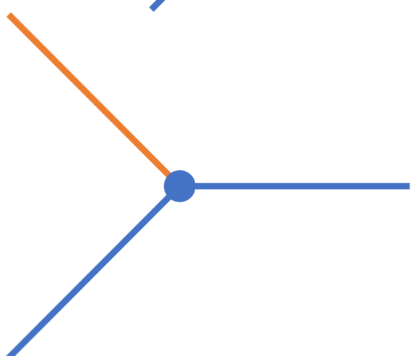
Tensor as a Message

use axiomatic form for positive providers or negative clients ("senders")

$$\frac{}{d : A, c' : B \vdash c : A \otimes B}\ \otimes RA$$







asynchronous message sending d : A along c : A ⊗ B with continuation c' : B

Interderivable with 2 cuts, 2 ids

$$\dfrac{\dfrac{}{A \vdash A}\ \text{id} \qquad \dfrac{}{B \vdash B}\ \text{id}}{A, B \vdash A \otimes B}\ \otimes R$$

because ⊗RA has no continuation an asynchronous semantics seems forced!

$$\dfrac{\Delta \vdash A \qquad \dfrac{\Delta' \vdash B \qquad \dfrac{}{A, B \vdash A \otimes B}\ \otimes RA}{\Delta', A \vdash A \otimes B}\ \text{cut}}{\Delta, \Delta' \vdash A \otimes B}\ \text{cut}$$

Positive Axiomatic Formulation

- Forces (?) asynchronous semantics
- Restore synchronous communication via mode-neutral shifts [Pf & Griffith'15]
- Can write all "sending" rules as axioms
  - Right rules for positive connectives
  - Left rules for negative connectives
- Can hide this from surface syntax, if desired

Outline

- Proofs as programs
- Linear sequent proofs as concurrent programs
  - Take 1: standard sequent calculus = synchronous communication
  - Take 2: unary $\otimes$R and $\multimap$R
  - Take 3: positive axiomatic form = asynchronous communication
- Adjoint logic
  - Structural properties and modes of truth
  - Declaration of Independence
- Concurrent operational semantics
  - Contraction, weakening, multicut, and multi-identity
  - Garbage collection

Structural Rules

- Weakening: do not need to use antecedents (affine logic)
- Contraction: may reuse antecedents (strict logic)
- Weakening + contraction = structural logic

$$\frac{\Delta \vdash e : C}{\Delta, c : A \vdash e : C}\ \mathsf{W} \qquad \frac{\Delta, c_1 : A, c_2 : A \vdash e : C}{\Delta, c : A \vdash e : C}\ \mathsf{C}$$

- Compromises the usual cut reduction and cut elimination!

Adjoint Logic [Reed'09]

- Would like to have our cake and eat it, too!
  - Allow weakening, contraction where desirable or necessary
- Challenges
  - How do we make system coherent: linear remains linear, etc.
  - Combination should be conservative over its parts
  - Logically: cut elimination, identity elimination
  - Operationally: session fidelity, global progress (even with recursion)
  - Uniform syntax and semantics

Modes of Truth

- Modes of truth k, m, n
- Every proposition A_m has an intrinsic mode of truth m
- Every mode m possesses structural properties $\sigma(m) \subseteq \{W, C\}$
  - It is possible to add exchange as an option, starting from ordered logic
- Modes are related by preorder ≤, where $m \leq k$ implies $\sigma(m) \subseteq \sigma(k)$

Modes of Truth, continued

- Syntax of propositions (and proofs) is uniform across all modes

$$\begin{array}{rcl} A_m, B_m & ::= & P_m \mid A_m \multimap B_m \mid \&\{\ell : A_m^\ell\}_{\ell \in L} \mid {\uparrow_k^m} A_k \\ & & \mid A_m \otimes B_m \mid \mathbf{1} \mid \oplus\{\ell : A_m^\ell\}_{\ell \in L} \mid {\downarrow_m^n} A_n \end{array}$$

- Shift ${\uparrow_k^m} A_k$ references proposition A_k in mode m, for m ≥ k
- Shift ${\downarrow_m^n} A_n$ references proposition A_n in mode m, for n ≥ m

