These are preliminary notes, containing only the necessary formalities. If I ever get around to it I will add more explanations

13.1 Motivation

Second-Order Propositional Logic ($\mathbf{P^2}$) is a simple propositional theory that provides quantification over a very limited range. It allows us to study some of the issues that arise in logics with quantifiers without having to deal with the full complexity of first-order logic. It's syntax is much simpler because it is a higher-order theory, which allows us to simulate all connectives with just implication and the universal quantifier.

Computationally, Second-Order Propositional Logic (or Quantified Boolean Formulas) is between Propositional Logic and First-Order logic. Satisfiability in Propositional Logic is $\mathcal{NP}$-complete, (satisfiability in) First-Order logic is undecidable. Second-Order Propositional Logic is still decidable, but $PSPACE$-complete.

13.2 Syntax of $\mathbf{P^2}$

In $\mathbf{P^2}$ we only need propositional variables, a constant bot (for *false*), implication $\supset$, and the universal quantifier $\forall$.

In the following exposition we will use the following symbols as meta-variables

$p, q, r, \ldots$	a propositional variable
$A, B, \ldots$	a $\mathbf{P^2}$ formula
$\Gamma, \Delta, \ldots$	a finite set of $\mathbf{P^2}$ formulas

13.2.1 Formulas of $\mathbf{P^2}$

The formulas of $\mathbf{P^2}$ are generated by

$$\begin{array}{rcll} V & = & p_0 \mid p_1 \mid p_2 \mid \cdots\cdots & \text{(a countably infinite set)} \\ A & = & V \mid \perp \mid A \supset A' \mid (\forall V)A \mid (A) & \end{array}$$

Examples: $(p_0 \supset p_1)$, $(\forall p_0)(p_0 \supset p_1)$, $(\forall p_1)(\forall p_2)((p_2 \supset p_2) \supset \perp)$

Intuitively, the meaning of $\mathbf{P^2}$ formulas is obvious.

Instead of $(\forall p)A$ we will sometimes use the notation $\forall p.A$. This notation uses fewer parentheses and is used in proof systems like Nuprl.

13.2.2 Increased Expressiveness

A formula like $(\forall p_0)\, p_0 \supset \perp$ states that it is impossible to make every propositional formula true. Statements of this nature could not be expressed in ordinary propositional logic.

13.2.3 Defined Connectives

Connectives like $\sim$, $\vee$, $\wedge$, and $\exists$ do not have to be included in the basic language of $\mathbf{P^2}$. Instead, the can can be defined in terms of $\perp$, $\supset$, and $\forall$:

$$\begin{array}{rcl} \sim A & \equiv & A \supset \perp \\ A \wedge B & \equiv & \sim(A \supset \sim B) \\ A \vee B & \equiv & (\sim A) \supset B \\ (\exists p) A & \equiv & \sim(\forall p \sim A) \end{array}$$

Some authors use the following definitions, which even make the constant $\perp$ a defined expression.

$$\begin{array}{rcl} \perp & \equiv & (\forall p) p \\ \sim A & \equiv & (\forall p)(A \supset p) \\ A \wedge B & \equiv & (\forall p)((A \supset (B \supset p)) \supset p) \\ A \vee B & \equiv & (\forall p)((A \supset p) \supset (B \supset p) \supset p) \\ (\exists p) A & \equiv & (\forall q)((\forall p. A \supset q) \supset q) \end{array}$$

13.3 Substitution

Substitution is the key to describing the meaning of quantified formulas as well as to formal reasoning about them. A formula of the form $(\forall p) A$ means that A must be true no matter what we put in – or substitute – for the variable p. In order to explain substitution, we need to understand the role of variable occurrences in a formula.

13.3.1 Free and Bound Variables

Quantified variables are considered to be *bound* in the formula that begins with the corresponding quantifier. Otherwise they are considered to be *free*. Free variables stand for arbitary propositional formulas, which means that the truth of the formula should not change if the variable is instantiated.

For A a formula of $\mathbf{P^2}$, the set of propositional variables that are free in A, denoted $FV(A)$, can be characterized by the following recursive definition:

$$\begin{array}{lcl} FV(\perp) & = & \emptyset \\ FV(p) & = & \{p\} \\ FV(A \supset B) & = & FV(A) \cup FV(B) \\ FV((\forall p) A) & = & FV(A) - \{p\} \end{array}$$

The set of all propositional variables that occur in A, $PV(A)$, can likewise be defined as

$$\begin{array}{lcl} PV(\bot) & = & \varnothing \\ PV(p) & = & \{p\} \\ PV(A{\supset}B) & = & PV(A){\cup}PV(B) \\ PV((\forall p)A) & = & PV(A){\cup}\{p\} \end{array}$$

