

IAoo8: Computational Logic

7. Modal Logic

Achim Blumensath

blumens@fi.muni.cz

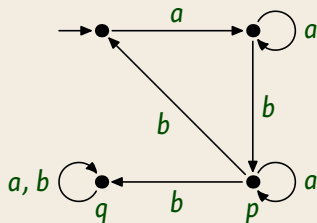
Faculty of Informatics, Masaryk University, Brno

Basic Concepts

Transition Systems

directed graph $\mathfrak{G} = \langle S, (E_a)_{a \in A}, (P_i)_{i \in I}, s_0 \rangle$ with

- ▶ **states** S
- ▶ **initial state** $s_0 \in S$
- ▶ **edge relations** E_a with **edge colours** $a \in A$ ('actions')
- ▶ **unary predicates** P_i with **vertex colours** $i \in I$ ('properties')



Modal logic

Propositional logic with **modal operators**

- ▶ $\langle a \rangle \varphi$ 'there exists an a -successor where φ holds'
- ▶ $[a] \varphi$ ' φ holds in every a -successor'

Notation: $\diamond \varphi, \square \varphi$ if there are no edge labels

Formal semantics

$\mathfrak{G}, s \models P$: iff $s \in P$

$\mathfrak{G}, s \models \varphi \wedge \psi$: iff $\mathfrak{G}, s \models \varphi$ and $\mathfrak{G}, s \models \psi$

$\mathfrak{G}, s \models \varphi \vee \psi$: iff $\mathfrak{G}, s \models \varphi$ or $\mathfrak{G}, s \models \psi$

$\mathfrak{G}, s \models \neg \varphi$: iff $\mathfrak{G}, s \not\models \varphi$

$\mathfrak{G}, s \models \langle a \rangle \varphi$: iff there is $s \rightarrow^a t$ such that $\mathfrak{G}, t \models \varphi$

$\mathfrak{G}, s \models [a] \varphi$: iff for all $s \rightarrow^a t$, we have $\mathfrak{G}, t \models \varphi$

Examples

$P \wedge \diamond Q$ 'The state is in P and there exists a transition to Q .'

$[a]\perp$ 'The state has no outgoing a -transition.'

Interpretations

- ▶ **Temporal Logic** talks about time:
 - ▶ states: points in time (discrete/continuous)
 - ▶ $\diamond\varphi$ 'sometime in the future φ holds'
 - ▶ $\square\varphi$ 'always in the future φ holds'
- ▶ **Epistemic Logic** talks about knowledge:
 - ▶ states: possible worlds
 - ▶ $\diamond\varphi$ ' φ might be true'
 - ▶ $\square\varphi$ ' φ is certainly true'

Examples: Temporal Logic

system $\mathcal{G} = \langle S, <, \bar{P} \rangle$

- ▶ “ P never holds.”

Examples: Temporal Logic

system $\mathcal{G} = \langle S, <, \bar{P} \rangle$

- ▶ “ P never holds.”

$$\neg P \wedge \neg \diamond P$$

- ▶ “After every P there is some Q .”

Examples: Temporal Logic

system $\mathfrak{G} = \langle S, <, \bar{P} \rangle$

- ▶ “ P never holds.”

$$\neg P \wedge \neg \diamond P$$

- ▶ “After every P there is some Q .”

$$(P \rightarrow \diamond Q) \wedge \square(P \rightarrow \diamond Q)$$

- ▶ “Once P holds, it holds forever.”

Examples: Temporal Logic

system $\mathfrak{G} = \langle S, <, \bar{P} \rangle$

- ▶ “ P never holds.”

$$\neg P \wedge \neg \diamond P$$

- ▶ “After every P there is some Q .”

$$(P \rightarrow \diamond Q) \wedge \square(P \rightarrow \diamond Q)$$

- ▶ “Once P holds, it holds forever.”

$$(P \rightarrow \square P) \wedge \square(P \rightarrow \square P)$$

- ▶ “There are infinitely many P .” (for the order $\langle \mathbb{N}, < \rangle$)

Examples: Temporal Logic

system $\mathfrak{G} = \langle S, <, \bar{P} \rangle$

- ▶ “ P never holds.”

$$\neg P \wedge \neg \diamond P$$

- ▶ “After every P there is some Q .”

$$(P \rightarrow \diamond Q) \wedge \square(P \rightarrow \diamond Q)$$

- ▶ “Once P holds, it holds forever.”

$$(P \rightarrow \square P) \wedge \square(P \rightarrow \square P)$$

- ▶ “There are infinitely many P .” (for the order $\langle \mathbb{N}, < \rangle$)

$$\square \diamond P$$

Translation to first-order logic

Proposition

For every formula φ of propositional modal logic, there exists a formula $\varphi^*(x)$ of first-order logic such that

$$\mathfrak{G}, s \models \varphi \quad \text{iff} \quad \mathfrak{G} \models \varphi^*(s).$$

Proof

Translation to first-order logic

Proposition

For every formula φ of propositional modal logic, there exists a formula $\varphi^*(x)$ of first-order logic such that

$$\mathfrak{G}, s \models \varphi \quad \text{iff} \quad \mathfrak{G} \models \varphi^*(s).$$

Proof

$$P^* := P(x)$$

$$(\varphi \wedge \psi)^* := \varphi^*(x) \wedge \psi^*(x)$$

$$(\varphi \vee \psi)^* := \varphi^*(x) \vee \psi^*(x)$$

$$(\neg\varphi)^* := \neg\varphi^*(x)$$

$$(\langle a \rangle \varphi)^* := \exists y [E_a(x, y) \wedge \varphi^*(y)]$$

$$([a] \varphi)^* := \forall y [E_a(x, y) \rightarrow \varphi^*(y)]$$

Bisimulation

\mathfrak{S} and \mathfrak{T} transition systems

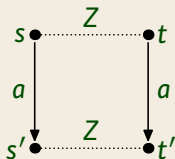
$Z \subseteq S \times T$ is a **bisimulation** if, for all $\langle s, t \rangle \in Z$,

(local) $s \in P \Leftrightarrow t \in P$

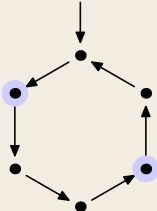
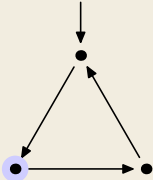
(forth) for every $s \rightarrow^a s'$, exists $t \rightarrow^a t'$ with $\langle s', t' \rangle \in Z$,

(back) for every $t \rightarrow^a t'$, exists $s \rightarrow^a s'$ with $\langle s', t' \rangle \in Z$.

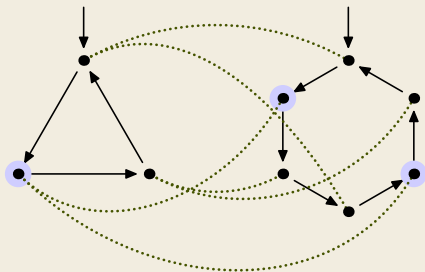
\mathfrak{S}, s and \mathfrak{T}, t are **bisimilar** if there is a bisimulation Z with $\langle s, t \rangle \in Z$.



