Lecture Notes on Substructural Deduction

15-317: Constructive Logic
Jonathan Sterling*

Lecture 20
November 13, 2017

In philosophy, the *subjective logic* is the study of valid inference: the correct deduction of conclusions from premises. We transcribe this activity using horizontal lines, where premises lie above the line and conclusions lie below the line:

$$\frac{\mathcal{J}_0 \quad \cdots \quad \mathcal{J}_n}{\mathcal{K}}\ R$$

The meaning of such an inference is that we can deduce $\mathcal{K}$ from $\mathcal{J}_0$ through $\mathcal{J}_n$. This is a schema for the deduction of *mathematical* knowledge, which remains true in permanence as soon as it has become true. Therefore, deductions of this kind may freely ignore, reuse or reorder known facts; this kind of inference is called *structural*, and notions of inference which are not closed under these operations are called *substructural*.

1 Linear Inference

What we have seen above is not, however, the full picture, which is captured instead by the notion of inference qua *local transformation* of states. In generality, such a view requires us to admit multiple conclusions, leading to the following schema for inference:

$$\frac{\mathcal{J}_0 \quad \cdots \quad \mathcal{J}_m}{\mathcal{K}_0 \quad \cdots \quad \mathcal{K}_n}\ R$$

*These notes are substantially influenced by Frank Pfenning's notes on Deductive Inference from 15-816 (2016).

The meaning of the above inference is that the transition R can take place when our state includes $\mathcal{J}_0$ through $\mathcal{J}_m$, and takes us to a new state which *instead* has $\mathcal{K}_0$ through $\mathcal{K}_n$. In this transition, the facts $\mathcal{J}_i$ have been consumed and replaced.

Under such a paradigm, a rule of inference expresses an atomic transition from one state to another; the premises capture a fragment of the state prior to the transition, and the conclusions capture the effect of that transition on the original state.

Ephemerality and persistence The above explanation captures the more general situation of knowledge which is not *persistent* but instead *ephemeral* (subject to loss or change). We call inference that involves ephemeral judgments *linear*.

The ephemeral version of deduction which we described above can be used to describe everyday situations, where the subjects of inference are not unchanging realities but are instead contingent and subject to change over time.

While the status of a statement like $\forall a, b, c : \mathbb{R}.\ a^2 + b^2 = c^2$ is permanent or persistent from the moment of its deduction, consider the statement *"China is on the socialist road"*. This assertion may be correct at one time and incorrect at another; it began to be true in 1949, but it ceased to be true in 1976. ***Truth is ephemeral.***

1.1 A Single Spark Can Start A Prairie Fire

Consider the old Chinese saying, *"A single spark can start a prairie fire."* We can capture this idea using linear inference using the following kinds of fact: spark, $\mathsf{on\_fire}(p)$ and $\mathsf{prairie}(p)$.

$$\frac{\mathsf{prairie}(p) \quad \mathsf{spark}}{\mathsf{prairie}(p) \quad \mathsf{on\_fire}(p)}$$

We can refine this a bit more by noticing that some p being a prairie is a *persistent* kind of fact: once true, it always remains true, and we can duplicate that fact as-needed. We will use underlines to indicate persistent facts in our states; we are also justified in omitting a persistent fact in the conclusion, since it is implicitly present.

$$\frac{\underline{\mathsf{prairie}(p)} \quad \mathsf{spark}}{\mathsf{on\_fire}(p)}$$

Choosing substructural (linear, ordered) deduction is not a restriction so much as an ability. ***Substructural logic generalizes structural logic.***

1.2 Commodity Exchange, the Universal Equivalent and the Money-Form of Value

In political economy, the *Value form* is initially comprised of two parts: the *relative form of value* and the *equivalent form of value* [MMF04]. These two moments of value can be observed when two commodities meet in the marketplace for exchange; for instance:

$$\frac{\textbf{20 yards of linen}}{\textbf{one coat}}$$

If a quantity of coats is fixed, then the relative form of value is the 20 yards of linen, and the equivalent form is the single coat; when the exchange is read in the other direction (and the quantity of linen is fixed), the relative form is the single coat and the equivalent form is the 20 yards of linen.

