Midterm Exam

15-836: Substructural Logics
Frank Pfenning

Thursday, October 12, 2023
150 points

Name:	Sample Solution	Andrew ID:	**fp**

Instructions

- This exam is closed-book, closed-notes.
- There are several appendices for reference.
- Reference pages will not be scanned (you may tear them off).
- Try to keep your answers inside the answer boxes to ensure proper scanning.
- You have 80 minutes to complete the exam.
- There are 5 problems.
- The maximal exam score is 150 points

	Ordered Inference	Cut	Message Passing	Replication	Adjoint Logic	
	Prob 1	Prob 2	Prob 3	Prob 4	Prob 5	Total
Score	**30**	**30**	**30**	**30**	**30**	**150**
Max	30	30	30	30	30	150

1 Ordered Inference (30 pts)

In this problem we use the binary representation of numbers as an ordered state. For example, the number $6 = (110)_2$ would be represented as $\epsilon\ 1\ 1\ 0$. We write $\Omega(n)$ for the binary representation of n, so $\Omega(6) = \epsilon\ 1\ 1\ 0$. You may assume that $\Omega(n)$ has no leading zeros. As a reminder, the definition of increment

$$\frac{0\ \mathsf{inc}}{1} \qquad \frac{1\ \mathsf{inc}}{\mathsf{inc}\ 0} \qquad \frac{\epsilon\ \mathsf{inc}}{\epsilon\ 1}$$

which satisfies that $\Omega(n)\ \mathsf{inc} \longrightarrow^* \Omega(n+1)$. In the tasks below, you may use inc and other auxiliary propositions and rules as you see fit.

Task 1 (15 pts) Define an ordered program to compute the bit parity of a binary number.

Initial state: $\Omega(n)\ \mathsf{par}$.

Final state: b where $b = 0$ means n had even parity and $b = 1$ means n had odd parity.

For example, $\epsilon\,1\,0\,1\ \mathsf{par} \longrightarrow^* 0$ and $\epsilon\,1\,0\ \mathsf{par} \longrightarrow^* 1$.

We use an auxiliary proposition xor (but there are many other correct solutions)

$$\frac{\mathsf{par}}{\mathsf{xor}\ 0} \qquad \frac{0\ \mathsf{xor}\ 0}{\mathsf{xor}\ 0} \qquad \frac{0\ \mathsf{xor}\ 1}{\mathsf{xor}\ 1} \qquad \frac{1\ \mathsf{xor}\ 0}{\mathsf{xor}\ 1} \qquad \frac{1\ \mathsf{xor}\ 1}{\mathsf{xor}\ 0} \qquad \frac{\epsilon\ \mathsf{xor}}{\cdot}$$

Task 2 (15 pts) Define an ordered program to compute the number of bits in a binary number.

Initial state: $\Omega(n)$ size, where n is a binary number with $n \geq 1$.

Final state: $\Omega(|n|)$ where $|n|$ is the number of bits (0, 1) in n.

For example, ϵ $\mathsf{size} \longrightarrow^* \epsilon$ and $\epsilon\,1\,0\,1\,0\,1\,1$ $\mathsf{size} \longrightarrow^* \epsilon\,1\,1\,0$.

We use the rules for inc from the problem statement.

$$\frac{1\ \mathsf{size}}{\mathsf{size}\ \mathsf{inc}} \qquad \frac{0\ \mathsf{size}}{\mathsf{size}\ \mathsf{inc}} \qquad \frac{\epsilon\ \mathsf{size}}{\epsilon}$$

2 Cut (30 pts)

In the distant past, in a misguided attempt to model stacks and/or queues, I decided to reformulate the rule of cut in ordered logic so it can only cut the proposition at the left end of the antecedents. I called the resulting rule lop_L. (You can find the rules of ordered logic in Appendix A.)

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \Omega_L \vdash A\end{array} \quad \begin{array}{c}\mathcal{E}\\ A\,\Omega_R \vdash C\end{array}}{\Omega_L\,\Omega_R \vdash C}\ \mathsf{lop}_L$$

Task 3 (10 pts) Recall: The usual proof of the admissibility of cut in ordered logic proceeds by . . .

