

Possible Worlds Semantics for Second-Order Modal Logic

MacKenzie Cumings

July 6, 2023

This is an attempt to define Second-Order Modal Logic in terms of possible worlds. The decision algorithm behind the web app <http://somerby.net/mack/logic> assumes that logical symbols have the meanings stated here.

There is not much in this document except definitions. I intend to write a complete paper on the subject someday. This paper would contain some evidence that these definitions are correct, a proof of decidability for a subset of symbolic that includes propositions, properties, and the necessity operator, and a decision algorithm for that subset. Also, since the meaning of the necessity operator given here is different from the meanings that it is normally given in modal predicate logic, the paper would also present definitions for necessity operators that capture these normal meanings that show the differences, show how they can be expressed in terms of the necessity operator used here, also show (what is already known) that predicate logics with these notions of necessity are not decidable. This would be a big project for me and I don't know when I will get around to doing it, so I've decided I should at least share the definitions.

1 Symbols

These are definitions for some symbols that appear in the definitions:

$$\mathcal{P}(S) := \text{the power set of } S \quad (1)$$

$$\bigcup S := \text{the union of all sets in } S \quad (2)$$

$$S^2 := S \times S := \{(x, y) : x \in S, y \in S\} \quad (3)$$

$$\lambda(\text{letters}).(\text{expression}) := \text{a function, where each letter is an argument to the function.} \quad (4)$$

$$\bigwedge_{m \in S} \phi := \phi \text{ is true for each } m \text{ in } S \quad (5)$$

$$\mathbb{P} := \text{the set of all possible worlds} \quad (6)$$

2 Meanings of Variables

ϕ is a syntactically valid statement of Second-Order Logic or a syntactically valid true/false expression within a statement of Second-Order Logic.

ψ is a syntactically valid statement of Second-Order Logic or a syntactically valid true/false expression within a statement of Second-Order Logic.

f is a function which is defined as a set of ordered pairs. The first items in the pairs are the domain of the function – the inputs or arguments that function is defined for – and the second items in the pairs are the range of the function – the outputs or results of the function.

w is a possible world.

o is an object in a possible world.

s is a symbol, usually a letter.

S is a set of symbols.

v is a function that, for a set of symbols (usually lowercase letters), maps each symbol to an object that exists in a possible world.

n is a function that, for a set of possible words, maps each possible world to a set of symbols (usually lowercase letters) that are supposed to represent propositions that are true in that possible world.

u is a function that, for a set of objects, maps each object to a set of symbols (usually uppercase letters) that are supposed to represent predicates that are true of that object.

b is a function that, for a set of pairs of objects, maps each pair to a set of symbols (usually uppercase letters) that are supposed to represent relations that exist between the pair.

3 Possible Worlds Semantics

A possible world is just a set of objects – the objects that exist in that world. Whether or not an object in a possible world is a subset of another object does not enter into these definitions.

Possible worlds are assumed not to be empty:

$$\neg \exists w, w \in \mathbb{P} \wedge w = \emptyset \quad (7)$$

Transworld identity is assumed to be impossible:

$$\forall o, \forall w_1, \forall w_2, (w_1 \in \mathbb{P} \wedge w_2 \in \mathbb{P} \wedge o \in w_1 \wedge w_1 \neq w_2) \rightarrow o \notin w_2 \quad (8)$$

A proposition symbol, or nullary predicate, is interpreted to be the set of possible world in which a proposition is true.

A property symbol, or unary predicate, is interpreted to be the set of objects which have a property.

A relation symbol, or binary predicate, is interpreted to be the set of pairs of objects which have a relation.

4 Translation

The definitions in Sections 5 through 8 are meant to be textual replacements that convert statements in conventional language of symbolic logic to statements about the set of possible worlds, \mathbb{P} , in the language of set theory. They are meant to be applied recursively to a statement, with parentheses added to the replacements to preserve order of operations. So, a definition of the form

$$\dots \phi \dots := \text{definition} \quad (9)$$

declares that, to translate “ $\dots \phi \dots$ ”, ϕ should first be translated, then every instance of ϕ in *definition* should be replaced with the result of that translation, and then the resulting expression should be surrounded by parentheses.

