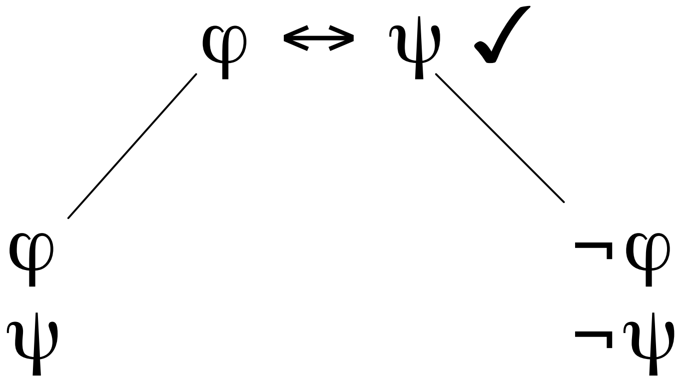
Tableau and Sequent Calculus for Predicate Logic

Robert Keller
February 2010

Aside: Additional Tree Rules: $\varphi \leftrightarrow \psi$

- This formula is retired and the tree **branches** with the formulas and their negations stacked:



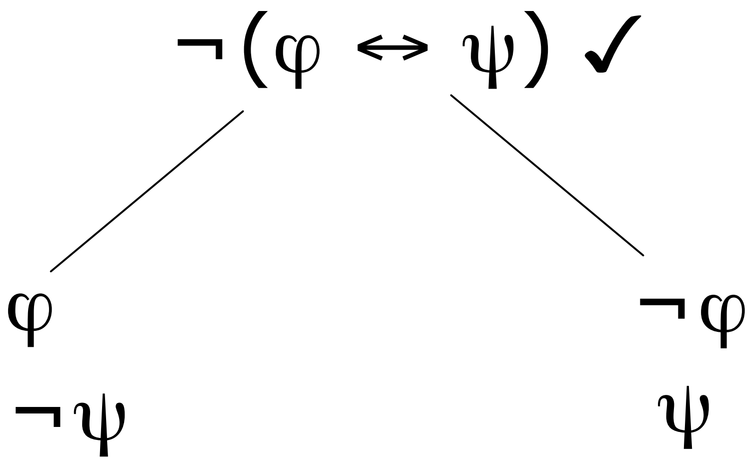
$\varphi \leftrightarrow \psi$ ✓

φ
ψ

$\neg\varphi$
$\neg\psi$

Additional Tree Rules: $\neg(\varphi \leftrightarrow \psi)$

- This formula is retired and the tree **branches**, with φ negated:



$\neg(\varphi \leftrightarrow \psi)$ ✓

φ

$\neg\psi$

$\neg\varphi$

ψ

Propositional Tree Rule Summary

stack	$\varphi \wedge \psi$ φ ψ	$\neg(\varphi \vee \psi)$ $\neg\varphi$ $\neg\psi$	$\neg(\varphi \rightarrow \psi)$ φ $\neg\psi$		$\neg\neg\varphi$ φ
split	$\varphi \vee \psi$ $\varphi \qquad \psi$	$\neg(\varphi \wedge \psi)$ $\neg\varphi \qquad \neg\psi$	$\varphi \rightarrow \psi$ $\neg\varphi \qquad \psi$	$(\varphi \leftrightarrow \psi)$ $\varphi \qquad \neg\varphi$ $\psi \qquad \neg\psi$	
				$\neg(\varphi \leftrightarrow \psi)$ $\varphi \qquad \neg\varphi$ $\neg\psi \qquad \psi$	

Quantifier Rules for Tableaux

¬∃ rule

$\neg \exists \nu \ \varphi$ ✓

$\forall \nu \ \neg \varphi$

$\neg\forall$ rule

$\neg\forall\nu\ \varphi$ ✓

$\exists\nu\ \neg\varphi$

∃ rule

$\exists\nu\ \varphi$ ✓

$\varphi[c/\nu]$

where c is a **new** constant not appearing in the tree.

The rule can be used only once per ∃ formula.

∀ rule

$\forall\nu\ \varphi$ **Does not get a check!!**

$\varphi[\tau/\nu]$

where τ is any term free to replace ν in φ.

This rule can used arbitrarily-many times for a ∀ formula.

Choice of Terms for ∀ rule

- Choose terms constructed of constants introduced by the ∃ rule

Example (already negated)

