# CM20019 – Complementary Course Notes

Sheet 3

November 2, 2006

*This sheet provides: 1) alternative presentations of some definitions you find in Richardson's course notes, 2) some extra material on topics that are treated in the course notes (Skolemization)and 2) some more notions and topics that you might not find in the course notes (Herbrand interpretations, and all what is related to that, for example).*

*For the syntax of first order logic, please refer to the course notes.*

## 1 Semantics of Predicate Logic

We will define the meaning of first order logic formulas. As in propositional logic formulas, it can be either true or false, i.e., first order logic is a two-valued logic. However, before being able to specify the meaning of formulas, the meaning of terms like $a$ or $f(a, b)$ needs to be defined.

**1.1 Definition.** An *algebra (or pre-interpretation)* $J$ for a first order language $\mathcal{L}$ based on the alphabet $\mathcal{A}$ consists of a non-empty set $D$, called the domain of $J$, and an assignment of a mapping $f^J : D^n \to D$ to each $n$-ary function symbol $f$ in the alphabet. This implies in particular that nullary function symbols, i.e., constants, are mapped to elements of the domain. Furthermore, non-nullary function symbols are mapped onto functions over the domain.

We can now define the meaning of a ground term $t$:

**1.2 Definition.** The meaning of a ground term $t$ under a pre-interpretation $J$ is:

$$t^J = \begin{cases} t^J & \text{if } t \text{ is a constant,} \\ f^J(t_1^J, \ldots, t_n^J) & \text{if } t \text{ is of the form } f(t_1, \ldots, t_n). \end{cases}$$

**1.3 Example.** Let $t = f(h(c), g(a, b))$ be a term. Its meaning depends on the interpretation. For example, let the domain $D$ be the set $\mathbf{N}$ of natural numbers together with the functions $+/2$, $-/2$ and $\sqrt{.}/1$ and the following assignment:

$$\begin{aligned} a^J &= 5, \\ b^J &= 3, \\ c^J &= 16, \\ f^J &= +, \\ g^J &= -, \\ h^J &= \sqrt{.} \end{aligned}$$

Then, $t^J = +(h(c)^J, g(a, b)^J) = +(\sqrt{c^J}, -(a^J, b^J)) = +(\sqrt{16}, -(5, 3)) = 6$.

Now, let the domain $D$ be a set of objects like *pens*, *pens with cap*, *boxes*, *lids*, *boxes with attached lids*, *boxes with pens*, etc. and consider the following assignment:

$$\begin{aligned} a^J &= lid, \\ b^J &= box, \\ c^J &= pen, \\ f^J &= put - into, \\ g^J &= connect - to, \\ h^J &= attach - cap. \end{aligned}$$

Then, $t^J = put - into(attach - cap(pen), connect - to(lid, box))$, which is part of an instruction to assemble and fill a certain box. The two examples reveal that terms may denote very different objects depending on the chosen interpretation.

We define then the meaning of non-ground terms. For this, we need the notion of *evaluation* (also called *state*, or *assignment*, and which resembles the notion of substitution.

**1.4 Remark.** In the following, given an alphabet $\mathcal{A}$ supporting a language $\mathcal{L}$, we will indicate the set of variables in the alphabet by $\mathcal{A}_V$, the set of function symbols by $\mathcal{A}_F$ and the set of predicate symbols by $\mathcal{A}_R$.

**1.5 Definition.** An *evaluation* over the domain $D$ of a pre-interpretation $J$ is a mapping $\sigma : \mathcal{A}_V \to D$, which assigns an element of $D$ to each variable occurring in $\mathcal{A}_V$ (that is the set of variables in the given language). As notation for evaluations, we will use the same as for substitutions.

