

Natural Deduction Rules for the Final Exam

Philosophy 12A

June 23, 2010

Rule of Assumptions:

$$j \quad (j) \quad p \quad \text{Assumption (or: Premise)}$$

where p may be any formula (or a premise of the sequent).

Rule of &-Elimination (&E):

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad p \ \& \ q \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad p \quad j \ \& \ E \\ \text{or} \\ a_1, \dots, a_n \quad (k) \quad q \quad j \ \& \ E \end{array}$$

Rule of &-Introduction (&I):

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad p \\ \vdots \\ b_1, \dots, b_u \quad (k) \quad q \\ \vdots \\ a_1, \dots, a_n, b_1, \dots, b_u \quad (m) \quad p \ \& \ q \quad j, k \ \& \ I \end{array}$$

Rule of \rightarrow -Elimination (\rightarrow E):

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad p \rightarrow q \\ \vdots \\ b_1, \dots, b_u \quad (k) \quad p \\ \vdots \\ a_1, \dots, a_n, b_1, \dots, b_u \quad (m) \quad q \quad j, k \rightarrow E \end{array}$$

Rule of \rightarrow -Introduction (\rightarrow I):

$$\begin{array}{l} j \quad (j) \quad p \quad \text{Assumption} \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad q \\ \vdots \\ \{a_1, \dots, a_n\}/j \quad (m) \quad p \rightarrow q \quad j, k \rightarrow I \end{array}$$

Rule of \sim -Elimination (\sim E):

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad \sim q \\ \vdots \\ b_1, \dots, b_u \quad (k) \quad q \\ \vdots \\ a_1, \dots, a_n, b_1, \dots, b_u \quad (m) \quad \wedge \quad j, k \sim E \end{array}$$

Rule of \sim -Introduction (\sim I):

$$\begin{array}{l} j \quad (j) \quad p \quad \text{Assumption} \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad \wedge \\ \vdots \\ \{a_1, \dots, a_n\}/j \quad (m) \quad \sim p \quad j, k \sim I \end{array}$$

Rule of Double Negation (DN):

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad \sim \sim p \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad p \quad j \text{ DN} \end{array}$$

Rule of \vee -Introduction (\vee I):

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad p \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad p \vee q \quad j \vee I \\ \text{or} \\ a_1, \dots, a_n \quad (k) \quad q \vee p \quad j \vee I \end{array}$$

Rule of \vee -Elimination (\vee E):

$$\begin{array}{l} a_1, \dots, a_n \quad (g) \quad p \vee q \\ \vdots \\ h \quad (h) \quad p \quad \text{Assumption} \\ \vdots \\ b_1, \dots, b_u \quad (i) \quad r \\ \vdots \\ j \quad (j) \quad q \quad \text{Assumption} \\ \vdots \\ c_1, \dots, c_w \quad (k) \quad r \\ \vdots \\ \mathcal{X} \quad (m) \quad r \quad g, h, i, j, k \vee E \end{array}$$

$$\mathcal{X} = \{a_1, \dots, a_n\} \cup \{b_1, \dots, b_u\}/h \cup \{c_1, \dots, c_w\}/j$$

Rule of Definition for \leftrightarrow (Df):

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad (p \rightarrow q) \& (q \rightarrow p) \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad p \leftrightarrow q \quad j \text{ Df} \end{array}$$

OR

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad p \leftrightarrow q \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad (p \rightarrow q) \& (q \rightarrow p) \quad j \text{ Df} \end{array}$$

Rule of \exists -Introduction:

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad \phi\tau \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad (\exists v)\phi v \quad j \exists I \end{array}$$

$\lceil(\exists v)\phi v\rceil$ is obtained from $\phi\tau$ by replacing *one or more* occurrences of τ in $\phi\tau$ by an individual variable v (*which must not occur in $\phi\tau$*) and then prefixing the quantifier $\lceil(\exists v)\rceil$.

Rule of \forall -Elimination:

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad (\forall v)\phi v \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad \phi\tau \quad j \forall E \end{array}$$

$\phi\tau$ is obtained from $\lceil(\forall v)\phi v\rceil$ by deleting the quantifier prefix $\lceil(\forall v)\rceil$ and then replacing *every* occurrence of v in the open sentence ϕv by *one and the same* constant τ .

