Predicate Calculus

The logic we have learned so far goes only a little bit beyond Aristotle's logic. The great leap forward was to extend the logic to encompass relations as well as properties.

Consider the sentence "Isaac is a son of Abraham." We can think of the sentence as attributing a property to Isaac, so that, if we use "i" to denote Isaac and "A" to designate the sons of Abraham, we can symbolize the sentence "Ai." We may also think of the sentence as attributing a property to Abraham, so that, using "a" to denote Abraham and "I" to designate Isaac's parents, we have "Ia." But we may also think of the sentence as about both Abraham and Isaac, saying of them that they stand in a certain relation. For this we need to go beyond the monadic predicate calculus, where we could only talk about properties, not about relations.

Let "S" denote the son-of relation. Then we can symbolize "Isaac is a son of Abraham" as "Sia." Using "m" for "Ishmael," we'll write "Sma" for "Ishmael is a son of Abraham." "Isaac is a son of Sara" will be "Sis." "Ishmael is a son of Abraham but not of Sara" will be "$(Sma \wedge \neg Sms)$."

To say "Abraham has a son" will be "$(\exists x)Sxa$." To say "Abraham is a son" is "$(\exists x)Sax$." To say "Abraham and Sara both have sons" will be "$((\exists x)Sxa \wedge (\exists x)Sxs)$," whereas to say "Abraham and Sara have a son" will be "$(\exists x)(Sxa \wedge Sxs)$." "Every son of Sara is a son of Abraham, but not every son of Abraham is a son of Sara" will be "$((\forall x)(Sxs \rightarrow Sxa) \wedge \neg(\forall x)(Sxa \rightarrow Sxs))$."

To say "Isaac is a son of Abraham," we write "Sia." If we say "Everyone is a son of Abraham," we are saying that everyone has the property that the sentence "Isaac is a son of Abraham" attributes to Isaac. To express this, we first replace the constant "i" by the variable "x" to get an open sentence which expresses the property of being a son of Abraham. Then, to say that everyone has this property, we prefix the universal quantifier "$(\forall x)$." The (closed) sentence "$(\forall x)Sxa$" says that everyone is a son of Abraham.

We write "$(\exists x)Sxa$" to say that Abraham has a son; the sentence attributes to Abraham the property of having a son. How would we attribute this property to everybody, saying that everyone has a son? We want to form an open sentence and prefix a universal quantifier, but we can't do this the way we did before,

by substituting the variable "x" for "a" and prefixing "(∀x)." This would give "(∀x)(∃x)Sxx." But there is nothing about this sentence to indicate that it means "Everyone has a son," rather than "Everyone is a son" or "Someone is his own son." To properly represent "Everyone has a son," we require a second variable. To represent the property of having a son, substitute a *new* variable for "a" in "(∃x)Sxa," getting the open sentence "(∃x)Sxy." To say that everyone has a son, prefix the universal quantifier "(∀y)," getting "(∀y)(∃x)Sxy."

To say "Everyone is a son," take the sentence "(∃x)Sax," which says that Abraham is a son, and substitute "y" for "a," getting an open sentence "(∃x)Syx," which represents the property of being a son. Then prefix the universal quantifier "(∀y)" to get "(∀y)(∃x)Syx."

"Someone is his own son" would naturally be said "(∃x)Sxx." As we shall see below, "(∀x)(∃x)Sxx" also means that someone is his own son, though it's a silly way to say it, since the initial "(∀x)" is entirely superfluous.

We'll symbolize "John loves Mary" as "Ljm," and "John is loved by Mary" as "Lmj." "John loves someone" will be "(∃x)Ljx," and "Everyone loves someone" will be "(∀y)(∃x)Lyx"; "There is somebody who loves someone" is "(∃y)(∃x)Lyx." "John is loved by someone" will be "(∃x)Lxj," while "Everyone is loved by someone" is "(∀y)(∃x)Lxy"; "Somebody is loved by someone" is "(∃y)(∃x)Lxy," which is logically equivalent to the formula that symbolizes "Somebody loves someone." "John loves everyone" is "(∀x)Ljx." "There is someone who loves everyone" is "(∃y)(∀x)Lyx," while "Everyone loves everyone" is "(∀y)(∀x)Lyx." "Mary is loved by everyone" is "(∀x)Lxm." "There is someone who is loved by everyone" is "(∃y)(∀x)Lxy." "Everyone is loved by everyone" is "(∀y)(∀x)Lxy," which is logically equivalent to the sentence that symbolizes "Everyone loves everyone."

"John loves someone who loves Mary" is "(∃x)(Ljx ∧ Lxm)." "Someone loves someone who loves Mary" is "(∃y)(∃x)(Lyx ∧ Lxm)." "Someone loves someone who loves someone" is "(∃z)(∃y)(∃x)(Lyx ∧

Lxz)." "Everyone loves someone who loves Mary" is "(∀y)(∃x)(Lyx ∧ Lxm)."

"John loves everyone who loves everyone" can be paraphrased "For any y, if y loves everyone, John loves y." "y loves everyone" is "(∀x)Lyx," and "John loves y" is "Ljy," so that "John loves everyone who loves everyone" is "(∀y)((∀x)Lyx → Ljy)." "Everyone loves everyone who loves everyone" is "(∀z)(∀y)((∀x)Lyx → Lzy)," while "Someone loves everyone who loves everyone" is "(∃z)(∀y)((∀x)Lyx → Lzy)."

"If John loves anyone, he loves Mary" is "((∃x)Ljx → Ljm)." "Everyone who loves anyone loves Mary" is "(∀y)((∃x)Lyx → Lym)." "John loves himself" is "Ljj." "If John loves anyone, he loves himself" is "((∃x)Ljx → Ljj)." "Everyone who loves anyone loves himself" is "(∀y)((∃x)Lyx → Lyy)."