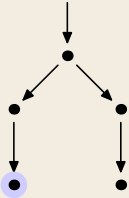
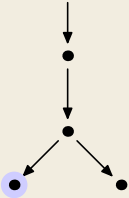
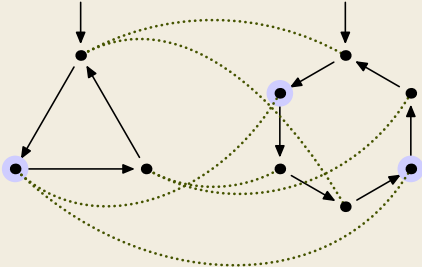
Examples



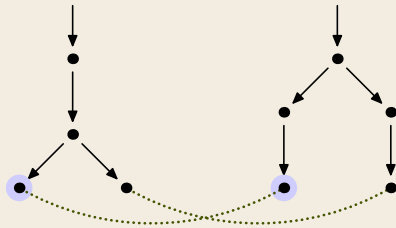
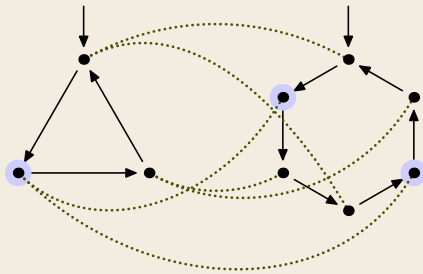
Examples



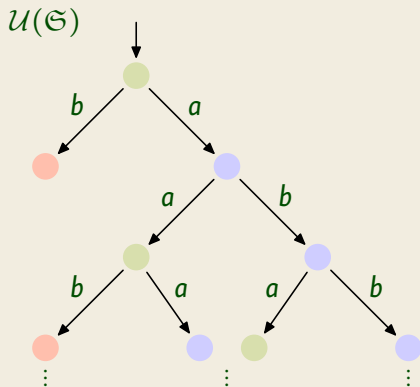
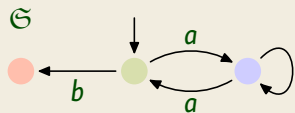
Examples



Examples



Unravelling



Lemma

\mathcal{G} and $\mathcal{U}(\mathcal{G})$ are bisimilar.

Bisimulation invariance

Theorem

Two **finite** transition systems \mathcal{G}, s and \mathcal{T}, t are **bisimilar** if, and only if,

$$\mathcal{G}, s \models \varphi \iff \mathcal{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

Bisimulation invariance

Theorem

Two **finite** transition systems \mathcal{G}, s and \mathcal{T}, t are **bisimilar** if, and only if,

$$\mathcal{G}, s \models \varphi \iff \mathcal{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

$$(\varphi = P) \quad s \in P \iff t \in P$$

Bisimulation invariance

Theorem

Two **finite** transition systems \mathcal{G}, s and \mathcal{T}, t are **bisimilar** if, and only if,

$$\mathcal{G}, s \models \varphi \iff \mathcal{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

$$(\varphi = P) \quad s \in P \iff t \in P$$

(boolean combinations) by inductive hypothesis

Bisimulation invariance

Theorem

Two **finite** transition systems \mathcal{G}, s and \mathcal{T}, t are **bisimilar** if, and only if,

$$\mathcal{G}, s \models \varphi \iff \mathcal{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

$$(\varphi = P) \quad s \in P \iff t \in P$$

(boolean combinations) by inductive hypothesis

$$(\varphi = \langle a \rangle \psi) \quad \mathcal{G}, s \models \langle a \rangle \psi$$

Bisimulation invariance

Theorem

Two **finite** transition systems \mathfrak{S}, s and \mathfrak{T}, t are **bisimilar** if, and only if,

$$\mathfrak{S}, s \models \varphi \iff \mathfrak{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

$$(\varphi = P) \quad s \in P \iff t \in P$$

(boolean combinations) by inductive hypothesis

$$(\varphi = \langle a \rangle \psi) \quad \mathfrak{S}, s \models \langle a \rangle \psi$$

$$\Rightarrow \text{ex. } s \xrightarrow{a} s' \text{ with } \mathfrak{S}, s' \models \psi$$

Bisimulation invariance

Theorem

Two **finite** transition systems \mathfrak{G}, s and \mathfrak{T}, t are **bisimilar** if, and only if,

$$\mathfrak{G}, s \models \varphi \iff \mathfrak{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

$$(\varphi = P) \quad s \in P \iff t \in P$$

(boolean combinations) by inductive hypothesis

$$(\varphi = \langle a \rangle \psi) \quad \mathfrak{G}, s \models \langle a \rangle \psi$$

$$\Rightarrow \text{ex. } s \xrightarrow{a} s' \text{ with } \mathfrak{G}, s' \models \psi$$

$$\mathfrak{G}, s \sim \mathfrak{T}, t \Rightarrow \text{ex. } t \xrightarrow{a} t' \text{ with } \mathfrak{G}, s' \sim \mathfrak{T}, t'$$

Bisimulation invariance

Theorem

Two **finite** transition systems \mathcal{G}, s and \mathcal{T}, t are **bisimilar** if, and only if,

$$\mathcal{G}, s \models \varphi \iff \mathcal{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

$$(\varphi = P) \quad s \in P \iff t \in P$$

(boolean combinations) by inductive hypothesis

$$(\varphi = \langle a \rangle \psi) \quad \mathcal{G}, s \models \langle a \rangle \psi$$

$$\Rightarrow \text{ex. } s \xrightarrow{a} s' \text{ with } \mathcal{G}, s' \models \psi$$

$$\mathcal{G}, s \sim \mathcal{T}, t \Rightarrow \text{ex. } t \xrightarrow{a} t' \text{ with } \mathcal{G}, s' \sim \mathcal{T}, t'$$

$$\Rightarrow \mathcal{T}, t' \models \psi$$

Bisimulation invariance

Theorem

Two **finite** transition systems \mathfrak{G}, s and \mathfrak{T}, t are **bisimilar** if, and only if,

$$\mathfrak{G}, s \models \varphi \iff \mathfrak{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Proof (for \Rightarrow) induction on φ

$$(\varphi = P) \quad s \in P \iff t \in P$$

(boolean combinations) by inductive hypothesis

$$(\varphi = \langle a \rangle \psi) \quad \mathfrak{G}, s \models \langle a \rangle \psi$$

$$\Rightarrow \text{ex. } s \xrightarrow{a} s' \text{ with } \mathfrak{G}, s' \models \psi$$

$$\mathfrak{G}, s \sim \mathfrak{T}, t \Rightarrow \text{ex. } t \xrightarrow{a} t' \text{ with } \mathfrak{G}, s' \sim \mathfrak{T}, t'$$

$$\Rightarrow \mathfrak{T}, t' \models \psi$$

$$\Rightarrow \mathfrak{T}, t \models \langle a \rangle \psi$$

Bisimulation invariance

Theorem

Two **finite** transition systems \mathfrak{G}, s and \mathfrak{T}, t are **bisimilar** if, and only if,

$$\mathfrak{G}, s \models \varphi \quad \Leftrightarrow \quad \mathfrak{T}, t \models \varphi, \quad \text{for every modal formula } \varphi .$$

Theorem

Every satisfiable modal formula has a model that is a finite tree.

Bisimulation invariance

Theorem

Two **finite** transition systems \mathfrak{S}, s and \mathfrak{T}, t are **bisimilar** if, and only if,

$$\mathfrak{S}, s \models \varphi \quad \Leftrightarrow \quad \mathfrak{T}, t \models \varphi, \quad \text{for every modal formula } \varphi.$$

Theorem

Every satisfiable modal formula has a model that is a finite tree.