> . . . nested induction, first on the structure of the cut proposition A, then on the structure of the derivations of the first and second premise ($\mathcal{D}$ and $\mathcal{E}$, respectively).

Task 4 (10 pts) Give an example case where the usual proof of the admissibility of cut would fail in an attempt to prove the admissibility of lop_L (in the system without cut or lop).

> In the situation below, we cannot push up the lop_L because A is no longer the leftmost antecedent. In other words, we cannot appeal to the induction hypothesis on A, $\mathcal{D}$, and $\mathcal{E}'$.
>
> $$\dfrac{\begin{array}{c}\mathcal{D}\\ \Omega_L \vdash A\end{array} \quad \dfrac{\begin{array}{c}\mathcal{E}'\\ B\,A\,\Omega_R \vdash C\end{array}}{A\,\Omega_R \vdash B \setminus C}\ \setminus R}{\Omega_L\,\Omega_R \vdash B \setminus C}\ \mathsf{lop}_L$$

Task 5 (10 pts) Either prove that lop_L is admissible (in the system without cut and lop) or give a counterexample to its admissibility.

> Since lop_L is a special case of cut, and cut is admissible, so is lop_L.

3 Message Passing (30 pts)

Consider the recursive type of unary natural numbers in linear logic with recursion, where proofs are interpreted as linear message-passing programs.

$$\mathsf{nat} = \oplus\{\mathsf{zero} : \mathbf{1}, \mathsf{succ} : \mathsf{nat}\}$$

According to a homework assignment we know this type is subject to weakening and contraction.

In the answers below you may use the mathematical syntax we have been using in lecture or the MPASS syntax. We won't hold you to all the lexical details, but it should be otherwise correct and linear. You can also use prior definition in later answers, and write auxiliary processes as you see fit. (You can find the statics and dynamics of MPASS in Appendix B.)

Task 6 (5 pts) Complete the definition of the zero process.

```
type nat = +{'zero : 1, 'succ : nat}
proc zero (x : nat) = send x 'zero ; send x ()
```

Task 7 (5 pts) Complete the definition of the successor process.

```
proc succ (x : nat) (y : nat) =
send x 'succ ; fwd x y
```

Task 8 (5 pts) Complete the following definition of *drop* that consumes its argument.

```
proc drop (u : 1) (x : nat) =
recv x ( 'zero => fwd u x
       | 'succ => call drop u x )
```

Task 9 (15 pts) Complete the following definition of *dup* that duplicates its argument. Recall that `A * B` is concrete syntax for $A \otimes B$.

```
proc dup (p : nat * nat) (x : nat) =
recv x ( 'zero => p1 <- call zero p1 ;
                  send p p1 ;
                  send p 'zero ; fwd p x
       | 'succ => p' <- call dup p' x ;
                  recv p' (x' =>
                  p1 <- call succ p1 x' ;
                  send p p1 ;
                  call succ p p') )
```

4 Replication (30 pts)

We can try to eliminate the exponential modality $!A$ from linear logic by defining instead

$$\nabla A = 1 \mathbin{\&} A \mathbin{\&} (\nabla A \otimes \nabla A)$$

Because this family of propositions is defined recursively, for this problem we allow *infinite derivations*. They are finitely represented as *circular proofs*, when possible. (You can find the rules for linear logic without the exponential in Appendix C.)

