CS Questions Around Formalisms

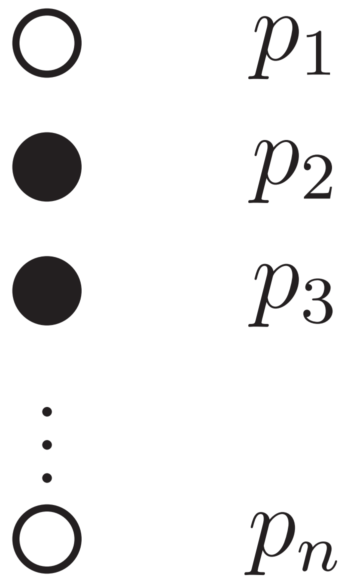
Does <u>my</u> program satisfy the spec?
Is there <u>any</u> program satisfying it?

} How difficult to check?

Can I express the property at all?

...and how complicated?

Propositional Logic



Syntax of Propositional Logic

The set $\mathsf{PL}[p_1, \ldots, p_n]$ of *propositional formulae over* $p_1, \ldots, p_n$ is freely generated as follows.

- $\top$, $\bot$, and all $p_i \in \{p_1, \ldots, p_n\}$ are propositional formulae *(so called "atomic formulae")*.
- If φ is a propositional formula, then so is $\neg\varphi$.
- If φ and ψ are propositional formulae, then so are $(\varphi \wedge \psi)$ and $(\varphi \vee \psi)$.

Evaluation of a Boolean Formula

For $\varphi \in \mathsf{PL}[p_1, \ldots, p_n]$ and $\underline{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$ we define $\varphi[\underline{a}] \in \{0,1\}$ as follows.

- $\top[\underline{a}] = 1$, $\bot[\underline{a}] = 0$, $p_i[\underline{a}] = a_i$
- $(\neg\varphi)[\underline{a}] = \neg\!\!\!\neg(\varphi[\underline{a}])$
- $(\varphi \wedge \psi)[\underline{a}] = (\varphi[\underline{a}]) \Cap (\psi[\underline{a}])$,
  $(\varphi \vee \psi)[\underline{a}] = (\varphi[\underline{a}]) \Cup (\psi[\underline{a}])$

The functions $\neg\!\!\!\neg, \Cap, \Cup$ are given by

x	y	$\neg\!\!\!\neg y$	$x \Cap y$	$x \Cup y$
0	0	1	0	0
0	1	0	0	1
1	0		0	1
1	1		1	1

Model Relation for Propositional Logic

Writing $\underline{a} \models \varphi$ for $\varphi[\underline{a}] = 1$ we obtain the following.

- $\underline{a} \models \top$ always holds and $\underline{a} \models \bot$ never holds
- $\underline{a} \models \neg\varphi$ holds iff $\underline{a} \models \varphi$ does *not* hold
- $\underline{a} \models \varphi \wedge \psi$ holds if $\underline{a} \models \varphi$ *and* $\underline{a} \models \psi$ both hold.
  $\underline{a} \models \varphi \vee \psi$ holds if $\underline{a} \models \varphi$ holds *or* $\underline{a} \models \psi$ holds.

Expressive Completeness

Theorem. For every $f\colon \{0,1\}^n \to \{0,1\}$ there is some $\varphi \in \mathsf{PL}[p_1, \ldots, p_n]$ such that for all $\underline{\mathrm{a}} \in \{0,1\}^n$ we have $f(\underline{\mathrm{a}}) = \varphi[\underline{\mathrm{a}}]$.

Expressive Completeness

Theorem. For every $f\colon \{0,1\}^n \to \{0,1\}$ there is some $\varphi \in \mathsf{PL}[p_1,\ldots,p_n]$ such that for all $\underline{\mathrm{a}} \in \{0,1\}^n$ we have $f(\underline{\mathrm{a}}) = \varphi[\underline{\mathrm{a}}]$.

In fact, φ can be chosen to be of the form

$$\bigvee_j \bigwedge_i \xi_{ij} \qquad \text{with } \xi_{ij} \in \{x_i, \neg x_i\}$$

(in "disjunctive normal form")

Other Complete Sets of Connectives

- $\wedge$, $\neg$. *Indeed,* $x \vee y = \neg((\neg x) \wedge (\neg y))$.
- $\vee$, $\neg$
- `nand` where

x	y	x `nand` y
0	0	1
0	1	1
1	0	1
1	1	0

Indeed, $\neg x = x$ `nand` x *and* $x \wedge y = \neg(x$ `nand` $y)$.

On Succinctness

Theorem. Let $\varepsilon > 0$. For large n, the fraction of functions

$$\{0,1\}^n \to \{0,1\}$$

that can be represented by formulae of size up to

$$2^{(1-\varepsilon)\cdot n}$$

tends to zero.

In other words, almost all function require exponentially large formulae.

Model Checking in Propositional Logic

Given: Propositional formula $\varphi \in \mathsf{PL}[p_1, \ldots, p_n]$ and $\underline{\mathrm{a}} \in \{0, 1\}^n$

Question: $\underline{\mathrm{a}} \models \varphi$?

Solvable in polynomial time (essentially $\mathcal{O}(|\varphi|)$): just compute truth value following the buildup of φ.

Satisfiability in Propositional Logic

Given: Propositional formula $\varphi \in \mathsf{PL}[p_1, \ldots, p_n]$.

Question: Is there some $\underline{\mathrm{a}} \in \{0,1\}^n$ such that $\underline{\mathrm{a}} \models \varphi$?

This problem is NP-complete.

NP

Definition. A problem is said to be in NP iff it can be solved by a non-deterministic Turing machine in polynomial time.

"NP *is verifying proofs*"

Conjecture. $\mathsf{P} \neq \mathsf{NP}$.