Rule of \forall -Introduction:

$$\begin{array}{l} a_1, \dots, a_n \quad (j) \quad \phi\tau \\ \vdots \\ a_1, \dots, a_n \quad (k) \quad (\forall v)\phi v \quad j \forall I \end{array}$$

Where τ is not in any of the formulae on lines a_1, \dots, a_n and v is not in $\phi\tau$. $\lceil(\forall v)\phi v\rceil$ is obtained by replacing *every* occurrence of the constant τ in $\phi\tau$ with the variable v and then prefixing the universal quantifier $\lceil(\forall v)\rceil$.

Rule of \exists -Elimination:

$$\begin{array}{l} a_1, \dots, a_n \quad (i) \quad (\exists v)\phi v \\ \vdots \\ j \quad (j) \quad \phi\tau \quad \text{Assumption} \\ \vdots \\ b_1, \dots, b_u \quad (k) \quad \mathcal{P} \\ \vdots \\ \mathcal{X} \quad (m) \quad \mathcal{P} \quad i, j, k \exists E \end{array}$$

Where τ is not in (i) $\lceil(\exists v)\phi v\rceil$, (ii) \mathcal{P} , or (iii) any of the formulae b_1, \dots, b_u (other than $\phi\tau$ itself). The set of line numbers at line m is: $\mathcal{X} = \{a_1, \dots, a_n\} \cup \{b_1, \dots, b_u\}/j$.

Rule of Sequent Introduction (SI):

Suppose the sequent $r_1, \dots, r_n \vdash s$ is a *substitution-instance* of a sequent $p_1, \dots, p_n \vdash q$ which appears in our Official List of Sequents and Theorems, and that the formulae r_1, \dots, r_n occur at lines j_1, \dots, j_n in a proof. Then we may infer s at line k, labeling the line ' j_1, \dots, j_n SI (Identifier)' and writing on the left all the numbers which appear on the left of lines j_1, \dots, j_n . The "Identifier" for a sequent is given by its name in our Official List of Sequents and Theorems (see below).

Rule of Theorem Introduction (TI):

If $\vdash s$ is a *substitution-instance* of some theorem $\vdash q$ which appears in our Official List of Sequents and Theorems, we may introduce a new line k into a proof with the formula s at it and no numbers on its left, labeling the line 'TI (Identifier)'. The *only* theorem in our Official List is LEM — see below.

Our Official List of Sequents and Theorems

$A \vee B, \sim A \vdash B$; or; $A \vee B, \sim B \vdash A$	(DS)
$A \rightarrow B, \sim B \vdash \sim A$	(MT)
$A \vdash B \rightarrow A$	(PMI)
$\sim A \vdash A \rightarrow B$	(PMI)
$A \vdash \sim \sim A$	(DN ⁺)
$\sim(A \& B) \dashv\vdash \sim A \vee \sim B$	(DeM)
$\sim(A \vee B) \dashv\vdash \sim A \& \sim B$	(DeM)
$\sim(\sim A \vee \sim B) \dashv\vdash A \& B$	(DeM)
$\sim(\sim A \& \sim B) \dashv\vdash A \vee B$	(DeM)
$A \rightarrow B \dashv\vdash \sim A \vee B$	(Imp)
$\sim(A \rightarrow B) \dashv\vdash A \& \sim B$	(Neg-Imp)
$A \& (B \vee C) \dashv\vdash (A \& B) \vee (A \& C)$	(Dist)
$A \vee (B \& C) \dashv\vdash (A \vee B) \& (A \vee C)$	(Dist)
$\wedge \vdash A$	(EFQ)
$A * B \vdash B * A$	(Com)
$\sim \sim A * \sim \sim B \dashv\vdash A * B$	(SDN)
$A * B \dashv\vdash \sim \sim A * B \dashv\vdash A * \sim \sim B$	(SDN)
$\vdash A \vee \sim A$	(LEM)
$(\forall v)\sim\phi v \dashv\vdash \sim(\exists v)\phi v$	(QS)
$(\exists v)\sim\phi v \dashv\vdash \sim(\forall v)\phi v$	(QS)
$\phi v \dashv\vdash \phi v'$	(AV)

Notes on Our Official List of Sequents and Theorems.

In (Com), '*' can be any binary LSL connective *except* ' \rightarrow '. In (SDN), '*' can be *any* binary LSL connective. In (AV), ϕv must be *closed*, and $\phi v'$ must be an *alphabetic variant* of ϕv . For example, ' $(\forall x)Fx$ ' and ' $(\forall y)Fy$ ' are both *closed* sentences (*i.e.*, they have *no free variables* in them), and they are *alphabetic variants* of each other, because they differ *only* in which individual variable (' x ' or ' y ') is used, but they have exactly the same *logical* (*i.e.*, *syntactical*) structure.