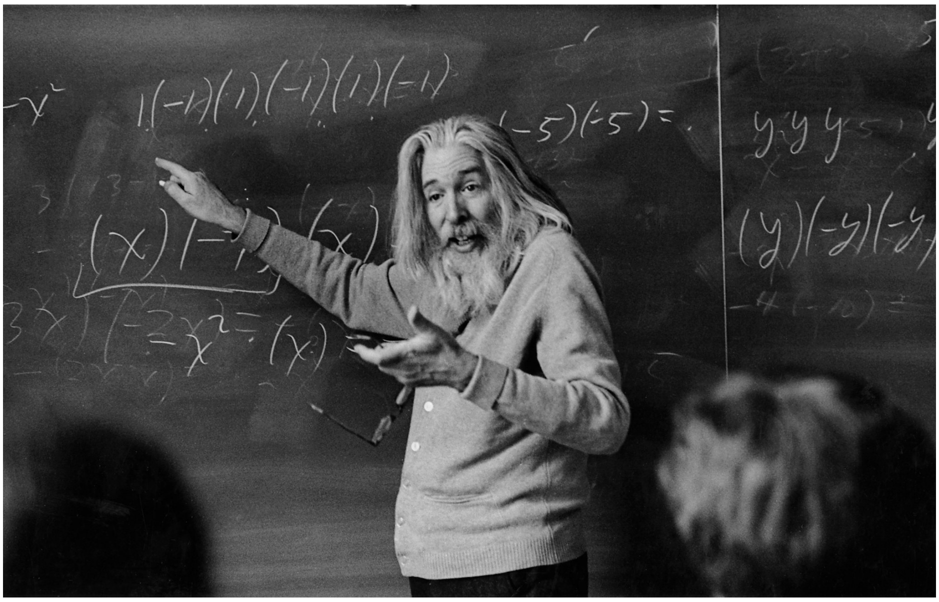
Raymond Smullyan's Tableaux
First-Order Logic



Courtesy of Luciano Serafini (FBK, Trento)

Reasoning tasks in FOL

In First order logics we have the same reasoning tatsk as in propositional logics (and any other logics)

Model checking

For a closed formula ϕ chesk if $\mathcal{I} \models \phi$

Satisfiability

Find an interpretation $\mathcal{I}$ that satisfies a closed formula ϕ. I.e., check if there is a $\mathcal{I}$ such that $\mathcal{I} \models \phi$.

Validity

Check if a formula ϕ is valid, i.e., if for all interpretations $\mathcal{I}$, $\mathcal{I} \models \phi$

Logical consequence

Check if a formula ϕ is a logical consequence of a set of formulas Γ, i.e., $\Gamma \models \phi$

Tableaux Calculus

- The Tableaux Calculus is an algorithm solving the problem of satisfiability.
- If a formula is satisfiable, then there exists an open branch in the tableaux of this formula.
- the procedure attempts to construct the tableaux for a formula. Sometimes it's not possible since the model of the formula is infinite.
- The basic idea is to incrementally build the model by looking at the formula, by decomposing it in a top/down fashion. The procedure exhaustively looks at all the possibilities, so that it can possibly prove that no model could be found for unsatisfiable formulas.

Semantic tableaux

Definition

A tableau is a rooted tree, where each node carries a first order sentence (closed formula), and the children of a node n are generated by applying a set of expansion rules to n or to one of the ancestors of n.