Conceivably one can come up with relative and equivalent values for all commodities, leading to a vast network of relationships between commodities; for instance, if one can establish an exchange rate between coats and pomagranates, then one can compose this with the one between linen and coats, establishing a certain quantity of pomagranates as the equivalent form of value to the relative form of value in linen.

However, the development of an advanced capitalist economy is essentially precluded by this design, because of the fact that the values of each commodity depend on the values of every other commodity that they can (transitively) be exchanged with. There are two main problems:

1. Whether one can actually exchange commodity A with commodity B is contingent on whether there is a chain of exchange relations that connects A and B; this is not guaranteed.

2. The units of exchange between two commodities may establish a non-whole-number relative value at a certain equivalent value. This can lead to unfair exchanges.

The answer to these problem is the development of a ***universal equivalent*** against which the relative form of value of all commodities is measured, i.e. a *classifying object* in the space of commodities. This universal equivalent is called **money**. When we have a universal equivalent, the simplest way to exchange two commodities is to use money as an intermediate form.

$$\begin{array}{c}\textsf{pretzel}\\ \hline\hline \textsf{quarter}\end{array} \qquad \begin{array}{c}\textsf{bagel}\\ \hline\hline \textsf{dollar}\quad\textsf{quarter}\end{array} \qquad \begin{array}{c}\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\\ \hline\hline \textsf{dollar}\end{array}$$

Now, suppose I have seven pretzels, and I want to convert these into bagels. Let us bring the pretzels to the market:

$$\begin{array}{c}
\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\\ \hline
\textsf{quarter}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\\ \hline
\textsf{quarter}\quad\textsf{quarter}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\\ \hline
\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\\ \hline
\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{pretzel}\quad\textsf{pretzel}\quad\textsf{pretzel}\\ \hline
\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{pretzel}\quad\textsf{pretzel}\\ \hline
\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{pretzel}\\ \hline
\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\\ \hline
\textsf{dollar}\quad\textsf{quarter}\quad\textsf{quarter}\quad\textsf{quarter}\\ \hline
\textsf{bagel}\quad\textsf{quarter}\quad\textsf{quarter}
\end{array}$$

Using the rules of inference, we first converted all of our pretzels into the universal equivalent; then, we converted this into as many bagels as possible. At the end, we received 50¢ in change from the bagel vendor, because our pretzels did not have the relative form of value of a whole-number equivalent quantity of bagel.

2 Ordered Inference

So far we have uncovered a notion of inference which describes the local transition of states, which are comprised of unordered collections of facts. We can refine this further, by allowing these facts to have a *location* relative to one another, and not generally allowing two facts to switch places. This opens up an entirely new realm of processes which can be captured through inference.

2.1 Example: A Stack Calculator

Ordered inference can be used to encode what is called *substructural operational semantics*, in which transitions for state machines are encoded as rules of inference [PS09]; the ordered character of facts naturally gives rise to a notion of control stack.

Consider the following following grammar of arithmetic expressions and stack frames:

$$
\begin{array}{lcl}
E & ::= & \bar{n} \mid \mathsf{plus}(E, E) \mid \mathsf{minus}(E, E) \\
K & ::= & \mathsf{plus}(\Box, E) \mid \mathsf{plus}(\bar{n}, \Box) \mid \mathsf{minus}(\Box, E) \mid \mathsf{minus}(\bar{n}, \Box)
\end{array}
$$

Now we define three kinds of fact or task:

1. $\mathsf{calc}(E)$ means "calculate E"
2. $\mathsf{ret}(\bar{n})$ means "return the numeral $\bar{n}$"
3. $\mathsf{cont}(K)$ means "resume calculation K when a value has been returned"

