MATHEMATICS 300 — FALL 2009

Introduction to Mathematical Reasoning

H. J. Sussmann

INSTRUCTOR'S NOTES

(September 22, 2009)

1 The precise definition of the notion of "propositional form"

REMARK. *The concept of "propositional form," or "well-formed propositional string" introduced here is that of* **"well-formed propositional string with all parentheses put in"**. *So, for example, the string* $(A \wedge B) \vee C$ *is well-formed, but the string* $A \wedge B \vee C$ *is not. In the book, rules are given for omitting parentheses. According to these rules,* $A \wedge B \vee C$ *is well-formed, because the rules tell you that in this case you can restore parentheses and read you had wanted to say* $3 \cdot (5+7)$*, then you have to put in the parentheses. The book also tells you how to omit parentheses for propositional forms involving* $\Rightarrow$ *and* $\Leftrightarrow$*. You can read the explanation of how this is done in the book, or in the section on "omitting and restoring parentheses" at the end of this handout. But my own recommendation is to leave the parentheses in. This makes for more cumbersome writing, but for easier reading.* $\diamondsuit$

Suppose we are given an *alphabet* $\mathcal{A}$, that is, a set of letters or letter-like symbols that will be called "propositional variables." For example, $\mathcal{A}$ could be $\{A, B, C\}$ (that is, the set consisting of the three letters A, B, C). Or $\mathcal{A}$ could be $\{A, B, C, D, E, F\}$. Or $\mathcal{A}$ could be $\{P, Q, R, S, T, U, V, W\}$. Or $\mathcal{A}$ could be an infinite set, say, the set consisting of the symbols P_1, P_2, P_3, and so on, so that P_n is a member of $\mathcal{A}$ for every natural number n.

Using the alphabet $\mathcal{A}$, we define "propositional strings" as follows: a **propositional string** with variables in $\mathcal{A}$ is a string S of symbols such that each symbol in S is either (a) a member of $\mathcal{A}$, or (b) one of the propositional connectives $\sim$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, or (c) a left parenthesis or a right parenthesis. For example, if $\mathcal{A} = \{A, B, C\}$, then the following are propositional strings:

A	$BAB)))((A\wedge \Rightarrow$
$A \wedge B$	$\sim A$
$AB\wedge$	$\sim A \wedge B$
$(A \wedge B \wedge C) \vee D$	$\sim (A \wedge B)$
$((A \wedge B) \wedge C) \vee D$	$(\sim A) \wedge B$
$\wedge\vee \sim (BA \Rightarrow$	$(\sim A)(\wedge B\,.$

We now want to specify exactly how to distinguish those propositional strings that are "well-formed" (that is, are acceptable representations of sentences) from those that are not well-formed. (For example, the strings

$$\begin{array}{l} A \\ A \wedge B \\ (A \wedge B \wedge C) \vee D \\ ((A \wedge B) \wedge C) \vee D \\ \sim A \\ \sim (A \wedge B) \\ (\sim A) \wedge B \end{array}$$

are well-formed, but the strings

$$\begin{array}{l} AB\wedge \\ \wedge\vee \sim (BA \Rightarrow \\ BAB)))((A\wedge \Rightarrow \\ \sim A \wedge B \\ (\sim A)(\wedge B \end{array}$$

are not. *But see the section on "omitting and restoring parentheses". According to the book's rules for parenthesis restoration, for the string* $\sim A \wedge B$ *you can restore parentheses and get* $(\sim A) \wedge B$*, so* $\sim A \wedge B$ *is well-formed.*)

In order to do this, we have to deal with a difficulty. When, for example, we combine the strings A and $B \wedge C$ by means of the connective $\vee$, we write $A \vee (B \wedge C)$. But when we combine $A \Rightarrow B$ and $B \wedge C$ by means of $\vee$, we write $(A \Rightarrow B) \vee (B \wedge C)$. In other words, when a string S is combined with other strings, then S remains unchanged if it is just a letter, but it is surrounded by parentheses if it is a string such as $A \Rightarrow B$, consisting of more than just one letter.