Variables don't name anything. They indicate places where a name has been taken from a sentence. We have different variables because we have different names, to indicate the places from which the different names have been taken. The variables are interchangeable. "John loves someone" can be written "(∃x)Ljx" or "(∃y)Ljy." "Everyone loves someone" can be written "(∀y)(∃x)Lyx" or "(∀x)(∃y)Lxy" or "∀z)(∃x)Lzx" or (∀z)(∃y)Lzy."

We now give an official description of the formal languages we shall be using. In addition to the familiar "∨," "∧," "→," "↔," "¬," "∀," "∃," "(," and ")," a *language for the predicate calculus* contains symbols of the following sorts:

> ***Individual constants*, usually lowercase letter from the beginning of the alphabet, sometimes with numerical subscripts, which serve the role of proper names in English; an individual constant names an individual.**
>
> ***Variables*, usually lowercase letters from the end of the alphabet, sometimes with numerical subscripts, which serve more-or-less the role of personal pronouns. The individual constants and the variables are referred to as the *individual symbols*.**

> ***One-place predicates***, usually uppercase letters, sometimes with numerical subscripts, which play the roles played by English intransitive verbs, adjectives, and common nouns; 1-place predicates represent properties and actions.
>
> ***Two-place predicates***, usually uppercase letters, sometimes with numerical subscripts, which serve the functions served by transitive verbs and by phrases like "is taller than" and "is a daughter of."
>
> ***Three-place predicates***, usually uppercase letters, sometimes with numerical subscripts, which behave like "______ is between ______ and ______" and "______ gave ______ to ______."
>
> And so on. For each positive number n, there is a category of ***n-place predicates***.

The language has to have infinitely many variables, but, other than that, any of these categories can be empty, though the language has to have at least one predicate. We assume that none of these categories of expressions overlap.

An ***atomic formula*** consists of an n-place predicate followed by n individual symbols. For example, "Lxy," "Lyx," "Ljx," and "Ljm."

The ***formulas*** comprise the smallest class of expressions which

> contains the atomic formulas;
> contains $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, and $(\varphi \leftrightarrow \psi)$, whenever it contains φ and ψ; and
> contains $\neg\varphi$, $(\forall v)\varphi$, and $(\exists v)\varphi$, for each variable v, whenever it contains φ.
>
> Unique Readability Lemma. A formula is built up out of atomic formulas ***in a unique way***.

An occurrence of a variable v in a formula is *bound* iff it occurs within some subformula that begins with either (∀v) or (∃v). If not bound, *free*. A formula with no free variables is called a *sentence*. For example, in "((∀x)Fxy ∨ (∃y)Fxy)," the first two occurrences of "x" are bound and the last one free, while the first occurrence of "y" is free and the other two are bound. Thus the quantifier "(∀x)" at the beginning of "(∀x)Fxy" binds the occurrence of "x" in "Fxy," but leaves the occurrence of "y" free.

Where φ is a formula, v a variable, and t an individual symbol, we write $\varphi^{v}/_{t}$ for the result of replacing each free occurrence of v in φ by t.

The terminology from the monadic predicate calculus – such terms as "disjunction," "molecular formula," and "universal formula" – is carried over directly.

The ideas behind the semantics for the predicate calculus are just the same as the ideas behind the semantics for the monadic predicate calculus, but, because of the presence of the extra variables, the details are quite a bit more complicated. So I want to put it off as long as possible. Hence, before we talk about semantics, let's talk a little bit about translations.

There are two principal principles governing translations. The first is that universal statements in English are generally rendered by a predicate calculus sentence consisting of a universal quantifier followed by a conditional. Thus "All squirrels are mammals" is translated "(∀x)(Sx → Mx)." For "(∀x)(Sx → Mx)" to be true, everything must satisfy "(Sx → Mx)," which is to say that nothing satisfies "Sx" without also satisfying "Mx"; which means that nothing is a squirrel which is not a mammal; which means that "All squirrels are mammals" is true.

"Everyone who loves Jill loves Clarissa" is "(∀x)(Lxj → Lxc).: "Every son of Sara is a son of Abraham" is "(∀x)(Sxs → Sxa)." "Everyone who sings and plays the ukelele has friends" is "(∀x)((Sx ∧ Ux) → (∃y)Fyx)." In each case, we see the pattern, universal quantifier followed by a conditional. "All As are Bs)" is "(∀x)(Ax → Bx)."

Rarely we see a universal quantifier followed by something other than a conditional. "Everyone is either a man, a woman, or a child" is translated by a universal quantifier followed by a disjunction, "$(\forall x)((Mx \vee Wx) \vee Cx)$." "Everyone drank, smoked, and had fun" is a universal quantifier followed by a conjunction, "$(\forall x)((Dx \wedge Sx) \wedge Fx)$." But such cases are exceptional. Whenever your translation is a universal quantifier followed by something other than a conditional, look again, more carefully, for such a translation is seldom right.

The second principal principle is that English existential sentences — that is, English sentences that make existence claims — are symbolized by an existential quantifier followed by a conjunction. "$(\exists x)(Sx \wedge Fx)$" translates "Some squirrels can fly," since it is true if and only if something satisfies "Sx" and "Fx," that is, it is true if and only if there is something that is both a squirrel and a thing that flies.

"Someone loves both Jill and Clarissa" is "$(\exists x)(Lxj \wedge Lxc)$." "Some sons Of Sara are sons of Abraham" is "$(\exists x)(Sxs \wedge Sxa)$." "Some people who play the ukelele have friends" is "$(\exists x)(Ux \wedge (\exists y)Fyx)$." In each case, we see a pattern: existential quantifier followed by conjunction. "Some As are Bs" is "$(\exists x)(Ax \wedge Bx)$."

There are exceptions to this pattern. "There are some who neither drank nor smoked" is "$(\exists x)\neg(Dx \vee Sx)$." But such exceptions are rare.

