Predicate Logic

Dave Dubin

February 6, 2017

Last week: propositional logic

- The propositional logic introduced last week used lower case letters to represent propositions: things that could be true or false.

Last week: propositional logic

- The propositional logic introduced last week used lower case letters to represent propositions: things that could be true or false.
- For example r might stand for the proposition "Romeo loves Juliet," and o might stand for "Othello loves Iago."

Last week: propositional logic

- The propositional logic introduced last week used lower case letters to represent propositions: things that could be true or false.
- For example r might stand for the proposition “Romeo loves Juliet,” and o might stand for “Othello loves Iago.”
- The proposition letters are joined by operators and parentheses according to the rules of a formal grammar to make logical expressions. For example, the expression $(r \wedge \neg o)$ would mean “Romeo loves Juliet, but Othello does not love Iago.”

Semantics of propositional logic

- Logic expressions also have truth values, but only with respect to particular truth value assignments to the proposition letters.

Semantics of propositional logic

- Logic expressions also have truth values, but only with respect to particular truth value assignments to the proposition letters.
- We model states of affairs two ways: first as a group of proposition letters with truth values assigned to them. For example, there are four possible states of affairs for the two propositions r and o.

Semantics of propositional logic

- Logic expressions also have truth values, but only with respect to particular truth value assignments to the proposition letters.
- We model states of affairs two ways: first as a group of proposition letters with truth values assigned to them. For example, there are four possible states of affairs for the two propositions r and o.
- The second way of understanding states of affairs is as valuations: functions from expressions to truth values. For example, there is only one of the four states of affairs that maps the expression $(r \wedge \neg o)$ to *true*. That is to say, only one of those states of affairs models that expression.

Predicate logic expressions

- Predicate logic (also called first order logic) uses the same parentheses and operators as propositional logic. But letters are used in three different ways.

Predicate logic expressions

- Predicate logic (also called first order logic) uses the same parentheses and operators as propositional logic. But letters are used in three different ways.
- Lower case letters from the beginning of the Latin alphabet represent specific individual things in the domain we're modeling. Think of them like proper names.

Predicate logic expressions

- Predicate logic (also called first order logic) uses the same parentheses and operators as propositional logic. But letters are used in three different ways.
- Lower case letters from the beginning of the Latin alphabet represent specific individual things in the domain we're modeling. Think of them like proper names.
- Lower case letters from the end of the alphabet (like x and y) are variables that can denote different individuals under different assignments—just like variables in algebraic expressions.

Predicate logic expressions

- Predicate logic (also called first order logic) uses the same parentheses and operators as propositional logic. But letters are used in three different ways.
- Lower case letters from the beginning of the Latin alphabet represent specific individual things in the domain we're modeling. Think of them like proper names.
- Lower case letters from the end of the alphabet (like x and y) are variables that can denote different individuals under different assignments—just like variables in algebraic expressions.
- Capital letters represent properties that an individual might have, classes they might belong to, or relations they might stand in. Think of them like relations in a relational database.

Predicate logic expressions

- Predicate logic (also called first order logic) uses the same parentheses and operators as propositional logic. But letters are used in three different ways.
- Lower case letters from the beginning of the Latin alphabet represent specific individual things in the domain we're modeling. Think of them like proper names.
- Lower case letters from the end of the alphabet (like x and y) are variables that can denote different individuals under different assignments—just like variables in algebraic expressions.
- Capital letters represent properties that an individual might have, classes they might belong to, or relations they might stand in. Think of them like relations in a relational database.
- Finally we have two new *quantifiers*: the symbol $\forall$ is read "for all" and $\exists$ is read "there exists."

Examples of predicate logic expressions

- Predicates take a particular number of arguments, and the order matters. Let Lxy stand for the binary predicate "x loves y," Vx stand for the unary predicate "x is a lover," and the propositional constants r, j, o, d, i stand for Romeo, Juliet, Othello, Desdemona, and Iago, respectively.

