Deduction
by Daniel Bonevac

Chapter 8 Identity and Functions

Introduction 1

- This chapter introduces two important extensions of Q that make it quite a bit more powerful.
- The first is the mathematical relation of identity (=). Up until now we would find it difficult to express a simple sentence like:
  - Marilyn Monroe is Norma Mortenson .

  Both of these terms are names, not predicates, so we could only have translated this as something like
  - Emn

  Where E is a predicate meaning “equals” or “identical to”.

  There is nothing really wrong with this, but it doesn’t allow us to make any inferences based on well known properties of identity. For example, if we know that Marilyn Monroe is Norma Mortenson, then we also know that Norma Mortenson is Marilyn Monroe. But right now we can’t prove that.

Introduction 2

- The second extension of Q is the idea of a <u>function</u>. A function is really just a specific kind of relation, but it is a relation with special properties. Basically, a function relates two objects as input to output. Functions are very important in mathematics, but they are also essential to our understanding of English.
- For example, until now if we wanted to express a statement like:
  - Mary is the mother of Jesus.
- We could only have done so like this:
  - Mmj
- This is fine, but it turns out to be very useful to be able to treat the phrase "mother of Jesus" as indicating a particular person, just as the phrase "square of 10" indicates a particular number.

Identity

- To introduce the concept of identity, we simply introduce two new symbols: $=$ and $\neq$.
- Identity is a binary relation. In Q it does not hold between predicates or sentences. Rather it holds between individual constants and variables.
  - $a = b$
  - $\exists x \forall y\ x \neq y$
- but not
  - $Fa = Gb$
  - $p = q$ (where p and q stand for sentences)
  - $F = G$

- Also note that while this is a formula:
  - $\neg a=a$
- We only use it when we have more than one negation associated with identity. For example:
  - $\neg a \neq a$

Identity vs. Predication

- In English, the word "is" can be used in two importantly different ways. To see this, consider the following statements.
  - Tony Stark is clever.
  - Tony Stark is Iron Man.
- The first use is what we call the <u>'is' of predication</u>. It simply predicates the property of cleverness to a person named Tony Stark: Ct
- The second use is what we call the <u>'is' of identity</u>. It says that Tony Stark and Iron Man are the same guy: $t = s$

Identity statements 1

- With identity we can now say all sorts of exciting things that we could not say before.
- For example:
  - Tony is the only one who is having fun.
- This sentence doesn't just say that Tony is having fun. It says that Tony is having fun, <u>and</u> nobody else is. We can express this as follows:
  - $Ft \,\&\, \forall x\, (Fx \rightarrow x{=}t)$.
- In English, this says:
  - Tony is having fun, and for all x, if x is having fun, then x is identical to Tony.

Identity statements 2

- Pages 227-229 show you ways that we can now characterize certain kinds of quantitative statements. For example:
  - There are at least two apples.
- can be expressed as
  - $\exists x \exists y\ ((Ax \ \&\ Ay)\ \&\ x \neq y)$
- Alternatively
  - There are at most two apples
- can be expressed as
  - $\forall x \forall y \forall x(\ (Ax \ \&\ Ay \ \&\ Az\ ) \rightarrow (x=y \lor y=z \lor x = z))$
- Notice that the author has here adopted the common convention of dropping parentheses for strings of conjunctions or disjunctions.

Identity statements 3

- We can also express superlatives like:
  - Big Brown is the fastest horse.
- This is essentially to say that Big Brown is faster than every horse not identical to Big Brown. Symbolically:
  - $\forall x$ ($x \neq b$, then Fbx)

Rules for identity

- We introduce new rules of inference for identity. The truth tree rules and the natural deduction rules are slightly different, but the basic ideas are the same.
- There are two new rules for each system.
  - One kind of rule allows you to make inferences from the fact that two things are identical. Specifically, if you know something like Fa and you know that a=b, then you can infer Fb.
  - The other kind or rule allows you to exploit the fact that something is always identical to itself: $a=a$.

Truth tree rules: identity (=)

- The first truth tree rule allows you to exploit the identity relation. It can be reused, so you only mark it with a * after you have used it. It works like this.

Fab	or	Fab	or	Faa
*a=b		*a=b		*a=b
Fbb		Faa		Fba

- Fbb is simply the result of substituting b for a in the first expression sentence Fab, and it is justified by the fact that a=b. You can created Faa by the same process as shown
- Importantly, you do not actually have to substitute for <u>every</u> occurrence of the constant, as shown by the third example.

Truth tree rules: negated identity (≠)

- Negated identity allows you to cancel any line that contains a statement of the form $a \neq a$. Because it is always the case that a=a, this is tantamount to a mathematical contradiction.
- Example:

*a=b
b=c
a≠c
*a=c
a≠a
x

Natural deduction rules: =I

- The first natural deduction rule for identity is called identity introduction or (=I). It is the analog of the negated identity rule for truth trees.
- =I simply allows you to introduce any sentence of the form
  - $a=a$
- whenever you like. There will be no line number in the justification. Simply write =I.

