Proof systems for Dynamic Predicate Logic

Frank Veltman

Department of Philosophy, University of Amsterdam

March 2000

1 Introduction

I am afraid this paper will appeal to only a select group of readers. One has to be well acquainted with Fitch style natural deduction for static predicate logic, one has to be well acquainted with dynamic predicate logic, and then, most importantly, one should be curious to know if and how a Fitch style natural deduction system for static predicate logic can be made dynamic. [1]

Of course, the most important step in going from a static to a dynamic proof system is to replace the elimination rule for the static existential quantifier pictured below[2] by an elimination rule that fits the dynamic quantifier.

Static elimination rule for $\exists$

$$\begin{array}{lll} \vdots & \vdots & \\ l & \exists x\varphi & \\ \vdots & \vdots & \\ m & [^a/x]\varphi \rightarrow \psi & \\ \vdots & \vdots & \\ n & \psi & E_\exists, l, m \end{array}$$

Dynamic elimination rule for $\exists$

$$\begin{array}{lll} \vdots & \vdots & \\ m & \exists x\varphi & \\ \vdots & \vdots & \\ n & \varphi & E_\exists, m \end{array}$$

1. I would not be surprised if this leaves only Jeroen, Martin, and Roel de Vrijer as interested readers. The four of us tried to answer these questions around 1990, when *DPL* was being developed. But at some point we gave up because (i) things turned out to be more complicated than we thought, and (ii) we had more urgent things to do. I tried again in 1997, during a sabbatical spent in Edinburgh. I never published the result.
2. Throughout I will take the system of natural deduction presented in L.T.F. Gamut, *Logic, Language, and Meaning, volume 1, Introduction to Logic* as the starting point

With an appeal to this rule, one can easily show, for example, that in dynamic predicate logic

$$\forall x(Sx \to Px) \models \exists y Sy \to Py.$$

Here is the derivation:

	1	$\forall x(Sx \to Px)$	premise
⌐	2	$\exists y Sy$	assumption
|	3	Sy	$\mathrm{E}_\exists$ 2
|	4	$Sy \to Py$	$\mathrm{E}_\forall$ 1
|	5	Py	$\mathrm{E}_\to$ 3,4
⌙	6	$\exists y Sy \to Py$	$\mathrm{I}_\to$

It is important to realize that the sequence of formulas constituting this derivation is a *text* in the language of *DPL*. The variable y occurring in the formula on line 3 is bound by the quantifier $\exists y$ occurring in the formula on line 4, and so are all occurrences of y in line 4 and 5.

2 Some obstacles

Unfortunately, changing the elimination rule for the existential quantifier is not the only thing one has to do to get a system that covers the dynamic notion of validity. Here are some problems that one has to deal with.

(1) How would you go about proving that

$$Ac.\, Bc \models \exists x Ax \wedge Bx$$

If you start like this :

1	Ac	premise
2	Bc	premise
3	$Ac \wedge Bc$	$\mathrm{I}_\wedge$ 1, 2
4	...	

and next, at line 4, apply the classic introduction rule for $\exists$, you end up with $\exists x(Ax \wedge Bx)$, which is not what you want. And if you first apply the introduction rule for $\exists$, like this:

1	Ac	premise
2	Bc	premise
3	$\exists x Ax$	$\mathrm{I}_\exists$ 1
4	...	

then there seems to be no way to get Bx at line 4.

What is needed here is a generalization of of $\mathrm{I}_\exists$. It will be a rule that can be applied not only formulas but also to texts, as follows:

1	Ac	premise
2	Bc	premise
3	$\exists x Ax$	
4	Bx	$\mathrm{I}_\exists$ 1-2
5	$\exists x Ax \wedge Bx$	$\mathrm{I}_\wedge$ 3, 4

So, lines 3 and 4 are added in one go, by an application of $\mathrm{I}_\exists$ to the text formed by the formulas on line 1 and line 2.

(2) Here is another problem. How would you go about showing that:

$$\exists x(Ax \wedge \exists x \neg Ax) \models \neg Ax$$

We cannot apply the new $\mathrm{E}_\exists$ as described above here, because that would give:

1	$\exists x(Ax \wedge \exists x \neg Ax)$	premise
2	$Ax \wedge \exists x \neg Ax$	$\mathrm{E}_\exists$
3	$\ldots$	

This way, the variable x occurring in the first conjunct of the formula on line 2 gets bound by the second existential quantifier on line 1 —- with disastrous consequences. (We should impose conditions on $\mathrm{E}_\exists$ that forbid this.)

The way out here is to introduce a new rule, *variable switch*, which enables us — under certain conditions — to replace a formula of the form $\exists x\varphi$ by a variant $\exists y[y/x]\varphi$. This is how things work out for the example at hand:

1	$\exists x(Ax \wedge \exists x \neg Ax)$	premise
2	$\exists y(Ay \wedge \exists x \neg Ax)$	variable switch, 1
3	$Ay \wedge \exists x \neg Ax$	$\mathrm{E}_\exists$, 2
4	$\exists x \neg Ax$	$\mathrm{E}_\wedge$, 3
5	$\neg Ax$	$\mathrm{E}_\exists$, 4

(3) As an example of a third obstacle, note that:

$$\exists y Ay \wedge \exists x Bx, Cy \models \exists y Ay \wedge Cy$$

Whereas,

$$\exists y Ay \wedge \exists x Bx, Cy, \exists y Ay \not\models Cy$$

So, somehow the step 4 in the next derivation is invalid

1	$\exists y Ay \wedge \exists x Bx$	premise
2	Cy	premise
3	$\exists y Ay$	premise
4	Cy	repetition 2

whereas step 4 in the following derivation is fine.

1	$\exists y Ay \wedge \exists x Bx$	premise
2	Cy	premise
3	$\exists y Ay$	$\mathrm{E}_\wedge$ 1
4	Cy	repetition 2
5	$\exists y Ay \wedge Cy$	$\mathrm{I}_\wedge$, 3, 4

Intuitively, the difference is this. The formula $\exists y Ay$ on the third line in the first of these derivations adds a further premise to the premises already given. It introduces a new discourse referent, and all that is said about it is that the

predicate A applies to it. So, to conclude that the predicate B applies to it, as is done in line 4 is invalid. The formula $\exists y Ay$ on the third line in the second derivation, on the other hand, is a conclusion drawn from the premises, and just recapitulates what was already said: There is an object with property A. And, yes, we already know about this object that it has property B as well.

It is not easy to characterize the difference formally. In the derivation we will have to keep track for each existential quantifier where it was first introduced, so that when it is just 'repeated' we can safely repeat things said about the referents introduced by it.

3 Preliminaries

Goal
To find a sound and complete Fitch style Natural Deduction System $\mathcal{N}$ for DPL.

Strategy
Develop complete Semantic Tableau system $\mathcal{S}$.
Use $\mathcal{S}$ to find complete Deductive Tableau system $\mathcal{D}$
Use $\mathcal{D}$ to find $\mathcal{N}$.

What the reader will find is a motivated description of $\mathcal{S}$, $\mathcal{D}$ and $\mathcal{N}$ plus proofs which show that

$$\pi \models \sigma \Rightarrow \pi \vdash_{\mathcal{S}} \sigma \Rightarrow \pi \vdash_{\mathcal{D}} \sigma \Rightarrow \pi \vdash_{\mathcal{N}} \sigma \Rightarrow \pi \models \sigma$$

Definition 1 (The languages of DPL)

(a) The languages of dynamic predicate logic share the following *logical vocabulary*: $\bot$ (falsum), $\neg$ (negation), $\wedge$ (conjunction); the pair of brackets $(,)$; a denumerable set of syntactic variables $V = \{x_0, x_1, x_2, \ldots\}$; existential quantifiers $\exists x$, for $x \in V$.

(b) A specific language L is identified with its *non-logical vocabulary*: a set of individual constants, and a for each n a set of n-place predicates.

The union of the set of variables and the set of individual constants of L is called the set of *terms* of L. The set of *formulas* of L is defined in the usual way.

The core language can be extended by defining additional logical constants. E.g., we can add '$\rightarrow$' (implication), '$\vee$' (disjunction), and '$\forall x$' (universal quantifiers). The choice of logical primitives is not as optional in DPL as it is in standard predicate logic.

Definition 2 (Extension of core syntax)

(a) $(\phi \rightarrow \psi) = \neg(\phi \wedge \neg\psi)$.
(b) $(\phi \vee \psi) = \neg(\neg\phi \wedge \neg\psi)$.
(c) $\forall x\phi = \neg\exists x\neg\phi$.

