IA008: Computational Logic

7. Modal Logic

Achim Blumensath
blumens@fi.muni.cz

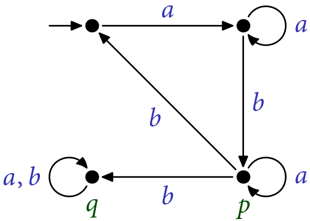
Faculty of Informatics, Masaryk University, Brno

Basic Concepts

Transition Systems

directed graph $\mathfrak{S} = \langle S, (E_a)_{a \in A}, (P_i)_{i \in I}, s_0 \rangle$ 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: $\Diamond\varphi, \Box\varphi$ if there are no edge labels

Formal semantics

$$\begin{aligned}
\mathfrak{S}, s \models P \quad &:\text{iff} \quad s \in P \\
\mathfrak{S}, s \models \varphi \wedge \psi \quad &:\text{iff} \quad \mathfrak{S}, s \models \varphi \text{ and } \mathfrak{S}, s \models \psi \\
\mathfrak{S}, s \models \varphi \vee \psi \quad &:\text{iff} \quad \mathfrak{S}, s \models \varphi \text{ or } \mathfrak{S}, s \models \psi \\
\mathfrak{S}, s \models \neg\varphi \quad &:\text{iff} \quad \mathfrak{S}, s \not\models \varphi \\
\mathfrak{S}, s \models \langle a \rangle \varphi \quad &:\text{iff} \quad \text{there is } s \to^{a} t \text{ such that } \mathfrak{S}, t \models \varphi \\
\mathfrak{S}, s \models [a]\varphi \quad &:\text{iff} \quad \text{for all } s \to^{a} t\text{, we have } \mathfrak{S}, t \models \varphi
\end{aligned}$$

Examples

$P \wedge \Diamond Q$	'The state is in P and there exists a transition to Q.'
$[a]\bot$	'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'
  - $\Box\varphi$ 'always in the future φ holds'
- **Epistemic Logic** talks about knowledge:
  - states: possible worlds
  - $\Diamond\varphi$ 'φ might be true'
  - $\Box\varphi$ 'φ is certainly true'

Examples: Temporal Logic

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

- "P never holds."

  $\neg \Diamond P$

- "After every P there is some Q."

  $\Box(P \rightarrow \Diamond Q)$

- "Once P holds, it holds forever."

  $\Box(P \rightarrow \Box P)$

- "There are infinitely many P."

  $\Box \Diamond P$

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

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

Proof

$$\begin{aligned}
P^* &:= P(x) \\
(\varphi \wedge \psi)^* &:= \varphi^*(x) \wedge \psi^*(x) \\
(\varphi \vee \psi)^* &:= \varphi^*(x) \vee \psi^*(x) \\
(\neg\varphi)^* &:= \neg\varphi^*(x) \\
(\langle a \rangle \varphi)^* &:= \exists y[E_a(x,y) \wedge \varphi^*(y)] \\
([a]\varphi)^* &:= \forall y[E_a(x,y) \rightarrow \varphi^*(y)]
\end{aligned}$$

Bisimulation

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

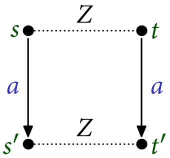
$Z \subseteq S \times T$ is a **bisimulation** if, for all $\langle s, t \rangle \in Z$,

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

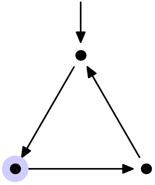
(forth) for every $s \to^a s'$, exists $t \to^a t'$ with $\langle s', t' \rangle \in Z$,

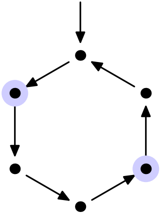
(back) for every $t \to^a t'$, exists $s \to^a s'$ with $\langle s', t' \rangle \in Z$.

