Logic of Programs	**Lecture 5**
CS 6860 Fall 2015	**Tues. Sept 8, 2015**

Lecture 5

Minimal Propositional Calculus Continued- Seeking Computational Content

The rule formats and suggestive rule names do not by themselves reveal *computational content*. We see suggestive examples of what it might be, but nothing definite. For example, we saw in Lecture 4 that the proof of $B \Rightarrow (A \Rightarrow B)$ has the form $\lambda(x.\lambda(y.x))$. If we take B to be the specific type $(D \Rightarrow D)$, we know that the proof object for $(D \Rightarrow D) \Rightarrow (A \Rightarrow (D \Rightarrow D))$ is $\lambda(x.\lambda(y.x))$. We know that we can supply a proof object for $(D \Rightarrow D)$, namely $\lambda(v.v)$. The rule name for this application is

$$\mathrm{ap}(\lambda(x.\lambda y.x); \lambda(v.v)),$$

and we know intuitively that the proof object for $A \Rightarrow (D \Rightarrow D)$ is $\lambda(y.\lambda(v.v))$. This suggests a "computation rule" for "application" $\mathrm{ap}(f; a)$. Namely, if f is $\lambda(x.\lambda(y.x))$ then $\mathrm{ap}(f; a)$ should result if $\lambda(y.a)$. This accords well with an evidence semantics and with operations on proof terms that *normalize* or simplify the proof structure.

One of the earliest understandings that there is computational content in proofs came from these *reduction operations* on proofs, ways of normalizing them (see Prawitz *Natural Deduction*, Dover, NY, 2006. Reprint of 1965 classic booklet). A particularly strong connection came in terms of a rule called *cut* which allows using "Lemmas" or "short cuts" in a proof. This rule is also called a *sequencing rule*.

$$\begin{array}{rll} x\!:\!H & \vdash G & \text{by } \mathrm{seq}(y.\_\_;\_\_) \\ x\!:\!H,\, y\!:\!C & \vdash G & \text{by } g(x,y) \\ x\!:\!H & \vdash C & \text{by } c(x) \end{array}$$

Simplify the proof term using $g(x, y)$, that is, reduce $\mathrm{seq}(y.g(x, y); c(x))$ to $g(x, c(x))$.

To understand (know) the *sense* or conceptual content of a proposition is to know what information counts as *evidence*.

If the proposition, say P, is not vacuous, then we can know some specific example of content or evidence. (False is vacuous, there is no evidence for it.)

3+2=5 in $\mathbb{N}$	Call $\mathbb{N}$ a *type*, 2,3,5 are members.
0=0 in $\mathbb{N}$	
0=1 in $\mathbb{N}$	We know there is no evidence.

Abstract objects – Euclidean Geometry as an example.

We want to talk about points as in the Euclidean Plane. We could agree on a notation.

> a, b in Points where $a \neq b$ in Point. We can imagine "virtual" evidence that points are distcint
>
> ab is a Segment.
>
> ab, bc are Segments with a common point. The evidence in this case is easy to "see." The common point is b.

In daily life: Today is Tuesday September 8, 2015.

> The moon is made of cheese.
>
> There is life on mars.
>
> Pluto is a planet; defined as such in ... and the definition has not changed.
>
> $\sqrt{2}$ is irrational.
>
> π is transcendental.

Evidence Semantics

We want to abstract from the notion of proof, in whatever format (Natural Deduction, Hilbert Style, Sequent Calculus, Refinement Logic (sequents done top-down from the goal)) and discuss a possibly broader notion of what it means *to have evidence that causes us to know that the conceptual content of a proposition is realized or made known by this evidence.*

Suppose for the propositional variables P_1, P_2, P_3, ... we understand *assignments of evidence* $\mathcal{E}$: Var $\to$ Evidence. The evidence can be coded as Terms in the type theory. For each *prop variable* P_i we assign a collection or type of Terms, $\mathcal{E}(P_i)$ is a type. If it is empty, then under the interpretation $\mathcal{E}$, P_i is considered false. If $\mathcal{E}(P_i)$ is non-empty, then the elements of $\mathcal{E}(P_i)$ are evidence indicating why P_i is believable, say some $p \in \mathcal{E}(P_i)$.