5 Binding Symbols

A statement may have free symbols in it – symbols that are not bound by any quantifier. In both First-Order and Second-Order Logics, free symbols represent constants. In First-Order Logic, all propositions, predicates and relations are free. To decide the validity of a statement, it must be decided whether it is true for all possible meanings of the free symbols. The functions defined in this section are used to bind symbols in such a way that the statement is considered for all possible meanings of the symbol.

$$\text{free-v}(\phi) := \text{an ordered list of free variables in } \phi \quad (10)$$

$$\text{free-n}(\phi) := \text{an ordered list of free nullary predicates in } \phi \quad (11)$$

$$\text{free-u}(\phi) := \text{an ordered list of free unary predicates in } \phi \quad (12)$$

$$\text{free-b}(\phi) := \text{an ordered list of free binary predicates in } \phi \quad (13)$$

$$\text{bind-v}(\phi) := \begin{cases} \phi & \text{if free-v}(\phi) = \emptyset \\ \text{bind-v}(\forall(\text{head}(\text{free-v}(\phi))), \phi) & \text{if free-v}(\phi) \neq \emptyset \end{cases} \quad (14)$$

$$\text{bind-n}(\phi) := \begin{cases} \phi & \text{if free-n}(\phi) = \emptyset \\ \text{bind-n}(\forall(\text{head}(\text{free-n}(\phi))), \phi) & \text{if free-n}(\phi) \neq \emptyset \end{cases} \quad (15)$$

$$\text{bind-u}(\phi) := \begin{cases} \phi & \text{if free-u}(\phi) = \emptyset \\ \text{bind-u}(\forall(\text{head}(\text{free-u}(\phi))), \phi) & \text{if free-u}(\phi) \neq \emptyset \end{cases} \quad (16)$$

$$\text{bind-b}(\phi) := \begin{cases} \phi & \text{if free-b}(\phi) = \emptyset \\ \text{bind-b}(\forall(\text{head}(\text{free-b}(\phi))), \phi) & \text{if free-b}(\phi) \neq \emptyset \end{cases} \quad (17)$$

6 Boolean Definitions

These definitions translate statements of symbolic logic into Boolean (true/false) statements:

6.1 Necessity

A statement is necessarily true if and only if it is true in all possible worlds.

$$\Box\phi := \bigwedge_{w \in \mathbb{P}} \phi \quad (18)$$

6.2 Universal Generalization

A universal generalization is true if and only if its predicate is true for all objects in the possible world that is assumed to be the current context.

$$\forall x, \phi := \bigwedge_{x \in w} \phi \quad (19)$$

6.3 Validity

A statement is valid if and only if it is true in all possible worlds for all possible assignments of objects in each possible world to constants, under all possible meanings for the free propositions, predicates and relations in the statement.

$$\text{valid}(\phi) := \text{bind-b}(\text{bind-n}(\text{bind-u}(\Box(\text{bind-v}(\phi)))))) \quad (20)$$

6.4 Propositions

A proposition is conflated with the set of possible worlds that it is true in. A proposition is true if and only if it includes the possible world that is assumed to be the current context.

$$p := w \in p \quad (21)$$

$$\forall p, \phi := \bigwedge_{p \in \mathcal{P}(\mathbb{P})} \phi \quad (22)$$

6.5 Properties

A property is conflated with the set of objects that have it. An object has a property if and only if it is a member the property.

$$Px := x \in P \quad (23)$$

$$\forall P, \phi := \bigwedge_{P \in \mathcal{P}(\cup \mathbb{P})} \phi \quad (24)$$

6.6 Relations

A relation is conflated with the set of ordered pairs of objects that have it. An ordered pair of objects has a relation if it is a member of the relation.

$$Rxy := (x, y) \in R \quad (25)$$

$$\forall R, \phi := \bigwedge_{R \in \mathcal{P}((\cup \mathbb{P})^2)} \phi \quad (26)$$

6.7 Other Operators

No special definition is needed for negation or conjunction. Because of (8), identity also does not need a special definition.

$$x=y := x = y \quad (27)$$

$$\neg\phi := \neg\phi \quad (28)$$

$$\phi \wedge \psi := \phi \wedge \psi \quad (29)$$

Operators that can be defined in terms of the operators above, such as ‘ \exists ’, ‘ \vee ’ and ‘ \diamond ’, are defined in the usual way.

7 Set Definitions

These definitions are based on the idea that the meaning of a statement is the set of possible worlds that it is true in. These definitions translate a statement of symbolic logic to an expression that identifies a set of possible worlds.