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

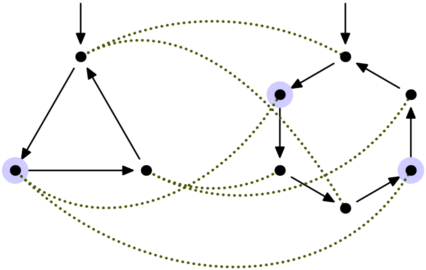


Examples

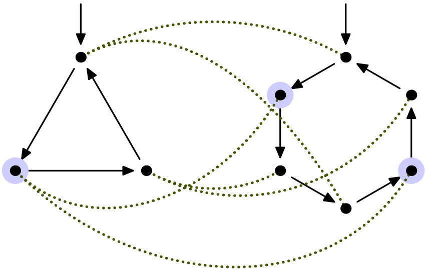


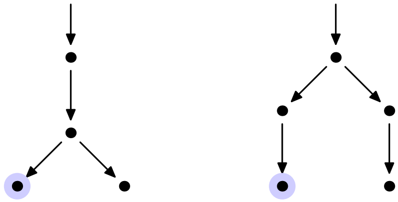


Examples

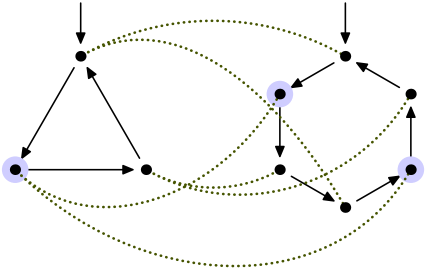


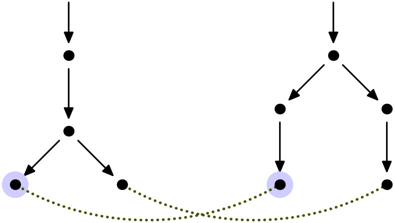
Examples



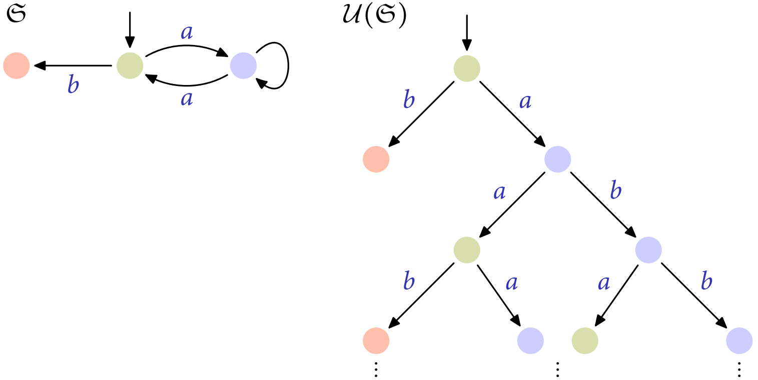


Examples





Unravelling



Lemma

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

Bisimulation invariance

Theorem

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

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

Proof (for $\Rightarrow$) induction on φ

$(\varphi = P)$ $s \in P \Leftrightarrow t \in P$

(boolean combinations) by inductive hypothesis

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

$\Rightarrow$ ex. $s \to^a s'$ with $\mathfrak{S}, s' \vDash \psi$

$\mathfrak{S}, s \sim \mathfrak{T}, t \quad \Rightarrow \quad$ ex. $t \to^a t'$ with $\mathfrak{S}, s' \sim \mathfrak{T}, t'$