NP-completeness

Definition. A problem L is NP-complete iff

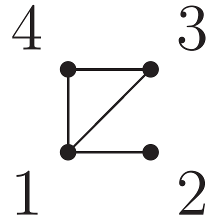
- it belongs to NP
- for any problem L' in NP there is an easy (say, in polynomial time) function f such that

$$x \in L' \text{ iff } f(x) \in L$$

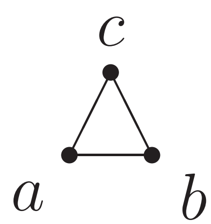
Example

Let $G = (\{1, 2, 3, 4\}, E)$ be an undirected graph. It can be described by formulae in $\mathsf{PL}[p_{12}, p_{13}, p_{14}, p_{23}, p_{24}, p_{34}]$ where p_{ij} expresses the fact that there is an edge from i to j.

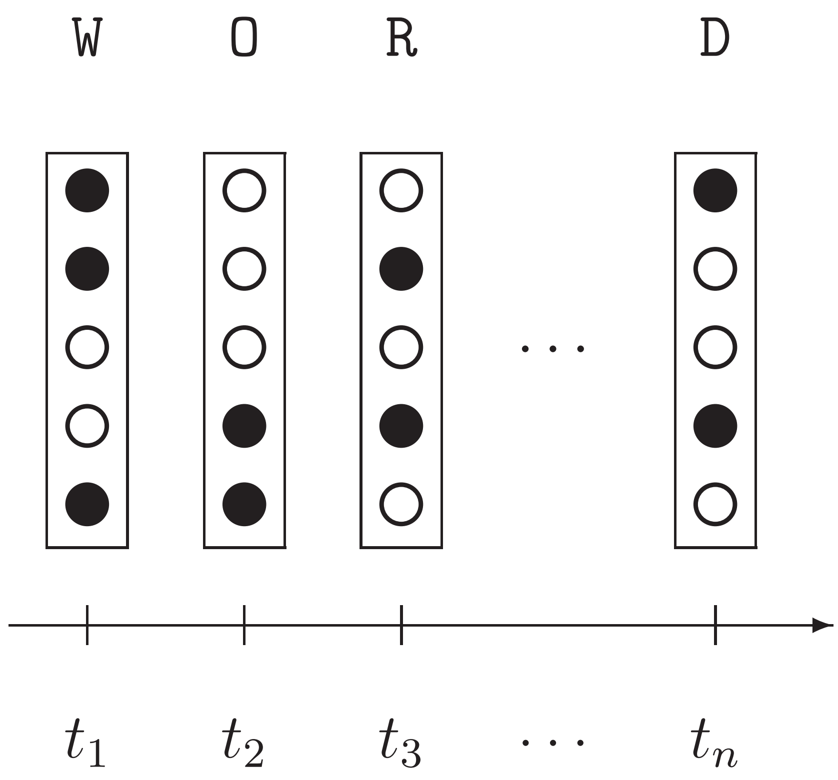
(a) Express: The graph is



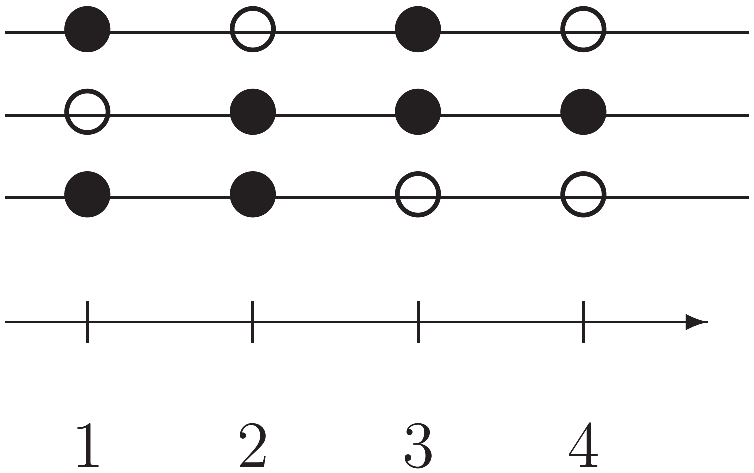
(b) Express: The graph contains some triangle.



First-Order Logic



Finite Trace Structures



$R = \{1, 3\}$

$Q = \{2, 3, 4\}$

$P = \{1, 2\}$

Universe $\{1, 2, 3, 4\}$

$$< \; = \; \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$$

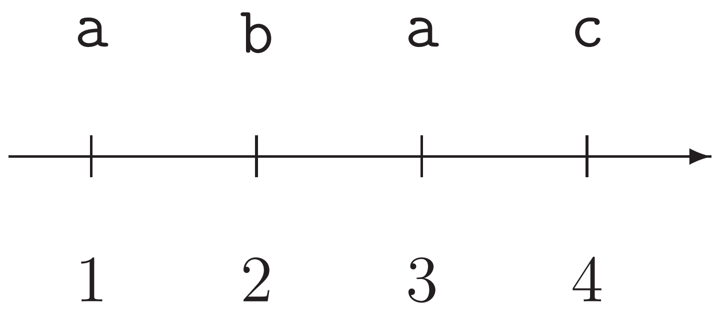
First-Order Structures

Definition. A first-order structure $\mathcal{A} = (A, R_1, R_2, \ldots)$ is given by

- A non-empty set A, called the "universe" of the structure
- Relations $R_1, R_2, \ldots$ on A.
  I.e., each $R_i \subseteq A^{n_i}$ for some n_i, called the arity.

Word Structures

$P_{\texttt{a}} = \{1,3\},\ P_{\texttt{b}} = \{2\},\ P_{\texttt{c}} = \{4\}$



Universe $\{1,2,3,4\}$

$< \; = \; \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$

Trace Structures and Word Structures

The universe is an initial segment of natural numbers, i.e., $A = \{1, 2, \ldots, \ell\}$ and $<$ the usual order on them.