The only difference between the English sentence "Some squirrels can fly" and "All squirrels can fly" is that the word "some" has been replaced by "all." So we would naturally anticipate that the only difference between their translations is that the existential quantifier "$(\exists x)$" is replaced by the universal quantifier "$(\forall x)$." But this would be a mistake. There are two differences between the sentences. The first is an existential quantifier followed by a conjunction, "$(\exists x)(Sx \wedge Fx)$," whereas the second is a universal quantifier followed by a conditional, "$(\forall x)(Sx \rightarrow Fx)$." Just replacing the existential by the universal quantifier gives "$(\forall x)(Sx \wedge Fx)$," which isn't what we want; it says that everything is a flying squirrel.

When I say "Everyone drank and smoked and had fun," I don't mean that everyone in the whole world drank, smoked, and had fun, I mean that everyone at the party did so. In this particular context, "everyone" means "every person at the party," so, when we translate the sentence, we need to understand "$(\forall x)$" in such a way that $(\forall x)\varphi$ is true if and only if everyone at the party satisfies φ. When I utter the sentence "Everyone drank and smoked and had fun," it's within a context in which it's clear that the only individuals we are interested in are the people at the party. The set of people at the party is said to be the *universe of discourse*, and "$(\forall x)$" is understood in such a way that $(\forall x)\varphi$ is true if and only if every member of the universe of discourse satisfies φ.

When we translate "Some people who play the ukelele have friends" as "$(\exists x)(Ux \wedge (\exists y)Fxy)$," we are understanding "$(\exists x)$" in such a way that $(\exists x)\varphi$ is true if and only if some *people* satisfy φ; that is, we are taking our universe of discourse to consist of people. If we didn't want to limit our horizon in this way, we could take our universe of discourse to include everything and translate the sentence "$(\exists x)(Px \wedge (Ux \wedge (\exists y)Fxy))$."

"All" and "only" are related in the same way as "if" and "only if." "All As are Bs" means "For any x, x is a B if x is an A," "$(\forall x)(Ax \rightarrow Bx)$." "Only As are Bs" means "For any x, x is a B only if x is an A," "$(\forall x)(Bx \rightarrow Ax)$." Thus "All Sola's pups can fly" is "$(\forall x)(Pxs \rightarrow Fx)$," whereas "Only Sola's pups can fly" is "$(\forall x)(Fx \rightarrow Pxs)$." "All and only Sola's pups can fly" can be written as the conjunction "All Sola's pups can fly and only Sola's pups can fly'" "$((\forall x)(Pxs \rightarrow Fx) \wedge (\forall x)(Fx \rightarrow Pxs)$"; or it can be written more concisely as "$(\forall x)(Pxs \leftrightarrow Fx)$." "Of all the dogs that live on Jefferson Street, only Sola's pups can fly" is "For all x, if x is a dog that lives on Jefferson Street, then x can fly only if x is one of Sola's pups," "$(\forall x)((Dx \wedge Jx) \rightarrow (Fx \rightarrow Pxs))$."

The quantifiers interact with the sentential calculus connectives in just the ways you'd expect. "Not every squirrel flies" is the negation of "Every squirrel flies," namely, "$\neg(\forall x)(Sx \rightarrow Fx)$." "No squirrel flies," which is the negation of "Some squirrel flies," is "$\neg(\exists x)(Sx \wedge Fx)$." "Not every squirrel

flies, but some do" is a conjunction, "$(\neg(\forall x)(Sx \rightarrow Fx) \wedge (\exists x)(Sx \wedge Fx))$." "Melissa won't dance unless everyone does" is a conditional "$(\neg(\forall x)Dx\ \neg Dm)$." "Melissa will come to the party if all the boys that hang out at Spike's Place come, but, otherwise, she'll stay home and watch TV" is "$(((\forall x)((Bx \wedge Sx) \rightarrow Px) \rightarrow Pm) \wedge (\neg(\forall x)((Bx \wedge Sx) \rightarrow Px) \rightarrow (Hm \wedge Wm)))$."

The trick to successful translation is to work in stages. For example, using "A," "H," and "C," respectively, to translate "admires," "hires," and "is a chauffeur," let's translate "Everyone who hires a chauffeur is admired by everyone who doesn't hire a chauffeur." The sentence is universal, its translation will consist of a universal quantifier followed by a conditional, "$(\forall x)$(x hires a chauffeur $\rightarrow$ x is admired by everyone who doesn't hire a chauffeur)." "x hires a chauffeur" is existential "$(\exists y)$(y is a chauffeur $\wedge$ x hires y)," that is "$(\exists y)(Cy \wedge Hxy)$." When we rewrite "x is admired by everyone who doesn't hire a chauffeur" in the active voice as "everyone who doesn't hire a chauffeur admires x," we see that it is a universal statement, "$(\forall y)$(y doesn't hire a chauffeur $\rightarrow$ y admires x)." 'y doesn't hire a chauffeur" is a negated existential claim, "$\neg(\exists z)(Cz \wedge Hyz)$." Putting the whole thing together, we get "$(\forall x)((\exists y)(Cy \wedge Hxy) \rightarrow (\forall y)(\neg(\exists z)(Cz \wedge Hzy) \rightarrow Ayx))$."

"Some admire all those who admire themselves, but some admire only those who don't admire themselves" is a conjunction. The first conjunct is existential, "$(\exists x)$x admires all those who admire themselves." "x admires all those who admire themselves" is universal, "for any y, if y admires herself, then x admires y," "$(\forall y)(Ayy \rightarrow Axy)$." The second conjunct is likewise existential, "$(\exists x)$x admires only those who don't admire themselves." "x admires only those who don't admire themselves" is universal, "for any y, x admires y only if y does not admire herself," "$(\forall y)(Axy \rightarrow \neg Ayy)$" Putting the pieces together, $((\exists x)(\forall y)(Ayy \rightarrow Axy) \wedge (\forall y)(Axy \rightarrow \neg Ayy))$."

