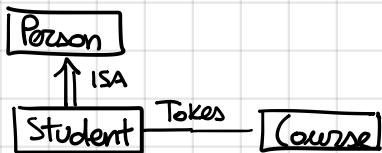


Formal Methods

FIRST ORDER LOGIC

FOL is the logic to speak about objects which are the domain of universe, concerned about properties and relationships. It also has functions including constants that denote objects.



$\text{Student}(x)$ = "x is a student"
 $\text{Person}(x)$
 $\text{Course}(x)$
 $\text{Takes}(x,y)$ = "x takes y"
 Nouns Arities

This is our kind of alphabet!

$$\begin{aligned}
 &\forall x \text{ St}(x) \rightarrow \text{Pe}(x) && [\text{ISA}] \\
 &\forall x (\exists y \text{ Takes}(x,y)) \rightarrow \text{St}(x) && [\text{First argument of "Takes" is a student}] \\
 &\quad -\alpha(x)- \\
 &\forall y (\exists y \text{ Takes}(x,y)) \rightarrow \text{Co}(y) && [\text{Second argument of "Takes" is a course}] \\
 &\quad \alpha(y) \\
 \Rightarrow &\forall x \forall y \text{ Takes}(x,y) \rightarrow \text{St}(x) \wedge \text{Co}(y), && \beta(x,y)
 \end{aligned}$$

Note: first 3 formulas are talking in general, not about any particular object (Student, Person or Course). Just Metadata → INTENSIONAL KNOWLEDGE

$\text{Student}(\text{John})$
 $\text{Takes}(\text{John}, \text{Math})$
 $\text{Person}(\text{Mozy})$

Γ (= gamma)
Constants → Data → EXTENSIONAL KNOWLEDGE

$\Gamma \models \text{Pe}(\text{John})$ ✓
 $\Gamma \models \text{Co}(\text{Math})$ ✓
 $\Gamma \models \text{St}(\text{Mozy})$ ✗
 $\Gamma \models \neg \text{St}(\text{Mozy})$ ✗

[Is John a person?]
} Because general rules can't lead to these two conclusions

$Vars = \{x_1, \dots, x_n\}$ set of individual variables. $Vars \subseteq \text{Terms}$ eg constants

FOL interpretation: $I = (\Delta^I, P_1^I, P_2^I, \dots, f_1^I, f_2^I, \dots)$ P_x^I = predicate f_x^I = formula
 Δ^I = domain, set of objects

Given I , an assignment is a function $\alpha: Vars \rightarrow \Delta^I$ that assigns to each variable $x \in Vars$ an object $\alpha(x) \in \Delta^I$ → $\hat{\alpha}: \text{Terms} \rightarrow \Delta^I$

- $\hat{\alpha}(x) = \alpha(x)$ if $x \in Vars$
- $\hat{\alpha}(f(t_1, \dots, t_k)) = f^I(\hat{\alpha}(t_1), \dots, \hat{\alpha}(t_k))$

$I, \alpha \models \varphi$: φ is true in interpretation I wrt an assignment α

Variable x in formula φ is free if x doesn't occur in scope of any quantifiers, bounded otherwise. φ is closed if has no free variable (sentence), open otherwise

A FOL query is an open formula. When φ is a query with free variables (x_1, \dots, x_k) then we write it as $\varphi(x_1, \dots, x_k)$ and say that φ has arity k .

Boolean query is a FOL query without free variables

LOGICAL TASKS

Validity: φ is valid iff $\forall I, d$ we have $I, d \models \varphi$

Satisfiability: φ is satisfiable iff $\exists I, d$ s.t. $I, d \models \varphi$, unsatisfiable otherwise

Logical Implication: φ logically implies ψ ($\varphi \models \psi$) iff $\forall I, d$ if $I, d \models \varphi$ then $I, d \models \psi$

Logical Equivalence: φ is logical equivalent to ψ iff $\forall I, d$ we have that $I, d \models \varphi$ iff $I, d \models \psi$ i.e. $\varphi \models \psi$ and $\psi \models \varphi$

QUERY EVALUATION

Checking if a formula is true given interpretation and assignment. To do so we have to consider finite alphabet (finite predicates, functions) and finite interpretation (Δ^I finite)

\Rightarrow Given query $\varphi(x_1, \dots, x_k)$, compute $\varphi^I = \{(d_1, \dots, d_k) \mid I, \langle d_1, \dots, d_k \rangle \models \varphi(x_1, \dots, x_k)\}$

Recognition problem: given finite I , a query and a tuple, check if the tuple is a solution.

Combined Complexity: complexity of $\{\langle I, d, \varphi \rangle \mid I, d \models \varphi\}$ i.e. I, d, φ all part of input

Data Complexity: All but φ (fixed) is part of input

Query Complexity: All but I (fixed) is part of input

Theorem: Complexity of $\{\langle I, d \rangle \mid I, d \models \varphi\}$ is

- TIME: polynomial

- SPACE: Log Space

This is the reason why FOL queries are used in databases!

Example

We consider FOL interpretations exactly as used in relational databases. This requires to drop functions except for constants. Moreover we assume that the interpretation of constants is the identity function, that is constants are interpreted as themselves. This allows us to drop also the interpretation of constants from our interpretations, which now have the form:

$$I = (\Delta^I, P_1^I, P_2^I, \dots, P_n^I).$$

Interpretation: I is as follows (also given in relational notation):

- $\Delta^I = \{john, paul, george, mick, ny, london, 0, 1, \dots, 100\}$
- $Person^I = \{(john, 30), (paul, 60), (george, 35), (mick, 35)\}$
- $Lives^I = \{(john, ny), (paul, ny), (george, london), (mick, london)\}$
- $Manages^I = \{(paul, john), (george, mick), (paul, mick)\}$

$Person^I$

name	age
john	30
paul	60
george	35
mick	35

$Lives^I$

name	city
john	ny
paul	ny
george	london
mick	london

$Manages^I$

boss	emp. name
paul	john
george	mick
paul	mick

FOL

DATABASE

Query: find name and age of persons who live in the same city as their boss.

$$\exists z, w. Person(z) \wedge Manages(z, w) \wedge Lives(z, w) \wedge Lives(w, Person(w))$$

"Find" or "Return" mean assign a value to open variable! → Query = open formula (as we said)

EXAMPLES

$S(s_id, name) \wedge B(b_id, color) \wedge R(s_id, b_id, date)$

① Find names of sailors who have reserved boat 103

$\exists x, z, w, t \ S(x, y) \wedge R(x, z, w) \wedge B(z, t) \wedge z = 103$

↳ this means we ONLY CARE of y i.e. "whatever x, z, w, t "

↳ needed to "force" z to be a boat id and not just a number

② find names of sailors who have reserved a red boat

$\exists sid, bid, d \ S(sid, n) \wedge R(sid, bid, d) \wedge B(bid, "red")$

③ Find colors of boat reserved by BOB

$\exists bid, sid, d \ B(bid, c) \wedge R(sid, bid, d) \wedge S(sid, "BOB")$

④ Find names of sailors who have reserved at least one boat

$\exists sid, bid, d, c \ S(sid, n) \wedge R(sid, bid, d) \wedge B(bid, c)$

⑤ Find names of sailors who reserved red and green

$\exists sid, bid, d, bid2, dz \ S(sid, n) \wedge R(sid, bid, d) \wedge B(bid, "RED") \wedge R(sid, bid2, dz) \wedge B(bid2, "GREEN")$

⑥ ... red OR green

$\exists sid, bid, d \ S(sid, n) \wedge R(sid, bid, d) \wedge (B(bid, "RED") \text{ OR } B(bid, "GREEN"))$

⑦ ... reserved AT LEAST 2 boats

$\exists sid, bid, bid2, ... \ S(sid, n) \wedge R(sid, bid, d) \wedge B(bid, c) \wedge R(sid, bid2, dz) \wedge B(bid2, cz) \wedge \neg(bid = bid2)$

⑧ ... not reserved red boats

$\exists x \ S(x, n) \wedge \neg \exists y, d \ R(x, y, d) \wedge B(y, "red")$

⑨ ... reserved ALL boat

$\exists x \ S(x, n) \wedge \forall y (\exists c \ B(y, c)) \rightarrow (\exists d \ R(x, y, d))$

⑩ ... all red boats

$\exists x \ S(x, n) \wedge \forall y \ B(y, "red") \rightarrow \exists d \ R(x, y, d)$

⑪ ... only red boats

$\exists x \ S(x, n) \wedge \forall y \ \exists d \ R(x, y, d) \rightarrow B(y, "red")$

⑫ ... exactly one boat

$\exists x \ S(x, n) \wedge (\exists y, d, c \ R(x, y, d) \wedge B(y, c)) \wedge (\forall y \forall d' (\exists d \ R(x, y, d)) \wedge (\exists d \ R(x, y, d) \rightarrow y = y'))$

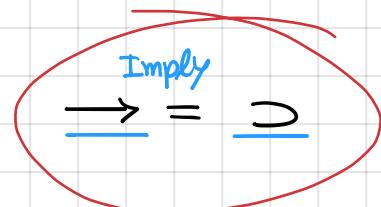


TABLEAU METHOD

Proving in mechanical manner that a given set of formulas is **not satisfiable**

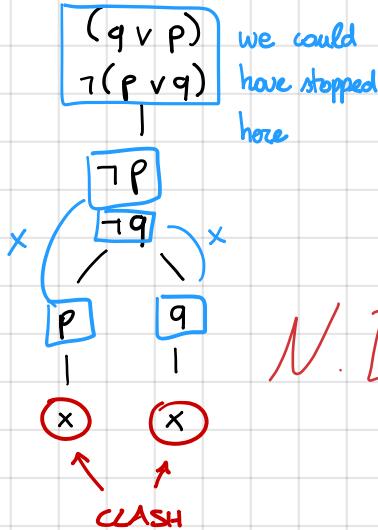
given: set of premises Γ and conclusion ϕ

task: prove $\Gamma \models \phi$

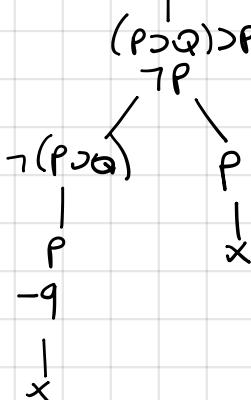
how: show $\Gamma \cup \{\neg\phi\}$ is not satisfiable i.e. add the complement of the conclusion to the premises and derive a contradiction

Th. $\Gamma \models \phi \iff \Gamma \cup \{\neg\phi\}$ is unsatisfiable

Eg. $\neg(q \vee p \supset p \vee q)$



Eg. $\models ((P \supset Q) \supset P) \supset P$
 $\neg(((P \supset Q) \supset P) \supset P)$



Note: order is inconsequential e.g. $q \vee p = p \vee q$ or $q \supset p$

Note: $\models \dots$ becomes $\neg\neg\dots$

α rules

$$\begin{array}{c} \frac{\phi \wedge \psi}{\phi} \\ \frac{\phi \wedge \psi}{\psi} \end{array} \quad \begin{array}{c} \frac{\neg(\phi \vee \psi)}{\neg\phi} \\ \frac{\neg(\phi \vee \psi)}{\neg\psi} \end{array} \quad \begin{array}{c} \frac{\neg(\phi \supset \psi)}{\phi} \\ \frac{\neg(\phi \supset \psi)}{\neg\psi} \end{array} \quad \begin{array}{c} \frac{\neg\neg\phi}{\phi} \end{array}$$

$\neg\neg$ -Elimination

β rules

$$\begin{array}{c} \frac{\phi \vee \psi}{\phi \mid \psi} \\ \frac{\phi \vee \psi}{\neg\phi \mid \neg\psi} \end{array} \quad \begin{array}{c} \frac{\neg(\phi \wedge \psi)}{\neg\phi \mid \neg\psi} \\ \frac{\neg(\phi \wedge \psi)}{\neg\phi \mid \psi} \end{array} \quad \begin{array}{c} \frac{\phi \supset \psi}{\neg\phi \mid \psi} \end{array}$$

Branch Closure

$$\frac{\phi}{\begin{array}{c} \neg\phi \\ X \end{array}}$$

Do α before β always

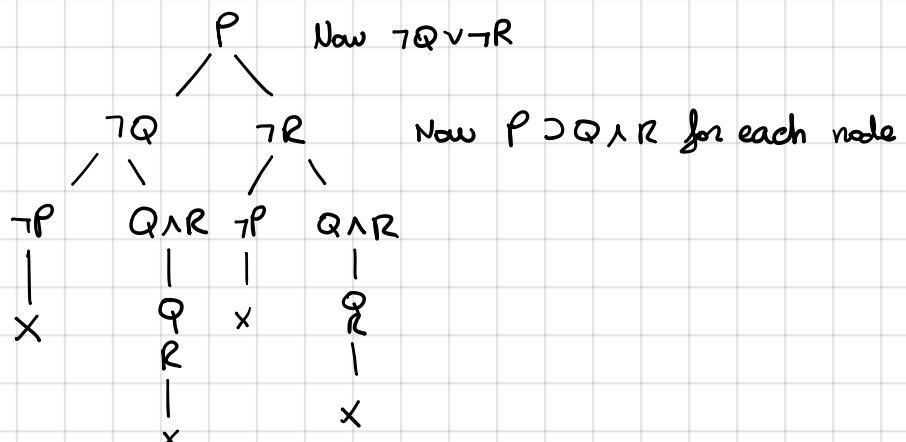
Note: These are the standard ("Smullyan-style") tableau rules.

We omit the rules for \equiv . We rewrite $\phi \equiv \psi$ as $(\phi \supset \psi) \wedge (\psi \supset \phi)$

Eg. $\models P \supset (Q \wedge R), \neg Q \vee \neg R \models \neg P$

$P \supset Q \wedge R \quad \neg Q \vee \neg R \quad \neg\neg P$

In a "node" with multiple formulas we compute one formula at a time (now $\neg\neg P$)



$\neg(P \vee Q \supset P \wedge Q)$ is satisfiable?

The Tableau shows us all possible interpretations $(\{P\}, \{Q\})$ that satisfy the formula.

e.g. $I = \{P\}$ means only P is true

Th. Soundness

If $\Gamma \vdash (\text{proves}) \Phi$ then $\Gamma \models \Phi$

Th. Completeness

If $\Gamma \models \Phi$ then $\Gamma \vdash \Phi$

yes, it is

Other rules for propositional logic (β rules)

$$\begin{array}{ll} \textcircled{1} \quad \Phi \equiv \Psi & \textcircled{2} \quad \neg(\Phi \equiv \Psi) \\ \frac{\Phi \quad \neg\Phi}{\Psi \quad \neg\Psi} & \frac{\Phi \quad \neg\Phi}{\neg\Psi \quad \Psi} \end{array}$$

δ rules:

$$\begin{array}{ll} \textcircled{1} \quad \exists x \Phi(x) & \textcircled{2} \quad \neg \forall x \Phi(x) \\ \frac{}{\Phi(c)} & \frac{}{\neg \Phi(c)} \end{array}$$

c = fresh constant = new constant not previously appearing in tableaux

δ rules:

$$\begin{array}{ll} \textcircled{1} \quad \neg \exists x \Phi(x) & \textcircled{2} \quad \forall x \Phi(x) \\ \frac{}{\neg \Phi(t)} & \frac{}{\Phi(t)} \end{array}$$

t = any term (not fresh)

This means that for every object of the domain the property $\Phi(x)$ should be true. A term t that occurs in the tableaux denotes an object of the domain.

Therefore Φ must be true for all terms t in the tableaux i.e. \forall can be applied as many times as one wants to any term.

Notation: $\Phi[x/t]$ = formula we get by substituting all x with t

e.g. $\forall x P(x,y)[x/b] = \forall x P(x,y)$

$P(x,y, f(x))[x/a] = P(a,y, f(a))$

$\exists x P(x,x) \wedge Q(x)[x/c] = \exists x P(x,x) \wedge Q(c)$

$\forall x P(x,y)[y/f(x)]$ = not allowed, x bounded to " \forall "

Example

To check if the formula

$(\exists x(P(x) \vee Q(x))) \equiv ((\exists x P(x)) \vee (\exists x Q(x)))$ is valid, we start with a tableaux with this formula:

valid

Tableaux only do UNSAT but...

$\neg((\exists x(Px \vee Qx)) \Leftrightarrow ((\exists x Px) \vee (\exists x Qx)))$	
	/ \
$\exists x(Px \vee Qx)$	
$\neg((\exists x Px) \vee (\exists x Qx))$	
	/ \
$\neg \exists x Px$	
$\neg \exists x Qx$	
$Pa \vee Qa$	
	/ \
Pa	Qa
$\neg Pa$	$\neg Qa$

We have clashes \rightarrow UNSAT

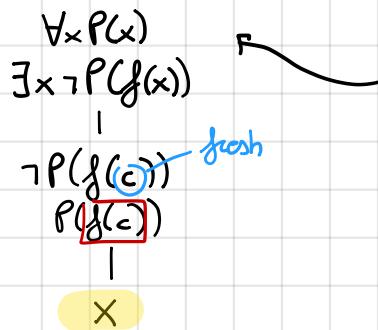
\Rightarrow original formula is valid

GENERAL RULE

• is Valid? \rightarrow negate and check if all branches are closed?

• is satisfiable? \rightarrow check if there is at least one open branch.

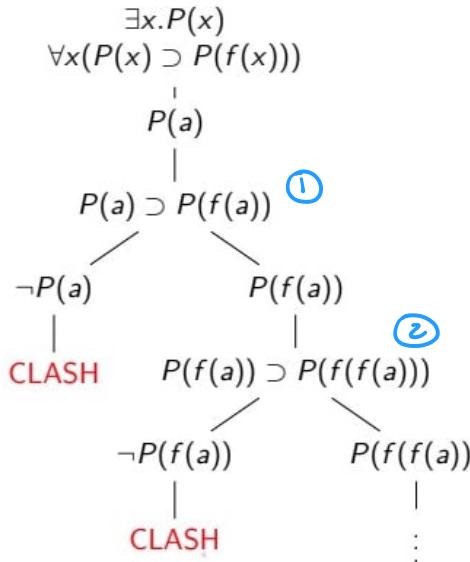
Ex. check if $\forall x P(x) \wedge \exists x \neg P(f(x))$ is SAT



Note: in cases like this, start with \exists and then use \forall to find clashes.

Eg. \forall applied multiple times

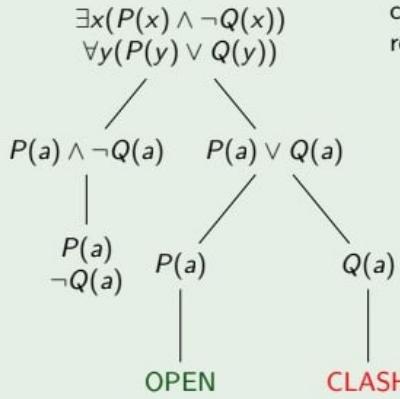
$$\exists x.P(x) \wedge \forall x(P(x) \supset P(f(x)))$$



Understand the model:

$$\exists x(P(x) \wedge \neg Q(x)) \wedge \forall y(P(y) \vee Q(y))$$

From the formulas appearing in the OPEN branch of the tableaux it is possible to construct a model for the root formula.



- $\Delta = \{a\}$, the constants appearing in the formulas
- $I(P) = \{a\}$, since the formula $P(a)$ appears in the open branch
- $I(Q) = \{\}$ since the formula $\neg Q(a)$ appears in the open branch

CONJUNCTIVE QUERIES

A query of form $\exists \vec{y} \text{ conj}(\vec{x}, \vec{y})$ where "conj" is a conjunction ("and") of atoms and EQUALITIES over free variable \vec{x} , the existentially quantified variable \vec{y} and possibly constants.

Relational alphabet:

`Person(name, age), Lives(person, city), Manages(boss, employee)`

Query: find the name and the age of the persons who live in the same city as their boss.

$$\begin{array}{l}
 \exists b, e, p_1, c_1, p_2, c_2. \text{Person}(n, a) \wedge \text{Manages}(b, e) \wedge \text{Lives}(p_1, c_1) \wedge \text{Lives}(p_2, c_2) \wedge \\
 n = p_1 \wedge n = e \wedge b = p_2 \wedge c_1 = c_2
 \end{array}
 \Rightarrow$$

Or simpler: $\exists b, c. \text{Person}(n, a) \wedge \text{Manages}(b, n) \wedge \text{Lives}(n, c) \wedge \text{Lives}(b, c)$

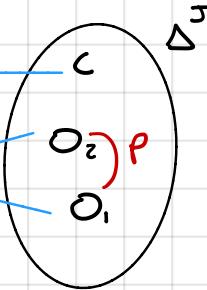
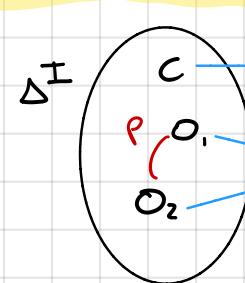
HOMOMORPHISM

Suppose two interpretations \mathcal{I} and \mathcal{J} on some alphabet.

A homomorphism from \mathcal{I} to \mathcal{J} is a mapping (function) $h: \Delta^{\mathcal{I}} \rightarrow \Delta^{\mathcal{J}}$ such that:

- $h(c^{\mathcal{I}}) = c^{\mathcal{J}}$

- $(o_1, \dots, o_k) \in P^{\mathcal{I}}$ implies $(h(o_1), \dots, h(o_k)) \in P^{\mathcal{J}}$



If the map covers the entire domain then it is called isomorphism. FOL is unable to distinguish between interpretations that are isomorphic.

Eg $\Delta^{\mathcal{I}} = \{v_1, v_2, v_3, v_4, v_5\}$

$$e^{\mathcal{I}} = \{(v_1, v_2)$$

$$(v_2, v_3)$$

$$(v_2, v_4)$$

$$(v_4, v_3)$$

$$(v_4, v_5)\}$$

$$\Delta^{\mathcal{J}} = \{q_1, q_2, q_3\}$$

$$e^{\mathcal{J}} = \{(q_1, q_2)$$

$$(q_2, q_3)$$

$$(q_3, q_3)$$

$$a^{\mathcal{J}} = q_1$$

$$b^{\mathcal{J}} = q_3$$

$$a^{\mathcal{I}} = v_1$$

$$b^{\mathcal{I}} = v_3$$

We want $h(v_1), h(v_2)$ etc

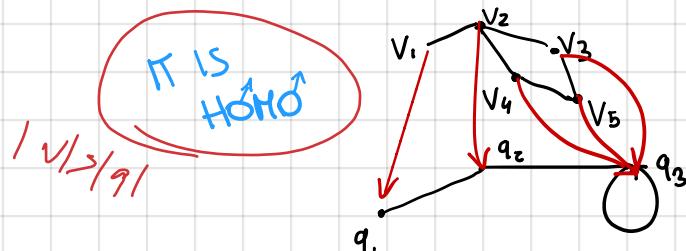
$$h(v_1) = q_1 \quad \text{because } h(c^{\mathcal{I}}) = c^{\mathcal{J}} \rightarrow h(a^{\mathcal{I}}) = a^{\mathcal{J}} \rightarrow h(v_1) = q_1 \quad (\text{some for } b^{\mathcal{I}}/b^{\mathcal{J}})$$

$$h(v_2) = q_2 \quad \text{because } (v_1, v_2) \in e^{\mathcal{I}} \rightarrow (h(v_1), h(v_2)) \in e^{\mathcal{J}} \text{ i.e we check } e^{\mathcal{J}} \text{ and search } q_2 \text{ in first position}$$

$$h(v_3) = q_3$$

$$h(v_4) = q_3 \quad \text{again } q_3 \text{ like above}$$

$$h(v_5) = q_3 \quad "$$



CANONICAL INTERPRETATION

(given a (boolean) CQ eg $q(c) \leftarrow E(c,y).E(y,z).E(z,c)$)

equal to $\exists y z \ E(c,y) \wedge E(y,z) \wedge E(z,c)$

can be transformed into a canonical interpretation

Def.: The canonical interpretation \mathcal{I}_q associated with q is the interpretation $\mathcal{I}_q = (\Delta^{\mathcal{I}_q}, P^{\mathcal{I}_q}, \dots, c^{\mathcal{I}_q}, \dots)$, where

- $\Delta^{\mathcal{I}_q} = \{x_1, \dots, x_n\} \cup \{c \mid c \text{ constant occurring in } q\}$,

i.e., all the variables and constants in q ;

- $c^{\mathcal{I}_q} = c$, for each constant c in q ;

- $(t_1, \dots, t_k) \in P^{\mathcal{I}_q}$ iff the atom $P(t_1, \dots, t_k)$ occurs in q .

Th. For boolean CQs, $\mathcal{I} \models q$ iff there exists a homomorphism from \mathcal{I}_q to \mathcal{I}

Observations:

① $\mathcal{I}_q \models q$ always true.

② If h - homomorphism from \mathcal{I} to \mathcal{I}_2 and h' from \mathcal{I}_2 to \mathcal{I}_3 then $h \circ h'$ is an homomorphism from \mathcal{I} to \mathcal{I}_3

Two interpretations I and J are **homomorphically equivalent** if exists homomorphism $h_{I,J}$ from I to J and $h_{J,I}$ from J to I .

CQs are unable to distinguish between interpretations that are homomorphic equivalent.

QUERY CONTAINMENT: given two FOL queries φ on ψ of some arity, φ is contained in ψ , denoted $\varphi \subseteq \psi$ if interpretations I and assignments α we have that $I, \alpha \models \varphi$ implies $I, \alpha \models \psi$.

For FOL queries, query containment is undecidable

For CQs, $q_1(\vec{x}) \subseteq q_2(\vec{x})$ can be reduced to query evaluation (NP complete)

How:

① Freeze the free variables i.e. consider them as constants

$\Rightarrow I, \vec{z} \models q_1(\vec{z})$ implies $I, \vec{z} \models q_2(\vec{z})$ $\forall I, \vec{z}$ where \vec{z} are new constants and I, \vec{z} extends I to new constants with $C^{I, \vec{z}} = \alpha(x)$

② Construct canonical interpretation $I_{q_1}(\vec{z})$ of CQ $q_1(\vec{z})$

③ Evaluate on $I_{q_1}(\vec{z})$ the CQ $q_2(\vec{z})$ i.e. check if $I_{q_1}(\vec{z}) \models q_2(\vec{z})$

$$\boxed{\begin{array}{l} \forall \vec{x} \ q_1(\vec{x}) \rightarrow q_2(\vec{x}) \\ \text{VALID} \end{array}}$$

Th. For CQs $q_1(\vec{x}) \subseteq q_2(\vec{x})$ iff $I_{q_1}(\vec{z}) \models q_2(\vec{z})$ where \vec{z} are new constants

Th. For CQs $I \models q$ iff $q_I \subseteq q$

Th. CQs containment is NP-complete

UNION OF CQ (UCQs)

UCQ has the form $\bigvee_{i=1 \dots n} \exists \vec{y}_i \text{ conj}_i(\vec{x}, \vec{y}_i)$ i.e. on OR of conjunctive queries.