Below, the letters π, σ (with or without subscripts) vary over finite, possibly empty sequences of formulas, which are also called texts. We will often write $\varphi_1 . \ldots . \varphi_n$ instead of $\langle \varphi_1, \ldots, \varphi_n \rangle$. If $\pi = \varphi_1 . \ldots . \varphi_n$ and $\sigma = \psi_1 . \ldots . \psi_m$, then $\pi . \sigma$ is short for $\langle \varphi_1, \ldots, \varphi_n, \psi_1 \ldots, \psi_m \rangle$.

Definition 3 (States) Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ be some model for a language of first order predicate logic.

(a) $\mathbf{1}_\mathcal{M}$, the set of *assignments* pertaining to $\mathcal{M}$, is the set of functions from V into $\mathcal{D}$. Whenever it is clear about which model $\mathcal{M}$ we are talking, we will write $\mathbf{1}$ rather than $\mathbf{1}_\mathcal{M}$.

(b) Let v be an assignment, $x \in V$, $d \in \mathcal{D}$. $v[x/d]$ is the assignment u such that $u(x) = d$, while for every $y \in V$: if $y \neq x$ then $u(y) = v(y)$.

(c) A *state* based on $\mathcal{M}$ is a set s of assignments. Here $\mathbf{1}$ can be thought of as the *mininimal state*, and $\emptyset$ as the maximal, or *absurd* state. In this connection we will often write '$\mathbf{0}$' rather than '$\emptyset$'.

(d) Let s be a state, $x \in V$, $d \in \mathcal{D}$.
$s[x/d] = \{u \in \mathbf{1} \mid u = v[x/d] \text{ for some } v \in s\}$.

Definition 4 (Dynamic interpretation) Let s be a state based on $\mathcal{M}$. For every formula φ the update $s[\varphi]$ of s with φ is recursively defined as follows:

(i) $s[Rt_1 \ldots t_n] = \{v \in s \mid \langle v(t_1), \ldots, v(t_n) \rangle \in \mathcal{I}(R)\}$.
(ii) $s[\neg\varphi] = \{v \in s \mid \{v\}[\varphi] = \mathbf{0}\}$
(iii) $s[\varphi \wedge \psi] = s[\phi][\psi]$.
(iv) $s[\exists x \varphi] = \cup_{d \in \mathcal{D}}(s[x/d][\varphi])$.

For a text $\pi = \varphi_1 . \ldots . \varphi_n$ we write $s[\pi]$ to abbreviate $s[\varphi_1] \ldots [\varphi_n]$.

Examples

- John has a new girlfriend. She is blond
  $\exists x Gjx . Bx$
  (Consider $\mathbf{1}[\exists x Gjx][Bx]$).
- John has no new girlfriend. She is blond (??)
  $\neg\exists x Gjx . Bx$
  (Consider $\mathbf{1}[\neg\exists x Gjx][Bx]$)
- If John has a new girlfriend, she is blond
  $\exists x Gjx \rightarrow Bx$
  $\forall x(Gjx \rightarrow Bx)$

In the sequel we need a slightly more general notion of entailment than the one you may be prepared for.

Definition 5 (Support) $s \models \pi$ iff for every $v \in s$, $\{v\}[\pi] \neq \mathbf{0}$.

Definition 6 (Entailment) $\pi \models \sigma$ iff for any model $\mathcal{M}$, and any s based on $\mathcal{M}$, $s[\pi] \models \sigma$.

3.1 Examples

- $\exists x Px \models Px$
- $\exists x Px . \exists x \neg Px \models \neg Px$
- $\exists x Px . \neg\neg \exists x \neg Px \models Px$
- $\exists x(Px \wedge Qx) \models \exists x Px . Qx$
- $\exists x Px \rightarrow Qx \models \forall x(Px \rightarrow Qx)$

Loss of structural properties:

- No Repetition: $\exists x Px . Px \wedge \exists x \neg Px \not\models Px$.
- No Monotony: $\exists x Px \models Px$, but $\exists x Px . \exists x \neg Px \not\models Px$.
- No Permutation: $\exists x \neg Px . \exists x Px \models Px$, but $\exists x Px . \exists x \neg Px \not\models Px$.
- No Cut: $\exists x Px . \exists x \neg Px \models \exists x Px$, and $\exists x Px . \exists x \neg Px . \exists x Px \models Px$, but $\exists x Px . \exists x \neg Px \not\models Px$

DPL is not an extension of standard predicate logic.
$\exists x((Ax \wedge \exists x Bx) \wedge Cx) \models_{cl} \exists x(Cx \wedge (Ax \wedge \exists x Bx))$, but
$\exists x((Ax \wedge \exists x Bx) \wedge Cx) \not\models \exists x(Cx \wedge (Ax \wedge \exists x Bx))$.

3.2 Scope and Binding

Definition 7 (Scope island) A formula φ is a *scope island* iff φ is of the form $\neg\psi$.

Definition 8 (Free and bound variables)

(a) A specific occurrence of x in π is *bound by* a specific occurrence of $\exists x$ in π iff
  (i) The occurrence of x is to the right of the occurrence of $\exists x$;
  (ii) There is no occurrence of a scope island φ in π such that the occurrence of $\exists x$ is inside the occurrence of φ and the occurrence of x is not;
  (iii) The occurrence of $\exists x$ is not to the left of another occurrence of $\exists x$ in π for which (i) and (ii) hold.

(b) A given occurrence of a quantifier $\exists x$ in a text π is *active after* π iff
  (i) There is no occurrence of a scope island φ in π such that the occurrence of $\exists x$ is inside the occurrence of φ;
  (ii) The occurrence of $\exists x$ is not to the left of another occurrence of $\exists x$ in π for which (i) holds.

(c) An occurrence of x is *free in* π iff the occurrence of x is not bound by an occurrence of $\exists x$ in π.

(d) An occurrence of x in σ is *free in* σ *after* π iff the occurrence of x in σ is free in $\pi . \sigma$.

(e) σ *depends on* π iff there is a free occurrence of a variable in σ which is not free in σ after π.

Definition 9 (Substitution)

(a) $[t/x, \pi](\sigma)$ is the result of substituting t for every occurrence of x in σ which is free in σ after π.
(b) y is *free for* x *in* σ *after* π iff y occurs free in $[y/x, \pi](\sigma)$ at all places where x occurs free in σ after π.

We will write '$[c/x](\sigma)$' rather than '$[c/x, \emptyset](\sigma)$'

Lemma 1
Let u and v be two assignments such that $u(x) = v(x)$ for all variables x occurring free in π. Then $\{u\}[\pi] \neq \emptyset$ iff $\{v\}[\pi] \neq \emptyset$

Moreover ...

Lemma 2
(a) Let x be free for y in π. Let s and s' be two states such that $s' = [y/s(x)]s$. Then $v \in s[[x/y]\pi]$ iff $v[y/s(x)] \in s'[\pi]]$
(b) Let x be free for y in π, and suppose that x does not occur free in π. Let s and s' be two states such that $s' = [x/s(y)]s$. Then $v \in s[\pi]$ iff $v[x/s(y)] \in s'[[x/y]\pi]]$

Definition 10 (Strong entailment)
(a) $i \approx_\pi j$ iff $i(x) = j(x)$ for all x such that $\exists x$ is active after π.
(b) $s \models_\pi \sigma$ iff $s \models \sigma$, and for all $i \in s$ and $j \in \{i\}[\sigma]$ it holds that $i \approx_\pi j$.
(c) $\pi \not\approx \sigma$ iff for any s $s[\pi] \models_\pi \sigma$.

Notice:
(a) If $\pi \models \sigma.\sigma'$, then $\pi \models \sigma$.
(b) It is not generally so that if $\pi \models \sigma.\sigma'$, then $\pi \models \sigma'$
(c) If $\pi \models \sigma.\sigma'$, and $\pi \not\approx \sigma$, then $\pi \models \sigma'$

4 Deductive Tableaus

A sequent is a pair $\langle \pi, \sigma \rangle$. We will write $\pi \Rightarrow \sigma$ rather than $\langle \pi, \sigma \rangle$

Call a text π *atomic* iff it consists of just one atomic sentence, and *simple* iff every formula that occurs in it is an atomic sentence. A sequent $\pi \Rightarrow \sigma$ is *simple* if π is simple and σ is atomic.

Definition 11 Let $\sigma = \varphi_1 . \ldots . \varphi_n$.
$\bigwedge \sigma =_{df} (\varphi_1 \wedge (\varphi_2 \wedge (\ldots \wedge (\varphi_{n-1} \wedge \varphi_n) \ldots))$.

Definition 12 (Deductive Tableau)
A deductive tableau for $\pi \Rightarrow \sigma$ is a tree consisting of sequents. This tree $\mathcal{T}$ is constructed as follows.

(i) The root of $\mathcal{T}$ is the sequent $\pi \Rightarrow \sigma$.