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

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

Bisimulation invariance

Theorem

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

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

Theorem

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

Definition

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

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

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 $[a]\varphi$

Models

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

$$s \to^a t \quad \text{implies} \quad A_s \subseteq A_t$$

Examples

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

Tableaux

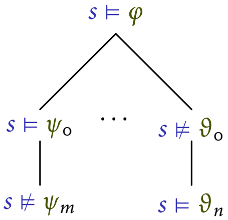
Tableau Proofs

Statements

$s \models \varphi \qquad s \not\models \varphi \qquad s \rightarrow^{a} t$

s, t state labels, φ a **modal formula**

Rules



Tableaux

Construction

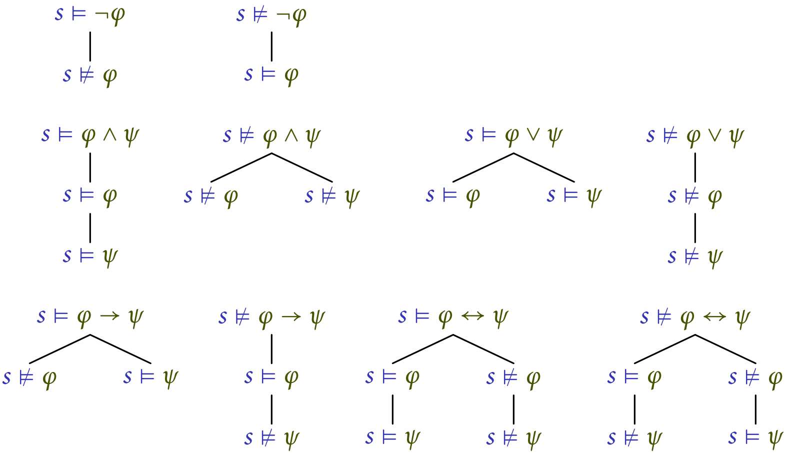
A **tableau** for a formula φ is constructed as follows:

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

Tableaux with premises Γ

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

Rules



s ⊨ ¬φ
s ⊭ φ
s ⊭ ¬φ
s ⊨ φ
s ⊨ φ ∧ ψ
s ⊨ φ
s ⊨ ψ
s ⊭ φ ∧ ψ
s ⊭ φ
s ⊭ ψ
s ⊨ φ ∨ ψ
s ⊨ φ
s ⊨ ψ
s ⊭ φ ∨ ψ
s ⊭ φ
s ⊭ ψ
s ⊨ φ → ψ
s ⊭ φ
s ⊨ ψ
s ⊭ φ → ψ
s ⊨ φ
s ⊭ ψ
s ⊨ φ ↔ ψ
s ⊨ φ
s ⊭ φ
s ⊨ ψ
s ⊭ ψ
s ⊭ φ ↔ ψ
s ⊨ φ
s ⊭ φ
s ⊭ ψ
s ⊨ ψ

Rules

$$\begin{array}{c} s \models \langle a \rangle \varphi \\ | \\ s \to^{a} t \\ | \\ t \models \varphi \end{array} \qquad \begin{array}{c} s \not\models \langle a \rangle \varphi \\ | \\ t' \not\models \varphi \end{array} \qquad \begin{array}{c} s \models [a] \varphi \\ | \\ t' \models \varphi \end{array} \qquad \begin{array}{c} s \not\models [a] \varphi \\ | \\ s \to^{a} t \\ | \\ t \not\models \varphi \end{array}$$

$$\begin{array}{c} s \models \forall x \varphi \\ | \\ s \models \varphi[x \mapsto u] \end{array} \qquad \begin{array}{c} s \not\models \forall x \varphi \\ | \\ s \not\models \varphi[x \mapsto c] \end{array} \qquad \begin{array}{c} s \models \exists x \varphi \\ | \\ s \models \varphi[x \mapsto c] \end{array} \qquad \begin{array}{c} s \not\models \exists x \varphi \\ | \\ s \not\models \varphi[x \mapsto u] \end{array}$$

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

Example $\varphi \vDash \Box\varphi$

$$s \nvDash \Box\varphi$$

$$s \to t$$

$$t \nvDash \varphi$$

$$t \vDash \varphi$$

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

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

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

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

$$s \models \Box\varphi$$

$$s \not\models \Box\psi$$

$$s \to t$$

$$t \not\models \psi$$

$$t \models \varphi$$

$$t \models \varphi \to \psi$$

$$t \not\models \varphi \qquad\qquad t \models \psi$$

Example $\vDash \Box\forall x\varphi \to \forall x \Box \varphi$

$$
\begin{array}{c}
s \nvDash \Box\forall x\varphi \to \forall x\Box\varphi \\
| \\
s \vDash \Box\forall x\varphi \\
| \\
s \nvDash \forall x\Box\varphi \\
| \\
s \nvDash \Box\varphi[x \mapsto c] \\
| \\
s \to t \\
| \\
t \nvDash \varphi[x \mapsto c] \\
| \\
t \vDash \forall x\varphi \\
| \\
t \vDash \varphi[x \mapsto c]
\end{array}
$$

Soundness and Completeness

Consequence

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

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

implies that

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

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.

Complexity

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.** $P \longrightarrow \bullet \longrightarrow \bullet \longrightarrow P, Q \longrightarrow Q \longrightarrow \bullet \longrightarrow \cdots$

Syntax

- atomic predicates $P, Q, \ldots$
- 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.

