SKETCHY NOTES FOR WEEK 4 OF BASIC LOGIC

We now leave the subject of propositional logic and turn to first-order logic, which will be our main object of study for the rest of the term. As a motivation for first-order logic, consider the following inference: given the hypotheses "Socrates is human" and "Everyone who is human is mortal", we conclude "Socrates is mortal". It is not so simple to formalise this argument in propositional logic, because to capture the sense of "Everyone who is human is mortal" we need to write down many implications of the general form "If Fred is human then Fred is mortal".

We would like to have a logical system in which we can discuss properties like being human or being mortal, and in which we have an economical way of saying for example that every object has some property. To start with we need to be a bit more precise about properties.

Definition 1. *Let A be a set. Then an n-*ary predicate on *A (or n-*ary relation on *A) is a set $R \subseteq A^n$.*

We are identifying the relation with the set of n-tuples for which it holds. If R is a relation we say that "$R(a_1, \dots a_n)$ is true" when $(a_1, \dots a_n) \in R$ and "$R(a_1, \dots a_n)$ is false" when $(a_1, \dots a_n) \notin R$.

Definition 2. *Let A be a set. Then an n-*ary function on *A is a function from A^n to A.*

In first order logic we typically start by fixing a *signature* (or *vocabulary*), which is a set of symbols that will be used in building up the formulae. Different signatures may be appropriate for different tasks: for we want to discuss number theory then a typical signature might contain symbols intended to denote the numbers zero and one, the operations of addition and multiplication, and the relations "less than" and equality.

Definition 3. *A* signature *is a triple $(\mathcal{C}, \mathcal{R}, \mathcal{F})$ of sets of symbols, together with a function* arity *from $\mathcal{R} \cup \mathcal{F}$ to $\mathbb{N}^+$.*

Given a signature V we will define sets of *terms and formulae*, which will be finite strings of symbols from the following list.

(1) The *connectives* $\neg$, $\vee$, $\wedge$, $\rightarrow$.
(2) The *variable symbols* $\{x_i : i \in \mathbb{N}\}$.
(3) The *quantifier symbols* $\forall$ and $\exists$.
(4) The set $\mathcal{C}$ of *constant symbols.*
(5) The set $\mathcal{R}$ of *relation symbols.*
(6) The set $\mathcal{F}$ of *function symbols.*
(7) The *punctuation marks*, that is to say the comma and the left and right brackets.

By convention we assume that these sets of symbols are *disjoint*, so that there is no possibility of confusion between different types of symbol.

In the setting of *first order logic with equality* we add a distinguished binary relation symbol $\equiv$ to the set of relation symbols from the signature.

We start by defining the *terms*. These are strings which will (eventually, when we get to semantics) be used to stand for objects; they are not themselves formulae, but are building blocks for formulae.

Definition 4. *The set of terms is defined by the following inductive structure:*

(1) *Every constant symbol c and variable symbol x is a term.*
(2) *If f is a function symbol with $arity(f) = m$ and $t_1, \dots t_m$ are terms then $f(t_1, \dots t_m)$ is a term.*

A term is said to be closed *if it contains no variable symbols.*

Now that we have defined the terms we can define the *atomic formulae*. These will be the basic objects in the inductive structure which defines formulae.

Definition 5. *An* atomic formula *is a string of form $R(t_1, \dots t_m)$ where R is a relation symbol, $arity(R) = m$ and $t_1, \dots t_m$ are terms.*

We are now in a position to define the set of formulae. The definition starts with the atomic formulae and builds up new formulae by using either the connectives (exactly as we used them in propositional logic) or the variables and quantifiers (this is new).

Definition 6. *The set of formulae is defined by the following inductive structure:*

(1) *If ϕ is an atomic formula then ϕ is a formula.*
(2) *If ϕ and ψ are formulae then $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$ $(\phi \to \psi)$ are formulae.*
(3) *If ϕ is a formula and x is a variable symbol then $(\exists x\ \phi)$ and $(\forall x\ \phi)$ are formulae.*

Next we describe how the language can be interpreted. Any non-trivial language has many different interpretations.