"Everyone who dated someone who dated either Harry or someone who dated Harry should be tested" is "$(\forall x)$(x dated someone who dated either Harry or someone who dated Harry $\rightarrow$ Tx)." "x dated someone who dated either Harry or someone who dated Harry" is $(\exists y)$(y dated either Harry or someone who dated

Harry $\wedge$ x dated y)." 'y dated either Harry or someone who dated Harry" is a disjunction "(Dyh $\vee$ y dated someone who dated Harry)." "y dated someone who dated Harry" is "$(\exists z)(Dzh \wedge Dyz)$." So the sentence is "$(\forall x)((\exists y)((Dyh \vee (\exists z)(Dzh \wedge Dyz)) \wedge Dxy) \rightarrow Tx)$."

For the most part, translation, while it may be complicated, is pretty straightforward. "Every," "all," and "each" are translated by universal quantifiers (generally followed by conditionals), while "some," "one," "at least one," "there exists," and "there are" are symbolized by existential quantifiers (generally followed by conjunctions).

"Any" presents special problems. In a simple sentence, "any" is universal; "Any dog can fly" is "$(\forall x)(Dx \rightarrow Fx)$." When "any" occurs in the consequent of a conditional, it's again translated "all"; "If Tarmin can fly, any dog can fly" is "$(Ft \rightarrow (\forall x)(Dx \rightarrow Fx))$." When "any" occurs in the antecedent of a conditional and it's paired with a pronoun that appears in the consequent, it's again translated "$(\forall x)$"; "If any dog can fly, she is exceptionally agile" is "$(\forall x)((Dx \wedge Fx) \rightarrow Ax)$." On the other hand, when "any" occurs in the antecedent of a conditional and it doesn't bind a quantifier that appears in the consequent, it's most natural to translate it as existential. "If any dog can fly, Tarmin can" is "$((\exists x)(Dx \wedge Fx) \rightarrow Ft)$." Contrast "If any dog can fly, Tarmin can" with "If every dog can fly, Tarmin can," translated "$((\forall x)(Dx \rightarrow Fx) \rightarrow Tx)$." You'd say the former if you believed that Tarmin is such a clever and agile dog that she'd learn to fly if any dog would; the latter is true just because Tarmin is a dog.

Sometimes, in complicated sentences, you can get "any" translated both as existential and as universal. "Any dog who chases any dog who chases any rabbit will be put in the pound" is "$(\forall x)((Dx \wedge (\exists y)((Dy \wedge (\exists z)(Rz \wedge Cyz)) \wedge C)) \rightarrow Px)$." Contrast it with the sentence "Some dog who chases some dog who chases some rabbit will be put in the pound," which is $(\exists x)((Dx \wedge (\exists y)((Dy \wedge (\exists z)(Rz \wedge Cyz) \wedge Cxy) \wedge Px)$," and with "Every dog who chases every dog that chases every rabbit will be put in the pound," which is "$(\forall x)((Dx \wedge (\forall y)((Dy \wedge (\forall z)(Rz \rightarrow Cyz)) \rightarrow Cxy)) \rightarrow Px)$."

Bertrand Russell's work suggests an admirably simple rule for translating "any": "Any" should always be translated by an initial universal quantifier; if there are several "any"s they should be translated by a block of initial universal quantifiers. Thus "If Tarmin can fly, any dog can" would be "$(\forall x)(Ft \rightarrow (Dx \rightarrow Fx))$," which is, as we shall see below, logically equivalent to "$(Ft \rightarrow (\forall x)(Dx \rightarrow Fx))$." "If any dog can fly, Tarmin can" should be "$(\forall x)((Dx \wedge Fx) \rightarrow Ft)$," which is logically equivalent to "$((\exists x)(Dx \wedge Fx) \rightarrow Ft)$." "Any dog who chases any dog who chases any rabbit will be put in the pound" is "$(\forall x)(\forall y)(\forall z)(((Dx \wedge (Dy \wedge (Rz \wedge Cyz))) \wedge Cxy) \rightarrow Px)$," which is logically equivalent to the symbolization we got before.

The way Russell thought about "any" isn't quite the way I've been describing. Russell thought that the word "any" was a schematic term, which could be filled in any way you liked. Thus the word "any" shouldn't be symbolized by a quantifier at all. "Any" should be symbolized by a free variable and a sentence containing "any" should be symbolized by an open sentence." "Any dog can fly" should be symbolized by the open sentence "(x is a dog $\rightarrow$ x can fly)," which is to be understood in the same way we understand the trigonometric law "$\tan^2 \theta + 1 = \sec^2 \theta$": any way you choose to replace θ with a numerical expression, you'll get a true sentence. But it's simpler and more straightforward to think of the trigonometric law as tacitly universally quantifier, "For any angle θ, $\tan^2 \theta + 1 = \sec^2 \theta$," and, likewise, to symbolize sentences containing "any" by the universal closure* of Russell's schema.

Russell's method doesn't work for "just any." "Not just anyone can join the Branded Peasant Club" is "$\neg(\forall x)Jx$," rather than "$(\forall x)\neg Jx$." "If just anyone can join the Branded Peasant Club, I wouldn't want to be a member" is $((\forall x)Jx \rightarrow \neg Wi)$," rather than "$(\forall x)(Jx \rightarrow \neg Wi)$."

A way to translate "any" that I find easier is to take advantage of the fact that English speakers have, from a very early age, an exquisitely well tuned ear for recognizing when two English sentences mean the same thing. Thus to translate a

---

* A universal closure of a formula is a sentence gotten by prefixing universal quantifiers.

sentence containing "any," try substituting "every" for "any," and see if the resulting sentence means the same thing; if so, translate "any" as "$\forall$." If not, try substituting "at least one"; when "any" means the same as "at least one," it should be translated "$\exists$." Thus "Any dog can fly" means the same as "Every dog can fly," so it's translated "$(\forall x)(Dx \rightarrow Fx)$." Similarly, "If Tarmin can fly, any dog can" means the same as "If Tarmin can fly, every dog can," translated "$(Ft \rightarrow (\forall x)(Dx \rightarrow Fx))$." On the other hand, "If any dog can fly, Tarmin can" doesn't mean "If every dog can fly, Tarmin can"; it means "If at least one dog can fly, Tarmin can," so it's translated "$((\exists x)(Dx \wedge Fx) \rightarrow Ft)$." "Any dog who chases any dog who chases any cat will be put into the pound" is "Every dog who chases at least one dog who chases at least one rabbit will be put into the pound," "$(\forall x)((Dx \wedge (\exists y)((Dy \wedge (\exists z)(Rz \wedge Cyz)) \wedge Cxy)) \rightarrow Px)$."