We now provide rules which, if executed in *ordered* inference, will derive $\mathsf{ret}(\bar{n})$ for some $\bar{n}$ from $\mathsf{calc}(E)$ for any E.

$$
\frac{\mathsf{calc}(\bar{n})}{\mathsf{ret}(\bar{n})}
\qquad
\frac{\mathsf{calc}(\mathsf{plus}(E_1, E_2))}{\mathsf{calc}(E_1) \quad \mathsf{cont}(\mathsf{plus}(\Box, E_2))}
\qquad
\frac{\mathsf{calc}(\mathsf{minus}(E_1, E_2))}{\mathsf{calc}(E_1) \quad \mathsf{cont}(\mathsf{minus}(\Box, E_2))}
$$

$$
\frac{\mathsf{ret}(\bar{n}) \quad \mathsf{cont}(\mathsf{plus}(\Box, E))}{\mathsf{calc}(E) \quad \mathsf{cont}(\mathsf{plus}(\bar{n}, \Box))}
\qquad
\frac{\mathsf{ret}(\bar{n}) \quad \mathsf{cont}(\mathsf{plus}(\bar{m}, \Box))}{\mathsf{ret}(\overline{n+m})}
$$

$$
\frac{\mathsf{ret}(\bar{n}) \quad \mathsf{cont}(\mathsf{minus}(\Box, E))}{\mathsf{calc}(E) \quad \mathsf{cont}(\mathsf{minus}(\bar{n}, \Box))}
\qquad
\frac{\mathsf{ret}(\bar{n}) \quad \mathsf{cont}(\mathsf{minus}(\bar{m}, \Box))}{\mathsf{ret}(\overline{n-m})}
$$

Observe that the *ordered* character of inference is crucial here. If we were allowed to reorder facts, the procedure would be non-deterministic and would in most cases not return the correct result: consider what would happen if we swapped two different stack frames!

Exercise 1 *In order to encode our control stack, we have expoited the fact that facts cannot be exchanged in ordered inference. Suppose we were working in* linear inference, *where facts can be exchanged; can you still find a way to encode the stack calculator?*

2.2 Lambek Calculus

The first application of ordered deduction was to capture the surface structure of natural language expressions in a compositional way, accounting for all the oddities of word order, gapping, and scrambling which pervade human languages. Initiated by Lambek [Lam58], this tradition has given birth to an explosion of new calculi and formalisms for ordered logic during the past half century, both in the style of lambda calculus [PP98, PP, MVF11, Mor12] and in combinatory style [Ste96, Bal02].

Lambek calculus begins by specifying a collection "atomic syntactic types" (base types) which represent the primitive parts of speech. Here are some basic ones:[1]

TYPE	MEANING	EXAMPLE
np	noun phrase	"people"
dp	determiner phrase	"the people", "she"
vp	verb phrase	"she serves the people"

Then, we add two connectives B / A and $A \backslash B$ (called "over" and "under" respectively) which are subject to the following rules of ordered inference:

$$\frac{M : B \,/\, A \quad N : A}{M\ N : B}\ /\mathsf{E} \qquad\qquad \frac{N : A \quad M : A \,\backslash\, B}{N\ M : B}\ \backslash\mathsf{E}$$

From the elimination rules above, these seem to be some kind of implication, which differ only in whether the argument should occur to the left or to the right. We use these connectives to generate the types of words, and then the rules of the logic specify the ways that words can be combined.

For instance, transitive verbs have type $(\mathsf{dp} \,\backslash\, \mathsf{vp}) \,/\, \mathsf{dp}$; that is, they are terms which take (first) a determiner phrase on the right (the object) and then another determiner phrase on the left (the subject), and then behave as a verb phrase (rougly a sentence). Adjectives have type $\mathsf{np} \,/\, \mathsf{np}$: if you place one before a noun phrase, it becomes a noun phrase. Determiners have type $\mathsf{dp} \,/\, \mathsf{np}$: if you place a determiner before a noun phrase, it becomes a determiner phrase.