Definition

A formula $\varphi(x)$ is **bisimulation invariant** if

$$\mathfrak{S}, s \sim \mathfrak{T}, t \quad \text{implies} \quad \mathfrak{S} \models \varphi(s) \Leftrightarrow \mathfrak{T} \models \varphi(t).$$

Theorem

A first-order formula φ is equivalent to a **modal formula** if, and only if, it is **bisimulation invariant**.

First-Order Modal Logic

Syntax

first-order logic with modal operators $\langle a \rangle \varphi$ and $[a] \varphi$

First-Order Modal Logic

Syntax

first-order logic with modal operators $\langle a \rangle \varphi$ and $[a] \varphi$

Models

transition systems where each state s is labelled with a Σ -structure \mathfrak{A}_s such that

$$s \rightarrow^a t \quad \text{implies} \quad \mathfrak{A}_s \subseteq \mathfrak{A}_t$$

First-Order Modal Logic

Syntax

first-order logic with modal operators $\langle a \rangle \varphi$ and $[a] \varphi$

Models

transition systems where each state s is labelled with a Σ -structure \mathfrak{A}_s such that

$$s \rightarrow^a t \text{ implies } \mathfrak{A}_s \subseteq \mathfrak{A}_t$$

Examples

- ▶ $\Box \forall x \varphi(x) \rightarrow \forall x \Box \varphi(x)$ is valid.
- ▶ $\forall x \Box \varphi(x) \rightarrow \Box \forall x \varphi(x)$ is not valid.

Tableaux

The Entailment Relation

Consequence

ψ is a **consequence** of Γ ($\Gamma \models \psi$) if, and only if, for all transition systems \mathfrak{S} ,

$$\mathfrak{S}, s \models \varphi, \quad \text{for all } s \in S \text{ and } \varphi \in \Gamma,$$

implies that

$$\mathfrak{S}, s \models \psi, \quad \text{for all } s \in S.$$

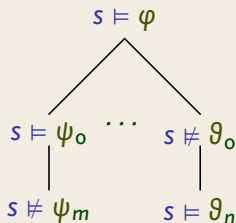
Tableau Proofs

Statements

$s \models \varphi$ $s \not\models \varphi$ $s \rightarrow^a t$

s, t state labels, φ a modal formula

Rules



Tableaux

Construction

A **tableau** for a formula φ is constructed as follows:

- ▶ start with $s_0 \neq \varphi$
- ▶ choose a branch of the tree
- ▶ choose a statement $s \models \psi / s \neq \psi$ on the branch
- ▶ choose a rule with head $s \models \psi / s \neq \psi$
- ▶ add it at the bottom of the branch
- ▶ repeat until every branch contains both statements $s \models \psi$ and $s \neq \psi$ for some formula ψ

Tableaux

Construction

A **tableau** for a formula φ is constructed as follows:

- ▶ start with $s_0 \not\models \varphi$
- ▶ choose a branch of the tree
- ▶ choose a statement $s \models \psi / s \not\models \psi$ on the branch
- ▶ choose a rule with head $s \models \psi / s \not\models \psi$
- ▶ add it at the bottom of the branch
- ▶ repeat until every branch contains both statements $s \models \psi$ and $s \not\models \psi$ for some formula ψ

Tableaux with premises Γ

- ▶ choose a branch, a state s on the branch, a premise $\psi \in \Gamma$, and add $s \models \psi$ to the branch

Rules

$$\begin{array}{c} s \models \neg\varphi \\ | \\ s \not\models \varphi \end{array}$$

$$\begin{array}{c} s \not\models \neg\varphi \\ | \\ s \models \varphi \end{array}$$

$$\begin{array}{c} s \models \varphi \wedge \psi \\ | \\ s \models \varphi \\ | \\ s \models \psi \end{array}$$

$$\begin{array}{c} s \not\models \varphi \wedge \psi \\ / \quad \backslash \\ s \not\models \varphi \quad s \not\models \psi \end{array}$$

$$\begin{array}{c} s \models \varphi \vee \psi \\ / \quad \backslash \\ s \models \varphi \quad s \models \psi \end{array}$$

$$\begin{array}{c} s \not\models \varphi \vee \psi \\ | \\ s \not\models \varphi \\ | \\ s \not\models \psi \end{array}$$

$$\begin{array}{c} s \models \varphi \rightarrow \psi \\ / \quad \backslash \\ s \not\models \varphi \quad s \models \psi \end{array}$$

$$\begin{array}{c} s \not\models \varphi \rightarrow \psi \\ | \\ s \models \varphi \\ | \\ s \not\models \psi \end{array}$$

$$\begin{array}{c} s \models \varphi \leftrightarrow \psi \\ / \quad \backslash \\ s \models \varphi \quad s \not\models \varphi \\ | \quad \quad | \\ s \models \psi \quad s \not\models \psi \end{array}$$

$$\begin{array}{c} s \not\models \varphi \leftrightarrow \psi \\ / \quad \backslash \\ s \models \varphi \quad s \not\models \varphi \\ | \quad \quad | \\ s \not\models \psi \quad s \models \psi \end{array}$$

Rules

$$s \models \langle a \rangle \varphi$$

$$s \xrightarrow{a} t$$

$$t \models \varphi$$

$$s \not\models \langle a \rangle \varphi$$

$$t' \not\models \varphi$$

$$s \models [a] \varphi$$

$$t' \models \varphi$$

$$s \not\models [a] \varphi$$

$$s \xrightarrow{a} t$$

$$t \not\models \varphi$$

$$s \models \forall x \varphi$$

$$s \models \varphi[x \mapsto u]$$

$$s \not\models \forall x \varphi$$

$$s \not\models \varphi[x \mapsto c]$$

$$s \models \exists x \varphi$$

$$s \models \varphi[x \mapsto c]$$

$$s \not\models \exists x \varphi$$

$$s \not\models \varphi[x \mapsto u]$$

t a new state, t' every state with entry $s \xrightarrow{a} t'$ on the branch,
 c a new constant symbol, u an arbitrary term

Example $\models \Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$

Example $\models \Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$

$s \not\models \Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$

$s \models \Box(\varphi \rightarrow \psi)$

$s \not\models \Box\varphi \rightarrow \Box\psi$

$s \models \Box\varphi$

$s \not\models \Box\psi$

$s \rightarrow t$

$t \not\models \psi$

$t \models \varphi$

$t \models \varphi \rightarrow \psi$

$t \not\models \varphi$

$t \models \psi$

Example $\varphi \models \Box\varphi$

Example $\varphi \models \Box\varphi$

$s \not\models \Box\varphi$

|

$s \rightarrow t$

|

$t \not\models \varphi$

|

$t \models \varphi$

Example $\models \Box \forall x \varphi \rightarrow \forall x \Box \varphi$

Example $\models \Box \forall x \varphi \rightarrow \forall x \Box \varphi$

$s \not\models \Box \forall x \varphi \rightarrow \forall x \Box \varphi$

$s \models \Box \forall x \varphi$

$s \not\models \forall x \Box \varphi$

$s \not\models \Box \varphi [x \mapsto c]$

$s \rightarrow t$

$t \not\models \varphi [x \mapsto c]$

$t \models \forall x \varphi$

$t \models \varphi [x \mapsto c]$

Soundness and Completeness

Theorem

A modal formula φ is a consequence of Γ if, and only if, there exists a tableau T for φ with premises Γ where every branch is contradictory.

