FORMAL SYSTEMS STUDIED IN THIS COURSE[1]

1. PS (Hunter's axiomatic theory for truth-functional logic):

Axiom-Schemata:

[PS1] $A \supset (B \supset A)$	
[PS2] $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$	[one-way distribution of $\supset$]
[PS3] $(\sim A \supset \sim B) \supset (B \supset A)$	[one-way contraposition]

Rule of Inference:

[MP] From A and $A \supset B$, infer B	[modus ponens]

2. QS (Hunter's axiomatic theory for quantificational logic without =)

Axiom-Schemata:

[QS1] $A \supset (B \supset A)$	
[QS2] $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$	[one-way distribution of $\supset$]
[QS3] $(\sim A \supset \sim B) \supset (B \supset A)$	[one-way contraposition]
[QS4] $(\bigwedge vA \supset At/v)$ if t is free for v in At/v	[universal instantiation]
[QS5] $(A \supset \bigwedge vA)$ if v does not occur free in A	[universal generalization]
[QS6] $(\bigwedge v(A \supset B) \supset (\bigwedge vA \supset \bigwedge vB))$	[one-way distribution of $\bigwedge$]
[QS7] If A is an axiom, then $\bigwedge vA$ is also an axiom.	[axiom universalization]

Rule of Inference:

[MP] From A and $A \supset B$, infer B	[modus ponens]

3. $QS^=$ (Hunter's axiomatic theory for quantificational logic with =)

$QS^=$ is the same as QS, except $QS^=$ also includes:

Bonus Axiom:

[$QS^=$1] $\bigwedge x\,(x = x)$	[universal self-identity]

Bonus Axiom-Schema:

[$QS^=$2] Any closure of $x = y \supset (A \supset B)$, where B is like A except y may replace any free x, as long as y is also free wherever it replaces x.	[indiscernibility of identicals]

[1] On notation, see the earlier handout "Main Formal Languages Used in this Course."

4. System H (Hunter's axiomatic theory for arithmetic)
Some of the axioms may seem unnecessary, but they promise to make the metaproofs easier.