Definition

The expansion rules for a first order semantic tableaux are those for the propositional semantic tableaux, extended with the following rules that deal with the quantifiers:

γ rules	$\dfrac{\forall x.\phi(x)}{\phi(t)}$	$\dfrac{\neg\exists x.\phi(x)}{\neg\phi(t)}$	Where t is a term free for x in ϕ
δ rules	$\dfrac{\neg\forall x.\phi(x)}{\neg\phi(c)}$	$\dfrac{\exists x.\phi(x)}{\phi(c)}$	where c is a new constant not previously appearing in the tableaux

Tableaux production rules for propositional logic

. . . for propositional connectives

α rules

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}} \qquad \frac{\neg(\phi \vee \psi)}{\begin{array}{c}\neg\phi \\ \neg\psi\end{array}} \qquad \frac{\neg\neg\phi}{\phi} \qquad \frac{\neg(\phi \supset \psi)}{\begin{array}{c}\phi \\ \neg\psi\end{array}}$$

β rules

$$\frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\phi \supset \psi}{\neg\phi \mid \psi} \qquad \frac{\neg(\phi \wedge \psi)}{\neg\phi \mid \neg\psi} \qquad \frac{\phi \equiv \psi}{\begin{array}{c|c}\phi & \neg\phi \\ \psi & \neg\psi\end{array}} \qquad \frac{\neg(\phi \equiv \psi)}{\begin{array}{c|c}\phi & \neg\phi \\ \neg\psi & \psi\end{array}}$$

Substitution $\phi[x/t]$

If $\phi(x)$ is a free variable and t is a term, we use the notation $\phi(t)$ instead of the more precise notation $\phi[x/t]$ to represent the substitution of x for t in ϕ.

Substitution

$\phi[x/t]$ denotes the formula we get by replacing each free occurrence of the variable x in the formula ϕ by the term t. This is admitted if t does not contain any variable y such that x occurs in the scope of a quantifier for y (i.e., in the scope of $\forall y$ or $\exists y$.

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x . P(x, y)[y/f(x)] &=
\end{aligned}
$$

Not allowed since $f(x)$ is not free for y in $\forall x . P(x, y)$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x . P(x, y)[y/f(x)] &=
\end{aligned}
$$

Not allowed since $f(x)$ is not free for y in $\forall x . P(x, y)$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] &= \text{Not allowed since } f(x) \text{ is not free for } y \text{ in } \forall x.P(x, y)
\end{aligned}
$$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] &= \text{Not allowed since } f(x) \text{ is not free for } y \text{ in } \forall x.P(x, y)
\end{aligned}
$$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] &=
\end{aligned}
$$

Not allowed since $f(x)$ is not free for y in $\forall x.P(x, y)$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] &=
\end{aligned}
$$

Not allowed since $f(x)$ is not free for y in $\forall x.P(x, y)$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] &=
\end{aligned}
$$

Not allowed since $f(x)$ is not free for y in $\forall x.P(x, y)$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{aligned}
P(x, y, f(x))[x/a] &= P(a, y, f(a)) \\
\forall x P(x, y)[x/b] &= \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] &= \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] &= P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] &=
\end{aligned}
$$

Not allowed since $f(x)$ is not free for y in $\forall x.P(x, y)$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{array}{rcl}
P(x, y, f(x))[x/a] & = & P(a, y, f(a)) \\
\forall x P(x, y)[x/b] & = & \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] & = & \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] & = & P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] & = &
\end{array}
$$

Not allowed since $f(x)$ is not free for y in $\forall x.P(x, y)$

Substitution $\phi[x/t]$

Example (of substitution)

$$
\begin{array}{rcl}
P(x, y, f(x))[x/a] & = & P(a, y, f(a)) \\
\forall x P(x, y)[x/b] & = & \forall x P(x, y) \\
\exists x P(x, x) \wedge Q(x)[x/c] & = & \exists x P(x, x) \wedge Q(c) \\
P(x, g(y))[y/f(x)] & = & P(x, g(f(x))) \\
\forall x.P(x, y)[y/f(x)] & = &
\end{array}
$$

Not allowed since $f(x)$ is not free for y in $\forall x.P(x, y)$

Universal quantification rule

$$\frac{\forall x \phi(x)}{\phi(t)}$$

- $\forall x \phi(x)$ means that for every object of the domain, the property $\phi(x)$ should be true.
- a term t that occurs in the tableaux denotes an object of the domain
- therefore, $\phi(t)$ must be true for all the terms t that occurs in the tableaux. I.e., the $\forall$ rule can be applied as many time as one want to any term that appear in the tableaux.