**1.6 Definition.** Given an interpretation $J$ and an evaluation $\sigma$, the meaning given by $J$ and $\sigma$ to a term $t$ is defined as follows:

$$t^{J,\sigma} = \begin{cases} \sigma(t) & \text{if } t \text{ is a variable,} \\ f^J(t_1^{J,\sigma}, \ldots, t_n^{J,\sigma}) & \text{if } t \text{ is of the form } f(t_1, \ldots, t_n). \end{cases}$$

Hence, pre-interpretations together with an evaluation map non-ground terms onto elements of $D$.

**1.7 Definition.** Let $e \in D$, $X \in \mathcal{A}_V$ and $\sigma$ be an evaluation. Then $\sigma[X := e]$ denotes the evaluation obtained from $\sigma$ by assigning $e$ to $X$ while leaving all other bindings unchanged, i.e.,

$$\sigma[X := e](Y) = \begin{cases} e & \text{if } X = Y \\ \sigma(Y) & \text{if } X \neq Y \end{cases}$$

We now extend the concept of an interpretation for propositional logic to first order formulas.

**1.8 Definition.** An interpretation $I$ for a first order language $\mathcal{L}$ based on the alphabet $\mathcal{A}$ consists of a pre-interpretation $J$ with a domain $D$ and an assignment of a set $p^I \subseteq D^n$ to each relation symbol $p/n$ belonging to the set of relation symbols $\mathcal{A}_R$. $p^I$ is said to be the *extension* of $p/n$ under $I$. $I$ is said to be based on $J$ and, for uniformity reasons, the mapping $f^J$ is denoted by $f^I$. $D$ is said to be the domain of $I$ as well.

**1.9 Example.** Consider a first order language with the binary predicate symbols $p/2$ and $q/2$. The interpretation $I$ with domain $\mathbf{N}$ and assignments

$$p^I = \{(n,m)|n,m \in \mathbf{N}, n < m\} \quad q^I = \{(n,m)|n,m \in \mathbf{N}, n > m\}$$

maps $p/2$ onto the "less than" and $q/2$ onto the "greater than" relation on natural numbers.

**1.10 Definition.** Given an interpretation $I$ and an evaluation $\sigma$ for a language $\mathcal{L}$, we define for all formulas F the relation $I, \sigma \models F$ (in words: evaluation $\sigma$ satisfies $F$ in $I$) as follows:
$I, \sigma \models p(t_1, \ldots, t_n)$ iff $(t_1^{I,\sigma}, \ldots, t_n^{I,\sigma}) \in p^I$,
$I, \sigma \models \neg F$ iff $I, \sigma \not\models F$,
$I, \sigma \models F_1 \wedge F_2$ iff $I, \sigma \models F_1$ and $I, \sigma \models F_2$,
$I, \sigma \models F_1 \vee F_2$ iff $I, \sigma \models F_1$ or $I, \sigma \models F_2$,
$I, \sigma \models F_1 \rightarrow F_2$ iff $I, \sigma \not\models F_1$ or $I, \sigma \models F_2$,
$I, \sigma \models F_1 \leftrightarrow F_2$ iff $I, \sigma \models F_1 \rightarrow F_2$ and $I, \sigma \models F_2 \rightarrow F_1$,
$I, \sigma \models (\exists X)F$ iff there exists $e \in D$ such that $I, \sigma[X := e] \models F$,
$I, \sigma \models (\forall X)F$ iff for all $e \in D$ it holds $I, \sigma[X := e] \models F$.

**1.11 Definition.** An interpretation $I$ is said to be a *model* for $F$, in symbols $I \models F$, iff for all evaluations $\sigma$ it holds $I, \sigma \models F$.

One should observe that if $F$ is closed, i.e., if $F$ is a sentence, then for all interpretations $I$ we find $I \models F$ iff $I, \sigma \models F$, where $\sigma$ is any evaluation. In other words, if $F$ is closed, then we do not have to worry at all about evaluations.

**1.12 Definition.** Let $\mathcal{F}$ be a set of formulae. The interpretation $I$ is said to be a model for $\mathcal{F}$ iff $I$ is a model for each $G \in \mathcal{F}$.

