# Miniprojects

15-836: Substructural Logics
Frank Pfenning

Due Friday, December 8
250 points

You may choose any of the following miniprojects and team up with a partner (which we recommend). Piazza can help in finding a partner.

Miniprojects are more open-ended than homework assignments, and solutions are often not clear cut. In evaluating the miniprojects, we will apply the following criteria:

**Correctness.** Any claimed property or theorem should hold, and your proof sketches should be correct. We expect the proofs to satisfy the standard of conference submissions, which means that their structure should be correct (for example, "*By nested induction first on $A$ and then on $\mathcal{D}$ and $\mathcal{E}$*"), and the (two or three) sample cases you present should exemplify your reasoning. You are responsible for the correctness of the cases you do not show. Similarly, claimed counterexamples should be exactly that, rather than the failure of a proof attempt. If you don't know if something is true, conjecture it.

**Elegance.** The presentation of any deductive system and auxiliary definitions should follow the principles established in this course. That said, there are many different styles of presentation in the literature with their own justifications and coherent principles that you may follow.

**Completeness.** Your development should reach a reasonably complete state. For example, if you present a sequent calculus you should prove the admissibility of cut and identity. Sometimes inference systems can be justified through their relation to previously studied ones, or judged by their own internal criteria such as preservation and progress. There are a few "further questions" given with each miniproject, but we do not expect that all of them would be answered, or even have a reasonable answer. Of course, the same might be true even for the core questions, so you have to be alert for (unintended) pitfalls.

**Presentation.** Your miniproject should tell a story and not just consist of a series of definitions and proofs. There should be an introduction that sets the stage, has relevant pointers to the literature, and summarizes your approach and main findings. You should use (generally small) examples to illustrate your systems. There should be a conclusion that states the related problems that you do not solve and possibly point to future work.

You may (and probably should) use and cite relevant prior work you can find but you don't need to do a major literature search. Putting prior work from a different context into the setting of this course may be a worthwhile miniproject as long as it is not just a matter of changing notation.

You should hand in a single file, `mp.pdf`, via Gradescope.

# 1 Ordered Logic

We have worked with ordered logic extensively, including cut elimination and modeling a variety of algorithms using ordered forward inference. But in our message passing interpretation and also our development of adjoint logic we have always assumed exchange.

1. Give a formulation of adjoint logic based on modes with three possible explicit structural rules: exchange, weakening, and contraction.
2. Prove cut and identity elimination. For readability, you may restrict yourself to the fragment without weakening and contraction (although it should hold for your full system).
3. What can we say about a mode without exchange, but weakening or contraction or both? Or do weakening and contraction for mode $m$ only "make sense" if we also assume that $m$ allows exchange?
4. What can we say about the interaction of two different modes that are both just ordered, that is, admit no structural rules?
5. Other possible directions:
   - What are suitable statics and dynamics for *message passing* with ordered and linear types? What does order express and enforce?
   - What are suitable statics and dynamics for *futures-based shared memory* with ordered and linear types? What does order express and enforce?
   - Forward inference for linear states corresponds to multiset rewriting while for ordered states it corresponds to string rewriting. What does forward inference look like if we both linear and ordered proposition in our state? Can you give one or more examples where such mixed rewriting is suitable for a given problem?
   - What if we also have structural propositions?

## 2 Classical Linear Logic

In Girard's seminal 1987 paper on *Linear Logic* he presents a single-sided classical linear logic using a dualizing operator on propositions. This dualizing operator is also popular in many papers on session types.

1. Give a message-passing interpretation of classical linear logic including recursion and exponentials.
2. Prove progress and preservation.
3. There are several interpretations of classical linear logic into intuitionistic linear logic (see, for example, O. Laurent, LICS 2018). What is the computational meaning of at least one of these translations?
4. Some related questions to investigate:
   - How does the meaning of $!A$ and $?A$ related to the computational interpretation of $\downarrow\uparrow A$ we have given for intuitionistic linear logic?
   - What would be a *shared memory* interpretation for classical linear logic (assuming it exists), allowing recursion? How do the exponentials fit into the picture.
   - There are two-sided formulations of classical linear logic without an explicit dualization operator. How does this impact the process language and its computational interpretation?
   - Do you see any connections between your interpretations (message passing or shared memory) and extant computational interpretations of *classical logic*?