Translation LTL to FO

$$
\begin{aligned}
P^* &:= P(x) \\
(\varphi \wedge \psi)^* &:= \varphi^*(x) \wedge \psi^*(x) \\
(\varphi \vee \psi)^* &:= \varphi^*(x) \vee \psi^*(x) \\
(\neg\varphi)^* &:= \neg\varphi^*(x) \\
(X\varphi)^* &:= \exists y[x < y \wedge \neg\exists z(x < z \wedge z < y) \wedge \varphi^*(y)] \\
(\varphi U \psi)^* &:= \exists y[x \leq y \wedge \psi^*(y) \wedge \forall z[x \leq z \wedge z < y \rightarrow \varphi^*(z)]]
\end{aligned}
$$

Linear Temporal Logic (LTL)

Theorem

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

Theorem

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

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

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 $\mathfrak{S}, s \vDash \varphi$ for CTL is **P-complete.** It can be done in

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

Data complexity: **NLOGSPACE-complete**

Theorem

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

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

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

Formula complexity: **PSPACE-complete**

Data complexity: **NLOGSPACE-complete**

Fixed points

Theorem (Knaster, Tarski)

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

$$\mathrm{lfp}(f) = \lim_{\alpha \to \infty} f^{\alpha}(\bot) \quad \text{and} \quad \mathrm{gfp}(f) = \lim_{\alpha \to \infty} f^{\alpha}(\top)$$

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

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

  $\mathrm{lfp}(f) = B$ and $\mathrm{gfp}(f) = (\mathbb{N} \setminus A) \cup B$
- $f(X) := \{ y \mid y \leq x \in X \}$

  fixed points: $\varnothing$, $\{0\}$, $\{0, 1\}, \dots, \{0, \dots, n\}, \dots, \mathbb{N}$
- $f(X) := \mathbb{N} \setminus X$ has no fixed points

Ordinals

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

3 Kinds

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

Proposition

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

Iteration

$$\begin{aligned} f^0(x) &:= x\,, \\ f^{\alpha+1}(x) &:= f(f^\alpha(x))\,, \\ f^\delta(x) &:= \sup_{\alpha<\delta} f^\alpha(x)\,, \quad \text{for limit ordinals } \delta\,. \end{aligned}$$

Proof

Monotonicity $f^{\alpha}(\bot) \leq f^{\beta}(\bot)$ for $\alpha \leq \beta$

$\bot \leq f(\bot)$

$f^{\alpha}(\bot) \leq f^{\beta}(\bot) \quad \Rightarrow \quad f^{\alpha+1}(\bot) \leq f^{\beta+1}(\bot)$

$f^{\alpha}(\bot) \leq f^{\delta}(\bot)$ for all $\alpha < \delta$

$\Rightarrow f^{\alpha+1}(\bot) \leq f^{\delta+1}(\bot)$

$\Rightarrow f^{\delta}(\bot) = \sup_{\alpha<\delta} f^{\alpha}(\bot) \leq f^{\delta+1}(\bot)$

$f^{\alpha}(\bot) \leq \sup_{\beta<\delta} f^{\beta}(\bot) = f^{\delta}(\bot)$

Existence exists α with $f^{\alpha}(\bot) = f^{\alpha+1}(\bot)$

Least fixed point

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

$\bot \leq a \quad \Rightarrow \quad f^{\alpha}(\bot) \leq f^{\alpha}(a) = a$

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

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

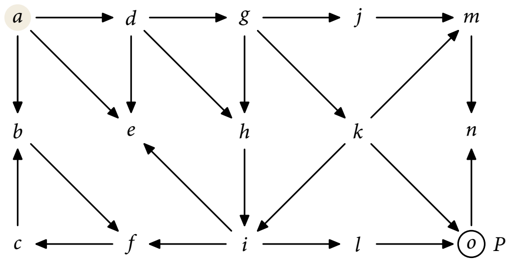
$$\mu X.\varphi(X): \quad X_0 := \varnothing, \quad X_{i+1} := F_\varphi(X_i)$$
$$\nu X.\varphi(X): \quad X_0 := S, \quad X_{i+1} := F_\varphi(X_i)$$