(ii) If a node $\pi' \Rightarrow \sigma'$ is simple or an axiom, then this node will have no successors.

(iii) If a node $\pi' \Rightarrow \sigma'$ is neither simple nor an axiom, then this node will have one or two immediate successors. For $\pi'' \Rightarrow \sigma''$ to be the one immediate successor of $\pi' \Rightarrow \sigma'$, it is neccessary that

$$\frac{\pi'' \Rightarrow \sigma''}{\pi' \Rightarrow \sigma'}$$

is a deductive tableau rule, while for $\pi'' \Rightarrow \sigma''$ and $\pi''' \Rightarrow \sigma'''$ to be the two immediate successors of $\pi' \Rightarrow \sigma'$ it is necessary that

$$\frac{\pi'' \Rightarrow \sigma'' \quad \pi''' \Rightarrow \sigma'''}{\pi' \Rightarrow \sigma'}$$

is a tableau rule.

The deductive tableau rules are given in the table below.

$\pi . Rt_1 \dots t_n . \pi' \Rightarrow Rt_1 \dots t_n$ (axiom)	Provided $Rt_1 \dots t_n$ does not depend on π'
$\dfrac{\pi . \varphi . \psi . \pi' \Rightarrow \sigma}{\pi . \varphi \wedge \psi . \pi' \Rightarrow \sigma}\ \mathrm{E}_\wedge$	Provided —
$\dfrac{\pi \Rightarrow \sigma . \varphi . \psi . \sigma'}{\pi \Rightarrow \sigma . \varphi \wedge \psi . \sigma'}\ \mathrm{I}_\wedge$	Provided —
$\dfrac{\pi . [^c/_x](\varphi . \pi') \Rightarrow [^c/_x, \varphi . \pi'](\sigma)}{\pi . \exists x \varphi . \pi' \Rightarrow \sigma}\ \mathrm{E}_\exists$	Provided c does not occur in π, φ, π', or σ
$\dfrac{\pi \Rightarrow \sigma . [^t/_x](\varphi . \sigma')}{\pi \Rightarrow \sigma . \exists x \varphi . \sigma'}\ \mathrm{I}_\exists$	Provided t is free for x in $\varphi . \sigma'$
$\dfrac{\pi . \pi' \Rightarrow \varphi}{\pi . \neg \varphi . \pi' \Rightarrow \bot}\ \mathrm{E}_\neg$	Provided φ does not depend on π'
$\dfrac{\pi . \varphi \Rightarrow \bot}{\pi \Rightarrow \neg \varphi}\ \mathrm{I}_\neg$	Provided —
$\dfrac{\pi \Rightarrow \neg\neg \varphi}{\pi \Rightarrow \varphi}\ \neg\neg$	Provided —
$\dfrac{\pi \Rightarrow \sigma . \varphi \wedge \psi . \sigma'}{\pi \Rightarrow \sigma . \varphi . \psi . \sigma'}$ sen	Provided —
$\dfrac{\pi \Rightarrow \sigma \quad \pi . \sigma \Rightarrow \sigma'}{\pi \Rightarrow \sigma . \sigma'}$ seq	Provided —

Definition 13 $\pi \vdash_{\mathcal{D}} \sigma$ iff there exists some finite deductive tableau with root $\pi \Rightarrow \sigma$ such that all sequents at the top are axioms.

Examples

(a) The example below shows why we cannot restrict our attention to sequents $\pi \Rightarrow \varphi$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ac.Bc \Rightarrow Ac \quad Ac.Bc.Ac \Rightarrow Bc}{Ac.Bc \Rightarrow Ac.Bc}\,\text{seq}}{Ac.Bc \Rightarrow \exists x Ax.Bx}\,\mathrm{I}_\exists}{Ac.Bc \Rightarrow \exists x Ax \wedge Bx}\,\mathrm{I}_\wedge}{Ac \wedge Bc \Rightarrow \exists x Ax \wedge Bx}\,\mathrm{E}_\wedge}{\exists x(Ax \wedge Bx) \Rightarrow \exists x Ax \wedge Bx}\,\mathrm{E}_\exists$$

(b) The rules $\neg\neg$ and sen, which differ from the other rules in that successor sequent is more complex than the predecessor, could be replaced by one rule that would allow for a strong form of reductio ad absurdum.

$$\dfrac{\pi.\neg\bigwedge\sigma \Rightarrow \bot}{\pi \Rightarrow \sigma}\,\text{reductio}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ac.Bc \Rightarrow Ac \quad Ac.Bc.Ac \Rightarrow Bc}{Ac.Bc \Rightarrow Ac.Bc}\,\text{seq}}{Ac.Bc \Rightarrow \exists x Ax.Bx}\,\mathrm{I}_\exists}{Ac \wedge Bc \Rightarrow \exists x Ax.Bx}\,\mathrm{E}_\wedge}{\exists x(Ax \wedge Bx) \Rightarrow \exists x Ax.Bx}\,\mathrm{E}_\exists}{\exists x(Ax \wedge Bx) \Rightarrow \exists x Ax \wedge Bx}\,\mathrm{I}_\wedge}{\exists x(Ax \wedge Bx).\neg(\exists x Ax \wedge Bx) \Rightarrow \bot}\,\mathrm{E}_\neg}{\neg(\exists x Ax \wedge Bx) \Rightarrow \neg\exists x(Ax \wedge Bx)}\,\mathrm{I}_\neg}{\neg\neg\exists x(Ax \wedge Bx).\neg(\exists x Ax \wedge Bx) \Rightarrow \bot}\,\mathrm{E}_\neg}{\neg\neg\exists x(Ax \wedge Bx) \Rightarrow \neg\neg(\exists x Ax \wedge Bx)}\,\mathrm{I}_\neg}{\neg\neg\exists x(Ax \wedge Bx) \Rightarrow \exists x Ax \wedge Bx}\,\neg\neg}{\neg\neg\exists x(Ax \wedge Bx) \Rightarrow \exists x Ax.Bx}\,\text{sen}$$

For practical purposes, it's pleasant to add the following rule:

$\dfrac{\pi \Rightarrow \bot}{\pi \Rightarrow \sigma}$ ex falso	Provided —

However, this rule is in fact not needed.

Proposition 1

(a) If $\pi \vdash_{\mathcal{D}} \sigma$, then $\pi.\pi' \vdash_{\mathcal{D}} \sigma$, provided that σ does not depend on π'.

(b) $\pi.\varphi \vdash_{\mathcal{D}} \varphi$, provided that φ does not depend on φ.

Proof:

(a) ...

(b) ...

Corollaries

(a) If $\pi \vdash_{\mathcal{D}} \bot$, then $\pi \vdash_{\mathcal{D}} \sigma$

(b) $\pi.\neg\varphi \vdash_{\mathcal{D}} \neg\varphi$

4.1 Intermediate Tableaus

With one exception the rules for intermediate tableaus are the same as the rules for deductive tableaus. The elimation rule for the existential quantifier

$\dfrac{\pi\,.\,[^c/_x](\varphi\,.\,\pi') \Rightarrow [^c/_x, \varphi\,.\,\pi'](\sigma)}{\pi\,.\,\exists x\varphi\,.\,\pi' \Rightarrow \sigma}\ \mathrm{E}_\exists$	Provided c does not occur in π, φ, π', or σ

is replaced by the following two rules:

$\dfrac{\pi\,.\,\exists y[^y/_x](\varphi\,.\,\pi') \Rightarrow [^y/_x, \varphi\,.\,\pi'](\sigma)}{\pi\,.\,\exists x\varphi\,.\,\pi' \Rightarrow \sigma}$ alf	Provided y is free for x in $\varphi\,.\,\pi'\,.\,\sigma$
$\dfrac{\pi\,.\,\varphi\,.\,\pi' \Rightarrow \sigma}{\pi\,.\,\exists x\varphi\,.\,\pi' \Rightarrow \sigma}$ pron	Provided x is free in $\varphi\,.\,\pi'\,.\,\sigma$ after π

Let's call the resulting system $\mathcal{D}'$. We will show now that if $\pi \vdash_{\mathcal{D}} \sigma$, then $\pi \vdash_{\mathcal{N}} \sigma$ by proving the following two lemmata.

Lemma 3
If $\pi \vdash_{\mathcal{D}} \sigma$, then $\pi \vdash_{\mathcal{D}'} \sigma$

Lemma 4
If $\pi \vdash_{\mathcal{D}'} \sigma$, then $\pi \vdash_{\mathcal{N}} \sigma$

5 Semantic Tableaus

A *generalised sequent* (g-sequent) is a pair $\langle \pi, \Delta \rangle$, where π is a text and Δ a set of texts. We will write and $\pi \Rightarrow \Delta$ rather than and $\langle \pi, \Delta \rangle$. A generalised sequent $\pi \Rightarrow \Delta$ is *simple* iff π is simple and each σ in Δ is atomic.