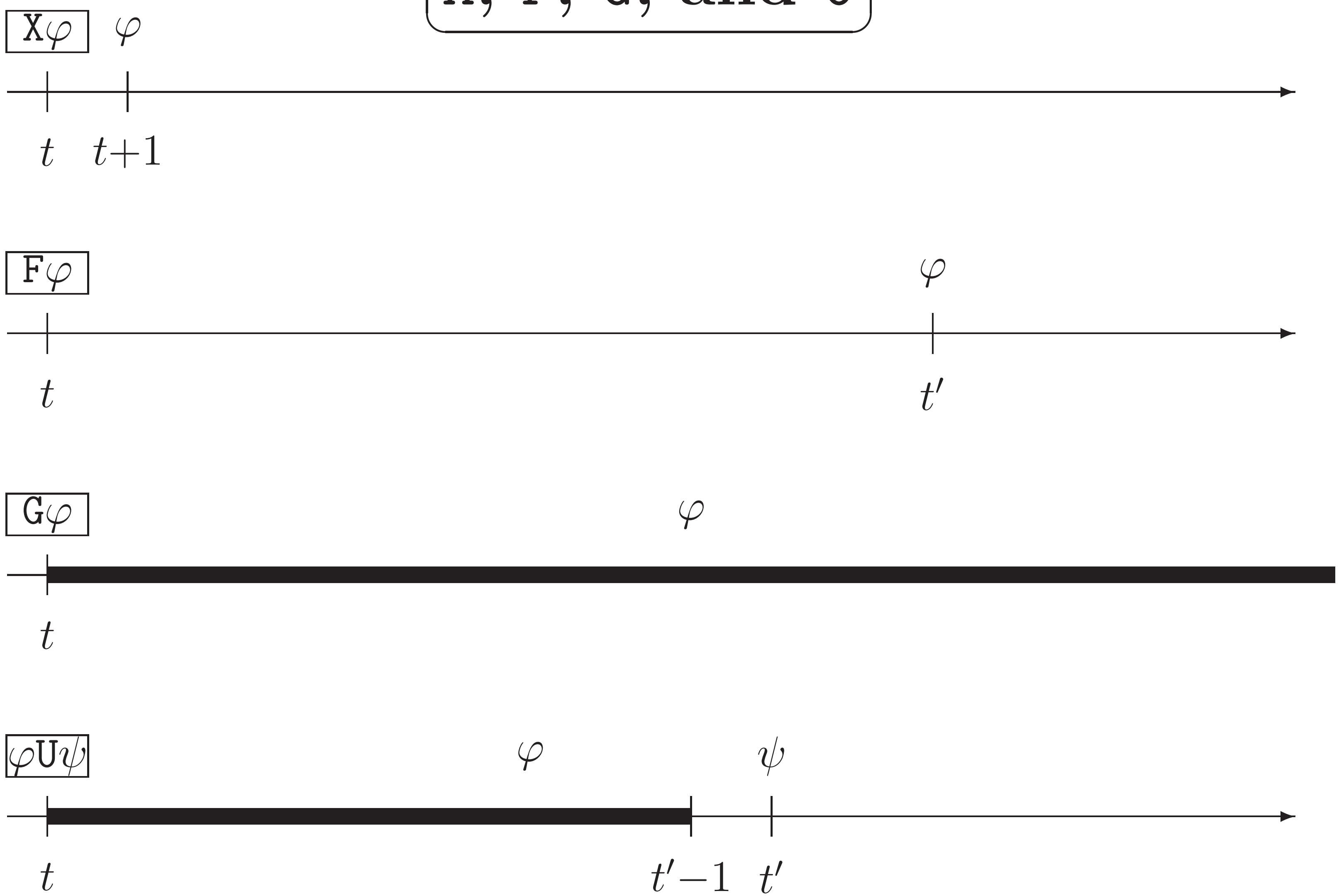
- finite trace structures are given by arbitrary unary relations P_1, P_2, ...
- word structures are given by unary relations P_a for $a \in \Sigma$ *partitioning* the univserse

Syntax of Linear Temporal Logic

The set $\mathsf{LTL}[p_1, \ldots, p_n]$ of *LTL-formulae* is freely generated

- $\top$, $\bot$, and all $p_i \in \{p_1, \ldots, p_n\}$
- If $\varphi, \psi \in \mathsf{LTL}[p_1, \ldots, p_n]$, then also $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$.
- ... and also
  - $\mathtt{X}\varphi$ "next"
  - $\mathtt{F}\varphi$ "finally"
  - $\mathtt{G}\varphi$ "globally"
  - $(\varphi \mathtt{U} \psi)$ "until"

X, F, G, and U



Semantics of LTL

Let $\mathcal{A} = (\{1, \ldots, \ell\}, <, P_1, P_2, \ldots, P_n)$ and $t \in \{1, \ldots, \ell\}$.
Define $\mathcal{A}, t \models \varphi$ for $\varphi \in \mathsf{LTL}[p_1, \ldots, p_2]$ inductively.

- $\mathcal{A}, t \models \top$; $\mathcal{A}, t \not\models \bot$; $\mathcal{A}, t \models p_i$ iff $t \in P_i$
- $\mathcal{A}, t \models \varphi \wedge \psi$ iff ...
- $\mathcal{A}, t \models \mathtt{X}\varphi$ iff $t < \ell$ and $\mathcal{A}, t{+}1 \models \varphi$
- $\mathcal{A}, t \models \mathtt{F}\varphi$ iff $\mathcal{A}, t' \models \varphi$ for some $t' \geq t$
- $\mathcal{A}, t \models \mathtt{G}\varphi$ iff $\mathcal{A}, t' \models \varphi$ for all $t' \geq t$
- $\mathcal{A}, t \models \varphi \mathtt{U} \psi$ iff there is some $t' \geq t$ s.t.
  $\mathcal{A}, t' \models \psi$ and $\mathcal{A}, t'' \models \varphi$ for all $t \leq t'' < t'$

Model Relation for Propositional Logic

Writing $\underline{\text{a}} \models \varphi$ for $\varphi[\underline{\text{a}}] = 1$ we obtain the following.

