Modal Logic

kankanray

Contents

1 Basic Concepts

1.1 Relational Structures

Definition 1.1. A relational structure $\mathfrak{F}$ is a tuple, where the first component is a non-empty set W called *universe*, and remaining components are relations on W. We assume there is at least one relation on W.

In the following 2 definitions, we assume W be a non-empty set and R a binary relation on W.

Definition 1.2. $R^+ := \bigcap\{R'|R'$ is a transitive binary relation on W and $R \subseteq R'\}$, is called the *transitive closure* of R.

Definition 1.3. $R^* := \bigcap\{R'|R'$ is a reflexive transitive binary relation on W and $R \subseteq R'\}$, is called the *reflexive transitive closure* of R.

Note that transitive closure of a binary relation has nice *finite steps* property, see Exercise 1.1.3.

Definition 1.4. A tree $\mathfrak{T}$ is a structural structure (T, S) where:
(i)T, the set of nodes, contains a unique $r \in T$ (root), such that $\forall t \in TS^*rt$.
(ii)For every $t \neq r$, there is a unique $t' \in T$, such that $St't$
(iii)$\forall t \neg S^+tt$, so S is *acyclic*.

Question 1.1. Why we define tree like that?

1.2 Modal Languages

Definition 1.5. Basic modal language:

-A set of proposition letters(or proposition symbols or propositional variables) Φ, whose elements are usually denoted p, q, r, and so on.

-A unary modal operator $\Diamond$.

Then the well-formed *formulas* ϕ of the basic modal language are given by the rule:

$\phi ::= p|\bot|\neg\phi|\psi \vee \phi|\Diamond\phi$,

where p ranges over elements pf Φ.

There is also a dual operator $\Box$ which is defined by $\Box\phi := \neg\Diamond\neg\phi$.

Moreover, we can define conjunction, implication, bi-implication, and the constant true as usual:

$\phi \wedge \psi := \neg(\neg\phi \vee \neg\psi)$,

$\phi \rightarrow \psi := \neg\phi \vee \psi$,

$\phi \leftrightarrow \psi := (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, and

$\top := \neg\bot$.

The following 3 definitions generalize the concept of basic modal language.

Definition 1.6. A *modal similarity type* is a pair $\tau = (O, \rho)$ where O is a non-empty set, and ρ is a function $O \rightarrow \mathbb{N}$. The elements of O are called *modal operators*; we use $\triangle$, $\triangle_0$, $\triangle_1$,..., to denote elements of O. The function ρ assigns to each operator $\triangle \in O$ a finite *arity* indicating the number of arguments $\triangle$ can be applied to.

So we often refer to *unary* triangles as *diamonds*, and denote them by $\Diamond_a$ or $\langle a \rangle$, where a is taken from some index set.

Definition 1.7. A *modal language* $ML(\tau, \Phi)$, with modal similarity type $\tau = (O, \rho)$ and a set of proposition letters Φ. The well-formed formulas are given by the rule:

$\phi := p|\bot|\neg\phi|\phi_1 \vee \phi_2|\triangle(\phi_1, ..., \phi_{\rho(\triangle)})$,

where p ranges over elements of Φ.

Definition 1.8. Dual operators for non-nullary triangles. For each $\triangle \in O$ the dual $\triangledown$ is defined as $\triangledown(\phi_0, ..., \phi_n) = \neg\triangle(\neg\phi_0, ..., \neg\phi_n)$. The dual of a triangle of arity at least 2 is called a *nabla*. A *box*(unary triangle-down) is written $\Box_a$ or $[a]$.

Definition 1.9. A *substitution* is a map $\sigma : \Phi \rightarrow Form(\tau.\Phi)$(formulas).

Then a substitution σ induces a map $(\cdot)^\sigma : Form(\tau, \Phi) \rightarrow Form(\tau, \Phi)$, which can be recursively defined as follows:

$\bot^\sigma = \bot$,

$p^\sigma = \sigma(p)$,

$(\neg\psi)^\sigma = \neg\psi^\sigma$,

$(\psi \vee \theta)^\sigma = \psi^\sigma \vee \theta^\sigma$,

$(\triangle(\psi_1, ..., \psi_n))^\sigma = \triangle(\psi_1^\sigma, ..., \psi_n^\sigma)$

1.3 Models and Satisfaction

Definition 1.10. A *frame* for basic modal language is a pair $\mathfrak{F} = (W, R)$ such that

(i) W is a non-empty set.
(ii) R is a binary relation on W.