7.1 Truth

ϕ is true when and where ϕ includes the possible world that is assumed in the current context.

$$\text{true}(w, \phi) := w \in \phi \quad (30)$$

7.2 Validity

ϕ is valid if the result of binding all of its free variables and symbols is an expression that identifies a set that includes all possible worlds.

$$\text{valid}(\phi) := \text{bind-b}(\text{bind-n}(\text{bind-u}(\Box(\text{bind-v}(\phi)))) = \mathbb{P} \quad (31)$$

7.3 Necessity

$\Box\phi$ is the set of all possible worlds if ϕ is the set of all possible worlds. Otherwise, it is the empty set.

$$\Box\phi := \begin{cases} \mathbb{P} & \text{if } \phi = \mathbb{P} \\ \emptyset & \text{if } \phi \neq \mathbb{P} \end{cases} \quad (32)$$

alternatively

$$\Box\phi := \bigcap_{w \in \mathbb{P}} \begin{cases} \mathbb{P} & \text{if } w \in \phi \\ \emptyset & \text{if } w \notin \phi \end{cases} \quad (33)$$

7.4 Universal Generalization

$\forall x, \phi$ is the set of possible worlds which are included in ϕ no matter what object in the current world is assigned to the symbol 'x'.

$$\forall x, \phi := \bigcap_{x \in w} \phi \quad (34)$$

7.5 Propositions

Since a proposition is a set of possible worlds, it needs no special definition.

A generalization of 'p' over ϕ is the set of possible worlds which are included in ϕ no matter what set of possible worlds is assumed for 'p'.

$$\forall p, \phi := \bigcap_{p \in \mathcal{P}(\mathbb{P})} \phi \quad (35)$$

7.6 Properties

If an object x has a property P, Px is defined as the set of all possible worlds. If it does not have property P, Px is defined as the empty set. This is because, in context, its truth or falsehood only depends on the object assigned to x and the property assigned to P and not the possible world assumed in the context.

$$Px := \begin{cases} \mathbb{P} & \text{if } x \in P \\ \emptyset & \text{if } x \notin P \end{cases} \quad (36)$$

$$\forall P, \phi := \bigcap_{P \in \mathcal{P}(\cup \mathbb{P})} \phi \quad (37)$$

7.7 Relations

If a pair (x,y) in a relation R , R_{xy} is defined as the set of all possible worlds. If it is not in a relation R , R_{xy} is defined as the empty set. This is because, in context, its truth or falsehood only depends on the objects assigned to x and y and the relation assigned to R and not the possible world assumed in the context.

$$R_{xy} := \begin{cases} \mathbb{P} & \text{if } (x,y) \in R \\ \emptyset & \text{if } (x,y) \notin R \end{cases} \quad (38)$$

$$\forall R, \phi := \bigcap_{R \in \mathcal{P}((\cup \mathbb{P})^2)} \phi \quad (39)$$

7.8 Conjunction

The conjunction of ϕ and ψ is their set intersection – the set of possible worlds which are contained in both ϕ and ψ .

$$\phi \wedge \psi := \phi \cap \psi \quad (40)$$

7.9 Negation

The negation of ϕ is its set complement – the set of possible worlds that are not in ϕ .

$$\neg \phi := \overline{\phi} \quad (41)$$

7.10 Other Operators

Because of (8), identity does not need a special definition.

$$x=y := x = y \quad (42)$$

Operators that can be defined in terms of the operators above, such as ‘ \exists ’, ‘ \forall ’ and ‘ \diamond ’, are defined in the usual way.

8 Definitions for Decidability

My plan for a proof of decidability involves proving that certain generalizations over infinite sets of objects or infinite sets of sets of objects are logically equivalent to generalizations over finite sets of strings of symbols. To make this proof easier, I defined the logical operators as functions of the variables w , v , n , u and b , which are described in section (2). In these definitions, mappings of things to strings of symbols are built up by “assigning” symbols to things, and “instantiating” variables with objects is achieved by building a mapping of variable symbols to objects.

8.1 Instantiation and Assignment

This function adds a symbol/object pair to a mapping of symbols to objects, associating the symbol with the object:

$$\text{instantiate}(v, s, o) := v \cup \{(s, o)\} \quad (43)$$

This function adds a symbol to the set of symbols associated with a thing t in a mapping of things to symbols if t belongs to a set of things T . If t does not belong to T , the set of symbols associated with it is not changed:

$$\text{assign}(f, s, T) := \{(t, S \cup \{s\}) : (t, S) \in f \wedge t \in T\} \cup \{(t, S) : (t, S) \in f \wedge t \notin T\} \quad (44)$$

Alternatively,

$$\text{assign}(f, s, T) := \{p : \exists t, \exists S, (t, S) \in f \wedge (t \in T \rightarrow p = (t, S \cup \{s\})) \wedge (t \notin T \rightarrow p = (t, S))\} \quad (45)$$

Each kind of expression in Second-Order Logic is defined as a function of parameters w , v , n , u , b , which have the meanings described in Section 2 above. These parameters are the context of the expression; they provide the meaning of all of the predicates and variables in the expression.

8.2 Propositions

A proposition symbol “p” is true in context if it is in the set of symbols that are associated with the possible world w .

$$“p” := \lambda w v n u b. 'p' \in n(w) \quad (46)$$

8.3 Properties

A predication of a property P on an object x is true in context if it is in the set of symbols that are associated with the object that is associated with the symbol ‘x’ in context.