Now, having put it off as long as possible, we turn to the semantics of the formal language. An *interpretation* is a function $\mathcal{A}$, assigning a value to "$\forall$," to each individual constant, and each predicate, so that

> **$\mathcal{A}$("$\forall$"), also written $|\mathcal{A}|$, is a nonempty set.**
> **$\mathcal{A}$(c), also written $c^{\mathcal{A}}$, is an element of $|\mathcal{A}|$, for each individual constant c.**
> **$\mathcal{A}$(R), also written $R^{\mathcal{A}}$, is a set of n-tuples from $|\mathcal{A}|$, for each n-place predicate R.**

[Here we identify an individual with its 1-tuple, so that <Socrates> = Socrates; if R is a 1-place predicate, $R^{\mathcal{A}}$ will be a subset of $|\mathcal{A}|$.]

The semantics for the full predicate calculus proceeds in basically the same way as the semantics for the monadic predicate calculus, though it's a little more complicated because of the presence of the extra variables. Instead of talking about a single individual satisfying a formula, we talk about satisfaction by a function assigning an individual to each of the variables. For example, if "S" represents the son-of relation, then a function which assign Isaac to "x" and Abraham to "y" will satisfy "Sxy."

A ***variable assignment*** for $\mathcal{A}$ is a function which assigns an element of $|\mathcal{A}'|$ to each of the variables. If σ is a variable assignment, define $Den_{\sigma,\mathcal{A}}(t)$, for each individual symbol t by

$$Den_{\sigma,\mathcal{A}}(t) = \mathcal{A}(t) \text{ if t is an individual constant}$$
$$= \sigma(t) \text{ if t is a variable}$$

If φ is an atomic formula of the form $Rt_1...t_n$ and σ is a variable assignment for an interpretation $\mathcal{A}$, we say that σ satisfies φ under $\mathcal{A}$ iff the n-tuple $<Den_{\sigma,\mathcal{A}}(t_1),...,Den_{\sigma,\mathcal{A}}(t_n)>$ is an element of $\mathcal{A}(R)$.

For example, if σ("x") is Isaac, σ("y") is Abraham, and σ("z") is Sara and if $\mathcal{A}$("S") is the son-of relation while $\mathcal{A}$("a") is Abraham's father Abram, we see that σ satisfies "Sxy," "Sxz," and "Sya," whereas it doesn't satisfy "Syx," "Sax," or "Saa."

σ satisfies a disjunction under $\mathcal{A}$ iff σ satisfies one or both disjuncts under $\mathcal{A}$.

σ satisfies a conjunction under $\mathcal{A}$ iff σ satisfies both conjuncts under $\mathcal{A}$.

σ satisfies a conditional under $\mathcal{A}$ iff either σ satisfies the consequent under $\mathcal{A}$ or σ fails to satisfy the antecedent under $\mathcal{A}$.

σ satisfies a biconditional under $\mathcal{A}$ iff either σ satisfies both components under $\mathcal{A}$ or σ satisfies neither component under $\mathcal{A}$.

σ satisfies a negation under $\mathcal{A}$ iff σ does not satisfy the negatum under $\mathcal{A}$.

Before we can give the condition of satisfaction for an initially quantified sentence, we need a definition: Where v is a variable and σ is a variable assignment for an interpretation $\mathcal{A}$, a ***v-variant*** of σ is a variable assignment which assigns the same value σ assigns to every variable other than v. Thus, if ρ is a v-variant of σ, then the only place σ and ρ might disagree is in what they assign to v; we express this by writing $\rho \approx_{v} \sigma$.

> σ satisfies $(\forall v)\varphi$ under $\mathcal{A}$ iff every v-variant of σ satisfies φ under $\mathcal{A}$.
>
> σ satisfies $(\exists v)\varphi$ under $\mathcal{A}$ iff at least one v-variant of σ satisfies φ under $\mathcal{A}$.

Example. Consider the language whose predicates are a 1-place predicate "R" and a 2-place predicate "E," and whose only individual constant is "j." Define an interpretation $\mathcal{A}$ for the language by stipulating:

> $|\mathcal{A}|$ = {U.S. presidents, Washington through Clinton}
> $\mathcal{A}$("E") = {<x,y>: x and y are presidents and x first took office earlier than y}
> $\mathcal{A}$("R") = {Republican presidents}
> $\mathcal{A}$("j') = Andrew Jackson

Let σ be the following variable assignment for $\mathcal{A}$:

> σ("x") = George Washington
> σ("y") = Abraham Lincoln
> σ("z") = Richard Nixon
> σ(every other variable) = Harry Truman

Thus σ satisfies "Exy" because Washington took office before Lincoln." σ doesn't satisfy "Ezw" because Nixon was later than Truman. σ satisfies the conditional "(Ezw → Rw)" because it fails to satisfy the antecedent. σ doesn't satisfy "(Eyz → Rj)," since it satisfies the antecedent – Lincoln was earlier than Nixon – but it doesn't satisfy the consequent – Jackson wasn't a Republican.

Now consider the variable assignment ρ with

> ρ("x") = George Washington
> ρ("y") = Ronald Reagan
> ρ("z") = Richard Nixon
> ρ(every other variable) = Harry Truman

Thus σ and ρ agree on the value they assign to every variable other than "y," so that ρ is a "y"-variant of σ. ρ satisfies "(Ry $\wedge$ Ezy)," since Reagan was a Republican who took office after Nixon. Since ρ is a "y"-variant of σ that satisfies "(Ry $\wedge$ Ezy)," σ satisfies "($\exists$y)(Ry $\wedge$ Ezy)." σ doesn't satisfy "(Ry $\wedge$ Ezy)," since Nixon took office later than Lincoln. Hence σ satisfies "(($\exists$y)(Ry $\wedge$ Ezy) $\wedge$ ¬(Ry $\wedge$ Ezy))."