$$\neg(\forall x\ p(x) \rightarrow \exists x\ p(x))$$

Example

$\neg(\forall x\ p(x) \rightarrow \exists x\ p(x))$ ✓

$\forall x\ p(x)$

$\neg\exists x\ p(x)$

Example

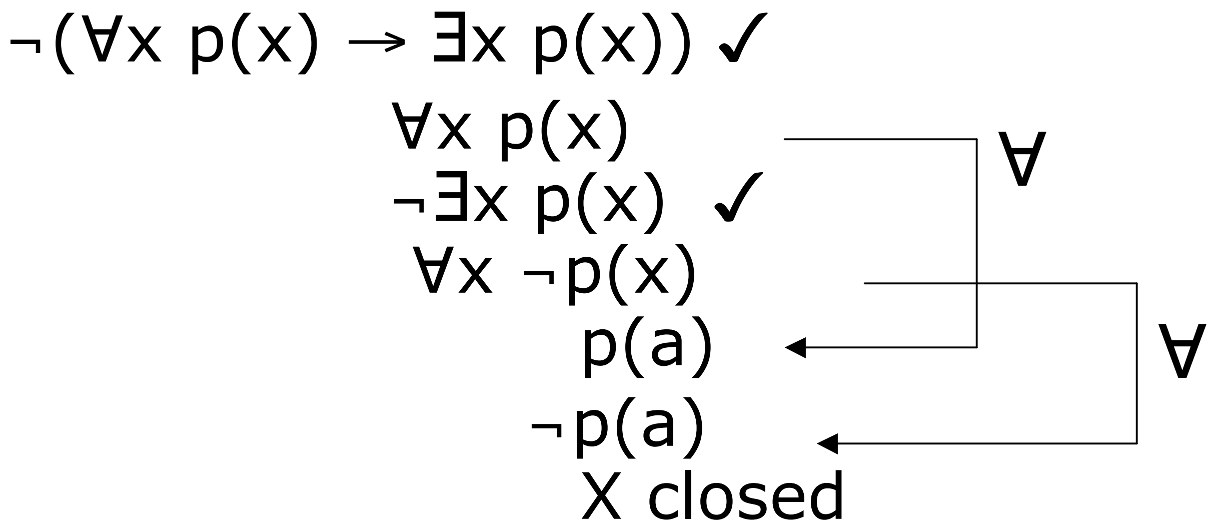
$\neg(\forall x\ p(x) \rightarrow \exists x\ p(x))$ ✓

$\forall x\ p(x)$

$\neg\exists x\ p(x)$ ✓

$\forall x\ \neg p(x)$

Example



$\neg(\forall x\, p(x) \rightarrow \exists x\, p(x))$ ✓

$\forall x\, p(x)$

$\neg\exists x\, p(x)$ ✓

$\forall x\, \neg p(x)$

$p(a)$

$\neg p(a)$

X closed

The root formula is not satisfiable.

Thus $\forall x\, p(x) \rightarrow \exists x\, p(x)$ is valid

Closure depended on appropriate choice of term to substitute for x in $\forall x\, \neg p(x)$.

Example

(in which ∀ rule is used twice from the same line)

$\forall x\ (\exists y\ P(x,y) \rightarrow \forall z\ P(z, x)),\ P(a,a)\ \vdash P(a, b)$

This time we number for better clarity.

1. $\forall x\ (\exists y\ P(x,y) \rightarrow \forall z\ P(z, x))$		premise
2. $P(a,a)$		premise
3. $\neg P(a, b)$		negated conclusion
4. $(\exists y\ P(a,y) \rightarrow \forall z\ P(z, a))$ √		1 with a for x
5. $(\exists y\ P(b,y) \rightarrow \forall z\ P(z, b))$		1 with b for x
6. $\neg\exists y\ P(a,y)$ √	$\forall z\ P(z, a)$	4
7. $\forall y \neg P(a, y)$	(cont'd next pg)	6
8. $\neg P(a, a)$		7

X closes (2, 8)

Example continued

1. ∀x (∃y P(x,y) → ∀z P(z, x))	premise
2. P(a,a)	premise
3. ¬P(a, b)	negated conclusion
4. (∃y P(a,y) → ∀z P(z, a)) √	1 with a for x
5. (∃y P(b,y) → ∀z P(z, b)) √	1 with b for x
6. ∀z P(z, a)	4

9. ¬∃y P(b,y) √	5	13. ∀z P(z, b)	5
10. ∀y¬P(b,y)	9	14. P(a, b)	13
11. ¬P(b,a)	11	X closes (3, 14)	
12. P(b, a)	10		
X closes (11, 12)			

Termination

- Unlike the propositional case, the predicate version of tableaux does not necessarily terminate. This is because the ∀ rule can be used arbitrarily-many times.

- It can be shown, however, that **if** the root formula is unsatisfiable, then **there exists** a closed tree for it.
  (If the root formula is satisfiable, the construction *might* not terminate.)

Example of Non-Termination

$$\forall x\, \exists y\, P(x, y) \vdash P(a, a)$$

1.	$\forall x\, \exists y\, P(x, y)$	premise
2.	$\neg P(a, a)$	conclusion negated
3.	$\exists y\, P(a, y)$ √	1, a for x
4.	$P(a, b)$	3, b for y
5.	$\exists y\, P(b, y)$ √	1, b for x
6.	$P(b, c)$	5, c for y
7.	$\exists y\, P(c, y)$ √	1, c for x
8.	$P(c, d)$	7, d for y
9.	$\exists y\, P(d, y)$ √	1, d for x
	. . .	. . .

Using the Tableau Method to Find a Model in the Predicate Calculus

- $\forall x\, (A(x) \vee B(x)) \rightarrow (\forall x\, A(x) \vee \forall x\, B(x))$
- This formula is not valid, so its negation should be satisfiable.