Examples of predicate logic expressions

- Predicates take a particular number of arguments, and the order matters. Let Lxy stand for the binary predicate "x loves y," Vx stand for the unary predicate "x is a lover," and the propositional constants r, j, o, d, i stand for Romeo, Juliet, Othello, Desdemona, and Iago, respectively.
- Lrj means "Romeo loves Juliet."

Examples of predicate logic expressions

- Predicates take a particular number of arguments, and the order matters. Let Lxy stand for the binary predicate "x loves y," Vx stand for the unary predicate "x is a lover," and the propositional constants r, j, o, d, i stand for Romeo, Juliet, Othello, Desdemona, and Iago, respectively.
- Lrj means "Romeo loves Juliet."
- $(Lrj \wedge \neg Loi)$ means "Romeo loves Juliet, but Othello doesn't love Iago."

Examples of predicate logic expressions

- Predicates take a particular number of arguments, and the order matters. Let Lxy stand for the binary predicate "x loves y," Vx stand for the unary predicate "x is a lover," and the propositional constants r, j, o, d, i stand for Romeo, Juliet, Othello, Desdemona, and Iago, respectively.
- Lrj means "Romeo loves Juliet."
- $(Lrj \land \neg Loi)$ means "Romeo loves Juliet, but Othello doesn't love Iago."
- $\forall x Lxd$ means "everyone loves Desdemona"

Examples of predicate logic expressions

- Predicates take a particular number of arguments, and the order matters. Let Lxy stand for the binary predicate "x loves y," Vx stand for the unary predicate "x is a lover," and the propositional constants r, j, o, d, i stand for Romeo, Juliet, Othello, Desdemona, and Iago, respectively.
- Lrj means "Romeo loves Juliet."
- $(Lrj \wedge \neg Loi)$ means "Romeo loves Juliet, but Othello doesn't love Iago."
- $\forall x Lxd$ means "everyone loves Desdemona"
- $\neg\exists x Lix$ means "Iago loves no one."

Examples of predicate logic expressions

- Predicates take a particular number of arguments, and the order matters. Let Lxy stand for the binary predicate "x loves y," Vx stand for the unary predicate "x is a lover," and the propositional constants r, j, o, d, i stand for Romeo, Juliet, Othello, Desdemona, and Iago, respectively.
- Lrj means "Romeo loves Juliet."
- $(Lrj \wedge \neg Loi)$ means "Romeo loves Juliet, but Othello doesn't love Iago."
- $\forall x Lxd$ means "everyone loves Desdemona"
- $\neg \exists x Lix$ means "Iago loves no one."
- $\forall x \forall y (Vx \rightarrow Lyx)$ means "everyone loves a lover."

Examples of predicate logic expressions

- Predicates take a particular number of arguments, and the order matters. Let Lxy stand for the binary predicate "x loves y," Vx stand for the unary predicate "x is a lover," and the propositional constants r, j, o, d, i stand for Romeo, Juliet, Othello, Desdemona, and Iago, respectively.
- Lrj means "Romeo loves Juliet."
- $(Lrj \land \neg Loi)$ means "Romeo loves Juliet, but Othello doesn't love Iago."
- $\forall x Lxd$ means "everyone loves Desdemona"
- $\neg \exists x Lix$ means "Iago loves no one."
- $\forall x \forall y (Vx \rightarrow Lyx)$ means "everyone loves a lover."
- $\forall x (Vx \leftrightarrow \exists z Lxz)$ means "a lover is someone who loves."

Quantifiers have scope

- The scope of a quantifier consists of the logical expression immediately following it. This means that one quantifier can be within the scope of another.

Quantifiers have scope

- The scope of a quantifier consists of the logical expression immediately following it. This means that one quantifier can be within the scope of another.
- Define Cx, Px, and Rxy as meaning x is a child, x is a pony, and x rode y, respectively.