As a normal "v", $I, \alpha \models \bigvee (\dots)$ iff $I, \alpha \models \exists y_i \text{ conj}_i(\vec{x}, \vec{y}_i)$ for some i i.e. at least one y_i satisfying

Also for UCQs we can have containment: $\{q_1 \dots q_n\} \subseteq \{q'_1 \dots q'_n\}$ iff for each q_i there is a q'_j s.t. $q_i \subseteq q'_j$

QUERY ANSWERING WITH INCOMPLETE INFORMATION

If we don't have complete information we can do $D \vdash Q$ (evaluation)

$\Rightarrow \forall I, I \models D$ implies $I \models Q$ (logical implication) i.e. for each model we must check if Q is true in it too

A common form in which D can be expressed is **INCOMPLETE DATABASE / NAEV TABLES** in which we can also find **labelled nulls** (unknown values)

Semantics of incomplete databases:

- A valuation function for nulls is a assignment function $\sigma : \text{Nulls} \rightarrow \text{Const}$ (essentially nulls are considered as individual variables in logic).
- We denote by $I, \sigma \models D$ the fact that for every tuple $(t_1, \dots, t_n) \in P$ for each table P we have $I, \sigma \models P(t_1, \dots, t_n)$.
- We define in logic the set of databases completing D as

$$\text{Models}(D) = \{I \mid \text{there exists a } \sigma \text{ such that } I, \sigma \models D\}$$

i.e. different values of null SQL can't use them!

Example

Employee		Manager		Employee		Manager			
①		mgr	mgd	name	Smith	mgr	mgd		
		Smith	null ₁	name	Smith	Smith	null ₁		
		null ₁	Brown	name	null ₁	Brown	Brown		
		Brown	Black	name	Brown	null ₂	Brown		
		Black		name	Black				
				name					
				name					
				name					
				name					

Answering a query in incomplete databases means computing the certain answer denoted as $\text{cert}(q, D)$, the set of tuples \vec{z} of constants of Const s.t. $\vec{z} \in q^I$ for every model I of D

- if q is boolean and D incomplete : $D \models q$ iff q evaluates to true in every model I of D ($D \not\models q$ otherwise)
- $D \not\models q$ is totally different from $D \models \neg q$

How to use CQs in incomplete DB?

- Each tuple in a table of D becomes an atom in conjunctive query q_D
- Each labelled null occurring in D become an existentially quantified variable in q_D

Th. $D \models q$ iff $q_D \subseteq q$ with $q = \text{boolean}(\cup)$ of conjunctive queries.

Th. $D \models q$ iff $I_{q_D} \models q$ → nulls become constants ($x_1, x_2 \dots$)

Example

$E(\text{mployee})$	$M(\text{anager})$	
name	mgr	mgd
Smith	Smith	Brown
$null_1$	$null_1$	
Brown	Brown	$null_2$

- Queries:

$$q_1(x, y) \leftarrow M(x, y)$$

$$q_2(x) \leftarrow \exists y. M(x, y)$$

$$q_3(x) \leftarrow \exists y_1, y_2, y_3. M(x, y_1) \wedge M(y_1, y_2) \wedge M(y_2, y_3)$$

$$q_4(x, y_3) \leftarrow \exists y_1, y_2. M(x, y_1) \wedge M(y_1, y_2) \wedge M(y_2, y_3)$$

- Answers:

$$q_1: \{ \}$$

$$q_2: \{ \text{Smith, Brown} \}$$

$$q_3: \{ \text{Smith} \}$$

$$q_4: \{ \}$$

For non boolean (U)CQs :

- evaluate q in D as it was a complete database
- filter all answers where nulls appears and remove them

UML CLASS DIAGRAM IN FOL

Let's start with an exercise:

→ The specification of this domain can be done in UML

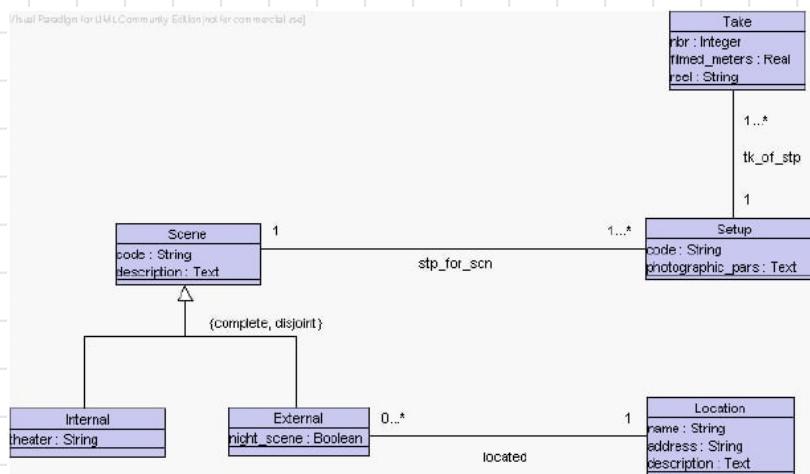
Requirements: We are interested in building a software application to manage filmed scenes for realizing a movie, by following the so-called "Hollywood Approach".

Every scene is identified by a code (a string) and it is described by a text in natural language.

Every scene is filmed from different positions (at least one), each of this is called a setup. Every setup is characterized by a code (a string) and a text in natural language where the photographic parameters are noted (e.g., aperture, exposure, focal length, filters, etc.). Note that a setup is related to a single scene.

For every setup, several takes may be filmed (at least one). Every take is characterized by a (positive) natural number, a real number representing the number of meters of film that have been used for shooting the take, and the code (a string) of the reel where the film is stored. Note that a take is associated to a single setup.

Scenes are divided into internals that are filmed in a theater, and externals that are filmed in a location and can either be "day scene" or "night scene". Locations are characterized by a code (a string) and the address of the location, and a text describing them in natural language.



All good but: not precise, verification done by humans, machine incomprehensible etc.

⇒ Use logic

Alphabet: $Scene(x)$, $Setup(x)$, $Take(x)$, $Internal(x)$, $External(x)$, $Location(x)$, $stp_for_scn(x, y)$, $tk_of_stp(x, y)$, $located(x, y)$, . . .

Axioms:

$$\forall x, y. code_{Scene}(x, y) \supseteq Scene(x) \wedge String(y)$$

$$\forall x, y. description(x, y) \supseteq Scene(x) \wedge Text(y)$$

$$\forall x, y. code_{Setup}(x, y) \supseteq Setup(x) \wedge String(y)$$

$$\forall x, y. photographic_pars(x, y) \supseteq Setup(x) \wedge Text(y)$$

$$\forall x, y. nbr(x, y) \supseteq Take(x) \wedge Integer(y)$$

$$\forall x, y. filmed_meters(x, y) \supseteq Take(x) \wedge Real(y)$$

$$\forall x, y. reel(x, y) \supseteq Take(x) \wedge String(y)$$

$$\forall x, y. theater(x, y) \supseteq Internal(x) \wedge String(y)$$

$$\forall x, y. night_scene(x, y) \supseteq External(x) \wedge Boolean(y)$$

$$\forall x, y. name(x, y) \supseteq Location(x) \wedge String(y)$$

$$\forall x, y. address(x, y) \supseteq Location(x) \wedge String(y)$$

$$\forall x, y. description(x, y) \supseteq Location(x) \wedge Text(y)$$

$$\forall x. Scene(x) \supseteq (1 \leq \#\{y \mid code_{Scene}(x, y)\} \leq 1)$$

$$\forall x, y. stp_for_scn(x, y) \supseteq Setup(x) \wedge Scene(y)$$

$$\forall x, y. tk_of_stp(x, y) \supseteq Take(x) \wedge Setup(y)$$

$$\forall x, y. located(x, y) \supseteq External(x) \wedge Location(y)$$

$$\forall x. Setup(x) \supseteq 1 \leq \#\{y \mid stp_for_scn(x, y)\} \leq 1$$

$$\forall y. Scene(y) \supseteq 1 \leq \#\{x \mid stp_for_scn(x, y)\}$$

$$\forall x. Take(x) \supseteq 1 \leq \#\{y \mid tk_of_stp(x, y)\} \leq 1$$

$$\forall x. Setup(y) \supseteq 1 \leq \#\{x \mid tk_of_stp(x, y)\}$$

$$\forall x. External(x) \supseteq 1 \leq \#\{y \mid located(x, y)\} \leq 1$$

$$\forall x. Internal(x) \supseteq Scene(x)$$

$$\forall x. External(x) \supseteq Scene(x)$$

$$\forall x. Internal(x) \supseteq \neg External(x)$$

$$\forall x. Scene(x) \supseteq Internal(x) \vee External(x)$$

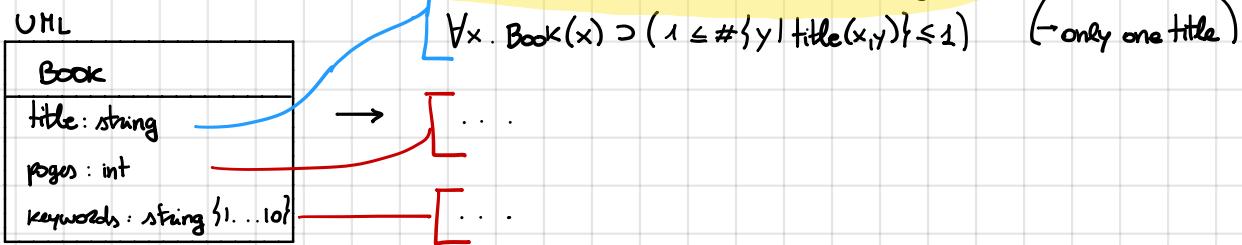
On the contrary: hard for humans, difficult to generate



- Assign formal semantics to constructs of the conceptual design diagram
- Use it as usual
- Read diagrams as logical theories

LOOK AT SLIDES FOR UML RECAP

ATTRIBUTES formalization



Each attribute is described using two FOL formulas

Second one (multiplicity) is a short hand for:

- At least 1 : $\exists y. a(x, y)$

- At most

 - 1 : $\forall y_1, y_2. a(x, y_1) \wedge a(x, y_2) \supset y_1 = y_2$

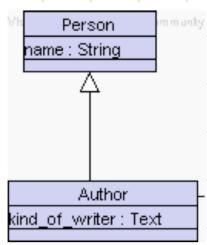
 - * : $\exists y_1, \dots, y_m. a(x, y_1) \wedge \dots \wedge a(x, y_m) \wedge y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge \dots \wedge y_{m-1} \neq y_m$

Some thing can be done with ASSOCIATIONS

eg $\forall x, y. \text{written_by}(x, y) \supset \text{Book}(x) \wedge \text{Author}(y)$

$\forall x. \text{Book}(x) \supset (\exists y. \text{written_by}(x, y))$

ISA



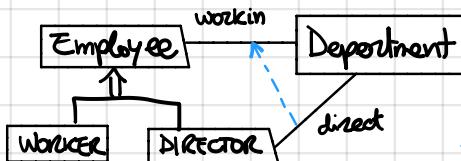
$\forall x. \text{Author}(x) \supset \text{Person}(x)$

$\forall x. \text{name}(x, y) \supset \text{Person}(x) \wedge \text{String}(y)$

$\forall x. \text{Author}(x) \supset (\exists y. \text{kind_of_writer}(x, y)) \wedge (\#y \leq 1)$

$\forall x, y. \text{kind_of_writer}(x, y) \supset \text{Author}(x) \wedge \text{Text}(y)$

This works also for "subsets" of associations



$\forall x, y. \text{Directs}(x, y) \supset \text{worksIn}(x, y)$

Note: direct inherits max multiplicity from worksIn

worksIn inherits min multiplicity from direct

This is logically implied anyway.

complete : $\forall x. E(x) \supset \text{Dir}(x) \vee \text{W}(x)$

disjoint : $\forall x. \text{W}(x) \supset \neg \text{Dir}(x)$ Note: viceversa not needed

Association Class = association with attributes

eg

Hosts
numb: int

$\forall x, y, z. \text{numbers}(x, y, z) \supset \text{hosts}(x, y) \wedge \text{int}(z)$

$\forall x, y. \text{hosts}(x, y) \supset (\exists z. \text{numbers}(x, y, z)) \wedge (\#z \leq 1)$

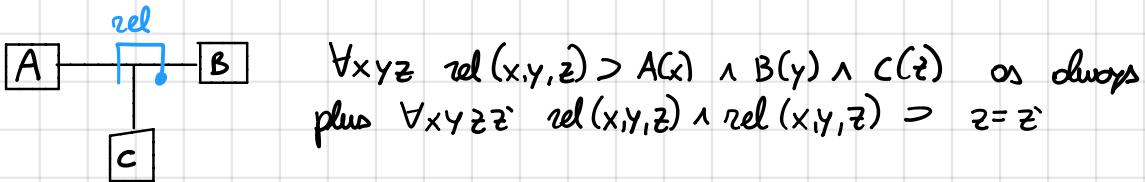
+ normal stuff of classic associations

KEY



$$\begin{aligned} \forall xy \text{ pd}(x,y) &\rightarrow p(x) \wedge d(y) \\ \forall xy \text{ pp}(x,y) &\rightarrow p(x) \wedge p(y) \end{aligned} \quad \left. \begin{array}{l} \text{+ multiplicity} \\ \text{+ } \end{array} \right\}$$

$$\forall x^* y z \text{ pd}(x,y) \wedge \text{pd}(y,z) \wedge \text{pp}(x,z) \wedge \text{pp}(y,z) \supset x = x^*$$



FORMS OF REASONING

Let Γ be the set of FOL assertions corresponding to UML diagram and $C(x)$ the predicate of class C .

C is consistent iff : $\Gamma \models \forall x C(x) \supset \text{false}$ i.e. there must exist a model of Γ where the extension of $C(x)$ is not the empty set

The diagram is consistent iff Γ is satisfiable i.e. Γ admits at least one model

C_1 subsumes C_2 if the class diagram implies that C_1 is a generalization of C_2 : $\Gamma \models \forall x C_2 \supset C_1(x)$

Two classes are equivalent if they denote the same set of instances whenever the conditions imposed by diagram are satisfied

CHECKING UML DIAGRAMS INSTANTIATIONS

An instantiation (object diagram) is an extension to all classes, associations and attributes, describing properties of single objects or relations between them → Describe actual data = database.

Two cases:

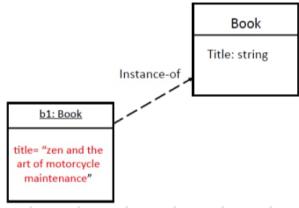
① NO ISA : diagram acts as constraints

② ISA : instantiation is partial, not explicit instantiate superclasses. Start from "complete" form.

① In this case the instantiation of the diagram is a first order interpretation of FOL :

$$I = (Obj^I, C^I_1, \dots, C^I_n, A^I_1, \dots, A^I_m, T^I_1, \dots, T^I_n, \alpha^I_1, \dots, \alpha^I_k)$$

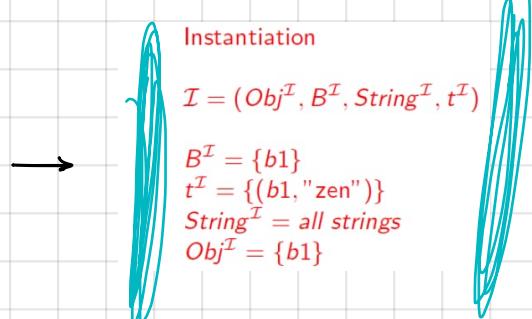
e.g.



$B(x), t(x, y), \text{String}(x)$

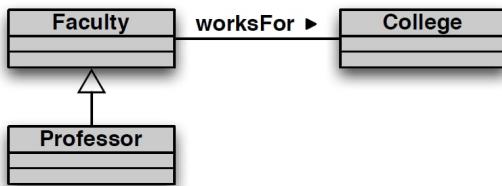
UML Class Diagram Axioms

$$\begin{aligned} \forall x, y. t(x, y) &\rightarrow B(x) \wedge \text{String}(y) \\ \forall x. B(x) &\rightarrow 1 \leq \#\{y \mid t(x, y)\} \leq 1 \end{aligned}$$



If the instantiation is correct we can query it just computing query q over I ignoring constraints Γ

② ISA - complete : each object in the subclass is also in the superclass



\mathcal{I} :
Faculty $^{\mathcal{I}}$ = { john, mary, paul }
Professor $^{\mathcal{I}}$ = { john, paul }
College $^{\mathcal{I}}$ = { collA, collB }
worksFor $^{\mathcal{I}}$ = { (john,collA), (mary,collB) }

ISA - NOT complete : for every object it is assumed that the instantiation state explicitly which are the most specific classes it is instance of

\mathcal{I} :
Faculty $^{\mathcal{I}}$: { mary } - incomplete!
Professor $^{\mathcal{I}}$: { john, paul }
College $^{\mathcal{I}}$: { collA, collB }
worksFor $^{\mathcal{I}}$: { (john,collA), (mary,collB) }

In this case, when answering query,
we have to use complete instantiation.
eg. John and Paul are still "Faculty" too

STRUCTURAL OPERATIONAL SEMANTICS OF PROGRAM

Evaluation Semantics: $(\delta, s) \rightarrow s'$ where δ is a program, s is memory state before evaluation and s' is the state after the evaluation.

How do we define this relation?

STRUCTURAL RULES

CONSEQUENT if SIDE-CONDITION
ANTECEDENT

$\Rightarrow \forall (\text{ANT} \wedge \text{SIDE} \supset \text{CONS})$

$$\text{Act : } \frac{(a, s) \longrightarrow s' \quad \text{if } s \models \text{Pre}(a) \text{ and } s' = \text{Post}(a, s)}{\text{true}}$$

special case: assignment $\frac{(x := v, s) \longrightarrow s' \quad \text{if } s' = s[x = v]}{\text{true}}$

$$\text{Skip : } \frac{(\text{skip}, s) \longrightarrow s}{\text{true}}$$

$$\text{Seq : } \frac{(\delta_1; \delta_2, s) \longrightarrow s'}{(\delta_1, s) \longrightarrow s'' \wedge (\delta_2, s'') \longrightarrow s'}$$

$$\text{if : } \frac{\begin{array}{c} (\text{if } \phi \text{ then } \delta_1 \text{ else } \delta_2, s) \longrightarrow s' \\ (\delta_1, s) \longrightarrow s' \end{array}}{\text{if } s \models \phi} \quad \frac{\begin{array}{c} (\text{if } \phi \text{ then } \delta_1 \text{ else } \delta_2, s) \longrightarrow s' \\ (\delta_2, s) \longrightarrow s' \end{array}}{\text{if } s \models \neg \phi}$$

$$\text{while : } \frac{\begin{array}{c} (\text{while } \phi \text{ do } \delta, s) \longrightarrow s \\ \text{true} \end{array}}{\text{if } s \models \neg \phi} \quad \frac{\begin{array}{c} (\text{while } \phi \text{ do } \delta, s) \longrightarrow s' \\ (\delta, s) \longrightarrow s'' \wedge (\text{while } \phi \text{ do } \delta, s'') \longrightarrow s' \end{array}}{\text{if } s \models \phi}$$

ex Compute S_f in following cases assuming that in S_0 we have $x=10$ and $y=0$

$$\bullet (x := x+1; x := x+z, S_0) \longrightarrow S_f$$

$$\hookrightarrow \frac{(x := x+1; x := x+z, S_0) \longrightarrow S_f}{(x := x+1, S_0) \rightarrow S_1 \wedge (x := x+z, S_1) \rightarrow S_f}$$

true true

$$S_1 : \quad x = 11$$

$$S_f : \quad x = 22$$

$$y = 0$$

$$y = 0$$

$$\bullet (x := x+1; \text{if } (x < 10) \text{ then } x := 0 \text{ else } x := 1; x := x+1, S_0) \longrightarrow S_f$$

$$\hookrightarrow \frac{(x := x+1; \text{if } (x < 10) \text{ then } x := 0 \text{ else } x := 1; x := x+1, S_0) \longrightarrow S_f}{\frac{x := x+1, S_0 \rightarrow S_1 \wedge (\text{if } \dots, x := x+1, S_1) \rightarrow S_f}{\frac{\text{true}}{(\text{if } \dots, S_1) \rightarrow S_2 \wedge (x+1, S_2) \rightarrow S_f}} \quad \frac{\text{true}}{\frac{x := 1, S_1 \rightarrow S_2}{\text{true}}}}}$$

$$S_1 : \quad x = 11$$

$$y = 0$$

$$S_2 : \quad x = 1$$

$$y = 0$$

$$S_f : \quad x = 12$$

$$y = 0$$

TRANSITION SEMANTICS

This kind of semantic just compute a single step: $\delta, s \rightarrow \delta', s'$

Given δ and s , compute the state s' and the program δ' that remains to be executed obtained by executing a single step of δ in s

Note: long arrow \longrightarrow for normal semantic, short arrow \rightarrow for transition semantics.

Γ \mathcal{E} is the empty program

$$Act : \frac{(a, s) \longrightarrow (\epsilon, s')}{\text{true}} \quad \text{if } s \models \text{Pre}(a) \text{ and } s' = \text{Post}(a, s)$$

special case: assignment $\frac{(x := v, s) \longrightarrow (\epsilon, s')}{\text{true}} \quad \text{if } s' = s[x = v]$

$$Skip : \frac{(\text{skip}, s) \longrightarrow (\epsilon, s)}{\text{true}}$$

$$Seq : \frac{(\delta_1; \delta_2, s) \longrightarrow (\delta'_1; \delta_2, s')}{(\delta_1, s) \longrightarrow (\delta'_1, s')}$$

$$\frac{(\delta_1; \delta_2, s) \longrightarrow (\delta'_2, s')}{(\delta_2, s) \longrightarrow (\delta'_2, s')} \quad \text{if } (\delta_1, s) \checkmark \rightarrow \text{if } \delta_1 \text{ is terminated} \\ = \mathcal{E}$$

$$if : \frac{(\text{if } \phi \text{ then } \delta_1 \text{ else } \delta_2, s) \longrightarrow (\delta'_1, s')}{(\delta_1, s) \longrightarrow (\delta'_1, s')} \quad \text{if } s \models \phi \quad \frac{(\text{if } \phi \text{ then } \delta_1 \text{ else } \delta_2, s) \longrightarrow (\delta'_2, s')}{(\delta_2, s) \longrightarrow (\delta'_2, s')} \quad \text{if } s \models \neg \phi$$

$$while : \frac{(\text{while } \phi \text{ do } \delta, s) \longrightarrow (\delta', \text{while } \phi \text{ do } \delta, s')}{(\delta, s) \longrightarrow (\delta', s')} \quad \text{if } s \models \phi$$

TERMINATION RULES

$$\epsilon : \frac{(\epsilon, s) \checkmark}{\text{true}}$$

$$Seq : \frac{(\delta_1; \delta_2, s) \checkmark}{(\delta_1, s) \checkmark \wedge (\delta_2, s) \checkmark}$$

$$if : \frac{(\text{if } \phi \text{ then } \delta_1 \text{ else } \delta_2, s) \checkmark}{(\delta_1, s) \checkmark} \quad \text{if } s \models \phi \quad \frac{(\text{if } \phi \text{ then } \delta_1 \text{ else } \delta_2, s) \checkmark}{(\delta_2, s) \checkmark} \quad \text{if } s \models \neg \phi$$

$$while : \frac{(\text{while } \phi \text{ do } \delta, s) \checkmark}{\text{true}} \quad \text{if } s \models \neg \phi \quad \frac{(\text{while } \phi \text{ do } \delta, s) \checkmark}{(\delta, s) \checkmark} \quad \text{if } s \models \phi$$

Eg Compute δ' , s' assuming in S_0 $x=10$ and $y=0$

$$\cdot (\text{while } (y < 2) \text{ do } \{x := x * 2, y := y + 1\}, S_0) \longrightarrow S_1 \quad \text{long arrow}$$

$(x := x * 2, S_0) \rightarrow S_1 \wedge (y := y + 1, S_1) \rightarrow S_2$

$(x := x * 2, S_0) \rightarrow S_2 \wedge (y := y + 1, S_2) \rightarrow S_1$ true

$(x := x * 2, S_1) \rightarrow S_3 \wedge (y := y + 1, S_3) \rightarrow S_4$ true

$(x := x * 2, S_2) \rightarrow S_4 \wedge (y := y + 1, S_4) \rightarrow S_3$ true

$S_2 = \begin{cases} x=20 \\ y=0 \end{cases} \quad S_1 = \begin{cases} x=20 \\ y=1 \end{cases}$

$S_4 = \begin{cases} x=40 \\ y=1 \end{cases} \quad S_3 = \begin{cases} x=40 \\ y=2 \end{cases}$

Something but with transition semantics

$$(\text{while } (y < 2) \text{ do } \{x := x * 2, y := y + 1\}, S_0) \rightarrow (\delta_1; \text{while } _, S_1)$$

$(x := x * 2, S_0) \rightarrow (\delta_1, S_1)$

$(x := x * 2, S_0) \rightarrow (\mathcal{E}, S_1)$

$$\delta_1 = \delta_2; y = y + 1 = \mathcal{E}; y = y + 1$$

$\delta_2 = \mathcal{E}$

$S_1 = \begin{cases} x=20 \\ y=0 \end{cases}$

$\mathcal{E}; y := y + 1; \text{while } _, S_1 \longrightarrow \delta_3, \text{while } _, S_2$

$$y := y + 1; \text{while } _, S_1 \longrightarrow \delta_3, S_2$$

$$y = y + 1, S_1 \longrightarrow (\mathcal{E}, S_2)$$

$$\delta_3 = \mathcal{E}, \text{while}$$

$S_2 = \begin{cases} x=40 \\ y=1 \end{cases}$

Continue in my nightmares...

To characterize a whole computation using single steps: $\xrightarrow{*}$

$$0 \text{ step : } \frac{(\delta, s) \xrightarrow{*} (\delta, s)}{\text{true}}$$

$$n \text{ step : } \frac{(\delta, s) \xrightarrow{*} (\delta'', s'')}{(\delta, s) \xrightarrow{*} (\delta', s') \wedge (\delta', s') \xrightarrow{*} (\delta'', s'')} \quad (\text{for some } \delta', s')$$

Notice that such relation is the reflexive-transitive closure of (single step) $\xrightarrow{*}$.

Th. For every while-program δ and states s and s_f :

$$(\delta, s_0) \longrightarrow s_f \equiv (\delta, s_0) \xrightarrow{*} (\delta_f, s_f) \wedge (\delta_f, s_f)^\checkmark \quad \text{for some } \delta_f$$

Why we do this transition semantic? Because it can handle concurrency.

- $(\delta_1 \parallel \delta_2)$ concurrent execution
 - if \emptyset then δ_1 , else δ_2 synchronized conditional
 - while \emptyset do δ synchronized loop
-] test is "free", no transition used for it.

$$\text{transition : } \frac{(\delta_1 \parallel \delta_2, s) \longrightarrow (\delta'_1 \parallel \delta_2, s')} {(\delta_1, s) \longrightarrow (\delta'_1, s')}$$

$$\frac{(\delta_1 \parallel \delta_2, s) \longrightarrow (\delta_1 \parallel \delta'_2, s')} {(\delta_2, s) \longrightarrow (\delta'_2, s')}$$

$$\text{termination : } \frac{(\delta_1 \parallel \delta_2, s)^\checkmark} {(\delta_1, s)^\checkmark \wedge (\delta_2, s)^\checkmark}$$

Presence of $\delta_1 \parallel \delta_2$ makes the transition relation **NONDETERMINISTIC**

HOARE LOGIC

Used to reason about the correctness of programs.

