Deduction
by Daniel Bonevac

Chapter 6 Quantified Truth Trees

The Language Q

- Before we start studying quantified truth trees, we need to formalize our understanding of the language of Q, i.e., quantificational logic. (Bonevac actually does this in Ch. 5 on p. 153)
- It's important to understand that Q is not a brand new language, but an extension of sentential logic. That means that if we want to we can still use our letters p, q, r, s, p_1, s_1, etc. to stand for entire sentences.
- So, for example, these are going to be perfectly well formed sentences in Q:
  - $p \rightarrow \forall xFx$
  - $p \vee \exists y \forall xFxy$
- Of course, that's only if 'p' is standing for an entire sentence. If 'p' is standing for a proper name, like Paul (which, as we'll see, is not officially permitted) then the above sentences are not well-formed at all.

Vocabulary of Q

- The complete vocabulary of Q is:
  - Sentence letters: p, q, r, s, p_1, q_1,…
  - n-ary predicate constants: F, G, H,…M, F_1, G_1…
  - Individual constants: a, b, c, ..., o, a_1, b_1…
  - Individual varibables: t, u, v, w, x, y, z, t_1, u_1…
  - Sentential connectives: ¬, &, v, →, ↔
  - Quantifiers ∀, ∃
  - Grouping Indicators (,)

Formation Rules

- Any sentence letter is a formula
- An n-ary predicate followed by n individual constants is a formula.
- If $\mathcal{A}$ is a formula, then $\neg\mathcal{A}$ is a formula.
- If A and B are formulas, then $(\mathcal{A} \rightarrow \mathcal{B})$, $(\mathcal{A} \vee \mathcal{B})$, $(\mathcal{A} \,\&\, \mathcal{B})$, $(\mathcal{A} \leftrightarrow \mathcal{B})$ are formulas.
- If $\mathcal{A}c$, is a formula with individual constant c, and ν is a variable that does not occur in $\mathcal{A}c$, the $\exists\nu\mathcal{A}\nu$ and $\forall\nu\mathcal{A}\nu$, are formulas.
- Every formula can be constructed from a finite number of applications of these rules.

Unbound variables

- The most important thing to understand about the formation rules is that what it tells you about the relation between variables and individual constants constants.
  - We can't have any free variables. All variables have to be bound by a quantifier. So this is not a formula in Q:
    - ∀xFxy
  - We can't have multiple quantifiers binding the same variable. This this is not a formula in Q:
    - ∃x∀xFxx
  - Quantifiers don't bind constants. This is not a formula in Q:
    - ∀xFa
  - Finally, we can not substitute variables for predicate constants. This is not a formula in Q:
    - ∀xXb
    - As Bonevac note, there are systems of logic that permit this. They are called second-order quantificational logic. Our system is first order, meaning it only quantifies over individual constants.

Truth trees in quantificational logic

- Truth trees in quantificational logic are just like truth trees in sentential logic, with the addition of some new rules to deal with the quantifiers.
- In quantificational logic, the truth trees method does is not actually a <u>decision procedure</u>, as it is in sentential logic. In other words, it does not guarantee a result. This is not a defect in the method. It is simply a proven fact about systems of quantificational logic that they are not decidable.

Instances

- Inference in quantificational logic proceeds by rules for instantiating, or creating instances.
- The existential and universal quantifiers are bound by different rules for fairly intuitive reasons.
  - An existentially quantified expression, like
    - $\exists xGx$

    where G means 'goofy' tells us only that at least one x is goofy. But we can not infer from this that any particular individual is goofy.
  - A universally quantified expression like
    - $\forall xGx$

    tells us that everything is goofy. That means we can create any instance we like: Ga, Gb, Gc, etc.