Definition 1.11. A *model* for the basic modal language is a pair $\mathfrak{M} = (\mathfrak{F}, V)$, where $\mathfrak{F}$ is a frame for the basic modal language, and V is a function assigning to each proposition letter p in Φ a subset $V(p)$ of W

Given a model $\mathfrak{M} = (\mathfrak{F}, V)$, we say that $\mathfrak{M}$ is *based on* the frame $\mathfrak{F}$, or that $\mathfrak{F}$ is the frame *underlying* $\mathfrak{M}$.

Now we discuss the semantics in basic modal language.

Definition 1.12. Suppose w is a state in a model $\mathfrak{M} = (W, R, V)$(i.e. w is an element of W). Then we inductively define the notion of a formula ϕ being satisfied(or true) in $\mathfrak{M}$ at state w as follows:

$\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$, where $p \in \Phi$,
$\mathfrak{M}, w \Vdash \perp$ never,
$\mathfrak{M}, w \Vdash \neg\phi$ iff not $\mathfrak{M}, w \Vdash \phi$,
$\mathfrak{M}, w \Vdash \phi \wedge \psi$ iff $\mathfrak{M}, w \Vdash \phi$ or $\mathfrak{M}, w \Vdash \psi$,
$\mathfrak{M}, w \Vdash \Diamond\phi$ iff for some $v \in W$ with Rwv we have $\mathfrak{M}, v \Vdash \phi$.

Then follows the definition, $\mathfrak{M}, w \Vdash \Box\phi$ iff for every $v \in W$ with Rwv we have $\mathfrak{M}, v \Vdash \phi$.

Finally we say a set of formulas is true at state w iff every formula in it is true at such state.

When $\mathfrak{M}$ is clear from the context, we write $w \Vdash \phi$ instead of $\mathfrak{M}, w \Vdash \phi$.

If it's not $\mathfrak{M}, w \Vdash \phi$, we may write $\mathfrak{M}, w \nVdash \phi$ or simply $w \nVdash \phi$.

Also, we may extend the valuation V from proposition letters to arbitrary formula ϕ:

$V(\phi) = \{w | \mathfrak{M}, w \Vdash \phi\}$.

There are some special concepts:

Definition 1.13. A formula ϕ is said to be *globally* or *universally true* in a model $\mathfrak{M}$ iff it is satisfied at all points in $\mathfrak{M}$.

A formula ϕ is *satisfiable* in a model $\mathfrak{M}$ iff there is *some* state in $\mathfrak{M}$ at which ϕ is true.

A formula ϕ is *falsifiable* or *refutable* in a model iff its negation is satisfiable.

A set Σ is globally true(satisfiable, respectively) in a model $\mathfrak{M}$ iff $\mathfrak{M}, w \Vdash \Sigma$ for all states w in $\mathfrak{M}$(some state w in $\mathfrak{M}$,respectively).

Now we discuss frames, models and satisfaction for modal languages of arbitrary similarity type.

Definition 1.14. Let τ be a modal similarity type. A *τ-frame* is a tuple $\mathfrak{F}$ consisting of the following ingredients:

(i) a non-empty set W,

(ii) for each $n \geq 0$ and each n-ary modal operator $\triangle$ in the similarity type τ and (n+1)-ary relation $R_{\triangle}$.

Definition 1.15. A *τ-model* is a pair $\mathfrak{M} = (\mathfrak{F}, V)$ where $\mathfrak{F}$ is a τ-frame and V is a valuation, with domain Φ and range $\mathcal{P}(W)$. We call W the universe of $\mathfrak{F}$.