Natural deduction: =E

- The second natural deduction rule for identity is the same as the identity truth tree rule.
- Here we call it =E. So, if you have a proof that looks like this:
  - n. Fab
  - m. a=b
- then on some later line you can write
  - o. Fbb =I, n,m.
- However you can only perform this rule on <u>constants</u>. You can't do this:
  - n. $\forall x \forall y Fxy$
  - m. $\forall x \forall y\ x=y$
  - o. $\forall x Fxx$ =I, n,m.

An example

- Let's do a translation example and then prove it using both methods.
  - Brian is the only one who doesn't shower.
  - Only someone who doesn't shower stinks.
  - So, if anyone stinks, it's Brian.
- We will translate this on the assumption that the <u>domain of discourse</u> is people. This means that every object referred to here is a person, so we don't have to keep using the predicate 'is a person'.
- On this assumption, the argument translates as follows.
  - $\forall y(\neg Sy \rightarrow y=b)$
  - $\forall x(Ts \rightarrow \neg Sx)$
  - $\forall x\ (Tx \rightarrow x=b)$

Truth tree solution

$$\forall y(\neg Sy \rightarrow y=b)$$

$$\forall x(Tx \rightarrow \neg Sx)$$

$$\neg\forall x\,(Tx \rightarrow x=b)$$

Truth tree solution

$\forall y(\neg Sy \rightarrow y=b)$

$\forall x(Tx \rightarrow \neg Sx)$

$\surd\ \neg\forall x\ (Tx \rightarrow x=b)$

$\surd\neg(Ta \rightarrow a=b)$

Ta

$a\neq b$

Truth tree solution

$*\forall y(\neg Sy \rightarrow y=b)$

$*\forall x(Tx \rightarrow \neg Sx)$

$\surd\ \neg\forall x\ (Tx \rightarrow x=b)$

$\surd\neg(Ta \rightarrow a=b)$

Ta

$a \neq b$

$\neg Sa \rightarrow a=b$

$Ta \rightarrow \neg Sa$

Truth tree solution

$* \forall y(\neg Sy \rightarrow y{=}b)$

$* \forall x(Tx \rightarrow \neg Sx)$

$* \neg\forall x\, (Tx \rightarrow x{=}b)$

$\surd \neg(Ta \rightarrow a{=}b)$

Ta

$a \neq b$

$\surd\ \neg Sa \rightarrow a{=}b$

$\surd\ Ta \rightarrow \neg Sa$

$\neg\ Ta$ | $\neg Sa$

x

$\neg\neg Sa$ | $a{=}b$

x | x

Truth tree solution

$* \forall y(\neg Sy \rightarrow y=b)$

$* \forall x(Tx \rightarrow \neg Sx)$

$\surd\ \neg\forall x\ (Tx \rightarrow x=b)$

$\surd \neg(Ta \rightarrow a=b)$

Ta

$a \neq b$

$\neg Sa \rightarrow a=b$

$\surd\ Ta \rightarrow \neg Sa$

$\neg\ Ta$ | $\neg Sa$

x

Natural Deduction Solution

1.	$\forall y\,(\neg Sy \rightarrow y{=}b)$	A
2.	$\forall x\,(Tx \rightarrow \neg Sx)$	A
3.	Show: $\forall x\,(Tx \rightarrow x{=}b)$	
4.	Show: $Ta \rightarrow a{=}b$	
5.	Ta	ACP
6.	$Ta \rightarrow \neg Sa$	$\forall$E,2
7.	$\neg Sa \rightarrow a{=}b$	$\forall$E,1
8.	$\neg Sa$	$\rightarrow$E, 5,6
9.	$a{=}b$	$\rightarrow$E, 7,8

Cancel show lines.

Example 2

- Here is a proof that requires the use of identity exploitation.
- Whatever is pooping on the porch is big. Marty is not big. So, Marty is not the one pooping on the porch.

- Translation:
- Let P stand for “pooping on the porch”

  $\forall x(Px \rightarrow Bx)$

  $\neg Bm$

  Therefore, $\forall x(Px \rightarrow x \neq m)$

Truth tree solution

$*\forall x(Px \rightarrow Bx)$

$\neg Bm$

$\surd\neg\, \forall x(Px \rightarrow x \neq m)$

$\surd\neg(Pa \rightarrow a \neq m)$

Pa

$\surd\neg a \neq m$

$*a = m$

$(Pm \rightarrow Bm)$

$\neg Pm$	Bm
Pm	$\neg Ba$
x	x

Natural deduction solution

1.	$\forall x(Px \rightarrow Bx)$	A
2.	$\neg Bm$	A
3.	~~Show~~ $\forall x(Px \rightarrow x \neq m)$	
4.	~~Show~~ $Pa \rightarrow a \neq m$	
5.	Pa	ACP
6.	~~Show~~ $a \neq m$	
7.	$a=m$	AIP
8.	$\neg Ba$	=E 2,7
9.	$Pa \rightarrow Ba$	$\forall$E, 1
10.	$\neg Pa$	$\rightarrow$E* 8,9
11.	Pa	R, 5