We start with some language we already saw (e.g. skip, assignment etc)

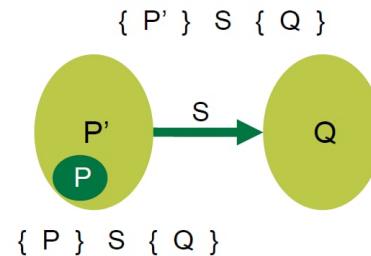
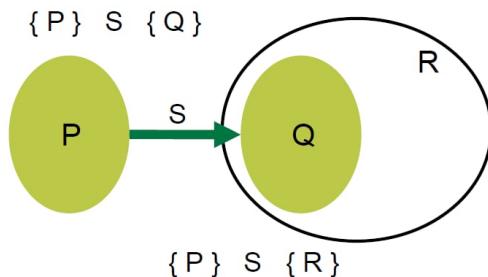
Given state $s \models P$ and $s' \models Q$, with program S we want to check that
 $s \models P \wedge S, s \rightarrow s' \supseteq s' \models Q$.

$\chi_P = \{s \mid s \models P\}$ = set of all states satisfying P
 $P(s) = s \in \chi_P$

P is stronger than Q , and is weaker than P if $P \Rightarrow Q$

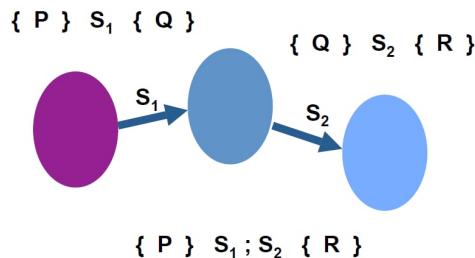
$\{P\} \text{ Program } \{Q\} = \forall s \quad s \models P \Rightarrow (\forall s' \quad s \text{ Pr } s' \Rightarrow s' \models Q)$
 $\delta, s \rightarrow s'$

Since the model does not express non-termination, we assume Pr terminates.
With this assumption we talk about partial correctness, total otherwise.



Post-condition weakening Rule:

$$\frac{\{P\} S \{Q\}, Q \Rightarrow R}{\{P\} S \{R\}}$$

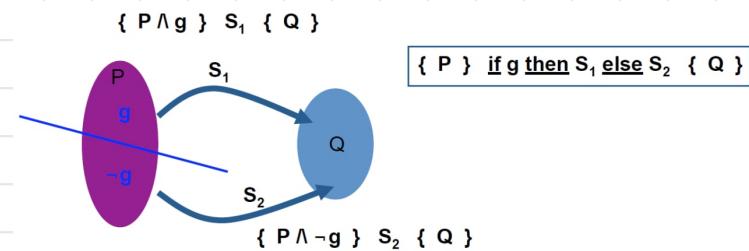


Pre-condition strengthening Rule:

$$\frac{P \Rightarrow P', \{P'\} S \{Q\}}{\{P\} S \{Q\}}$$

Premise 1, 2
 \downarrow
Conclusion

$$\frac{\{P\} S_1 \{Q\}, \{Q\} S_2 \{R\}}{\{P\} S_1; S_2 \{R\}}$$



$$\frac{\{P \wedge g\} S_1 \{Q\}, \{P \wedge -g\} S_2 \{Q\}}{\{P\} \text{if } g \text{ then } S_1 \text{ else } S_2 \{Q\}}$$

WEAKEST PRECONDITION

wp: Statement \times Pred \rightarrow Pred

Let $W = \text{wp}(S, Q)$:

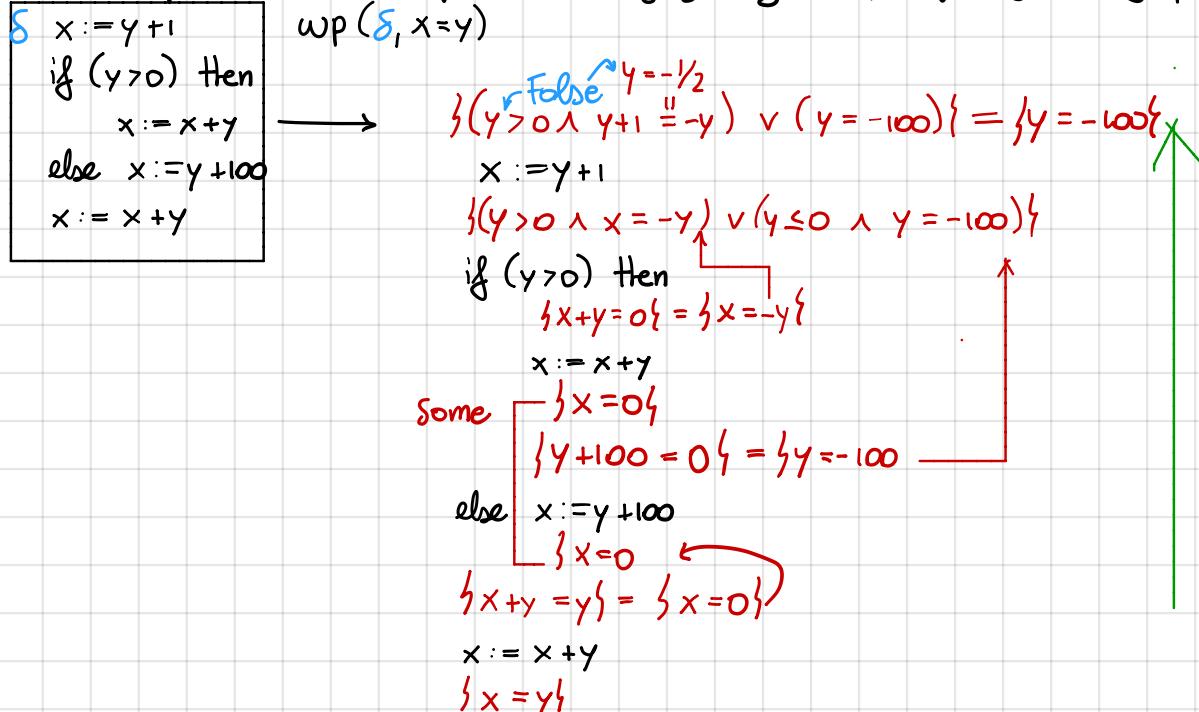
- **Reachability:** from any $S \models W$ if $S \sqsubseteq S'$ then $S' \models Q$
- **Maximality:** $S \sqsubseteq S'$ and $S' \models Q$ implies $S \models W$

$$\Rightarrow \{P\} S \{Q\} = P \Rightarrow \text{wp}(S, Q)$$

- wp skip $Q = Q$
- wp ($x := e$) $Q = Q[e/x]$
- $\text{wp}((S_1 ; S_2), Q) = \text{wp}(S_1, (\text{wp}(S_2, Q)))$
- $\text{wp}((\text{if } g \text{ then } S_1 \text{ else } S_2), Q) = g \wedge \text{wp}(S_1, Q) \vee \neg g \wedge \text{wp}(S_2, Q)$

Note: No while!

Ex. Compute the weakest precondition of getting $x=y$ from following program



WHILE

$$\begin{aligned} P \Rightarrow I \\ \{g \wedge I\} S \{I\} \\ I \wedge \neg g \Rightarrow Q \end{aligned}$$

setting up
invariance
exit condition

$$\{P\} \text{while } g \text{ do } S \{Q\}$$

Ex Check if following Hoare triple is correct using os invariant ($0 \leq i \wedge 0 \leq j \wedge i+j \leq 5$)

$\{i=0 \text{ AND } j=5\} \text{ while } (i < 5) \text{ do } \{j=j-1; i:=i+1\} \{j=0\}$

Solution: $P = \{i=0 \text{ AND } j=5\}$

$Q = \{j=0\}$

$I = \{0 \leq i \wedge 0 \leq j \wedge i+j \leq 5\}$

1. Check $P \triangleright I$

$i=0 \wedge j=5 \supset 0 \leq i \wedge 0 \leq j \wedge i+j \leq 5 \quad \checkmark$

3. Check $\{g \wedge I\} \triangleright Q$

$i \geq 5 \wedge 0 \leq i \wedge 0 \leq j \wedge i+j \leq 5 \supset j=0 \quad \checkmark$

2. Check $\{g \wedge I\} \delta \{I'\} = g \wedge I \supset \text{wp}(\delta, I)$

$i < 5 \wedge 0 \leq i \wedge 0 \leq j \wedge i+j \leq 5 \supset \text{wp}(\delta, I) \quad \{0 \leq i+1 \wedge 0 \leq j-1 \wedge i+1+j-1 \leq 5\}$

$j := j-1$

$\{0 \leq i+1 \wedge 0 \leq j \wedge i+1+j \leq 5\}$

$i = i+1$

$\{0 \leq i \wedge 0 \leq j \wedge i+j \leq 5\}$

Checking i and j we can say that the Hoare triple is correct using I as invariant

Ex Compute weakest precondition for getting $\{x=0\}$ for following program:

```

x := 50+y
if (x > 50) then {
    if (y > 0) then {
        x := x - y
    } else y := y
}
else x := x + y
y := y + 50
  
```

$\{y = 25\}$

$\{50 = 0\} \quad \{y \leq 50 \wedge y = -25\}$

$\{50+y > 50 \wedge y = 50+y \vee 50+y \leq 50 \wedge 50+y+y = 0\}$

$x := 50+y \quad \{x > 50 \wedge y = x \vee x \leq 50 \wedge x+y = 0\}$

$x > 50 \wedge (y > 0 \wedge x = y \vee y \leq 0 \wedge x = 0) \vee (x \leq 50 \wedge x+y = 0)$

$\rightarrow \text{if } (x > 50) \text{ then } \{y > 0 \wedge x = y \vee y \leq 0 \wedge x = 0\}$

$\text{if } (y > 0) \text{ then }$

$\{x-y = 0\}$

$x := x - y$

$\{x = 0\}$

$\{x = 0\}$

$\text{else } y := y$

$\{x = 0\}$

$\{x+y = 0\}$

$\rightarrow \text{else } x := x + y$

$\{x = 0\}$

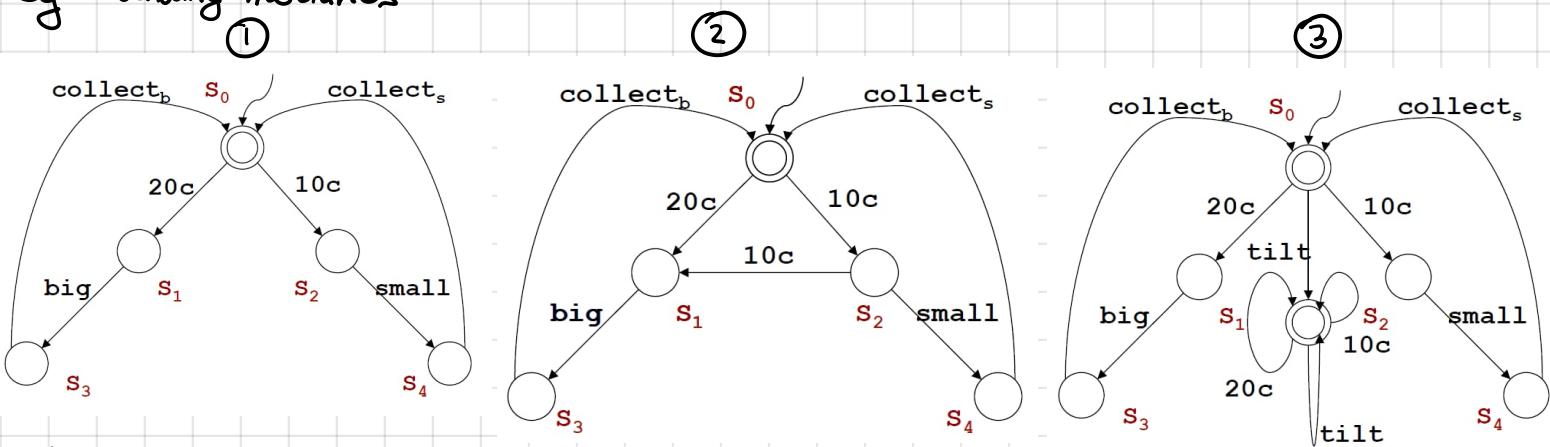
$y := y + 50$

$\{x = 0\}$

TRANSITION SYSTEMS AND BISIMULATION

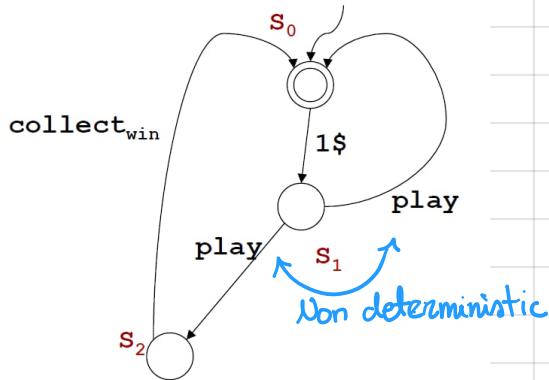
Graph of transitions. Termination and reachability (of final state) let us query the system
(fixpoint theory, model checking)

Eg vending machines



Eg slot machine

NON DETERMINISTIC



A transition system TS is a tuple $T = \langle A, S, S^0, \delta, F \rangle$ where :

- $A = \text{set of actions}$
- $S = \text{set of states}$
- $S^0 \subseteq S = \text{set of initial states}$
- $\delta \subseteq S \times A \times S = \text{transition relation}$
- $F \subseteq S = \text{set of final states}$

+ equivalent relations!

REACHABILITY

A binary relation R is a **reachability-like relation** iff :

- $(s, s) \in R$
- if $\exists a, s' \text{ s.t. } s \xrightarrow{a} s' \wedge (s, s') \in R$ then $(s, s') \in R$

A state s_0 reaches a state s_f iff \forall reachability-like relations R we have $(s_0, s_f) \in R$

Algorithm ComputingReachability

Input: transition system TS

Output: the **reachable-from** relation (the smallest reachability-like relation)

Body

```

R = ∅
R' = {(s, s) | s ∈ S}
while (R ≠ R') {
    R := R'
    R' := R' ∪ {((s, s'') | ∃ s', a. s →a s' ∧ (s', s'') ∈ R)}
}
return R'
```

YdoB

BISIMULATION

Two transition systems are **bisimilar** if they have the same behaviour:

- locally they look indistinguishable
- every action that can be done on one of them can also be done on the other remaining indistinguishable.

R is a **bisimulation** iff $(s, t) \in R$ implies that:

- s is final iff t is final
- \forall actions a :
 - if $s \xrightarrow{a} s'$ then $\exists t' \text{ s.t. } t \xrightarrow{a} t' \text{ and } (s', t') \in R$
 - if $t \xrightarrow{a} t'$ then $\exists s' \text{ s.t. } s \xrightarrow{a} s' \text{ and } (s', t') \in R$

A state s_0 is bisimilar (or equivalent) to state t_0 iff there exists a bisimulation between the initial states s_0 and t_0 .

LEAST AND GREATEST FIXPOINTS

Consider $X = f(X)$ where f is an operator from 2^S to 2^S (2^S = set of all subsets of S). Every solution E of this equation is called fixpoint of f . Smallest and greatest solution are called least and greatest fixpoints.

Every E s.t. $f(E) \subseteq E$ is called pre-fixpoint

Every E s.t. $E \subseteq f(E)$ is called post-fixpoint

Operator from subset of status to subset of status

f is monotonic if $E_1 \subseteq E_2$ implies $f(E_1) \subseteq f(E_2)$. In this case:

- there exists a unique least fixpoint of f which is given by: $\bigcap \{E \subseteq S \mid f(E) \subseteq E\}$
- there exists a unique greatest fixpoint of f which is given by: $\bigcup \{E \subseteq S \mid E \subseteq f(E)\}$

pre-fixpoints

post-fixpoints

APPROXIMATES

The approximates for a least fixpoint $L = \bigcap \{E \subseteq S \mid f(E) \subseteq E\}$ are as follow:

$$\begin{aligned} z_0 &= \emptyset \\ z_i &= f(z_{i-1}) \\ z_2 &= f(z_1) \end{aligned} \quad \left\{ \begin{array}{l} \forall i, z_i \subseteq z_{i+1}, \quad \forall i \quad z_i \subseteq L \\ \dots \end{array} \right.$$

Th. If for some n $z_{n+1} = z_n$ then $z_n = L$

Least fixpoint algorithm

```

 $Z_{old} := \emptyset;$ 
 $Z := f(Z_{old});$ 
while ( $Z \neq Z_{old}$ ) {
     $Z_{old} := Z;$ 
     $Z := f(Z);$ 
}

```

Similar for greatest fixpoint $G = \bigcup \{E \subseteq S \mid E \subseteq f(E)\}$.

$$\begin{aligned} z_0 &= S \\ z_1 &= f(z_0) \\ z_2 &= f(z_1) \end{aligned} \quad \left\{ \begin{array}{l} \forall i \quad z_{i+1} \subseteq z_i, \quad \forall i \quad G \subseteq z_i \\ \dots \end{array} \right. \quad \text{Contrary.}$$

Th. If for some n , $z_{n+1} = z_n$ then $z_n = G$

Greatest fixpoint algorithm

```

 $Z_{old} := S;$ 
 $Z := f(Z_{old});$ 
while ( $Z \neq Z_{old}$ ) {
     $Z_{old} := Z;$ 
     $Z := f(Z);$ 
}

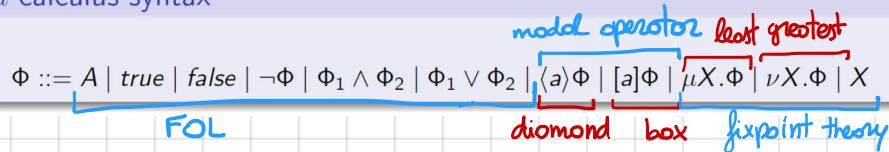
```

μ - CALCULUS

Logic based on fixpoints. Composed by:

- Propositions: to denote properties of the global store in a given configuration
- Modalities: to denote the capability of performing certain actions in a given configuration
- Least/Greatest Fixpoints: to denote "temporal" properties of the system (induction/coinduction)

μ -calculus syntax



Diamond = exist a state where ϕ holds

Box = every states imply ϕ

For formulae of the form $\mu X . \Phi$ and $\nu X . \Phi$ we require syntactic monotonicity of Φ .

Every occurrence of variable X in Φ must be within the scope of an even number of negation signs.

In μ -calculus, given the requirement of syntactic monotonicity, the least fixpoint and the greatest one **ALWAYS** exist.

Transition System: $T = (S, \{R_a \mid a \in A\}, \Pi)$

P = given proposition set

$\rightarrow S$ = set of states

A = given set of atomic actions

R_a = family of relations

Π = mapping from P to a subset of S

Valuation: given transition system T, a valuation V on T is a mapping from variables in Var to subsets of states in T.

μ -calculus semantics

$(A)_V^T$	$= \Pi(A) \subseteq S$
$(X)_V^T$	$= V(X) \subseteq S$
$(\text{true})_V^T$	$= S$
$(\text{false})_V^T$	$= \emptyset$
$(\neg \Phi)_V^T$	$= S - (\Phi)_V^T$
$(\Phi_1 \wedge \Phi_2)_V^T$	$= (\Phi_1)_V^T \cap (\Phi_2)_V^T$
$(\Phi_1 \vee \Phi_2)_V^T$	$= (\Phi_1)_V^T \cup (\Phi_2)_V^T$
$(\langle a \rangle \Phi)_V^T$	$= \{s \in S \mid \exists s'. (s, s') \in R_a \text{ and } s' \in (\Phi)_V^T\}$
$([a] \Phi)_V^T$	$= \{s \in S \mid \forall s'. (s, s') \in R_a \text{ implies } s' \in (\Phi)_V^T\}$
$(\mu X . \Phi)_V^T$	$= \bigcap \{\mathcal{E} \subseteq S \mid (\Phi)_{V[X \leftarrow \mathcal{E}]}^T \subseteq \mathcal{E}\}$ least fp. of Φ
$(\nu X . \Phi)_V^T$	$= \bigcup \{\mathcal{E} \subseteq S \mid \mathcal{E} \subseteq (\Phi)_{V[X \leftarrow \mathcal{E}]}^T\}$ greatest fp. of Φ

Given $T = (S, \{R_a \mid a \in A\}, \Pi)$

• $(\Phi)_V^T$ = set of states that satisfy Φ

Examples:

- 1) $\langle \text{next} \rangle \text{true}$ = capability of making a next-transition
- 2) $[\text{next}] \text{false}$ = inability of making any next-transition
- 3) $\langle \text{next} \rangle \text{true} \wedge [\text{next}] P$ = next-transition are allowed and all reach states where P holds
- 4) $\mu X . P \vee \langle \text{next} \rangle X$ = exists an evolution of the system s.t. P eventually holds.
- 5) $\nu X . P \wedge [\text{next}] X$ = $\neg \mu X . \neg P \vee \langle \text{next} \rangle X$ = invariance of P under all of the evolutions of the system.
- 6) $\mu X . P \vee (\langle \text{next} \rangle \text{true} \wedge [\text{next}] X)$ = for all evolutions of the system P eventually holds
- 7) $\nu X . \mu Y . (P \wedge \langle \text{next} \rangle X) \vee \langle \text{next} \rangle Y$ = strong fairness, there exists a run where P is true infinitely often

We are interested in **model checking**: querying the transition system.

Let $T = (S, \{R_a\}_{a \in A}, \Pi)$ be a transition system, let $s \in S$ be one of its states and ϕ be a closed (no free variables) μ -calculus formula. The related **model checking problem** is to verify whether $s \in (\phi)_V^T$ where V is any valuation since ϕ is closed. Also written $T, s \models \phi$ or just $s \models \phi$.

μ -calculus model checking algorithm

$\llbracket A \rrbracket_V^T = \Pi(A)$	$\llbracket \langle a \rangle \Phi \rrbracket_V^T = \text{PREE}(a, \llbracket \Phi \rrbracket_V^T)$
$\llbracket X \rrbracket_V^T = V(X)$	$\llbracket [a]\Phi \rrbracket_V^T = \text{PREA}(a, \llbracket \Phi \rrbracket_V^T)$
$\llbracket \text{true} \rrbracket_V^T = S$	$\llbracket \mu X. \Phi \rrbracket_V^T = \text{LFP}_X. \llbracket \Phi \rrbracket_V^T$
$\llbracket \text{false} \rrbracket_V^T = \emptyset$	$\llbracket \nu X. \Phi \rrbracket_V^T = \text{GFP}_X. \llbracket \Phi \rrbracket_V^T$
$\llbracket \neg \Phi \rrbracket_V^T = S - \llbracket \Phi \rrbracket_V^T$	
$\llbracket \Phi_1 \wedge \Phi_2 \rrbracket_V^T = \llbracket \Phi_1 \rrbracket_V^T \cap \llbracket \Phi_2 \rrbracket_V^T$	
$\llbracket \Phi_1 \vee \Phi_2 \rrbracket_V^T = \llbracket \Phi_1 \rrbracket_V^T \cup \llbracket \Phi_2 \rrbracket_V^T$	

$\text{PreE}(a, E) = \text{existential } a\text{-preimage of } E = \{s \in S \mid \exists s' \cdot (s, s') \in R_a \text{ and } s' \in E\}$
 $\text{PreA}(a, E) = \text{universal } a\text{-preimage of } E = \{s \in S \mid \forall s' \cdot (s, s') \in R_a \text{ implies } s' \in E\}$

Procedure $\text{LFP}_X. \llbracket \Phi \rrbracket_V^T$

```

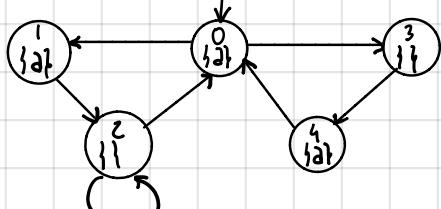
 $\mathcal{X}_{old} := \llbracket \text{False} \rrbracket_V^T;$ 
 $\mathcal{X} := \llbracket \Phi \rrbracket_V^T[x \leftarrow \mathcal{X}_{old}]$ ;
while ( $\mathcal{X} \neq \mathcal{X}_{old}$ ) {
     $\mathcal{X}_{old} := \mathcal{X}$ ;
     $\mathcal{X} := \llbracket \Phi \rrbracket_V^T[x \leftarrow \mathcal{X}_{old}]$ ;
}
return  $\mathcal{X}$ ;
```

Procedure $\text{GFP}_X. \llbracket \Phi \rrbracket_V^T$

```

 $\mathcal{X}_{old} := \llbracket \text{True} \rrbracket_V^T$ ;
 $\mathcal{X} := \llbracket \Phi \rrbracket_V^T[x \leftarrow \mathcal{X}_{old}]$ ;
while ( $\mathcal{X} \neq \mathcal{X}_{old}$ ) {
     $\mathcal{X}_{old} := \mathcal{X}$ ;
     $\mathcal{X} := \llbracket \Phi \rrbracket_V^T[x \leftarrow \mathcal{X}_{old}]$ ;
}
return  $\mathcal{X}$ ;
```

Ex Model check the μ -calculus formula
transition system.



Let's define X_i the i -th approximation of X

$[X_i] = \text{states that make formula true}$

$[X_0] = [\text{false}] = \emptyset$

$$\begin{aligned}
 [X_1] &= [\neg a \vee \text{next} X_0] = [\neg a] \cup [\text{next} X_0] \\
 &= [\neg a] \cup \text{PreE}(\text{next}, [X_0]) \cup \emptyset \\
 &= \{2, 3\} \cup \emptyset = \{2, 3\}
 \end{aligned}$$

This set MUST get bigger!

$$[X_2] = [\neg a \vee \text{next} X_1] = [\neg a] \cup \text{PreE}(\text{next}, [X_1]) = \{2, 3\} \cup \{0, 1, 2\} = \{0, 1, 2, 3\}$$

$$[X_3] = [\neg a \vee \text{next} X_2] = [\neg a] \cup \text{PreE}(\text{next}, [X_2]) = \{2, 3\} \cup \{0, 1, 2, 4\} = \{0, 1, 2, 3, 4\}$$

$$[X_4] = [\neg a \vee \text{next} X_3] = [\neg a] \cup \text{PreE}(\text{next}, [X_3]) = \{2, 3\} \cup \{0, 1, 2, 3, 4\} = \{0, 1, 2, 3, 4\}$$

Again! OK

Equal!

Least Fix Point

Since our initial state is in (the "solution"), we can say that is true that from initial state we can reach a state where $\neg a$ is true.