Definition 14 (Generalised Entailment) Let Δ be a set of texts. $\pi \models \Delta$ iff the following holds for any state s: for every $i \in s[\pi]$, there exists some $\sigma \in \Delta$ such that $\{i\}[\sigma] \neq \emptyset$.

The definition says that $\pi \models \{\sigma\}$ iff for any s, $s[\pi]$ supports σ. So, when Δ is a singleton, we are back in the old case. We will always write '$\pi \models \sigma$' rather than $\pi \models \{\sigma\}$.

Semantic tableaus are constructed by applying the following rules.

$\pi\,.\,Rt_1\ldots t_n\,.\,\pi' \Rightarrow \Delta, Rt_1\ldots t_n$ (axiom)	Provided $Rt_1\ldots t_n$ does not depend on π'
$\dfrac{\pi\,.\,\varphi\,.\,\psi\,.\,\pi' \Rightarrow \Delta}{\pi\,.\,\varphi\wedge\psi\,.\,\pi' \Rightarrow \Delta}\ \mathrm{E}_\wedge$	Provided —
$\dfrac{\pi \Rightarrow \Delta, \sigma\,.\,\varphi\,.\,\psi\,.\,\sigma'}{\pi \Rightarrow \Delta, \sigma\,.\,\varphi\wedge\psi\,.\,\sigma'}\ \mathrm{I}_\wedge$	Provided —
$\dfrac{\pi\,.\,[^c/_x](\varphi\,.\,\pi') \Rightarrow [^c/_x,\varphi\,.\,\pi'](\Delta)}{\pi\,.\,\exists x\varphi\,.\,\pi' \Rightarrow \Delta}\ \mathrm{E}_\exists$	Provided c does not occur in π, φ, π', or Δ
$\dfrac{\pi \Rightarrow \Delta, \sigma\,.\,[^t/_x](\varphi\,.\,\sigma')}{\pi \Rightarrow \Delta, \sigma\,.\,\exists x\varphi\,.\,\sigma'}\ \mathrm{I}_\exists$	Provided t is free for x
$\dfrac{\pi\,.\,\pi' \Rightarrow \Delta, \varphi}{\pi\,.\,\neg\varphi\,.\,\pi' \Rightarrow \Delta}\ \mathrm{E}_\neg$	Provided φ does not depend on π'
$\dfrac{\pi\,.\,\varphi \Rightarrow \Delta}{\pi\,.\ \Rightarrow \Delta, \neg\varphi}\ \mathrm{I}_\neg$	Provided no σ in Δ depends on φ
$\dfrac{\pi \Rightarrow \Delta, \sigma' \quad \pi\,.\,\sigma' \Rightarrow \Delta, \sigma''}{\pi \Rightarrow \Delta, \sigma'\,.\,\sigma''}\ \mathrm{seq}$	Provided no σ in Δ depends on σ'

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ac\,.\,Bc \Rightarrow Ac \quad Ac\,.\,Bc\,.\,Ac \Rightarrow Bc}{Ac\,.\,Bc \Rightarrow Ac\,.\,Bc}\ \mathrm{seq}}{Ac\,.\,Bc \Rightarrow \exists x Ax\,.\,Bx}\ \mathrm{I}_\exists}{Ac \wedge Bc \Rightarrow \exists x Ax\,.\,Bx}\ \mathrm{E}_\wedge}{\exists x(Ax \wedge Bx) \Rightarrow \exists x Ax\,.\,Bx}\ \mathrm{E}_\exists}{\Rightarrow \exists x Ax\,.\,Bx, \neg\exists x(Ax \wedge Bx)}\ \mathrm{I}_\neg}{\neg\neg\exists x(Ax \wedge Bx) \Rightarrow \exists x Ax\,.\,Bx}\ \mathrm{E}_\neg$$

Proposition 2
If $\pi \vdash_{\mathcal{S}} \sigma$, then $\pi \vdash_{\mathcal{D}} \sigma$
Proof: The proposition follows from the following claim:

$$\text{If } \pi \vdash_{\mathcal{S}} \sigma_1, \ldots, \sigma_n, \text{ then } \pi\,.\,\neg\bigwedge\sigma_1\,.\,\ldots\,.\,\neg\bigwedge\sigma_n \vdash_{\mathcal{D}} \bot$$

The claim is proved with induction on the number of rules employed in in the closed semantic tableau for $\pi \Rightarrow \sigma_1, \ldots, \sigma_n$.

6 Rigid Semantic Tableaus

A rigid sequent (r-sequent) is a pair $\langle \pi, \Sigma \rangle$, where π is a text and Σ a sequence of texts. We will write $\pi \Rightarrow \Sigma$ rather than $\langle \pi, \Sigma \rangle$.

Rules for r-sequents are given in the table below.

$\pi\,.\,Rc_1\ldots c_n\,.\,\pi' \Rightarrow \Sigma; Rc_1\ldots c_n; \Sigma'$ (axiom)	
$\dfrac{\pi\,.\,\varphi\,.\,\psi\,.\,\pi' \Rightarrow \Sigma}{\pi\,.\,\varphi\wedge\psi\,.\,\pi' \Rightarrow \Sigma}\ \mathrm{E}_\wedge$	Provided π is simple
$\dfrac{\pi \Rightarrow \Sigma; \varphi\,.\,\psi\,.\,\sigma; \Sigma'}{\pi \Rightarrow \Sigma; \varphi\wedge\psi\,.\,\sigma; \Sigma'}\ \mathrm{I}_\wedge$	Provided $\pi \Rightarrow \Sigma$ is simple
$\dfrac{\pi\,.\,[^c/_x](\varphi\,.\,\pi') \Rightarrow [^c/_x, \varphi\,.\,\pi'](\Sigma)}{\pi\,.\,\exists x\varphi\,.\,\pi' \Rightarrow \Sigma}\ \mathrm{E}_\exists$	Provided π is simple, and c does not occur in π, φ, π', or Σ
$\dfrac{\pi \Rightarrow \Sigma; [^c/_x](\varphi\,.\,\sigma); \Sigma'; \exists x\varphi\,.\,\sigma}{\pi \Rightarrow \Sigma; \exists x\varphi\,.\,\sigma; \Sigma'}\ \mathrm{I}_\exists$	Provided $\pi \Rightarrow \Sigma$ is simple
$\dfrac{\pi\,.\,\pi' \Rightarrow \Sigma; \varphi}{\pi\,.\,\neg\varphi\,.\,\pi' \Rightarrow \Sigma}\ \mathrm{E}_\neg$	Provided π is simple
$\dfrac{\pi\,.\,\varphi \Rightarrow \Sigma; \Sigma'}{\pi \Rightarrow \Sigma; \neg\varphi; \Sigma'}\ \mathrm{I}_\neg$	Provided $\pi \Rightarrow \Sigma$ is simple
$\dfrac{\pi \Rightarrow \Sigma; \varphi; \Sigma' \quad \pi\,.\,\varphi \Rightarrow \Sigma; \sigma; \Sigma'}{\pi \Rightarrow \Sigma; \varphi\,.\,\sigma; \Sigma'}\ \mathrm{seq}$	Provided $\pi \Rightarrow \Sigma$ is simple, and φ is a negation or atomic

Note that these rules are special cases of the semantic tableau rules.

Definition 15 (Rigid Tableau)
Let $c_1, c_2, \ldots$ be an enumeration of a countable set of individual constants. Given this enumeration, a *rigid tableau* for $\pi \Rightarrow \Sigma$ is a tableau constructed according to the rules given above provided that:

(i) Whenever the rigid rule $\mathrm{E}_\exists$ is applied, i.e. whenever a node

$$\pi\,.\,\exists x\varphi\,.\,\pi' \Rightarrow \Sigma$$

is succeeded by a node

$$\pi\,.\,[^c/_x](\varphi\,.\,\pi') \Rightarrow [^c/_x, \varphi\,.\,\pi'](\Sigma)$$

then the individual constant c is the first of $c_1, c_2, \ldots$ that does not occur in π, φ, π', or Σ.

(ii) Whenever the rigid rule $\mathrm{I}_\exists$ is applied, i.e., whenever a node

$$\pi \Rightarrow \Sigma; \exists x\varphi\,.\,\sigma; \Sigma'$$

is succeeded by a node

$$\pi \Rightarrow \Sigma; [^c/_x](\varphi\,.\,\sigma); \Sigma'; \exists x\varphi\,.\,\sigma$$

then the individual c is the first of $c_1, c_2, \ldots$ such that $[^c/_x](\varphi\sigma)$ has not already been introduced by an application of $\mathrm{I}_\exists$ as a text in the right-hand side of any of the predecessors of the sequent $\pi \Rightarrow \Sigma; \exists x\varphi\,.\,\sigma; \Sigma'$.