What about the satisfaction? Similar to the case for basic modal language, we define the following:

Definition 1.16. When ϕ is satisfied in a state w in the model $\mathfrak{M}$, we write $\mathfrak{M}, w \Vdash \phi$. We define such relations inductively:

The clauses for atomic and boolean cases are the same for the basic modal language.

As for the modal case, when $n = \rho(\triangle) > 0$ we define

$\mathfrak{M}, w \Vdash \triangle(\phi_1, ..., \phi_n)$ iff for some $v_1, ..., v_n \in W$ with $R_{\triangle} w v_1 v_2 ... v_n$ we have, for each i, $\mathfrak{M}, w \Vdash \phi_i$.

With this definition, we know when $\triangledown(\phi_1, ..., \phi_n)$ is satisfied:

$\mathfrak{M}, w \Vdash \triangledown(\phi_1, ..., \phi_n)$ iff for every $v_1, ..., v_n \in W$ with $R_{\triangle} w v_1 ..., v_n$, there exists an $i \in \{1, 2, ..., n\}$ such that $\mathfrak{M}, v_i \Vdash \phi_i$.

Finally, if $\rho(\triangle) = 0$, then $R_{\triangle}$ is an unary relation, so we define: $\mathfrak{M}, w \Vdash \triangle$ iff $w \in R_{\triangle}$.

The concepts of globally true and so on are defined similarly.

Then it follows several examples, and I really hate them.

Now we discuss the concept of validity, which describe the situations when a formula ϕ is true in any valuation.

Definition 1.17. A formula ϕ is *valid in a state w in a frame* $\mathfrak{F}$(notation: $\mathfrak{F}, w \Vdash \phi$) iff ϕ is true in every model $(\mathfrak{F}, V)$ based on $\mathfrak{F}$.

ϕ is *valid in a frame* $\mathfrak{F}$(notation: $\mathfrak{F} \Vdash \phi$) iff ϕ is valid at every state in W.

ϕ is *valid on a class of frames* F(notation: $\mathsf{F} \Vdash \phi$) iff it is valid on every frame $\mathfrak{F}$ in F.

Finally, we say ϕ is *valid* iff it is valid on the class of all frames.

By the way, the set of all formulas that are valid in a class of frames F is called the *logic* of F(notation:Λ_{F}).

1.4 General Frames

Definition 1.18. Given an (n+1)-ary relation R on a set W, we define the following n-ary operation m_R on the power set $\mathcal{P}(W)$ of W:

$m_R(X_1, ..., X_n) = \{w \in W | R w w_1 ... w_n$ for some $w_1 \in X_1, ..., w_n \in X_n\}$.

Definition 1.19. General Frame

Let τ be a modal similarity type. A*general τ-frame* is pair $(\mathfrak{F}, A)$, where $\mathfrak{F} = (W, R_\triangle)_{\triangle \in \tau}$ is a τ-frame, and A is a non-empty collection of admissible subsets of W closed under the following operations:

(i) union: if $X, Y \in A$, then $X \cup Y \in A$.

(ii) relative complement: if $X \in A$, then W
$X \in A$.

(iii) modal operations: if $X_1, ..., X_n \in A$, then $m_{R_\triangle}(X_1, ..., X_n) \in A$ for all $\triangle \in \tau$.

A *modal based on a general frame* is a triple $(\mathfrak{F}, A, V)$ where $(\mathfrak{F}, A)$ is a general frame and V is a valuation satisfying the constraint that for each proposition letter p, $V(p)$ is an element of A. Valuations satisfying this constraint are called *admissible* for $(\mathfrak{F}, A)$.

Definition 1.20. Validity

A formula ϕ is *valid at a state w in a general frame* $(\mathfrak{F}, A)$(notation: $(\mathfrak{F}, A), w \Vdash \phi$) iff ϕ is true at w in every admissible model.

ϕ is *valid*(notation: $(\mathfrak{F}, A) \Vdash \phi$) iff ϕ is valid in every state.

ϕ is valid in a class of general frames G(notation: $\mathsf{G} \Vdash \phi$) iff it's valid in every general frame in G.

Finally, we say ϕ is *g-valid* iff it's valid in all general frames. We will see that ϕ is valid iff it's g-valid.