Quantifiers have scope

- The scope of a quantifier consists of the logical expression immediately following it. This means that one quantifier can be within the scope of another.
- Define Cx, Px, and Rxy as meaning x is a child, x is a pony, and x rode y, respectively.
- The ambiguous English sentence, “Every child was riding a pony” expresses two different propositions.

Quantifiers have scope

- The scope of a quantifier consists of the logical expression immediately following it. This means that one quantifier can be within the scope of another.
- Define Cx, Px, and Rxy as meaning x is a child, x is a pony, and x rode y, respectively.
- The ambiguous English sentence, "Every child was riding a pony" expresses two different propositions.
- We can express the first in logical form as $\exists x(Px \wedge \forall y(Cy \rightarrow Ryx))$. On this interpretation, there is some particular pony (or ponies) that every child rode.

Quantifiers have scope

- The scope of a quantifier consists of the logical expression immediately following it. This means that one quantifier can be within the scope of another.
- Define Cx, Px, and Rxy as meaning x is a child, x is a pony, and x rode y, respectively.
- The ambiguous English sentence, "Every child was riding a pony" expresses two different propositions.
- We can express the first in logical form as $\exists x(Px \wedge \forall y(Cy \to Ryx))$. On this interpretation, there is some particular pony (or ponies) that every child rode.
- We can express the second as $\forall y(Cy \to \exists x(Px \wedge Ryx))$. On this interpretation, every child was riding some pony, but no particular pony was necessarily ridden by every child.

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.
- A syntactically ambiguous sentence has more than one parse. We'll have more to say about that in a few weeks, but one example is "I saw the man on the hill with the telescope."

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.
- A syntactically ambiguous sentence has more than one parse. We'll have more to say about that in a few weeks, but one example is "I saw the man on the hill with the telescope."
- A semantically ambiguous parse has more than one interpretation, even with the same grammatical parse. Consider this argument from LeBlanc and Wisdom:

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.
- A syntactically ambiguous sentence has more than one parse. We'll have more to say about that in a few weeks, but one example is "I saw the man on the hill with the telescope."
- A semantically ambiguous parse has more than one interpretation, even with the same grammatical parse. Consider this argument from LeBlanc and Wisdom:
    - All fathers are parents;

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.
- A syntactically ambiguous sentence has more than one parse. We'll have more to say about that in a few weeks, but one example is "I saw the man on the hill with the telescope."
- A semantically ambiguous parse has more than one interpretation, even with the same grammatical parse. Consider this argument from LeBlanc and Wisdom:
  - All fathers are parents;
  - All artists are dreamers;

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.
- A syntactically ambiguous sentence has more than one parse. We'll have more to say about that in a few weeks, but one example is "I saw the man on the hill with the telescope."
- A semantically ambiguous parse has more than one interpretation, even with the same grammatical parse. Consider this argument from LeBlanc and Wisdom:
  - All fathers are parents;
  - All artists are dreamers;
  - Therefore, all fathers of artists are parents of artists and fathers of dreamers.

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.
- A syntactically ambiguous sentence has more than one parse. We'll have more to say about that in a few weeks, but one example is "I saw the man on the hill with the telescope."
- A semantically ambiguous parse has more than one interpretation, even with the same grammatical parse. Consider this argument from LeBlanc and Wisdom:
  - All fathers are parents;
  - All artists are dreamers;
  - Therefore, all fathers of artists are parents of artists and fathers of dreamers.
- Define the unary predicates Ax and Dx as "x is an artist" and "x is a dreamer," respectively.

Translating to logical form

- Many English sentences admit more than one logical form. We say they are either syntactically or semantically *ambiguous*.
- A syntactically ambiguous sentence has more than one parse. We'll have more to say about that in a few weeks, but one example is "I saw the man on the hill with the telescope."
- A semantically ambiguous parse has more than one interpretation, even with the same grammatical parse. Consider this argument from LeBlanc and Wisdom:
  - All fathers are parents;
  - All artists are dreamers;
  - Therefore, all fathers of artists are parents of artists and fathers of dreamers.
- Define the unary predicates Ax and Dx as "x is an artist" and "x is a dreamer," respectively.
- Define the binary predicates Fxy and Pxy as "x is the father of y" and "x is the parent of y," respectively.