8.4 Relations

$$“P_x” := \lambda w v n u b. 'P' \in u(v('x')) \quad (47)$$

A relation R between objects x and y is true in context if it is in the set of symbols that are associated with the pair of objects that are associated with symbols ‘x’ and ‘y’ in context.

$$“R_{xy}” := \lambda w v n u b. 'R' \in b(v('x'), v('y')) \quad (48)$$

8.5 Identity

x and y are identical if the symbols ‘x’ and ‘y’ are associated with the same object in context.

$$“x=y” := \lambda w v n u b. v('x') = v('y') \quad (49)$$

8.6 Negation

$\neg\phi$ is true if ϕ , in the same context, is false. ϕ gets the same context as $\neg\phi$ because $\neg\phi$ passes its arguments to ϕ unchanged.

$$“\neg\phi” := \lambda w v n u b. \neg\phi(w, v, n, u, b) \quad (50)$$

8.7 Conjunction

$\phi \wedge \psi$ is true if both ϕ and ψ are true in the same context. ϕ and ψ get the same context as $\phi \wedge \psi$ because $\phi \wedge \psi$ passes its arguments to ϕ and ψ unchanged.

$$“\phi \wedge \psi” := \lambda w v n u b. \phi(w, v, n, u, b) \wedge \psi(w, v, n, u, b) \quad (51)$$

8.8 Necessity

ϕ is necessarily true if it is true in every possible world. ϕ does not always get the same context as $\Box\phi$; it is considered in every possible world instead of just the current possible world. The assignment of objects to free variables in ϕ does not change, nor do the objects change.

$$“\Box\phi” := \lambda w_1 v n u b. \bigwedge_{w_2 \in \mathbb{P}} \phi(w_2, v, n, u, b) \quad (52)$$

8.9 Universal Generalization

A universal generalization is true if and only if, for every object that is in the current possible world, ϕ is true when instantiated with the object.

$$“\forall x, \phi” := \lambda w v n u b. \bigwedge_{o \in w} \phi(w, \text{instantiate}(v, 'x', o), n, u, b) \quad (53)$$

8.10 Generalization over Propositions

A proposition is conflated with the set of possible worlds in which it is true. Consequently, a generalization over propositions is true if and only if, for every combination of possible worlds, ϕ is true when, for each possible world in the combination, ‘p’ is added to the list of propositions which are true in it.

$$\text{“}\forall p, \phi\text{”} := \lambda wvnub. \bigwedge_{P \in \mathcal{P}(\mathbb{P})} \phi(w, v, \text{assign}(n, \text{‘p’}, P), u, b) \quad (54)$$

8.11 Generalization over Properties

A property is conflated with the set of objects that have it. Consequently, a generalization over properties is true if and only if, for every combination of objects, ϕ is true when, for each object in the combination, ‘P’ is added to the list of properties that the object has.

$$\text{“}\forall P, \phi\text{”} := \lambda wvnub. \bigwedge_{p \in \mathcal{P}(\cup \mathbb{P})} \phi(w, v, n, \text{assign}(u, \text{‘P’}, p), b) \quad (55)$$

8.12 Generalization over Relations

A relation is conflated with the set of pairs of objects that have it. Consequently, a generalization over relations is true if and only if, for every combination of pairs of objects, ϕ is true when, for each pair in the combination, ‘R’ is added to the list of relations that the pair has.

$$\text{“}\forall R, \phi\text{”} := \lambda wvnub. \bigwedge_{r \in \mathcal{P}((\cup \mathbb{P})^2)} \phi(w, v, n, u, \text{assign}(b, \text{‘R’}, r)) \quad (56)$$

8.13 Other Operators

Operators that can be defined in terms of the operators above, such as ‘ \exists ’, ‘ \vee ’ and ‘ \diamond ’, are defined in the usual way.

8.14 Validity

Validity is defined similarly to how it is defined in Section 6.3, but since the other definitions are functions, $\text{valid}(\phi)$ is a function application (a function call) that passes initial mappings and an empty world to the function that is the result of binding all of the free symbols in ϕ .

$$\begin{aligned} \text{valid}(\phi) &:= \text{bind-b}(\text{bind-n}(\text{bind-u}(\square(\text{bind-v}(\phi)))) \\ &(\emptyset, \emptyset, \{(w, \emptyset) : w \in \mathbb{P}\}, \{(o, \emptyset) : o \in \cup \mathbb{P}\}, \{((a, b), \emptyset) : a \in \cup \mathbb{P} \wedge b \in \cup \mathbb{P}\}) \end{aligned} \quad (57)$$