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

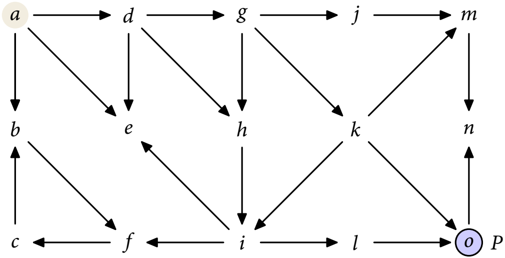
Examples

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



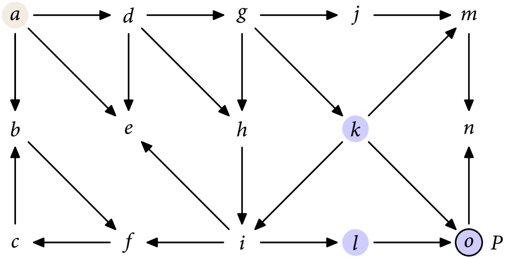
Examples

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



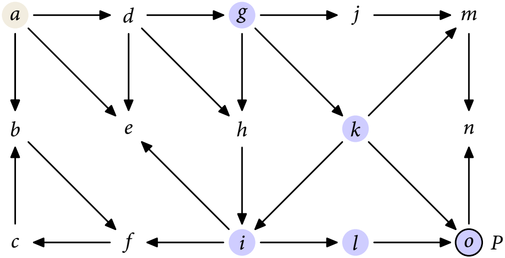
Examples

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



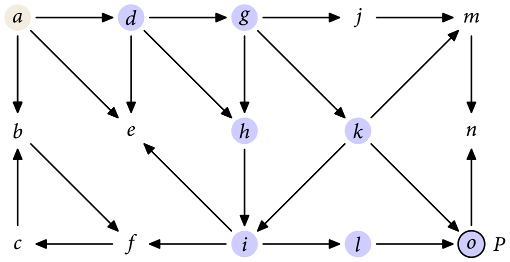
Examples

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



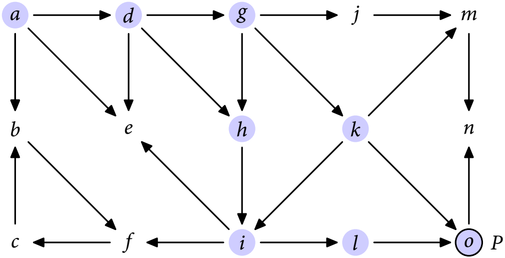
Examples

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



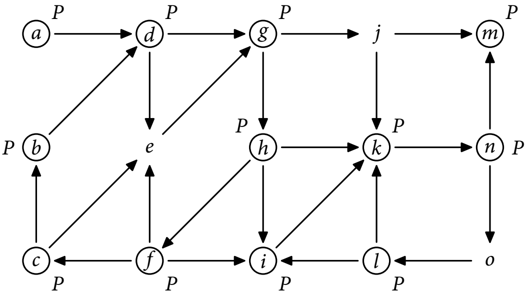
Examples

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



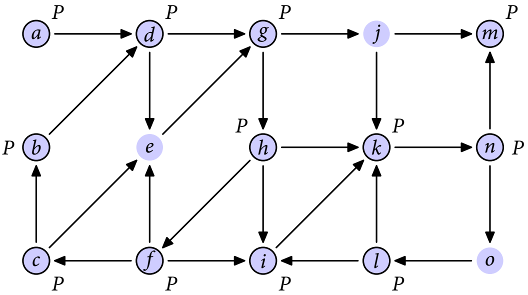
Examples

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



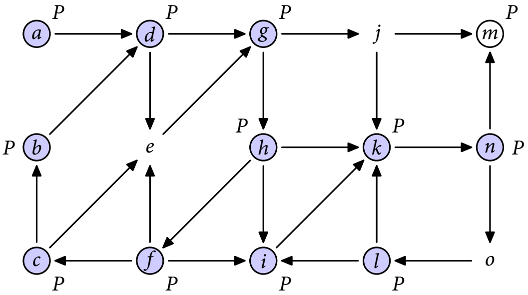
Examples

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



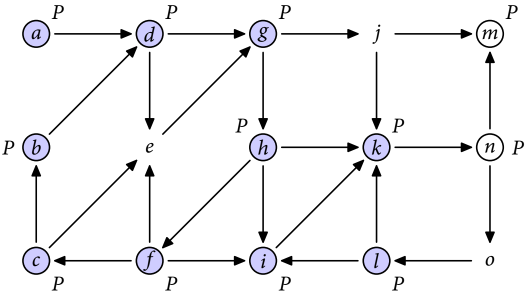
Examples

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



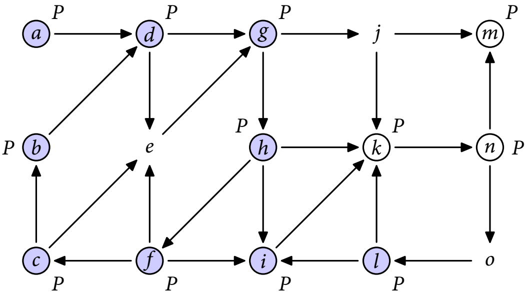
Examples

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



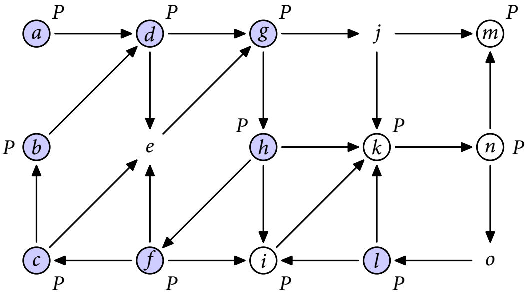
Examples

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



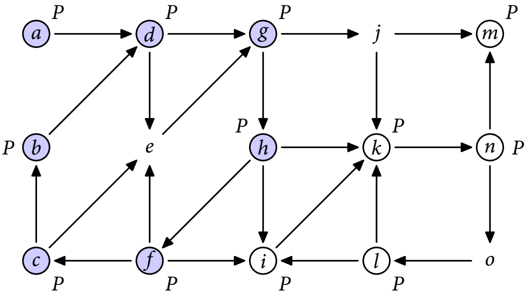
Examples

$\nu X(P \wedge \Diamond X)$ 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



Expressive power

Theorem

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

Proof (for CTL)

$$
\begin{aligned}
P^* &:= P \\
(\varphi \wedge \psi)^* &:= \varphi^* \wedge \psi^* \\