Conjunctions like "and" can be assigned the type $A \,\backslash\, (A \,/\, A)$ for any syntactic type A.

[1]In modern linguistics, there are many more syntactic types! But these will allow us to work through some basic examples. Note also that in these notes we diverge slightly from the atomic types chosen by Lambek, preferring a presentation more aligned with the modern understanding of natural language syntax.

2.2.1 Parsing

With these connectives in place, ordered inference gives an operational semantics to parsing problems. To set up a parsing problem, the initial state of facts is given as a sequence of words together with their syntactic type; then, one tries to use the rules of Lambek calculus to derive a single proposition, vp. Here is an example:

$$\dfrac{\dfrac{\dfrac{\dfrac{\textit{Women} : \mathsf{dp} \quad \textit{hold up} : (\mathsf{dp} \backslash \mathsf{vp}) / \mathsf{dp} \quad \textit{half} : \mathsf{dp} / \mathsf{dp} \quad \textit{the} : \mathsf{dp} / \mathsf{np} \quad \textit{sky} : \mathsf{np}}{\textit{Women} : \mathsf{dp} \quad \textit{hold up} : (\mathsf{dp} \backslash \mathsf{vp}) / \mathsf{dp} \quad \textit{half} : \mathsf{dp} / \mathsf{dp} \quad \textit{the sky} : \mathsf{dp}}\ {/}\mathsf{E}}{\textit{Women} : \mathsf{dp} \quad \textit{hold up} : (\mathsf{dp} \backslash \mathsf{vp}) / \mathsf{dp} \quad \textit{half the sky} : \mathsf{dp}}\ {/}\mathsf{E}}{\textit{Women} : \mathsf{dp} \quad \textit{hold up half the sky} : \mathsf{dp} \backslash \mathsf{vp}}\ {/}\mathsf{E}}{\textit{Women hold up half the sky} : \mathsf{vp}}\ {\backslash}\mathsf{E}$$

Remark 1 *The reader may be uncomfortable with the fact that we have started with* $\textit{hold up} : (\mathsf{dp} \backslash \mathsf{vp}) / \mathsf{dp}$ *as a single lexeme, rather than as a compound phrase formed using the verb* $\textit{hold}$ *and the verb modifier* $\textit{up}$*. Unfortunately, it is not yet possible for us to account for this kind of gapping construction, which requires the machinery developed in [MVF11].*

Remark 2 *Why is* $\textit{women} : \mathsf{dp}$ *a determiner phrase rather than a noun phrase? Technically, it is really a plural noun phrase which is adjoined to a silent plural determiner; we will learn to account for phenomena analogous to this in §2.2.3.*

2.2.2 Derived Rules and Type Raising

For the sake of space, fix the following abbreviations:

$$\mathsf{tv} \triangleq (\mathsf{dp} \backslash \mathsf{vp}) / \mathsf{dp} \qquad \text{(transitive verb)}$$

$$\mathsf{iv} \triangleq \mathsf{dp} \backslash \mathsf{vp} \qquad \text{(intransitive verb)}$$

$$\mathsf{d} \triangleq \mathsf{dp} / \mathsf{np} \qquad \text{(determiner)}$$

$$\mathsf{conj}[A] \triangleq A \backslash (A / A) \qquad \text{(conjunction)}$$

Try to parse the sentence, *"Lenin opposes and Kerensky supports the War."*, assuming the following initial state, writing $?$ for an unconstrained syntactic type:

$$\textit{Lenin} : \mathsf{dp} \quad \textit{opposes} : \mathsf{tv} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports} : \mathsf{tv} \quad \textit{the} : \mathsf{d} \quad \textit{war} : \mathsf{np}$$