Quantifiers

- To understand the truth tree rules you first you first need to understand that quantifiers are in effect a kind of connective.
- Because truth tree rules only allow you to operate on the main connective, this applies to the quantifier as well.
- The scope of the quantifier is that part of the formula (including the quantifier itself) to which the quantifier applies. Consider the following formula:
  - (∀xFx v ∃yGy)

  Here the main connective is v. The scope of the universal quantifier is ∀xFx . The scope of the existential quantifier is ∃yGy
- But consider the following formula:
  - ∀x∃y(Fxy v Fyx)

  Now the main connective is actually the universal quantifier. It is the main connective, or has the widest scope, because it is the outermost quantifier.
- Importantly, then, with quantifiers, the requirement that we always operate on the main connective means that when a quantifier is a main connective, we are always operating on the outermost quantifier.

Existential (E)

- The rule for instantiating the existential quantifier is a kind of trick. Recall that problem is basically that since with an existentially quantified expression like ∃xGx, we only know that some unidentified x is G. So we basically just give this x a name, that is not the name of anything else we know.
- Here's a helpful way to think about this. Comedians know that when you are telling a joke, it's important not to just: "So this guy goes into a bar.. and then just keep referring to him as 'this guy'. To make it funnier give him a names. But to make sure you know they're not talking about anybody real, they'll often say something like: "So this guy goes into a bar. Let's call him René. So the bartender says to René "How 'bout a drink?" and René says "I think not." And he suddenly disappears."
- Of course, that joke is only funny if you know it's René Descartes you're talking about, so it actually undermines the point of the above paragraph. The important thing is to introduce a name that is the name of no one in particular.
- In logic, we take advantage of this technique by instantiating existentials only to <u>brand new constants</u>. The constant has to be <u>new</u> to the proof.

Existential (E) 2

- So, suppose you are testing the following inference for validity.
  - ∃xFx ∴ Fa
- You begin by negating the conclusion as before and trying to generate a contradiction

  ∃xFx

  ¬ Fa
- Clearly, the only way to get a contradiction is to instantiate to Fa. But, of course you can't do that, because the constant 'a' is not new to the proof. All you can do is instantiate to a new constant, like 'b'.

  √∃xFx

  ¬ Fa

  Fb

  and the line here remains open.

Negated existential (¬E)

- Negated existentials are not restricted in the same way that non negated existentials are.
- The reason for this is that a negated existential like
  - ¬∃xFx

  tells us that there is nothing at all that has F. So we are just as free to say ¬Fa as ¬Fb or any other constant we like.
- Interestingly, what this means is that we never really fully dispatch a negated existential, which is why we use *, not √. The asterisk is a temporary dispatch mark. It means that we can instantiate the negated existential as many times as we like.

Example using (¬E)

- $\neg\exists x\, Fx \rightarrow Ga$, $\neg\exists x\, Gx$ ∴ Show $\exists x\, Fx$

$\neg\exists x\, Fx \rightarrow Ga$

$\neg\exists x\, Gx$

$\neg\, \exists x\, Fx$

Example using (¬E)

- $\neg\exists x\, Fx \rightarrow Ga$, $\neg\exists x\, Gx$ ∴ Show $\exists x\, Fx$

$\neg\exists x\, Fx \rightarrow Ga$

*$\neg\exists x\, Gx$

$\neg\exists x\, Fx$

$\neg Ga$

Example using (¬E)

- ¬∃x Fx → Ga, ¬∃x Gx ∴ Show ∃x Fx

¬∃x Fx → Ga
*¬∃x Gx
*¬ ∃x Fx
¬ Ga
¬ Fa

- Notice here that we could have gone to ¬Fb or any constant we like, but we took advantage of the unrestricted nature of (¬E) to to get a contradiction.

Example using (¬E)

- $\neg\exists x\, Fx \rightarrow Ga,\ \neg\exists x\, Gx\ \therefore$ Show $\exists x\, Fx$

$\surd\neg\exists x\, Fx \rightarrow Ga$

$*\neg\exists x\, Gx$

$*\neg\ \exists x\, Fx$

$\neg\ Ga$

$\neg\ Fa$

$\neg\neg\exists x\, Fx$	Ga
x	x

Universal (U)

- Instantiating a universal is an entirely free move as well. A statement like $\forall x Fx$ says that everything has F, hence it follows that Fa, Fb, Fc, etc.
- For this reason, we also use an * rather than a √ to dispatch a universally quantified expression.