$\neg(\forall x\, (A(x) \vee B(x)) \rightarrow (\forall x\, A(x) \vee \forall x\, B(x)))$ ✓

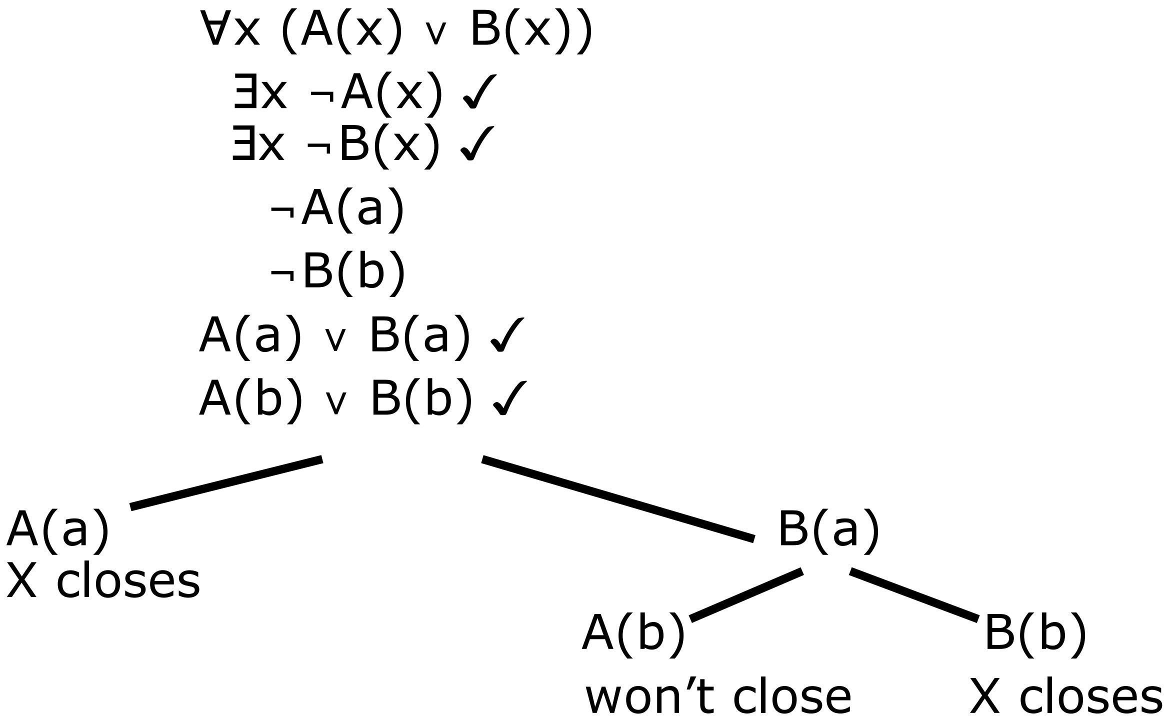
$\forall x\, (A(x) \vee B(x))$

$\neg(\forall x\, A(x) \vee \forall x\, B(x))$ ✓

$\neg\forall x\, A(x)$ ✓
$\neg\forall x\, B(x)$ ✓

$\exists x\, \neg A(x)$
$\exists x\, \neg B(x)$

Using the Tableau Method to Find a Model in the Predicate Calculus



∀x (A(x) ∨ B(x))

∃x ¬A(x) ✓

∃x ¬B(x) ✓

¬A(a)

¬B(b)

A(a) ∨ B(a) ✓

A(b) ∨ B(b) ✓

A(a)
X closes

B(a)

A(b)
won’t close

B(b)
X closes

Conclusion: **There is a model for the negation** with domain {a, b}, in which “¬A(a), A(b), B(a), and ¬B(b)” (translated into the appropriate interpretation notation).

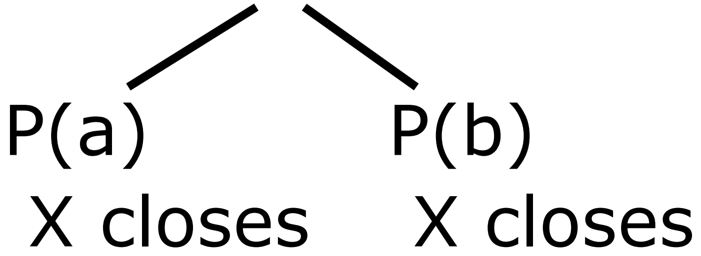
Handling Equality in Tableaux

- If an open path has a node $t_1 = t_2$, then for any unchecked node containing t_1, add on the path a formula in which t_1 is replaced with t_2 (and vice-versa).