- $\underline{\text{a}} \models \top$ always holds and $\underline{\text{a}} \models \bot$ never holds
- $\underline{\text{a}} \models \neg\varphi$ holds iff $\underline{\text{a}} \models \varphi$ does *not* hold
- $\underline{\text{a}} \models \varphi \land \psi$ holds if $\underline{\text{a}} \models \varphi$ *and* $\underline{\text{a}} \models \psi$ both hold.
  $\underline{\text{a}} \models \varphi \lor \psi$ holds if $\underline{\text{a}} \models \varphi$ holds *or* $\underline{\text{a}} \models \psi$ holds.

LTL Model Checking

t		t		$t{+}1$
$\mathcal{A}, t \models \mathtt{X}\varphi$	iff			$\mathcal{A}, t{+}1 \models \varphi$
$\mathcal{A}, t \models \mathtt{F}\varphi$	iff	$\mathcal{A}, t \models \varphi$	or	$\mathcal{A}, t{+}1 \models \mathtt{F}\varphi$
$\mathcal{A}, t \models \mathtt{G}\varphi$	iff	$\mathcal{A}, t \models \varphi$	and	$\mathcal{A}, t{+}1 \models \mathtt{G}\varphi$
				in case $t + 1 \leq \ell$
$\mathcal{A}, t \models \varphi\mathtt{U}\psi$	iff	$\mathcal{A}, t \models \psi$	or	
		$\mathcal{A}, t \models \varphi$	and	$\mathcal{A}, t{+}1 \models \varphi\mathtt{U}\psi$

Example for LTL Model Checking

Formula $\mathtt{G}(y \rightarrow y\mathtt{U}r)$. *That this* $\mathtt{G}(\neg y \vee y\mathtt{U}r)$.

$\mathcal{A}$									
r	1	1	1	0	0	0	0	0	1
y	0	0	1	0	0	0	1	1	0
g	0	0	0	1	1	1	0	0	0
$\mathtt{G}(y \rightarrow y\mathtt{U}r)$									
$y \rightarrow y\mathtt{U}r$									
$y\mathtt{U}r$									
$\neg y$									

LTL and Automata

For φ construct DFA $\mathcal{M}_\varphi$ with $L(\mathcal{M}_\varphi) = \{w | w^{-1} \models \varphi\}$.

States: sets of sub-formulae of φ.

indicating with formulae hold at a given position

Transitions: Given

- previous state $\{\psi \mid \mathcal{A}, t{+}1 \models \psi\}$
- t'th letter of w, i.e., local properties of $\mathcal{A}$ at time t

determine $\{\psi \mid \mathcal{A}, t \models \psi\}$ by the rules seen.

Example

For each of the following formulae, decide whether they hold at the first letter of the given words!

- a and Xa

  `baab`   `abc`   `aaa`   `a`

- Fa

  `bbbbbba`   `ba`   `a`

- Ga

  `baaaaaaaaaaaaaa`   `aaa`   `a`   `aaaaaab`

- aUb

  `aaaaab`   `abc`   `bcaaab`   `b`

Example

Consider the language with the predicates r_1, r_2, g_1, g_2 with the interpretation that r_1 and r_2 express that process 1 and 2, respectively, are requesting access to a shared resource, and g_1 and g_2 express that access to the shared resource is granted for process 1 and 2.

Formalise the following statements.

- "No two requests are granted at the same time."
- "Every request will eventually be granted."
- "Every request by process 1 will be granted in the next round."

Example

Consider a traffic light. In our formalisation, we will use the variables r, y, g for the events the red/yellow/green light is on.

Formalise the following events.

- "There is always at least one light on"
- "It is always the case, that you will get a green light sometimes"
- "Whenever there's a yellow light, it will stay till a red light shows up"

First-Order Structures

Definition. A first-order structure $\mathcal{A} = (A, R_1, R_2, \ldots)$ is given by

- A non-empty set A, called the "universe" of the structure
- Relations $R_1, R_2, \ldots$ on A.
  I.e., each $R_i \subseteq A^{n_i}$ for some n_i, called the arity.

Syntax of First-Order Logic

The set $\mathsf{FOL}[R_1, \ldots, R_n]$ of first-order formulae over the relation symbols $R_1, \ldots, R_n$ is freely generated as follows.

- $(x = y)$ for variables x, y
- $R_j x_{i_1} \ldots x_{i_{n_j}}$ if R_j is of arity n_j
- $\top$, $\bot$, $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$ for formulae ϕ, ψ
- $\forall x \varphi$ and $\exists x \varphi$ for φ a formula and x a variable

Semantics of First-Order Logic

Let $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots)$ and $\eta\colon V \to A$.
Define $\mathcal{A}, \eta \models \varphi$ for $\varphi \in \mathsf{FOL}[R_1, \ldots]$ inductively.

- $\mathcal{A}, \eta \models x = y$ iff $\eta(x) = \eta(y)$
- $\mathcal{A}, \eta \models R_1 y_1 \ldots y_n$ iff $(\eta(y_1), \ldots, \eta(y_n)) \in R_i^{\mathcal{A}}$
- $\mathcal{A}, \eta \models \varphi \wedge \psi$ iff $\ldots$
- $\mathcal{A}, \eta \models \forall x \varphi$ iff for all $a \in A$ we have $\mathcal{A}, \eta_x^a \models \varphi$
- $\mathcal{A}, \eta \models \exists x \varphi$ iff for some $a \in A$ we have $\mathcal{A}, \eta_x^a \models \varphi$

LTL and First-Order Logic

$\varphi \in \mathsf{LTL}[p_1, \ldots, p_n]$	$\tilde{\varphi}(t) \in \mathsf{FOL}[<, P_1, \ldots, P_n]$
p_i	$P_i(t)$
$\mathtt{X}\varphi$	$\exists t' (\chi_{\text{next}}(t, t') \wedge \tilde{\varphi}(t'))$
$\mathtt{F}\varphi$	$\exists t' (t \leq t' \wedge \tilde{\varphi}(t'))$
$\mathtt{G}\varphi$	$\forall t' (t \leq t' \wedge \tilde{\varphi}(t'))$
$\varphi \mathtt{U} \psi$	$\exists t' (\; t \leq t' \wedge \tilde{\psi}(t') \wedge$
	$\forall t'' (((t \leq t'') \wedge (t'' < t')) \rightarrow \tilde{\varphi}(t)) \;)$

Negation Normal Form

Lemma.