Soundness and Completeness

Theorem

A modal formula φ is a consequence of Γ if, and only if, there exists a tableau T for φ with premises Γ where every branch is contradictory.

Theorem

Satisfiability for propositional modal logic is in **deterministic linear space**.

Theorem

Satisfiability for first-order modal logic is **undecidable**.

Temporal Logics

Linear Temporal Logic (LTL)

Speaks about **paths**. $P \longrightarrow \bullet \longrightarrow \bullet \longrightarrow P, Q \longrightarrow Q \longrightarrow \bullet \longrightarrow \dots$

Syntax

- ▶ atomic predicates P, Q, \dots
- ▶ boolean operations \wedge, \vee, \neg
- ▶ next $X\varphi$
- ▶ until $\varphi U \psi$
- ▶ finally $F\varphi := \top U \varphi$
- ▶ generally $G\varphi := \neg F \neg \varphi$

Examples

- | | |
|-------------------------|---|
| FP | a state in P is reachable |
| GFP | we can reach infinitely many states in P |
| $(\neg P)U(P \wedge Q)$ | the first reachable state in P is also in Q |

Linear Temporal Logic (LTL)

Theorem

Let L be a set of paths. The following statements are equivalent:

- ▶ L can be defined in LTL.
- ▶ L can be defined in first-order logic.
- ▶ L can be defined by a star-free regular expression.

Linear Temporal Logic (LTL)

Theorem

Let L be a set of paths. The following statements are equivalent:

- ▶ L can be defined in LTL.
- ▶ L can be defined in first-order logic.
- ▶ L can be defined by a star-free regular expression.

Translation LTL to FO

$$P^* := P(x)$$

$$(\varphi \wedge \psi)^* := \varphi^*(x) \wedge \psi^*(x)$$

$$(\varphi \vee \psi)^* := \varphi^*(x) \vee \psi^*(x)$$

$$(\neg\varphi)^* := \neg\varphi^*(x)$$

$$(X\varphi)^* := \exists y[x < y \wedge \neg\exists z(x < z \wedge z < y) \wedge \varphi^*(y)]$$

$$(\varphi U \psi)^* := \exists y[x \leq y \wedge \psi^*(y) \wedge \forall z[x \leq z \wedge z < y \rightarrow \varphi^*(z)]]$$

Linear Temporal Logic (LTL)

Theorem

Satisfiability of LTL formulae is **PSPACE-complete**.

Theorem

Model checking $\mathfrak{G}, s \models \varphi$ for LTL is **PSPACE-complete**. It can be done in

time $\mathcal{O}(|S| \cdot 2^{\mathcal{O}(|\varphi|)})$ or **space** $\mathcal{O}((|\varphi| + \log |S|)^2)$.

Linear Temporal Logic (LTL)

Theorem

Satisfiability of LTL formulae is **PSPACE-complete**.

Theorem

Model checking $\mathfrak{G}, s \models \varphi$ for LTL is **PSPACE-complete**. It can be done in

$$\text{time } \mathcal{O}(|S| \cdot 2^{\mathcal{O}(|\varphi|)}) \quad \text{or} \quad \text{space } \mathcal{O}((|\varphi| + \log |S|)^2).$$

Formula complexity: **PSPACE-complete**

Data complexity: **NLOGSPACE-complete**

Computation Tree Logic (CTL and CTL*)

Applies LTL-formulae to the branches of a tree.

Syntax (of CTL*)

- ▶ **state formulae** φ :

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid A\psi \mid E\psi$$

- ▶ **path formulae** ψ :

$$\psi ::= \varphi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \neg\psi \mid X\psi \mid \psi U \psi \mid F\psi \mid G\psi$$

Examples

EFP a state in P is reachable

AFP every branch contains a state in P

EGFP there is a branch with infinitely many P

EGEFP there is a branch such that we can reach P from every of its states

Computation Tree Logic (CTL and CTL*)

Theorem

Satisfiability for CTL is **EXPTIME-complete**.

Model checking $\mathfrak{G}, s \models \varphi$ for CTL is **P-complete**. It can be done in

time $\mathcal{O}(|\varphi| \cdot |S|)$ or **space** $\mathcal{O}(|\varphi| \cdot \log^2(|\varphi| \cdot |S|))$.

Data complexity: **NLOGSPACE-complete**

Computation Tree Logic (CTL and CTL*)

Theorem

Satisfiability for CTL is **EXPTIME-complete**.

Model checking $\mathfrak{G}, s \models \varphi$ for CTL is **P-complete**. It can be done in
time $\mathcal{O}(|\varphi| \cdot |S|)$ or **space** $\mathcal{O}(|\varphi| \cdot \log^2(|\varphi| \cdot |S|))$.

Data complexity: **NLOGSPACE-complete**

Theorem

Satisfiability for CTL* is **2EXPTIME-complete**.

Model checking $\mathfrak{G}, s \models \varphi$ for CTL* is **PSPACE-complete**. It can be done in

time $\mathcal{O}(|S|^2 \cdot 2^{\mathcal{O}(|\varphi|)})$ or **space** $\mathcal{O}(|\varphi|(|\varphi| + \log|S|)^2)$.

Formula complexity: **PSPACE-complete**

Data complexity: **NLOGSPACE-complete**

Fixed points

Theorem (Knaster, Tarski)

Let $\langle A, \leq \rangle$ be a **complete** partial order and $f : A \rightarrow A$ **monotone**.
Then f has a **least** and a **greatest fixed point** and

$$\text{lfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\perp) \quad \text{and} \quad \text{gfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\top)$$

Fixed points

Theorem (Knaster, Tarski)

Let $\langle A, \leq \rangle$ be a **complete** partial order and $f : A \rightarrow A$ **monotone**.
Then f has a **least** and a **greatest fixed point** and

$$\text{lfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\perp) \quad \text{and} \quad \text{gfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\top)$$

Examples $\langle \mathcal{P}(\mathbb{N}), \subseteq \rangle$

- $f(X) := (X \setminus A) \cup B$

Fixed points

Theorem (Knaster, Tarski)

Let $\langle A, \leq \rangle$ be a **complete** partial order and $f : A \rightarrow A$ **monotone**.
Then f has a **least** and a **greatest fixed point** and

$$\text{lfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\perp) \quad \text{and} \quad \text{gfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\top)$$

Examples $\langle \mathcal{P}(\mathbb{N}), \subseteq \rangle$

- $f(X) := (X \setminus A) \cup B$

$$\text{lfp}(f) = B \quad \text{and} \quad \text{gfp}(f) = (\mathbb{N} \setminus A) \cup B$$

- $f(X) := \{y \mid y \leq x \in X\}$

Fixed points

Theorem (Knaster, Tarski)

Let $\langle A, \leq \rangle$ be a **complete** partial order and $f : A \rightarrow A$ **monotone**.
Then f has a **least** and a **greatest fixed point** and

$$\text{lfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\perp) \quad \text{and} \quad \text{gfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\top)$$

Examples $\langle \mathcal{P}(\mathbb{N}), \subseteq \rangle$

- $f(X) := (X \setminus A) \cup B$

$$\text{lfp}(f) = B \quad \text{and} \quad \text{gfp}(f) = (\mathbb{N} \setminus A) \cup B$$

- $f(X) := \{y \mid y \leq x \in X\}$

fixed points: $\emptyset, \{0\}, \{0, 1\}, \dots, \{0, \dots, n\}, \dots, \mathbb{N}$