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$
- All artists are dreamers: $\forall x(Ax \rightarrow Dx)$

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$
- All artists are dreamers: $\forall x(Ax \rightarrow Dx)$
- Therefore, all fathers of artists are parents of artists and fathers of dreamers:
  $\forall wx((Fxw \wedge Aw) \rightarrow \exists yz((Pxy \wedge Ay) \wedge (Fxz \wedge Dz)))$

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$
- All artists are dreamers: $\forall x(Ax \rightarrow Dx)$
- Therefore, all fathers of artists are parents of artists and fathers of dreamers:
  $\forall wx((Fxw \wedge Aw) \rightarrow \exists yz((Pxy \wedge Ay) \wedge (Fxz \wedge Dz)))$

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$
- All artists are dreamers: $\forall x(Ax \rightarrow Dx)$
- Therefore, all fathers of artists are parents of artists and fathers of dreamers:
  $\forall wx((Fxw \wedge Aw) \rightarrow \exists yz((Pxy \wedge Ay) \wedge (Fxz \wedge Dz)))$

That translation is not only natural, it also gives us a valid argument. But on reflection, a strict reading of the first premise gives us this translation:

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$
- All artists are dreamers: $\forall x(Ax \rightarrow Dx)$
- Therefore, all fathers of artists are parents of artists and fathers of dreamers:
  $\forall wx((Fxw \wedge Aw) \rightarrow \exists yz((Pxy \wedge Ay) \wedge (Fxz \wedge Dz)))$

That translation is not only natural, it also gives us a valid argument. But on reflection, a strict reading of the first premise gives us this translation:

- All fathers are parents: $\forall xy(Fxy \rightarrow \exists zPxz)$

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$
- All artists are dreamers: $\forall x(Ax \rightarrow Dx)$
- Therefore, all fathers of artists are parents of artists and fathers of dreamers:
  $\forall wx((Fxw \wedge Aw) \rightarrow \exists yz((Pxy \wedge Ay) \wedge (Fxz \wedge Dz)))$

That translation is not only natural, it also gives us a valid argument. But on reflection, a strict reading of the first premise gives us this translation:

- All fathers are parents: $\forall xy(Fxy \rightarrow \exists zPxz)$

Translating to logical form

It seems natural to translate the argument this way:

- All fathers are parents: $\forall xy(Fxy \rightarrow Pxy)$
- All artists are dreamers: $\forall x(Ax \rightarrow Dx)$
- Therefore, all fathers of artists are parents of artists and fathers of dreamers:
  $\forall wx((Fxw \wedge Aw) \rightarrow \exists yz((Pxy \wedge Ay) \wedge (Fxz \wedge Dz)))$

That translation is not only natural, it also gives us a valid argument. But on reflection, a strict reading of the first premise gives us this translation:

- All fathers are parents: $\forall xy(Fxy \rightarrow \exists zPxz)$

In other words, "All fathers of anyone are parents of *someone*. And the resulting argument is *not* valid.

Inference and validity

- Just as with propositional logic, a conclusion follows validly from premises if it cannot be false when the premises are true.

Inference and validity

- Just as with propositional logic, a conclusion follows validly from premises if it cannot be false when the premises are true.
- As with translation, our common sense and domain knowledge can set us up for surprises. Consider (courtesy of Prof. Smullyan) :

Inference and validity

- Just as with propositional logic, a conclusion follows validly from premises if it cannot be false when the premises are true.
- As with translation, our common sense and domain knowledge can set us up for surprises. Consider (courtesy of Prof. Smullyan) :
  - Everyone loves a lover;

Inference and validity

- Just as with propositional logic, a conclusion follows validly from premises if it cannot be false when the premises are true.
- As with translation, our common sense and domain knowledge can set us up for surprises. Consider (courtesy of Prof. Smullyan) :
    - Everyone loves a lover;
    - Romeo loves Juliet;

Inference and validity

- Just as with propositional logic, a conclusion follows validly from premises if it cannot be false when the premises are true.
- As with translation, our common sense and domain knowledge can set us up for surprises. Consider (courtesy of Prof. Smullyan) :
    - Everyone loves a lover;
    - Romeo loves Juliet;
    - Therefore, Othello loves Iago.