$\mathcal{A}, \eta \models \neg\forall x\varphi$ iff $\mathcal{A}, \eta \models \exists x\neg\varphi$

$\mathcal{A}, \eta \models \neg\exists x\varphi$ iff $\mathcal{A}, \eta \models \forall x\neg\varphi$

Recall from propositional logic.

$\mathcal{A}, \eta \models \neg(\varphi \wedge \psi)$ iff $\mathcal{A}, \eta \models (\neg\varphi) \vee (\neg\psi)$

$\mathcal{A}, \eta \models \neg(\varphi \vee \psi)$ iff $\mathcal{A}, \eta \models (\neg\varphi) \wedge (\neg\psi)$

$\mathcal{A}, \eta \models \neg\neg\varphi$ iff $\mathcal{A}, \eta \models \varphi$

Prenex Normal Form

Lemma. Assume $x \notin \mathsf{fv}(\psi)$.

$\mathcal{A}, \eta \models (\forall x \varphi) \wedge \psi$ iff $\mathcal{A}, \eta \models \forall x (\varphi \wedge \psi)$

$\mathcal{A}, \eta \models (\forall x \varphi) \vee \psi$ iff $\mathcal{A}, \eta \models \forall x (\varphi \vee \psi)$

$\mathcal{A}, \eta \models (\exists x \varphi) \wedge \psi$ iff $\mathcal{A}, \eta \models \exists x (\varphi \wedge \psi)$

$\mathcal{A}, \eta \models (\exists x \varphi) \vee \psi$ iff $\mathcal{A}, \eta \models \exists x (\varphi \vee \psi)$

Words—Spot the difference!

a a b a a *vs* a a a a a

---

a a b a a *vs* a a a a b

---

a b a c a *vs* a a b c a

Ehrenfeucht-Fraïssé Games

The game is played on configurations

$$\underbrace{(A, R_1^{\mathcal{A}}, R_2^{\mathcal{A}}, \ldots)}_{\mathcal{A}}, a_1, \ldots, a_k \;\mid\; \underbrace{(B, R_1^{\mathcal{B}}, R_2^{\mathcal{B}}, \ldots)}_{\mathcal{B}}, b_1, \ldots, b_k$$

where $a_1, \ldots, a_k \in A$ and $b_1, \ldots, b_k \in B$.

In each round, Spoiler picks either $a_{k+1} \in A$ or $b_{k+1} \in B$. Then Duplicator picks the other.

Duplicator needs to keep the invariants

- $a_i = a_j$ iff $b_i = b_j$
- $R_i^{\mathcal{A}}(a_{i_1}, \ldots, a_{i_\ell})$ iff $R_i^{\mathcal{B}}(b_{i_1}, \ldots, b_{i_\ell})$

Quantifier-Rank

Definition.

$$\mathsf{qr}(\top) = \mathsf{qr}(\bot) = \mathsf{qr}(x = y) = \mathsf{qr}(Rx \ldots z) = 0$$

$$\mathsf{qr}(\neg\varphi) = \mathsf{qr}(\varphi)$$

$$\mathsf{qr}(\varphi \wedge \psi) = \mathsf{qr}(\varphi \vee \psi) = \max\{\mathsf{qr}(\varphi), \mathsf{qr}(\psi)\}$$

$$\mathsf{qr}(\forall x\varphi) = \mathsf{qr}(\exists x\varphi) = \mathsf{qr}(\varphi) + 1$$

$$\mathsf{FOL}_k[R_1, \ldots, R_n] = \{\varphi \in \mathsf{FOL}[R_1, \ldots, R_n] \mid \mathsf{qr}(\varphi) \leq k\}$$

$\mathcal{A}, \vec{a} \equiv_k \mathcal{B}\vec{b}$ iff $\mathcal{A}, \vec{a} \equiv_{\mathsf{FOL}_k[\ldots]} \mathcal{B}, \vec{b}$

Ehrenfeucht-Fraïssé Theorem

Theorem. For any configuration $\mathcal{A}, \vec{a} \mid \mathcal{B}, \vec{b}$ in an EF-game over finite structures, the following are equivalent.

- Duplicator can survive m more rounds.
- $\mathcal{A}, \vec{a} \equiv_m \mathcal{B}, \vec{b}$

Winning Condition for $\mathcal{A}, \vec{a}$

Define $\chi_{m,\mathcal{A},\vec{a}}$ with $\mathsf{qr}(\chi_{m,\mathcal{A},\vec{a}}) \leq m$ s.t. $\mathcal{B}, \vec{b} \models \chi_{m,\mathcal{A},\vec{a}}(\vec{x})$ iff Duplicator has a strategy for m round in $\mathcal{A}, \vec{a} \mid \mathcal{B}, \vec{b}$.

$$\chi_{m+1,\mathcal{A},\vec{a}}(\vec{x}) = \Big(\bigwedge_{a \in A} \exists y \chi_{m,\mathcal{A},\vec{a},a}\Big) \wedge \Big(\forall y \bigvee_{a \in A} \chi_{m,\mathcal{A},\vec{a},a}\Big)$$

$$\chi_{0,\mathcal{A},\vec{a}}(\vec{x}) = \bigwedge_{\substack{\mathcal{A}, \vec{a} \models \varphi \\ \varphi \text{ atomic}}} \varphi \quad \wedge \bigwedge_{\substack{\mathcal{A}, \vec{a} \not\models \varphi \\ \varphi \text{ atomic}}} \neg\varphi$$

Example

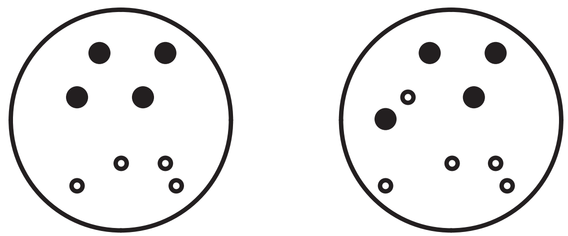
Let $\mathcal{A} = (\{1, 2, 3\}, <, P_a, P_b)$ be the structure for the word `aab`, i.e. $P_a = \{1, 2\}, P_b = \{3\}$.