Definition 7. *Given a signature, a* structure *$\mathcal{M}$ for that signature is given by the following data:*

(1) *A nonempty set M (usually referred to as the* underlying set *of the structure $\mathcal{M}$).*
(2) *For each constant symbol c, an element $c^{\mathcal{M}}$ of the set M.*
(3) *For each function symbol f, a function $f^{\mathcal{M}}$ from $M^{arity(f)}$ to M.*
(4) *For each relation symbol R, an $arity(R)$-ary relation $R^{\mathcal{M}} \subseteq M^{arity(R)}$.*

Note: In the setting of first order logic with equality, we demand that the special symbol $\equiv$ is interpreted by the equality relation.

Ultimately we aim to define a relation "$\mathcal{M} \models \phi$" (ϕ is true in $\mathcal{M}$) between structures and (certain) formulae. If $\mathcal{M}$ is a structure for a signature then we define the *expanded language for $\mathcal{M}$* by adding a constant c_a for each $a \in M$. We use the shorthand $\mathcal{M}$-term for the cumbrous "term of the expanded language for $\mathcal{M}$", and so on for other concepts.

We start by defining how to interpret the closed $\mathcal{M}$-terms. Notice that these terms are generated by an inductive structure which starts off with the elements of $\mathcal{C} \cup \{c_a : a \in M\}$, and then generates more complex terms by applying function symbols.

Definition 8. *We define $\tau^{\mathcal{M}}$ for τ a closed $\mathcal{M}$-term by induction.*

- *If $\tau = c$ for some constant symbol $c \in \mathcal{C}$, $\tau^{\mathcal{M}} = c^{\mathcal{M}}$.*

- *If $\tau = c_a$ for $a \in M$, $\tau^{\mathcal{M}} = a$.*
- *If $\tau = f(t_1, \ldots t_m)$ for some function symbol f and closed $\mathcal{M}$-terms $t_1, \ldots t_m$, then $\tau^{\mathcal{M}} = f^{\mathcal{M}}(t_1^{\mathcal{M}}, \ldots t_m^{\mathcal{M}})$.*

We continue defining the satisfaction relation. We need to make precise the idea of *substituting* a term for a variable in a term or a formula.

Given a term τ and variable x, we define an operation on terms "substitute τ for x". Informally this means replacing every appearance of x by τ.

Definition 9. *Given terms μ and τ and a variable symbol x, we define $\mu[x/\tau]$ by recursion:*

(1) *If μ is a constant symbol or variable symbol other than x, $\mu[x/\tau] = \mu$.*
(2) *If μ is x, then $\mu[x/\tau] = \tau$.*
(3) *If μ is $f(t_1, \ldots t_p)$ for a function symbol f and terms $t_1, \ldots t_p$ then $\mu[x/\tau] = f(t_1[x/\tau], \ldots t_p[x/\tau])$.*

A routine induction shows that $\mu[x/\tau]$ is a term.

In general it matters in what order we do substitutions.

$$x_1[x_1/x_2][x_2/x_3] = x_3, \;\; x_1[x_2/x_3][x_1/x_2] = x_2.$$

Now we define what it means to substitute a term x for a variable τ in a formula. We can't just replace every appearance of x by τ, because this does not always give a formula and also it is not semantically sensible.