Proposition 3
If $\pi \vdash_{\mathcal{R}} \Sigma$, then $\pi \vdash_{\mathcal{S}} \{\sigma \mid \sigma \text{ occurs in the sequence } \Sigma\}$

Lemma 5

(i) Suppose there is some c not occurring in $\varphi\,.\,\psi'\,.\,\sigma$ such that $i \in s[\pi\,.\,[^c/_x](\varphi\,.\,\pi')]$ and $\{i\}[[^c/_x, \varphi\,.\,\pi'](\sigma)] = \emptyset$. Then $i \in s[\pi\,.\,\exists x\varphi\,.\,\pi']$ and $\{i\}[\sigma] = \emptyset$.

(ii) Suppose there is some $i \in s[\pi\,.\,\pi']$, such that $\{i\}[\varphi] = \emptyset$. Then $i \in s[\pi\,.\,\neg\varphi\,.\,\pi']$.

Proposition 4
Fix an enumeration $c_1, c_2, \ldots$ of a countable set of individual constants. Let $\pi \Rightarrow \Sigma$ be any sequent. Given the enumeration, there exists exactly one rigid tableau for $\pi \Rightarrow \Sigma$.

Proposition 5
Suppose the rigid tableau $\mathcal{T}$ for $\pi \Rightarrow \Sigma$ contains an open branch $\mathcal{B}$.
Consider the model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ given by

(a) $\mathcal{D} = \{c \mid c \text{ occurs in any of the sequents constituting } \mathcal{B}\}$

(b) $\mathcal{I}(c_i) = c_i$

(c) If R is an n-place predicate, then $\langle c_{i_1}, \ldots, c_{i_n} \rangle \in \mathcal{I}(R)$ iff there is some node $\pi \Rightarrow \Sigma$ on $\mathcal{B}$ such that $\pi = \pi'\,.\,Rc_{i_1}, \ldots, c_{i_n}\,.\,\pi''$.

Now consider any sequent $\pi \Rightarrow \sigma$ such that $\pi \Rightarrow \sigma_1; \ldots; \sigma; \ldots; \sigma_n$ occurs on the branch $\mathcal{B}$.

Claim: There is some $i \in \mathbf{1}[\pi]$ such that $\{i\}[\sigma] = \mathbf{0}$.

Proof of the claim:
The proof is by induction on the complexity of $\pi \Rightarrow \sigma$, where the complexity of $\pi \Rightarrow \sigma$ is given by the number $m = 2j + l$ where j and l are determined as follows.

- j is the number of logical constants ($\neg$, $\wedge$, $\exists$) occurring in the sequence $\pi\,.\,\sigma$.
- l is the number of formulas in σ.

(i) $m = 1$. Notice that in this case $\pi \Rightarrow \sigma$ is simple. Given the defintion of $\mathcal{M}$, it holds that $\mathbf{1}[\pi] = \mathbf{1}$, and $\mathbf{1}[\pi\,.\,\sigma] = \mathbf{0}$.

(ii) Now assume that the claim holds for any sequent the complexity of which is not larger than k. Consider a sequent $\pi \Rightarrow \sigma$ with complexity $k + 1$. There are three major subcases:

Case (a): π is not simple. Then $\pi \Rightarrow \sigma_1; \ldots; \sigma; \ldots; \sigma_n$is derived from its immediate predecessor by an application of either of the following rules:

– **$\mathbf{E}_\wedge$:** This case is easy. $\pi \Rightarrow \sigma_1; \ldots; \sigma; \ldots; \sigma_n$ is of the form

$$\pi'\,.\,(\varphi \wedge \psi)\,.\,\pi'' \Rightarrow \sigma_1; \ldots; \sigma; \ldots; \sigma_n$$

while its immediate predecessor is

$$\pi'.\varphi.\psi.\pi'' \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$$

Given the semantic rule for $\wedge$-formulas it holds that if $i \in s[\varphi.\psi]$, then $i \in s[\varphi \wedge \psi]$. Given the induction hypothesis we may assume that there is some $i \in \mathbf{1}[\pi.\varphi.\psi.\pi']$ such that $\{i\}[\sigma] = \mathbf{0}$. Hence, there is some $i \in \mathbf{1}[\pi.(\varphi \wedge \psi).\pi']$ such that $\{i\}[\sigma] = \mathbf{0}$.

– **$\mathbf{E}_\exists$:** In this case $\pi \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$ is of the form

$$\pi'.\exists x\varphi.\pi'' \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$$

while its immediate predecessor is

$$\pi'.[^c/x](\varphi.\pi'') \Rightarrow [^c/x,\varphi.\pi'](\sigma_1);\ldots;[^c/x;\varphi.\pi'](\sigma);\ldots;[^c/x,\varphi.\pi'](\sigma_n)$$

– **$\mathbf{E}_\neg$:** In this case we have that $\pi \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$ is of the form

$$\pi'.\neg\varphi.\pi'' \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$$

whereas its immediate predecesoor is $\pi \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$ is of the form

$$\pi'.\pi'' \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n;\varphi$$

Case (b): π is simple, and σ is not active. In this case there must be a predecessor of the form $\pi' \Rightarrow \sigma$ on the branch $\mathcal{B}$ where π' is simple and σ is active.

Case (c): π is simple, and σ is active. Then $\pi \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$is derived from its immediate predecessor by an application of either of the following rules:

– **$\mathbf{I}_\wedge$:** In this case $\pi \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$ is of the form

$$\pi \Rightarrow \sigma_1;\ldots;(\varphi \wedge \psi).\sigma';\ldots;\sigma_n$$

while its immediate predecessor is

$$\pi \Rightarrow \sigma_1;\ldots;\varphi.\psi.\sigma';\ldots;\sigma_n$$

– **$\mathbf{I}_\exists$:** In this case $\pi \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$ is of the form

$$\pi \Rightarrow \sigma_1;\ldots;\exists x\varphi.\sigma;\ldots;\sigma_n$$

while its immediate predecessor is

$$\pi \Rightarrow \sigma_1;\ldots;[^c/x](\varphi.\sigma');\ldots;\sigma_n;\exists x\varphi.\sigma'$$

– **$\mathbf{I}_\neg$:** In this case $\pi \Rightarrow \sigma_1;\ldots;\sigma;\ldots;\sigma_n$ is of the form

$$\pi \Rightarrow \sigma_1;\ldots;\neg\varphi;\ldots;\sigma_n$$

while its immediate predecessor is

$$\pi.\varphi \Rightarrow \sigma_1;\ldots\ldots;\sigma_n$$

– seq: This case has two subsubcases:

(a) $\pi \Rightarrow \sigma_1; \ldots; \sigma; \ldots; \sigma_n$ is either of the form

$$\pi \Rightarrow \sigma_1; \ldots; \neg\varphi.\sigma'; \ldots; \sigma_n$$

while its two immediate predecessors are

$$\pi \Rightarrow \sigma_1; \ldots; \neg\varphi; \ldots; \sigma_n$$

and

$$\pi.\neg\varphi \Rightarrow \sigma_1; \ldots; \sigma'; \ldots; \sigma_n$$

(b) $\pi \Rightarrow \sigma_1; \ldots; \sigma; \ldots; \sigma_n$ is of the form

$$\pi \Rightarrow \sigma_1; \ldots; Ra_1 \ldots a_n.\sigma'; \ldots; \sigma_n$$

while its two immediate predecessors are

$$\pi \Rightarrow \sigma_1; \ldots; Ra_1 \ldots a_n; \ldots; \sigma_n$$

and

$$\pi.Ra_1 \ldots a_n \Rightarrow \sigma_1; \ldots; \sigma'; \ldots; \sigma_n$$

Theorem 1 (Completeness)
If $\pi \models \Sigma$, then $\pi \vdash_{\mathcal{R}} \Sigma$

7 Natural Deduction

Examples.

(a) Note that $Rxy\,.\,\exists xAx\,.\,\exists yBy \models \exists x\exists yRxy$.
But how to derive

1	Rxy	premise
2	$\exists xAx$	premise
3	$\exists yBy$	premise
4	$\exists yRxy$	???

(b) Note that $\exists xAx\,.\,Bx\,.\,\exists xAx \not\models Bx$
However, $\exists xAx\,.\,Bx \models \exists xAx\,.\,Bx$
and $\exists xAx\,.\,Bx \models Bx$

(c) How to derive

1	Ac	premise
2	Bc	premise
3	$\exists xAx$	
4	Bx	

Let $\varphi_1\,.\,\ldots\,.\,\varphi_k$ be a text. A *natural deduction* from $\varphi_1\,.\,\ldots\,.\,\varphi_k$ is a sequence $S_1, S_2, \ldots$ of *steps* $S_n = \langle F_n, A_n, Q_n\rangle$ such that

(i) F_n is a formula;

(ii) $A_n \subseteq \{k \mid k \leq n\}$; if $k \in A_n$ then the formula F_k is one of the *assumptions at* n;

(iii) Q_n is a partial function assigning a natural number, called the *(quantifier) index* $Q_n(\exists x, i)$, to the i-th occurrence of the quantifier $\exists x$ in the formula F_n.