Example: Intuitionistic Linear Logic

- Unrestricted mode U with σ(U) = {W,C}
- Linear mode L with σ(L) = {}
- L < U
- Mode U populated only by shifts

$$
\begin{array}{lcl}
A_{\mathsf{U}} & ::= & \uparrow_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{L}} \\
A_{\mathsf{L}} & ::= & A_{\mathsf{L}} \multimap B_{\mathsf{L}} \mid \cdots \mid \downarrow_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{U}} \\
!A_{\mathsf{L}} & \triangleq & \downarrow_{\mathsf{L}}^{\mathsf{U}} \uparrow_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{L}}
\end{array}
$$

Example: LNL [Benton'94]

- Unrestricted mode U with σ(U) = {W,C}, with all connectives
- Linear mode L with σ(L) = {}
- L < U

$$
\begin{array}{lcl}
m & ::= & L \mid U \quad \text{with} \quad L < U \\
A_m, B_m & ::= & P_m \mid A_m \multimap B_m \mid \&\{\ell : A_\ell\}_{\ell \in L} \mid \uparrow_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{L}} \\
 & & \mid A_m \otimes B_m \mid \mathbf{1} \mid \oplus\{\ell : A_\ell\}_{\ell \in L} \mid \downarrow_{\mathsf{L}}^{\mathsf{U}} A_{\mathsf{U}} \\
A_{\mathsf{U}} \rightarrow B_{\mathsf{U}} & \triangleq & A_{\mathsf{U}} \multimap B_{\mathsf{U}}
\end{array}
$$

Other Examples

- Intuitionistic S4 ~ staged computation
  - $U < V$, $\sigma(U) = \sigma(V) = \{W,C\}$
- Lax logic ~ monadic encapsulation
  - $X < U$, $\sigma(X) = \sigma(U) = \{W,C\}$
- Subexponential logic
  - Like adjoint logic, distinguished mode L
  - All other modes m contain only $\uparrow_L^m A_L$
  - $!^m A_L \triangleq \downarrow_L^m \uparrow_L^m A_L$

The Declaration of Independence

- Key to obtaining coherence, conservativity, cut elimination, uniformity
- $\Psi ::= \varepsilon \mid A_m \mid \Psi_1, \Psi_2$ where ‘,’ is associative, commutative, w/unit ε
- $\Psi \geq m$ if $k \geq m$ for every A_k in Ψ

$\Psi \vdash A_m$ is well-formed only if $\Psi \geq m$

- Truth of A_m must be independent from all A_k for $k \ngeq m$

Rules for Adjoint Logic

- Logical rules unchanged, stay at same, but arbitrary mode m
- Structural rules depend on mode properties

$$\frac{W \in \sigma(m) \quad \Delta \vdash e : C_r}{\Delta, c : A_m \vdash e : C_r}\ \mathrm{W}$$

$$\frac{C \in \sigma(m) \quad \Delta, c_1 : A_m, c_2 : A_m \vdash e : C}{\Delta, c : A_m \vdash e : C_r}\ \mathrm{C}$$

Rules for Shift

- Derived entirely from declaration of independence

$$\frac{\Psi \vdash A_k}{\Psi \vdash {\uparrow}_k^m A_k}\ {\uparrow}R \qquad \frac{k \geq r \quad \Psi, A_k \vdash C_r}{\Psi, {\uparrow}_k^m A_k \vdash C_r}\ {\uparrow}L$$

$$\frac{\Psi \geq n \quad \Psi \vdash A_n}{\Psi \vdash {\downarrow}_m^n A_n}\ {\downarrow}R \qquad \frac{\Psi, A_n \vdash C_r}{\Psi, {\downarrow}_m^n A_n \vdash C_r}\ {\downarrow}L$$

Judgmental Rules

- Cut and identity are the most interesting rules
- Standard argument for cut elimination does not work in the presence of explicit contraction
- Return to Gentzen (1935): Multicut

Multicut

- Some restrictions forced by declaration of independence

$$\frac{\begin{array}{l}\overline{c} = (c_1, \ldots, c_n) \\ \Psi \geq m \geq r\end{array} \qquad \Psi \vdash \overline{c} : A_m \quad \Psi', c_1{:}A_m, \ldots, c_n{:}A_m \vdash e : C_r}{\Psi, \Psi' \vdash e : C_r} \; \mathsf{mcut}^*$$