Existential quantification rule

$$\frac{\exists x \phi(x)}{\phi(c)} \text{ for a new constant } c$$

- $\exists x \phi(x)$ means that for some object of the domain, the property $\phi(x)$ should be true.
- we don't know which object of the domain has the property ϕ, we know only that there is one.
- this means that this rule cannot be applied to the terms that already occur in the tableaux, since otherwise we would introduce an unjustified joiche on the element that has the property ϕ.
- the trick is to introduce a term to denote an unconditioned objects (sometimes called "fresh" constant/variable) for denoting an "unknown" object, i.e., an object on which we haven't done any commitment.
- therefore we allow only to infer $\phi(c)$ form $\exists x \phi(x)$, where c is fresh. Only one application is possible.

Open and Closed Branches

- a tableaux rooted with ϕ is a method to search an interpretation that satisfy ϕ
- Every branch of a tableaux with root equal to ϕ, corresponds to an attempt to find an interpretation $\mathcal{I}$ that satisfies ϕ.
- The interpretation corresponding to a branch b of a tableaux should satisfy all the formulas that appear in the branch.
- If the branch contains two opposite literals, i.e. $P(t_1, \ldots, t_n)$ and $\neg P(t_1, \ldots, t_n)$, then the branch cannot correspond to an interpretation, since there is no interpretation that satisfy at the same time $P(t_1, \ldots, t_n)$ and $\neg P(t_1, \ldots, t_n)$. So we can consider this attempt to find an interpretation failed. In this case we say that the branch is closed.
- if in a branch b all the rules has been applied and there is no opposite literals, then this branch corresponds to an interpretation. We call such a branch open

Open and Closed Branches

Definition

- A branch of a tableau is said to be closed if it contains a a pair of formulas ϕ and $\neg\phi$.
- A branch of a tableau is said to be open if it is not closed.
- A tableau is said to be closed if each of its paths is closed.

The tableaux method

1. To test a formula ϕ for validity, form tableau starting with $\neg\phi$. If the tableau closes off, then ϕ is logically valid.
2. To test whether ϕ is a logical consequence of Γ form a tableau starting with each formula in Γ and $\neg\phi$. If the tableau closes off, then ϕ is indeed a logical consequence of Γ.
3. To test a set of formulas Γ is satisfiable, form a tableau starting with Γ or equivalently an unsigned If the tableau closes off, then Γ is not satisfiable. If the tableau does not close off, then Γ is satisfiable, and from any open branch we can read off an interpretation satisfying Γ.

Example

Example

To check if the formula $(\exists x(P(x) \vee Q(x))) \equiv ((\exists x P(x)) \vee (\exists x Q(x)))$ is ~~satisfiable~~ valid, we start with a tableaux with this formula:

$$
\begin{array}{c}
\neg\,((\exists x\,(Px \vee Qx)) \Leftrightarrow ((\exists x\,Px) \vee (\exists x\,Qx))) \\
\begin{array}{cc}
/ & \backslash \\
\begin{array}{c}
\exists x\,(Px \vee Qx) \\
\neg\,((\exists x\,Px) \vee (\exists x\,Qx)) \\
\neg\,\exists x\,Px \\
\neg\,\exists x\,Qx \\
Pa \vee Qa \\
\begin{array}{cc}
/ & \backslash \\
Pa & Qa \\
\neg\,Pa & \neg\,Qa
\end{array}
\end{array}
&
\begin{array}{c}
\neg\,\exists x\,(Px \vee Qx) \\
(\exists x\,Px) \vee (\exists x\,Qx) \\
\begin{array}{cc}
/ & \backslash \\
\exists x\,Px & \exists x\,Qx \\
Pb & Qc \\
\neg\,(Pb \vee Qb) & \neg\,(Pc \vee Qc) \\
\neg\,Pb & \neg\,Pc \\
\neg\,Qb & \neg\,Qc
\end{array}
\end{array}
\end{array}
\end{array}
$$