In a making a natural deduction we have to keep track at each step S_{n+1} of the index of the quantifier by which a free occurrence of the variable x in F_{n+1} is going to be bound. $C_n(x)$ is supposed to tell this. More precisely:

C_n is a partial function assigning a natural number to some of the variables as follows.

(a) If no occurrence of $\exists x$ in F_n is active after F_n, then $C_n(x) = C_m(x)$, where m is the largest number smaller than n such that $C_m(x)$ is defined and $A_m \subseteq A_n$;

(b) If the i-th occurrence of $\exists x$ in F_n is active after F_n, then $C_n(x) = Q_n(\exists x, i)$.

Usually, the largest number smaller than n such that $C_m(x)$ is defined and $A_m \subseteq A_n$ happens to be $n-1$. The only exception is the case where in S_n an assumption is withdrawn.

Another useful abbreviation is given by $M_n(\exists x)$, where $M_n(\exists x)$ denotes the maximal number in $\{l \mid l = Q_m(\exists x, i)$ for some i and some $m < n\}$

To qualify as a natural deduction the following conditions should be fulfilled:

Adding a premise For $1 \leq n \leq k$ the following must hold:

- $F_n = \varphi_n$.
- $A_n = \emptyset$.

- For every x and i such that $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = M_{n-1}(\exists x) + i$.

Each F_n is called a premise.

For $n > k$ either of the following must hold:

Making an assumption

- $A_n = A_{n-1} \cup \{n\}$.
- For every x and i such that $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = M_{n-1}(\exists x) + i$.

In this case we say that F_n is an assumption *made* at step n.

Repetition There is some $m < n$ such that

- $F_m = F_n$
- $A_m \subseteq A_{n-1} = A_n$.
- For every x such that x occurs free in F_n, $C_{n-1}(x) = C_{m-1}(x)$;
- For every x and i such that $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = Q_m(\exists x, i)$.

In this case we say that $F_n . \ldots . F_{n+l}$ is obtained by repetition.

Examples

$\exists x Ax . Bx \models Bx$;
$\exists x Ax . Bx \models \exists x Ax \wedge Bx$;
$\exists x Ax . Bx . \exists x Ax \not\models Bx$;
$\exists x Ax . Bx . \exists x Ax \models \exists x Ax \wedge Bx$.

1	$\exists^1 x Ax$	premise
2	Bx	premise
3	Bx	repetition, 2

1	$\exists^1 x Ax$	premise
2	Bx	premise
3	$\exists^1 x Ax \wedge Bx$	$\mathrm{I}_\wedge$, 1,2

1	$\exists^1 x Ax$	premise
2	Bx	premise
3	$\exists^2 x Ax$	premise
4	Bx	???

1	$\exists^1 x Ax$	premise
2	Bx	premise
3	$\exists^2 x Ax$	premise
4	$\exists^1 x Ax \wedge Bx$	$\mathrm{I}_\wedge$, 1,2

Elimination of $\exists x$ There is some $m < n$ such that

- $F_m = \exists x F_n$.
- $A_m \subseteq A_{n-1} = A_n$.
- $C_{n-1}(x) = Q_m(\exists x, 1)$;
  for every y such that $y \neq x$, and y occurs free in F_n, $C_{n-1}(y) = C_{m-1}(y)$.
- If $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = Q_m(\exists x, i+1)$.
  For every $y \neq x$ and i such that $\exists y$ occurs at least i times in F_n, $Q_n(\exists y, i) = Q_m(\exists y, i)$.

In this case we say that F_n is obtained from F_m by $\mathrm{E}_\exists$.

Introduction of $\exists x$ There exist t, $\chi_0, \chi_1 \ldots \chi_l$, and $m < n - l$ such that

- $F_m . F_{m+1} . \ldots . F_{m+l} = [^t/_x](\chi_0 . \chi_1 . \ldots . \chi_l)$
  $F_n . F_{n+1} . \ldots . F_{n+l} = \exists x \chi_0 . \chi_1 . \ldots . \chi_l$
- $A_m \subseteq A_{m+1} \subseteq \ldots \subseteq A_{m+l} \subseteq A_{n-1} = A_n = \ldots = A_{n+l}$.
- For every y such that y occurs free in $\exists x \chi_0 . \ldots . \chi_l$, $C_{n-1}(y) = C_{m-1}(y)$.
- $Q_n(\exists x, 1) = M_{n-1}(\exists x) + 1$;
  for $i > 1$, $Q_n(\exists x, i) = Q_m(\exists x, i-1)$;
  for every y such that $y \neq x$, $Q_n(\exists y, i) = Q_m(\exists y, i)$.
  For every $1 \leq j \leq l$, and every y, $Q_{n+j}(\exists y, i) = Q_{m+j}(\exists y, i)$.

In this case we say that $F_n . \ldots . F_{n+l}$ is obtained from $F_m . \ldots . F_{m+l}$ by $\mathrm{I}_\exists$

Example

$\exists x Rxx \models \exists x \exists y Rxy \wedge Ryx$

1	$\exists^1 x Rxx$	premise
2	Rxx	$\mathrm{E}_\exists$, 1
3	Rxx	repetition, 2
4	$\exists^1 y Rxy$	
5	Ryx	$\mathrm{I}_\exists$, 2–3
6	$\exists^2 x \exists^1 y Rxy$	
7	Ryx	$\mathrm{I}_\exists$, 4–5
8	$\exists^2 x \exists^1 y Rxy \wedge Ryx$	$\mathrm{I}_\wedge$, 6,7

Variable switch There exist y, x, χ, and m, l with $m < n - l$ such that

- $F_m = \exists x \chi$
  $F_n = \exists y [^y/_x](\chi)$
  $F_{n+1} . \ldots . F_{n+l} = [^y/_x, \chi](F_{m+1} . \ldots . F_{m+l})$
- $A_m \subseteq A_{m+1} \subseteq \ldots \subseteq A_{m+l} \subseteq A_{n-1} = A_n = \ldots = A_{n+l}$.
- For every z such that z occurs free in $F_n . \ldots . F_{n+l}$, $C_{n-1}(z) = C_{m-1}(z)$.
- $Q_n(\exists y, 1) = M_{n-1}(\exists y) + 1$; for $i > 1$, $Q_n(\exists y, i) = Q_m(\exists y, i-1)$;
  $Q_n(\exists x, i) = Q_m(\exists x, i+1)$;
  for every z such that $z \neq x$ and $z \neq y$, $Q_n(\exists z, i) = Q_m(\exists y, i)$.
  For every $1 \leq j \leq l$, and every z, $Q_{n+j}(\exists z, i) = Q_{m+j}(\exists z, i)$.

In this case we say that $F_n . \ldots . F_{n+l}$ is obtained from $F_m . \ldots . F_{m+l}$ by *variable switch*.

Example

$\exists x(Ax \wedge \exists x \neg Ax) \models \neg Ax$

1	$\exists^1 x(Ax \wedge \exists^2 x \neg Ax)$	premise
2	$\exists^1 y(Ay \wedge \exists^2 x \neg Ax)$	variable switch, 1
3	$Ay \wedge \exists^2 x \neg Ax$	$\mathrm{E}_\exists$, 2
4	$\exists^2 x \neg Ax$	$\mathrm{E}_\wedge$, 3
5	$\neg Ax$	$\mathrm{E}_\exists$, 4

Elimination of $\wedge$ There exist and $m < n$ such that

- $F_m = F_n \wedge F_{n+1}$.
- $A_m \subseteq A_{n-1} = A_n = A_{n+1}$.
- For every x such that x occurs free in F_m, $C_{n-1}(x) = C_{m-1}(x)$;
- For every x and i such that $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = Q_m(\exists x, i)$.
  For every x and i such that $\exists x$ occurs at least i times in F_{n+1}, $Q_{n+1}(\exists x, i) = Q_m(\exists x, i + j)$, where j is the number of times $\exists x$ occurs in F_n.

In this case we say that F_n is obtained from F_m by $\mathrm{E}_\wedge$.