What if we change formula to $\vee X. \neg a \vee \text{next} X$?

$$[X_0] = \text{entire set} = \{0, 1, 2, 3, 4\}$$

$$[X_1] = [\neg a \vee \text{next} X] = [\neg a] \cup \text{PreE}(\text{next}, [X_0]) = \{2, 3\} \cup \{0, 1, 2, 3, 4\} = \{0, 1, 2, 3, 4\}$$

Greatest Fix Point

$[v X. \top \wedge \langle \text{next} \rangle X] = \top$ always true

$$[x_0] = \{1, 2, 3, 4\}$$

In this case the set MUST shrink!

$$[x_1] = [\top \wedge \langle \text{next} \rangle x_0] = [\top] \wedge \text{PreE}(\text{next}, [x_0]) = \{2, 3\} \cap \{0, 2, 3, 4\} = \{2, 3\}$$

$$[x_2] = [\top \wedge \langle \text{next} \rangle x_1] = [\top] \wedge \text{PreE}(\text{next}, [x_1]) = \{2, 3\} \cap \{0, 1, 2\} = \{2\}$$

$$[x_3] = [\top \wedge \langle \text{next} \rangle x_2] = [\top] \wedge \text{PreE}(\text{next}, [x_2]) = \{2, 3\} \cap \{1, 2\} = \{2\}$$

Equal! (Greatest Fixpoint)

Since our initial state is not in the fixpoint the formula is false

$[\mu X. \top \vee [\text{next}] X] = \text{all executions reach } \top \text{ eventually}$

$$[x_0] = \emptyset$$

$$[x_1] = [\top \vee [\text{next}] x_0] = [\top] \cup \text{PreA}(\text{next}, [x_0]) = \{2, 3\} \cup \{\emptyset\} = \{2, 3\}$$

$$[x_2] = [\top \vee [\text{next}] x_1] = [\top] \cup \text{PreA}(\text{next}, [x_1]) = \{2, 3\} \cup \{1\} = \{1, 2, 3\}$$

$$[x_3] = [\top \vee [\text{next}] x_2] = [\top] \cup \text{PreA}(\text{next}, [x_2]) = \{2, 3\} \cup \{0, 1\} = \{0, 1, 2, 3\}$$

$$[x_4] = [\top \vee [\text{next}] x_3] = [\top] \cup \text{PreA}(\text{next}, [x_3]) = \{2, 3\} \cup \{0, 1, 2, 4\} = \{0, 1, 2, 3, 4\}$$

$$[x_5] = [\top \vee [\text{next}] x_4] = [\top] \cup \text{PreA}(\text{next}, [x_4]) = \{2, 3\} \cup \{0, 1, 2, 3, 4\} = \{0, 1, 2, 3, 4\}$$

Least Fix Point

Initial state in solution so formula is true

$[v X \mu Y. (\top \wedge \langle \text{next} \rangle X \vee [\text{next}] Y)]$

$$[x_0] = \{0, 1, 2, 3, 4\}$$

$$[x_1] = [\mu Y (\top \wedge \langle \text{next} \rangle x_0 \vee [\text{next}] Y)] = \{0, 1, 3, 4\}$$

$$[y_{10}] = \emptyset$$

$$[y_{11}] = [\top \wedge \langle \text{next} \rangle x_0 \vee [\text{next}] y_{10}] = [\top] \wedge \text{PreE}(\text{next}, [x_0]) \cup \text{PreA}(\text{next}, y_{10}) = \{0, 1, 4\} \cap \{0, 1, 2, 3, 4\} \cup \emptyset = \{0, 1, 4\}$$

$$[y_{12}] = [\top \wedge \langle \text{next} \rangle x_0 \vee [\text{next}] y_{11}] = [\top] \wedge \text{PreE}(\text{next}, [x_0]) \cup \text{PreA}(\text{next}, y_{11}) = \{0, 1, 4\} \cup \{4, 3\} = \{0, 1, 3, 4\}$$

$$[y_{13}] = [\top \wedge \langle \text{next} \rangle x_0 \vee [\text{next}] y_{12}] = [\dots] = \{0, 1, 4\} \cup \{0, 3, 4\} = \{0, 1, 3, 4\}$$

$$[x_2] = [\mu Y (\top \wedge \langle \text{next} \rangle X, \vee [\text{next}] Y)] = \{0, 3, 4\}$$

$$[y_{20}] = \emptyset$$

$$[y_{21}] = [\top \wedge \langle \text{next} \rangle X, \vee [\text{next}] y_{20}] = \{0, 1, 4\} \cap \{0, 2, 3, 4\} \cup \emptyset = \{0, 4\}$$

$$[y_{22}] = [\top \wedge \langle \text{next} \rangle X_1 \vee [\text{next}] Y_{21}] = \{0, 4\} \cup \{3, 4\} = \{0, 3, 4\}$$

$$[y_{23}] = [\top \wedge \langle \text{next} \rangle X_1 \vee [\text{next}] Y_{22}] = \{0, 4\} \cup \{3, 4\} = \{0, 3, 4\}$$

$$[x_3] = [\mu Y (\top \wedge \langle \text{next} \rangle X_2 \vee [\text{next}] Y)] = \{0, 3, 4\}$$

$$[y_{30}] = \emptyset$$

$$[y_{31}] = [\top \wedge \langle \text{next} \rangle X_2 \vee [\text{next}] Y_{30}] = \{0, 1, 4\} \cap \{0, 2, 3, 4\} \cup \emptyset = \{0, 4\}$$

$$[y_{32}] = [\top \wedge \langle \text{next} \rangle X_2 \vee [\text{next}] Y_{31}] = \{0, 4\} \cup \{3, 4\} = \{0, 3, 4\}$$

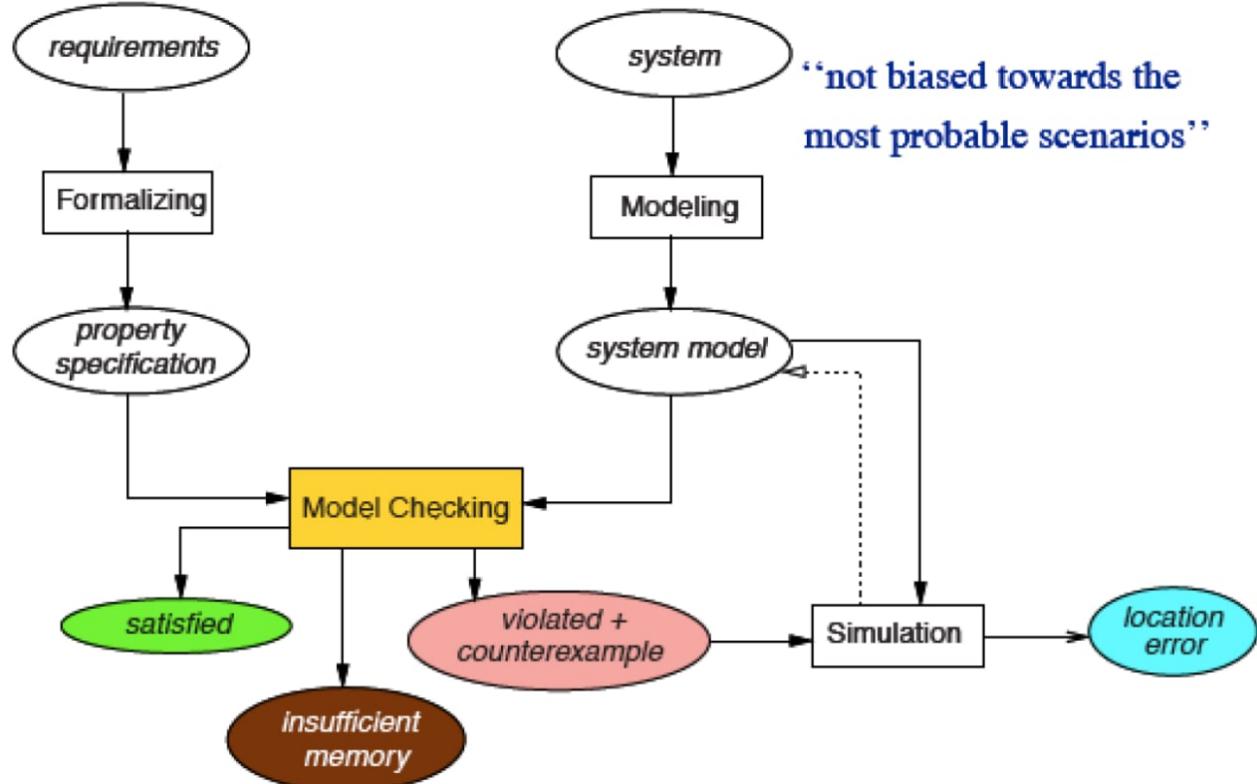
$$[y_{33}] = [\top \wedge \langle \text{next} \rangle X_2 \vee [\text{next}] Y_{32}] = \{0, 4\} \cup \{3, 4\} = \{0, 3, 4\}$$



$X_2 = X_3 \rightarrow$ Greatest Fix Point

Initial state in solution \rightarrow formula satisfied!

System Verification: check whether a system fulfills the qualitative requirements that have been identified. Checking we are building something in the right way.



LINEAR TEMPORAL LOGIC (LTL)

Only one timeline *✓*

$\bigcirc \varphi$	φ is true in the <u>next</u> moment in time
$\Box \varphi$	φ is true in <u>all future</u> moments
$\Diamond \varphi$	φ is true in <u>some</u> future moment
$\varphi \sqcup \psi$	φ is true <u>until</u> ψ is true

SEMANTIC

$\langle M, i \rangle \models p$ iff $p \in I(i)$ for $p \in \Sigma = \text{set of atomic propositions}$

$\langle M, i \rangle \models \neg \varphi$ iff $\langle M, i \rangle \not\models \varphi$

$\langle M, i \rangle \models \varphi \wedge \psi$ iff $\langle M, i \rangle \models \varphi$ and $\langle M, i \rangle \models \psi$

$\langle M, i \rangle \models \varphi \vee \psi$ iff $\langle M, i \rangle \models \varphi$ or $\langle M, i \rangle \models \psi$

$\langle M, i \rangle \models \varphi \Rightarrow \psi$ iff if $\langle M, i \rangle \models \varphi$ then $\langle M, i \rangle \models \psi$

NEXT

Provides a constraint on the next moment in time. $\langle M, i \rangle \models \bigcirc \varphi$ iff $\langle M, i+1 \rangle \models \varphi$

e.g. $(\text{socd} \wedge \neg \text{rich}) \Rightarrow \bigcirc \text{socd}$

EVENTUALLY / SOMETIMES

φ will be true but we don't know when. $\langle M, i \rangle \models \Diamond \varphi$ iff $\exists j (j \geq i) \wedge \langle M, j \rangle \models \varphi$

e.g. $(\neg \text{resigned} \wedge \text{socd}) \Rightarrow \Diamond \text{famous}$

ALWAYS

Invariant properties. $\langle M, i \rangle \models \Box \varphi$ iff $\forall j \text{ if } (j \geq i) \text{ then } \langle M, j \rangle \models \varphi$

e.g. lottery-win $\Rightarrow \Box \text{ rich}$

UNTIL

$\langle M, i \rangle \models \varphi \sqcup \psi$ iff $\exists j (j \geq i) \wedge \langle M, j \rangle \models \psi \wedge \forall k (i \leq k \leq j) \Rightarrow \langle M, k \rangle \models \varphi$

e.g. start lecture \Rightarrow talk \sqcup end-lecture

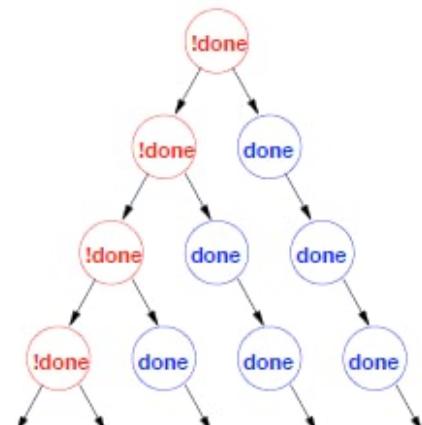
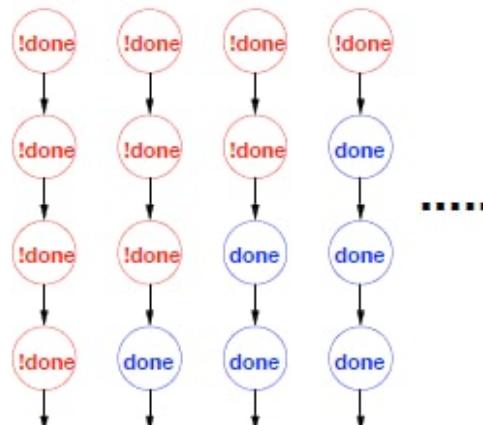
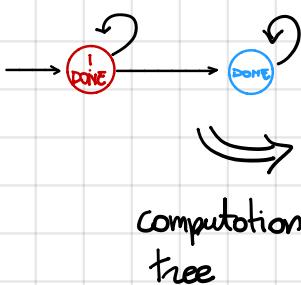
born \Rightarrow alive \sqcup dead (φ cont last forever and ψ sooner or later will happen)

Note:

$$1) \neg \Box \varphi \equiv \Diamond \neg \varphi$$

$$2) \Diamond \varphi \equiv \top \sqcup \varphi$$

true



Some times symbols are translated with letters:

\diamond → Future

\square → Globally in the future

\circ → neXtime

COMPUTATION TREE LOGIC (CTL)

Based on the possibility of several "branches" (as we saw before).

CTL explicitly introduces path quantifiers:

- all paths = A
- exists a path = E

Use some temporal operators as LTL that can be combined with quantifiers:

Universal modalities AF, AG, AX, AU
Existential modalities EF, EG, EX, EU

Syntax for single state:

$\mathcal{KM}, s_i \models \neg\varphi$	iff $\mathcal{KM}, s_i \not\models \varphi$	Same as before.
$\mathcal{KM}, s_i \models \varphi \wedge \psi$	iff $\mathcal{KM}, s_i \models \varphi$ and $\mathcal{KM}, s_i \models \psi$	
$\mathcal{KM}, s_i \models \varphi \vee \psi$	iff $\mathcal{KM}, s_i \models \varphi$ or $\mathcal{KM}, s_i \models \psi$	
$\mathcal{KM}, s_i \models \varphi \Rightarrow \psi$	iff if $\mathcal{KM}, s_i \models \varphi$ then $\mathcal{KM}, s_i \models \psi$	
$\mathcal{KM}, s_i \models \top$		
$\mathcal{KM}, s_i \not\models \perp$		

For a path $\pi = (s_i, s_{i+1}, \dots)$ outgoing from state s_i :

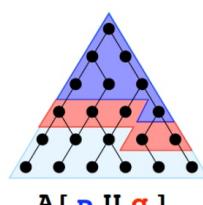
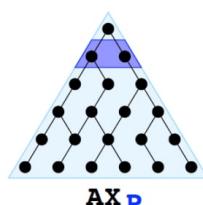
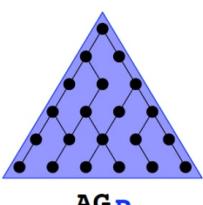
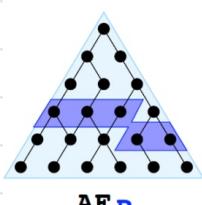
$\mathcal{KM}, s_i \models \mathbf{AX}\varphi$	iff $\forall \pi = (s_i, s_{i+1}, \dots) \quad \mathcal{KM}, s_{i+1} \models \varphi$
$\mathcal{KM}, s_i \models \mathbf{EX}\varphi$	iff $\exists \pi = (s_i, s_{i+1}, \dots) \quad \mathcal{KM}, s_{i+1} \models \varphi$
$\mathcal{KM}, s_i \models \mathbf{AG}\varphi$	iff $\forall \pi = (s_i, s_{i+1}, \dots) \quad \forall j \geq i. \mathcal{KM}, s_j \models \varphi$
$\mathcal{KM}, s_i \models \mathbf{EG}\varphi$	iff $\exists \pi = (s_i, s_{i+1}, \dots) \quad \forall j \geq i. \mathcal{KM}, s_j \models \varphi$
$\mathcal{KM}, s_i \models \mathbf{AF}\varphi$	iff $\forall \pi = (s_i, s_{i+1}, \dots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \varphi$
$\mathcal{KM}, s_i \models \mathbf{EF}\varphi$	iff $\exists \pi = (s_i, s_{i+1}, \dots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \varphi$
$\mathcal{KM}, s_i \models (\varphi \mathbf{AU} \psi)$	iff $\forall \pi = (s_i, s_{i+1}, \dots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \psi$ and $\forall i \leq k < j : \mathcal{M}, s_k \models \varphi$
$\mathcal{KM}, s_i \models (\varphi \mathbf{EU} \psi)$	iff $\exists \pi = (s_i, s_{i+1}, \dots) \quad \exists j \geq i. \mathcal{KM}, s_j \models \psi$ and $\forall i \leq k < j : \mathcal{M}, s_k \models \varphi$

finally P

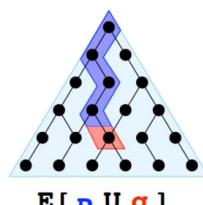
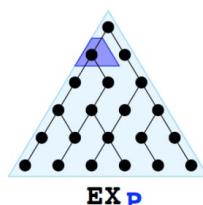
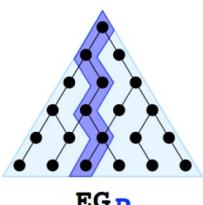
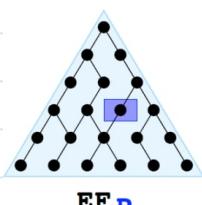
globally P

next P

P until q



Logic of CTL is based on combining these 8 operators!



MODEL CHECKING

Construct denotation of $\varphi = \text{set of states where formula holds}$ and then compare it with set of initial states. Denotation of $\varphi = [\varphi]$

$\exists p \leftrightarrow \langle \text{next} \rangle p$ } \rightarrow we can use μ -calculus, we must "translate" to μ -calculus.
 $\forall p \leftrightarrow [\text{next}] p$ }

CTL	μ -calculus
p	p
$\vee, \wedge, \rightarrow$	$\vee, \wedge, \rightarrow$
$\exists \forall \varphi$	$\langle \text{next} \rangle \dot{+} (\varphi)$
$\forall \exists \varphi$	$[\text{next}] \dot{+} (\varphi)$
$\exists F \varphi$	$\mu z \dot{+} (\varphi) \vee \langle \text{next} \rangle z$
$A F \varphi$	$\mu z \dot{+} (\varphi) \vee [\text{next}] z$
$E G \varphi$	$\nu z \dot{+} (\varphi) \wedge \langle \text{next} \rangle z$
$A G \varphi$	$\nu z \dot{+} (\varphi) \wedge [\text{next}] z$
$\varphi E U \psi$	$\mu z \dot{+} (\varphi) \wedge (\dot{+}(\psi) \wedge \langle \text{next} \rangle z)$
$\varphi A U \psi$	$\mu z \dot{+} (\varphi) \vee (\dot{+}(\psi) \wedge [\text{next}] z)$

Eg Model checking

$$AG(p \rightarrow AFq)$$

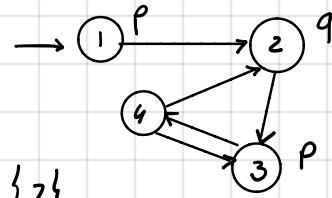
$$\underline{d = AFq \rightarrow \mu z q \vee [\text{next}] z = \{1, 2\}}$$

$$[z_0] = \emptyset$$

$$[z_1] = [q \vee \langle \text{next} \rangle z_0] = [q] \cup \text{PreA}(\text{next}, [z_0]) = \{2\}$$

$$[z_2] = [q \vee \langle \text{next} \rangle z_1] = [q] \cup \text{PreA}(\text{next}, [z_1]) = \{2\} \cup \{1\} = \{1, 2\}$$

$$[z_3] = [q \vee \langle \text{next} \rangle z_2] = [q] \cup \text{PreA}(\text{next}, [z_2]) = \{2\} \cup \{1\} = \{1, 2\}$$



$$\beta = \neg p \vee d \rightarrow [\neg p] \cup [d] = \{1, 2, 4\}$$

$$\gamma = AG \beta \rightarrow \nu z \beta \wedge [\text{next}] z = \emptyset$$

$$[z_0] = \{1, 2, 3, 4\}$$

$$[z_1] = [\beta \wedge \langle \text{next} \rangle z_0] = [\beta] \cap \text{PreA}(\text{next}, [z_0]) = \{1, 4\} \cap \{1, 2, 3, 4\} = \{1, 2, 4\}$$

$$[z_2] = [\beta \wedge \langle \text{next} \rangle z_1] = [\beta] \cap \text{PreA}(\text{next}, [z_1]) = \{1, 2, 4\} \cap \{1, 3\} = \{1\}$$

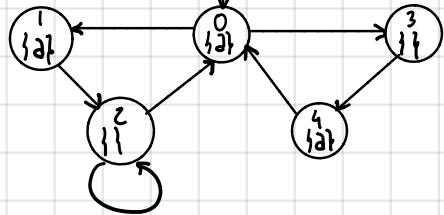
$$[z_3] = [\beta \wedge \langle \text{next} \rangle z_2] = [\beta] \cap \text{PreA}(\text{next}, [z_2]) = \{1, 2, 4\} \cap \emptyset = \emptyset$$

$$[z_4] = [\beta \wedge \langle \text{next} \rangle z_3] = [\beta] \cap \text{PreA}(\text{next}, [z_3]) = \{1, 2, 4\} \cap \emptyset = \emptyset$$

$\gamma \models \beta$? No because initial state not in γ



Ex Model check CTL formula $\text{EF}(\text{AG}(\alpha \supset \text{AX}\beta))$ showing its translation in μ -calculus



$$\begin{aligned} [\alpha] &= [\exists x \neg \alpha] \\ [\beta] &= [Ax \alpha] \\ [\gamma] &= [\alpha \supset \beta] \\ [\delta] &= [AG(\gamma)] \\ [\eta] &= [EF(\delta)] \end{aligned}$$

$$\alpha = EX \gamma \rightarrow \langle \text{next} \rangle \gamma$$

$$[\alpha] = [\langle \text{next} \rangle \gamma] = \text{PreE}(\text{next}, [-\alpha]) = \{0, 1, 2\}$$

$$\beta = AE \alpha \rightarrow [-] \alpha$$

$$[\beta] = [-\alpha] = \text{PreA}(-, [\alpha]) = \{1, 2, 4\}$$

$$[\gamma] = [\alpha \supset \beta] = [\neg \alpha \vee \beta] = [\neg \alpha] \cup [\beta] = \{2, 3\} \cup \{1, 2, 4\} = \{1, 2, 3, 4\}$$

$$\delta = \underbrace{AG \gamma}_z = \gamma \wedge Ax \underbrace{AG \gamma}_z \rightarrow z = \gamma \wedge Ax z \rightarrow \sqrt{z} \cdot \gamma \wedge [-] z \quad \text{gfp}$$

$$[\delta] = [AG \gamma] = [\sqrt{z} \gamma \wedge [-] z] = \emptyset$$

$$[z_0] = \{0, 1, 2, 3, 4\}$$

$$[z_1] = [\gamma \wedge [-] z_0] = [\gamma \wedge \text{PreA}(-, [z_0])] = \{1, 2, 3, 4\} \cap \{0, 1, 2, 3, 4\} = \{1, 2, 3, 4\}$$

$$[z_2] = [\gamma \wedge [-] z_1] = [\gamma \wedge \text{PreA}(-, [z_1])] = \{1, 2, 3, 4\} \cap \{0, 1, 3\} = \{1, 3\}$$

$$[z_3] = [\gamma \wedge [-] z_2] = [\gamma \wedge \text{PreA}(-, [z_2])] = \{1, 2, 3, 4\} \cap \{0\} = \emptyset$$

$$[z_4] = [\gamma \wedge [-] z_3] = [\gamma \wedge \text{PreA}(-, [z_3])] = \{1, 2, 3, 4\} \cap \emptyset = \emptyset \quad \text{gfp}$$

$$\eta = EF \delta = \delta \vee EX EF \delta \rightarrow z = \gamma \vee EX z \rightarrow \mu z \delta \vee \langle \neg \rangle z \quad \text{lfp}$$

$$[\eta] = [EF \delta] = [\mu z \delta \vee \langle \neg \rangle z] = \emptyset$$

$$[z_0] = \emptyset$$

$$[z_1] = [\delta \vee \langle \neg \rangle z_0] = [\delta] \cup [\text{PreE}(-, [z_0])] = \emptyset \cup \emptyset = \emptyset \quad \text{lfp}$$

$$\alpha = \langle \neg \rangle \gamma$$

$$\beta = [-] \alpha$$

$$\gamma = \alpha \supset \beta$$

$$\delta = \sqrt{z} \gamma \wedge [-] z$$

$$\eta = \mu z \delta \vee \langle \neg \rangle z$$

$T \models \phi ?$ = Is formula true in transition system?

More precisely in state 0: $T_0 \models \phi ?$

$0 \in [\phi] = [\eta] ?$ No \rightarrow formula not true in the transition system

AUTOMATA THEORETIC LTL MODEL CHECKING

Σ^w = set of infinite words, infinite sequences of letters from alphabet Σ

Σ^* = set of finite words.

$L \subseteq \Sigma^*$ = finite language eg words ending with a

$L \subseteq \Sigma^w$ = infinite language eg words containing a finite number of a

Language recognition problem: determine whether a word w belongs to language L → Use **automata**
We say $w \in L(D) \subseteq \Sigma^*$ if the corresponding run p in the automaton D ends in a final state

Nondeterministic finite-state automaton N accepts $w \in L(N)$ if at least one run is accepting since more than one run is possible on some word.

$N = \{Q, \Sigma, I, \delta, F\}$ with:

Q = finite set of states

Σ = finite alphabet

$I \subseteq Q$ = set of initial states

$F \subseteq Q$ = set of final states

$\delta: Q \times \Sigma \rightarrow 2^Q$ = nondeterministic transition function

UNION

Given two NFA $N_1 = \{Q_1, \Sigma, I_1, \delta_1, F_1\}$ and $N_2 = \{Q_2, \Sigma, I_2, \delta_2, F_2\}$, the union automaton

$N_1 \cup N_2 = \{Q, \Sigma, I, \delta, F\}$ is defined as

$Q = Q_1 \cup Q_2$

$I = I_1 \cup I_2$

$F = F_1 \cup F_2$

$\delta(q, \sigma) = \begin{cases} \delta_1(q, \sigma) & q \in Q_1 \\ \delta_2(q, \sigma) & q \in Q_2 \end{cases}$

Intuitively, $N_1 \cup N_2$ chooses nondeterministically to execute either N_1 or N_2

Note: Q_1 and Q_2 must be disjoint

PRODUCT

In some way, $N_1 \times N_2$ is defined as

$Q = Q_1 \times Q_2$

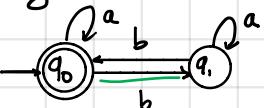
$I = I_1 \times I_2$

$F = F_1 \times F_2$

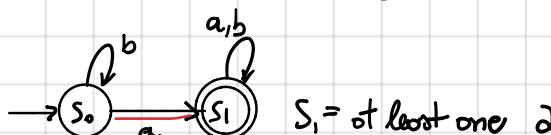
$\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$

$N_1 \times N_2$ executes N_1 and N_2 in parallel

Eg

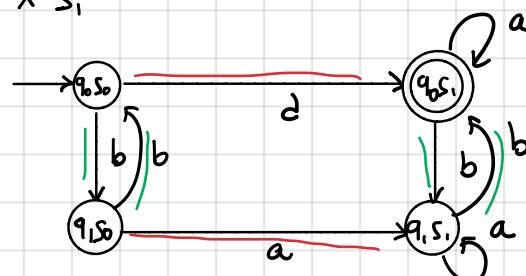


D_1 = recognizes even number of b



S_1 = at least one a

$D_1 \times S_1$



Th. For every regular expression α there exists a NFA N_α s.t. $L(\alpha) = L(N_\alpha)$ 5
For every NFA N there exists a regular expression α_N s.t. $L(\alpha_N) = L(N)$

BÜCHI AUTOMATA

Deterministic (DBA) and Nondeterministic (NBA) Büchi automata are like DFA and NFA but they read infinite words $w \in \Sigma^\omega$

As there is no lost state in the corresponding runs p , the acceptance condition is to visit a final state in F infinitely many times

Th. The language $L = \{w \in \Sigma^\omega : w \text{ contains finitely many } 2\}$ can be recognized by a NBA but not by any DBA

Th. For a given NBA N , there exists a NBA \bar{N} s.t. $L(\bar{N}) = \Sigma^\omega \setminus L(N)$

Union of two NBAs works exactly as for NFA.