- Write down $\chi_{1,\mathcal{A},1}$.
- Does $\texttt{aaab}, 1 \models \chi_{1,\mathcal{A},1}$?

Example: Unstructured Sets

If $\mathcal{A} = (A)$ and $\mathcal{B} = (B)$ are structures over the empty signature, then

$$\mathcal{A} \equiv_m \mathcal{B} \text{ iff } (|A| = |B| \text{ or } |A|, |B| \geq m).$$

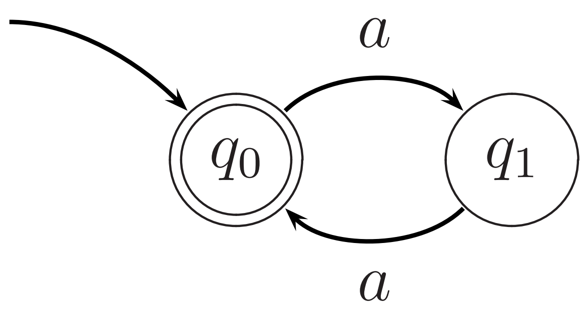


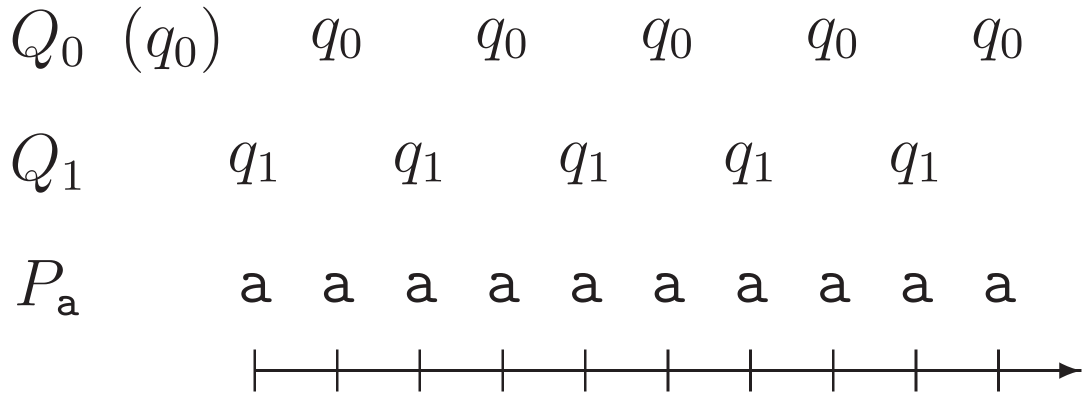
EF-Games over Linear Orders

Let $m \geq 1$ be a natural number, and $\mathcal{A} = (A, <^{\mathcal{A}})$ and $\mathcal{B} = (B, <^{\mathcal{B}})$ be linear orderings of lengths ℓ_A and ℓ_B, respectively.

Then $\mathcal{A} \equiv_m \mathcal{B}$ iff $(\ell_A = \ell_B$ or $\ell_A, \ell_B \geq 2^m - 1)$.

Parity of the Word Length





Syntax of Monadic Second-Order Logic

The set $\mathsf{MSO}[R_1, \dots, R_n]$ of first-order formulae over the relation symbols $R_1, \dots, R_n$ is freely generated as follows.

- $(x = y)$ for variables x, y
- $R_j x_{i_1} \dots x_{i_{n_j}}$ if R_j is of arity n_j
- Xx for X predicate variable
- $\top$, $\bot$, $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$ for formulae ϕ, ψ
- $\forall x \varphi$ and $\exists x \varphi$ for φ a formula and x a variable
- $\forall X \varphi$ and $\exists X \varphi$ for φ a fla and X predicate var

Semantics of Monadic Second-Order Logic

Let $\mathcal{A} = (A, R_1^{\mathcal{A}}, \dots)$ and $\eta \colon V \to A$, and $H \colon V^{(1)} \to \mathfrak{P}(A)$.
Define $\mathcal{A}, H, \eta \models \varphi$ for $\varphi \in \mathsf{MSO}[R_1, \dots]$ inductively.

- $\mathcal{A}, H, \eta \models x = y$ iff $\eta(x) = \eta(y)$
  $\mathcal{A}, H, \eta \models R_1 y_1 \dots y_n$ iff $(\eta(y_1), \dots, \eta(y_n)) \in R_i^{\mathcal{A}}$
  $\mathcal{A}, H, \eta \models Xx$ iff $\eta(x) \in H(X)$
- $\mathcal{A}, H, \eta \models \varphi \wedge \psi$ iff $\dots$
  $\mathcal{A}, H, \eta \models \forall x \varphi$ iff $\mathcal{A}, H, \eta_x^a \models \varphi$ for all $a \in A$
- $\mathcal{A}, H, \eta \models \forall X \varphi$ iff $\mathcal{A}, H_X^U, \eta \models \varphi$ for all $U \in \mathfrak{P}(A)$
  $\mathcal{A}, H, \eta \models \exists X \varphi$ iff $\mathcal{A}, H_X^U, \eta \models \varphi$ for some $U \in \mathfrak{P}(A)$

