Truth Tables for Predicate Logic

G52DOA - Derivation of Algorithms

Venanzio Capretta

Truth Tables for Predicate Logic

In predicate logic, we can construct truth tables for a proposition relative to a specific interpretation.
Let D be the domain, and $d_0, d_1, d_2, \ldots$ all its elements.
The interpretation of a unary predicate P is given by a truth table:

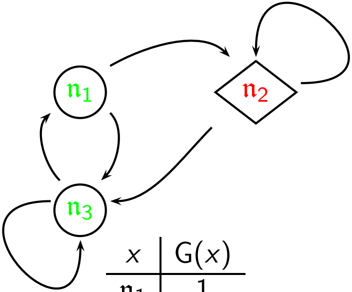
x	$P(x)$
d_0	1 or 0
d_1	1 or 0
d_2	1 or 0
$\vdots$	$\vdots$

Truth table for binary relation

The interpretation of a binary predicate R is also a truth table:

x	y	$R(x,y)$
d_0	d_0	1 or 0
	d_1	1 or 0
	d_2	1 or 0
	$\vdots$	$\vdots$
d_1	d_0	1 or 0
	d_1	1 or 0
	d_2	1 or 0
	$\vdots$	$\vdots$
$\vdots$		$\vdots$

Truth tables for a graph model



x	$G(x)$
$\mathfrak{n}_1$	1
$\mathfrak{n}_2$	0
$\mathfrak{n}_3$	1

x	y	$E(x,y)$
$\mathfrak{n}_1$	$\mathfrak{n}_1$	0
	$\mathfrak{n}_2$	1
	$\mathfrak{n}_3$	1
$\mathfrak{n}_2$	$\mathfrak{n}_1$	0
	$\mathfrak{n}_2$	1
	$\mathfrak{n}_3$	1
$\mathfrak{n}_3$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	0
	$\mathfrak{n}_3$	1

Truth Tables for complex propositions

x	$\mathsf{E}(x,x) \to \neg\mathsf{G}(x)$
$\mathfrak{n}_1$	1
$\mathfrak{n}_2$	1
$\mathfrak{n}_3$	0

x	$\neg\mathsf{G}(x) \to \mathsf{E}(x,x)$
$\mathfrak{n}_1$	1
$\mathfrak{n}_2$	1
$\mathfrak{n}_3$	1

x	$\neg\mathsf{E}(x,x) \wedge \neg\mathsf{G}(x)$
$\mathfrak{n}_1$	0
$\mathfrak{n}_2$	0
$\mathfrak{n}_3$	0

x	y	$\mathsf{G}(x) \wedge \neg\mathsf{G}(y) \to \mathsf{E}(x,y)$
$\mathfrak{n}_1$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	1
	$\mathfrak{n}_3$	1
$\mathfrak{n}_2$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	1
	$\mathfrak{n}_3$	1
$\mathfrak{n}_3$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	0
	$\mathfrak{n}_3$	1

Truth Tables for universal quantifications

To determine the truth table of $\forall x, A[x]$:

- Check the value of $A[x]$ for all assignments to x;
- If they are all 1, then $\forall x, A[x]$ is 1;
- If at least one is 0, then $\forall x, A[x]$ is 0.

In the example:

- $\forall x, (\mathsf{E}(x,x) \to \neg\mathsf{G}(x))$ has value 0;
- $\forall x, (\neg\mathsf{G}(x) \to \mathsf{E}(x,x))$ has value 1;
- $\forall x, (\neg\mathsf{E}(x,x) \land \neg\mathsf{G}(x))$ has value 0;

Truth Tables for existential quantifications

To determine the truth table of $\exists x, A[x]$:

- Check the value of $A[x]$ for all assignments to x;
- If at least one is 1, then $\exists x, A[x]$ is 1;
- If they are all 0, then $\exists x, A[x]$ is 0.

In the example:

- $\exists x, (\mathsf{E}(x, x) \rightarrow \neg \mathsf{G}(x))$ has value 1;
- $\exists x, (\neg \mathsf{G}(x) \rightarrow \mathsf{E}(x, x))$ has value 1;
- $\exists x, (\neg \mathsf{E}(x, x) \wedge \neg \mathsf{G}(x))$ has value 0.

Free and Bound Variables

Variables in a formula can be *free* or *bound*:

- x is *bound* if it is in the scope of $\forall x$ or $\exists x$;
- Otherwise it is *free*.

In $\forall x, R(x, y)$, x is bound, y is free.

Names of bound variables are irrelevant. They can be changed:

$\forall x, R(x, y)$ is the same as $\forall z, R(z, y)$.