- If W ϵ σ(m), then n = 0 is allowed
- If C ϵ σ(m), then n > 1 is allowed
- Provider is "aware of" all client channels $\overline{\text{c}}$

Multi-identity

- Provider might be communicating with multiple clients

$$\frac{}{d : A_m \vdash \overline{c} : A_m}\ \text{id}^*$$

- If W ϵ σ(m), then $|\overline{c}| = 0$ is allowed
- If C ϵ σ(m), then $|\overline{c}| > 1$ is allowed

Outline

- Proofs as programs
- Linear sequent proofs as concurrent programs
  - Take 1: standard sequent calculus = synchronous communication
  - Take 2: unary ⊗R and –oR
  - Take 3: positive axiomatic form = asynchronous communication
- Adjoint logic
  - Structural properties and modes of truth
  - Declaration of Independence
- Concurrent operational semantics
  - Contraction, weakening, multicut, and multi-identity
  - Garbage collection

Process Interpretation

- Transitions do not care about modes at runtime
- However, some rules care about the number of clients a process has
- In this uniform semantics, computation at all structural properties is implemented by message passing
- Transitions of process configurations mimic cut reductions (not cut elimination)
- Shifts send and receive ’shift’ messages which synchronize [Pf & Griffith’15]

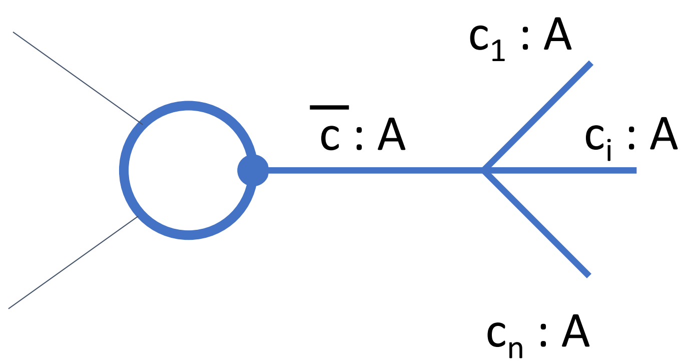
Multicut, Process Interpretation

one provider with multiple clients

$$\frac{\Delta \vdash \overline{c} : A \quad \Delta', c_1{:}A, \ldots, c_n{:}A \vdash e : C}{\Delta, \Delta' \vdash e : C}\ \text{mcut}$$

$\overline{c} = c_1, \ldots, c_n$

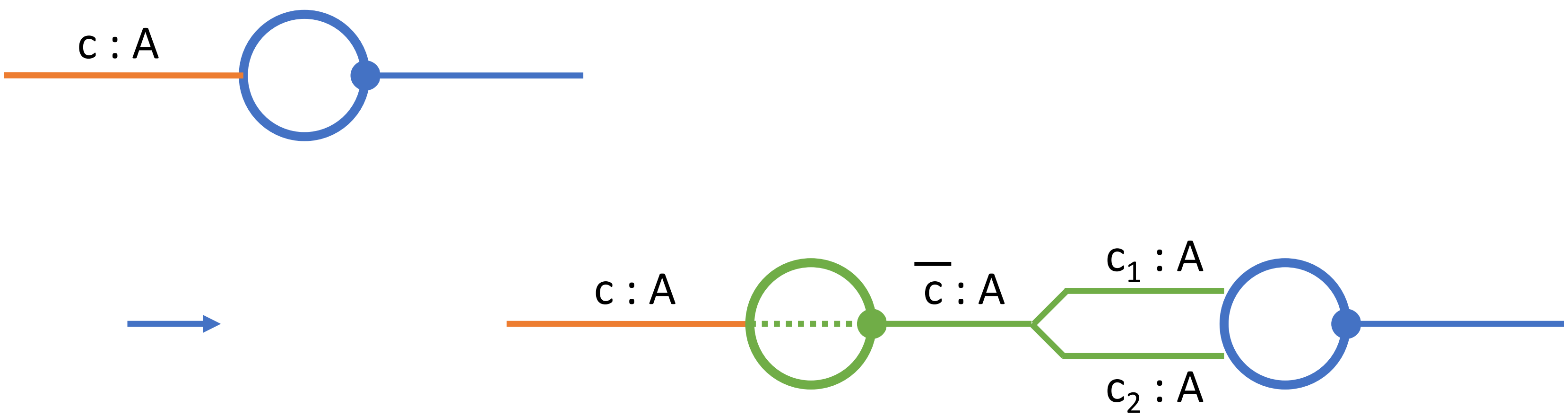
provider should know about all client channels