Representing Automata Runs in MSO

Theorem. Let $\mathcal{L} \subset \Sigma^+$ be regular. Then there is an $\mathsf{MSO}[<, P_a, \ldots]$-formula φ such that

$$w \in \mathcal{L} \text{ iff } w \models \varphi$$

Run of an Automaton

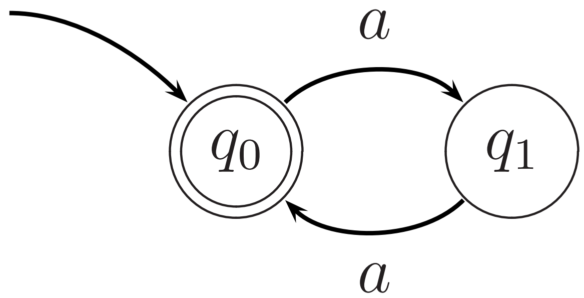
Let $\mathfrak{A} = (Q, I, \Delta, F)$ be an NFA, $Q = \{q_0, \ldots, q_n\}$.
"$\mathfrak{A}$ has an accepting run": $\exists X_0 \ldots \exists X_n (\varphi_i \wedge \varphi_s \wedge \varphi_f)$

$$\varphi_i \equiv \forall x(\text{“}x \text{ first”} \rightarrow \bigwedge_j (X_j(x) \rightarrow \bigvee_{q_i \in I, (q_i, a, q_j) \in \Delta} P_a(x)))$$

$$\varphi_s \equiv \forall x \forall y(\chi_{\mathsf{next}}(x, y) \rightarrow \bigwedge_j (X_j(y) \rightarrow \bigvee_{(q_i, a, q_j \in \Delta)} (X_i(x) \wedge P_a(y))))$$

$$\varphi_f \equiv \forall x(\text{“}x \text{ last”} \rightarrow \bigvee_{q_j \in F} X_j(x))$$

Example: $(aa)^*$



$$\exists X_0 \exists X_1 [\ \forall x((\forall y. x \leq y) \rightarrow (X_1(x) \wedge P_a(x)))\wedge$$
$$\forall x \forall y(((x < y) \wedge \neg \exists z(x < z \wedge z < y)) \rightarrow$$
$$((X_0(y) \rightarrow (X_1(x) \wedge P_a(y)))\wedge$$
$$(X_1(y) \rightarrow (X_0(x) \wedge P_a(y)))))\wedge$$
$$\forall x((\forall y. y \leq x) \rightarrow X_0(y))]$$

MSO Games

$$\mathcal{A}, U_1, \ldots, U_k, a_1, \ldots, a_\ell \mid \mathcal{B}, V_1, \ldots, V_k, b_1, \ldots, b_\ell$$

where $U_1, \ldots, U_k \subset A$, $V_1, \ldots, V_k \subset B$.

Spoiler can choose between two types of moves.

- choose $a_{\ell+1} \in A$, or $b_{\ell+1} \in B$
- choose $U_{k+1} \subset A$, or $V_{k+1} \subset B$

Duplicator needs to keep the invariants

- $a_i = a_j$ iff $b_i = b_j$; $R_i^{\mathcal{A}}(a_{i_1}, \ldots, a_{i_\ell})$ iff $R_i^{\mathcal{B}}(b_{i_1}, \ldots, b_{i_\ell})$
- $a_i \in U_j$ iff $b_i \in V_j$

MSO Games—The Theorem

For any configuration $\mathcal{A}, \vec{U}, \vec{a} \mid \mathcal{B}, \vec{V}, \vec{b}$ in an MSO-game over finite structures, the following are equivalent.

- Duplicator can survive m more rounds.
- $\mathcal{A}, \vec{U}, \vec{a} \equiv_m^{MSO} \mathcal{B}, \vec{V}, \vec{b}$

Winning Condition for $\mathcal{A}, \vec{U}, \vec{a}$

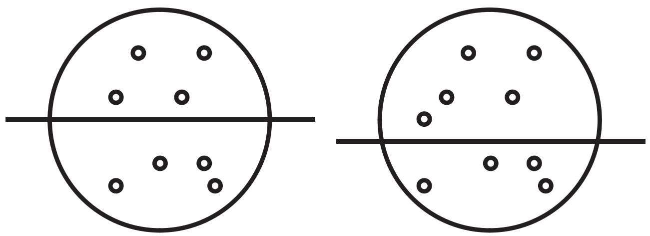
Define $\chi_{m,\mathcal{A},\vec{U},\vec{a}}$ with $\mathsf{qr}(\chi_{m,\mathcal{A},\vec{U},\vec{a}}) \leq m$ s.t.
$\mathcal{B}, \vec{V}, \vec{b} \models \chi_{m,\mathcal{A},\vec{U},\vec{a}}(\vec{x})$ iff Duplicator has a strategy for m rounds in $\mathcal{A}, \vec{U}, \vec{a} \mid \mathcal{B}, \vec{V}, \vec{b}$.

$$\begin{aligned}
\chi_{m+1,\mathcal{A},\vec{U},\vec{a}}(\vec{X}, \vec{x}) = \ & (\bigwedge_{a \in A} \exists y \chi_{m,\mathcal{A},\vec{U},\vec{a},a}) \wedge (\forall y \bigvee_{a \in A} \chi_{m,\mathcal{A},\vec{U},\vec{a},a}) \\
& \wedge (\bigwedge_{U \subset A} \exists Y \chi_{m,\mathcal{A},\vec{U},U,\vec{a}}) \wedge (\forall Y \bigvee_{U \subset A} \exists Y \chi_{m,\mathcal{A},\vec{U},U,\vec{a}}) \\
\chi_{0,\mathcal{A},\vec{U},\vec{a}}(\vec{x}) = \ & \bigwedge_{\substack{\mathcal{A}, \vec{a} \models \varphi \\ \varphi \text{ atomic}}} \varphi \quad \wedge \bigwedge_{\substack{\mathcal{A}, \vec{a} \not\models \varphi \\ \varphi \text{ atomic}}} \neg\varphi
\end{aligned}$$

Example: Unstructured Sets

If $\mathcal{A} = (A)$ and $\mathcal{B} = (B)$ are structures over the empty signature, then

$$\mathcal{A} \equiv_m^{MSO} \mathcal{B} \text{ iff } (|A| = |B| \text{ or } |A|, |B| \geq 2^{m-1}).$$



MSO-Definability and Regular Language

For $m \geq 0$ we note that

- there are only finitely many $\equiv_m^{MSO}$ classes $[\![\mathcal{A}_w]\!]$, and
- the $\equiv_m^{MSO}$ class of wu only depends on that of w and u.