# 2 Normal Forms

The notion of semantic equivalence introduced for predicate logic extend to first order formulas. Below you find further equivalences for first order formulas. If the final four laws are not applicable because $X$ occurs free in $G$ then all free occurrences of $X$ can be replaced with a new variable $Y$ which does not occur in $F$ and $G$ or anywhere else in the context. Such a renaming ($\alpha$-conversion) is *equivalence preserving* and allows the laws to become applicable.

$$\begin{array}{rcll}
F & \equiv & (\forall X)F & \\
\neg(\forall X)F & \equiv & (\exists X)\neg F & \\
\neg(\exists X)F & \equiv & (\forall X)\neg F & \\
(\forall X)F \wedge (\forall X)G & \equiv & (\forall X)(F \wedge G) & \\
(\exists X)F \vee (\exists X)G & \equiv & (\exists X)(F \vee G) & \\
(\forall X)(\forall Y)F & \equiv & (\forall Y)(\forall X)F & \\
(\exists X)(\exists Y)F & \equiv & (\exists Y)(\exists X)F & \\
(\forall X)F \wedge G & \equiv & (\forall X)(F \wedge G) & \text{if } X \text{ is not free in } G \\
(\forall X)F \vee G & \equiv & (\forall X)(F \vee G) & \text{if } X \text{ is not free in } G \\
(\exists X)F \wedge G & \equiv & (\exists X)(F \wedge G) & \text{if } X \text{ is not free in } G \\
(\exists X)F \vee G & \equiv & (\exists X)(F \vee G) & \text{if } X \text{ is not free in } G
\end{array}$$

The *replacement theorem* for propositional logic extends to first order logic (recall the notation used in the Complementaty course note 2):

**2.1 Theorem.** *If the first-order formulas $G$ and $H$ are semantically equivalent, so are $F[G]$ and $F[G/H]$.*

Besides semantic equivalences, which are often called *model-preserving transformations*, there are additional transformations which preserve only the *validity* or the *satisfiability* of formulas. These transformations are called *validity-preserving* and *satisfiability-preserving* respectively.

The so-called *Skolemization* is based on the following observation: A formula of the form

$$(\exists X_1)\dots(\exists X_n)(\forall Y)F$$

is valid iff

$$(\exists X_1)\dots(\exists X_n)(F[Y := f(X_1,\dots,X_n)])$$

is valid, where the so-called Skolem function $f$ must not occur anywhere else in $F$.

Analogously, a formula of the form

$$(\forall X_1)\dots(\forall X_n)(\exists Y)F$$

is unsatisfiable iff

$$(\forall X_1)\dots(\forall X_n)(F[Y := f(X_1,\dots,X_n)])$$

is unsatisfiable.

**2.1 Remark.** The results above have formal proofs (that we will not see) and are known under the name of Skolem Theorems. It is easy to see at least that the model is not preserved in this transformation. After Skolemization, the formula so obtained has some symbols more (all the *newly* introduced Skolem function symbols): they have no meaning in the interpretations of the original formula. In other words, the *language* of the Skolemized formula is bigger than the language of the original formula.

We see an example of application of this transformation:

**2.2 Example.** Skolemize the formula below, preserving unsatisfiability:

$$(\exists X)(\forall Y)(\forall U)(\exists Z)(p(X,U)\vee p(Y,Z)).$$

We need to apply the second transformation, eliminating existential quantifiers. We eliminate them one at the time, scanning them from left to right.