Practicing with Semantic Tableaux

Exercize

Show with the method of semantic tableaux that the following formulas are valid:

- $\forall x P(x) \supset \neg \exists x \neg P(x)$
- $\forall x (P(x) \vee A) \supset (\forall x P(x) \vee A)$ when x is not free in A
- $\exists x (P(x) \supset \forall x P(x))$
- $\exists x \forall y P(x, y) \supset \forall y \exists x P(x, y)$

Practicing with Semantic Tableaux

Solution

$$\neg(\forall x P(x) \supset \neg \exists x \neg P(x))$$
$$\forall x P(x)$$
$$\neg\neg \exists x \neg P(x)$$
$$\exists x \neg P(x)$$
$$\neg P(a)$$
$$P(a)$$
$$\times$$

$$\neg(\forall x (P(x) \vee A) \supset (\forall x P(x) \vee A))$$
$$\forall x (P(x) \vee A)$$
$$\neg(\forall x P(x) \vee A)$$
$$\neg \forall x P(x)$$
$$\neg A$$
$$\neg P(a)$$
$$P(a) \vee A$$
$$P(a) \qquad\qquad A$$
$$\times \qquad\qquad \times$$

Practicing with Semantic Tableaux

Solution

$$\neg\exists x(P(x) \supset \forall x P(x))$$
$$|$$
$$\neg(P(a) \supset \forall x P(x))$$
$$|$$
$$P(a)$$
$$\neg\forall x P(x)$$
$$|$$
$$\color{red}{\neg P(b)}$$
$$|$$
$$\neg(P(b) \supset \forall x P(x))$$
$$|$$
$$\color{red}{P(b)}$$
$$\neg\forall x P(x)$$
$$|$$
$$\times$$

$$\neg(\exists x \forall y P(x,y) \supset \forall y \exists x P(x,y))$$
$$|$$
$$\exists x \forall y P(x,y)$$
$$\neg\forall y \exists x P(x,y)$$
$$|$$
$$\forall y P(a,y)$$
$$|$$
$$\neg\exists x P(x,b)$$
$$|$$
$$\color{red}{P(a,b)}$$
$$|$$
$$\color{red}{\neg P(a,b)}$$
$$|$$
$$\times$$

Example

Example

Check if $\forall x P(x) \wedge \exists x \neg P(f(x))$ is valid/satisfiable/unsatisfiable.

Solution

$$\forall x P(x) \wedge \exists x \neg P(f(x))$$
$$|$$
$$\forall x P(x)$$
$$\exists x \neg P(f(x))$$
$$|$$
$$\neg P(f(c))$$

Example

Example

Check if $\forall x P(x) \wedge \exists x \neg P(f(x))$ is valid/satisfiable/unsatisfiable.

Solution

$$\forall x P(x) \wedge \exists x \neg P(f(x))$$

$$|$$

$$\forall x P(x)$$
$$\exists x \neg P(f(x))$$

$$|$$

$$\neg P(f(c))$$

Now to expand $\forall x P(x)$, we can use any ground term t. possible choices: c, $f(c)$, $f(f(c))$, ... we choose $f(c)$ because we want to create a clash with $\neg P(f(c))$.

Example (Cont'd)

Example

Check if $\forall x P(x) \wedge \exists x \neg P(f(x))$ is valid/satisfiable/unsatisfiable.