Surprised?

- Everyone loves a lover

Surprised?

- Everyone loves a lover
    - $\forall x \forall y (Vx \rightarrow Lyx)$

Surprised?

- Everyone loves a lover
  - $\forall x \forall y (Vx \rightarrow Lyx)$
  - $\forall x (Vx \leftrightarrow \exists z Lxz)$

Surprised?

- Everyone loves a lover
    - $\forall x \forall y (Vx \rightarrow Lyx)$
    - $\forall x (Vx \leftrightarrow \exists z Lxz)$
- Romeo loves Juliet

Surprised?

- Everyone loves a lover
  - $\forall x \forall y (Vx \rightarrow Lyx)$
  - $\forall x (Vx \leftrightarrow \exists z Lxz)$
- Romeo loves Juliet
  - Lrj

Surprised?

- Everyone loves a lover
  - $\forall x \forall y (Vx \rightarrow Lyx)$
  - $\forall x (Vx \leftrightarrow \exists z Lxz)$
- Romeo loves Juliet
  - Lrj
- Therefore, Othello loves Iago.

Surprised?

- Everyone loves a lover
  - $\forall x \forall y (Vx \rightarrow Lyx)$
  - $\forall x (Vx \leftrightarrow \exists z Lxz)$
- Romeo loves Juliet
  - Lrj
- Therefore, Othello loves Iago.
  - Loi

Formal grammar for first order logic

From your reading for this week:

- $\mathbf{v} ::= x|y|z| \ldots$

Formal grammar for first order logic

From your reading for this week:

- $\mathbf{v} ::= x|y|z| \ldots$
- $\mathbf{c} ::= a|b|c| \ldots$

Formal grammar for first order logic

From your reading for this week:

- $\mathbf{v} ::= x|y|z|\ldots$
- $\mathbf{c} ::= a|b|c|\ldots$
- $\mathbf{t} ::= \mathbf{v}|\mathbf{c}$

Formal grammar for first order logic

From your reading for this week:

- $\mathbf{v} ::= x|y|z|\ldots$
- $\mathbf{c} ::= a|b|c|\ldots$
- $\mathbf{t} ::= \mathbf{v}|\mathbf{c}$
- $\mathbf{P} ::= P|Q|R|\ldots$

Formal grammar for first order logic

From your reading for this week:

- $\mathbf{v} ::= x|y|z|\ldots$
- $\mathbf{c} ::= a|b|c|\ldots$
- $\mathbf{t} ::= \mathbf{v}|\mathbf{c}$
- $\mathbf{P} ::= P|Q|R|\ldots$
- $\mathbf{Atom} ::= \mathbf{Pt}_1 \ldots \mathbf{t}_n$ where n is the arity of $\mathbf{P}$

From your reading for this week:

- $\mathbf{v} ::= x|y|z|\ldots$
- $\mathbf{c} ::= a|b|c|\ldots$
- $\mathbf{t} ::= \mathbf{v}|\mathbf{c}$
- $\mathbf{P} ::= P|Q|R|\ldots$
- $\mathbf{Atom} ::= \mathbf{Pt}_1 \ldots \mathbf{t}_n$ where n is the arity of $\mathbf{P}$
- $\varphi ::= \mathbf{Atom}|\neg\varphi|(\varphi \wedge \varphi)|(\varphi \vee \varphi)|(\varphi \rightarrow \varphi)|(\varphi \leftrightarrow \varphi)|\forall\mathbf{v}\varphi|\exists\mathbf{v}\varphi$