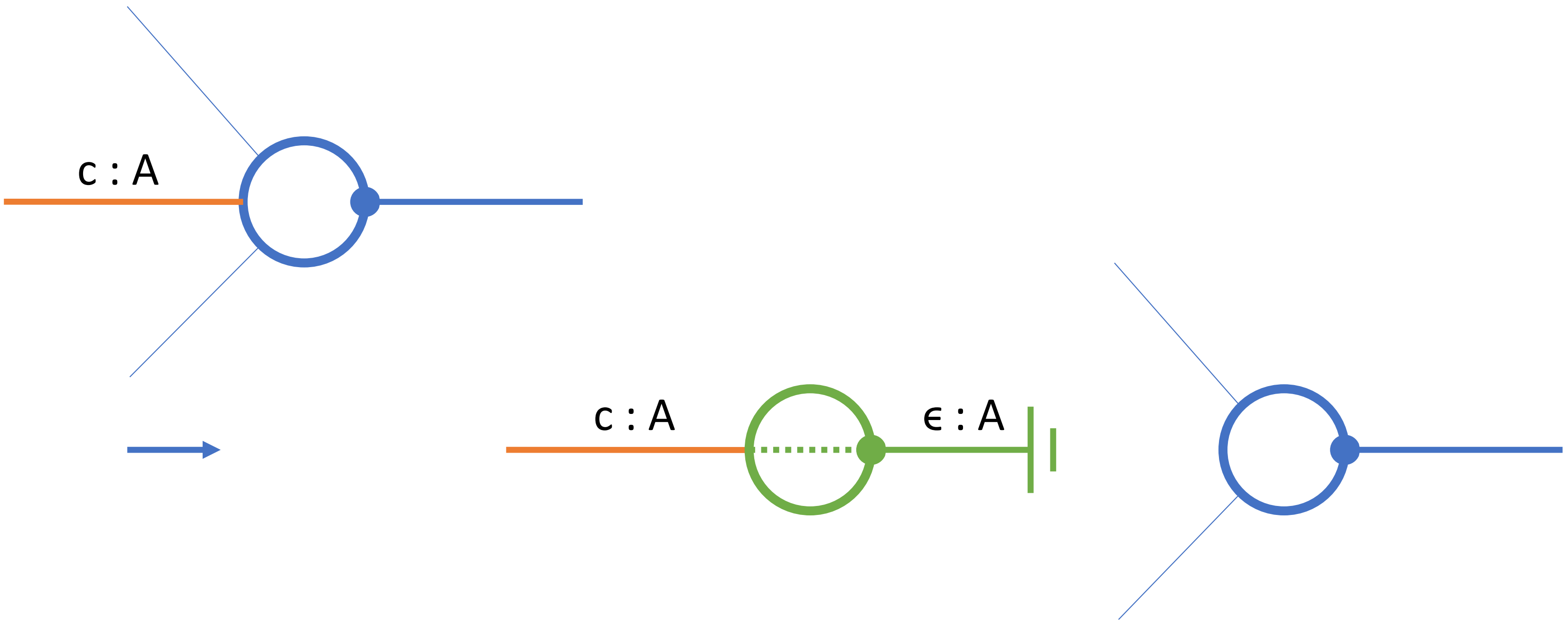
without modes, to emphasize dynamic mode independence

Key Idea: Contraction = Identity + Multicut

$$\dfrac{\dfrac{}{c : A \vdash \overline{c} : A}\ \text{id} \qquad \Delta, c_1 : A, c_2 : A \vdash e : C}{\Delta, c : A \vdash e : C}\ \text{mcut}$$