Introduction of $\wedge$ There is some m with $m < n - 1$ such that

- $F_n = F_m \wedge F_{m+1}$.
- $A_m \subseteq A_{m+1} \subseteq A_{n-1} = A_n$.
- For every x such that x occurs free in F_n, $C_{n-1}(x) = C_{m-1}(x)$.
- Q_n is determined as follows:
  Suppose $\exists x$ occurs j times in F_m. Then for every $i \leq j$, $Q_n(\exists x, i) = Q_m(\exists x, i)$, and for every $i > j$, if $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = Q_{m+1}(\exists x, i - j)$.

In this case we say that F_n is obtained from F_m by $\mathrm{I}_\wedge$.

Elimination of $\bot$ There is some $m \leq n$

- $F_m = \bot$.
- $A_m \subseteq A_{n-1} = A_n$.
- For every x and i such that $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) \leq M_{n-1}(\exists x) + i$.

In this case we say that F_n is obtained from F_m by $\mathrm{E}_\bot$.

Elimination of $\neg$ There exist χ and l, m with $l \leq m \leq n$ such that

- $F_l = \neg\chi$, $F_m = \chi$, and $F_n = \bot$.
- $A_l \subseteq A_m \subseteq A_{n-1} = A_n$.
- For every x such that x occurs free in χ, $C_{l-1}(x) = C_{m-1}(x)$

In this case we say that F_n is obtained from F_m by $\mathrm{I}_\bot$.

Introduction of $\neg$ There is some $m < n$ such that

- F_m is an assumption made at step m, $F_{n-1} = \bot$, and $F_n = \neg F_m$.
- $A_m = A_{n-1}$, and $A_n = A_{n-1} \backslash \{m\}$
- For every x and i such that $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = Q_m(\exists x, i)$;

In this case we say that F_n is obtained from F_m by $\mathrm{I}_\neg$.
The assumption χ made at step m is *withdrawn* at step n.

Elimination of $\neg\neg$ There is some $m < n$ such that

- $F_m = \neg\neg F_n$.
- $A_m \subseteq A_{n-1} = A_n$.
- For every x such that x occurs free in F_n, $C_{n-1}(x) = C_{m-1}(x)$;
- For every x and i such that $\exists x$ occurs at least i times in F_n, $Q_n(\exists x, i) = Q_m(\exists x, i)$.

In this case we say that F_n is obtained from F_m by $\mathrm{E}_{\neg\neg}$.

Example
$\neg(\exists x Ax \wedge \neg Bx) . \exists x Ax \models Bx$

1	$\neg(\exists^1 x Ax \wedge \neg Bx)$	premise
2	$\exists^2 x Ax$	premise
3	$\neg Bx$	assumption
4	$\exists^2 x Ax \wedge \neg Bx$	$\mathrm{I}_\wedge$
5	$\bot$	$\mathrm{E}_\neg$, 1, 4
6	$\neg\neg Bx$	$\mathrm{I}_\neg$
7	Bx	$\neg\neg$, 6

Definition 16
Consider a natural deduction. We define T_n, also called the *text at* the n-th step, to be the sequence of formulas $\chi_1 . \chi_2 . \cdots$ determined as follows:

(i) $\chi_1 = F_i$ for the smallest $i \leq n$ such that $A_i \subseteq A_n$;
(ii) If $\chi_k = F_j$, then $\chi_{k+1} = F_i$ for the smallest i such that $j \leq i \leq n$ and $A_i \subseteq A_n$.

Definition 17
$\varphi_1 . \ldots . \varphi_k \vdash_{\mathcal{N}} \psi_1 . \ldots . \psi_l$ iff there exists a natural deduction from $\varphi_1 . \ldots . \varphi_k$ such that for some n the following holds:

(i) $F_n . \ldots . F_{n+l} = \psi_1 . \ldots . \psi_l$;
(ii) for each i such that $n \leq i \leq n + l$, $A_i = \emptyset$;
(iii) for each x occurring free in $F_n . \ldots . F_{n+l}$, $C_n(x) = C_k(x)$.

Definition 18
Consider a natural deduction from $\varphi_1 . \cdots \varphi_k$.
Consider $T_n = \varphi_1 . \cdots \varphi_k . \chi_1 . \ldots . \chi_m$ for some n such that $k \leq n$.
Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ be some model and suppose there are assignments $v_0, \ldots, v_m$ pertaining to $\mathcal{M}$ such that

(i) $v_{i+1} \in \{v_i\}[\chi_{i+1}]$ for every $0 \leq i < m$;
(ii) for all x and all $0 \leq i, j \leq m$ such that $C_i(x) = C_j(x)$ it holds that $v_i(x) = v_j(x)$.

In this case we say that the sequence $v_0, \ldots, v_m$ *follows* the deduction up to n.

Lemma 6
Consider a natural deduction from $\varphi_1 . \ldots . \varphi_k$.
Consider $T_k = \varphi_1 . \ldots . \varphi_k$.
Let $\mathcal{M}$ be any model $\langle \mathcal{D}, \mathcal{I} \rangle$.
Any sequence of assignments $v_0, \ldots, v_k$ pertaining to $\mathcal{M}$ which has the property mentioned under (i) above follows the deduction up to k.

Lemma 7
Consider a natural deduction from $\varphi_1 . \ldots . \varphi_k$.
Consider $T_n = \varphi_1 . \ldots . \varphi_m$ for some n such that $k \leq n$, and assume F_{n+1} is not an assumption.
Let $\mathcal{M}$ be any model.
Then any sequence $v_0, \ldots, v_m$ pertaining to $\mathcal{M}$ which follows the deduction up to n can be extended to a sequence $v_0, \ldots, v_m, v_{m+1}$ which follows the deduction op to $n + 1$.

Theorem 2 (Soundness)
If $\pi \vdash_{\mathcal{N}} \sigma$, then $\pi \models \sigma$

8 Sequent system 'Utrecht'

$\dfrac{\pi; [^c/_x]\varphi; [^c/_x]\pi' \Rightarrow [^c/_x]\sigma}{\pi; \exists x; \varphi; \pi' \Rightarrow \sigma}\ \mathrm{E}_\exists$	Provided c does not occur in π, φ, π', or σ
$\dfrac{\pi \Rightarrow \sigma; [^t/_x]\varphi; [^t/_x]\sigma'}{\pi \Rightarrow \sigma; \exists x; \varphi; \sigma'}\ \mathrm{I}_\exists$	Provided t is free for x

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg\varphi; \varphi; \psi \Rightarrow}{\neg\varphi; \varphi \wedge \psi \Rightarrow}\ \text{seq}}{\neg\varphi \Rightarrow \neg(\varphi \wedge \psi)}\ \mathrm{I}_\neg}{\neg\neg(\varphi \wedge \psi); \neg\varphi \Rightarrow}\ \mathrm{E}_\neg}{\neg\neg(\varphi \wedge \psi) \Rightarrow \neg\neg\varphi}\ \mathrm{I}_\neg \qquad \neg\neg(\varphi \wedge \psi); \neg\neg\varphi \Rightarrow \varphi}{\neg\neg(\varphi \wedge \psi) \Rightarrow \varphi}\ \text{c-cut}$$

However,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg\psi; \varphi; \psi \Rightarrow (??)}{\neg\psi; \varphi \wedge \psi \Rightarrow}\ \text{seq}}{\neg\psi \Rightarrow \neg(\varphi \wedge \psi)}\ \mathrm{I}_\neg}{\neg\neg(\varphi \wedge \psi); \neg\psi \Rightarrow}\ \mathrm{E}_\neg}{\neg\neg(\varphi \wedge \psi) \Rightarrow \neg\neg\psi}\ \mathrm{I}_\neg \qquad \neg\neg(\varphi \wedge \psi); \neg\neg\psi \Rightarrow \psi}{\neg\neg(\varphi \wedge \psi) \Rightarrow \psi}\ \text{c-cut}$$

Meaning To know the meaning of a sentence is to know the change it brings about in the information state of anyone who accepts the news conveyed by it. In other words, the meaning of a sentence is a function from information states into information states.

Notation Let s be an information state and φ a sentence with meaning $[\varphi]$. We write $s[\varphi]$ for the information state that results when s is updated with φ.

Acceptance Sometimes the information conveyed by φ will already be subsumed by s. In such a case, we say that φ is *accepted* in s, or that s *supports* φ, and we write this as:

$$s \models \varphi$$

Logical validity Updating any state s with the premises $[\varphi_1], \ldots, [\varphi_n]$ in that order, yields a state in which the conclusion ψ is to be accepted.

$$\varphi_1, \ldots, \varphi_n \models \psi \text{ iff for any } s, s[\varphi_1], \ldots, [\varphi_n] \models \psi$$

More examples

- $\exists x Px$. `might` $\neg Px$ is inconsistent.
- $\exists x Px$. `might` $\forall y \neg Py$ is inconsistent.
- $\exists x Px . \forall y$ `might` $\neg Py$ is consistent.
- $\exists x Px . \forall y$ `might` $(y \neq x)$ is consistent.
- $\exists x(Px \wedge \forall y$ `might` $(y \neq x))$ is inconsistent.