Solution

$$\forall x P(x) \wedge \exists x \neg P(f(x))$$

$$|$$

$$\forall x P(x)$$
$$\exists x \neg P(f(x))$$

$$|$$

$$\neg P(f(c))$$

$$|$$

$$P(f(c))$$

$$|$$

$$\times$$

Example of tableaux

Example

$$\exists x(P(x) \wedge \neg Q(x)) \wedge \forall y(P(y) \vee Q(y))$$

$$\exists x(P(x) \wedge \neg Q(x))$$
$$\forall y(P(y) \vee Q(y))$$

$$P(a) \wedge \neg Q(a)$$

$$P(a)$$
$$\neg Q(a)$$

$$P(a) \vee Q(a)$$

$P(a)$ — OPEN

$Q(a)$ — CLASH

Termination

For certain formulas the tableaux can go on for ever generating an infinite tree. Consider, for instance the tableaux for the formula $\exists x.P(x) \wedge (\forall x P(x) \supset P(f(x)))$

$$\exists x.P(x) \wedge \forall x(P(x) \supset P(f(x)))$$

$$\exists x.P(x)$$
$$\forall x(P(x) \supset P(f(x)))$$

$$P(a)$$

$$P(a) \supset P(f(a))$$

$\neg P(a)$ — CLASH

$P(f(a))$

$$P(f(a)) \supset P(f(f(a)))$$

$\neg P(f(a))$ — CLASH

$P(f(f(a)))$

$\vdots$

Tableaux

Exercize

G4ive tableau proofs for the following logical consequences:

- $\forall x.P(x) \lor \forall x.Q(x) \models \neg\exists x(\neg P(x) \land \neg Q(x))$
- $\models \exists x.(P(x) \lor Q(x)) \equiv \exists x.P(x) \lor \exists x.Q(x)$

Some definition for tableaux

Definition (Closed branch)

A closed branch is a branch which contains a formula and its negation.

Definition (Open branch)

An open branch is a branch which is not closed

Definition (Closed tableaux)

A tableaux is closed if all its branches are closed.

Definition

Let ϕ be a first-order formula and Γ a finite set of such formulas. We write $\Gamma \vdash \phi$ to say that there exists a closed tableau for $\Gamma \cup \{\neg\phi\}$

Soundness and completeness

Theorem (Soundness)

$$\Gamma \vdash \phi \quad \Longrightarrow \quad \Gamma \models \phi$$

Theorem (Completeness)

$$\Gamma \models \phi \quad \Longrightarrow \quad \Gamma \vdash \phi$$

Remark

The mere existence of a closed tableau does not mean that we have an effective method to build it! Concretely: we don't know how often and in which way we have to apply] the γ-rules ($\forall x \phi(x) \Rightarrow \phi[x/t]$), and what term to use in the substitution.

Example

Check via tableaux if the validity/satisfiability of the formula $\phi = \forall x, y(P(x) \supset Q(y)) \supset (\exists x P(x) \supset \forall y Q(y))$

Solution

$\neg(\forall xy(P(x) \supset Q(y)) \supset (\exists x P(x) \supset \forall y Q(y)))$

|

$\forall xy(P(x) \supset Q(y))$
$\neg(\exists x P(x) \supset \forall y Q(y))$

|

$\exists x P(x)$
$\neg \forall y Q(y)$

$P(a)$

|

$\neg Q(b)$

|

$P(a) \supset Q(b)$

/ \

$\neg P(a)$ | $Q(b)$

| |

CLASH *CLASH*

We try, vith the tableaux, to build a model for the negation of ϕ. Since the tableaux ends with all CLASHES, there is no such a model. In other words, for all $\mathcal{I}$, $\mathcal{I} \not\models \neg\phi$. Which implies that for all $\mathcal{I}$, $\mathcal{I} \models \phi$, i.e., that ϕ is valid.

Infinite domains

- Differently from Prop. Logic, in FOL, models can be infinite.
- There are formulas which are satisfied only by infinite models. For instance the following formula[1]

$$\phi = \left(\begin{array}{rl} \forall x \neg R(x,x) & \wedge \\ \forall xyz.(R(x,y) \wedge R(y,z) \supset R(x,z)) & \wedge \\ \forall x. \exists y. R(x,y) & \end{array} \right)$$

- If we build a tableaux for such a formula, searching for a model, we will end up in an infinite tableaux.