Example: Equality in Tableau

- $P(a) \vee P(b), \neg P(a) \vdash \neg(a=b)$
- Proof:

1. $P(a) \vee P(b)$	premise
2. $\neg P(a)$	premise
3. $\neg\neg(a=b)$ ✓	negated conclusion
4. $a=b$	3, $\neg\neg$
5. $\neg P(b)$	2, 4, = rule



$P(a)$	$P(b)$	1
X closes	X closes	

Example: Equality in Tableau

- $a = b \vdash P(a, b) \rightarrow P(b, a)$
- Proof:

1. $a = b$	premise
2. $\neg(P(a, b) \rightarrow P(b, a))$ ✓	negated conclusion
3. $P(a, b)$	2
4. $\neg P(b, a)$	2
5. $P(a, a)$	3, 1, = rule
6. $\neg P(a, a)$	4, 1, = rule
X closes(5, 6)	

Sequent Calculus for Predicates

- As with the tableau method, the propositional rules for Sequent Calculus will be augmented with four new rules for quantifiers.

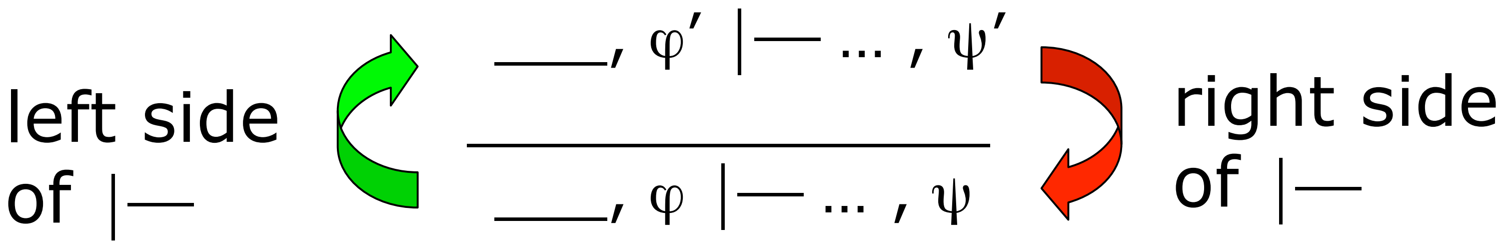
- As before, the Sequent Calculus rules have a correspondence with the tableau proof rules.

- Whereas the tableau shows negation explicitly, in Sequent Calculus it may be implicit, depending on which side of the turnstile a formula appears.

- Negated formulas in tableaux generally correspond to formulas on the right of the turnstile in Sequent Calculus.

Mindset for Remembering Sequent Calculus

- Think of the sequent(s) above the line as being **sufficient** to prove the one below.
- Think of the “information flow” as shown in the diagram.



left side of |—

$$\frac{\_\_\_, \varphi' \;|\!\!— \ldots , \psi'}{\_\_\_, \varphi \;|\!\!— \ldots , \psi}$$

right side of |—

∀|— rule

$$\frac{\varphi[t/x],\ \_\_\_\_\ \vdash \ldots}{\forall x\ \varphi,\ \_\_\_\_\ \vdash \ldots}\ \forall\vdash$$

where t is any term.

This rule parallels the **∀-Elimination** rule of natural deduction.

It says that … can be proved from $\forall x\varphi$, ___ provided that it can be proved from $\varphi[t/x]$, ___.

∃|— rule

$$\frac{\varphi[x_0/x],\ \_\_\_\_\ \mid\!\!\text{—}\ \ldots}{\exists x\ \varphi,\ \_\_\_\_\ \mid\!\!\text{—}\ \ldots}\ \exists\mid\!\!\text{—}$$

where x_0 is a fresh variable not occurring in … .

This rule parallels the **∃-Elimination** rule of natural deduction.

It says that … can be proved from ∃ xφ, ___ provided that it can be proved from $\varphi[x_0/x]$, ___.

$\vdash \forall$ rule

$$\frac{\_\_\_\_ \vdash \ldots, \varphi[x_0/x]}{\_\_\_\_ \vdash \ldots, \forall x\, \varphi} \vdash \forall$$

where x_0 is a fresh variable not occurring in … .

This rule parallels the **∀-Introduction** rule of natural deduction.

It states that to prove $\forall x \varphi$ it suffices to prove $\varphi[x_0/x]$ where x_0 is an an arbitrary fresh variable.

|— ∃ rule

$$\frac{\_\_\_\_ \;\vdash \ldots, \varphi[t/x]}{\_\_\_\_ \;\vdash \ldots, \exists x\ \varphi}\ \vdash \exists$$

where t is an term free for x in φ.

This rule parallels the **∃-Introduction rule** of natural deduction.

It states that to prove ∃xφ it suffices to prove φ[t/x] where t is any term.