- $f(X) := \mathbb{N} \setminus X$

Fixed points

Theorem (Knaster, Tarski)

Let $\langle A, \leq \rangle$ be a **complete** partial order and $f : A \rightarrow A$ **monotone**.
Then f has a **least** and a **greatest fixed point** and

$$\text{lfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\perp) \quad \text{and} \quad \text{gfp}(f) = \lim_{\alpha \rightarrow \infty} f^\alpha(\top)$$

Examples $\langle \mathcal{P}(\mathbb{N}), \subseteq \rangle$

- $f(X) := (X \setminus A) \cup B$

$$\text{lfp}(f) = B \quad \text{and} \quad \text{gfp}(f) = (\mathbb{N} \setminus A) \cup B$$

- $f(X) := \{y \mid y \leq x \in X\}$

fixed points: $\emptyset, \{0\}, \{0, 1\}, \dots, \{0, \dots, n\}, \dots, \mathbb{N}$

- $f(X) := \mathbb{N} \setminus X$ has no fixed points

Ordinals

$0, 1, 2, 3, \dots$

Ordinals

$0, 1, 2, 3, \dots, \omega$

Ordinals

$0, 1, 2, 3, \dots, \omega, \omega + 1, \omega + 2, \dots$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$
 $\omega^3, \dots \omega^4, \dots \omega^5, \dots$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

ω^ω

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

$\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

$\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots \epsilon, \dots$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

$\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots \epsilon, \dots \omega_1, \dots \omega_2, \dots$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

$\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots \epsilon, \dots \omega_1, \dots \omega_2, \dots$

3 Kinds

- 0

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

$\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots \epsilon, \dots \omega_1, \dots \omega_2, \dots$

3 Kinds

- 0
- successor $\alpha + 1$

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$

$\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$

$\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots \epsilon, \dots \omega_1, \dots \omega_2, \dots$

3 Kinds

- 0
- successor $\alpha + 1$
- limit δ

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$
 $\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$
 $\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots \varepsilon, \dots \omega_1, \dots \omega_2, \dots$

3 Kinds

- 0
- successor $\alpha + 1$
- limit δ

Proposition

Every non-empty set of ordinals has a least element.

Ordinals

$0, 1, 2, 3, \dots \omega, \omega + 1, \omega + 2, \dots \omega + \omega = \omega^2, \omega^2 + 1, \omega^2 + 2, \dots$
 $\omega^3, \dots \omega^4, \dots \omega^5, \dots \omega\omega = \omega^2, \dots \omega^3, \dots \omega^4, \dots$
 $\omega^\omega, \dots \omega^{\omega^\omega}, \dots \omega^{\omega^{\omega^\omega}}, \dots \epsilon, \dots \omega_1, \dots \omega_2, \dots$

3 Kinds

- 0
- successor $\alpha + 1$
- limit δ

Iteration

$$f^0(x) := x,$$
$$f^{\alpha+1}(x) := f(f^\alpha(x)),$$
$$f^\delta(x) := \sup_{\alpha < \delta} f^\alpha(x),$$

for limit ordinals δ .
(for gfp, take inf instead of sup)

Proof

Monotonicity $f^\alpha(\perp) \leq f^\beta(\perp)$ for $\alpha \leq \beta$

induction on α

$(\alpha = 0)$ $\perp \leq f^\beta(\perp)$ (\perp is the least element)

Proof

Monotonicity $f^\alpha(\perp) \leq f^\beta(\perp)$ for $\alpha \leq \beta$

induction on α

$(\alpha = 0)$ $\perp \leq f^\beta(\perp)$ (\perp is the least element)

$(\alpha = \alpha' + 1)$ If $\beta = \beta' + 1$, we have

$$f^{\alpha'}(\perp) \leq f^{\beta'}(\perp) \Rightarrow f^{\alpha'+1}(\perp) \leq f^{\beta'+1}(\perp). \quad (f \text{ is monotone})$$

Proof

Monotonicity $f^\alpha(\perp) \leq f^\beta(\perp)$ for $\alpha \leq \beta$

induction on α

$(\alpha = 0)$ $\perp \leq f^\beta(\perp)$ (\perp is the least element)

$(\alpha = \alpha' + 1)$ If $\beta = \beta' + 1$, we have

$$f^{\alpha'}(\perp) \leq f^{\beta'}(\perp) \Rightarrow f^{\alpha'+1}(\perp) \leq f^{\beta'+1}(\perp). \quad (f \text{ is monotone})$$

If β is a limit, we have

$$f^\alpha(\perp) \leq \sup_{\gamma < \beta} f^\gamma(\perp) = f^\beta(\perp). \quad (\text{definition of supremum})$$

Proof

Monotonicity $f^\alpha(\perp) \leq f^\beta(\perp)$ for $\alpha \leq \beta$

induction on α

$(\alpha = 0)$ $\perp \leq f^\beta(\perp)$ (\perp is the least element)

$(\alpha = \alpha' + 1)$ If $\beta = \beta' + 1$, we have

$$f^{\alpha'}(\perp) \leq f^{\beta'}(\perp) \Rightarrow f^{\alpha'+1}(\perp) \leq f^{\beta'+1}(\perp). \quad (f \text{ is monotone})$$

If β is a limit, we have

$$f^\alpha(\perp) \leq \sup_{\gamma < \beta} f^\gamma(\perp) = f^\beta(\perp). \quad (\text{definition of supremum})$$

$(\alpha \text{ limit})$ For $\gamma < \alpha$, we have $f^\gamma(\perp) \leq f^\beta(\perp)$. Hence,

$$f^\alpha(\perp) = \sup_{\gamma < \alpha} f^\gamma(\perp) \leq f^\beta(\perp). \quad (\text{inductive hypothesis})$$

Proof

Existence

exists α with $f^\alpha(\perp) = f^{\alpha+1}(\perp)$ (there are only $|A|$ different values)

Proof

Existence

exists α with $f^\alpha(\perp) = f^{\alpha+1}(\perp)$ (there are only $|A|$ different values)

Least fixed point

$a = f(a)$ fixed point, $f^\alpha(\perp) = f^{\alpha+1}(\perp)$

Proof

Existence

exists α with $f^\alpha(\perp) = f^{\alpha+1}(\perp)$ (there are only $|A|$ different values)

Least fixed point

$a = f(a)$ fixed point, $f^\alpha(\perp) = f^{\alpha+1}(\perp)$

$\perp \leq a$

Proof

Existence

exists α with $f^\alpha(\perp) = f^{\alpha+1}(\perp)$ (there are only $|A|$ different values)

Least fixed point

$a = f(a)$ fixed point, $f^\alpha(\perp) = f^{\alpha+1}(\perp)$

$\perp \leq a \Rightarrow f^\alpha(\perp) \leq f^\alpha(a) = a$ (f^α monotone, by induction on α)

The modal μ -calculus (L_μ)

Adds recursion to modal logic.

Syntax

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle a \rangle \varphi \mid [a] \varphi \mid \mu X. \varphi(X) \mid \nu X. \varphi(X)$$

(X positive in $\mu X. \varphi(X)$ and $\nu X. \varphi(X)$)

The modal μ -calculus (L_μ)

Adds recursion to modal logic.