(\varphi \vee \psi)^* &:= \varphi^* \vee \psi^* \\
(\neg\varphi)^* &:= \neg\varphi^* \\
(EX\varphi)^* &:= \Diamond\varphi^* \\
(AX\varphi)^* &:= \Box\varphi^* \\
(E\varphi U\psi)^* &:= \mu X[\psi^* \vee (\varphi^* \wedge \Diamond X)] \\
(A\varphi U\psi)^* &:= \mu X[\psi^* \vee (\varphi^* \wedge \Box X)]
\end{aligned}
$$

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** $\mathcal{O}\big((|\varphi| \cdot |S|)^{|\varphi|}\big)$.

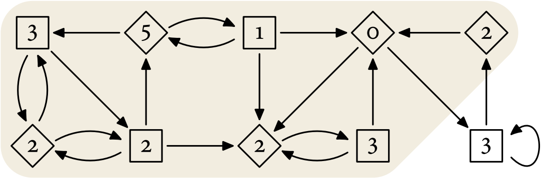
(The satisfiability algorithm uses tree automata and parity games.)

Parity Games

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

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

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

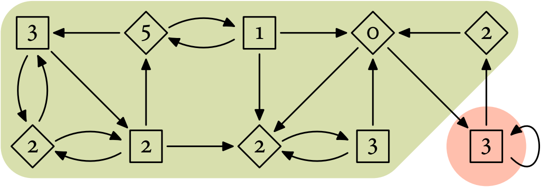


Parity Games

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

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

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



Parity Games

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

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

$\liminf_{n\to\infty} \Omega(v_n)$ 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 $\mathfrak{S}, s_0 \models \varphi$? ($\varphi$ μ-formula in negation normal form)

Positions

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

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

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

$$\psi = \psi_0 \wedge \psi_1 , \qquad \psi = P \text{ and } s \in P ,$$
$$\psi = [a]\psi_0 , \qquad \psi = \neg P \text{ and } s \notin P .$$