So

$$
\begin{aligned}
Q &= \{[\![\mathcal{A}_w]\!] \mid w \text{ a word}\} \\
\delta\colon \quad Q \times \Sigma &\quad\longrightarrow\quad Q \\
([\![\mathcal{A}_w]\!], a) &\quad\longmapsto\quad [\![\mathcal{A}_{wa}]\!]
\end{aligned}
$$

defines a finite automaton.

Büchi's Theorem

The following are equivalent for word languages $\mathcal{L} \subset \Sigma^+$.

- $\mathcal{L}$ is regular
- $\mathcal{L}$ is MSO definable, i.e., there is an MSO formula φ and

$$\mathcal{L} = \{w \mid w \models \varphi\}$$

Alternative Proof

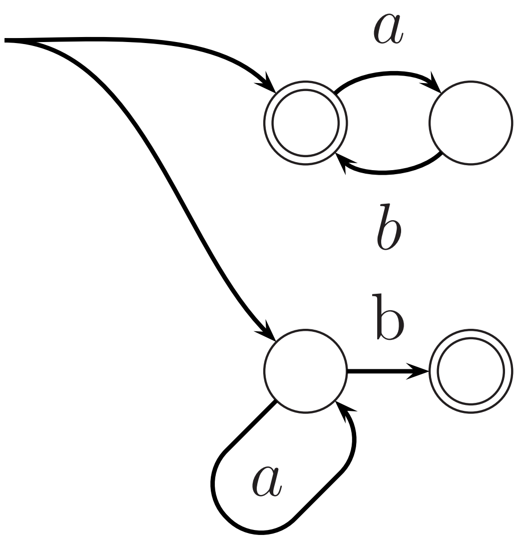
NFAs are closed under

- intersection $\mathcal{L} \cap \mathcal{L}'$
- union $\mathcal{L} \cup \mathcal{L}'$
- complement
  e.g., via power-set construction to get a DFA
- projection of the alphabet
  $\{\pi(a_1) \ldots \pi(a_n) \mid a_1 \ldots a_n \in \mathcal{L}\}$

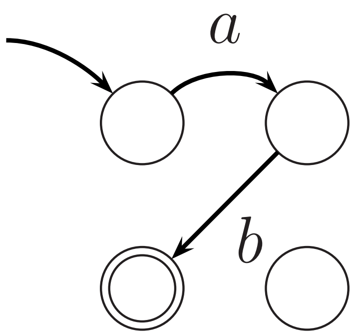
of their languages.

Moreover, this closure is effective.

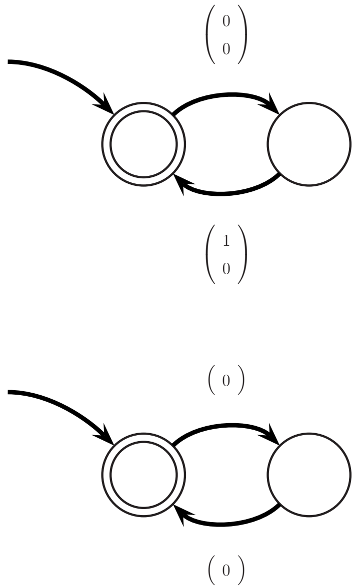
NFA Closure Properties



$$Q = Q_0 \cup Q_1$$

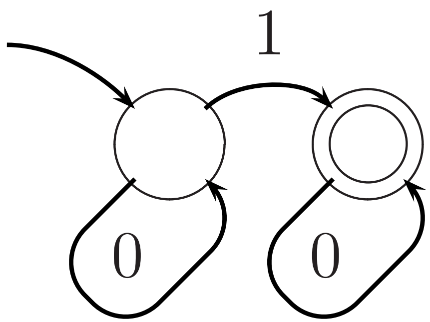


$$Q = Q_0 \times Q_1$$

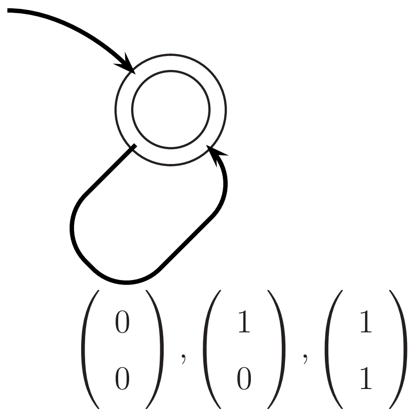


$$Q = Q_0$$

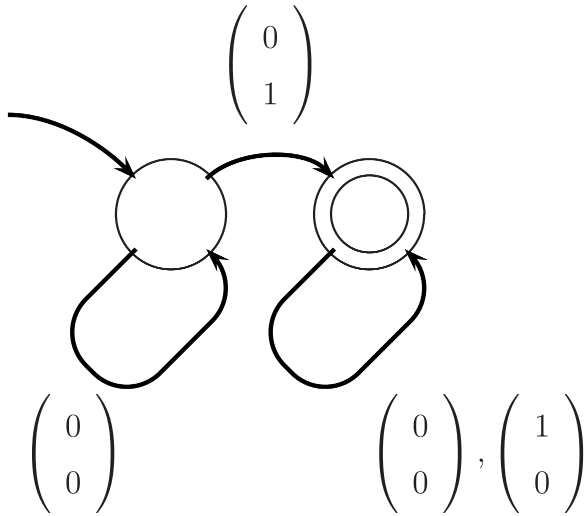
Primitives



"X is a singleton"



$X \subset Y$



"$x < y$"

Presburger Arithmetic

$X = 13$	1	0	1	1	0	0	0	0	0	0
$X = 30$	0	1	1	1	1	0	0	0	0	0
$Z = 43$	1	1	0	1	0	1	0	0	0	0