Syntax

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle a \rangle \varphi \mid [a] \varphi \mid \mu X. \varphi(X) \mid \nu X. \varphi(X)$$

(X positive in $\mu X. \varphi(X)$ and $\nu X. \varphi(X)$)

Semantics

$$F_\varphi(X) := \{ s \in S \mid \mathfrak{G}, s \models \varphi(X) \}$$

$$\mu X. \varphi(X) : X_0 := \emptyset, \quad X_{i+1} := F_\varphi(X_i)$$

$$\nu X. \varphi(X) : X_0 := S, \quad X_{i+1} := F_\varphi(X_i)$$

The modal μ -calculus (L_μ)

Adds recursion to modal logic.

Syntax

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle a \rangle \varphi \mid [a] \varphi \mid \mu X. \varphi(X) \mid \nu X. \varphi(X)$$

(X positive in $\mu X. \varphi(X)$ and $\nu X. \varphi(X)$)

Semantics

$$F_\varphi(X) := \{ s \in S \mid \mathfrak{G}, s \models \varphi(X) \}$$

$$\mu X. \varphi(X) : X_0 := \emptyset, \quad X_{i+1} := F_\varphi(X_i)$$

$$\nu X. \varphi(X) : X_0 := S, \quad X_{i+1} := F_\varphi(X_i)$$

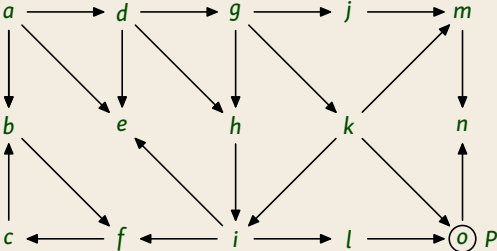
Examples

$\mu X(P \vee \diamond X)$ a state in P is reachable

$\nu X(P \wedge \diamond X)$ there is a branch with all states in P

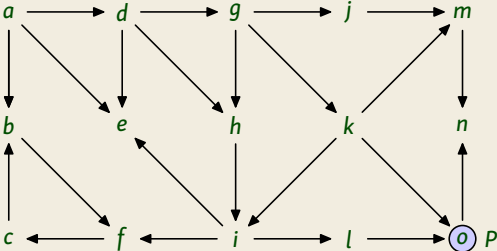
Examples

$\mu X(P \vee \Diamond X)$ a state in P is reachable



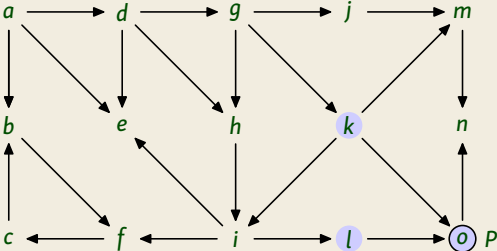
Examples

$\mu X(P \vee \Diamond X)$ a state in P is reachable



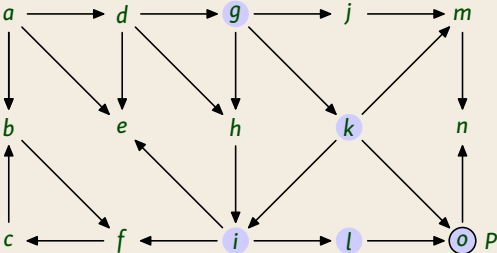
Examples

$\mu X(P \vee \Diamond X)$ a state in P is reachable



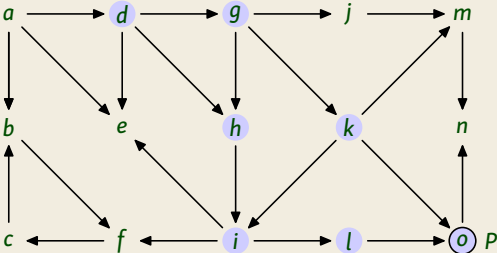
Examples

$\mu X(P \vee \Diamond X)$ a state in P is reachable



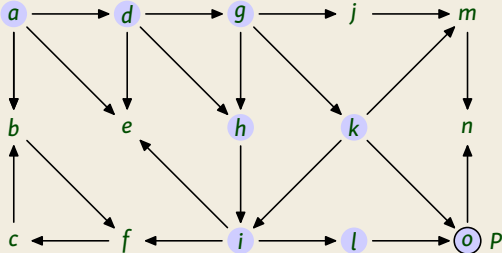
Examples

$\mu X(P \vee \Diamond X)$ a state in P is reachable



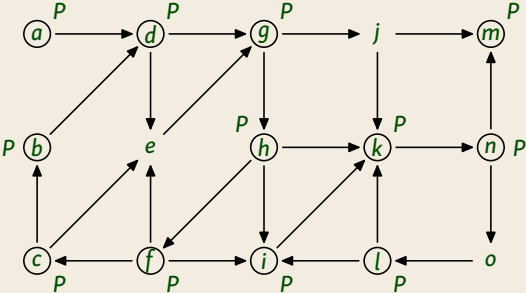
Examples

$\mu X(P \vee \Diamond X)$ a state in P is reachable



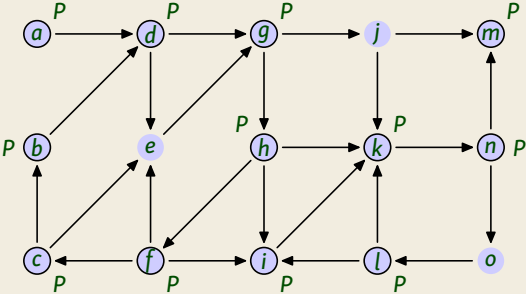
Examples

$\nu X(P \wedge \Diamond X)$ there is a branch with all states in P



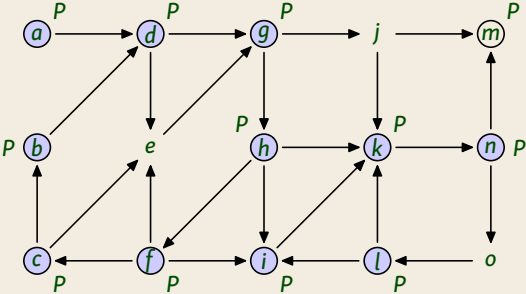
Examples

$\forall X(P \wedge \Diamond X)$ there is a branch with all states in P



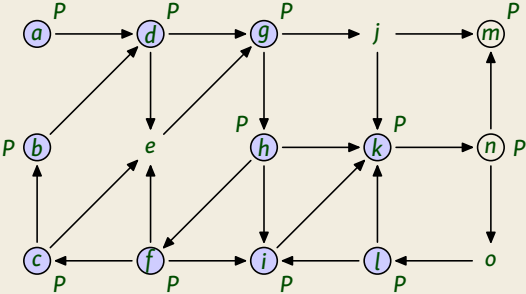
Examples

$\nu X(P \wedge \diamond X)$ there is a branch with all states in P



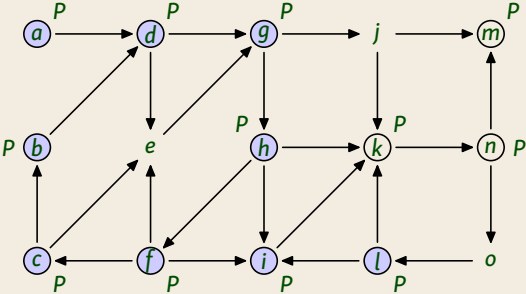
Examples

$\nu X(P \wedge \diamond X)$ there is a branch with all states in P



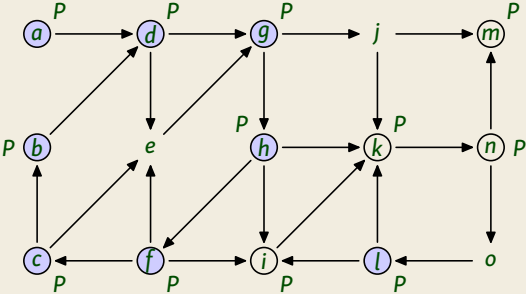
Examples

$\nu X(P \wedge \Diamond X)$ there is a branch with all states in P



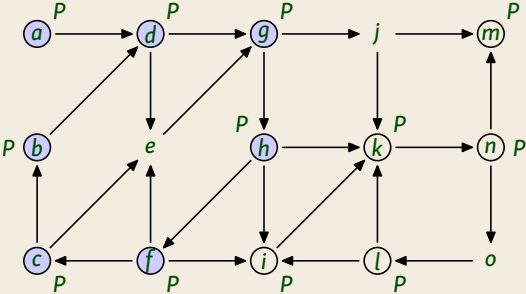
Examples

$\forall X(P \wedge \Diamond X)$ there is a branch with all states in P



Examples

$\forall X(P \wedge \Diamond X)$ there is a branch with all states in P



Expressive power

Theorem

For every CTL*-formula φ there exists an equivalent formula φ^* of the modal μ -calculus.

Expressive power

Theorem

For every CTL*-formula φ there exists an equivalent formula φ^* of the modal μ -calculus.

Proof (for CTL)

$$P^* := P$$

$$(\varphi \wedge \psi)^* := \varphi^* \wedge \psi^*$$

$$(\varphi \vee \psi)^* := \varphi^* \vee \psi^*$$

$$(\neg\varphi)^* := \neg\varphi^*$$

$$(EX\varphi)^* := \diamond\varphi^*$$

$$(AX\varphi)^* := \square\varphi^*$$

$$(E\varphi U\psi)^* := \mu X[\psi^* \vee (\varphi^* \wedge \diamond X)]$$

$$(A\varphi U\psi)^* := \mu X[\psi^* \vee (\varphi^* \wedge \square X \wedge \diamond T)]$$

The modal μ -calculus (L_μ)

Theorem

A regular tree language can be defined in the **modal μ -calculus** if, and only if, it is **bisimulation invariant**.

Theorem