Task 10 (10 pts) Construct three derived *left* rules for ∇, applying inversion to the premises as much as possible. Because ∇ is defined recursively, you may not be able to eliminate ∇ or other connectives from the premises as we usually expect. Assign names ∇L_1, ∇L_2 and ∇L_3 to your rules. You do not need to show the derivation, just the final rules.

$$\frac{\Delta \vdash C}{\Delta, \nabla A \vdash C}\,\nabla L_1 \qquad \frac{\Delta, A \vdash C}{\Delta, \nabla A \vdash C}\,\nabla L_2 \qquad \frac{\Delta, \nabla A, \nabla A \vdash C}{\Delta, \nabla A \vdash C}\,\nabla L_3$$

Task 11 (10 pts) Construct one derived *right* rule for ∇, applying inversion to the premises as much as possible. Again, ∇ or other connectives may remain in the premises. Assign the name ∇R to your rule. You do not need to show the derivation, just the final rule.

$$\frac{\Delta \vdash \mathbf{1} \quad \Delta \vdash A \quad \Delta \vdash \nabla A \otimes \nabla A}{\Delta \vdash \nabla A}\,\nabla R$$

Task 12 (10 pts) Show that the identity on ∇A is admissible by constructing a (possibly circular) proof of $\nabla A \vdash \nabla A$ using the identity only at A.

$$\dfrac{\dfrac{\dfrac{}{\cdot \vdash \mathbf{1}}\,\mathbf{1}R}{\nabla A \vdash \mathbf{1}}\,\nabla L_1 \qquad \dfrac{\dfrac{}{A \vdash A}\,\mathsf{id}_A}{\nabla A \vdash A}\,\nabla L_2 \qquad \dfrac{\dfrac{\overset{(x)}{\nabla A \vdash \nabla A} \quad \overset{(x)}{\nabla A \vdash \nabla A}}{\nabla A, \nabla A \vdash \nabla A \otimes \nabla A}\,\otimes R}{\nabla A \vdash \nabla A \otimes \nabla A}\,\nabla L_3}{\nabla A \vdash \nabla A \quad (x)}\,\nabla R$$

5 Adjoint Logic (30 pts)

We have been reading the sequent calculus rules bottom-up, as if were are constructing proofs in a goal-directed way. When writing a proof goal $\Delta \vdash A_m$ we *presuppose* that $\Delta \geq m$, and this property must be preserved reading rules bottom-up. (You can find the usual rules for adjoint logic in Appendix D.)

Rules may also be read top-down for different kinds of proof search procedures.

Task 13 (30 pts) Fill in additional premises as needed in the following rules so that the conclusion satisfies our presupposition and the correct structural properties whenever all premises do. Do not add any redundant conditions. Write "(none)" when no conditions are needed.

The well-formedness conditions on the upshift $m \geq k$ is optional, because it could be interpreted as being enforced in the syntax rather than in the rules.

$$\frac{\boxed{\text{(none)}}}{A_m \vdash A_m}\ \mathsf{id} \qquad \frac{\boxed{\text{(none)}} \quad \Delta \vdash A_m \quad \Delta', A_m \vdash C_r}{\Delta, \Delta' \vdash C_r}\ \mathsf{cut}$$

$$\frac{\boxed{\mathsf{W} \in \sigma(m), m \geq r} \quad \Delta \vdash C_r}{\Delta, A_m \vdash C_r}\ \mathsf{weaken} \qquad \frac{\boxed{\mathsf{C} \in \sigma(m)} \quad \Delta, A_m, A_m \vdash C_r}{\Delta, A_m \vdash C_r}\ \mathsf{contract}$$

$$\frac{\boxed{\Delta \geq m \quad (m \geq k)} \quad \Delta \vdash A_k}{\Delta \vdash {\uparrow}_k^m A_k}\ {\uparrow}R \qquad \frac{\boxed{\text{(none)} \quad (m \geq k)} \quad \Delta, A_k \vdash C_r}{\Delta, {\uparrow}_k^m A_k \vdash C_r}\ {\uparrow}L$$

