SKETCHY NOTES FOR WEEK 2 OF BASIC LOGIC

The point of unique readability is to enable definition by recursion. The idea of a recursive definition is this: we will specify a function G on $C = \mathcal{C}(B, K)$ by specifying what G does on B, and specifying how to compute $G(F(a_0, \dots a_{k-1}))$ from $a_0, \dots a_{k-1}$ and $G(a_0), \dots G(a_{k-1})$. Unique readability will ensure that this definition makes sense by avoiding situations where some element of C might get assigned more than one value.

Before proving a general theorem about recursive definitions, we analyse uniquely readable structures. Recall that in general $\mathcal{C}(B, K) = \bigcup_n C_n$, where $C_0 = B$ and $C_{n+1} = C_n \bigcup \{F(\vec{a}) : F \in K, \vec{a} \in C_n\}$.

Note on notation: I am using "$\vec{a}$" as shorthand for an argument tuple $(a_0, \dots a_{k-1})$ and "$\vec{a} \in C_n$" as shorthand for the assertion that all entries a_i are in C_n.

Now suppose that (B, K) is a uniquely readable inductive structure for C and that $c \in C_{n+1} \setminus C_n$. By the definition of unique readability and the fact that $c \in C \setminus B$, there must be a unique $F \in K$ and a unique tuple $\vec{a} \in C$ such that $c = F(\vec{a})$. By the definition of C_{n+1} there must be $F' \in K$ and $\vec{a}' \in C_n$ with $c = F'(a')$, and the uniqueness tells us that $F' = F$ and $\vec{a} = \vec{a}$. Summarising: if $c \in C_{n+1} \setminus C_n$ then it's generated in exactly one way, by applying a function in K to a tuple with entries in C_n.

One more note: we will be recursively defining a function G from C to some fixed "codomain" D. The point of fixing the codomain in advance is to make sure that the recursive definition does not break down by getting into a situation where we don't know how to compute $G(F(a_0, \dots a_{k-1}))$ from $a_0, \dots a_{k-1}$ and $G(a_0), \dots G(a_{k-1})$.

Theorem 1. *Let (B, K) be a uniquely readable inductive structure for C, and let D be some set. Let $G_0 : B \to D$ and for each $F \in K$ let $H_F : C^k \times D^k \to D$ where k is the arity of F. Then there is a unique function $G : C \to D$ such that*

- *For $c \in B$, $G(c) = G_0(c)$.*
- *For $c \in C$, $G(c) = H_F(a_0, \dots a_{k-1}, G(a_0), \dots G(a_{k-1}))$ for the unique $F \in K$ and $(a_0, \dots a_{k_1}) \in C^k \cap \mathrm{dom}(F)$ with $c = F(a_0, \dots a_{k-1})$.*

Proof. The proof that there is at most one G with these properties is a straightforward induction on the inductive structure. If G and G' are two such functions then the induction hypothesis is that $G(c) = G'(c)$. The base case holds because $G(c) = G_0(c) = G'(c)$ for all $c \in B$, and the induction step holds because if $G(a_i) = G'(a_i)$ for all i then

$$G(F(a_0, \dots a_{k-1})) = H_F(a_0, \dots a_{k-1}, G(a_0), \dots G(a_{k-1})) =$$

$$H_F(a_0, \dots a_{k-1}, G'(a_0), \dots G'(a_{k-1})) = G'(F(a_0, \dots a_{k-1})).$$

To show that there exists at least one G, we construct a suitable function in stages, where at stage n we define the function on C_n. We will define functions $G_n : C_n \to D$ by recursion on n, making sure that $G_{n+1} \upharpoonright C_n = G_n$.

Base: G_0 is the given function $G_0 : C \to D$.

Recursion step: Suppose we defined $G_n : C_n \to D$. We define $G_{n+1}(c) = G_n(c)$ for $c \in C_n$. For $c \in C_{n+1} \setminus C_n$ we define $G_{n+1}(c) = H_F(a_0, \dots a_{k-1}, G_n(a_0), \dots G_n(a_{k-1}))$.

for the unique function $F \in K$ and tuple $(a_0, \ldots a_{k-1})$ such that $c = F(a_0, \ldots, a_{k-1})$: this makes sense because (as we observed already) the tuple $(a_0, \ldots a_{k-1})$ consists of elements of C_n, so that the values $G_n(a_i)$ were all defined already.