Satisfiability of μ -calculus formulae is **decidable** and complete for **exponential time**.

Model checking $\mathfrak{S}, s \models \varphi$ for the modal μ -calculus can be done in **time** $\mathcal{O}((|\varphi| \cdot |S|)^{|\varphi|})$.

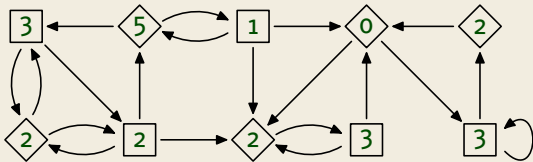
(The satisfiability algorithm uses tree automata and parity games.)

Parity Games

$$\mathcal{G} = \langle V_{\diamond}, V_{\square}, E, \Omega \rangle \quad \Omega : V \rightarrow \mathbb{N}$$

Infinite plays v_0, v_1, \dots are **won** by Player \diamond if

$\liminf_{n \rightarrow \infty} \Omega(v_n)$ is even.

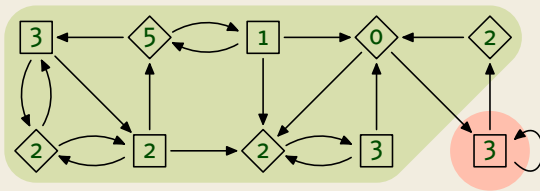


Parity Games

$$\mathcal{G} = \langle V_{\diamond}, V_{\square}, E, \Omega \rangle \quad \Omega : V \rightarrow \mathbb{N}$$

Infinite plays v_0, v_1, \dots are **won** by Player \diamond if

$\liminf_{n \rightarrow \infty} \Omega(v_n)$ is even.



Parity Games

$$\mathfrak{G} = \langle V_{\diamond}, V_{\square}, E, \Omega \rangle \quad \Omega : V \rightarrow \mathbb{N}$$

Infinite plays v_0, v_1, \dots are **won** by Player \diamond if

$$\liminf_{n \rightarrow \infty} \Omega(v_n) \text{ is even.}$$

Theorem

Parity games are **positionally determined**: from each position some player has a positional/memory-less winning strategy.

Parity Games

$$\mathcal{G} = \langle V_{\diamond}, V_{\square}, E, \Omega \rangle \quad \Omega : V \rightarrow \mathbb{N}$$

Infinite plays v_0, v_1, \dots are **won** by Player \diamond if

$$\liminf_{n \rightarrow \infty} \Omega(v_n) \text{ is even.}$$

Theorem

Parity games are **positionally determined**: from each position some player has a positional/memory-less winning strategy.

Theorem

Computing the winning region of a parity game with n positions and d priorities can be done in time $n^{\mathcal{O}(\log d)}$.

Model-Checking Games

game for $\mathfrak{G}, s_0 \models \varphi?$ (φ μ -formula in negation normal form)

Model-Checking Games

game for $\mathfrak{G}, s_0 \models \varphi?$ (φ μ -formula in negation normal form)

Positions

Player \diamond : $\langle s, \psi \rangle$ for $s \in S$ and ψ a subformula

$$\begin{array}{lll} \psi = \psi_0 \vee \psi_1, & \psi = P \text{ and } s \notin P, & \psi = \mu X. \psi_0, \\ \psi = \langle a \rangle \psi_0, & \psi = \neg P \text{ and } s \in P, & \psi = \nu X. \psi_0, \\ & & \psi = X. \end{array}$$

Player \square : $[s, \psi]$ for $s \in S$ and ψ a subformula

$$\begin{array}{ll} \psi = \psi_0 \wedge \psi_1, & \psi = P \text{ and } s \in P, \\ \psi = [a] \psi_0, & \psi = \neg P \text{ and } s \notin P. \end{array}$$

Model-Checking Games

game for $\mathfrak{G}, s_0 \models \varphi?$ (φ μ -formula in negation normal form)

Positions

Player \diamond : $\langle s, \psi \rangle$ for $s \in S$ and ψ a subformula

$$\begin{array}{lll} \psi = \psi_0 \vee \psi_1, & \psi = P \text{ and } s \notin P, & \psi = \mu X. \psi_0, \\ \psi = \langle a \rangle \psi_0, & \psi = \neg P \text{ and } s \in P, & \psi = \nu X. \psi_0, \\ & & \psi = X. \end{array}$$

Player \square : $[s, \psi]$ for $s \in S$ and ψ a subformula

$$\begin{array}{ll} \psi = \psi_0 \wedge \psi_1, & \psi = P \text{ and } s \in P, \\ \psi = [a] \psi_0, & \psi = \neg P \text{ and } s \notin P. \end{array}$$

Initial position $\langle s_0, \varphi \rangle$ or $[s_0, \varphi]$

Model-Checking Games

game for $\mathfrak{G}, s_0 \models \varphi$? (φ μ -formula in negation normal form)

Edges ((s, ψ) means either $\langle s, \psi \rangle$ or $[s, \psi]$.)