Examples:

$$\begin{array}{lcc} FV(p_0{\supset}p_1) & = & \{p_0,p_1\} \\ PV(p_0{\supset}p_1) & = & \{p_0,p_1\} \\ FV((\forall p_0)(p_0{\supset}p_1)) & = & \{p_1\} \\ PV((\forall p_0)(p_0{\supset}p_1)) & = & \{p_0,p_1\} \\ FV((\forall p_1)(\forall p_2)((p_2{\supset}p_2){\supset}\bot)) & = & \varnothing \\ PV((\forall p_1)(\forall p_2)((p_2{\supset}p_2){\supset}(\forall p_3 p_1))) & = & \{p_1,p_2,p_3\} \end{array}$$

We can extend the definitions of FV and PV to finite sets of formulas by taking $FV(\Gamma) = \bigcup_{A\in\Gamma} FV(A)$ and likewise by taking $PV(\Gamma) = \bigcup_{A\in\Gamma} PV(A)$.
For sequents, the definitions are $FV(\Delta{\vdash}\Gamma) = FV(\Delta{\cup}\Gamma)$ and $PV(\Delta{\vdash}\Gamma) = PV(\Delta{\cup}\Gamma)$.

13.4 Defining Substitution

Substitution $A|^p_B$ is the replacement of *all* occurrences of the variable p in A by the formula B. There are a few issues, however, that one needs to be aware of.

Variables that are bound by a quantifier, must not be replaced, as this would change the meaning. $((\exists p)(p{\supset}{\sim}q))|^p_q$ should not result in $((\exists p)(q{\supset}{\sim}q))$ as the former is a tautology (choose $p = \bot$) while the latter depends on the value of q (and ths is only satisfiable).

In the same way, a variable must not be replaced by a bound variable, as this may change the meaning of the formula. For instance, the formula $(\exists q)((p{\supset}q)\wedge(q{\supset}p))$ is a tautology (choose $q = p$), but defining $(\exists q)((p{\supset}q)\wedge(q{\supset}p))|^p_{\sim q}$ as $(\exists q)((\sim q{\supset}q)\wedge(q{\supset}{\sim}q)$ is unsatisfiable.

The formal definition takes both issues into account. In the former case, nothing will be substituted, in the latter case, variable *capture* is avoided by renaming the bound variable first.

Given formulas A and B of $\mathbf{P^2}$ and a propositional variable p, the $\mathbf{P^2}$ formula $A|^p_B$ ("A with B substituted for p") is, as usual, defined recursively:

$$\begin{array}{rcll} \bot|^p_B & = & \bot & \\ p|^p_B & = & B & \\ q|^p_B & = & q & (q{\neq}p) \\ (A{\supset}A')|^p_B & = & (A|^p_B){\supset}(A'|^p_B) & \\ ((\forall p)A)|^p_B & = & \forall pA & \\ ((\forall q)A)|^p_B & = & \forall q(A|^p_B) & (q{\neq}p,\ q\notin FV(B)) \\ ((\forall q)A)|^p_B & = & \forall q'(A|^q_{q'}|^p_B) & (q{\neq}p,\ q\in FV(B),\ q'\notin PV(A,B,p)) \end{array}$$

Examples:

$$\begin{aligned}
(p_0 \supset p_1)\,|^{p_0}_{p_2 \supset p_3} &= ((p_2 \supset p_3) \supset p_1) \\
(p_0 \supset (p_0 \supset p_1))\,|^{p_0}_{p_3} &= (p_3 \supset (p_3 \supset p_1)) \\
(p_0 \supset p_0)\,|^{p_0}_{p_0 \supset p_0} &= ((p_0 \supset p_0) \supset (p_0 \supset p_0)) \\
(p_0 \supset (\forall p_0 (p_0 \supset p_0)))\,|^{p_0}_{p_1} &= (p_1 \supset (\forall p_0 (p_0 \supset p_0))) \\
(\forall p_0 (p_0 \supset p_3))\,|^{p_3}_{p_0} &= (\forall p_1 (p_1 \supset p_0))
\end{aligned}$$

Again one can extend substitution to finite sets of formulas and then to sequents by letting $\Gamma|^p_B = \{\, A|^p_B \mid A \in \Gamma \,\}$ and $(\Delta \vdash \Gamma)|^p_B = (\Delta|^p_B) \vdash (\Gamma|^p_B)$.

Computer scientists often use the notation $A[B/p]$ instead of $A|^p_B$ to denote the substitution of variables by formulas, while mathematicians like Smullyan prefer $A|^p_B$. In the following we will use the latter for explaining the semantics of formulas (where variables are replaced by truth values) while we use the former to explain the proof system (which replaces variables by formulas).