Definition 19 (Support) A state s supports a text π iff $\{i\}[\pi] \neq \emptyset$ for every $i \in s$.

Proposition 6 (Soundness Deductive Tableaus) If $\pi \vdash_{\mathcal{D}} \sigma$, then $\pi \models \sigma$

Proof: Check the following:

(i) If $Rt_1 \dots t_1$ does not depend on π', then $\pi . Rt_1 \dots t_n . \pi' \models Rt_1 \dots t_n$.
(ii) If $\pi . \varphi . \psi . \pi' \models \sigma$, then $\pi . \varphi \wedge \psi . \pi' \models \sigma$.
(iii) If $\pi \models \sigma . \varphi . \psi . \sigma'$, then $\pi \models \sigma . \varphi \wedge \psi . \sigma'$.
(iv) If $\pi . [^c/_x](\varphi . \pi') \models [^c/_x, \varphi . \pi'](\sigma)$, and c does not occur in π, φ, π', or σ, then $\pi . \exists x \varphi . \pi' \models \sigma$.
(v) If $\pi \models \sigma . [^t/_x](\varphi . \sigma')$, and t is free for x in $\varphi . \sigma'$, then $\pi \models \sigma . \exists x \varphi . \sigma'$.
(vi) If $\pi . \pi' \models \varphi$, and φ does not depend on π', then $\pi . \neg \varphi . \pi' \models \sigma$.
(vii) If $\pi . \varphi \models \emptyset$, then $\pi \models \neg \varphi$.
(viii) If $\pi . \neg \varphi \models \emptyset$, then $\pi \models \varphi$.
(ix) If $\pi \models \sigma$ and $\pi . \sigma \models \sigma'$, then $\pi \models \sigma . \sigma'$.

Elimination rule for $\wedge$

$\vdots$	π'	
m	$\varphi \wedge \psi$	
$\vdots$	π	
n	φ	$E_\wedge, m$
$n+1$	ψ	$E_\wedge, m$

Conditions:
1. φ does not depend on $\varphi \wedge \psi . \tau(\pi)$;
2. ψ does not depend on $\psi . \tau(\pi)$.

Introduction rule for $\wedge$

$\vdots$	π'	
m	φ	
$m+1$	ψ	
$\vdots$	π	
n	$\varphi \wedge \psi$	$I_\wedge, m, m+1$

Conditions:
1. φ does not depend on $\psi . \tau(\pi)$;
2. ψ does not depend on $\tau(\pi)$.

Elimination rule for $\neg$

$\vdots$	π''	
k	φ	
$\vdots$	π'	
m	$\neg \varphi$	
$\vdots$	π	
n	$\bot$	$E_\neg, k, m$

Conditions:

1. $\neg\varphi$ does not depend on $\varphi\,.\,\tau(\pi')$

Introduction rule for $\neg$

$$\begin{array}{|ll|ll}
\hline
\vdots & \pi & & \\
m & \varphi & \text{assumption} & \\
\vdots & \pi' & & \\
n-1 & \bot & & \\
\hline
n & \neg\varphi & & I_\neg
\end{array}$$

Conditions:

1. —

Introduction rule for $\exists$

$$\begin{array}{lll}
\vdots & \pi' & \\
m & [^t/_x]\varphi & \\
\vdots & \pi & \\
n & \exists x\varphi & I_\exists, m
\end{array}$$

Conditions:

1. t is free for x in φ;
2. $\exists x\varphi$ does not depend on $[t/x]\varphi\,.\,\tau(\psi)$.

Elimination rule for $\exists$

$$\begin{array}{lll}
\vdots & \pi' & \\
m & \exists x\varphi & \\
\vdots & \pi & \\
n & \varphi & E_\exists, m
\end{array}$$

Conditions:

1. φ does not depend on $\varphi\,.\,\tau(\pi)$.

$\neg\neg$-rule

$$\begin{array}{lll}
\vdots & \pi' & \\
m & \neg\neg\varphi & \\
\vdots & \pi & \\
n & \varphi & \neg\neg, m
\end{array}$$

Conditions:

1. φ does not depend on $\tau(\pi)$.

Recapitulation rule

$$\begin{array}{lll}
\vdots & \pi' & \\
m & \exists x\varphi & \\
\vdots & \pi & \\
n & \exists y[y/x]\varphi & R_\exists, m
\end{array}$$

Conditions:

1. y is free for x in φ;
2. $\exists y[y/x]\varphi$ does not depend on $\varphi.\tau(\pi)$.

Proof:

(a) and (b) are left to the reader.

(c)

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi.\neg\varphi \Rightarrow \chi}{\pi.\neg\varphi.\neg\chi \Rightarrow \bot}\,\mathrm{E}_\neg}{\pi.\neg\chi \Rightarrow \neg\neg\varphi}\,\mathrm{I}_\neg \qquad \pi.\neg\chi.\neg\neg\varphi \Rightarrow \varphi}{\pi.\neg\chi \Rightarrow \varphi} \qquad \dfrac{\dfrac{\dfrac{\pi.\varphi.\neg\psi \Rightarrow \chi}{\pi.\varphi.\neg\psi.\neg\chi \Rightarrow \bot}\,\mathrm{E}_\neg}{\pi.\varphi.\neg\chi \Rightarrow \neg\neg\psi}\,\mathrm{I}_\neg \qquad \pi.\varphi.\neg\chi.\neg\neg\psi \Rightarrow \psi}{\pi.\varphi.\neg\chi \Rightarrow \psi}\,\mathrm{seq}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi.\neg\chi \Rightarrow \varphi.\psi}{\pi.\neg\chi \Rightarrow \varphi \wedge \psi}\,\mathrm{I}_\wedge}{\pi.\neg\chi.\neg(\varphi\wedge\psi) \Rightarrow \bot}\,\mathrm{E}_\neg}{\pi.\neg(\varphi\wedge\psi) \Rightarrow \neg\neg\chi}\,\mathrm{I}_\neg \qquad \pi.\neg(\varphi\wedge\psi).\neg\neg\chi}{\pi.\neg(\varphi\wedge\psi) \Rightarrow \chi}$$

(d)

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi.\neg[^c/_x]\varphi \Rightarrow \psi}{\pi.\neg\psi.\neg[^c/_x]\varphi \Rightarrow \bot}\,\mathrm{E}_\neg}{\pi.\neg\psi \Rightarrow \neg\neg[^c/_x]\varphi}\,\mathrm{I}_\neg \qquad \pi.\neg\psi.\neg\neg[^c/_x]\varphi \Rightarrow [^c/_x]\varphi}{\dfrac{\pi.\neg\psi \Rightarrow [^c/_x]\varphi}{\pi.\neg\psi \Rightarrow \exists x\varphi}\,\mathrm{I}_\exists}\,\mathrm{E}_\neg}{\pi.\neg\psi.\neg\exists x\varphi \Rightarrow \bot}\,\mathrm{I}_\neg \qquad \pi.\neg\exists x\varphi.\neg\neg\psi \Rightarrow \psi$$

$$\pi.\neg\exists x\varphi \Rightarrow \psi$$

Example

	1	$\neg\neg\exists x(Ax \wedge Bx)$	premise
┌	2	$\neg(\exists xAx \wedge Bx)$	assumption
│ ┌	3	$\exists x(Ax \wedge Bx)$	assumption
│ │	4	$Ax \wedge Bx$	$\mathrm{E}_\exists$, 3
│ │	5	Ax	$\mathrm{E}_\wedge$, 4
│ │	6	Bx	$\mathrm{E}_\wedge$, 4
│ │	7	$\exists xAx$	$\mathrm{I}_\exists$, 5
│ │	8	Bx	$\mathrm{I}_\exists$, 6,7
│ │	9	$\exists xAx \wedge Bx$	$\mathrm{I}_\wedge$, 7, 8
│ └	10	$\bot$	$\mathrm{E}_\neg$, 2,9
│	11	$\neg\exists x(Ax \wedge Bx)$	$\mathrm{I}_\neg$
└	12	$\bot$	$\mathrm{E}_\neg$, 1,11
	13	$\neg\neg(\exists xAx \wedge Bx)$	$\mathrm{I}_\neg$
	14	$\exists xAx \wedge Bx$	$\neg\neg$, 13
	15	$\exists xAx$	$\mathrm{E}_\wedge$, 14
	16	Bx	$\mathrm{E}_\wedge$, 14,15