$$\langle s, \psi_0 \vee \psi_1 \rangle \rightarrow (s, \psi_i),$$

$$[s, \psi_0 \wedge \psi_1] \rightarrow (s, \psi_i),$$

$$\langle s, \mu X. \psi \rangle \rightarrow \psi,$$

$$\langle s, \nu X. \psi \rangle \rightarrow \psi,$$

$$\langle s, X \rangle \rightarrow \langle s, \mu X. \psi \rangle \text{ or } \langle s, \nu X. \psi \rangle,$$

$$\langle s, \langle a \rangle \psi \rangle \rightarrow (t, \psi) \quad \text{for every } s \rightarrow^a t,$$

$$[s, [a] \psi] \rightarrow (t, \psi) \quad \text{for every } s \rightarrow^a t.$$

Model-Checking Games

game for $\mathfrak{G}, s_0 \models \varphi$? (φ μ -formula in negation normal form)

Edges ($\langle s, \psi \rangle$ means either $\langle s, \psi \rangle$ or $[s, \psi]$.)

$$\langle s, \psi_0 \vee \psi_1 \rangle \rightarrow (s, \psi_i),$$

$$[s, \psi_0 \wedge \psi_1] \rightarrow (s, \psi_i),$$

$$\langle s, \mu X. \psi \rangle \rightarrow \psi,$$

$$\langle s, \nu X. \psi \rangle \rightarrow \psi,$$

$$\langle s, X \rangle \rightarrow \langle s, \mu X. \psi \rangle \text{ or } \langle s, \nu X. \psi \rangle,$$

$$\langle s, \langle a \rangle \psi \rangle \rightarrow (t, \psi) \quad \text{for every } s \rightarrow^a t,$$

$$[s, [a] \psi] \rightarrow (t, \psi) \quad \text{for every } s \rightarrow^a t.$$

Priorities (all other priorities big)

$$\Omega(\langle s, \mu X. \psi \rangle) := 2k + 1, \quad \text{if inside of } k \text{ fixed points.}$$

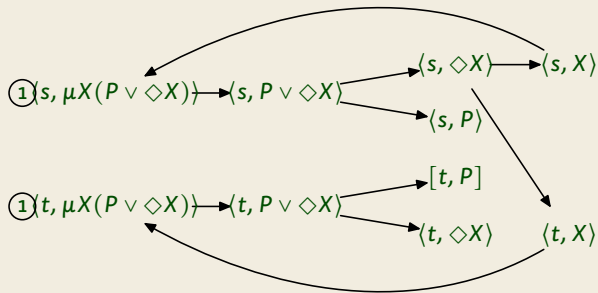
$$\Omega(\langle s, \nu X. \psi \rangle) := 2k.$$

Model-Checking Games

$$\mathcal{G} = \begin{array}{c} \curvearrowright \\ \textcircled{s} \longrightarrow \textcircled{t} \end{array} P \quad \varphi = \mu X(P \vee \diamond X)$$

Model-Checking Games

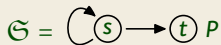
$$\mathcal{G} = \begin{array}{c} \textcircled{1} \\ \textcircled{s} \end{array} \rightarrow \begin{array}{c} \textcircled{2} \\ \textcircled{t} \end{array} P \quad \varphi = \mu X(P \vee \diamond X)$$



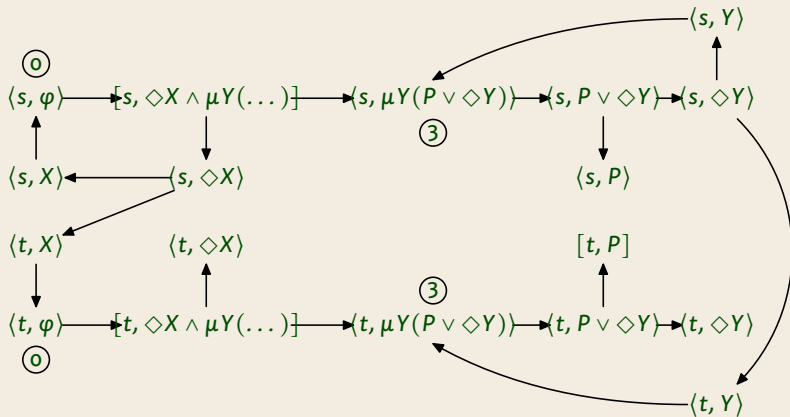
Model-Checking Games

$$\mathcal{G} = \begin{array}{c} \text{---} \\ \curvearrowright \\ \textcircled{s} \longrightarrow \textcircled{t} \end{array} P \quad \varphi = \nu X(\diamond X \wedge \mu Y(P \vee \diamond Y))$$

Model-Checking Games



$$\varphi = \nu X(\diamond X \wedge \mu Y(P \vee \diamond Y))$$



Description Logics

Description Logic

General Idea

Extend **modal logic** with operations that are **not bisimulation-invariant**.

Applications

Knowledge representation, deductive databases, system modelling, semantic web

Ingredients

- ▶ **individuals**: elements (Anna, John, Paul, Marry,...)
- ▶ **concepts**: unary predicates (person, male, female,...)
- ▶ **roles**: binary relations (has_child, is_married_to,...)
- ▶ **TBox**: terminology definitions
- ▶ **ABox**: assertions about the world

Example

TBox

$\text{man} := \text{person} \wedge \text{male}$

$\text{woman} := \text{person} \wedge \text{female}$

$\text{father} := \text{man} \wedge \exists \text{has_child}.\text{person}$

$\text{mother} := \text{woman} \wedge \exists \text{has_child}.\text{person}$

ABox

$\text{man}(\text{John})$

$\text{man}(\text{Paul})$

$\text{woman}(\text{Anna})$

$\text{woman}(\text{Marry})$

$\text{has_child}(\text{Anna}, \text{Paul})$

$\text{is_married_to}(\text{Anna}, \text{John})$

Syntax

Concepts

$$\varphi ::= P \mid \top \mid \perp \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \forall R\varphi \mid \exists R\varphi \mid (\geq nR) \mid (\leq nR)$$

Terminology axioms

$$\varphi \sqsubseteq \psi \quad \varphi \equiv \psi$$

TBox Axioms of the form $P \equiv \varphi$.

Assertions

$$\varphi(a) \quad R(a, b)$$

Extensions

- ▶ operations on roles: $R \cap S, R \cup S, R \circ S, \neg R, R^+, R^*, R^-$
- ▶ extended number restrictions: $(\geq nR)\varphi, (\leq nR)\varphi$

Algorithmic Problems

- ▶ **Satisfiability:** Is φ satisfiable?
- ▶ **Subsumption:** $\varphi \models \psi$?
- ▶ **Equivalence:** $\varphi \equiv \psi$?
- ▶ **Disjointness:** $\varphi \wedge \psi$ unsatisfiable?

All problems can be solved with standard methods like **tableaux** or **tree automata**.

Semantic Web: OWL (functional syntax)

```
Ontology(  
  Class(pp:man    complete  
        intersectionOf(pp:person pp:male))  
  Class(pp:woman  complete  
        intersectionOf(pp:person pp:female))  
  Class(pp:father complete  
        intersectionOf(pp:man  
          restriction(pp:has_child pp:person)))  
  Class(pp:mother complete  
        intersectionOf(pp:woman  
          restriction(pp:has_child pp:person)))  
  Individual(pp:John  type(pp:man))  
  Individual(pp:Paul  type(pp:man))  
  Individual(pp:Anna  type(pp:woman)  
            value(pp:has_child    pp:Paul)  
            value(pp:is_married_to pp:John))  
  Individual(pp:Marry type(pp:woman))  
)
```