But be careful not to *capture* free variables:

$\forall x, R(x, y)$ is NOT the same as $\forall y, R(y, y)$.

Truth Tables of formulas with free variables

Truth table: one row for every assignment to *free* variables.
For *bound* variables: make subtables:

x	y	$\mathsf{G}(x) \wedge \neg\mathsf{G}(y) \rightarrow \mathsf{E}(x,y)$	$\forall y, (\mathsf{G}(x) \wedge \neg\mathsf{G}(y) \rightarrow \mathsf{E}(x,y))$
	$\mathfrak{n}_1$	1	
$\mathfrak{n}_1$	$\mathfrak{n}_2$	1	1
	$\mathfrak{n}_3$	1	
	$\mathfrak{n}_1$	1	
$\mathfrak{n}_2$	$\mathfrak{n}_2$	1	1
	$\mathfrak{n}_3$	1	
	$\mathfrak{n}_1$	1	
$\mathfrak{n}_3$	$\mathfrak{n}_2$	0	0
	$\mathfrak{n}_3$	1	

Soundness and Completeness Theorem

A proposition A is provable if and only if it is true in every model (it is a tautology).

Specifically:

- Soundness: If A is provable, then it is true in all models;
- Completeness: If A is true in all models, then it is provable.

Therefore, if we have a proposition A and we want to find whether it is provable/tautological; we have to try to either prove it or find a *countermodel:* a model in which it is false.

Substitution

Given a proposition A, possibly containing the variable x, and a term t we define:

$A[t/x]$: the proposition A where all *free* occurrences of the variable x have been replaced with t

Example:

$$(R(x,y) \rightarrow P(x) \vee Q(y))[f(x)/y]$$
$$= R(x,f(x)) \rightarrow P(x) \vee Q(f(x))$$

Variable Capture

Warnings:

Don't replace bound variables:

$$(R(x,y) \rightarrow \exists y, (P(x) \vee Q(y)))[f(x)/y]$$
$$= R(x, f(x)) \rightarrow \exists y, (P(x) \vee Q(y))$$

Avoid variable capture by renaming bound variables:

$$(R(x,y) \rightarrow \exists x, (P(x) \vee Q(y)))[f(x)/y]$$
$$= (R(x,y) \rightarrow \exists z, (P(z) \vee Q(y)))[f(x)/y]$$
$$= R(x, f(x)) \rightarrow \exists z, (P(z) \vee Q(f(x)))$$
$$\neq R(x, f(x)) \rightarrow \exists x, (P(x) \vee Q(f(x)))$$

Universal Quantification

We reformulate the rules for quantifiers in terms of substitution.

Introduction

$$
\begin{array}{r|l}
 & \vdots \\
n & \quad \boxed{\begin{array}{ll} y & \qquad\qquad \text{eigenvariable} \\ \vdots & \\ A[y/x] & \end{array}} \\
 & \vdots \\
k & \forall x, A \qquad\qquad \forall_i \; n - m
\end{array}
$$

Elimination

$$
\begin{array}{r|l}
 & \vdots \\
n & \forall x, A \\
 & \vdots \\
k & A[t/x] \quad \forall_e \; n
\end{array}
$$

Existential Quantification

Introduction

$$
\begin{array}{r|ll}
 & \vdots & \\
n & A[t/x] & \\
 & \vdots & \\
k & \exists x, A & \exists_i\ n
\end{array}
$$

Elimination

$$
\begin{array}{r|ll}
 & \vdots & \\
h & \exists x, A & \\
 & \vdots & \\
n & \quad y & \text{eigenvariable} \\
n+1 & \quad A[y/x] & \text{assumption} \\
 & \quad \vdots & \\
m & \quad B & \\
 & \vdots & \\
k & B & \exists_e\ h, n-m
\end{array}
$$

Equality

There is one relation that is used in every signature and always has the same interpretation: *Equality*.
We can write it in the standard form: $\mathrm{Eq}(x, y)$.
But it is traditional to write: $x = y$.
Equality is always interpreted as the *sameness* relation: the relation that every object has only with itself.

Rules for Equality

Equality has its own natural derivation rules:

Introduction

$$
\begin{array}{r|ll}
 & \vdots & \\
k & t = t & \quad =_i
\end{array}
$$

Elimination

$$
\begin{array}{r|ll}
 & \vdots & \\
n & t_1 = t_2 & \\
 & \vdots & \\
m & A[t_1/x] & \\
 & \vdots & \\
k & A[t_2/x] & \quad =_e \ n, m
\end{array}
$$

The elimination rule states that if $t_1 = t_2$, then we can replace t_1 with t_2 in any proposition.