Let τ be an "x"-variant of σ. Then τ("y") = σ("y") = Lincoln. We don't know who τ("x") is, but, whoever τ("x") is, we know that τ won't satisfy "(Rx $\wedge$ Exy)," because there were no Republican presidents earlier than Lincoln. Since every "x"-variant of σ satisfies "¬(Rx $\wedge$ Exy," σ satisfies "($\forall$x)¬(Rx $\wedge$ Exy)."

Let μ be the variable assignment given by

μ("x") = Bill Clinton
μ("y") = Abraham Lincoln
μ("z") = Richard Nixon
μ(every other variable) = Harry Truman

Thus μ is an "x"-variant of σ. Let ν be a "y"-variant of μ. Even without knowing who ν("y") is, we can be confident that ν satisfies "(Ry $\rightarrow$ Eyx)," since, if ν("y") is any member of |$\mathcal{A}$| other than Clinton, ν will satisfy the consequent, whereas, if ν("y") happens to be Clinton, ν will fail to satisfy the antecedent. Since every "y"-variant of μ satisfies "(Ry $\rightarrow$ Eyx)," μ satisfies "($\forall$y)(Ry $\rightarrow$ Eyx)." Since μ is an "x"-variant of σ which satisfies "($\forall$y)(Ry $\rightarrow$ Eyx)," we see that σ satisfies "($\exists$x)($\forall$y)(Ry $\rightarrow$ Eyx)." Indeed, every variable assignment satisfies "($\exists$x)($\forall$y)(Ry $\rightarrow$ Eyx)." "($\exists$x)($\forall$y)(Ry $\rightarrow$ Eyx)" is a true sentence and, as we shall now see, a true sentence is satisfied by every variable assignment, whereas a false sentence is satisfied by nothing.⊠

> **Lemma on Irrelevant Variables. If σ and ρ are variable assignments for $\mathcal{A}$ which assign the same values to all the variables that occur free in φ, then σ satisfies φ under $\mathcal{A}$ iff ρ does.**

The proof proceeds in the usual way. We let Σ be the set of formulas φ such that, whenever σ and ρ are variable assignments that agree on the values they assign the variables that occur free in φ, σ satisfies φ iff ρ satisfies φ. Then we show that Σ contains the atomic formulas and that it's closed under disjunction, conjunction, negation, forming conditionals, forming biconditionals, universal quantification, and existential quantification. The proof is so similar to the proof that the Law of Bivalence holds for the monadic predicate calculus that there's no point in going through it again here.☒

> **Definition. A sentence is *true* under $\mathcal{A}$ iff it is satisfied under $\mathcal{A}$ by every variable assignment for $\mathcal{A}$. A sentence is *false* under $\mathcal{A}$ iff it is satisfied by no variable assignment for $\mathcal{A}$.**

The Lemma on Irrelevant Variables has the following immediate consequences:

> **Principle of Bivalence. Under a given interpretation, every sentence is either true or false.**

> **Corollary. For any sentence φ, interpretation $\mathcal{A}$, and variable assignment σ for $\mathcal{A}$, φ is true under iff φ is satisfied by σ under $\mathcal{A}$.**

> **Corollary. A universal sentence $(\forall v)\varphi$ is true under $\mathcal{A}$ iff every variable assignment for $\mathcal{A}$ satisfies φ under $\mathcal{A}$. An existential sentence $(\exists v)\varphi$ is true under $\mathcal{A}$ iff some variable assignment satisfies φ under $\mathcal{A}$.**

If φ is a formula with only the variable v free, we can continue to talk about an individual satisfying φ, just as we did when we were working on the monadic predicate calculus. Where is an interpretation and *a* a member of $|\mathcal{A}|$, we will say that *a satisfies* φ under $\mathcal{A}$ to mean that every variable assignment σ with $\sigma(v) = a$ satisfies φ under $\mathcal{A}$. According to the Lemma on Irrelevant Variables, this will hold whenever there is even one variable assignment with $\sigma(v) = a$ that satisfies φ.

The proofs of the following results are virtually unchanged from the monadic predicate calculus:

> **Substitution Principle.** If $\mathcal{A}(c) = \sigma(v)$, then σ satisfies $\varphi^{v}/_{c}$ under $\mathcal{A}$ iff σ satisfies φ under $\mathcal{A}$.

> **Locality Principle.** Let $\mathcal{A}$ and $\mathcal{B}$ be two interpretations which have the same universe of discourse and which assign the same values to all the individual constants and predicates that occur within the formula φ. Then, for any variable assignment σ, σ satisfies φ under $\mathcal{A}$ iff σ satisfies φ under $\mathcal{B}$.

While most of our results from the monadic predicate calculus carry over to the full predicate calculus, not everything is the same. Thus, we saw that every consistent MPC sentence is true in an interpretation with a finite universe. The same is not true for the full predicate calculus. Thus consider the following sentence:

$$((\forall x)(\exists y)Lxy \wedge ((\forall x)(\forall y)(\forall z)((Lxy \wedge Lyz) \rightarrow Lxz) \wedge (\forall x)\neg Lxx))$$