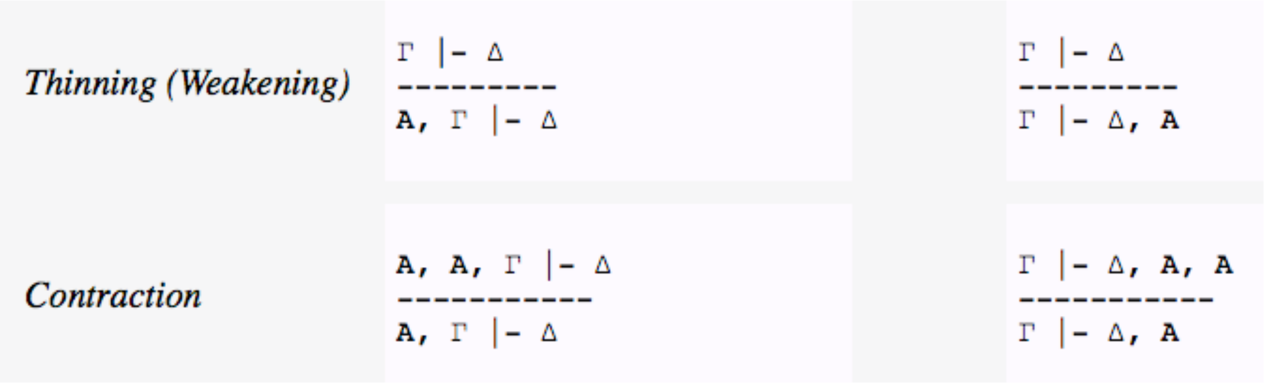
Sequent Calculus Quantifier Rule Summary

	Left	Right
∃	$\dfrac{\varphi[x_0/x],\ \_\_\_\_ \ \vdash \ldots}{\exists x\ \varphi,\ \_\_\_\_ \ \vdash \ldots}\ \exists\vdash$	$\dfrac{\_\_\_\_ \ \vdash \ldots,\ \varphi[t/x]}{\_\_\_\_ \ \vdash \ldots,\ \exists x\ \varphi}\ \vdash\exists$
∀	$\dfrac{\varphi[t/x],\ \_\_\_\_ \ \vdash \ldots}{\forall x\ \varphi,\ \_\_\_\_ \ \vdash \ldots}\ \forall\vdash$	$\dfrac{\_\_\_\_ \ \vdash \ldots,\ \varphi[x_0/x]}{\_\_\_\_ \ \vdash \ldots,\ \forall x\ \varphi}\ \vdash\forall$

Variations

- See the tutorial by Alexander Sakharov: http://sakharov.net/sequent.html

- In particular, some versions uses the ability to “thin” and “contract” a set of formulas.
- We may use these on occasion.



Thinning (Weakening)	Γ |- Δ ---------- A, Γ |- Δ	Γ |- Δ ---------- Γ |- Δ, A
Contraction	A, A, Γ |- Δ ------------ A, Γ |- Δ	Γ |- Δ, A, A ------------ Γ |- Δ, A

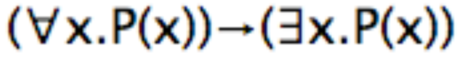
Rule Correspondence

Tableau Rule	Sequent Calculus Rule (going backward/upward)
$\neg\exists$ replace with $\forall\neg$	$\vdash \exists$ specializes term on right
$\neg\forall$ replace with $\exists\neg$	$\vdash \forall$ specializes term on right to fresh var
$\exists$ introduces constant	$\exists \vdash$ specializes term on left to fresh var
$\forall$ uses term	$\forall \vdash$ specializes term on left

Sequent Calculus Examples

∀x p(x) → ∃x p(x) revisited

using JAPE



$(\forall x.P(x))\rightarrow(\exists x.P(x))$

Sequent Calculus Examples

$\vdash \rightarrow$ rule

$$\begin{array}{c} \forall x.P(x) \vdash \exists x.P(x) \\ \hline \vdash\rightarrow \\ (\forall x.P(x))\rightarrow(\exists x.P(x)) \end{array}$$

Sequent Calculus Examples

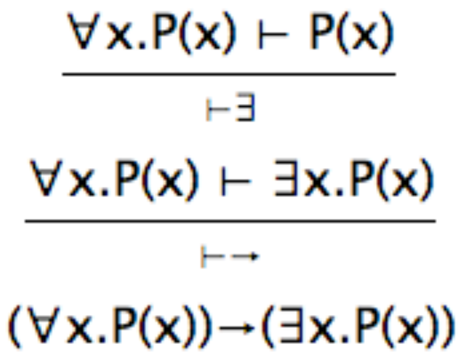
- |— ∃ rule (introduces term _B for x)

$$\begin{array}{c} \forall x.P(x) \vdash P(\_B) \\ \hline \vdash\exists \\ \forall x.P(x) \vdash \exists x.P(x) \\ \hline \vdash\to \\ (\forall x.P(x))\to(\exists x.P(x)) \end{array}$$

Sequent Calculus Examples

unifying _B with term x (all that is available!?)