First, to eliminate $(\exists X)$, we see how many *universal* quantifiers are preceding it (are at its left), because this will determine the arity of the Skolem function we are going to introduce for $X$. There are 0, so we introduce a *new* Skolem function of arity zero (a constant), say $a_s/0$, and we rewrite the formula replacing this new term wherever there was an $X$. We get

$$(\forall Y)(\forall U)(\exists Z)(p(a_s,U)\vee p(Y,Z)).$$

Next, we eliminate $(\exists U)$. This is preceded by 2 universal quantifiers, so we will need to introduce a new Skolem function of arity 2, say $f_s/2$. The *term* we are going to use for $Z$ must depend on the variables in the universal quantifiers preceding it: so we will use the term $f_s(Y, U)$. We get:

$$(\forall Y)(\forall U)(p(a_s, U) \vee p(Y, f_s(Y, U))).$$

The formula so obtained is unsatisfiable iff the original one was.

You may repeat the exercise preserving validity, or invent formulae where to practice the algorithm.

Conjunctive and disjunctive normal forms, are again very important. In propositional logic we have seen an algorithm preserving logical equivalence (and in addition to that, the course notes illustrate another algorithm based on the truth table). We can now extend the first algorithm returning conjunctive normal forms presented for propositional logic to the first order case. Please observe that the algorithm will introduce some steps which will not necessarily maintain logical equivalence, so the full picture is different than in propositional logic.

**Algorithm**
`Input:` A first order formula $F$.
`Output:` A first order formula $G$ in conjunctive normal form such that $G$ is unsatisfiable iff $F$ is unsatisfiable.

1. Universally close the formula.
2. Eliminate all equivalence signs using the equivalence law.
3. Eliminate all implication signs using the implication law.
4. Eliminate all negation signs except those in front of atoms using the de Morgan laws, the double negation laws and the first two laws of the list shown above.
5. Apply the final four laws shown of the list shown above to obtain a formula which consists of a sequence of quantifiers and a quantifier-free formula.
6. Eliminate existential quantifiers by Skolemization.
7. Drop the remaining universal quantifiers.
8. Distribute all disjunctions over conjunctions using the second distributive law.

The formula obtained after step 5 is in a so-called *prenex normal form*, consisting of a prenex, i.e., a sequence of quantifiers, and a quantifier-free formula called a matrix . One should observe that the formula obtained after Skolemization contains only universal quantifiers. Hence, these quantifiers are superfluous and may be dropped. As in propositional logic, the clausal form denotes the set of set notations for the formulas in conjunctive normal form.

**2.2 Theorem.** *If $F_1$ is a conjunctive normal form of a first order formula $F$, then $F_1$ is unsatisfiable iff $F$ is unsatisfiable.*

*Proof.* (Sketch of proof) This result follows immediately from the fact that variables which occur freely are interpreted as universally closed variables (step 1), steps 2-5 and 8 are model-preserving, step 6 preserves satisfiability and step 7 is just an abbreviation. □

**2.3 Example.** Purpose of this example is twofold: we want to see how to interpret a formula, and we want to see an application of the algorithm above. Consider the formula $F$:

$$(\forall V)(q(V,a) \to (\exists W)(q(W,a) \wedge (\forall X)(\forall Y)(p(h(f(X,Y)),W) \to p(h(f(g(X),g(Y))),V)).$$

Consider an interpretation $I$ with domain $\mathbf{R}$ and the following assignment:

$$\begin{aligned} a^I &= 0 \in \mathbf{R}, \\ f^I &= - : \mathbf{R} \times \mathbf{R} \to \mathbf{R}, \\ g^I &= g : \mathbf{R} \to \mathbf{R}, \\ h^I &= |.| : \mathbf{R} \to \mathbf{R}, \\ q^I &= \{(n,m) | n,m \in \mathbf{R}, n > m\}, \\ p^I &= \{(n,m) | n,m \in \mathbf{R}, n < m\}. \end{aligned}$$

In other words: $a, f, h$ are respectively mapped to the zero, subtraction, and absolute value in the reals, and $g$ to a generic function $g$ in the reals; moreover, $q, p$ are respectively mapped to the greater than and less than relations in the reals.