To make this precise, we introduce the following notation: if S is a propositional string consisting of a single symbol in $\mathcal{A}$, then S^* will stand for S. And if S is any other propositional string, then S^* will stand for (S). For example, if S is A then S^* is A, but if S is $A \wedge B$ then S^* is $(A \wedge B)$.

Now, here are the rules defining "well-formed formula" (wff):

- WFF1. If S consists of a single letter in $\mathcal{A}$, then S is a wff.
- WFF2. If S is a wff then $\sim S^*$ is a wff.
- WFF3. If S and T are wff's then $S^* \Rightarrow T^*$ and $S^* \Leftrightarrow T^*$ are wff's.
- WFF4. If n is a natural number, $n > 1$, and $S_1, S_2, \cdots, S_n$ are wff's, then $S_1^* \wedge S_2^* \wedge \cdots \wedge S_n^*$ and $S_1^* \vee S_2^* \vee \cdots \vee S_n^*$ are wff's.
- WFF5. Only those strings that are obtainable by repeated applications of WFF1, WFF2, WFF3 and WFF4 are wff's.

Example 1. Let $\mathcal{S}$ be the string $\sim (P \wedge Q)$. Let us prove that $\mathcal{S}$ is well-formed.

1. By Rule WFF1, the one-letter strings P, Q, are well-formed. (*At this point, we know that* P *and* Q *are well-formed.*)

2. By Rule WFF4 (with $n = 2$), the string $P \wedge Q$ is well-formed, because P and Q are well-formed. (*At this point, we know that* P, Q, *and* $P \wedge Q$ *are well-formed.*)
3. By Rule WFF2 the string $\sim (P \wedge Q)$ is well-formed, because $P \wedge Q$ is well-formed. (*At this point, we have found out that* $\mathcal{S}$ *is well-formed, so our proof is complete.*)

Example 2. Let $\mathcal{S}$ be the string

$$((P \Rightarrow Q) \wedge (P \Rightarrow R) \wedge (Q \Rightarrow (\sim R))) \Rightarrow (\sim (P \vee Q)) \ .$$

Let us prove that $\mathcal{S}$ is well-formed.

1. By Rule WFF1, the one-letter strings P, Q, R are well-formed. (*At this point, we know that* P, Q, R *are well-formed.*)
2. By Rule WFF3 the strings $P \Rightarrow Q$ and $P \Rightarrow R$ are well-formed, because P, Q and R are well-formed. (*At this point, we know that* P, Q, R, $P \Rightarrow Q$, *and* $P \Rightarrow R$ *are well-formed.*)
3. By Rule WFF4 (with $n = 2$), the string $P \wedge Q$ is well-formed, because P and Q are well-formed. (*At this point, we know that* P, Q, R, $P \Rightarrow Q$, $P \Rightarrow R$, *and* $P \wedge Q$ *are well-formed.*)
4. By Rule WFF2, the string $\sim R$ and the string $\sim (P \wedge Q)$ are well-formed, because R and $P \wedge Q$ are well-formed. (*At this point, we know that* P, Q, R, $P \Rightarrow Q$, $P \Rightarrow R$, $P \wedge Q$, $\sim R$, *and* $\sim (P \wedge Q)$ *are well-formed.*)
5. By Rule WFF3, the string $Q \Rightarrow (\sim R)$ is well-formed, because Q and $\sim R$ are well-formed. (*At this point, we know that* P, Q, R, $P \Rightarrow Q$, $P \Rightarrow R$, $P \wedge Q$, $\sim R$, $\sim (P \wedge Q)$, *and* $Q \Rightarrow (\sim R)$ *are well-formed.*)
6. By Rule WFF4, with $n = 3$, the string