Note this x is distinct from the one in ∀x.P(x).



$$\dfrac{\dfrac{\forall x.P(x) \vdash P(x)}{\overset{\vdash\exists}{\forall x.P(x) \vdash \exists x.P(x)}}}{\overset{\vdash\to}{(\forall x.P(x))\to(\exists x.P(x))}}$$

Sequent Calculus Examples

∀|— rule

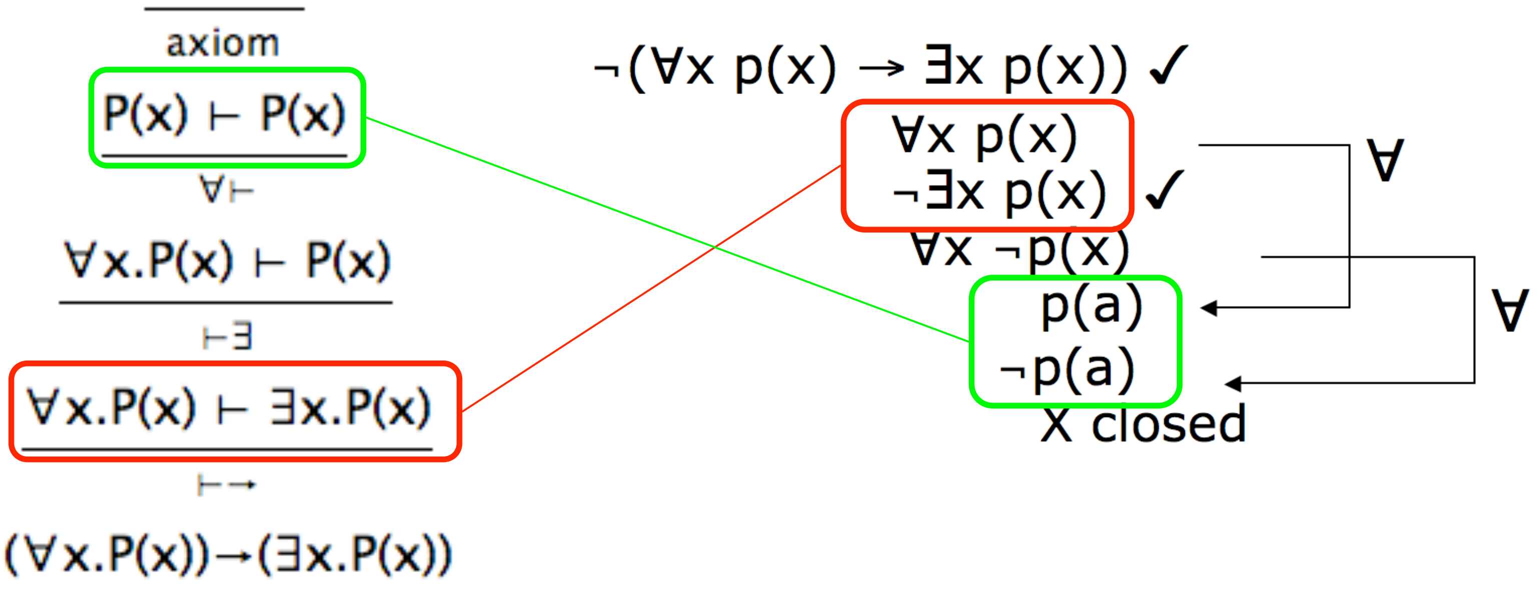
$$\begin{array}{c}
\underline{P(\_B) \vdash P(x)} \\
\forall\vdash \\
\underline{\forall x.P(x) \vdash P(x)} \\
\vdash\exists \\
\underline{\forall x.P(x) \vdash \exists x.P(x)} \\
\vdash\rightarrow \\
(\forall x.P(x)) \rightarrow (\exists x.P(x))
\end{array}$$

Sequent Calculus Examples

Unifying _B with x gives an axiom

$$\begin{array}{c}
\hline
\text{axiom} \\
P(x) \vdash P(x) \\
\hline
\forall\vdash \\
\forall x.P(x) \vdash P(x) \\
\hline
\vdash\exists \\
\forall x.P(x) \vdash \exists x.P(x) \\
\hline
\vdash\rightarrow \\
(\forall x.P(x)) \rightarrow (\exists x.P(x))
\end{array}$$

Compare with Tableau



axiom

$P(x) \vdash P(x)$

$\forall\vdash$

$\forall x.P(x) \vdash P(x)$

$\vdash\exists$

$\forall x.P(x) \vdash \exists x.P(x)$

$\vdash\rightarrow$

$(\forall x.P(x))\rightarrow(\exists x.P(x))$

$\neg(\forall x\ p(x) \rightarrow \exists x\ p(x))$ ✓

$\forall x\ p(x)$

$\neg\exists x\ p(x)$ ✓

$\forall x\ \neg p(x)$

$p(a)$

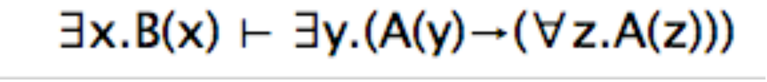
$\neg p(a)$

X closed

As before, could use block tableau to make the connection clearer.

Another Sequent Proof Example (Here contraction will be used.)

The LHS is just a way to get and individual.
(This theory in JAPE does not have a “top” T.)



$\exists x.B(x) \vdash \exists y.(A(y) \rightarrow (\forall z.A(z)))$

Another Sequent Proof Example

m is a fresh variable
(essentially this is ∃ elim.)

$$\frac{B(m) \vdash \exists y.(A(y) \rightarrow (\forall z.A(z)))}{\exists x.B(x) \vdash \exists y.(A(y) \rightarrow (\forall z.A(z)))}\ \exists\vdash$$

Another Sequent Proof Example

(Here contraction is used to keep a copy of a formula in the RHS for later.