A Ordered Logic

$$\frac{}{A \vdash A}\ \mathsf{id}_A \qquad \frac{}{P \vdash P}\ \mathsf{id}^*_P \qquad \frac{\Omega \vdash A \quad \Omega_L\ A\ \Omega_R \vdash C}{\Omega_L\ \Omega\ \Omega_R \vdash C}\ \mathsf{cut}_A$$

$$\frac{\Omega_1 \vdash A \quad \Omega_2 \vdash B}{\Omega_1\ \Omega_2 \vdash A \bullet B}\ {\bullet}R \qquad \frac{\Omega_1\ A\ B\ \Omega_2 \vdash C}{\Omega_1\ (A \bullet B)\ \Omega_2 \vdash C}\ {\bullet}L$$

$$\frac{A\ \Omega \vdash B}{\Omega \vdash A \setminus B}\ {\setminus}R \qquad \frac{\Omega_A \vdash A \quad \Omega_L\ B\ \Omega_R \vdash C}{\Omega_L\ \Omega_A\ (A \setminus B)\ \Omega_R \vdash C}\ {\setminus}L$$

$$\frac{\Omega\ A \vdash B}{\Omega \vdash B \,/\, A}\ /R \qquad \frac{\Omega_A \vdash A \quad \Omega_L\ B\ \Omega_R \vdash C}{\Omega_L\ (B \,/\, A)\ \Omega_A\ \Omega_R \vdash C}\ /L$$

$$\frac{\Omega \vdash A \quad \Omega \vdash B}{\Omega \vdash A \mathbin{\&} B}\ \&R \qquad \frac{\Omega_L\ A\ \Omega_R \vdash C}{\Omega_L\ (A \mathbin{\&} B)\ \Omega_R \vdash C}\ \&L_1 \quad \frac{\Omega_L\ B\ \Omega_R \vdash C}{\Omega_L\ (A \mathbin{\&} B)\ \Omega_R \vdash C}\ \&L_1$$

$$\frac{\Omega \vdash A}{\Omega \vdash A \oplus B}\ {\oplus}R_1 \quad \frac{\Omega \vdash B}{\Omega \vdash A \oplus B}\ {\oplus}R_2 \qquad \frac{\Omega_L\ A\ \Omega_R \vdash C \quad \Omega_L\ B\ \Omega_R \vdash C}{\Omega_L\ (A \oplus B)\ \Omega_R \vdash C}\ {\oplus}L$$

$$\frac{}{\cdot \vdash \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Omega_L\ \Omega_R \vdash C}{\Omega_L\ (\mathbf{1})\ \Omega_R \vdash C}\ \mathbf{1}L$$

$$\frac{}{\Omega \vdash \top}\ {\top}R \qquad \text{no } {\top}L \text{ rules}$$

$$\text{no } \mathbf{0}R \text{ rules} \qquad \frac{}{\Omega_L\ (\mathbf{0})\ \Omega_R \vdash C}\ \mathbf{0}L$$

B Linear Message Passing

Statics for MPass.

$$\frac{\Delta \vdash P(x) :: (x : A) \quad \Delta', x : A \vdash Q(x) :: (z : C)}{\Delta, \Delta' \vdash x_A \leftarrow P(x) \ ; \ Q(x) :: (z : C)}\ \mathsf{cut} \qquad \frac{}{y : A \vdash \textbf{fwd}\ x\ y :: (x : A)}\ \mathsf{id}$$

$$\frac{}{\cdot \vdash \textbf{send}\ x\ (\,) :: (x : \mathbf{1})}\ \mathbf{1}R \qquad \frac{\Delta \vdash Q :: (z : C)}{\Delta, x : \mathbf{1} \vdash \textbf{recv}\ x\ ((\,) \Rightarrow Q) :: (z : C)}\ \mathbf{1}L$$

$$\frac{\Delta \vdash P :: (x : A_k) \quad (k \in L)}{\Delta \vdash \textbf{send}\ x\ k \ ; P :: (x : \oplus\{\ell : A_\ell\}_{\ell \in L})}\ \oplus R \qquad \frac{\Delta, x : A_\ell \vdash Q_\ell :: (z : C) \quad (\forall \ell \in L)}{\Delta, x : \oplus\{\ell : A_\ell\}_{\ell \in L} \vdash \textbf{recv}\ x\ (\ell \Rightarrow Q_\ell)_{\ell \in L} :: (z : C)}\ \oplus L$$