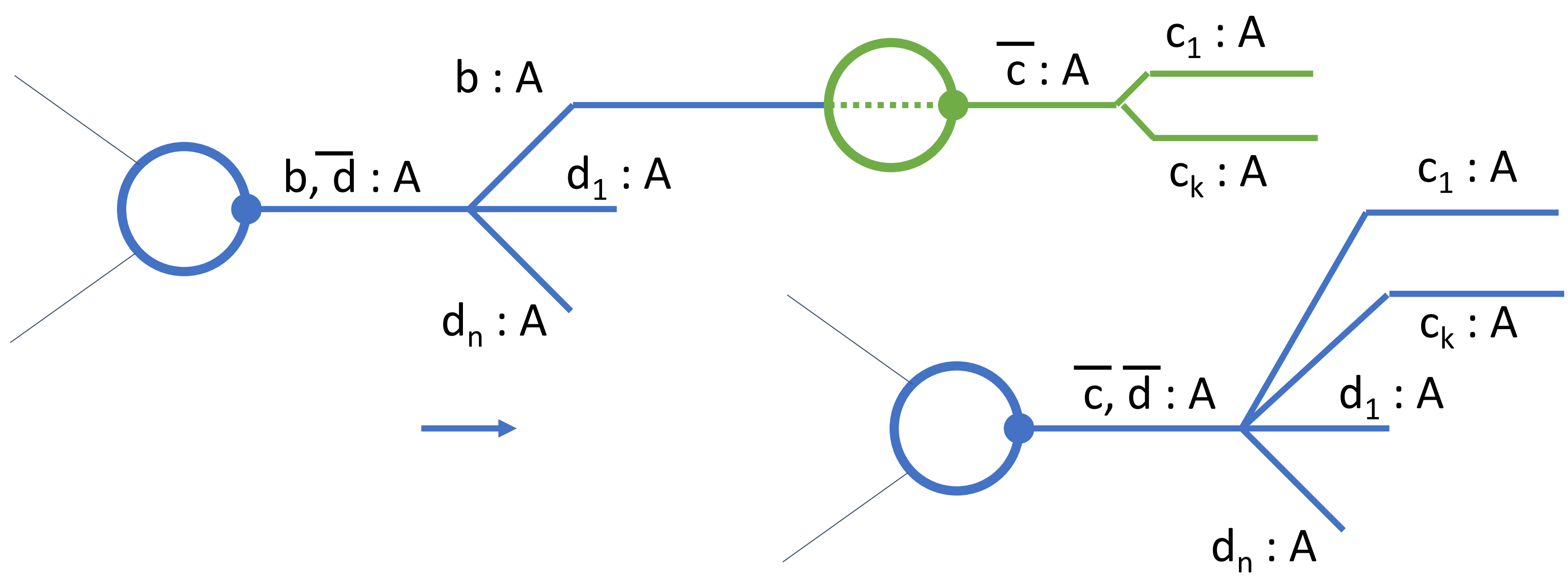


Key Idea: Weakening = Identity + Multicut



Key Idea: Identity Propagation

- Helps to implement both drop (W) and copy (C)
- Distributed garbage collection “for free”