$$(P \Rightarrow Q) \wedge (P \Rightarrow R) \wedge (Q \Rightarrow (\sim R))$$

is well-formed, because $P \Rightarrow Q$, $P \Rightarrow R$, and $Q \Rightarrow (\sim R)$ are well-formed. (*At this point, we know that* P, Q, R, $P \Rightarrow Q$, $P \Rightarrow R$, $P \wedge Q$, $\sim R$, $\sim (P \wedge Q)$, $Q \Rightarrow (\sim R)$, *and* $(P \Rightarrow Q) \wedge (P \Rightarrow R) \wedge (Q \Rightarrow (\sim R))$ *are well-formed.*)
7. By Rule WFF3, the string

$$((P \Rightarrow Q) \wedge (P \Rightarrow R) \wedge (Q \Rightarrow (\sim R))) \Rightarrow (\sim (P \vee Q))$$

is well-formed, because $(P \Rightarrow Q) \wedge (P \Rightarrow R) \wedge (Q \Rightarrow (\sim R))$ and $\sim (P \vee Q)$ are well-formed. (*At this point, we have finally shown that* $\mathcal{S}$ *is well-formed, so our proof is finished*)

Question for you to think about. Let $\mathcal{S}$ be the string

$$((P \Rightarrow Q) \wedge (P \Rightarrow R) \wedge (Q \Rightarrow (\sim R)) \Rightarrow (\sim (P \vee Q)) \ .$$

How would you prove that $\mathcal{S}$ is *not* well-formed? (*This is not easy. You need a clever idea. Think of how the number of left parentheses in a wff compares with the number of right parentheses.*)

1.1 Omitting and restoring parentheses

If you do not want to write too many parentheses, you could define a notion of **well-formed string with omission of parentheses**, as the book does. The idea for doing this is, simply, that you can omit parentheses if the rules for restoring parentheses given in the book will lead you back to the formula you want. To make precise sense of this is not easy. I will try do it in a subsequent handout. In the meanwhile, feel free to write propositional forms the way the book does, but I recommend that you do it my way. For example, for the propositional form that I would write as $(P\wedge Q)\vee R$, you can also write $P\wedge Q\vee R$, because according to the parenthesis restoration rules of the book, this is read as $(P\wedge Q)\vee R$. But if you want to write $P\wedge(Q\vee R)$, then you have to leave it as $P\wedge(Q\vee R)$, because if you omit the parentheses it would be read as $(P\wedge Q)\vee R$, which is not what you want. In general, the order of precedence of the logical connectives is as follows: $\sim$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, and when you have several $\Rightarrow$s, or several $\Leftrightarrow$s, they go from left to right. (When you have several $\wedge$s or $\vee$s, you do not need to put parentheses except at the top level.)

Thus, for example, in the string

$$P\Rightarrow Q\wedge\sim R\wedge\sim(P\Rightarrow Q)\Rightarrow P\Rightarrow R$$

we begin the parenthesis restoration with the $\sim$s, and rewrite our string as

$$P\Rightarrow R\vee Q\wedge(\sim R)\wedge(\sim(P\Rightarrow Q))\Rightarrow P\Rightarrow R\,.$$

Then we move on to the $\wedge$s, and get

$$P\Rightarrow R\vee(Q\wedge(\sim R)\wedge(\sim(P\Rightarrow Q)))\Rightarrow P\Rightarrow R\,.$$

Next we move on the the $\vee$s, and obtain

$$P\Rightarrow(R\vee(Q\wedge(\sim R)\wedge(\sim(P\Rightarrow Q))))\Rightarrow P\Rightarrow R\,.$$

Then we put a pair of parentheses for the leftmost $\Rightarrow$, obtaining

$$(P\Rightarrow(R\vee(Q\wedge(\sim R)\wedge(\sim(P\Rightarrow Q)))))\Rightarrow P\Rightarrow R\,.$$

After that, we put a pair of parentheses for the second leftmost $\Rightarrow$, obtaining

$$((P\Rightarrow(R\vee(Q\wedge(\sim R)\wedge(\sim(P\Rightarrow Q)))))\Rightarrow P)\Rightarrow R\,.$$

And we are done.