**Constructive Logic (15-317), Spring 2022**
**Recitation 14: Linearly lining up for food (4-20-2022)**

One day, you visit a bakery you heard about from a good friend. Inside is an amazing selection of baked goods at all kinds of price ranges. Upon taking a look around, you just know that you would eat everything in the shop if only you could afford it. After taking a step back and pondering the sad state of your wallet, you begin to browse the inventory to figure out what you can buy with your limited pastry budget of five dollars.

Going forward, let's represent each dollar as the proposition $D$ and represent each unit of your boundless appetite as the proposition $A$.

**Task 1.** Suppose that there is some proposition, $P$, you want to prove using just your money and appetite as assumptions. What do judgements of this form look like in our sequent calculus-like notation? Remember, $D$ is a *linear* assumption while $A$ is *unrestricted*.

**Solution 1:** $D\ res, D\ res, D\ res, D\ res, D\ res, !A\ res \Vdash P\ true$

## Let us eat cake

You stumble upon the last remaining specialty cupcake of the day. After staring at it for far too long, you realize you must buy it. Once it is in your hands, you motion to take a bite. But wait! If you eat it now, will you be able to take it home and enjoy it later? Let's find out!

Let's represent the two outcomes as the propositions $S_{\text{take cake home}}$ and $S_{\text{eat cake now}}$ and represent the cupcake as $C$.

**Task 2.** Given $C \multimap S_{\text{take cake home}}$ and $C \multimap S_{\text{eat cake now}}$ as unrestricted assumptions, prove that you cannot generate both $S_{\text{take cake home}}$ and $S_{\text{eat cake now}}$ simultaneously.

**Solution 2:** $C \multimap S_{\text{take cake home}}, C \multimap S_{\text{eat cake now}}; C \Vdash S_{\text{take cake home}} \otimes S_{\text{eat cake now}}$

When the proof branches on $\otimes R$, the last cupcake $C$ can only go to one branch, since it is a linear resource. Then, when applying $\multimap L$, we can generate either $S_{\text{eat cake now}}$ or $S_{\text{take cake home}}$ but not both.

**Task 3.** Given the same assumptions, prove that the two outcomes are alternatives, represented as an alternative conjunction.

**Solution 3:**

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma; C \Vdash C}\ init \quad \dfrac{}{\Gamma; S_{\text{take cake home}} \Vdash S_{\text{take cake home}}}\ init}{\Gamma; C, C \multimap S_{\text{take cake home}} \Vdash S_{\text{take cake home}}}\ {\multimap}L}{\Gamma; C \Vdash S_{\text{take cake home}}}\ copy \quad \dfrac{\dfrac{\dfrac{}{\Gamma; C \Vdash C}\ init \quad \dfrac{}{\Gamma; S_{\text{eat cake now}} \Vdash S_{\text{eat cake now}}}\ init}{\Gamma; C, C \multimap S_{\text{eat cake now}} \Vdash S_{\text{eat cake now}}}\ {\multimap}L}{\Gamma; C \Vdash S_{\text{eat cake now}}}\ copy}{\Gamma; C \Vdash S_{\text{take cake home}} \& S_{\text{eat cake now}}}\ \&R$$

Where $\Gamma = \{C \multimap S_{\text{take cake home}}, C \multimap S_{\text{eat cake now}}\}$.

**Task 4.** Given this alternative conjunction as a linear assumption, prove one of the outcomes.

**Solution 4:**

$$\dfrac{\dfrac{}{\cdot; S_{\text{take cake home}} \Vdash S_{\text{take cake home}}}\ init}{\cdot; S_{\text{take cake home}} \& S_{\text{eat cake now}} \Vdash S_{\text{take cake home}}}\ \&L$$

## Near the end

While on the long and linear bus ride home, you realize you didn't spend as much money as you thought you would. Not finding the day's events to be enough practice and regretting the fact that you didn't buy more, you begin to think about the following problems:

**Task 5.** Prove $\Vdash A \multimap B \multimap A$.

**Solution 5:** This sequent is not provable:

$$\dfrac{\dfrac{\text{fail}}{\cdot; A\ \mathit{res}, B\ \mathit{res} \Vdash A\ \mathit{true}}}{\cdot;\cdot \Vdash A \multimap B \multimap A\ \mathit{true}} \multimap R \times 2$$

**Task 6.** Prove $\Vdash A \& \top \multimap A$ and $\Vdash A \multimap A \& \top$.

**Solution 6:**

$$\dfrac{\dfrac{\dfrac{}{\cdot; A\ \mathit{res} \Vdash A\ \mathit{true}}\ \mathit{init}}{\cdot; A \& \top\ \mathit{res} \Vdash A\ \mathit{true}}\ \& L_1}{\cdot;\cdot \Vdash A \& \top \multimap A\ \mathit{true}} \multimap R$$

$$\dfrac{\dfrac{\dfrac{}{\cdot; A\ \mathit{res} \Vdash A\ \mathit{true}}\ \mathit{init} \qquad \dfrac{}{\cdot; A\ \mathit{res} \Vdash \top\ \mathit{true}}\ \top R}{\cdot; A\ \mathit{res} \Vdash A \& \top\ \mathit{true}}\ \& R}{\cdot;\cdot \Vdash A \multimap A \& \top\ \mathit{true}} \multimap R$$

**Task 7.** Prove $\Vdash A \otimes \mathbf{1} \multimap A$ and $\Vdash A \multimap A \otimes \mathbf{1}$.

**Solution 7:**

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot; A\ \mathit{res} \Vdash A\ \mathit{true}}\ \mathit{init}}{\cdot; A\ \mathit{res}, \mathbf{1}\ \mathit{res} \Vdash A\ \mathit{true}}\ \mathbf{1}L}{\cdot; A \otimes \mathbf{1}\ \mathit{res} \Vdash A\ \mathit{true}}\ \otimes L}{\cdot;\cdot \Vdash A \otimes \mathbf{1} \multimap A\ \mathit{true}} \multimap R$$

$$\dfrac{\dfrac{\dfrac{}{\cdot; A\ \mathit{res} \Vdash A\ \mathit{true}}\ \mathit{init} \qquad \dfrac{}{\cdot;\cdot \Vdash \mathbf{1}\ \mathit{true}}\ \mathbf{1}R}{\cdot; A\ \mathit{res} \Vdash A \otimes \mathbf{1}\ \mathit{true}}\ \otimes R}{\cdot;\cdot \Vdash A \multimap A \otimes \mathbf{1}\ \mathit{true}} \multimap R$$

## Turing Machines in Ordered Logic

Consider the encoding of turing machine in ordered inference where the proposition $q_i$ is the current state of the machine, $a_j$ is the contents in a cell on the tape and \$ represents the two endmarkers of the tape. We imagine that the contents of the current cell is that tothe right of $q$.

We can represent the initial context for the initial state $q_0$ as

$$\$ \; q_0 \; a_0 \; \ldots \; a_n \; \$$$

where $a_0 \ldots a_n$ is the input word on the tape. We allows denote the proposition ␣ to be a blank symbol.

If the current state of the machine is

$$\$ \; a_0 \; q \; a_1 \; \$$$

and if the transition function specifies to write symbol $a_1'$, transition to state $q'$, and move right, then the resulting configuration would be

$$\$ \; a_0 \; a_1' \; q' \; \$$$

. If the transition function instead specified to move left instead of right, then the resulting configuration would be

$$\$ \; q \; a_0 \; a_1' \; \$$$

.

**Task 8.** Use ordered inference rules to encode a procedure that will allow the machine to move left, move right, extend the tape when we reach the boundary, and eliminate $q_f$ when we reach the final state.

**Solution 8:** http://www.cs.cmu.edu/ crary/317-f18/lectures/21-ordered.pdf