The sentence is certainly consistent, as we can see from considering an interpretation $\mathcal{A}$ with $|\mathcal{A}|$ = {natural numbers} and $\mathcal{A}$("L") = the less-than relation on the natural numbers. But the sentence isn't true under any interpretation with a finite universe. To see this, take an interpretation $\mathcal{B}$ under which the sentence is true. Take $a_0 \in |\mathcal{B}|$. Because "$(\forall x)(\exists y)Lxy$" is true under $\mathcal{B}$, there must exist an element a_1 of $|\mathcal{B}|$ such that $\langle a_0,a_1\rangle \in \mathcal{B}$("L"); because "$(\forall x)\neg Lxx$" is true under $\mathcal{B}$, a_1 must be different from a_0. Because "$(\forall x)(\exists y)Lxy$" is true under $\mathcal{B}$, there must exist an element a_2 of $|\mathcal{B}|$ with $\langle a_1,a_2\rangle \in \mathcal{B}$("L"). Because the sentence "$(\forall x)(\forall y)(\forall z)((Lxy \wedge Lyz) \rightarrow Lxz)$" is true in $\mathcal{B}$, $\langle a_0,a_2\rangle$ must be in $\mathcal{B}$("L"). It follows from the fact that "$(\forall x)\neg Lxx$" is true in $\mathcal{B}$ that a_2 is distinct from both a_0 and a_1. Because "$(\forall x)(\exists y)Lxy$" is true in $\mathcal{B}$, there must exist an element a_3 of $|\mathcal{B}|$ such that $\langle a_2,a_3\rangle \in \mathcal{B}$("L"). Because "$(\forall x)(\forall y)(\forall z)((Lxy \wedge Lyz) \rightarrow Lxz)$" is true in $\mathcal{B}$, $\langle a_1,a_3\rangle$ and $\langle a_0,a_3\rangle$ must both be in $\mathcal{B}$("L"). Since "$(\forall x)\neg Lxx$" is true in , a_3 must be distinct from

a_0, from a_1, and from a_2. And so on. We find a_4 with $\langle a_3, a_4`\rangle \in \mathcal{B}(\text{"L"})$ and we show that a_4 is different from a_0, a_1, a_2, and a_3. Then we find a_5 with $\langle a_4, a_5\rangle \in \mathcal{B}(\text{"L"})$ and we show that a_5 is different from a_0, a_1, a_2, a_3, and a_4. The process continues forever, so we conclude that $|\mathcal{B}|$ must be infinite.

For the monadic predicate calculus, we found that there was a mechanical procedure for testing whether a given sentence is valid, though the method, we must admit, was too cumbersome to be of much practical use. One of the fundamental results in the theory of computability is that there is no such procedure for the full predicate calculus:

> Church's Theorem. There can be no algorithm for testing whether a given sentence is valid in the predicate calculus.

What an amazing result! The theorem tells us, not merely that, as a matter of fact, right now today no one has written down an algorithm for testing validity in the predicate calculus. It tells us that there will never be such an algorithm, no matter how clever people become in the future. Church's Theorem is a fundamental limitation, like the facts that no one will ever travel faster than light and no one will ever carry out measurements more precise than permitted by the Heisenberg Uncertainty Principle. Unfortunately, we won't be able to discuss Church's Theorem more fully here.

A *decision procedure* for validity in the predicate calculus would be a mechanical procedure by which, for a given sentence, we can test, one way or another, whether the sentence is valid. Church’s Theorem tells us that there is no such procedure. There is, however, a *proof procedure*, a method by which any sentence that is valid can be shown to be valid. Namely, we derive the sentence from the empty set, using the full-predicate-calculus version of the rules of derivation we learned for the MPC. If a sentence is valid, we can derive it form the empty set, and, if we can derive a sentence from the empty set, it is valid. So if a sentence is valid, we have a way of showing it's valid. The trouble is that an invalid sentence is invalid. If a sentence is invalid, we won't be able to prove it. But the fact that we haven't been able to produce a proof for a given sentence doesn't show that the sentence is unprovable. Maybe there is a proof, but we haven't been clever enough or patient enough to produce it.

The fact that we have worked for seventy-two hours – or seventy-two years – at trying to prove a sentence without success doesn't prove that the sentence isn't provable, so it doesn't prove that the sentence isn't valid.

For validity in the full predicate calculus, we have no decision procedure, but we have a proof procedure. We'll take that up in the next chapter. In the meantime, we'll learn a little more about the semantics of the predicate calculus.

We've already stipulated that two sentences are logically equivalent iff they are true under precisely the same interpretations. Let us now generalize this notion to formulas that contain free variables.

> Definition. Two formulas are *materially equivalent* with respect to an interpretation $\mathcal{A}$ just in case they are satisfied in $\mathcal{A}$ by precisely the same variable assignments for $\mathcal{A}$.
>
> Proposition. Suppose that φ_1 and φ_2 are materially equivalent with respect to $\mathcal{A}$. Let ψ_1 and ψ_2 be two formulas that are alike except that ψ_1 contains φ_1 as a subformula at some places where ψ_2 contains φ_2. Then ψ_1 and ψ_2 are materially equivalent with respect to $\mathcal{A}$.

Proof: Suppose not. Let ψ_1 be a simplest formula such that there exists a formula ψ_2 which is just like ψ_1 except that ψ_1 contains φ_1 at some places where ψ_2 contains φ_2 such that ψ_1 and ψ_2 are not materially equivalent with respect to $\mathcal{A}$. ψ_1 must not be equal to φ_1, since if it were, ψ_2 would be identical to φ_2 and, by hypothesis, φ_1 and φ_2 are materially equivalent with respect to $\mathcal{A}$. So ψ_1 must contain φ_1 as a proper part. That means that ψ_1 can't be atomic, since atomic formulas don't have any formulas as proper parts. There are seven other possibilities:

ψ_1 is a conjunction, say $(\chi_1 \wedge \theta_1)$. Then ψ_2 has the form $(\chi_2 \wedge \theta_2)$, where either $\chi_2 = \chi_1$ or else they differ in that χ_1 contains φ_1 at some or more places where χ_2 contains φ_2. Similarly for θ_2. Because ψ_1 is a simplest formula for which the theorem we're trying to prove fails, we know that χ_1 is materially equivalent to χ_2 and

that θ_1 is materially equivalent to θ_2. For any variable assignment σ, we have:

> **σ satisfies $(\chi_1 \wedge \theta_1)$ with respect to $\mathcal{A}$**
> **iff σ satisfies both χ_1 and θ_1 with respect to $\mathcal{A}$**
> **iff σ satisfies both χ_2 and θ_2 with respect to $\mathcal{A}$**
> **iff σ satisfies $(\chi_2 \wedge \theta_2)$ with respect to $\mathcal{A}$.**