The meaning of the formula $F$, under this interpretation is given by rewriting function and relations symbols with the symbols of the intended interpretation, and by replacing variables $V$ and $W$ by $\epsilon$ and $\delta$ respectively. Let's call this $F^I$

$$(\forall \epsilon)(\epsilon > 0 \to (\exists \delta)(\delta > 0 \wedge (\forall X, Y)(|X - Y| < \delta \to |g(X) - g(Y)| < \epsilon))).$$

So, under $I$, the meaning of $F$ is the statement "$g$ is a continuous function in the reals".

Now, let's see how the algorithm works. We can either take as working case $F$ or $F^I$, simply we want to return a conjunctive normal form. Let's work on $F^I$ (because it is easier to see how the transformation works having clear the meaning, i.e. *continuity*).

Eliminating implications yields:

$$(\forall \epsilon)(\neg \epsilon > 0 \vee (\exists \delta)(\delta > 0 \wedge (\forall X, Y)(\neg |X - Y| < \delta \vee |g(X) - g(Y)| < \epsilon))).$$

Moving the quantifiers to the prenex yields:

$$(\forall \epsilon)(\exists \delta)(\forall X, Y)(\neg \epsilon > 0 \vee (\delta > 0 \wedge (\neg |X - Y| < \delta \vee |g(X) - g(Y)| < \epsilon))).$$

Skolemization yields:

$$(\forall \epsilon)(\forall X, Y)(\neg \epsilon > 0 \vee (f_\delta(\epsilon) > 0 \wedge (\neg |X - Y| < f_\delta(\epsilon) \vee |g(X) - g(Y)| < \epsilon))),$$

where $f_\delta(\epsilon)$ is a new Skolem function symbol. Now, drop universal quantifiers:

$$\neg \epsilon > 0 \vee (f_\delta(\epsilon) > 0 \wedge (\neg |X - Y| < f_\delta(\epsilon) \vee |g(X) - g(Y)| < \epsilon)),$$

and distribute disjunction over conjunction, completing the transformation:

$$(\neg \epsilon > 0 \vee (f_\delta(\epsilon) > 0) \quad \wedge \quad (\neg \epsilon > 0 \vee \neg |X - Y| < f_\delta(\epsilon) \vee |g(X) - g(Y)| < \epsilon))).$$

The first two steps and the final step are model-preserving. The third step is satisfiability-preserving and the fourth step is simply an abbreviation.

# 3 Logical Entailment

The logical entailment relation $\models$ defined for propositional logic between sets of formulas and formulas extends to first order formulas. In other words, $F \models G$ iff all models of $F$ are also models for $G$. Likewise, the concepts of validity, satisfiability, falsifiability and unsatisfiability can be extended to first order logic.

Finally, the deduction theorem and the theorem relating validity of a formula to satisfiability of its negated version hold for universally closed first order formulas as well.

In contrast to propositional logic, the question of whether a first order formula is a logical consequence of a set of formulas is *undecidable*. This can be formally shown by reducing logical entailment in first order logic to some known undecidable problem like the halting problem for Turing machines, Post's correspondence problem or the question of whether the intersection of the languages generated by two context-free grammars is empty.

But logical entailment in first order logic is semidecidable in that if a formula $G$ is logically entailed by a set $F$ of formulas then there is a procedure which eventually terminates and reports this fact. On the other hand, if $G$ is not entailed by $F$ then this procedure may either terminate and report this fact, or run forever.

According to the definition of logical entailment, if we want to show that $F \models G$ holds, then we have to show that each model for $F$ is also a model for $G$. But, unfortunately, for first order formulas there are so many interpretations and each interpretation is based on so many pre-interpretations that we cannot possibly consider all of them. Luckily, we have the possibility of considering only a single pre-interpretation, the so-called Herbrand pre-interpretation.

