

Constructive Logic (15-317), Fall 2016

Assignment 10: Ordered and Subsingleton Logic

Jon Sterling

Due Tuesday, December 5, 2017, 1:30pm

This assignment is due at the beginning of class on the above date and must be submitted electronically via autolab. Submit your homework as a pdf file. **After submitting via autolab, please check the submission's contents to ensure it contains what you expect. No points can be given to a submission that isn't there.**

1 Ordered Logic

In ordered natural deduction, we had the following elimination rule for the *under* connective:

$$\frac{A \quad A \setminus B}{B} \setminus E$$

In the ordered sequent calculus, however, we had the following left rule:

$$\frac{\Omega \Longrightarrow A \quad \Omega_L B \quad \Omega_R \Longrightarrow C}{\Omega_L \Omega (A \setminus B) \Omega_R \Longrightarrow C} \setminus L$$

We could also have tried another version of the left rule, which we will call $\setminus L'$, which more closely mirrored the natural deduction elimination form:

$$\frac{\Omega_L B \quad \Omega_R \Longrightarrow C}{\Omega_L A (A \setminus B) \Omega_R \Longrightarrow C} \setminus L'$$

Task 1 (10 pts). Assuming *cut*, show that these two versions of the left rule are equivalent. That is, show that $\setminus L'$ is a derived rule for the sequent calculus which has just $\setminus L$, and show that $\setminus L$ is a derived rule for the sequent calculus which has just $\setminus L'$.

Task 2 (10 pts). As is the case with many seemingly sensible rules, the new left rule $\setminus L'$ will actually destroy the logical character of the sequent calculus. In

particular, the version of sequent calculus with $\backslash L'$ and not $\backslash L$ does not enjoy the admissibility of cut, and thence does not validate the cut elimination theorem.

Demonstrate a counterexample to cut elimination in the calculus with $\backslash L'$; to be precise, this should be a sequent $\Gamma \Longrightarrow C$ for some specific Γ and specific C which can be proved with *cut* but has no cut-free proof.

2 Subsingleton Logic

Consider the encoding of binary numbers in ordered inference where the proposition $b0$ represents a 0 bit, $b1$ represents a 1 bit, and $\$$ represents the the end of a binary string; in this encoding, we represent the binary number 1011 with the ordered state $\$ b1 b0 b1 b1$.

Task 3 (10 pts). Use *ordered inference rules* to encode a procedure that decides whether a binary string contains an even or odd number of 1-bits in it.

Specifically, introducing a new atomic proposition *par* together with rules such that when \vec{A} is the encoding of a binary string in standard form, one can derive $\frac{\vec{A} \text{ par}}{\$ b0}$ iff the string \vec{A} has an even number of 1-bits, and one can derive

$\frac{\vec{A} \text{ par}}{\$ b1}$ iff \vec{A} has an odd number of 1-bits.

You may freely introduce any auxiliary propositions and rules that you require.

Task 4 (10 pts). Rewrite your solution to Task 3 in the form of a well-typed ordered concurrent program in the subsingleton fragment. You may freely make any definitions you require, including recursive definitions of session types and concurrent programs.

Your solution should include the definition of a type of bit strings *bits*, together with a program *par* that has the type $bits \vdash \text{par} : bits$.