Axioms
All axioms from the Axiom-Schemata in $\mathrm{QS}^{=}2$, as well as…

1. $\bigwedge x\, (x = x)$ [universal self-identity, i.e., $\mathrm{QS}^{=}1$]
2. $\bigwedge x \bigwedge y \bigwedge z\, (x = y \supset (y = z \supset x = z))$ [transitivity of =, 1^{st} clause]
3. $\bigwedge x \bigwedge y \bigwedge z\, (x = y \supset (x = z \supset y = z))$ [transitivity of =, 2^{nd} clause]
4. $\bigwedge x \bigwedge y\, (x = y \supset \mathrm{S}x = \mathrm{S}y))$ [successor is a total function]
5. $\bigwedge x\, (0 < \mathrm{S}x)$ [0 is less than any successor]
6. $\bigwedge x\, (\sim x < 0)$ [nothing is less than 0]
7. $\bigwedge x \bigwedge y\, (x < y \supset \mathrm{S}x < \mathrm{S}y)$ [ordering of successors, 1^{st} clause]
8. $\bigwedge x \bigwedge y\, (x < y \supset \sim x = y)$ [def. of <, 1^{st} clause]
9. $\bigwedge x \bigwedge y\, (x < y \supset \sim y = x)$ [def. of <, 2^{nd} second clause]
10. $\bigwedge x \bigwedge y\, (x < y \supset \sim y < x)$ [< is asymmetric]
11. $\bigwedge x \bigwedge y\, (\sim x < y \supset (\sim y < x \supset x = y))$ [≥ is antisymmetric]
12. $\bigwedge x \bigwedge y\, (y < \mathrm{S}x \supset (y < x \vee y = x)$ [ordering of successors, 2^{nd} clause]
13. $\bigwedge x\, (x + 0 = x)$ [recursive def. of +, basis clause]
14. $\bigwedge x \bigwedge y\, (x + \mathrm{S}y = \mathrm{S}(x + y))$ [recursive def. of +, inductive clause]
15. $\bigwedge x \bigwedge y \bigwedge z \bigwedge w\, (x = y \supset (z + x = w \supset z + y = w))$ [+ is a function, 1^{st} clause]
16. $\bigwedge x \bigwedge y \bigwedge z \bigwedge w\, (x = y \supset (z = x + w \supset z = y + w))$ [+ is a function, 2^{nd} clause]
17. $\bigwedge x\, (x \cdot 0 = 0)$ [recursive def. of ·, basis clause]
18. $\bigwedge x \bigwedge y\, (x \cdot \mathrm{S}y = (x \cdot y) + x)$ [recursive def. of ·, inductive clause]
19. $\bigwedge x \bigwedge y \bigwedge z \bigwedge w\, (x = y \supset (z \cdot x = w \supset z \cdot y = w))$ [· is a function, 1^{st} clause]
20. $\bigwedge x \bigwedge y \bigwedge z \bigwedge w\, (x = y \supset (z = x \cdot w \supset z = y \cdot w))$ [· is a function, 2^{nd} clause]
21. $\bigwedge x\, (\mathrm{P}x0 = \mathrm{S}0)$ [recursive def. of P, basis clause][2]
22. $\bigwedge x \bigwedge y\, (\mathrm{P}x\mathrm{S}y = \mathrm{P}xy \cdot x)$ [recursive def. of P, inductive clause]
23. $\bigwedge x \bigwedge y \bigwedge z \bigwedge w\, (x = y \supset (\mathrm{P}zx = w \supset \mathrm{P}zy = w))$ [P is a function]
24. $\bigwedge x \bigwedge y \bigwedge z \bigwedge w\, [((y < x \wedge x = y + z) \vee (\sim y < x \wedge z = 0)) \supset$
$(((y < x \wedge x = y + w) \vee (\sim y < x \wedge w = 0)) \supset z = w)]$ [secures that $x - y$ is unique, if $x \geq y$]

[2] Hunter intends 'P' to express exponentiation.

5. Q (Robinson Arithmetic)
Do not confuse Q in this context, which is a formal <u>theory</u> *of arithmetic, with Hunter's language Q, which is a formal* <u>language</u> *for quantificational logic.*

We assume here that the language of Robinson Arithmetic is the same as the language of H.

<u>Axioms</u>

1. $\bigwedge x \bigwedge y\ (\mathrm{S}x = \mathrm{S}y \supset x = y)$ [successor is a function]
2. $\bigwedge x\ (\mathrm{S}x \neq 0)$ [0 is not a successor]
3. $\bigwedge x\ (x \neq 0 \supset \bigvee y\ (x = \mathrm{S}y))$ [all other numbers are successors]
4. $\bigwedge x\ (x + 0 = x)$ [recursive def. of +, basis clause]
5. $\bigwedge x \bigwedge y\ (x + \mathrm{S}y = \mathrm{S}(x + y))$ [recursive def. of +, inductive clause]
6. $\bigwedge x\ (x \cdot 0 = 0)$ [recursive def. of ·, basis clause]
7. $\bigwedge x \bigwedge y\ (x \cdot \mathrm{S}y = (x \cdot y) + x)$ [recursive def. of ·, inductive clause]

<u>Rules of Inference</u>
Usually, all the standard inference rules from quantificational logic are fair game. Hunter (p. 238) understands Robinson arithmetic as axioms 1-7 added to $\mathrm{QS}^{=}$.

6. PA (Peano Arithmetic)
PA is the same as Q, except that PA also includes the following axiom-schema:

1. $\bigwedge \mathrm{F}\ ([\mathrm{F}0 \wedge \bigwedge x\ (\mathrm{F}x \supset \mathrm{FS}x)] \supset \bigwedge y\ \mathrm{F}y)$ [axiom of induction]