$$\frac{\Delta \vdash P :: (x : B)}{\Delta, w : A \vdash \textbf{send}\ x\ w \ ; P :: (x : A \otimes B)}\ \otimes R^* \qquad \frac{\Delta', y : A, x : B \vdash Q(y) :: (z : C)}{\Delta', x : A \otimes B \vdash \textbf{recv}\ x\ (y \Rightarrow Q(y)) :: (z : C)}\ \otimes L$$

$$\frac{\Delta \vdash P_\ell :: (x : A_\ell) \quad (\forall \ell \in L)}{\Delta \vdash \textbf{recv}\ x\ (\ell \Rightarrow P_\ell)_{\ell \in L} :: (x : \&\{\ell : A_\ell\}_{\ell \in L})}\ \& R \qquad \frac{\Delta, x : A_k \vdash Q :: (z : C) \quad (k \in L)}{\Delta, x : \&\{\ell : A_\ell\}_{\ell \in L} \vdash \textbf{send}\ x\ k \ ; Q :: (z : C)}\ \& L$$

$$\frac{\Delta, y : A \vdash P :: (x : B)}{\Delta \vdash \textbf{recv}\ x\ (y \Rightarrow P(y)) :: (x : A \multimap B)}\ \multimap R \qquad \frac{\Delta', x : B \vdash Q :: (z : C)}{\Delta', w : A, x : A \multimap B \vdash \textbf{send}\ x\ w \ ; Q :: (z : C)}\ \multimap L^*$$

$$\frac{p\ (x : A)\ \overline{(y : B)} = P(x, \overline{y}) \in \Sigma}{\overline{(y : B)} \vdash \textbf{call}\ x\ \overline{y} :: (x : A)}\ \mathsf{call}$$

Dynamics for MPass.

$$\begin{array}{lll}
\mathsf{proc}(x_A \leftarrow P(x) \ ; \ Q(x)) & \longrightarrow & \mathsf{proc}(P(a)), \mathsf{proc}(Q(a)) \quad (a \text{ fresh}) \\
\mathsf{proc}(P(b)), \mathsf{proc}(\textbf{fwd}\ a\ b) & \longrightarrow & \mathsf{proc}(P(a)) \\
\mathsf{proc}(\textbf{fwd}\ a\ b), \mathsf{proc}(Q(a)) & \longrightarrow & \mathsf{proc}(Q(b)) \\
\mathsf{proc}(\textbf{send}\ x\ k \ ; P), \mathsf{proc}(\textbf{recv}\ x\ (\ell \Rightarrow Q_\ell)_{\ell \in L}) & \longrightarrow & \mathsf{proc}(P), \mathsf{proc}(Q_k) \qquad (k \in L) \\
\mathsf{proc}(\textbf{send}\ x\ (\,)), \mathsf{proc}(\textbf{recv}\ x\ ((\,) \Rightarrow Q)) & \longrightarrow & Q \\
\mathsf{proc}(\textbf{send}\ a\ b \ ; P), \mathsf{proc}(\textbf{recv}\ a\ (y \Rightarrow Q(y))) & \longrightarrow & \mathsf{proc}(P), \mathsf{proc}(Q(b)) \\
\mathsf{proc}(\textbf{call}\ p\ a\ \overline{b}) & \longrightarrow & \mathsf{proc}(P(a, \overline{b})) \\
 & & \text{where } p\ (x : A)\ \overline{(y : B)} = P(x, \overline{y}) \in \Sigma
\end{array}$$

C Linear Logic (without Exponential)

$$\frac{\Delta \vdash A \quad \Delta', A \vdash C}{\Delta, \Delta' \vdash C}\ \mathsf{cut}_A \qquad \frac{}{A \vdash A}\ \mathsf{id}_A \qquad \frac{}{P \vdash P}\ \mathsf{id}_P^*$$

$$\frac{\Delta \vdash A}{\Delta \vdash A \oplus B}\ \oplus R_1 \quad \frac{\Delta \vdash B}{\Delta \vdash A \oplus B}\ \oplus R_2 \qquad \frac{\Delta, A \vdash C \quad \Delta, B \vdash C}{\Delta, A \oplus B \vdash C}\ \oplus L$$

$$\text{no } \mathbf{0}R \text{ rule} \qquad \frac{}{\Delta, \mathbf{0} \vdash C}\ \mathbf{0}L$$