Product $N_1 \otimes N_2$ called synchronous product is defined as:

$$Q = Q_1 \times Q_2 \times \{1, 2\} \quad \text{bit to flag which automaton is focusing on}$$

$$I = I_1 \times I_2 \times \{1\}$$

$$F = F_1 \times F_2 \times \{1\}$$

$$\delta((q_1, q_2, z), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma), 1) & \text{if } q_1 \notin F_1 \\ (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma), z) & \text{if } q_1 \in F_1 \end{cases}$$

$$\delta((q_1, q_2, z), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma), z) & \text{if } q_2 \notin F_2 \\ (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma), 1) & \text{if } q_2 \in F_2 \end{cases}$$

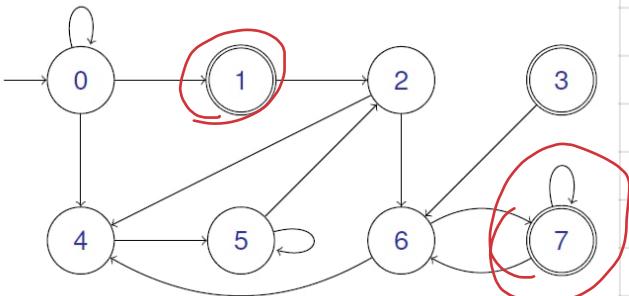
$N_1 \otimes N_2$ switches the index every time a final state is hit

A language is called ω -regular if it is the union of expressions of the form $\alpha \cdot \beta^\omega$ with α and β being regular languages

NONEMPTYNESS

Given an NFA N decide whether $L(N) \neq \emptyset \Rightarrow$ Does a word w accepted by N exist?
 ↳ w is accepted by N iff exists a run whose path starts from 0 and ends in a final state
 \Rightarrow Nonemptiness \Rightarrow Reachability

REACHABILITY WITH FIX-POINT THEORY



$$\begin{aligned} F &= \{1, 7\} \subseteq Q \\ Z &= F \vee \text{next } Z \quad (= \text{reachability}) \\ \downarrow \\ \mu Z. (F \vee \text{next } Z) &= \text{Reach}(F) \end{aligned}$$

Given a NBA instead we not only have to find a path from 0 to final state but must visit the final state infinitely many times. \rightarrow Recurrent reachability.

Using fixpoint theory:

$$\begin{aligned} Y &= \text{Reach}(F \wedge \text{next } Y) \\ \nu Y. (\text{Reach}(F \wedge \text{next } Y)) &\rightarrow \nu Y. (\mu Z ((F \wedge \text{next } Y) \vee \text{next } Z)) \end{aligned}$$



[A Generalized NBA (GNBA) is like a normal NBA except that $F = (F_1, F_2, \dots, F_n)$
 A run ρ in N is accepting iff it visits every F_i infinitely often]

Th. For an LTL formula φ we can construct a GNBA $N\varphi$ s.t $\mathcal{L}(N\varphi) = \mathcal{L}(\varphi)$
 To do this we need:

Fischer-Ladner Closure: for a given LTL formula φ , the FS-closure of φ denoted $cl(\varphi)$ is the set of subformulas of φ and their negations. Defined as follow:

- $\varphi \in cl(\varphi)$
- if $\psi \in cl(\varphi)$ then $\neg\psi \in cl(\varphi)$
- if $\psi_1, \psi_2 \in cl(\varphi)$ then $\psi_1, \psi_2 \in cl(\varphi)$
- if $X\psi \in cl(\varphi)$ then $\psi \in cl(\varphi)$
- if $\psi_1, \psi_2 \in cl(\varphi)$ then $\psi_1, \psi_2 \in cl(\varphi)$

eg $\varphi = p \wedge ((x_p) \vee q)$

$$cl(\varphi) = \{p \wedge ((x_p) \vee q), \neg(p \wedge ((x_p) \vee q)), p, \neg p, (x_p) \vee q, \neg((x_p) \vee q), x_p, \neg x_p, q, \neg q\}$$

A set $\alpha \subseteq cl(\varphi)$ is called atom if it is maximally consistent, that is:

- $\forall \psi \in cl(\varphi)$ either $\psi \in \alpha \vee \neg\psi \in \alpha$
- $\psi_1, \psi_2 \in \alpha \iff \psi_1, \psi_2 \in \alpha$

eg $\varphi = p \cup q$ $\mathcal{L}(\varphi) = \{p \cup q, \neg(p \cup q), p, \neg p, q, \neg q\}$

→ Atoms: $\alpha_1 = \{p \cup q, p, q\}$ $\alpha_2 = \{p \cup q, p, \neg q\}$ $\alpha_3 = \{p \cup q, \neg p, q\}$ $\alpha_4 = \{p \cup q, \neg p, \neg q\}$
 $\alpha_5 = \{\neg(p \cup q), p, q\}$ $\alpha_6 = \{\neg(p \cup q), p, \neg q\}$ $\alpha_7 = \{\neg(p \cup q), \neg p, q\}$ $\alpha_8 = \{\neg(p \cup q), \neg p, \neg q\}$

A state α (eg α_1) gives information on which subformulas of φ need to be satisfied when computation starts from α itself (for α_1 , $p \cup q \wedge p \wedge q$ must be true)

Every atom containing φ is an initial state ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$)

To move from atom α to α' ($\sigma \in \Sigma = 2^{\text{Prop}}$) i.e. $\alpha' \in \mathcal{E}(\alpha, \sigma)$ we have to check:

① $\sigma = \alpha \cap \text{Prop}$ → move only if you read something consistent

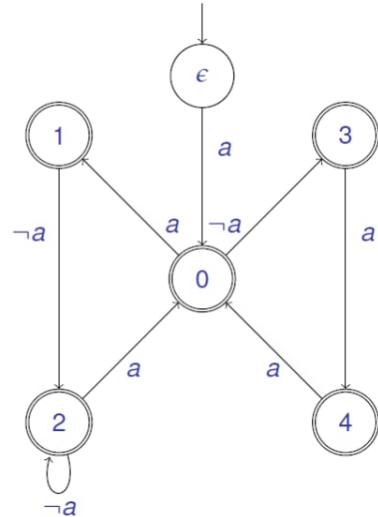
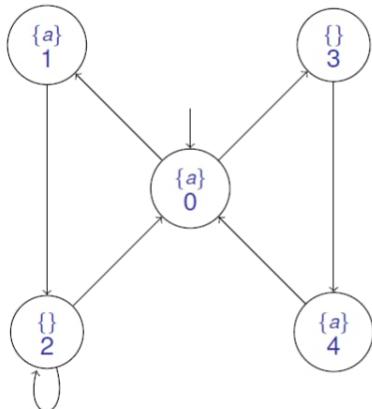
② $X\psi \in \alpha$ iff $\psi \in \alpha'$ → from state with $X\psi$ we can reach only states with ψ

③ $\psi_1 \vee \psi_2 \in \alpha$ iff either $\psi_1 \in \alpha$ or both $\psi_1 \in \alpha$ and $\psi_1 \vee \psi_2 \in \alpha'$

↳ in this case we can move wherever we want because we "verified" α

Final States: every subformula $\psi_1 \vee \psi_2$ in α holds on acceptance and thus contributes to F with the set $F_{\psi_1 \vee \psi_2} = \{\alpha \in Q : \psi_2 \in \alpha \text{ or } \neg(\psi_1 \vee \psi_2) \in \alpha\}$

FROM LABELED TRANSITION SYSTEM TO NBA



For a given LTS Γ and an LTL formula φ , Model Checking is the problem of verifying that all executions of Γ satisfy φ . $\Rightarrow \Gamma \models \varphi \iff \mathcal{L}(\Gamma) \subseteq \mathcal{L}(\varphi)$

$$\cdot \Gamma \rightarrow N_\Gamma \quad \mathcal{L}(\Gamma) = \mathcal{L}(N_\Gamma)$$

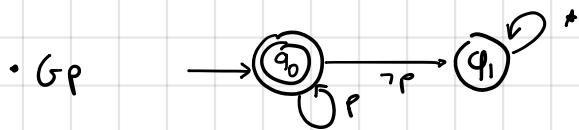
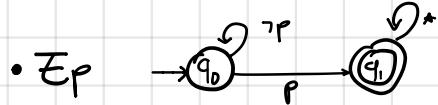
$$\cdot \varphi \rightarrow N_\varphi \quad \mathcal{L}(\varphi) = \mathcal{L}(N_\varphi)$$

$$\cdot \mathcal{L}(\Gamma) \subseteq \mathcal{L}(\varphi) \iff \mathcal{L}(N_\Gamma) \subseteq \mathcal{L}(N_\varphi) \iff \mathcal{L}(N_\Gamma) \cap \mathcal{L}(\overline{N}_\varphi) = \emptyset$$

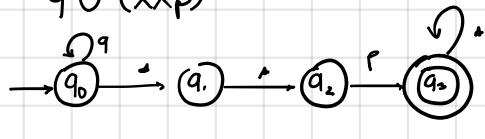
$$\rightarrow \mathcal{L}(\overline{N}_\varphi) = \mathcal{L}(N_\varphi)$$

! Can we avoid the need of complement of a NBA?

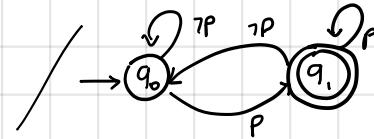
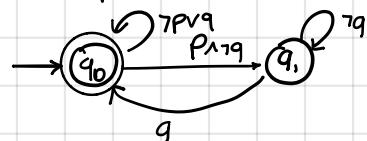
Ex From LTL to (G) NBA



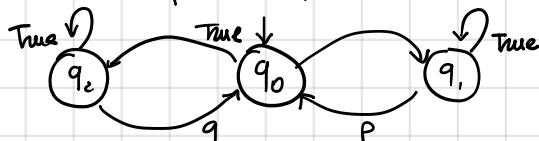
• $q \vee (xx_p)$



• $G(p \rightarrow Fq)$



• $GFp \wedge GFq$



$$F_1 = \{q_1\}$$

$$F_2 = \{q_2\}$$

LTL model checking algorithms takes :

- a model T
- a formula φ
- and returns
- YES if $T \models \varphi$
- NO and a counter example if $T \not\models \varphi$

Consider a model T and an LTL property φ
 $T \models \varphi$ if \forall paths π of T it holds that $\pi \models \varphi$ namely
if $\pi \in \mathcal{L}(\varphi)$
 $\rightarrow T \models \varphi \Leftrightarrow \mathcal{L}(T) \subseteq \mathcal{L}(\varphi)$
 $\Leftrightarrow \mathcal{L}(T) \cap \overline{\mathcal{L}(\varphi)} = \emptyset$
 $\Leftrightarrow \mathcal{L}(T) \cap \mathcal{L}(\neg \varphi) = \emptyset$

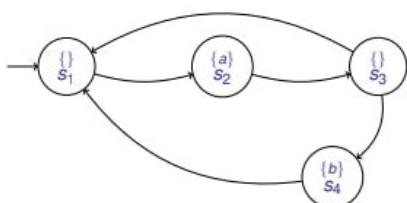
① construct N_T automaton from LTS

② construct $N_{\neg \varphi}$ automaton from LTL formula

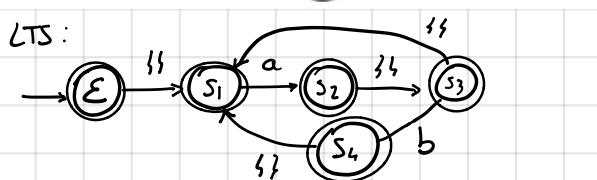
③ construct product automaton $N_{T, \neg \varphi} = N_T \otimes N_{\neg \varphi}$ } YES if $\mathcal{L}(N_{T, \neg \varphi}) = \emptyset$

④ Solve nonemptiness problem $\mathcal{L}(N_{T, \neg \varphi}) \neq \emptyset$ } NO otherwise

Ex.



$$Xa \wedge (G(b \rightarrow Xa)) \wedge Fa$$

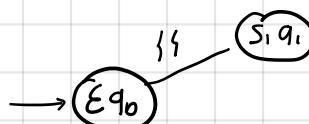


LTL : $\varphi = Xa \wedge (G(b \rightarrow Xa)) \wedge Fa$
 $\rightarrow \neg \varphi = \neg \varphi_1 \vee \neg \varphi_2 \vee \neg \varphi_3 = \neg Xa \vee \neg G(b \rightarrow Xa) \vee \neg Fa$
 $= X \neg a \vee (b \wedge X \neg a) \vee G \neg a$

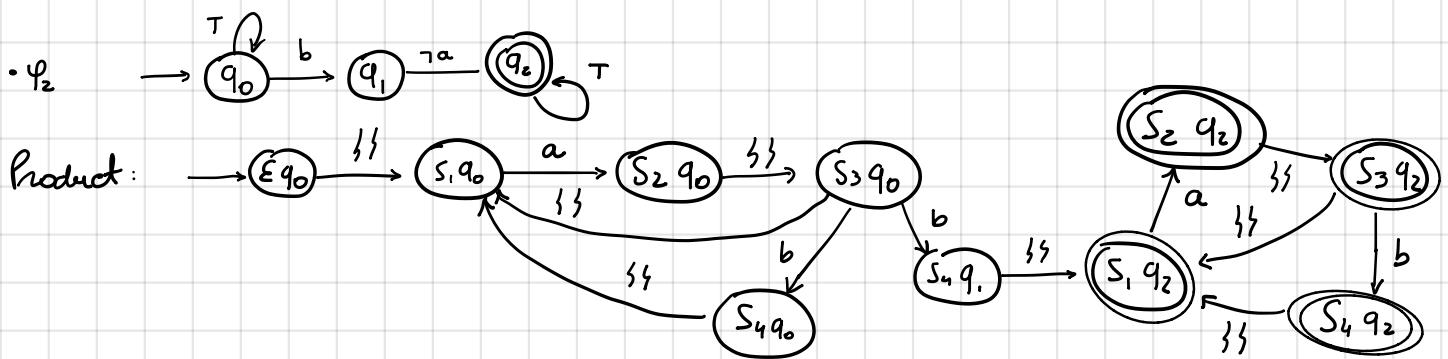
We just need one of φ_1, φ_2 and φ_3 to be "true"



Product (LTS always accepting !) :



From S1 we move only with "a"
From q1 we move only with "\neg a"
 \rightarrow STUCK \rightarrow empty



$$B = \{S_1 q_2, (S_2 q_2), (S_3 q_2), (S_4 q_2)\}$$

Recurrent Reachability:

$$\text{Buchi}(B) = \bigvee Y (\text{Reach}(B \wedge \text{next}^* Y)) \\ = \bigvee Y \mu Z ((B \wedge \text{next}^* Y) \vee \text{next}^* Z)$$

$$[\text{Reach}(F) = \mu Z (F \vee \text{next}^* Z)]$$

Y_0 = all states

$Z_0 = \emptyset$

$$Z_1 = (B \wedge \text{next}^* Y_0) \cup \text{next}^* Z_0 = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2\} \cup \{\text{all states} \setminus E q_0\} = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2\}$$

$$Z_2 = (B \wedge \text{next}^* Y_1) \cup \text{next}^* Z_1 = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2\} \cup \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2, S_1 q_1\} = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2, S_1 q_1\}$$

$$Z_3 = (B \wedge \text{next}^* Y_2) \cup \text{next}^* Z_2 = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2\} \cup \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2, S_1 q_1, S_3 q_0\} \\ = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2, S_1 q_1, S_3 q_0\}$$

$$Z_4 = (B \wedge \text{next}^* Y_3) \cup \text{next}^* Z_3 = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2, S_1 q_1, S_3 q_0, S_2 q_0\}$$

$$Z_5 = (B \wedge \text{next}^* Y_4) \cup \text{next}^* Z_4 = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2, S_1 q_1, S_3 q_0, S_2 q_0, S_1 q_0\}$$

$$Z_6 = (B \wedge \text{next}^* Y_5) \cup \text{next}^* Z_5 = \{S_1 q_2, S_2 q_2, S_3 q_2, S_4 q_2, S_1 q_1, S_3 q_0, S_2 q_0, S_1 q_0, E q_0, S_4 q_0\}$$

$$Z_7 = (B \wedge \text{next}^* Y_6) \cup \text{next}^* Z_6 = Z_6$$

L.f.p.

$$Y_1 = Y_0 \quad g.f.p.$$

$\Rightarrow G(b \rightarrow x_a)$ is false in LTS $\rightarrow \varphi$ not satisfied in LTS

Ex Check whether CQ q_1 is contained in CQ q_2 reporting canonical DBs and homomorphism

$q_1 \leftarrow \text{edge}(r, g), \text{edge}(g, b), \text{edge}(b, r)$

$q_2 \leftarrow \text{edge}(x, y), \text{edge}(y, z), \text{edge}(z, v), \text{edge}(v, w), \text{edge}(w, z)$

There are no free variable \rightarrow no need to freeze them!

I_{q_1} = canonical DB of q_1

e	
r	g
g	b
b	r

For second query we need to given assignments for existential variables

α

We start from $\alpha(x)$ and we assign to it (at random) r

$$\alpha(x) = r$$

$$\rightarrow \text{edge}(x, y) = \text{edge}(r, y)$$

we check on canonical DB of q_1 a predicate starting with r and "guess" $\alpha(y)$: only row starting with r has a g at second place

$$\alpha(y) = g$$

We do the same for $\text{edge}(y, z)$. $\alpha(y) = g$, there is only [g b] $\rightarrow g(z) = b$

$\text{edge}(z, x) \rightarrow \text{edge}(b, r)$ that is in canonical DB ✓

$\text{edge}(z, v)$, $\alpha(z) = b \rightarrow g(v) = r$

$\text{edge}(v, w)$, $\alpha(v) = r \rightarrow g(w) = g$

$\text{edge}(w, z) \rightarrow \text{edge}(g, b)$ is in canonical DB ✓

Satisfy assignment of existential variables.

I_{q_2}

e	
x	y
y	z
z	x
z	v
v	w
w	z

Finding homomorphism means extend assignment to constants. But in our case there aren't.

So? Nothing to do

$$h = \hat{\alpha} = \alpha$$

└ α on constants

To check homomorphism we need to check that

$$h(c^{I_{q_1}}) = h(c^{I_{q_2}}) \wedge \text{constants}$$

but since we don't have ... is true

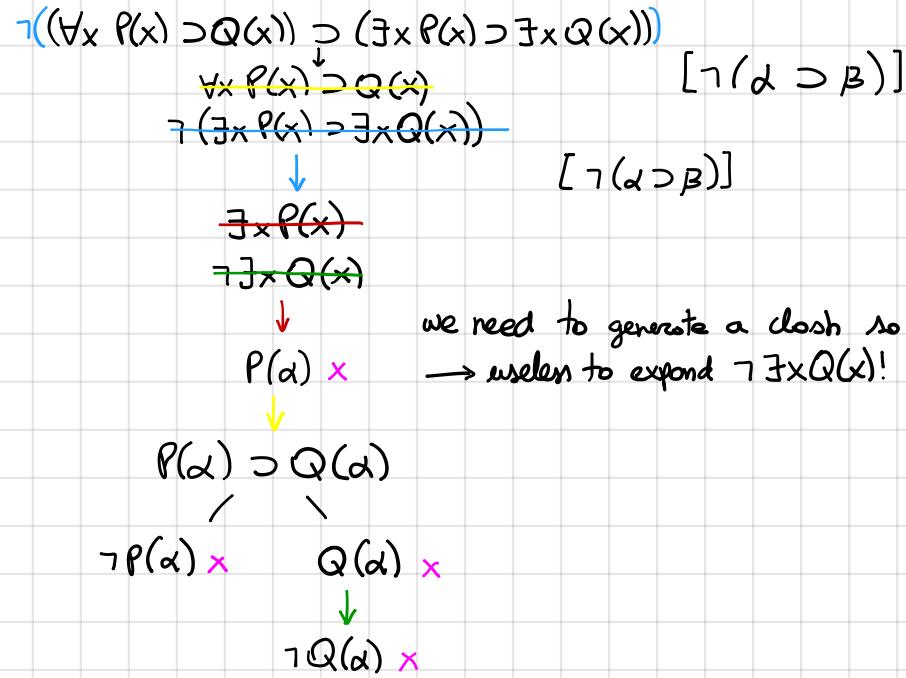
$(x, y) \in e^{I_{q_2}} \Rightarrow h(x), h(y) \in e^{I_{q_1}}$ how to check? Simply use assignments: $h(x), h(y) \rightarrow [r g]$
 $[r g]$ is $e^{I_{q_1}}$ ✓

Do so every entry in $e^{I_{q_2}}$. They should be all true, mistakes otherwise.

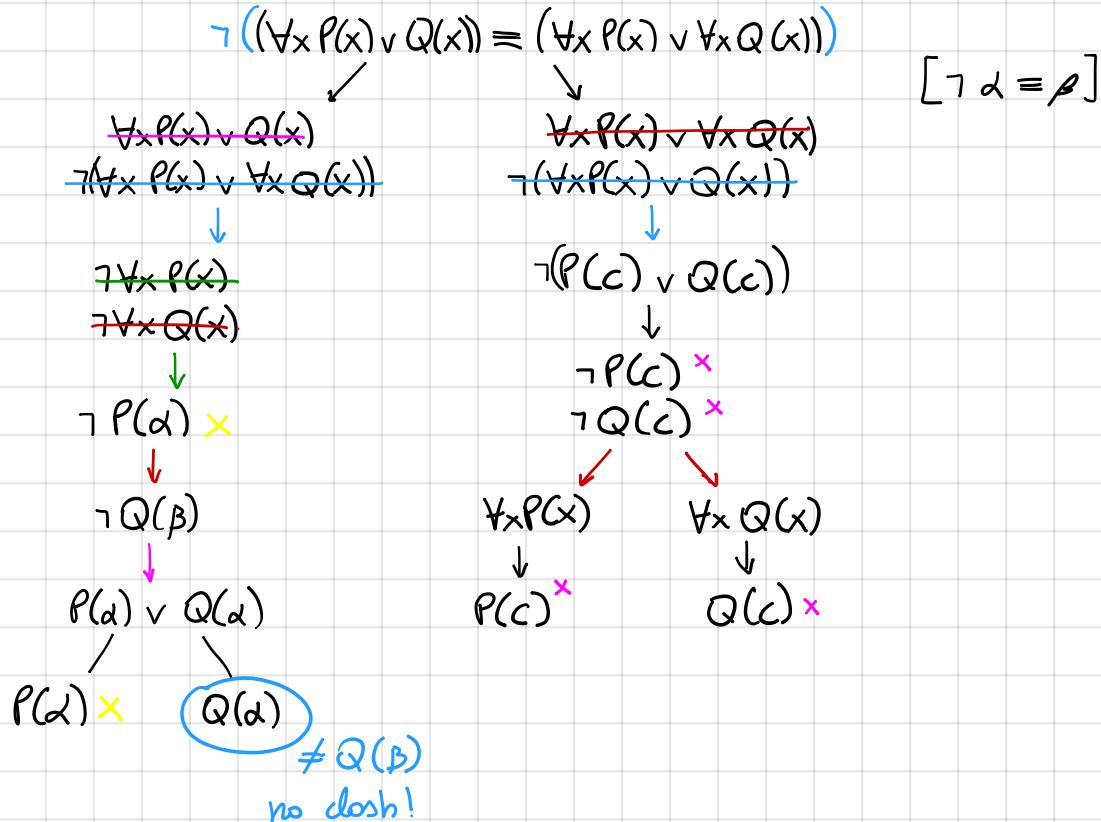
Ex Check whether the following FOL formula is valid using tableau:

$$(\forall x P(x) \supset Q(x)) \supset (\exists x P(x) \supset \exists x Q(x))$$

Tableaux can only solve satisfiability problems, not validity
→ negate the formula and check unsatisfiability



Ex Check if valid using tableau and if not exhibit an interpretation that is a counter example

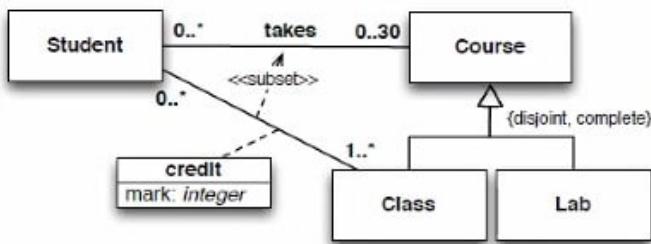


$$\Delta^I = \text{all constants} = \{\alpha, \beta\}$$

$$P^I = \{\beta\}$$

$$Q^I = \{\alpha\}$$

Exercise 1. Express the following UML class diagram in FOL.



Exercise 2. Consider the above UML class diagram and the following (partial) instantiation.

Student	Class	Lab	credit/mark	takes
peter paul mary jane	calculus AI FM algorithms	IoT lab db lab hacking lab	peter algorithm 30 paul calculus 27 mary algorithms 28 mary AI 30 jane FM 30 jane algorithms 30	peter IoT lab paul IoT lab mary FM jane db lab jane hacking lab jane IoT lab

1. Check whether the instantiation (once completed) is correct (and explain why it is or it is not).
2. Express in FOL and evaluate the following queries:
 - (a) Return students that have taken at least 3 courses.
 - (b) Return students that have taken only classes.
 - (c) Check if there exists a student that has taken all labs.
 - (d) Check if there is a student that has taken all classes, but not for credit.

①

$$\begin{aligned}
 & St(x), Course(x), Clas(x), Lob(x), takes(x,y), credit(x,y), mark(x,y,z), int(x) \\
 & \forall x, y \ takes(x,y) \supset St(x) \wedge Course(y) \quad \boxed{\text{takes}} \\
 & \forall x \ St(x) \supset \#\{y | takes(x,y)\} \leq 30 \quad \boxed{1 \leq \dots \leq 30 \text{ to be more precise, but logic will do for us.}} \\
 & \forall x, y \ credit(x,y) \supset St(x) \wedge clas(y) \quad \boxed{\text{credit}} \\
 & \forall x \ St(x) \supset 1 \leq \#\{y | credit(x,y)\} \\
 & \forall x, y \ credit(x,y) \supset takes(x,y) \\
 & \forall x, y, z \ mark(x,y,z) \supset credit(x,y) \wedge int(z) \quad \boxed{\text{mark}} \\
 & \forall x, y \ credit(x,y) \supset \exists z \ mark(x,y,z) \\
 & \forall x, y \ credit(x,y) \supset \forall z, z' \ mark(x,y,z) \wedge mark(x,y,z') \supset z = z' \\
 & \forall x \ Clas(x) \supset Course(x) \\
 & \forall x \ Lob(x) \supset Course(x) \\
 & \forall x \ Clas(x) \supset \neg Lob(x) \quad [\text{OR viceversa}] \quad \boxed{\text{disjoint}} \\
 & \forall x \ Course(x) \supset (Clas(x) \vee Lob(x)) \quad \boxed{\text{complete}}
 \end{aligned}$$

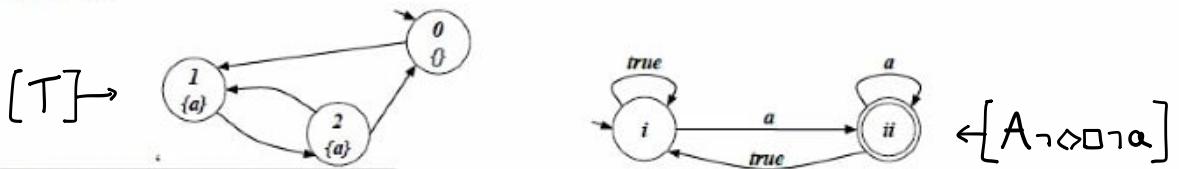
②

1. Use chase algorithm (apply/check ISA on instantiation 1 time more than the max length of ISA chain in diagram (1 in our case))
Instantiation is correct (every clas and lob is also a course, create the table)

2.