You will quickly find that it cannot be done! Certain parts of the sentence can be parsed, but no matter what we try, we will get stuck. For instance:

$$\dfrac{\dfrac{\dfrac{\textit{Lenin} : \mathsf{dp} \quad \textit{opposes} : \mathsf{tv} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports} : \mathsf{tv} \quad \textit{the} : \mathsf{d} \quad \textit{war} : \mathsf{np}}{\textit{Lenin} : \mathsf{dp} \quad \textit{opposes} : \mathsf{tv} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports} : \mathsf{tv} \quad \textit{the war} : \mathsf{dp}}\,/\mathsf{E}}{\textit{Lenin} : \mathsf{dp} \quad \textit{opposes} : \mathsf{tv} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports the war} : \mathsf{iv}}\,/\mathsf{E}}{\textit{Lenin} : \mathsf{dp} \quad \textit{opposes} : \mathsf{tv} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky supports the war} : \mathsf{vp}}\,\backslash\mathsf{E}$$

And that is as far as we can possibly get. To get further, we need to introduce *type raising*. First, let's add introduction rules for the *over* and *under* connectives:

$$\dfrac{\begin{array}{c}\overline{\cdots u : A} \\ \vdots \\ M : B\end{array}}{\lambda u.M : B \,/\, A}\,/\mathsf{I}^u \qquad\qquad \dfrac{\begin{array}{c}\overline{u : A \cdots} \\ \vdots \\ M : B\end{array}}{\lambda u.M : A \setminus B}\,\backslash\mathsf{I}^u$$

In the above rules, we use ellipses to indicate that the hypothesis must be the rightmost or the leftmost assumption respectively. Now, for any syntactic types A, X we can derive the following *type raising* rule:

$$\dfrac{M : A}{M^{\uparrow} : X \,/\, (A \setminus X)}\,\mathsf{raise} \triangleq \dfrac{\dfrac{M : A \quad \overline{v : A \setminus X}}{v\ M : X}\,\backslash\mathsf{E}}{\lambda v.v\ M : X \,/\, (A \setminus X)}\,/\mathsf{I}^v$$

Another important derived rule is *composition*:

$$\dfrac{M : A \,/\, B \quad N : B \,/\, C}{M; N : A \,/\, C}\,\mathsf{cmp} \triangleq \dfrac{\dfrac{M : A \,/\, B \quad \dfrac{N : B \,/\, C \quad \overline{u : C}}{N\ v : B}\,/\mathsf{E}}{M\ (N\ v) : A}\,/\mathsf{E}}{\lambda v.M\ (N\ v) : A \,/\, C}\,/\mathsf{I}^v$$

Now we are equipped to try our derivation again.