[1] To verify this, suppose that $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ is an interpretation that satisfies ϕ, and suppose that $|\Delta| = n$ for some finite number n. Consider the sequence $\langle d_1, d_2, d_3, \ldots d_{n+1} \rangle$ of $n+1$ elements of Δ, such that $\langle d_i, d_{i+1} \rangle \in R^{\mathcal{I}}$. This sequence exists, because for every d there is always a d' with $\langle d, d' \rangle \in R^{\mathcal{I}}$, since $\mathcal{I} \models \forall x. \exists y. R(x,y)$. $\mathcal{I} \models \forall xyz.(R(x,y) \wedge R(y,z) \supset R(x,z))$, implies that $R^{\mathcal{I}}$ is transitive, and therefore for all $0 \leq i < j \leq n+1$, $\langle d_i, d_j \rangle \in R^{\mathcal{I}}$. The fact that Δ contains at most n elements implies that for some $1 \leq i < j \leq n+1$, $d_i = d_j$, which means that $\langle d_i, d_i \rangle \in R^{\mathcal{I}}$ for some $1 \leq i \leq n$. But this contraddicts the fact that $\mathcal{I} \models \forall x \neg R(x,x)$.

Infinite tableaux

Exercize

Build a tableaux for
$\forall x \neg R(x,x) \wedge \forall xyz.(R(x,y) \wedge R(y,z) \supset R(x,z)) \wedge \forall x. \exists y. R(x,y)$

Solution

$$\begin{array}{c} \forall x \neg R(x,x) \wedge \forall xyz.(R(x,y) \wedge R(y,z) \supset R(x,z)) \wedge \forall x. \exists y. R(x,y) \\ | \\ \forall x \neg R(x,x) \\ \forall xyz.(R(x,y) \wedge R(y,z) \supset R(x,z)) \\ \forall x. \exists y. R(x,y) \\ | \\ \exists y. R(a_0, y) \\ | \\ R(a_0, a_1) \\ | \\ \exists y. R(a_1, y) \\ | \\ \vdots \end{array}$$

By applying the γ-rule to the axiom $\forall x \exists y (R(x,y))$, we generate $\exists y R(a_0, y)$ for an initial constant a_0, and by applying the δ-rule to this last formula we generate a new individual a_1. This allow to apply the γ-rule again to $\forall x \exists y R(x,y)$, obtaining $\exists y R(a_1, y)$, and again by applying δ-rule to this new formula we generate another constant a_2. The process can go on infinitively without reaching any clash

Example of tableaux

Example

$\exists x(P(x) \wedge \neg Q(x)) \wedge \forall y(P(y) \vee Q(y))$

$\exists x(P(x) \wedge \neg Q(x))$
$\forall y(P(y) \vee Q(y))$

$P(a) \wedge \neg Q(a)$ $\quad$ $P(a) \vee Q(a)$

$P(a)$
$\neg Q(a)$

$P(a)$ $\quad$ $Q(a)$

OPEN $\quad$ CLASH

From the formulas appearing in the OPEN branch of the tableaux it is possible to construct a model for the root formula.

- $\Delta = \{a\}$, the constants appearing in the formulas
- $I(P) = \{a\}$, since the formula $P(a)$ appears in the open branch
- $I(Q) = \{\}$ since the formula $\neg Q(a)$ appears in the open branch

Termination fo FO tableaux

In contrast to what happens in propositional logic, the tableau construction is not guaranteed to terminate.
If the formula ϕ that labels the root is unsatisfiable, in which case the construction is guaranteed to terminate and the tableau is closed.
If the formula ϕ that labels the root is satisfiable then either the construction is guaranteed to terminate and the tableau is open, or the construction does not terminate.