Functions

- It is surprisingly easy to introduce functions into Q, and they actually require no new rules.
- You probably remember that in mathematics we sometimes represent a function as follows
  - $f(x)$
- For example, we might say that
  - $f(x) = x^2$
- This would allow us to write, for example that
  - $f(3) = 9$
- We basically do the same thing in Q, with the provision that variable be bound.
- So, suppose we let
  - $f(x) =$ "father of x"
- Now suppose we know that the father of Sarah is Michael. We can represent this as:
  - $f(s) = m$

Understanding function symbolism

- As we noted in the beginning, functions are basically just certain kinds of relations, and function symbolism is just a way of representing a relation that is conducive to certain kinds of inferences.
- The statement: "Michael is the father of Sarah" can be represented as
  - Fms

or

  - f(s) = m

- By representing the statement in the latter form we can combine function statements with the rules of identity to produce inferences we could not produce before.
- If you recall the distinction between identity and predication, you may note that it's not entirely correct to say that the two expressions above mean <u>exactly</u> the same thing.
- To preserve the distinction, we can say that ms means "Michael fathered Sarah," so that the father-offspring relation is being predicated of Michael and Sarah.
- On the other hand f(s) = m says that the "Michael is identical to the father of Sarah"

Understanding function symbolism 2

- Because function symbols go in the place of constants and variables, they are treated in the same manner. This is why we don't need any new rules for them.
- For example, if we say something like:
  - Billy's father loves Sarah's mother.
- We would write this as:
  - Lf(b)m(s)
- Read as: The father of Billy loves the mother of Sarah.
- If we knew that Billy's father is Alvin and that Sarah's mother is Denise, then we would write this as
  - f(b) = a
  - m(s) = d
- We can then use =E to show that.
  - Lad

Understanding function symbolism 3

- The other interesting thing about functions is that they can be iterated.
- Again, you may remember from mathematics that we can say things like this.
  - f(g(x))
- Suppose $f(x) = x^2$ and $g(x) = 1/x$
  - then $f(g(2))= ¼$.
- In logic we can represent functions of functions in the same way. For example, we can represent "paternal grandfather" as g(x) or as f(f(x)).
- Hence, we can write
  - $g(x) = f(f(x))$
- Hence, if we know that Alvin is the paternal grandfather of Betty, we'll be able to easily prove that Alvin is the father of the father of Betty.

Function conditions

- It’s important to understand that functions are special kinds of relations. There are two conditions they must meet.
- The first is called the existence condition.
- The second is called the uniqueness conditions.

The existence condition

- The existence condition on functions requires that every input produces an output.
- The relation “father of” (understood as biological father) is reasonably construed as a function because every human has a biological father.
- On the other hand “brother of” is not a function, because not every person has a brother.

The uniqueness condition

- The uniqueness condition requires that a function have <u>exactly</u> one output for every input.
- Again, “father of” (biological) is reasonably construed as a function because everyone has exactly one father.
- But even though “grandfather of” meets the existence condition, it does not meet the uniqueness condition, because everyone has two biological grandfathers.

Perils of ignoring existence and uniqueness

- We can get into a lot of trouble ignoring these two conditions. It's a lot like ignoring the fact that you can't divide by zero. Division is technically not a function because the fact that you can't divide by zero means it fails the existence condition, at least if division is defined on the set of integers. Here's the proof in case you have forgotten it:

1.	$a=b$	A (a and b not equal to 0)
2.	Show: $2 = 1$	
3.	$a^2=ab$	multiplying both sides equally.
4.	$a^2 – b^2 = ab - b^2$	subtracting from both sides equally.
5.	$(a – b)(a +b) = b(a –b)$	distributive property of multiplication
6.	$(a + b) = b$	dividing both sides by (a-b)
7.	$(b+b) = b$	=E, 1,6
8.	$2b=b$	addition
9.	$2 = 1$	divide both sides equally

Bogus function proofs

- In the same way, you can do bogus function proofs if you ignore these conditions. For example, if you ignore the uniqueness condition you can prove the folowing:
  - Juan's dog is his pet.
  - Juan's cat is his pet.
  - Therefore, Juan's dog is Juan's cat.
- This inference would never go through if "dog of" and "cat of" and "pet of" were translated as binary predicates. But if they are allowed to pose as functions, we would translate the argument and prove it as follows:

1. $d(j) = p(j)$	A
2. $c(j) = p(j)$	A
3. Show $d(j) = p(j)$	
4. $d(j) = c(j)$	=E, 1,2