- $\exists y_1, y_2, y_3 \ St(x) \wedge takes(x, y_1) \wedge takes(x, y_2) \wedge takes(x, y_3) \wedge y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge y_2 \neq y_3$
- $St(x) \wedge \forall y \ Lob(y) \supset takes(x, y)$
- $\exists x \ St(x) \wedge \forall y \ Clas(y) \supset (takes(x, y) \wedge \neg credit(x, y))$

Exercise 6 (optional). Model check the LTL formula $\diamond \square \neg a$ against the following transition system, by considering that the Büchi automaton for $\neg(\diamond \square \neg a)$ is the one below:



¹The student can get the maximum grade even without doing Exercise 6.

\forall traces of $T \supset$ traces satisfy $\diamond \square \neg a$

$\forall t \in \mathcal{L}(A_T) \Rightarrow t \in \mathcal{L}(A_{\neg \diamond \square \neg a})$

$\hookrightarrow \exists t \in \mathcal{L}(A_T) \wedge t \notin \mathcal{L}(A_{\neg \diamond \square \neg a})$

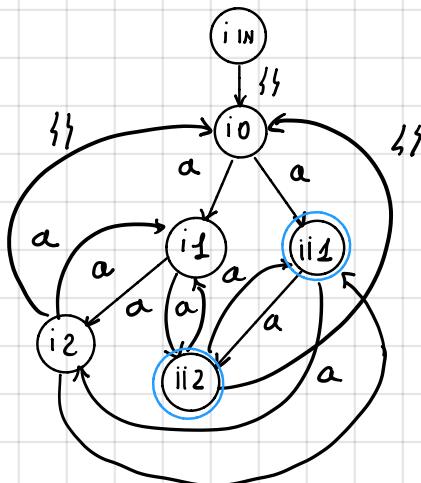
→ find complementary $A_{\diamond \square \neg a}$

We must check emptiness of $A_T \wedge A_{\neg \diamond \square \neg a}$

$\exists t \in \mathcal{L}(A_T) \wedge t \in \mathcal{L}(A_{\neg \diamond \square \neg a})$

check emptiness of $A_T \wedge A_{\neg \diamond \square \neg a}$

$A_T \wedge A_{\neg \phi}$



Non emptiness:

$$\vee X \mu Y (\text{Acc} \wedge \leftrightarrow X) \vee \leftrightarrow Y$$

$X_0 = \text{All states}$

$$X_1 = \mu Y (\text{Acc} \wedge \leftrightarrow X_0) \vee \leftrightarrow Y \quad - \text{All states}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = ([\text{Acc}] \cap \text{PreE}(-, [X_0])) \cup \text{PreE}(-, [Y_{10}])$$

$$= \{ii_1, ii_2\} \cup \emptyset = \{ii_1, ii_2\}$$

$$[Y_{12}] = \{ii_1, ii_2\} \cup \text{PreE}(-, [Y_{11}])$$

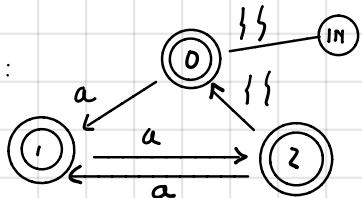
$$< \{ii_1, ii_2\} \cup \{i_0, i_1, i_2, ii_1, ii_2\} = \{i_0, i_1, i_2, ii_1, ii_2\}$$

$$[Y_{13}] = \{ii_1, ii_2\} \cup \text{PreE}(-, [Y_{12}])$$

$$= \text{All states least fixpoint}$$

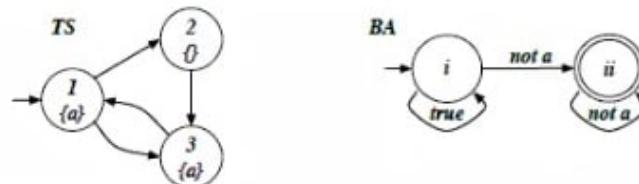
→ greatest fixpoint

$A_T :$



Not empty! $\Rightarrow \exists t. t$ is generated by $T \wedge t \models \neg(\square \neg a)$
 $\Rightarrow T \not\models \phi$

Exercise 5. Consider the transition system TS below. Model check the LTL formula $\square \diamond a$, by considering that the Büchi automaton BA for $\neg \square \diamond a$ (i.e., $\diamond \square \neg a$) is the one below:



$$\forall t. t \in T \supseteq t \models \phi$$

$$L(T) \subseteq L(\phi)$$

$$L(T) \cap L(\neg \phi) = \emptyset$$

$$L(A_T) \cap L(A_{\neg \phi}) = \emptyset$$

$$L(A_T \wedge A_{\neg \phi}) = \emptyset$$

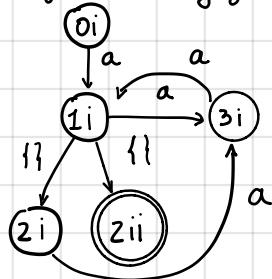
① Build A_T :

- all states are final
- add "input node" (0)
- move "actions" on edges (a and $\exists i$)



② Build $A_T \wedge A_{\neg \phi}$ → given:

- run both automata from "input" and "i" ($(0i)$)
- final state if final in both automata



③ Check non-emptiness using "magic formula"

→ final state

$$\vee X \mu Y (Acc \wedge c \rightarrow X) \vee c \rightarrow Y$$

$$[X_0] = S$$

$$[X_1] = \mu Y (Acc \wedge c \rightarrow X) \vee c \rightarrow Y = \emptyset \text{ greatest fixpoint}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = ([Acc] \wedge \text{PreE}(-, [X_0])) \cup \text{PreE}(-, [Y_{10}])$$

$$= (\{zii\} \cap \{0i, 1i, 2ii, 3i\}) \cup \emptyset = \emptyset \text{ least fixpoint}$$

$$\text{Empty! } \Rightarrow \exists t \in T \wedge t \models \neg \square \diamond a \Rightarrow T \models \phi$$

General rule: if initial state ($0i$)

- is NOT in fixpoint $\rightarrow T \models \phi$ ($L(N_{T,\neg \phi}) = \emptyset$)
- is in fixpoint $\rightarrow T \not\models \phi$ ($L(N_{T,\neg \phi}) \neq \emptyset$)

RECAP RULES & μ -calculus CONVERSATIONS

$$\alpha \text{ rules}$$

$\frac{\phi \wedge \psi}{\phi}$	$\frac{\neg(\phi \vee \psi)}{\neg\phi}$	$\frac{\neg(\phi \supset \psi)}{\phi}$
ψ	$\neg\psi$	$\neg\psi$

$$\neg\neg\text{-Elimination}$$

$$\frac{\neg\neg\phi}{\phi}$$

$$\beta \text{ rules}$$

$$\frac{\phi \vee \psi}{\phi \mid \psi}$$

$$\frac{\neg(\phi \wedge \psi)}{\neg\phi \mid \neg\psi}$$

$$\frac{\phi \supset \psi}{\neg\phi \mid \psi}$$

$$\text{Branch Closure}$$

$$\frac{\phi}{\begin{matrix} \neg\phi \\ X \end{matrix}}$$

$$\frac{\Phi \equiv \Psi}{\begin{matrix} \Phi \mid \neg\Phi \\ \Psi \mid \neg\Psi \end{matrix}}$$

$$\frac{\neg(\Phi \equiv \Psi)}{\begin{matrix} \Phi \mid \neg\Phi \\ \neg\Psi \mid \Psi \end{matrix}}$$

$$\frac{\exists x \ \phi(x)}{\phi(c)}$$

$$\frac{\neg \forall x \ \phi(x)}{\neg \phi(c)}$$

c = fresh constant = new constant

not previously appearing in tableau

$$\frac{\neg \exists x \ \phi(x)}{\neg \phi(t)}$$

$$\frac{\forall x \ \phi(x)}{\phi(t)}$$

t - only term = not fresh

CTL \rightarrow μ -calculus

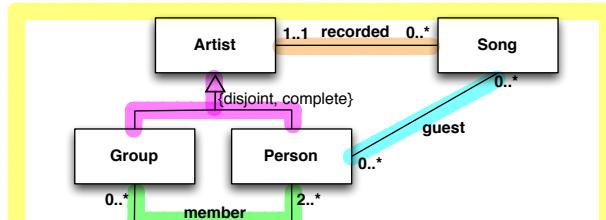
φ	ρ
$\vee, \wedge, \rightarrow$	$\vee, \wedge, \rightarrow$
$\exists x \varphi$	$\langle \text{next} \rangle t(\varphi)$
$\forall x \varphi$	$[\text{next}] t(\varphi)$
$\exists F \varphi$	$\mu z \ t(\varphi) \vee \langle \text{next} \rangle z$
$\forall F \varphi$	$\mu z \ t(\varphi) \vee [\text{next}] z$
$\exists G \varphi$	$\nu z \ t(\varphi) \wedge \langle \text{next} \rangle z$
$\forall G \varphi$	$\nu z \ t(\varphi) \wedge [\text{next}] z$
$\varphi \exists U \psi$	$\mu z \ t(\psi) \wedge (t(\varphi) \wedge \langle \text{next} \rangle z)$
$\varphi \forall U \psi$	$\mu z \ t(\psi) \vee (t(\varphi) \wedge [\text{next}] z)$

E \rightarrow has always $\langle - \rangle$
 A \rightarrow has always $[-]$

F \rightarrow $\mu \dots \vee \dots$
 G \rightarrow $\nu \dots \wedge \dots$

Note : $A \supset B \rightarrow \neg A \vee B$

Exercise 1. Express the following UML class diagram in *FOL*.

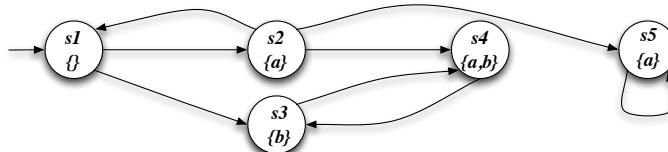


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation.

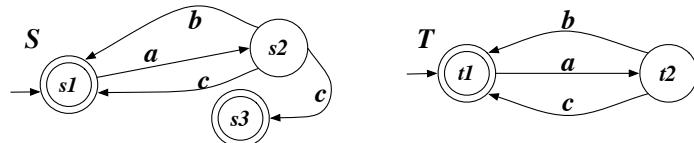
	Person		member		Song	
<i>Group</i>	John	Be	John	Be	I wanna be your man (original)	
	Paul	RS	Paul	Be	I wanna be your man (cover)	
	George		George	Be		
	Ringo		Ringo	Be		
	Mick		Mick	RS		
	Keith		Keith	RS		
<i>recorded</i>	RS	I wanna be your man (original)	John	I wanna be your man (original)		
	Be	I wanna be your man (cover)	Paul	I wanna be your man (original)		
<i>guest</i>						

1. Check whether the above instantiation, once completed, is correct, and explain why it is or it is not.
2. Express in *FOL* and evaluate the following queries:
 - (a) Return groups to with more than 3 members.
 - (b) Return person that are guest of all songs that they, or group they are member of, did not recorded.
 - (c) Check whether there are no songs recoded by groups whose members also participated as guests to the song.

Exercise 3. Model check the Mu-Calculus formula $\nu X.\mu Y.((b \wedge \langle \text{next} \rangle X) \vee \langle \text{next} \rangle Y)$ and the CTL formula $EG(AX(\neg a \vee AFb))$ (showing its translation in Mu-Calculus) against the following transition system:



Exercise 4. Consider the following two transition systems:



Write the definition of bisimilarity and compute the bisimilarity relation for the two transition systems.

Exercise 5. Check whether the following Hoare triple is correct, using as *invariant* $i \leq 10$.

{ $i=0$ } while($i < 10$) do $i := i + 1$ { $i = 10$ }

1) ALPHABET: $A(x)$, $G(x)$, $P(x)$, $S(x)$, $\text{member}(x,y)$, $\text{guest}(x,y)$, $\text{recorded}(x,y)$

ISA: $\forall x A(x) \supset G(x) \vee P(x)$

$\forall x G(x) \supset \neg P(x)$

$\forall x G(x) \supset A(x)$

$\forall x P(x) \supset A(x)$

member: $\forall xy \text{ member}(x,y) \supset G(x) \wedge P(y)$

$\forall x G(x) \supset z \in \#\{y \mid \text{member}(x,y)\}$

guest: $\forall xy \text{ guest}(x,y) \supset P(x) \wedge S(y)$

recorded: $\forall xy \text{ recorded}(x,y) \supset A(x) \wedge S(y)$

$\forall y S(y) \supset 1 \leq \#\{x \mid \text{recorded}(x,y)\} \leq 1$

2) 1. To check if instantiation is correct, we apply chose algorithm until we reach a fixpoint (no more modifications possible)

$I_0 = \emptyset$

$I_1 = I$

$I_2 = I_1 \cup \{\text{group} \cup \text{person}\} = \text{create Artist table}$

$I_3 = I_2 \cup \emptyset = I_2 \quad \text{fixpoint (complete instantiation)}$

The complete instantiation is correct because all axioms are made true.

2. $\neg \exists yy'y'' \text{ member}(x,y) \wedge \text{member}(x,y') \wedge \text{member}(x,y'') \wedge \neg(y=y') \wedge \neg(y'=y'') \wedge \neg(y=y'')$
 $\Rightarrow \{Be\}$
- $P(x) \wedge \forall y \text{ guest}(x,y) \supset \neg \text{recorded}(x,y) \vee (\forall z \text{ member}(z,x) \supset \text{recorded}(z,y))$
 $\Rightarrow \{\text{Paul, John}\}$
- $\neg \exists xyz \text{ recorded}(x,y) \wedge G(x) \supset \exists z \text{ member}(x,z) \wedge \text{guest}(z,y)$
 $\Rightarrow \{\emptyset\}$

4) Two transition systems are bisimilar if:

- locally they look equal
- each action that can be done on one can also be done on the other.

$R_0 = \text{cartesian product} = \{(s_i, t_i)(s_j, t_j) | (s_i, t_i) \in S_1 \wedge (s_j, t_j) \in S_2\}$

$R_1 = \text{remove pairs that violate local conditions on final states}$

$= \{(s_i, t_i)(s_j, t_j) | (s_j, t_j) \in F_2\}$

$R_2 = \text{remove pairs that allow action only on one of the two states}$

$= \{(s_i, t_i)(s_j, t_j) | (s_i, t_i) \in F_1\}$

$R_3 = \text{remove pairs that lead to pairs no more in the list}$

$= \{(s_i, t_i)(s_j, t_j) | (s_j, t_j) \in R_2\}$

(s_i, t_i) belongs to gfp $\rightarrow S$ and T are bisimilar

$$\begin{aligned} 5) \quad P &= \{i = 0\} \\ Q &= \{i = 10\} \\ \delta &= \{i = i + 1\} \\ G &= \{i < 10\} \\ I &= \{i \leq 10\} \end{aligned}$$

- Check if $P \supseteq I$

$$i = 0 \supseteq i \leq 10 \quad \checkmark$$

- Check $\neg G \wedge I \supseteq Q$

$$i \leq 10 \wedge i \leq 10 \supseteq i = 10 \quad \checkmark \quad (i = 10)$$

- Check $G \wedge I \supseteq \text{Wp}(\delta, I)$

$$\begin{aligned} i < 10 \wedge i \leq 10 \supseteq & \quad i \leq 9 \quad \checkmark \quad (i = 9) \\ & i = i + 1 \\ & i \leq 10 \end{aligned}$$

I is an invariant \rightarrow the Hoare triple is correct!

$$3) \quad \vee X \mu Y ((b \wedge c \rightarrow X) \vee c \rightarrow Y)$$

$$[X_0] = S$$

$$[X_1] = \mu Y ((b \wedge c \rightarrow X) \vee c \rightarrow Y) = \{1, 2, 3, 4\}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = ([b] \cap \text{PreE}(-, [X_0])) \cup \text{PreE}(-, [Y_{10}])$$

$$= (\{3, 4\} \cap S) \cup \emptyset = \{3, 4\}$$

$$[Y_{12}] = ([b] \cap \text{PreE}(-, [X_0])) \cup \text{PreE}(-, [Y_{11}])$$

$$= \{3, 4\} \cup \{1, 2, 3, 4\} = \{1, 2, 3, 4\}$$

$$[Y_{13}] = ([b] \cap \text{PreE}(-, [X_0])) \cup \text{PreE}(-, [Y_{12}]) \quad \boxed{\text{least fixpoint}}$$

$$= \{3, 4\} \cup \{1, 2, 3, 4\} = \{1, 2, 3, 4\}$$

$$[X_2] = \mu Y ((b \wedge c \rightarrow X) \vee c \rightarrow Y) = \{1, 2, 3, 4\} \quad \text{greatest fixpoint}$$

$$[Y_{20}] = \emptyset$$

$$[Y_{21}] = ([b] \cap \text{PreE}(-, [X_2])) \cup \text{PreE}(-, [Y_{20}])$$

$$= (\{3, 4\} \cap \{1, 2, 3, 4\}) \cup \emptyset = \{3, 4\}$$

$$[Y_{22}] = ([b] \cap \text{PreE}(-, [X_2])) \cup \text{PreE}(-, [Y_{21}])$$

$$= \{3, 4\} \cup \{1, 2, 3, 4\} = \{1, 2, 3, 4\}$$

$$[Y_{23}] = ([b] \cap \text{PreE}(-, [X_2])) \cup \text{PreE}(-, [Y_{22}]) \quad \boxed{\text{least fixpoint}}$$

$$= \{3, 4\} \cup \{1, 2, 3, 4\} = \{1, 2, 3, 4\}$$

Initial state in solution (greatest fixpoint) $\rightarrow \mathcal{T} \models \phi$

$$\mathcal{E}G(\text{AX}(\overbrace{\neg a \vee \underline{AFb}}^{\beta}) \underbrace{\alpha}_{\gamma})$$

$$\alpha = AFb = \mu X b \vee \neg X = \{3, 4\}$$

$$[X_0] = \emptyset$$

$$[X_1] = [b] \cup \text{PreA}(-, [X_0])$$

$$= \{3, 4\} \cup \emptyset = \{3, 4\}$$

$$[X_2] = [b] \cup \text{PreA}(-, [X_1])$$

$$= \{3, 4\} \cup \{3, 4\} = \{3, 4\}$$

least fixpoint

$$\beta = \neg a \vee \alpha = \{1, 3, 4\}$$

$$= \{1, 3\} \cup \{3, 4\} = \{1, 3, 4\}$$

$$\gamma = AX\beta = \neg X\beta = \text{PreA}(-, [\beta]) = \{3, 4\}$$

$$\mathcal{E}G\gamma = \vee X \gamma \wedge \neg X$$

$$[X_0] = S$$

$$[X_1] = [\gamma] \cap \text{PreE}(-, [X_0])$$

$$= \{3, 4\} \cap S = \{3, 4\}$$

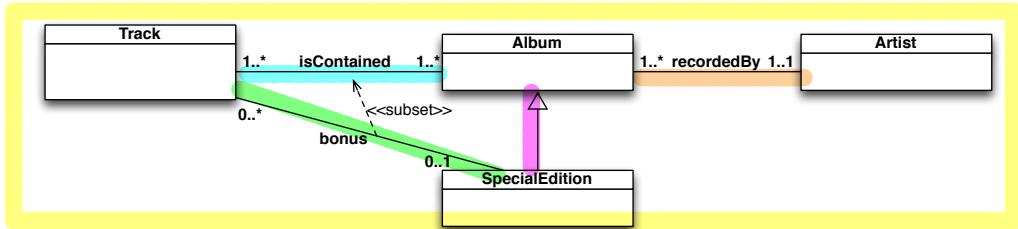
$$[X_2] = [\gamma] \cap \text{PreE}(-, [X_1])$$

$$= \{3, 4\} \cap \{1, 2, 3, 4\} = \{3, 4\}$$

greatest fixpoint

Initial states doesn't belong to solution (greatest fixpoint) $\rightarrow \mathcal{T} \not\models \phi$

Exercise 1. Express the following UML class diagram in *FOL*.

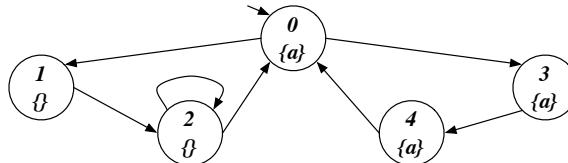


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation.

Track	Album	SpEd	Artist	isContained	bonus	recordedBy
t1 t2 t3 t4 t5 t6	a1 a2 a3	s1 s2	bt rs	t1 a1 t2 a1 t3 a1 t1 a2 t4 a2 t5 a2 t5 a3	t5 s1 t6 s2	a1 bt a2 bt a3 rs s1 rs s2 bt

1. Check whether the above instantiation, once completed, is correct, and explain why it is or it is not.
2. Express in *FOL* the following queries and evaluate them over the completed instantiation:
 - (a) Return the tracks that are contained in an album and a special edition of the same artist.
 - (b) Return those artist that have recorded only albums that are not special editions.
 - (c) Check if there is a track appearing in all special editions.

Exercise 3. Model check the Mu-Calculus formula $\nu X.\mu Y.((\neg a \wedge \langle next \rangle X) \vee ([next]Y))$ and the CTL formula $EF(AG(a \supset EXAX \neg a))$ (showing its translation in Mu-Calculus) against the following transition system:



Exercise 4. Consider the following two transition systems:



Write the definition of bisimilarity and compute the bisimilarity relation for the two transition systems.

Exercise 5. Compute the certain answers to the following CQs over the following incomplete database (naive tables), and discuss how you obtained the result:

$$q() \leftarrow \text{lives}(x, y), \text{incountry}(y, z)$$

$$q(x, z) \leftarrow \text{lives}(x, y), \text{incountry}(y, z)$$

lives	
person	city
null ₀	null ₁
null ₂	null ₃
null ₄	null ₅
mary	null ₅

incountry	
city	country
null ₁	IT
null ₃	null ₆
null ₅	JP

1) **ALPHABET**: $T(x)$, $A(x)$, $\text{Art}(x)$, $S(x)$, $\text{bonus}(x,y)$, $\text{contained}(x,y)$, $\text{recorded}(x,y)$

ISA: $\forall x \ S(x) \supset A(x)$

BONUS: $\forall xy \ \text{bonus}(x,y) \supset T(x) \wedge S(y)$

$\forall x \ T(x) \supset \#\{y \mid \text{bonus}(x,y)\} \leq 1$

$\forall xy \ \text{bonus}(x,y) \supset \text{contained}(x,y)$

CONTAINED: $\forall xy \ \text{contained}(x,y) \supset T(x) \wedge A(y)$

$\forall x \ T(x) \supset 1 \leq \#\{y \mid \text{contained}(x,y)\}$

$\forall y \ A(y) \supset 1 \leq \#\{x \mid \text{contained}(x,y)\}$

RECORDED: $\forall xy \ \text{recorded}(x,y) \supset A(x) \wedge \text{Art}(y)$

$\forall x \ A(x) \supset 1 \leq \#\{y \mid \text{recorded}(x,y)\} \leq 1$

$\forall y \ \text{Art}(y) \supset 1 \leq \#\{x \mid \text{recorded}(x,y)\}$

2) Complete instantiation (merge Album with SE tables and Contained with Bonus) is correct because all axioms are made true

- $\exists y y' z z' \ \text{bonus}(x,y) \wedge \text{contained}(x,y') \wedge \text{recorded}(y,z) \wedge \text{recorded}(y',z') \wedge z = z'$
 $\Rightarrow \{\dagger_5\}$
- $\forall x \ \text{recorded}(x,y) \supset \neg S(x)$
 $\Rightarrow \{\emptyset\}$
- $\exists x \ \forall y \ S(y) \supset \text{bonus}(x,y)$
 $\Rightarrow \{\emptyset\}$

4) Two transition systems are bisimilar if:

- locally they look equal
- each action done on one of them can be done also on second one

R_0 = cartesian product = $\{(t_1, q_1)(t_1, q_2)(t_1, q_3)(t_2, q_1)(t_2, q_2)(t_2, q_3)\}$

R_1 = remove pairs that violate local condition on final states
 $= \{(t_1, q_1)(t_2, q_2)(t_2, q_3)\}$

R_2 = remove pairs that can "accept" action only on one of the two states
 $= \{(t_1, q_1)(t_2, q_2)\}$

R_3 = remove pairs that lead to pairs no more in the list] greatest fixpoint
 $= \{(t_1, q_1)(t_2, q_2)\}$

(t_1, q_1) belongs to gfp $\rightarrow T$ and Q are bisimilar

$$3) \nu X \mu Y ((\neg a \wedge \leftrightarrow X) \vee (\neg Y))$$

$$[X_0] = S$$

$$[X_1] = \mu Y ((\neg a \wedge \leftrightarrow X) \vee (\neg Y)) = \{1, 2\}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = ([\neg a] \cap \text{PreE}(-, [X_0])) \cup \text{PreA}(-, [Y_{10}])$$

$$= \{1, 2\} \cap \{S\} \cup \emptyset = \{1, 2\}$$

$$[Y_{12}] = ([\neg a] \cap \text{PreE}(-, [X_0])) \cup \text{PreA}(-, [Y_{11}]) \rightarrow \text{least fixpoint}$$

$$= \{1, 2\} \cup \{1\} = \{1, 2\}$$

$$[X_2] = \mu Y ((\neg a \wedge \leftrightarrow X) \vee (\neg Y)) = \{1, 2\}$$

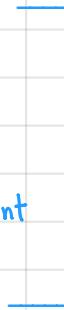
$$[Y_{20}] = \emptyset$$

$$[Y_{21}] = ([\neg a] \cap \text{PreE}(-, [X_1])) \cup \text{PreA}(-, [Y_{20}])$$

$$= \{1, 2\} \cap \{0, 1, 2\} \cup \emptyset = \{1, 2\}$$

$$[Y_{22}] = ([\neg a] \cap \text{PreE}(-, [X_1])) \cup \text{PreA}(-, [Y_{21}]) \rightarrow \text{last fixpoint}$$

$$= \{1, 2\} \cup \{1\} = \{1, 2\}$$



greatest fixpoint

Initial state not in solution $\rightarrow T \not\models \phi$

δ

$$CTL : \exists F (AG(a \supset \exists AX \neg a))$$

α

$$\alpha = \neg a = \text{PreA}(-, [a]) = \{3, 4\}$$

$$\beta = \exists X \alpha = \leftrightarrow \alpha - \text{PreE}(-, \{3, 4\}) = \{0, 3\}$$

$$\gamma = \alpha \supset \beta = \neg a \vee \beta = \{1, 2\} \cup \{0, 3\} = \{0, 1, 2, 3\}$$

$$\delta = AG \gamma = \nu X \gamma \wedge \neg X = \emptyset$$

$$[X_0] = S$$

$$[X_1] = \{0, 1, 2, 3\} \cap \text{PreA}(-, [X_0]) = \{0, 1, 2, 3\}$$

$$[X_2] = \{0, 1, 2, 3\} \cap \text{PreA}(-, [X_1]) = \{0, 1, 2\}$$

$$[X_3] = \{0, 1, 2, 3\} \cap \text{PreA}(-, [X_2]) = \{1, 2\}$$

$$[X_4] = \{0, 1, 2, 3\} \cap \text{PreA}(-, [X_3]) = \{1\}$$

$$[X_5] = \{0, 1, 2, 3\} \cap \text{PreA}(-, [X_4]) = \emptyset$$

$$\exists F \phi = \mu X \phi \vee \leftrightarrow X$$

$$[X_0] = \emptyset$$

$$[X_1] = \emptyset \cup \emptyset = \emptyset$$

Initial state not in solution $\rightarrow T \not\models \phi$

⑤ $q() \leftarrow \text{lives}(x,y), \text{incountry}(y,z)$

- Evaluate query on DB as it was complete : True (eg $\{\text{null}_0, \text{null}_1, \text{, } \text{IT}\}$)

• Remove null tuples : nothing to do since it is a boolean query

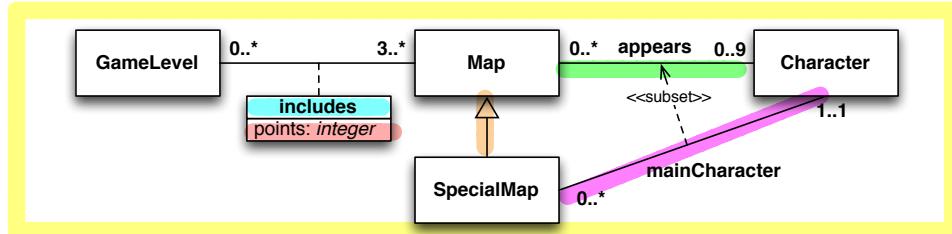
$q(x,z) \leftarrow \text{lives}(x,y), \text{incountry}(y,z)$

- Evaluate query on DB as it was complete : $\{\text{null}_0, \text{IT}, (\text{null}_2, \text{null}_6), (\text{null}_4, \text{JP}), (\text{moy}, \text{JP})\}$

• Remove null tuples : $\{(\text{moy}, \text{JP})\}$

We remove tuples because the certain answer is constituted by tuples of constants.