The other cases are similar.<u>X</u>

> **Definition. Two formulas are *logically equivalent* iff they are materially equivalent with respect to every interpretation.**

> **Corollary. Suppose that φ_1 and φ_2 are logically equivalent, and let ψ_1 and ψ_2 be two formulas that are alike except that ψ_1 contains φ_1 as a subformula at some places where ψ_2 contains φ_2. Then ψ_1 and ψ_2 are logically equivalent.**

> **Definition. A formula is said to be in *prenex form* if its quantifiers (if it has any) all occur at the very beginning of the formula, followed by a formula that is quantifier-free.**

> **Theorem. For any formula, there is a logically equivalent formula in prenex form.**

Proof: For a given formula, perform the following operations. First, replace all subformulas of the form $(\psi \leftrightarrow \chi)$ by $((\psi \rightarrow \chi) \wedge (\chi \rightarrow \psi))$. This will still give you a formula logically equivalent to the one you started with. Next, change the bound variables so that no variable occurs both free and bound within the formula, and so that no variable occurs within a quantifier more than once. A change of bound variables will leave you with a formula equivalent to the one you started with. Finally, repeatedly apply the following lemma to pull all the quantifiers to the front, working from the outside in:

> **Lemma. Suppose that the variable v doesn't appear in the formula ψ. For any formula φ we have:**
>
> **$((\forall v)\varphi \wedge \psi)$ is logically equivalent to $(\forall v)(\varphi \wedge \psi)$.**

$(\psi \wedge (\forall v)\varphi)$ is logically equivalent to $(\forall v)(\psi \wedge \varphi)$.
$((\exists v)\varphi \wedge \psi)$ is logically equivalent to $(\exists v)(\varphi \wedge \psi)$.
$(\psi \wedge (\exists v)\varphi)$ is logically equivalent to $(\exists v)(\psi \wedge \varphi)$.
$((\forall v)\varphi \vee \psi)$ is logically equivalent to $(\forall v)(\varphi \vee \psi)$.
$(\psi \vee (\forall v)\varphi)$ is logically equivalent to $(\forall v)(\psi \vee \varphi)$.
$((\exists v)\varphi \vee \psi)$ is logically equivalent to $(\exists v)(\varphi \vee \psi)$.
$(\psi \vee (\exists v)\varphi)$ is logically equivalent to $(\exists v)(\psi \vee \varphi)$.
$((\forall v)\varphi \rightarrow \psi)$ is logically equivalent to $(\exists v)(\varphi \rightarrow \psi)$.
$(\psi \rightarrow (\forall v)\varphi)$ is logically equivalent to $(\forall v)(\psi \rightarrow \varphi)$.
$((\exists v)\varphi \rightarrow \psi)$ is logically equivalent to $(\forall v)(\varphi \rightarrow \psi)$.
$(\psi \rightarrow (\exists v)\varphi)$ is logically equivalent to $(\exists v)(\psi \rightarrow \varphi)$.
$\neg(\forall v)\varphi$ is logically equivalent to $(\exists v)\neg\varphi$.
$\neg(\exists v)\varphi$ is logically equivalent to $(\forall v)\neg\varphi$.

Example: Let's find a prenex equivalent to the formula "$((\forall x)(\exists y)Fxy \leftrightarrow (\exists x)Fzx)$." First, we get rid of the "$\leftrightarrow$":

$$(((\forall x)(\exists y)Fxy \rightarrow (\exists x)Fzx) \wedge ((\exists x)Fzx \rightarrow (\forall x)(\exists y)Fxy))$$

Next, we change bound variables:

$$(((\forall x)(\exists y)Fxy \rightarrow (\exists w)Fzw) \wedge ((\exists v)Fzv \rightarrow (\forall u)(\exists t)Fut))$$

Now we start moving the quantifiers forward:

$$((\exists x)((\exists y)Fxy \rightarrow (\exists w)Fzw) \wedge ((\exists v)Fzv \rightarrow (\forall u)(\exists t)Fut))$$

$$((\exists x)(\forall y)(Fxy \rightarrow (\exists w)Fzw) \wedge ((\exists v)Fzv \rightarrow (\forall u)(\exists t)Fut))$$

$$((\exists x)(\forall y)(\exists w)(Fxy \rightarrow Fzw) \wedge ((\exists v)Fzv \rightarrow (\forall u)(\exists t)Fut))$$

$((\exists x)(\forall y)(\exists w)(Fxy \to Fzw) \wedge (\forall v)(Fzv \to (\forall u)(\exists t)Fut))$

$((\exists x)(\forall y)(\exists w)(Fxy \to Fzw) \wedge (\forall v)(\forall u)(Fzv \to (\exists t)Fut))$

$((\exists x)(\forall y)(\exists w)(Fxy \to Fzw) \wedge (\forall v)(\forall u)(\exists t)(Fzv \to Fut))$

$(\exists x)((\forall y)(\exists w)(Fxy \to Fzw) \wedge (\forall v)(\forall u)(\exists t)(Fzv \to Fut))$

$(\exists x)(\forall y)((\exists w)(Fxy \to Fzw) \wedge (\forall v)(\forall u)(\exists t)(Fzv \to Fut))$

$(\exists x)(\forall y)(\exists w)((Fxy \to Fzw) \wedge (\forall v)(\forall u)(\exists t)(Fzv \to Fut))$

$(\exists x)(\forall y)(\exists w)(\forall v)((Fxy \to Fzw) \wedge (\forall u)(\exists t)(Fzv \to Fut))$

$(\exists x)(\forall y)(\exists w)(\forall v)(\forall u)((Fxy \to Fzw) \wedge (\exists t)(Fzv \to Fut))$

$(\exists x)(\forall y)(\exists w)(\forall v)(\forall u)(\exists t)((Fxy \to Fzw) \wedge (Fzv \to Fut))$

There were lots of other possibilities for the order in which we chose to bring the quantifiers forward, so there were lots of other answers we might have gotten here (all logically equivalent, of course.)X