Initial position

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

Model-Checking Games

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

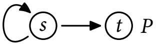
Edges (((s, ψ) means either $\langle s, \psi\rangle$ or $[s, \psi]$.)

$$\begin{aligned}
\langle s, \psi_0 \vee \psi_1 \rangle &\to (s, \psi_i)\,, \\
[s, \psi_0 \wedge \psi_1] &\to (s, \psi_i)\,, \\
\langle s, \mu X.\psi \rangle &\to \psi\,, \\
\langle s, \nu X.\psi \rangle &\to \psi\,, \\
\langle s, X \rangle &\to \langle s, \mu X.\psi \rangle \text{ or } \langle s, \nu X.\psi \rangle\,, \\
\langle s, \langle a \rangle \psi \rangle &\to \langle t, \psi \rangle \quad \text{for every } s \to^a t\,, \\
[s, [a]\psi] &\to \langle t, \psi \rangle \quad \text{for every } s \to^a t\,.
\end{aligned}$$

Priorities (all other priorities big)

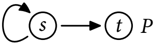
$$\begin{aligned}
\Omega(\langle s, \mu X.\psi \rangle) &:= 2k + 1\,, \qquad \text{if inside of } k \text{ fixed points.} \\
\Omega(\langle s, \nu X.\psi \rangle) &:= 2k\,.
\end{aligned}$$

Model-Checking Games

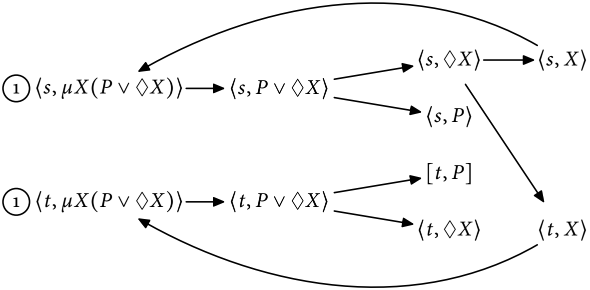
$\mathfrak{S} =$ 

$\varphi = \mu X(P \vee \Diamond X)$

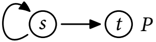
Model-Checking Games

$\mathfrak{S} =$ 

$\varphi = \mu X(P \vee \Diamond X)$



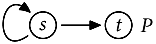
Model-Checking Games

$\mathfrak{S}$ = 

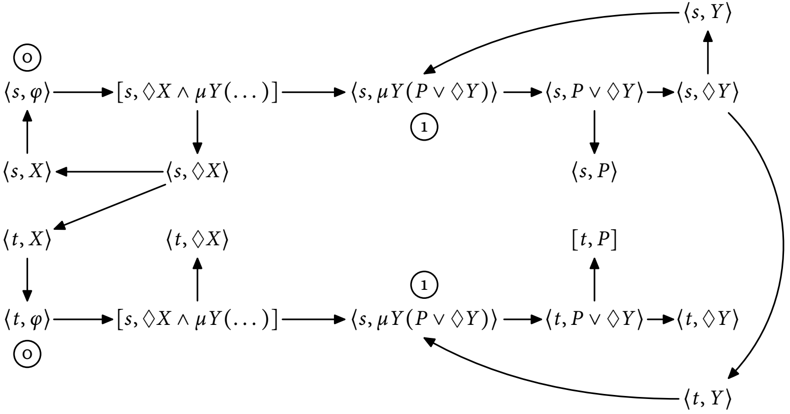
$$\varphi = \nu X(\Diamond X \land \mu Y(P \lor \Diamond Y))$$

Model-Checking Games

$\mathfrak{S}$ =



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



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

$$\begin{aligned}\text{man} &:= \text{person} \wedge \text{male}\\ \text{woman} &:= \text{person} \wedge \text{female}\\ \text{father} &:= \text{man} \wedge \exists \text{has\_child.person}\\ \text{mother} &:= \text{woman} \wedge \exists \text{has\_child.person}\end{aligned}$$

ABox

man(John)

man(Paul)

woman(Anna)

woman(Marry)

has_child(Anna, Paul)

is_married_to(Anna, John)

Syntax

Concepts

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

Terminology axioms

$$\varphi \sqsubseteq \psi \qquad \varphi \equiv \psi$$

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

Assertions

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

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))
)
```