Exercise 1. Express the following UML class diagram in *FOL*.

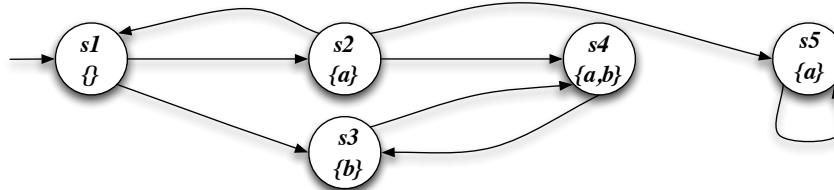


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation.

Map	SpecialMap	Character	appears	mainCharacter
artica bush	city desert	adrian bob charline	adrian adrian adrian bob charline artica	artica bush desert city artica
				adrian charline city desert

1. Check whether the instantiation (once completed) is correct (and explain why it is or it is not).
2. Express in *FOL* and evaluate the following queries:
 - (a) Return the maps with at least 3 distinct characters.
 - (b) Return the characters that appear in maps only as main characters.
 - (c) Check if there exists a map where all characters appears.

Exercise 3. Model check the Mu-Calculus formula $\nu X.\mu Y.((a \wedge \langle \text{next} \rangle X) \vee [\text{next}]Y)$ and the CTL formula $EF(\neg a \supset EXAGb)$ (showing its translation in Mu-Calculus) against the following transition system:



Exercise 4. Check whether the following Hoare triple is correct, using as *invariant* ($i + j = 9$).

{ $i=0$ AND $j=9$ } while ($i < 10$) do ($i := i+1$; $j=j-1$) { $j < 0$ }

Exercise 5. Given the following conjunctive queries:

```

q1(x) :- edge(x,y), edge(y,y), edge(x,z), edge(y,z), edge(z,y).
q2(x) :- edge(x,y), edge(y,z), edge(x,v), edge(v,z), edge(v,y).
    
```

check whether q_1 is contained into q_2 , explaining the technique used and, in case of containment, showing the homomorphism between the canonical databases.

1) ALPHABET: $G(x)$, $M(x)$, $SM(x)$, $C(x)$, includes (x, y) Integer(x)

ISA: $\forall x \ SM(x) \supset M(x)$

main character: $\forall x y \ mc(x, y) \supset C(x) \wedge SM(y)$

$\forall y \ SM(y) \supset 1 \leq \#\{x \mid mc(x)\} \leq 1$

$\forall x y \ mc(x, y) \supset opp(x, y)$

opposite: $\forall x y \ opp(x, y) \supset C(x) \wedge M(y)$

$\forall y \ M(y) \supset \#\{x \mid opp(x, y)\} \leq 9$

includes: $\forall x y \ inc(x, y) \supset G(x) \wedge M(y)$

$\forall x \ G(x) \supset 3 \leq \#\{y \mid inc(x, y)\}$

$\forall x y \ inc(x, y) \supset 1 \leq \#\{z \mid points(x, y, z)\} \leq 1$

points: $\forall x y z \ points(x, y, z) \supset inc(x, y) \wedge \text{Integer}(z)$

2) The completed instantiation (merge mc with opp table) is correct because makes all axioms true.

- $\exists x x' x'' opp(x, y) \wedge opp(x', y) \wedge opp(x'', y) \wedge x \neq x' \wedge x \neq x'' \wedge x' \neq x''$
 $\Rightarrow \{\emptyset\}$

- $\forall y \ opp(x, y) \supset mc(x, y)$

$\Rightarrow \{\emptyset\}$

- $\exists y \ \forall x \ C(x) \supset opp(x, y)$

$\Rightarrow \{\emptyset\}$

4) $I = \{i + j = 9\}$

$P = \{i = 0 \wedge j = 9\}$

$Q = \{j < 0\}$

$\delta = \{i = i+1; j = j-1\}$

$G = \{i < 10\}$

• Check $P \supset I$

$i = 0 \wedge j = 9 \supset i + j = 9 \quad \checkmark$

• Check $\neg G \wedge I \supset Q$

$i \geq 10 \wedge i + j = 9 \supset j < 0 \quad \checkmark \quad i = 10 \quad j = -1$

• Check $G \wedge I \supset wp(\delta, I)$

$i < 10 \wedge i + j = 9 \supset wp(\delta, I) \rightarrow$

$i < 10 \wedge i + j = 9 \supset i + j = 9 \quad \checkmark$

$\{i + i + 1 = 9\}$

$i = i + 1$

$\{i + j - 1 = 9\}$

$j = j - 1$

$\{i + j = 9\}$

I is an invariant so the Moore triple is correct!

$$3) \vee X \mu Y ((a \wedge \neg \rightarrow X) \vee \neg Y)$$

$$[X_0] = S$$

$$[X_1] = \mu Y ((a \wedge \neg \rightarrow X) \vee \neg Y) = S \rightarrow \text{greatest fixpoint}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = ([a] \wedge \text{PreE}(-, [X_0])) \cup \text{PreA}(-, [Y_{10}])$$

$$= \{2, 4, 5\} \cap \{5\} \cup \{\emptyset\} = \{2, 4, 5\}$$

$$[Y_{12}] = ([a] \wedge \text{PreE}(-, [X_0])) \cup \text{PreA}(-, [Y_{11}])$$

$$= \{2, 4, 5\} \cap \{3, 5\} = \{2, 3, 4, 5\}$$

$$[Y_{13}] = ([a] \wedge \text{PreE}(-, [X_0])) \cup \text{PreA}(-, [Y_{11}])$$

$$= \{2, 4, 5\} \cup \{1, 2, 3, 4, 5\} = S \text{ least fixpoint}$$

Initial state in solution $\rightarrow T \models \phi$

$$\text{CTL: } \text{EF}(\neg a \supset \overbrace{\text{EX} \underline{A \wedge b}}^{\alpha})$$

$$\alpha = \vee X b \wedge \neg X = \{3, 4, 5\}$$

$$[X_0] = S$$

$$[X_1] = [b] \cap \text{PreA}(-, [X_0])$$

$$= \{3, 4, 5\} \cap S = \{3, 4, 5\}$$

$$[X_2] = [b] \cap \text{PreA}(-, [X_1])$$

$$= \{3, 4, 5\} \cap \{2, 3, 4, 5\} = \{3, 4, 5\} \text{ greatest fixpoint}$$

$$\beta = \neg \rightarrow \alpha = \text{PreE}(-, \{3, 4, 5\}) = S$$

$$\gamma = a \vee \beta = \{2, 4, 5\} \cup \{5\} = S$$

$$\text{EF } \gamma = \mu X (\gamma \vee \neg \rightarrow X) = \{S\}$$

$$[X_0] = \emptyset$$

$$[X_1] = S \cup \text{PreE}(-, [X_0]) = S \text{ least fixpoint}$$

Initial state in solution $\rightarrow T \models \phi$

5)

1) Freeze free variables

$$\begin{aligned} q_1(a) &= e(a,y) \ e(y,y) \ e(a,z) \ e(y,z) \ e(z,y) \\ q_2(a) &= e(a,y) \ e(y,z) \ e(a,v) \ e(v,z) \ e(v,y) \end{aligned}$$

2) Canonical interpretations

$$I_{q_1} = \begin{cases} \Delta^{I_{q_1}} = \{a, y, z\} \\ a^{I_{q_1}} = a \\ E^{I_{q_1}} = \{(a,y)(y,y)(a,z)(y,z)(z,y)\} \end{cases}$$

$$I_{q_2} = \begin{cases} \Delta^{I_{q_2}} = \{a, y, z, v\} \\ a^{I_{q_2}} = a \\ E^{I_{q_2}} = \{(a,y)(y,z)(a,v)(v,z)(v,y)\} \end{cases}$$

3) Find homomorphism from I_{q_2} to I_{q_1} .

[CM ?? theorem]



$$h(a) = a$$

$$h(a,y) = (a, ?) \rightarrow h(y) = y$$

$$h(y,z) = (y, ?) \rightarrow h(z) = y$$

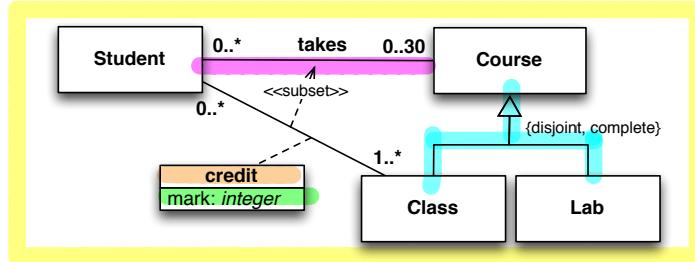
$$h(a,v) = (a, ?) \rightarrow h(v) = z$$

$$h(v,z) = (z, y) \quad \checkmark$$

$$h(v,y) = (z, y) \quad \checkmark$$

Homomorphism exists $\Rightarrow q_1 \subseteq q_2$

Exercise 1. Express the following UML class diagram in *FOL*.

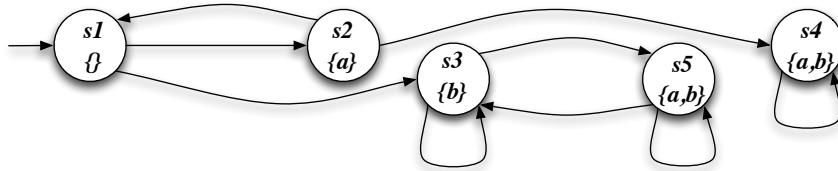


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation.

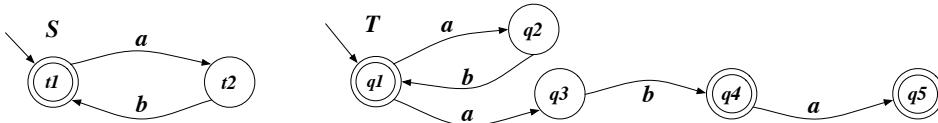
Student	Class	Lab	credit/mark	takes
peter	calculus	IoT lab	peter	IoT lab
paul	AI	db lab	paul	IoT lab
mary	FM	hacking lab	mary	FM
jane	algorithms		mary	db lab
			jane	hacking lab
			jane	IoT lab

1. Check whether the instantiation (once completed) is correct (and explain why it is or it is not).
2. Express in *FOL* and evaluate the following queries:
 - (a) Return students that have taken at least 3 courses.
 - (b) Return students that have taken only classes.
 - (c) Check if there exists a student that has taken all labs.
 - (d) Check if there is a student that has taken all classes, but not for credit.

Exercise 3. Model check the Mu-Calculus formula $\nu X.\mu Y.((a \wedge [next]X) \vee [next]Y)$ and the CTL formula $EF(\neg a \supset (EX a \wedge EX AG b))$ (showing its translation in Mu-Calculus) against the following transition system:



Exercise 4. Consider the following two transition systems:



Write the definition of bisimilarity and compute the bisimilarity relation for the two transition systems.

Exercise 5. Given the following conjunctive queries:

```
q1(x) :- edge(x,y), edge(y,z), edge(z,x).
q2(x) :- edge(x,y), edge(x,w), edge(y,z), edge(z,x), edge(z,v), edge(v,y), edge(v,w), edge(w,z).
```

check whether q_1 is contained into q_2 , explaining the technique used and, in case of containment, showing the homomorphism between the canonical databases.

1) ALPHABET: $s(x), c(x), cl(x), l(x)$, credit(x, y), mark(x, y, z), Integer(x)

ISA: $\forall x \ c(x) \supset cl(x) \vee l(x)$

$\forall x \ cl(x) \supset \neg l(x)$

$\forall x \ cl(x) \supset c(x)$

$\forall x \ l(x) \supset c(x)$

TAKES: $\forall xy \ tokens(x, y) \supset s(x) \wedge c(y)$

$\forall x \ s(x) \supset \#\{y \mid tokens(x, y)\} \leq 30$

CREDIT: $\forall xy \ credit(x, y) \supset s(x) \wedge cl(y)$

$\forall x \ s(x) \supset 1 \leq \#\{y \mid credit(x, y)\}$

$\forall xy \ credit(x, y) \supset 1 \leq \#\{z \mid mark(x, y, z)\} \geq 1$

$\forall xy \ credit(x, y) \supset tokens(x, y)$

MARK: $\forall xyz \ mark(x, y, z) \supset credit(x, y) \wedge Integer(z)$

2) Completed instantiation (odd table for course unifying clos and ldo and odd missing line in "tokens" from "credit" eg Petri - algorithm) is correct because all axioms are true.

- $\exists yy'y \ tokens(x, y) \wedge token(x, y') \wedge tokens(x, y'') \wedge y \neq y' \wedge y \neq y'' \wedge y' = y''$

$\Rightarrow \{ \text{Jones} \}$

- $\forall y \ tokens(x, y) \supset cl(y)$

$\Rightarrow \{ \text{Moby} \}$

- $\exists x \forall y \ l(y) \supset tokens(x, y)$

$\Rightarrow \{ \text{Jones} \}$

- $\exists x \forall y \ cl(y) \supset tokens(x, y) \wedge \neg credit(x, y)$

$\Rightarrow \{ \emptyset \}$

5) $q_1 \subseteq q_2$?

- Freeze free variables (x)

- Build canonical interpretation of q_1 and q_2

$$I_{q_1} = \begin{cases} \Delta^{I_{q_1}} = \{x, y, z\} \\ x^{I_{q_1}} = x \\ \Xi^{I_{q_1}} = \{(x, y), (y, z), (z, x)\} \end{cases}$$

$$I_{q_2} = \begin{cases} \Delta^{I_{q_2}} = \{x, y, z, v, w\} \\ x^{I_{q_2}} = x \\ \Xi^{I_{q_2}} = \{(x, y), (x, w), (y, z), (z, x), (z, v), (v, y), (v, w), (w, z)\} \end{cases}$$

- Check if $I_{q_1} \models I_{q_2} \rightarrow$ find homomorphism from I_{q_2} to I_{q_1}

$$- h(x_2) = x_1$$

$$- h(x_1, y_2) = (x_1, ?) \rightarrow h(y_2) = y_1$$

$$- h(x_2, w_2) = (x_1, ?) \rightarrow h(w_2) = y_1$$

$$- h(y_2, z_2) = (y_1, ?) \rightarrow h(z_2) = z_1$$

$$- h(z_2, x_2) = (z_1, x_1) \quad \checkmark$$

$$- h(z_2, v_2) = (z_1, ?) \rightarrow h(v_2) = x_1$$

$$- h(v_2, y_2) = (x_1, y_1) \quad \checkmark$$

$$- h(v_2, w_2) = (x_1, y_1) \quad \checkmark$$

$$- h(w_2, z_2) = (y_1, z_1) \quad \checkmark$$

Exists a homomorphism

$$\Rightarrow q_1 \subseteq q_2$$

$$3) \vee X \mu Y ((\alpha \wedge \neg X) \vee \neg Y)$$

$$[X_0] = S$$

$$[X_1] = \mu Y ((\alpha \wedge \neg X) \vee \neg Y) = \{2, 4, 5\}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = ([\alpha] \cap \text{PreA}(-, [X_0])) \cup \text{PreA}(-, [Y_{10}])$$

$$= \{3, 5, 4\} \cap \{S\} \cup \{\emptyset\} = \{3, 5, 4\}$$

$$[Y_{12}] = ([\alpha] \cap \text{PreA}(-, [X_0])) \cup \text{PreA}(-, [Y_{11}])$$

$$= \{3, 5, 4\} \cup \{4\} = \{3, 5, 4\}$$

$$[X_2] = \mu Y ((\alpha \wedge \neg X) \vee \neg Y) = \{4\}$$

$$[Y_{20}] = \emptyset$$

$$[Y_{21}] = ([\alpha] \cap \text{PreA}(-, [X_1])) \cup \text{PreA}(-, [Y_{20}])$$

$$= \{3, 4, 5\} \cap \{4\} \cup \{\emptyset\} = \{4\}$$

$$[Y_{22}] = ([\alpha] \cap \text{PreA}(-, [X_1])) \cup \text{PreA}(-, [Y_{21}])$$

$$= \{4\} \cup \{4\} = \{4\}$$

$$[X_3] = \mu Y ((\alpha \wedge \neg X) \vee \neg Y) = \{4\}$$

$$[Y_{30}] = \emptyset$$

$$[Y_{31}] = ([\alpha] \cap \text{PreA}(-, [X_2])) \cup \text{PreA}(-, [Y_{30}])$$

$$= \{3, 4, 5\} \cap \{4\} \cup \{\emptyset\} = \{4\}$$

$$[Y_{32}] = ([\alpha] \cap \text{PreA}(-, [X_2])) \cup \text{PreA}(-, [Y_{31}])$$

$$= \{3, 4, 5\} \cap \{4\} \cup \{4\} = \{4\}$$

→ greatest fixpoint

Initial state not in solution $\rightarrow T \not\models \phi$

$$\text{CTL: } EF(\neg a \Rightarrow (\overbrace{EX a \wedge \overbrace{EX AG b}}^{\beta}))$$

$$\alpha = \vee X b \wedge \neg X = \{3, 4, 5\}$$

$$[X_0] = S$$

$$[X_1] = [b] \cap \text{PreA}(-, [X_0])$$

$$= \{3, 4, 5\} \cap S = \{3, 4, 5\}$$

$$[X_2] = [b] \cap \text{PreA}(-, [X_1])$$

$$= \{3, 4, 5\} \cap \{3, 4, 5\} = \{3, 4, 5\}$$

→ greatest fixpoint

$$\beta = \neg \alpha = \text{PreE}(-, [\alpha]) = S$$

$$\gamma = \neg \alpha = \text{PreE}(-, [\alpha]) = S$$

$$\delta = S \cap S = S$$

$$\omega = \alpha \vee S = S$$

$$EF(\omega) = \mu X \omega \vee \neg X = S$$

$$[X_0] = \emptyset$$

$$[X_1] = [\omega] \cup \text{PreE}(-, [X_0])$$

$$= S \cup \emptyset = S \rightarrow \text{least fixpoint}$$

Initial state in solution $\rightarrow T \models \phi$

4) Two transition system are bisimilar if:

- locally they look equal
- each action done on one of them can be done also on the second one

$$R_0 = \text{congestion product} = \{(t_1, q_1), (t_1, q_2), (t_1, q_3), (t_1, q_4), (t_1, q_5), (t_2, q_1), (t_2, q_2), (t_2, q_3), (t_2, q_4), (t_2, q_5)\}$$

R_1 = remove pairs that violate local condition on final state

$$= \{(t_1, q_1)(t_1, q_4)(t_1, q_5)(t_2, q_1)(t_2, q_2)(t_2, q_3)(t_2, q_4)(t_2, q_5)\}$$

R_2 = remove actions that can be done only on one of the two states

$$= \{(t_1, q_1)(t_1, q_4)(t_2, q_2)(t_2, q_3)\}$$

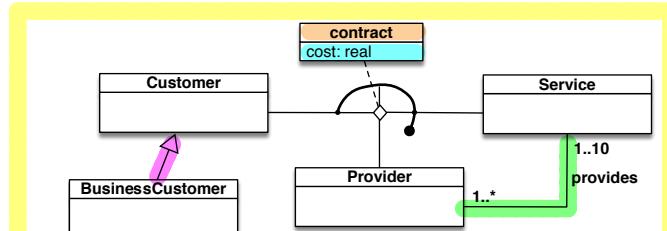
R_3 = remove pairs that lead to pairs no more in the list

$$= \{(t_1, q_1)(t_2, q_2)\}$$

$$R_4 = \text{some} = \{(t_1, q_1)(t_2, q_2)\}$$

(t_1, q_1) belong to greatest fixpoint $\rightarrow S$ and T are bisimilar

Exercise 1. Express the following UML class diagram in FOL:

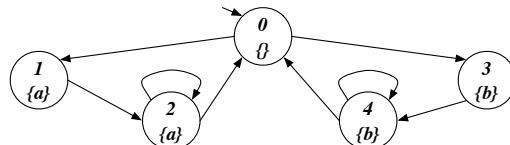


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation:

Customer	BCustomers	Services	Provider	provides	contacts/cost
c1 c2 c3 c4	b1 b2 b3	s1 s2 s3	p1 p2	p1 s1 p1 s2 p1 s3 p2 s2	c1 s1 p1 90.0 c1 s2 p1 80.0 c1 s3 p1 50.0 b2 s1 p2 170,0 b2 s2 p2 100,0

1. Check whether the above instantiation, once completed, is correct, and explain why it is or it is not.
2. Express in FOL the following queries and evaluate them over the completed instantiation:
 - (a) Check that, for every provider x and service y involved in a contract, provider x does provide service y .
 - (b) Return those customers that have contracts only for services provided by $p2$.
 - (c) Return those customers that have a contract for with all providers.

Exercise 3. Model check the Mu-Calculus formula $\nu X.\mu Y.((b \wedge [next]X) \vee (a \wedge \langle next \rangle Y))$ and the CTL formula $EF(AG(a \supset EXAX \neg a))$ (showing its translation in Mu-Calculus) against the following transition system:



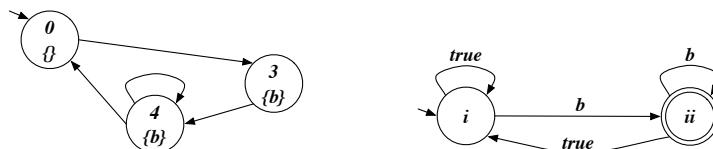
Exercise 4. Check whether the Hoare triple below is correct, by using $(x \geq 0 \wedge y \geq 0 \wedge x + y = 31)$ as invariant:

$$\{x = 31 \wedge y = 0\} \text{ while}(x > 0) \text{ do } (x := x - 1; y := y + 1) \{y = 31\}$$

Exercise 5. Check whether the following FOL formula is valid, by using tableaux:

$$(\exists x.P(x) \vee \exists x.Q(x)) \equiv \exists x.(P(x) \vee Q(x))$$

Exercise 6 (optional). Model check the LTL formula $\diamond \square \neg b$ against the following transition system, by considering that the Büchi automaton for $\neg(\diamond \square \neg b)$ is the one below:



1) ALPHABET: $c(x)$, $b(x)$, $p(x)$, $s(x)$, $\text{cont}_z(x,y,z)$, $\text{cost}(x,y,z,w)$, $\text{Real}(x)$

ISA: $\forall x \ b(x) \supset c(x)$

PROVIDES: $\forall xy \ \text{prov}(x,y) \supset p(x) \wedge s(y)$

$\forall x \ P(x) \supset 1 \leq \#\{y \mid \text{prov}(x,y)\} \geq 10$

$\forall y \ S(y) \supset 1 \leq \#\{x \mid \text{prov}(x,y)\}$

CONTRACT: $\forall xyz \ \text{cont}_z(x,y,z) \supset c(x) \wedge s(y) \wedge p(z)$

$\forall xyz \ \text{cont}_z(x,y,z) \wedge \text{cont}_z(x,y,z) \supset z = z$

$\forall xyz \ \text{cont}_z(x,y,z) \supset 1 \leq \#\{w \mid \text{cost}(x,y,z,w)\} \geq 1$

COST: $\forall xyzw \ \text{cost}(x,y,z,w) \supset \text{cont}_z(x,y,z) \wedge \text{Real}(w)$

2) Completed instantiation is correct because all axioms are true

- $\forall xy \ \text{cont}_z(z,y,x) \supset \text{prov}(x,y)$

\Rightarrow false [$\text{cont}_z(b_2, s_1, p_2)$ but not $\text{prov}(p_2, s_1)$]

- $\exists yz \ \text{cont}_z(x,y,z) \supset \text{prov}(p_2, y)$

$\Rightarrow \{\emptyset\}$

- $\exists z \forall y \ S(y) \supset \text{cont}_z(x,y,z)$

$\Rightarrow \{\emptyset\}$

5) $\neg [(\exists x P(x) \vee \exists x Q(x)) \equiv \exists x (P(x) \vee Q(x))]$

$$\begin{array}{c} \neg [(\exists x (P(x) \vee Q(x)) \equiv \exists x (P(x) \vee Q(x))] \\ \downarrow \quad \downarrow \\ \begin{array}{c} \exists x P(x) \vee \exists x Q(x) \\ \neg (\exists x (P(x) \vee Q(x))) \end{array} \quad \begin{array}{c} \neg (\exists x P(x) \vee \exists x Q(x)) \\ \exists x (P(x) \vee Q(x)) \end{array} \\ \downarrow \quad \downarrow \\ \begin{array}{c} \exists x P(x) \quad \exists x Q(x) \\ \downarrow \quad \downarrow \\ P(\alpha) \times \quad Q(\beta) \times \\ \downarrow \quad \downarrow \\ \neg (P(\alpha) \vee Q(\beta)) \quad \neg (P(\alpha) \vee Q(\beta)) \\ \downarrow \quad \downarrow \\ \times \neg P(\alpha) \quad \neg P(\alpha) \\ \neg Q(\beta) \quad \neg Q(\beta) \times \end{array} \quad \begin{array}{c} P(\alpha) \vee Q(\alpha) \\ \downarrow \quad \downarrow \\ P(\alpha) \times \quad Q(\alpha) \times \\ \downarrow \quad \downarrow \\ \neg \exists P(x) \quad \neg \exists Q(x) \\ \downarrow \quad \downarrow \\ \neg Q(\alpha) \quad \neg P(\alpha) \times \end{array} \end{array} \end{array}$$