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

$$\frac{}{\cdot \vdash \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Delta \vdash C}{\Delta, \mathbf{1} \vdash C}\ \mathbf{1}L$$

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

$$\frac{}{\Delta \vdash \top}\ \top R \qquad \text{no } \top L \text{ rule}$$

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

D Adjoint Logic

Syntax with $\ell \geq m \geq k$ and $\sigma(m) \subseteq \{\mathsf{W}, \mathsf{C}\}$.

$$A_m ::= P_m \mid A_m \to B_m \mid A_m \times B_m \mid \mathbf{1} \mid A_m \mathbin{\&} B_m \mid \top \mid A_m + B_m \mid \mathbf{0} \mid {\uparrow}_k^m A_k \mid {\downarrow}_m^\ell A_\ell$$

Rules.

$$\frac{\mathsf{W} \in \sigma(m) \quad \Delta \vdash C_r}{\Delta, A_m \vdash C_r}\ \mathsf{weaken} \qquad \frac{\mathsf{C} \in \sigma(m) \quad \Delta, A_m, A_m \vdash C_r}{\Delta, A_m \vdash C_r}\ \mathsf{contract}$$

$$\frac{}{A_m \vdash A_m}\ \mathsf{id} \qquad \frac{\Delta \geq m \geq r \quad \Delta \vdash A_m \quad \Delta', A_m \vdash C_r}{\Delta, \Delta' \vdash C_r}\ \mathsf{cut}$$

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

$$\frac{\Delta \geq \ell \quad \Delta \vdash A_\ell}{\Delta \vdash {\downarrow}_m^\ell A_\ell}\ {\downarrow}R \qquad \frac{\Delta, A_\ell \vdash C_r}{\Delta, {\downarrow}_m^\ell A_\ell \vdash C_r}\ {\downarrow}L$$

$$\frac{\Delta, A_m \vdash B_m}{\Delta \vdash A_m \to B_m}\ {\to}R \qquad \frac{\Delta \geq m \quad \Delta \vdash A_m \quad \Delta', B_m \vdash C_r}{\Delta, \Delta', A_m \to B_m \vdash C_r}\ {\to}L$$

$$\frac{\Delta \vdash A_m \quad \Delta' \vdash B_m}{\Delta, \Delta' \vdash A_m \times B_m}\ {\times}R \qquad \frac{\Delta, A_m, B_m \vdash C_r}{\Delta, A_m \times B_m \vdash C_r}\ {\times}L$$

$$\frac{}{\cdot \vdash \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Delta \vdash C_r}{\Delta, 1 \vdash C_r}\ \mathbf{1}L$$

$$\frac{\Delta \vdash A_m \quad \Delta \vdash B_m}{\Delta \vdash A_m \mathbin{\&} B_m}\ {\&}R \qquad \frac{\Delta, A_m \vdash C_r}{\Delta, A_m \mathbin{\&} B_m \vdash C_r}\ {\&}L_1 \quad \frac{\Delta, B_m \vdash C_r}{\Delta, A_m \mathbin{\&} B_m \vdash C_r}\ {\&}L_2$$

$$\frac{}{\Delta \vdash \top}\ \top R \qquad \text{no } \top L \text{ rule}$$

$$\frac{\Delta \vdash A_m}{\Delta \vdash A_m + B_m}\ {+}R_1 \quad \frac{\Delta \vdash B_m}{\Delta \vdash A_m + B_m}\ {+}R_2 \qquad \frac{\Delta, A_m \vdash C_r \quad \Delta, B_m \vdash C_r}{\Delta, A_m + B_m \vdash C_r}\ {+}L$$

$$\text{no } \mathbf{0}R \text{ rule} \qquad \frac{}{\Delta, \mathbf{0} \vdash C_r}\ \mathbf{0}L$$