Key Ideas: Multicast and Copy-on-Receive

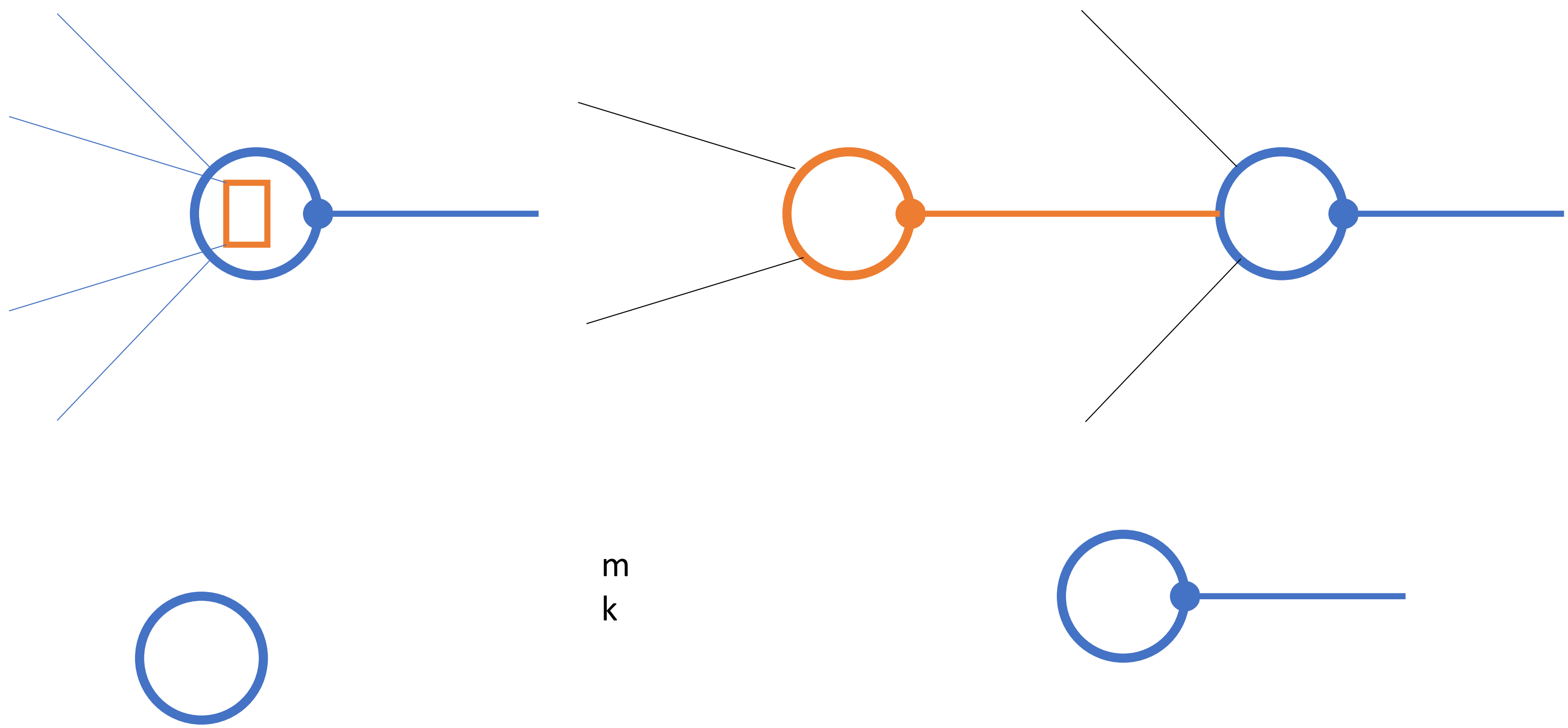
- Positive types (e.g., $A \otimes B$) do multicast, if there are multiple clients
- Negative types (e.g., A –o B) copy-on-receive, if there are multiple clients
- Previous work on !A only performed copy-on-receive

Ongoing and Future Work

- Exploit independence principle for non-uniform semantics
  - S = shared, U = unrestricted, L = linear with $S \leq U$, $U \leq S$, $L < U$, $L < S$ and $\sigma(S) = \sigma(U) = \{C,W\}$, $\sigma(L) = \{\}$ [Balzer & Pf, 2017]
  - H = locally heap-allocated, L = linear, $L < H$ with $\sigma(H) = ?$, $\sigma(L) = \{\}$
  - Proof irrelevance, intensional equality [Pf'01]
  - Ghost messages for contracts and reasoning about distributed computation
- Is there a general theorem about non-uniform compatibility?
- Surface syntax? (Uniform is possible, up to a point)
- General implementation

Summary

- Adjoint logic combines logics coherently and conservatively
- Shifts compose to a monad (one order) and comonad (other order)
- Declaration of Independence enables key results
  - Cut elimination, identity elimination
  - Conservative extension over combined fragments
- Uniform message-passing semantics via multicut and identity
  - Contraction = identity + multicut for n > 1
  - Weakening = identity + multicut for n = 0
  - Distributed garbage collection = identity propagation


m
k