Negated Universal (¬U)

- A negated universal like $\neg\forall xFx$ tells us that <u>not everything</u> is F, in other words, there is at least one thing that is not F. Hence, it is restricted in the same way that an existentially quantified expression is.
- Namely, when you instantiate a negated universal, the constant you use must be <u>entirely new</u> to the proof.

Example using (U) and (¬U)

- $\forall x \neg \forall y(Fy \rightarrow Gx) \quad \therefore \quad \exists x \exists y(Fy \,\&\, \neg Gx)$

$$\forall x \neg \forall y(Fy \rightarrow Gx)$$
$$\neg \exists x \exists y(Fy \,\&\, \neg Gx)$$

  - Negate the conclusion in the usual way.

Example using (U) and (¬U)

$*\forall x \neg \forall y(Fy \rightarrow Gx)$

$\neg \exists x \exists y(Fy \,\&\, \neg Gx)$

$\neg \forall y\,(Fy \rightarrow Ga)$

- We make this move first because even though both lines are unrestricted, we see that the new line is not. In general, we want to take care of our restricted moves first, as this will maximize the opportunity to generate contradictions in the fewest number of steps.

Example using (U) and (¬U)

*∀x¬∀y(Fy → Gx)
¬∃x∃y(Fy & ¬Gx)
√¬∀y(Fy → Ga)
¬(Fb →Ga)

- Note that we had to introduce a new constant here because (¬U) is restricted.

Example using (U) and (¬U)

*∀x¬∀y(Fy → Gx)

¬∃x∃y(Fy & ¬Gx)

√¬∀y(Fy → Ga)

√ ¬(Fb → Ga)

Fb

¬Ga

- This, of course, is just the rule for →.

Example using (U) and (¬U)

*$\forall x \neg \forall y(Fy \rightarrow Gx)$

*$\neg \exists x \exists y(Fy \,\&\, \neg Gx)$

√$\neg \forall y(Fy \rightarrow Ga)$

√ $\neg(Fb \rightarrow Ga)$

Fb

$\neg Ga$

$\neg \exists y(Fy \,\&\, \neg Ga)$

- This is an unrestricted move, and we go to Ga with the hope of getting a contradiction.

Example using (U) and (¬U)

*∀x¬∀y(Fy → Gx)
*¬∃x∃y(Fy & ¬Gx)
√¬∀y(Fy → Ga)
√ ¬(Fb → Ga)
Fb
¬Ga
*¬ ∃y(Fy & ¬Ga)
¬(Fb & ¬Ga)

- This, too is an unrestricted move, and we go to Fb with the hope of getting a contradiction.

Example using (U) and (¬U)

$*\forall x \neg \forall y(Fy \rightarrow Gx)$

$*\neg \exists x \exists y(Fy \,\&\, \neg Gx)$

$\surd \neg \forall y(Fy \rightarrow Ga)$

$\surd\ \neg(Fb \rightarrow Ga)$

Fb

$\neg Ga$

$*\neg\ \exists y(Fy \,\&\, \neg Ga)$

$\surd\ \neg(Fb \,\&\, \neg Ga)$

$\neg Fb$	$\neg\neg Ga$
x	x

- Finally, the ¬& rule and we're done.

Strategies

- In addition to the strategies from truth trees in sentential logic, there are two strategies relating to the quantifiers.

1. Apply the restricted rules (E and ¬U) before the unrestricted ones (¬E and U)
2. Don’t introduce new constants unnecessarily. In other words, when using the unrestricted rules, remember that you can, and often should, instantiate them to constants already in the proof.