

Logic in Action

Chapter 8: Validity Testing

`http://www.logicinaction.org/`

For the predicate logic case

The tableau method can be used also to decide the validity of inferences in predicate logic.

We already know how to deal with logical connectives (\neg , \wedge , \vee , \rightarrow , \leftrightarrow).

We just need to know how to deal with **quantifiers** (\exists , \forall).

Quantifiers (1)

\exists	$\exists x\varphi(x)$	\circ				\circ	$\exists x\varphi(x)$
	$\varphi(a)$	\oplus	\circ			\circ	$\varphi(a_1), \dots, \varphi(a_n)$
	For a new a					For all existing a_1, \dots, a_n	

\forall	$\forall x\varphi(x)$	\circ				\circ	$\forall x\varphi(x)$
	$\varphi(a_1), \dots, \varphi(a_n)$	\circ				\oplus	$\varphi(a)$
	For all existing a_1, \dots, a_n					For a new a	

Existential claims:	$\exists x\varphi(x)$	\circ			\circ	$\forall x\varphi(x)$
Universal claims:		\circ	$\exists x\varphi(x)$		$\forall x\varphi(x)$	\circ

Quantifiers (2)

What if we have a universal claim, but no names?

- We add a new element (because we do not allow empty domains).

$\forall x \varphi(x)$	\circ		\circ	$\exists x \varphi(x)$
$\varphi(a)$	\oplus		\oplus	$\varphi(a)$
	\circ		\circ	
For a new a				For a new a

Quantifiers (3)

Important observation.

- Every time a new name is introduced ($\overset{+}{\circ}$), we should **reactivate** every previous universal claim.

Recommendations

When working with predicate tableau, try to follow this order:

- 1 Work with logical connectives (\neg , \wedge , \vee , \rightarrow , \leftrightarrow).
- 2 Then, when working with existential claims.
- 3 Finally work with universal claims.

To practice

Which of the following statements are true?

- $\forall x(Px) \models \neg \exists x(\neg Px)$
- $\neg \exists x(Px) \models \forall x(\neg Px)$
- $\forall x \exists y Rxy \models \forall x Rxx$
- $\forall x \forall y Rxy \models \forall x Rxx$
- $\forall x \forall y Rxy, Rab \models Raa$
- $\forall x(Px \rightarrow Qx) \vee \forall y(Qy \rightarrow Py) \models \forall x \forall y((Px \wedge Qy) \rightarrow (Qx \vee Py))$
- $\forall x Px \rightarrow \forall x Qx \models \forall x(Px \rightarrow Qx)$
- $\forall x(Px \rightarrow Qx) \models \forall x Px \rightarrow \forall x Qx$
- $\exists y \forall x Rxy \models \forall x \exists y Rxy$
- $\forall x(Px \rightarrow Qx), \exists x(Px \wedge Rx) \models \exists x(Qx \wedge Rx)$
- $\forall x(Px \rightarrow Qx), \exists x(\neg Px \wedge Rx) \models \exists x(\neg Qx \wedge Rx)$
- $\neg \exists x(Px \wedge Qx), \forall x(Qx \rightarrow Rx) \models \neg \exists x(Px \wedge Rx)$
- $\forall x(Px \rightarrow Qx), \forall x(Qx \rightarrow Rx), \forall x(Rx \rightarrow Px) \models \forall x(Qx \wedge Px)$

Can we always find a counterexample?

Consider the following inference

$$\frac{\forall y \exists x Rxy}{\exists y \forall x Rxy}$$

- What does the inference says?
- Is it valid?
- Can you find a counterexample without using the tableau method?
- Can you find a counterexample with the tableau method?

What can we do?

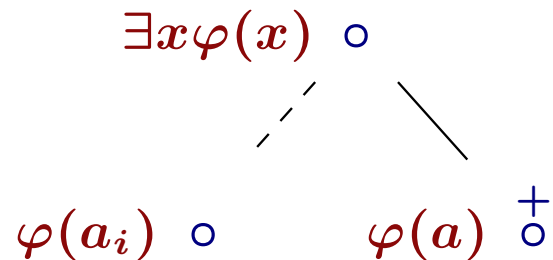
The problem

- For existential claims, we always introduce a new name.
- But maybe one of the previous names is useful.

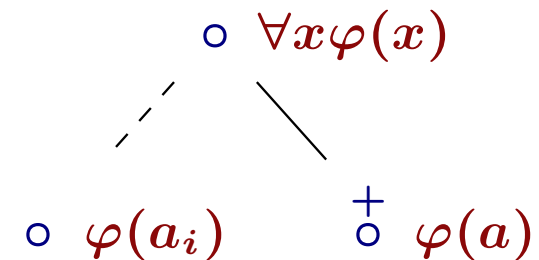
The solution

- For existential claims, we will now consider the possibility of a previous name being the adequate one.

Extended rules for existential claims



For an existing a_i and a new a



For an existing a_i and a new a

What happen now with $\frac{\forall y \exists x Rxy}{\exists y \forall x Rxy}$?

Can we always find a counterexample?

Consider the following inference

$$\frac{\forall y \exists x Rxy, \forall x \forall y \forall z ((Rxy \wedge Ryz) \rightarrow Rxz)}{\exists x \exists y (Rxy \wedge Ryx)}$$

- What does the inference says?
- Is it valid?
- Can you find a counterexample without using the tableau method?
- Can you find a counterexample with the tableau method?

What can we do?

The problem

- The tableau method tries to build counterexamples step by step, introducing at most one new name at each step.
- Hence, every model we built is **finite**.
- There are invalid inferences whose counterexamples are **infinite** models.

Important observations

- ① The **tableau** method attempts to build a model (domain and relations) with the specified requirements.
- ② The presented **tableau** method is **complete** for **proving validity** in **predicate** logic: if an inference with predicate formulas is valid, then its tableau will be closed.
- ③ The presented **tableau** method is **not complete** for **finding counterexamples** in **predicate** logic: if an inference with predicate formulas is not valid *and its counterexamples is an infinite model*, the tableau will not find it.
- ④ The presented **tableau** method **cannot** generate **every counterexample** of an invalid inference in **predicate** logic.