Finally we define G by setting $G(c) = G_n(c)$ for the least n such that $c \in C_n$. Since we made sure that $G_{n+1} \restriction C_n = G_n$, $G_m \restriction C_n = G_n$ for all $m \geq n$, and so easily $G(c) = G_n(c)$ for *every* n such that $c \in C_n$.

Now we verify that this G works. The proof is an easy induction on n, where the induction hypothesis states that $G(c)$ satisfies the defining properties for all $c \in C_n$.

Base case: $n = 0$. When $c \in B = C_0$, we defined $G(c) = G_0(c)$.

Successor step: Suppose that $G(c)$ is as required for all $c \in C_n$. Let $c \in C_{n+1}$. If $c \in C_n$ then there is nothing to do. If $c \in C_{n+1} \setminus C_n$ then $c = F(a_0, \ldots a_{k-1})$ for $F \in K$ and $a_i \in C_n$, and we have

$$G(c) = G_{n+1}(c) = H_F(a_0, \ldots a_{k-1}, G_n(a_0), \ldots G_n(a_{k-1})) =$$
$$H_F(a_0, \ldots a_{k-1}, G(a_0), \ldots G(a_{k-1}))$$

$\square$

Remark: We worded the statement of the Theorem to make the recursive character of the definition rather explicit. An equivalent formulation (closer to what we use in practice) says:

Given G_0 and H_F for $F \in K$ as above, there is a unique $G : C \to D$ satisfying the equations:

- $G_0 = G \restriction B$.
- $G(F(a_0, \ldots a_{k-1})) = H_F(a_0, \ldots a_{k-1}, G(a_0), \ldots G(a_{k-1}))$ for all $F \in K$ and all $(a_0, \ldots a_{k-1}) \in \mathrm{dom}(F) \cap C^k$.

Now for a blizzard of definitions.

We fix a set $\{T, F\}$ of *truth values* and define various operations on the set of truth values.

- $\neg F = T$, $\neg T = F$.
- $T \vee T = T \vee F = F \vee T = T$, $F \vee F = F$.
- $F \wedge F = F \wedge T = T \wedge F = F$, $T \wedge T = T$.
- $T \to T = F \to T = F \to F = T$, $T \to F = F$.

It is important to understand that these are finitary functions, and are not the same as the corresponding symbols in the alphabet for propositional logic: the symbols *denote* the functions, in rather the same way that "James Cummings" *is* a string of 14 ascii symbols but *denotes* a math professor.

Definition 1. *A* truth assignment *is a function f from $\{A_n : n \in \mathbb{N}\}$ to $\{T, F\}$.*

To each truth assignment f we will associate a function F from the set of wff's to $\{T, F\}$ by the following recursive definition:

(1) $F(A_n) = f(A_n)$.
(2) $F((\alpha \wedge \beta)) = F(\alpha) \wedge F(\beta)$.
(3) $F((\alpha \vee \beta)) = F(\alpha) \vee F(\beta)$.
(4) $F((\alpha \to \beta)) = F(\alpha) \to F(\beta)$.
(5) $F((\neg\alpha)) = \neg F(\alpha)$.

Note: In class we used f for the original truth assignment and F for the induced function from wff's to truth values to underline the point that they are distinct functions. In practice we usually just use the same symbol for both of them, and (unlike in class) we have followed this convention below.

Definition 2. (1) *A* theory *is a set of wff's.*

(2) *If f is a truth assignment and α is a wff then f* satisfies *α ($f \models \alpha$) if and only if $f(\alpha) = T$.*
(3) *If S is a theory and f is a truth assignment, then f* satisfies *S ($f \models S$) if and only if $f \models \alpha$ for all $\alpha \in S$.*
(4) *If S is a theory and α is a wff, then S* entails *α ($S \models \alpha$) if and only if for every truth assignment f, if $f \models S$ then $f \models \alpha$.*
(5) *If S and S' are theories then S* entails *S' ($S \models S'$) if and only if $S \models \alpha$ for all $\alpha \in S'$.*
(6) *A wff α is a* syllogism *if and only if $f \models \alpha$ for every truth assignment f.*

Example: $A_0 \vee \neg A_0$ is a syllogism, $\{A_0, A_0 \to A_1\}$ entails A_1.

Note: In class we started a discussion of proofs, but since it was informal and incomplete I will leave that to next week's notes.