CSCI 1323 (Discrete Structures), Spring 2003

Derivation Rules for Predicate Logic

Inference rules

From	Can derive	Name/abbreviation	Restrictions on use
$(\forall x)P(x)$	$P(t)$, where t is a variable or constant symbol	Universal instantiation — ui	If t is a variable, it must not fall within the scope of a quantifier for t.
$(\exists x)P(x)$	$P(t)$, where t is a variable or constant symbol not previously used in proof sequence	Existential instantiation — ei	Must be the first rule used that introduces t.
$P(x)$	$(\forall X)P(x)$	Universal generalization — ug	$P(x)$ has not been deduced from any hypotheses in which x is a free variable, nor has $P(x)$ been deduced by ei from any wff in which x is a free variable.
$P(x)$ or $P(a)$, where $P(a)$ is a constant symbol	$(\exists X)P(x)$	Existential generalization — eg	To go from $P(a)$ to $(\exists x)P(x)$, x must not appear in $P(a)$.

Temporary hypotheses

Given hypotheses $P_1, \ldots P_n$, you can derive $T \rightarrow S$ in a proof sequence as follows:

1. Introduce T as a "temporary hypotheses": Write T as the n-th step of the proof sequence, with a justification of "temporary hyp".
2. Prove S given T and the hypotheses: Apply derivation rules to steps 1 through n of the sequence to produce steps $n+1$ through m, where step m derives S. Indent these steps to indicate that they depend on a temporary hypothesis.
3. Derive $T \rightarrow S$, giving as a justification "temp. hyp discharged".

You then continue with the main proof sequence as usual, except that the indented steps ($n+1$ through m) cannot be used to derive subsequent steps in the sequence.

Equivalence rule

Expression	Equivalent to	Name/abbreviation
$((\exists x)A(x))'$	$(\forall x)(A(x)')$	Negation — neg