# 4 Herbrand Interpretations

**4.1 Definition.** Given an alphabet $\mathcal{A}$ which contains at least one constant and a first order language $\mathcal{L}$ based on $\mathcal{A}$, the *Herbrand universe* $T(\mathcal{A}_F)$ is the set of ground terms which can be built from $\mathcal{A}_F$. The *Herbrand base* $A(\mathcal{A}_R, \mathcal{A}_F)$ is the set of ground atoms which can be built from $\mathcal{A}_R$ an $\mathcal{A}_F$.

If the given alphabet doesn't contain a constant, we can arbitrarily introduce one.

**4.2 Example.** Let $\mathcal{A}_F = \{a/0, f/1\}$ and $\mathcal{A}_R = \{p/2\}$. Then,

$$T(\mathcal{A}_F) = \{a, f(a), f(f(a)), \dots\}$$

and

$$A(\mathcal{A}_R, \mathcal{A}_F) = \{p(a,a), p(f(a),a), p(a,f(a)), \dots\}.$$

**4.3 Definition.** The *Herbrand pre-interpretation* or Herbrand algebra for a first order language $\mathcal{L}$ based on $\mathcal{A}$ consists of the domain $T(\mathcal{A}_F)$ and an assignment which assigns to each $n$-ary function symbol $f$ a function $T(\mathcal{A}_F)^n \to T(\mathcal{A}_F)$ by mapping each $n$-tuple of ground terms $(t_1, \dots, t_n)$ to the ground term $f(t_1, \dots, t_n)$. One should observe that there is only a unique Herbrand pre-interpretation.

**4.4 Definition.** A *Herbrand interpretation* $I$ for a first order language $\mathcal{L}$ based on $\mathcal{A}$ consists of the Herbrand pre-interpretation and an assignment of a set $p^I$ of $n$-tuples to each relation symbol $p/n$ occurring in $\mathcal{A}_R$. In other words, different Herbrand interpretations differ in the set of tuples assigned to the relation symbols. A *Herbrand model* for a set of formulas $F$ is a Herbrand interpretation which is a model for $F$.

There is a one-to-one correspondence between each Herbrand interpretation $I$ and a particular subset of the Herbrand base $A(\mathcal{A}_R, \mathcal{A}_F)$ defined by

$$\{p(t_1, \ldots, t_n) | p \in \mathcal{A}_R \text{ is an } n\text{-ary relation symbol and } (t_1, \ldots, t_n) \in p^I\}.$$

This allows us to identify Herbrand interpretations with subsets of the Herbrand base with the understanding that each element of a subset is mapped to true by the interpretation and, vice versa, if a ground atom is mapped to true than it must occur in that subset.

**4.5 Example.** Consider the following set of formulae:

$$\mathcal{F} = \{p(a, a), (\forall X, Y)(p(X, Y) \rightarrow p(X, f(Y)))\}.$$

The Herbrand universe and Herbrand base have been already calculated in Exercise 4.2. The following is a Herbrand model for $\mathcal{F}$:

$$\{p(a, a), p(a, f(a)), p(a, f(f(a))), \ldots\}.$$

Herbrand proved in 1930 that for each interpretation $J$ over some domain $D$ for a formula $F$ in clausal form there exists a Herbrand interpretation $I$ for $F$ such that if $J$ satisfies $F$, then $I$ satisfies $F$ as well. From this result it is straightforward to prove the following theorem:

**4.1 Theorem.** *A first order formula $F$ in clause form is unsatisfiable iff $F$ is false under all Herbrand-interpretations.*

**4.6 Remark.** This result tells us how logical consequence can be shown. Using the Deduction Theorem logical consequence is mapped onto validity. In order to show the validity of a formula, the formula is negated and the negated formula is shown to be unsatisfiable (see Theorem in the previous complementary course note). In doing so, the negated formula is transformed into clause form. According to Theorem 2.2 this transformation is satisfiability-preserving. Finally we check for unsatisfiability under all Herbrand-interpretations using one of the negative calculi we are going to see next (for example, resolution).