This observation was a result of failing to complete the proof without the copy.)

$$\dfrac{\dfrac{B(m) \vdash \begin{array}{l}\exists y.(A(y)\to(\forall z.A(z))),\\ \exists y.(A(y)\to(\forall z.A(z)))\end{array}}{B(m) \vdash \exists y.(A(y)\to(\forall z.A(z)))}\ \vdash\text{contract}}{\exists x.B(x) \vdash \exists y.(A(y)\to(\forall z.A(z)))}\ \exists\vdash$$

Another Sequent Proof Example

Need a term for _B1.
(Essentially this is ∃ intro.)

$$\begin{array}{c}
B(m) \vdash \begin{array}{l} A(\_B1)\rightarrow(\forall z.A(z)), \\ \exists y.(A(y)\rightarrow(\forall z.A(z))) \end{array} \\
\hline
\vdash\exists \\
B(m) \vdash \begin{array}{l} \exists y.(A(y)\rightarrow(\forall z.A(z))), \\ \exists y.(A(y)\rightarrow(\forall z.A(z))) \end{array} \\
\hline
\vdash\text{contract} \\
B(m) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z))) \\
\hline
\exists\vdash \\
\exists x.B(x) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z)))
\end{array}$$

Another Sequent Proof Example

Unify _B with m:

$$\dfrac{\dfrac{\dfrac{B(m) \vdash \begin{array}{l} A(m)\to(\forall z.A(z)), \\ \exists y.(A(y)\to(\forall z.A(z))) \end{array}}{B(m) \vdash \begin{array}{l} \exists y.(A(y)\to(\forall z.A(z))), \\ \exists y.(A(y)\to(\forall z.A(z))) \end{array}}\ {\vdash}\exists}{B(m) \vdash \exists y.(A(y)\to(\forall z.A(z)))}\ {\vdash}\text{contract}}{\exists x.B(x) \vdash \exists y.(A(y)\to(\forall z.A(z)))}\ \exists{\vdash}$$

Another Sequent Proof Example

(Essentially want ∀ intro.)
Need a fresh var for z.

$$
\begin{array}{c}
\begin{matrix} B(m), \\ A(m) \end{matrix} \vdash \begin{matrix} \forall z.A(z), \\ \exists y.(A(y)\to(\forall z.A(z))) \end{matrix} \\ \hline
{\scriptstyle \vdash\to} \\
B(m) \vdash \begin{matrix} A(m)\to(\forall z.A(z)), \\ \exists y.(A(y)\to(\forall z.A(z))) \end{matrix} \\ \hline
{\scriptstyle \vdash\exists} \\
B(m) \vdash \begin{matrix} \exists y.(A(y)\to(\forall z.A(z))), \\ \exists y.(A(y)\to(\forall z.A(z))) \end{matrix} \\ \hline
{\scriptstyle \vdash\text{contract}} \\
B(m) \vdash \exists y.(A(y)\to(\forall z.A(z))) \\ \hline
{\scriptstyle \exists\vdash} \\
\exists x.B(x) \vdash \exists y.(A(y)\to(\forall z.A(z)))
\end{array}
$$

Another Sequent Proof Example

m1 is a fresh var for z.
Now use ∃ intro.

$$
\begin{array}{c}
B(m), A(m) \vdash A(m1), \exists y.(A(y) \to (\forall z.A(z))) \\
\hline
\vdash\forall \\
B(m), A(m) \vdash \forall z.A(z), \exists y.(A(y) \to (\forall z.A(z))) \\
\hline
\vdash\to \\
B(m) \vdash A(m) \to (\forall z.A(z)), \exists y.(A(y) \to (\forall z.A(z))) \\
\hline
\vdash\exists \\
B(m) \vdash \exists y.(A(y) \to (\forall z.A(z))), \exists y.(A(y) \to (\forall z.A(z))) \\
\hline
\vdash\text{contract} \\
B(m) \vdash \exists y.(A(y) \to (\forall z.A(z))) \\
\hline
\exists\vdash \\
\exists x.B(x) \vdash \exists y.(A(y) \to (\forall z.A(z)))
\end{array}
$$

Another Sequent Proof Example

The payoff for the earlier contraction step:

$$\begin{array}{c}
B(m), A(m) \vdash A(\_B1)\rightarrow(\forall z.A(z)), A(m1) \\ \hline
\vdash\exists \\
B(m), A(m) \vdash A(m1), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\forall \\
B(m), A(m) \vdash \forall z.A(z), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\rightarrow \\
B(m) \vdash A(m)\rightarrow(\forall z.A(z)), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\exists \\
B(m) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z))), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\text{contract} \\
B(m) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\exists\vdash \\
\exists x.B(x) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z)))
\end{array}$$

unify _B1, m1:

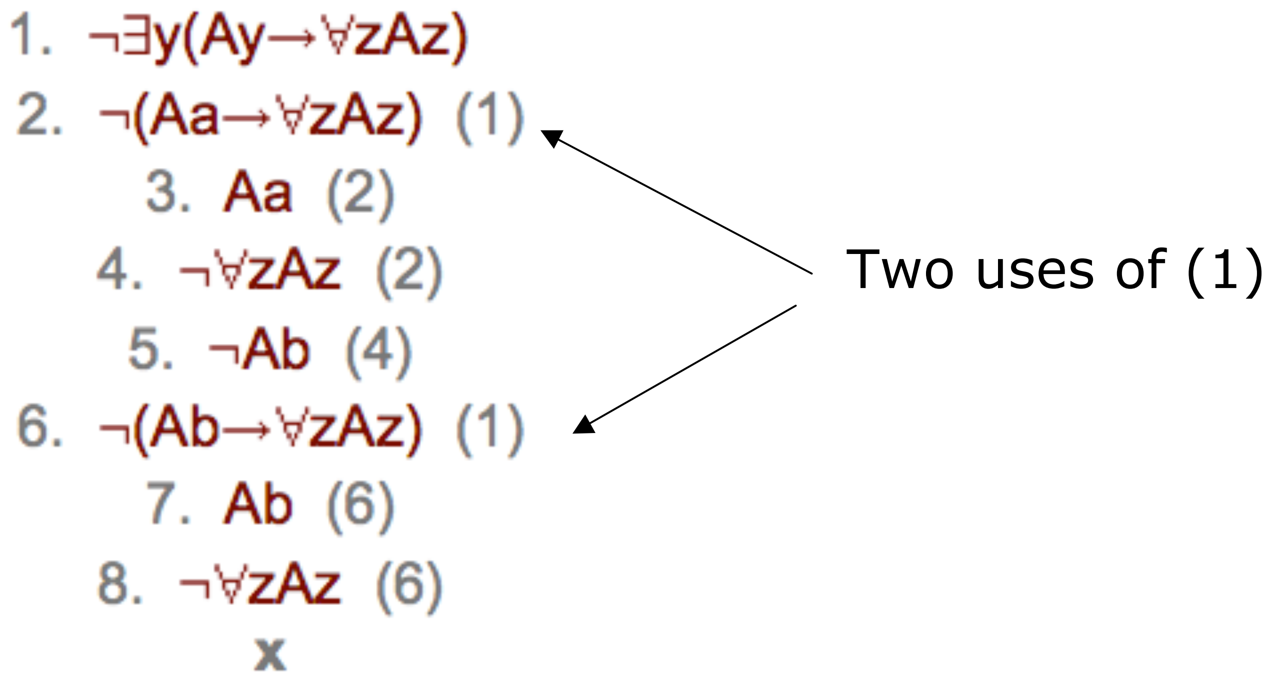
$$\begin{array}{c}
B(m), A(m) \vdash A(m1)\rightarrow(\forall z.A(z)), A(m1) \\ \hline
\vdash\exists \\
B(m), A(m) \vdash A(m1), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\forall \\
B(m), A(m) \vdash \forall z.A(z), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\rightarrow \\
B(m) \vdash A(m)\rightarrow(\forall z.A(z)), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\exists \\
B(m) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z))), \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\vdash\text{contract} \\
B(m) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z))) \\ \hline
\exists\vdash \\
\exists x.B(x) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z)))
\end{array}$$

Proof Concluded

$$\begin{array}{c}
\hline
\text{axiom}\\
B(m),\ A(m),\ A(m1) \vdash \forall z.A(z),\ A(m1)\\
\hline
\vdash\rightarrow\\
B(m),\ A(m) \vdash A(m1)\rightarrow(\forall z.A(z)),\ A(m1)\\
\hline
\vdash\exists\\
B(m),\ A(m) \vdash A(m1),\ \exists y.(A(y)\rightarrow(\forall z.A(z)))\\
\hline
\vdash\forall\\
B(m),\ A(m) \vdash \forall z.A(z),\ \exists y.(A(y)\rightarrow(\forall z.A(z)))\\
\hline
\vdash\rightarrow\\
B(m) \vdash A(m)\rightarrow(\forall z.A(z)),\ \exists y.(A(y)\rightarrow(\forall z.A(z)))\\
\hline
\vdash\exists\\
B(m) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z))),\ \exists y.(A(y)\rightarrow(\forall z.A(z)))\\
\hline
\vdash\text{contract}\\
B(m) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z)))\\
\hline
\exists\vdash\\
\exists x.B(x) \vdash \exists y.(A(y)\rightarrow(\forall z.A(z)))
\end{array}$$

Contrast Tableau Proof

The LHS premise is not really necessary,
since constants can be introduced freely.
This is equivalent to assuming a non-empty domain.
This prover combines some steps (which?):



1. $\neg\exists y(Ay \to \forall zAz)$
2. $\neg(Aa \to \forall zAz)$ (1)
    3. Aa (2)
    4. $\neg\forall zAz$ (2)
    5. $\neg Ab$ (4)
6. $\neg(Ab \to \forall zAz)$ (1)
    7. Ab (6)
    8. $\neg\forall zAz$ (6)

x

Two uses of (1)

http://www.umsu.de/logik/trees/