$$
\begin{array}{cl}
\textit{Lenin} : \mathsf{dp} \quad \textit{opposes} : \mathsf{tv} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports} : \mathsf{tv} \quad \textit{the} : \mathsf{d} \quad \textit{war} : \mathsf{np} & \\
\hline
\textit{Lenin} : \mathsf{dp} \quad \textit{opposes} : \mathsf{tv} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports} : \mathsf{tv} \quad \textit{the war} : \mathsf{dp} & /\mathsf{E} \\
\hline
\textit{Lenin}^{\uparrow} : \mathsf{vp} \,/\, (\mathsf{dp} \setminus \mathsf{vp}) \quad \textit{opposes} : (\mathsf{dp} \setminus \mathsf{vp}) \,/\, \mathsf{dp} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports} : \mathsf{tv} \quad \textit{the war} : \mathsf{dp} & \mathsf{raise} \\
\hline
\textit{Lenin}^{\uparrow}; \textit{opposes} : \mathsf{vp} \,/\, \mathsf{dp} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky} : \mathsf{dp} \quad \textit{supports} : \mathsf{tv} \quad \textit{the war} : \mathsf{dp} & \mathsf{cmp} \\
\hline
\textit{Lenin}^{\uparrow}; \textit{opposes} : \mathsf{vp} \,/\, \mathsf{dp} \quad \textit{and} : \mathsf{conj}[?] \quad \textit{Kerensky}^{\uparrow} : \mathsf{vp} \,/\, (\mathsf{dp} \setminus \mathsf{vp}) \quad \textit{supports} : \mathsf{tv} \quad \textit{the war} : \mathsf{dp} & \mathsf{raise} \\
\hline
\textit{Lenin}^{\uparrow}; \textit{opposes} : \mathsf{vp} \,/\, \mathsf{dp} \quad \textit{and} : (\mathsf{vp} \,/\, \mathsf{dp}) \setminus ((\mathsf{vp} \,/\, \mathsf{dp}) \,/\, (\mathsf{vp} \,/\, \mathsf{dp})) \quad \textit{Kerensky}^{\uparrow}; \textit{supports} : \mathsf{vp} \,/\, \mathsf{dp} \quad \textit{the war} : \mathsf{dp} & \mathsf{cmp} \\
\hline
\textit{Lenin}^{\uparrow}; \textit{opposes and} : (\mathsf{vp} \,/\, \mathsf{dp}) \,/\, (\mathsf{vp} \,/\, \mathsf{dp}) \quad \textit{Kerensky}^{\uparrow}; \textit{supports} : \mathsf{vp} \,/\, \mathsf{dp} \quad \textit{the war} : \mathsf{dp} & \setminus\mathsf{E} \\
\hline
\textit{Lenin}^{\uparrow}; \textit{opposes and Kerensky}^{\uparrow}; \textit{supports} : \mathsf{vp} \,/\, \mathsf{dp} \quad \textit{the war} : \mathsf{dp} & /\mathsf{E} \\
\hline
\textit{Lenin}^{\uparrow}; \textit{opposes and Kerensky}^{\uparrow}; \textit{supports the war} : \mathsf{vp} &
\end{array}
$$

Exercise 2 *Try to derive the each of following rules, or conjecture that it is impossible.*

$$
\frac{A}{X \,/\, (X \,/\, A)} \qquad\qquad \frac{A}{(X \,/\, A) \setminus X} \qquad\qquad \frac{A}{(A \setminus X) \setminus X}
$$

2.2.3 Using Persistent Propositions

Consider adding a new atomic syntactic type $\mathsf{np}_{\mathit{mass}}$ for *mass nouns*. Among other things, mass nouns differ from ordinary nouns in that their determiner is silent (they do not need to use "the"): that is, this determiner appears in the parse tree, but it does not appear in the surface syntax.

This seems to poses a problem for parsing: we will not have the determiner in our state initially, since we don't know in advance where we will need it. The solution is to regard this determiner as *persistent*, and simply add $\underline{\varnothing : \mathsf{dp} \,/\, \mathsf{np}_{\mathit{mass}}}$ to all our states. Then, whenever we need a mass noun determiner, we can freely add it in the appropriate spot.

Consider the sentence, "*We demand peace and bread and land!*", where all three demands are mass nouns. To parse this, we want to derive a rule of the following shape:

$$
\frac{\underline{\varnothing : \mathsf{dp} \,/\, \mathsf{np}_{\mathit{mass}}} \quad \textit{we} : \mathsf{dp} \quad \textit{demand} : \mathsf{tv} \quad \textit{peace} : \mathsf{np}_{\mathit{mass}} \quad \textit{and} : \mathsf{conj}[\mathsf{np}_{\mathit{mass}}] \quad \textit{bread} : \mathsf{np}_{\mathit{mass}} \quad \textit{and} : \mathsf{conj}[\mathsf{np}_{\mathit{mass}}] \quad \textit{land} : \mathsf{np}_{\mathit{mass}}}{\underline{\varnothing : \mathsf{dp} \,/\, \mathsf{np}_{\mathit{mass}}} \quad ??? : \mathsf{vp}}
$$

Stop and try to derive this before turning the page.