We are interested in models given by the map:

Val: P_i: Var $\to \mathcal{E}(P_i)$. Val is a function that assigns to propositional variable P_i a type $\mathcal{E}(P_i)$ a type $\mathcal{E}(P_i)$ consisting of evidence.

PC is the Proposition Calculus. It comes in three "flavors."

Minimal PC (mPC) vs. Intuitionistic PC (iPC) vs. Classical PC (PC)

Let $\sim A == (A \Rightarrow \bot)$ for a designated constant $\bot$.

mPC: $\sim A \vee \sim B \Rightarrow \sim (A \,\&\, B)$

$$\begin{array}{rl} \vdash ((A \Rightarrow\bot) \vee (B \Rightarrow\bot)) \Rightarrow (A \,\&\, B) \Rightarrow\bot & \lambda(d.\lambda(ab.\text{spread}(ab;\, a, b.\text{decide}(d; na.\text{ap}(na; a); nb.\text{ap}(nb; b))))) \\ d\!:\!(A \Rightarrow\bot) \vee (B \Rightarrow\bot),\, ab\!:\!A \,\&\, B \quad \vdash \bot & \\ a\!:\!A,\, b\!:\!B \quad \vdash \bot & \text{decide}(d; \_\_) \\ na\!:\!A \Rightarrow\bot \quad \vdash \bot & \text{ap}(na; a) \\ nb\!:\!B \Rightarrow\bot \quad \vdash \bot & \text{ap}(nb; b) \end{array}$$

Here is an argument that uses *ex falso quodlibet* and requires it. The rule is

$$H,\ f\!:\!\text{False},\ H' \vdash G \text{ by any}(f).$$

$$\begin{array}{rll}
\vdash A \vee B \Rightarrow \sim A \Rightarrow B & \text{by } \lambda(d.\lambda(na.\text{decide}(d; a.\text{any}(ap(na; a); b.b)))) \\
d\!:\!A \vee B,\ na\!:\!(A \Rightarrow \text{False}) & \vdash B & \text{by decide} \\
a\!:\!A,\ na\!:\!(A \Rightarrow \text{False}) & \vdash B & \text{by ap}(na; a) \\
 & \vdash A & \text{by } a \\
v\ :\ \text{False} & \vdash b & \text{by any}(v) \\
b\!:\!B & \vdash B & \text{by } b
\end{array}$$

Note $(\sim A \Rightarrow B) \Rightarrow A \vee B$ requires the law of excluded middle on A, i.e. $A \vee \sim A$. We will discuss this later.

Here are some other propositions of mPC

- $A \,\&\, B \Rightarrow \sim (A \,\&\, \sim B)$
- $(A \Rightarrow \sim B) \Rightarrow (B \Rightarrow\!\sim A)$
- $\sim\!\sim (A \,\&\, B) \Leftrightarrow \sim\!\sim A \,\&\, \sim\!\sim B$
- $\sim\!\sim A \,\&\, \sim\!\sim B \Rightarrow \sim\!\sim (A \,\&\, B)$
- $\sim\!\sim (A \,\&\, B) \Rightarrow \sim\!\sim A \,\&\, \sim\!\sim B$
- $\sim\!\sim (A \Rightarrow B) \Rightarrow (\sim\!\sim A \Rightarrow \sim\!\sim B)$

Note, in classical logic we can use $\sim\!\sim A \Leftrightarrow A$ to simplify these considerably.

In due course we will examine a very important result of Harvey Friedman that mPC can express iPC, i.e. we can embed iPC in mPC. We think there is a new "semantics" proof of this important result. We will sketch this in later a later class.