Definition 10. *Given a formula ϕ, a term τ and a variable symbol x, we define $\phi[x/\tau]$ by recursion:*

(1) *If ϕ is an atomic formula $R(t_1, \ldots t_p)$, then $\phi[x/\tau] = R(t_1[x/\tau], \ldots t_p[x/\tau])$.*
(2) *If $\phi = (\psi_0 @ \psi_1)$ for formulae ψ_i and a connective $@ \in \{\wedge, \vee, \rightarrow\}$ then $\phi[x/\tau] = (\psi_0[x/\tau] @ \psi_1[x/\tau])$.*
(3) *If $\phi = (\neg\psi)$ for a formula ψ, then $\phi[x/\tau] = (\neg\psi[x/\tau])$.*
(4) *(The trickiest case!) If $\phi = (Qy\psi)$ for a quantifier $Q \in \{\forall, \exists\}$, then $\phi[x/\tau] = \phi$ if y is x and $\phi[x/\tau] = (Qy\psi[x/\tau])$ if y is not x.*

The intuition for the last clause: Appearances of y in ψ are "bound" by quantifying over y, and not availaable for substitution. A variable appearance is called "free" if it is not bound by a quantifier. A variable may make both free and bound apearances in a single formula.

Definition 11. *We define a function Free by recursion, first on terms and then on formulae. Free computes the set of variables appearing freely in a term or formula.*

Terms: Free is given by the recursion

(1) *$Free(c) = \emptyset$ for constant symbols c.*
(2) *$Free(x) = \{x\}$.*
(3) *$Free(f(t_1, \ldots t_p)) = Free(t_1) \cup \ldots \cup Free(t_p)$.*

Formulae: Free is given by the recursion

(1) *$Free(R(t_1, \ldots t_p)) = Free(t_1) \cup \ldots \cup Free(t_p)$ for atomic formulae.*
(2) *$Free((\phi @ \psi)) = Free(\phi) \cup Free(\psi)$, where $@$ is a binary connective.*
(3) *$Free((\neg\phi)) = Free(\phi)$.*
(4) *$Free((Qx\phi)) = Free(\phi) \setminus \{x\}$ where Q is a quantifier.*

Now we can define the class of formulae which have well-defined truth values.

Definition 12. *A formula ϕ is a* sentence *if and only if $Free(\phi) = \emptyset$.*

We are finally ready to define satisfaction. The definition is by recursion (with a twist!) on sentences in the expanded language.

Definition 13. *Let V be a vocabulary and let $\mathcal{M}$ be a structure for V. Then we define a relation $\mathcal{M} \models \phi$ for sentences ϕ of the expanded language.*

$\phi = R(t_1, \dots t_p)$ *where R is a relation symbol. Since ϕ is a sentence all the t_i are closed $\mathcal{M}$-terms. $\mathcal{M} \models \phi$ if and only if $(t_1^{\mathcal{M}}, \dots t_p^{\mathcal{M}}) \in R^{\mathcal{M}}$.*

$\phi = (\phi_1 \wedge \phi_2)$. *$\mathcal{M} \models \phi$ if and only if $\mathcal{M} \models \phi_1$ and $\mathcal{M} \models \phi_2$.*

$\phi = (\phi_1 \vee \phi_2)$. *$\mathcal{M} \models \phi$ if and only if $\mathcal{M} \models \phi_1$ or $\mathcal{M} \models \phi_2$.*

$\phi = (\phi_1 \rightarrow \phi_2)$. *$\mathcal{M} \models \phi$ if and only if whenever $\mathcal{M} \models \phi_1$, then $\mathcal{M} \models \phi_2$.*

$\phi = (\neg\psi)$. *$\mathcal{M} \models \phi$ if and only if $\mathcal{M} \not\models \psi$.*

$\phi = (\exists x\psi)$. *$\mathcal{M} \models \phi$ if and only if there exists $a \in M$ such that $\mathcal{M} \models \psi[x/c_a]$.*

$\phi = (\forall x\psi)$. *$\mathcal{M} \models \phi$ if and only if for every $a \in M$, $\mathcal{M} \models \psi[x/c_a]$.*

This definition is not quite covered by the Recursion theorem we proved earlier in the term. The problem is that even though the system which generates formulae is uniquely readable, we have defined the relation for $(Qx\psi)$ (Q a quantifier) in terms of the relation for various formulae $\psi[x/c_a]$ rather than the formula ψ. How can we prove it is a valid recursive definition?