Solution:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand} : \mathsf{tv} \quad \mathit{peace} : \mathsf{np}_{mass} \quad \mathit{and} : \mathsf{conj}[\mathsf{np}_{mass}] \quad \mathit{bread} : \mathsf{np}_{mass} \quad \mathit{and} : \mathsf{conj}[\mathsf{np}_{mass}] \quad \mathit{land} : \mathsf{np}_{mass}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand} : \mathsf{tv} \quad \mathit{peace} : \mathsf{np}_{mass} \quad \mathit{and} : \mathsf{conj}[\mathsf{np}_{mass}] \quad \mathit{bread\ and} : \mathsf{np}_{mass} / \mathsf{np}_{mass} \quad \mathit{land} : \mathsf{np}_{mass}}\,\backslash\mathsf{E}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand} : \mathsf{tv} \quad \mathit{peace} : \mathsf{np}_{mass} \quad \mathit{and} : \mathsf{conj}[\mathsf{np}_{mass}] \quad \mathit{bread\ and\ land} : \mathsf{np}_{mass}}\,/\mathsf{E}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand} : \mathsf{tv} \quad \mathit{peace\ and} : \mathsf{np}_{mass} / \mathsf{np}_{mass} \quad \mathit{bread\ and\ land} : \mathsf{np}_{mass}}\,\backslash\mathsf{E}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand} : \mathsf{tv} \quad \mathit{peace\ and\ bread\ and\ land} : \mathsf{np}_{mass}}\,/\mathsf{E}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand} : \mathsf{tv} \quad \varnothing : \mathsf{dp} / \mathsf{np}_{mass} \quad \mathit{peace\ and\ bread\ and\ land} : \mathsf{np}_{mass}}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand} : \mathsf{tv} \quad \varnothing\ \mathit{peace\ and\ bread\ and\ land} : \mathsf{dp}}\,/\mathsf{E}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we} : \mathsf{dp} \quad \mathit{demand}\ \varnothing\ \mathit{peace\ and\ bread\ and\ land} : \mathsf{iv}}\,/\mathsf{E}}{\underline{\varnothing : \mathsf{dp} / \mathsf{np}_{mass}} \quad \mathit{we\ demand}\ \varnothing\ \mathit{peace\ and\ bread\ and\ land} : \mathsf{vp}}\,\backslash\mathsf{E}
$$

References

[Bal02] Jason Baldridge. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, 2002.

[Lam58] Joachim Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958.

[MMF04] K. Marx, E. Mandel, and B. Fowkes. *Capital: A Critique of Political Economy*. Number Volume 1 in Penguin classics. Penguin Books Limited, 2004.

[Mor12] G.V. Morrill. *Type Logical Grammar: Categorial Logic of Signs*. Springer Netherlands, 2012.

[MVF11] Glyn Morrill, Oriol Valentín, and Mario Fadda. The displacement calculus. *Journal of Logic, Language and Information*, 20(1):1–48, Jan 2011.

[Pfe04] Frank Pfenning. Substructural operational semantics and linear destination-passing style. In W.-N. Chin, editor, *Proceedings of the 2nd Asian Symposium on Programming Languages and Systems (APLAS'04)*, page 196, Taipei, Taiwan, November 2004. Springer-Verlag LNCS 3302. Abstract of invited talk.

[PP] Jeff Polakow and Frank Pfenning. Natural deduction for intuitionistic non-commutative linear logic. Submitted. Revised and extended version of abstract from TLCA'99.

[PP98] Jeff Polakow and Frank Pfenning. Ordered linear logic programming. Technical Report CMU-CS-98-183, Department of Computer Science, Carnegie Mellon University, December 1998.

[PS09] Frank Pfenning and Robert J. Simmons. Substructural operational semantics as ordered logic programming. In *Proceedings of the 24th Annual Symposium on Logic in Computer Science (LICS 2009)*, pages 101–110, Los Angeles, California, August 2009. IEEE Computer Society Press.

[Ste96] Mark Steedman. *Surface structure and interpretation*. Linguistic inquiry monographs, 30. MIT Press, 1996.