## 3 Polarized Logic

We can *polarize* essentially any logic by requiring explicit modalities to switch from positive to negative and negative to positive propositions. For example, polarized linear propositions would be

$$\begin{array}{lll} \text{Negatives} & A^-, B^- & ::= & P^- \mid A^+ \multimap B^- \mid \&\{\ell : A_\ell^-\}_{\ell \in L} \mid \Uparrow A^+ \\ \text{Positives} & A^+, B^+ & ::= & P^+ \mid A^+ \otimes B^+ \mid \mathbf{1} \mid \oplus\{\ell : A_\ell^+\}_{\ell \in L} \mid \Downarrow A^- \end{array}$$

We have written double-arrow shifts because these are not quite the shifts of adjoint logic as can be seem from $A^+ \multimap B^-$ that switches between polarities without a shift.

1. Give a formulation of polarized adjoint logic.
2. Is there a uniform notion of shift, or do we have different kinds of shift with different rules and properties?
3. Sketch the proof of cut elimination (where you may limit yourself to a 2-mode linear/nonlinear calculus if you wish).
4. Give a dynamics for the polarized adjoint calculus (restricted to the 2-mode linear/nonlinear fragment if you wish). What are the differences to the usual dynamics?
5. Further questions to investigate:
   - In Levy's call-by-push-value, which is formulated in natural deduction, the typing judgments are $\Gamma^+ \vdash e : A^-$ ($e$ is a computation) and $\Gamma^+ \vdash v : A^+$ ($v$ is a value). Does your formulation provide a rational reconstruction of the restriction to purely positive contexts? What would this mean for a message passing or shared memory interpretation?
   - How does your dynamics relate to Levy's call-by-push-value?

## 4 Bunched Logic

*Bunched logic* (BI) is an alternative to adjoint logic, also mixing linear and structural logic. It has found significant applications as the basis for *separation logic* for reasoning about imperative programs. In bunched logic, the antecedents are built up as trees from a linear combination $\Delta \,,\, \Delta'$ where the comma corresponds to a linear $\otimes$-like connective, and $\Delta \,;\, \Delta'$ where the semicolon corresponds to a structural conjunction $\wedge$. There are also units for comma and semicolon corresponding to $\mathbf{1}$ and $\top$, respectively.

1. Write out a sequent calculus for (intuitionistic) BI.
2. Sketch the proof of the admissibility of cut and identity.
3. Give a process assignment to sequent calculus. Strive either for a message-passing or a shared memory interpretation in a style similar to what we have used in this course.
4. Sketch the proofs of progress and preservation.
5. Further potential questions to investigate
   - How does BI compare to the mixed linear/nonlinear logic?
   - How do the dynamics relate?
   - Can we generalize BI to have multiple modes with different structural rules in analogy with adjoint logic?

## 5 Contextual Logic

Originally, contextual modal type theory (Nanevski at al, ToCL 2008) was developed for two main purposes: support metaprogramming with open code, and capture the concept of metavariable type-theoretically. It generalized the constructor $\mathbf{quote}\ M : \Box A$ ($M$ is a closed term of type $A$) from modal logic to $\mathbf{box}\,(\Gamma.\,M)$ ($M$ may only depend on the variables in context $\Gamma$).

Actually, we have already been using metavariables without internalizing them into the type system because all top level definitions, written as $p\,(x : A)\,(y_1 : A_1)\,\ldots\,(y_n : A_n) = P(x, y_1, \ldots, y_n)$ essentially give $p$ a contextual type $[y_1 : A_1, \ldots, y_n : A_n](x : A)$. That is, in MPass and SAX, process variables are metavariables.

1. First, in a mixed linear/nonlinear logic: What is the right decomposition of $[\Delta]A$ (which requires a generalization of $\downarrow$ or $\uparrow$ or both)?
2. Prove admissibility of cut and identity in the sequent calculus formulation of contextual mixed linear/nonlinear logic.
3. Give an operational interpretation, either under a message passing or a shared memory interpretation.
4. Further questions to investigate:
   - Can we generalize the contextual linear/nonlinear logic to a full adjoint logic?
   - What additional computational or logical phenomena can be modeled by contextualizing adjoint logic (if any)?