$\neg \Gamma \text{ unsat} \Rightarrow \Gamma \text{ valid}$

$$3) \vee X \mu Y ((b \wedge \neg X) \vee (a \wedge \neg \rightarrow Y))$$

$$[X_0] = S$$

$$[X_1] = \mu Y ((b \wedge \neg X) \vee (a \wedge \neg \rightarrow Y)) = \{3, 4\}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = ([b] \cap \text{PreA}(-, [X_0])) \cup ([a] \cap \text{PreE}(-, [Y_{10}]))$$

$$= \{3, 4\} \cap \{5\} \cup \{1, 2\} \cap \emptyset = \{3, 4\}$$

$$[Y_{12}] = ([b] \cap \text{PreA}(-, [X_0])) \cup ([a] \cap \text{PreE}(-, [Y_{11}])) \\ = \{3, 4\} \cup \{1, 2\} \cap \{0, 3, 4\} = \{3, 4\}$$

least fixpoint

$$[X_2] = \mu Y ((b \wedge \neg X) \vee (a \wedge \neg \rightarrow Y)) = \{3\}$$

$$[Y_{20}] = \emptyset$$

$$[Y_{21}] = ([b] \cap \text{PreA}(-, [X_1])) \cup ([a] \cap \text{PreE}(-, [Y_{20}]))$$

$$= \{3, 4\} \cap \{3\} \cup \{1, 2\} \cap \emptyset = \{3\}$$

$$[Y_{22}] = ([b] \cap \text{PreA}(-, [X_1])) \cup ([a] \cap \text{PreE}(-, [Y_{21}])) \\ = \{3\} \cup \{1, 2\} \cap \{0\} = \{3\}$$

least fixpoint

$$[X_3] = \mu Y ((b \wedge \neg X) \vee (a \wedge \neg \rightarrow Y)) = \emptyset$$

$$[Y_{30}] = \emptyset$$

$$[Y_{31}] = ([b] \cap \text{PreA}(-, [X_2])) \cup ([a] \cap \text{PreE}(-, [Y_{30}]))$$

$$= \{3, 4\} \cap \emptyset \cup \{1, 2\} \cap \emptyset = \emptyset$$

least fixpoint

Initial state not in solution $\rightarrow T \not\models \phi$

$$\overbrace{\text{CTL: } EF(AG(a \supset \exists X A x \supset a))}^{\delta} \\ \underbrace{\quad \quad \quad \quad \quad \quad}_{\alpha \beta \gamma \delta \epsilon}$$

$$\alpha = \neg a = \{3, 4\}$$

$$\beta = \neg \rightarrow \alpha = \{0, 3, 4\}$$

$$\gamma = \neg a \vee \beta = \{0, 3, 4\} \cup \{0, 3, 4\} = \{0, 3, 4\}$$

$$\delta = \vee X \delta \wedge \neg X = \emptyset$$

$$[X_0] = \{5\}$$

$$[X_1] = [\delta] \cap \text{PreA}(-, [X_0])$$

$$= \{0, 3, 4\} \cap \{5\} = \{0, 3, 4\}$$

$$[X_2] = [\delta] \cap \text{PreA}(-, [X_1])$$

$$= \{0, 3, 4\} \cap \{3, 4\} = \{3, 4\}$$

$$[X_3] = [\delta] \cap \text{PreA}(-, [X_2])$$

$$= \{3, 4\} \cap \{3\} = \{3\}$$

$$[X_4] = [\delta] \cap \text{PreA}(-, [X_3])$$

$$= \{3, 4\} \cap \{\emptyset\} = \emptyset$$

greatest fixpoint

$$EF(\delta) = \mu X \delta \vee \neg \rightarrow X = \emptyset$$

$$[X_0] = \emptyset$$

$$[X_1] = [\delta] \cup \text{PreE}(-, [X_0])$$

$= \emptyset \cup \emptyset$ least fixpoint

Initial state not in solution $\rightarrow T \not\models \phi$

$$I = \{x \geq 0 \wedge y \geq 0 \wedge x+y=31\}$$

$$P = \{x=31 \wedge y=0\}$$

$$G = \{x > 0\}$$

$$\bar{G} = \{x = x-1; y = y+1\}$$

$$Q = \{y = 31\}$$

- Check $P \supseteq I$

$$x=31 \wedge y=0 \supseteq x \geq 0 \wedge y \geq 0 \wedge x+y=31 \quad \checkmark$$

- Check $\neg G \wedge I \supseteq Q$

$$x \leq 0 \wedge \underbrace{x \geq 0}_{x=0} \wedge y > 0 \wedge x+y=31 \supseteq y=31 \quad \checkmark$$

- Check $\{G \wedge I\} \delta \{I\} = G \wedge I \supseteq W_P(\delta, I)$

$$x > 0 \wedge x \geq 0 \wedge y \geq 0 \wedge x+y=31 \supseteq W_P(\delta, I) \quad \curvearrowright$$

$$x > 0 \wedge x \geq 0 \wedge y \geq 0 \wedge x+y=31 \supseteq x \geq 1 \wedge y \geq -1 \wedge x+y=31 \quad \checkmark$$

$$\{x \geq 1 \wedge y \geq -1 \wedge x+y=31\}$$

$$x = x-1$$

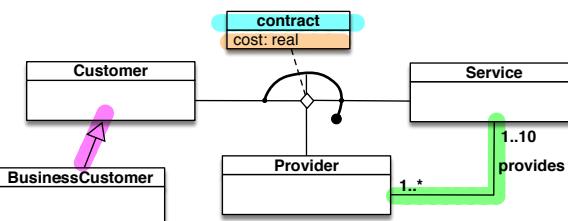
$$\{x \geq 0 \wedge y \geq -1 \wedge x+y=30\}$$

$$y = y+1$$

$$\{x \geq 0 \wedge y \geq 0 \wedge x+y=31\}$$

I is an invariant \rightarrow the hoare triple is correct!

Exercise 1. Express the following UML class diagram in FOL:

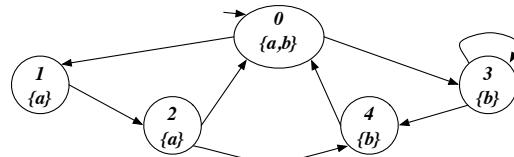


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation:

Customer	BCustomers	Services	Provider	provides	contacts/cost
c1 c2 c3 c4	b1 b2 b3	s1 s2 s3	p1 p2	p1 s1 p1 s2 p1 s3 p2 s2	c1 s1 p1 90.0 c1 s2 p1 80.0 c1 s3 p1 50.0 b2 s1 p2 170,0 b2 s2 p2 100,0

1. Check whether the above instantiation, once completed, is correct, and explain why it is or it is not.
2. Express in FOL the following queries and evaluate them over the completed instantiation:
 - (a) Check whether there is a customer with contract with two providers for the same service.
 - (b) Return those customers that have contracts only for one service.
 - (c) Return those customers that have contracts with the same provider for all their services.

Exercise 3. Model check the Mu-Calculus formula $\nu X.\mu Y.((a \wedge [next]X) \vee (b \wedge [next]Y))$ and the CTL formula $AF(EG(a \supset EXAXb))$ (showing its translation in Mu-Calculus) against the following transition system:



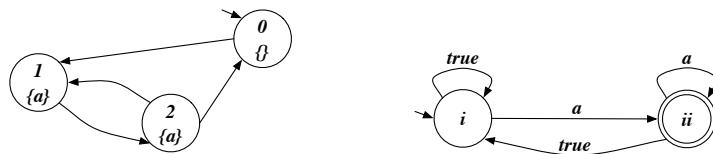
Exercise 4. Check whether CQ q_1 is contained in CQ q_2 , reporting canonical DBs and homomorphism:

$$\begin{aligned} q_1(x_r) &\leftarrow e(x_r, x_g), e(x_g, x_b), e(x_b, x_r). \\ q_2(x) &\leftarrow e(x, y), e(y, z), e(z, x), e(z, v)e(v, w), e(w, z). \end{aligned}$$

Exercise 5. Check whether the following FOL formula is valid, by using tableaux:

$$(\forall x.P(x) \supset Q(x)) \supset (\exists x.P(x) \supset \exists x.Q(x))$$

Exercise 6 (optional).¹ Model check the LTL formula $\diamond \square \neg a$ against the following transition system, by considering that the Büchi automaton for $\neg(\diamond \square \neg a)$ is the one below:



¹The student can get the maximum grade even without doing Exercise 6.

1) **Alphabet:** $C(x), B(x), P(x), S(x), \text{real}(x)$
 $\text{cont}_r(x, y, z)$
 $\text{cost}(x, y, z, w)$

ISA: $\forall x B(x) \supset C(x)$

PROVIDES: $\forall x, y \text{ provides}(x, y) \supset P(x) \wedge S(y)$
 $\forall x P(x) \supset (\exists y \mid \text{provides}(x, y)) \leq 1$
 $\forall y S(y) \supset (\exists x \mid \text{provides}(x, y)) \leq 1$

CONTRACT: $\forall x, y, z \text{ cont}_r(x, y, z) \supset C(x) \wedge S(y) \wedge P(z)$

$\forall x, y, z, z' \text{ cont}_r(x, y, z) \wedge \text{cont}_r(x, y, z') \supset z = z'$

$\forall x, y, z \text{ cont}_r(x, y, z) \supset (\exists w \mid \text{cost}(x, y, z, w) \leq 1)$

COST: $\forall x, y, z, w \text{ cost}(x, y, z, w) \supset \text{cont}(x, y, z) \wedge \text{real}(w)$

2) Instantiation is correct because satisfies all axioms in Γ

- $\exists y, z, z' \text{ cont}_r(x, y, z) \wedge \text{cont}_r(x, y, z') \wedge z \neq z'$
 No need to check $C(x), S(y), P(z)$ alone because they are implied in $\text{cont}_r(x, y, z)$
 $\Rightarrow \{\emptyset\}$
- $\exists y, z \text{ cont}_r(x, y, z) \wedge [\forall y' \exists z \text{ cont}_r(x, y, z) \wedge \exists z' \text{ cont}_r(x, y', z') \supset y = y']$
 $\Rightarrow \{\emptyset\}$
- $\exists p \forall y \exists z \text{ cont}_r(x, y, z) \supset \text{cont}_r(x, y, p)$
 $\Rightarrow \{c_1, b_2\}$

5) $\neg(\forall x P(x) \supset Q(x)) \supset (\exists x P(x) \supset \exists x Q(x))$

$$\begin{array}{c}
 \neg(\forall x P(x) \supset Q(x)) \\
 \downarrow \\
 \neg(\exists x P(x) \supset \exists x Q(x)) \\
 \downarrow \\
 \exists x P(x) \\
 \neg \exists x Q(x) \\
 \downarrow \\
 P(\alpha) \\
 \neg P(\alpha) \times \\
 \downarrow \\
 P(\alpha) \supset Q(\alpha) \\
 / \quad \backslash \\
 \neg P(\alpha) \times \quad Q(\alpha) \times \\
 \downarrow \\
 \neg Q(\alpha) \times
 \end{array}
 \quad
 \begin{array}{l}
 [\neg(\alpha \supset \beta)] \\
 [\neg(\alpha \supset \beta)]
 \end{array}$$

we need to generate a clash so we need a $\neg P(\alpha)$
 $P(\alpha) \times \rightarrow$ useful to expand $\neg \exists x Q(x)$!

$\neg \Gamma$ is unsat. $\rightarrow \Gamma$ is valid!

$$3) \vee X \mu Y ((a \wedge \neg X) \vee (b \wedge \neg Y))$$

$$[X_0] = S$$

$$[X_1] = \mu Y ((a \wedge \neg X) \vee (b \wedge \neg Y)) = \{S_0, S_1, S_2, S_4\}$$

$$[Y_{10}] = \emptyset$$

$$[Y_{11}] = (a \wedge \neg X) \vee (b \wedge \neg Y)$$

$$= ([a] \cap \text{PreA}(-, [X_0])) \cup ([b] \cap \text{PreA}(-, [Y_{10}]))$$

$$= (\{S_0, S_1, S_2\} \cap \{S_0, S_1, S_2, S_3, S_4\}) \cup (\{S_0, S_3, S_4\} \cap \{\emptyset\}) = \{S_0, S_1, S_2\}$$

$$[Y_{12}] = (([a] \cap \text{PreA}(-, [X_0])) \cup ([b] \cap \text{PreA}(-, [Y_{11}])))$$

$$= \{S_0, S_1, S_2\} \cup \{S_4\} = \{S_0, S_1, S_2, S_4\}$$

$$[Y_{13}] = (([a] \cap \text{PreA}(-, [X_0])) \cup ([b] \cap \text{PreA}(-, [Y_{12}]))) \rightarrow \text{least fixpoint}$$

$$= \{S_0, S_1, S_2\} \cup \{S_4\} = \{S_0, S_1, S_2, S_4\}$$

$$[X_2] = \mu Y ((a \wedge \neg X) \vee (b \wedge \neg Y)) = \{S_1, S_2\}$$

$$[Y_{20}] = \emptyset$$

$$[Y_{21}] = (a \wedge \neg X_1) \vee (b \wedge \neg Y_{20})$$

$$= ([a] \cap \text{PreA}(-, [X_1])) \cup ([b] \cap \text{PreA}(-, [Y_{20}]))$$

$$= (\{S_0, S_1, S_2\} \cap \{S_1, S_2, S_4\}) \cup (\{S_0, S_3, S_4\} \cap \{\emptyset\}) = \{S_1, S_2\}$$

$$[Y_{22}] = ([a] \cap \text{PreA}(-, [X_1])) \cup ([b] \cap \text{PreA}(-, [Y_{21}]))$$

$$= \{S_1, S_2\} \cup \{S_1\} = \{S_1, S_2\}$$

$$[X_3] = \mu Y ((a \wedge \neg X) \vee (b \wedge \neg Y)) = \{S_1\}$$

$$[Y_{30}] = \emptyset$$

$$[Y_{31}] = (a \wedge \neg X_2) \vee (b \wedge \neg Y_{30})$$

$$= ([a] \cap \text{PreA}(-, [X_2])) \cup ([b] \cap \text{PreA}(-, [Y_{30}]))$$

$$= (\{S_0, S_1, S_2\} \cap \{S_1\}) \cup (\{S_0, S_3, S_4\} \cap \{\emptyset\}) = \{S_1\}$$

$$[Y_{32}] = ([a] \cap \text{PreA}(-, [X_2])) \cup ([b] \cap \text{PreA}(-, [Y_{31}]))$$

$$= \{S_1\} \cup \{\emptyset\} = \{S_1\}$$

$$[X_4] = \mu Y ((a \wedge \neg X) \vee (b \wedge \neg Y)) = \{\emptyset\} \text{ greatest fixpoint}$$

$$[Y_{40}] = \emptyset$$

$$[Y_{41}] = (a \wedge \neg X_3) \vee (b \wedge \neg Y_{40})$$

$$= ([a] \cap \text{PreA}(-, [X_3])) \cup ([b] \cap \text{PreA}(-, [Y_{40}]))$$

$$= (\{S_0, S_1, S_2\} \cap \{\emptyset\}) \cup (\{S_0, S_3, S_4\} \cap \{\emptyset\}) = \{\emptyset\} \text{ least fixpoint}$$

Initial stage not in solution $\Rightarrow T \not\models \phi$

$$CTL: AF(EG(a \supseteq EXAx_b))$$

$$\alpha = Ax_b = [-]b = \text{PreA}(-, [b]) = \{s_2, s_3, s_4\}$$

$$\beta = \exists X \alpha = \neg \rightarrow \alpha = \text{PreE}(-, [\alpha]) = \{s_0, s_1, s_2, s_3\}$$

$$\gamma = \alpha \supseteq \beta = \neg \alpha \vee \beta = \{s_3, s_4\} \cup \{s_0, s_1, s_2, s_3\} = S$$

$$\sigma = EG \gamma = vX \gamma \wedge \neg \rightarrow X$$

$$[x_0] = S$$

$$[x_1] = [\gamma] \wedge \text{PreE}(-, [x_0]) = S \cap S = \{s\} \text{ fixpoint}$$

$$AF \sigma = \mu X \sigma \vee [-]X$$

$$[x_0] = \emptyset$$

$$[x_1] = [\sigma] \cup \text{PreA}(-, [x_0]) = \{s\} \text{ fixpoint}$$

Initial state in reduction $\Rightarrow T \models \phi$

4) $q_1 \subseteq q_2$?

- Freeze free variables (x_r and x)

- Build canonical interpretation of q_1 and q_2

$$I_{q_1} = \begin{cases} \Delta^{I_{q_1}} = \{x_r, x_g, x_b\} \\ x_r^{I_{q_1}} = x_r \\ E^{I_{q_1}} = \{(x_r, x_g), (x_g, x_b), (x_b, x_r)\} \end{cases}$$

$$I_{q_2} = \begin{cases} \Delta^{I_{q_2}} = \{x, y, z, v, w\} \\ x^{I_{q_2}} = x \\ E^{I_{q_2}} = \{(x, y), (y, z), (z, x), (z, v), (v, w), (w, z)\} \end{cases}$$

- Check if $I_{q_1} \models q_2 \rightarrow$ find homomorphism from I_{q_2} to I_{q_1}

On order, check all $e \in E^{I_{q_2}}$

- $h(x) = x_r$ because constants

- $h(x, y) = (x_r, ?)$ \rightarrow find one e starting with $x_r \checkmark \rightarrow (x_r, x_g) \rightarrow h(y) = x_g$

- $h(y, z) = (x_g, ?)$ \rightarrow again $\checkmark \rightarrow (x_g, x_b) \rightarrow h(z) = x_b$

- $h(z, x) = (x_b, x_r)$ \rightarrow is present in $E^{I_{q_1}}$ \checkmark

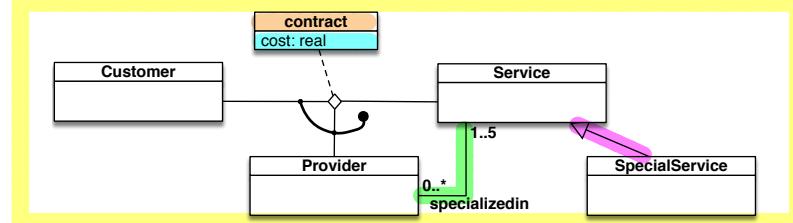
- $h(z, v) = (x_b, ?)$ \rightarrow find e starting with $x_b \checkmark \rightarrow (x_b, x_r) \rightarrow h(v) = x_r$

- $h(v, w) = (x_r, ?)$ $\rightarrow \dots \checkmark \rightarrow (x_r, x_g) \rightarrow h(w) = x_g$

- $h(w, z) = (x_g, x_b)$ \rightarrow is present in $E^{I_{q_1}}$ \checkmark

Exists one homomorphism $\Rightarrow q_1 \subseteq q_2$

Exercise 1. Express the following UML class diagram in FOL:

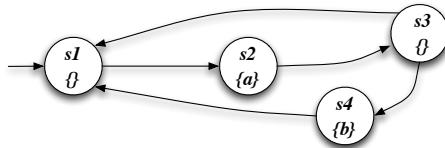


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation:

Customer	Service	SpecialService	Provider	specializedin	contacts / cost
c1 c2 c3 c4	s1 s2 s3	ss1 ss2	p1 p2	p1 s1 p1 s2 p1 s3 p2 ss1 p2 ss2	c1 p1 s1 90.0 c1 p2 s2 80.0 c2 p1 s1 50.0 c3 p2 ss1 170.0 c2 p2 ss2 100.0

1. Check whether the above instantiation, once completed, is correct, and explain why it is or it is not.
2. Express in FOL the following queries and evaluate them over the completed instantiation:
 - (a) Return those providers that have contracts with at least two customers.
 - (b) Return those providers that have contracts only services they are specialized in.
 - (c) Return those providers that have contracts all services they are specialized in.
 - (d) Check whether there exists a customer with contracts for all services.

Exercise 3. Model check the Mu-Calculus formula $\nu X. \mu Y. ((a \wedge \langle \text{next} \rangle X) \vee (\neg b \wedge \langle \text{next} \rangle Y))$ and the CTL formula $AG(AF a \wedge EF b \wedge EG \neg b)$ (showing its translation in Mu-Calculus) against the following transition system:



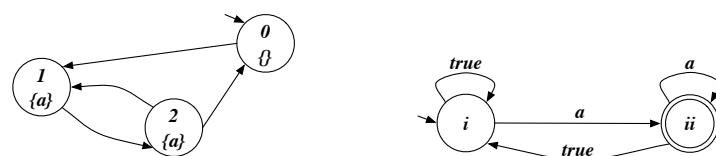
Exercise 4. Check whether CQ q_1 is contained in CQ q_2 , reporting canonical DBs and homomorphism:

$$\begin{aligned} q_1() &\leftarrow \text{edge}(r, g), \text{edge}(g, b), \text{edge}(b, r). \\ q_2() &\leftarrow \text{edge}(x, y), \text{edge}(y, z), \text{edge}(z, x), \text{edge}(z, v), \text{edge}(v, w), \text{edge}(w, z). \end{aligned}$$

Exercise 5. Check whether the following FOL formula is valid, by using tableaux:

$$(\forall x. \forall y. P(x, y) \supset Q(x)) \equiv (\forall x. (\exists y. P(x, y)) \supset Q(x))$$

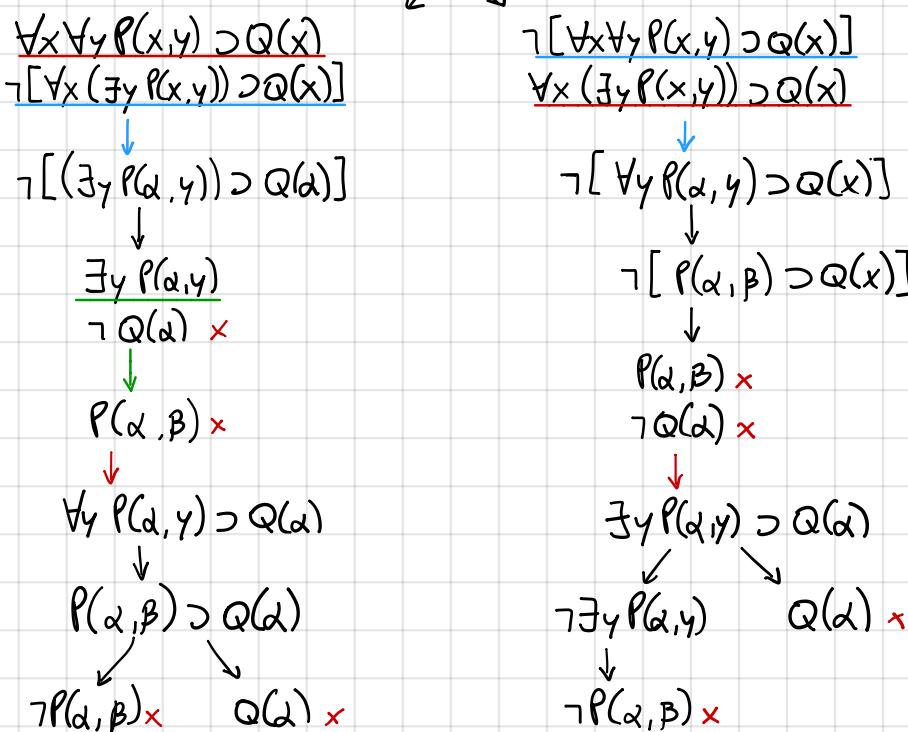
Exercise 6 (optional).¹ Model check the LTL formula $\diamond \square \neg a$ against the following transition system, by considering that the Büchi automaton for $\neg(\diamond \square \neg a)$ is the one below:



¹The student can get the maximum grade even without doing Exercise 6.

- 1) ALPHABET: $c(x)$, $s(x)$, $p(x)$, $ss(x)$, $\text{cont}_k(x, y, z)$, $\text{cost}(x, y, z, w)$, $\text{Red}(x)$
- ISA: $\forall x \ ss(x) \supset s(x)$
- SPECIALIZED IN: $\forall x, y \ \text{spec}(x, y) \supset p(x) \wedge s(x)$
 $\forall x \ p(x) \supset 1 \leq \#\{y \mid \text{spec}(x, y)\} \leq 5$
- CONTRACT: $\forall x, y, z \ \text{cont}_2(x, y, z) \supset c(x) \wedge p(y) \wedge s(z)$
 $\forall x, y, z, z' \ \text{cont}_2(x, y, z) \wedge \text{cont}_2(x, y, z') \supset z = z'$
 $\forall x, y, z \ \text{cont}_2(x, y, z) \supset 1 \leq \#\{w \mid \text{cost}(x, y, z, w)\} \leq 1$
- COST: $\forall x, y, z, w \ \text{cost}(x, y, z, w) \supset \text{cont}_2(x, y, z) \wedge \text{Red}(w)$
- 2) The completed instantiation is correct because all axioms are made true.
- $\exists x, x', z \ \text{cont}_2(x, y, z) \wedge \text{cont}_2(x', y, z) \wedge \neg(x = x')$
 $\Rightarrow \{p_1, p_2\}$
 - $\forall x \ \exists z \ \text{cont}_2(x, y, z) \supset \text{spec}(y, z)$
 $\Rightarrow \{p_1\}$
 - $\forall y \ \text{spec}(x, y) \supset \exists z \ \text{cont}_2(x, y, z)$
 $\Rightarrow \{p_2\}$
 - $\exists y \ \forall z \ s(z) \supset \text{cont}_2(x, y, z)$
 $\Rightarrow \{\emptyset\}$

5) $\neg ((\forall x \forall y p(x, y) \supset q(x)) \equiv (\forall x (\exists y p(x, y)) \supset q(x)))$



All branches close \rightarrow Unsat $\neg \phi \models \phi$ valid

$$3) \vee X \mu Y ((a \wedge c \rightarrow X) \vee (\neg b \wedge c \rightarrow Y))$$

$$[x_0] = S$$

$$[x_1] = \mu Y ((a \wedge c \rightarrow X) \vee (\neg b \wedge c \rightarrow Y)) = \{S_1, S_2, S_3\}$$

$$[y_{10}] = \{\emptyset\}$$

$$[y_{11}] = (a \wedge c \rightarrow x_0) \vee (\neg b \wedge c \rightarrow y_{10})$$

$$= ([a] \wedge \text{PreE}(-, [x_0])) \vee ([\neg b] \wedge \text{PreE}(-, [y_{10}]))$$

$$= \{S_2\} \cap \{S_4\} \cup (\{S_1, S_2, S_3\} \cap \{\emptyset\}) = \{S_2\}$$

$$[y_{12}] = ([a] \wedge \text{PreE}(-, [x_0])) \vee (\neg b \wedge \text{PreE}(-, [y_{11}]))$$

$$= \{S_2\} \cap \{S_4\} \cup (\{S_1, S_2, S_3\} \cap \{S_4\}) = \{S_1, S_2\}$$

$$[y_{13}] = ([a] \wedge \text{PreE}(-, [x_0])) \vee (\neg b \wedge \text{PreE}(-, [y_{12}]))$$

$$= \{S_2\} \cup (\{S_1, S_2, S_3\} \cap \{S_1, S_3, S_4\}) = \{S_1, S_2, S_3\}$$

$$[y_{14}] = ([a] \wedge \text{PreE}(-, [x_0])) \vee (\neg b \wedge \text{PreE}(-, [y_{13}]))$$

$$= \{S_2\} \cup (\{S_1, S_2, S_3\} \cap \{S_3\}) = \{S_1, S_2, S_3\}$$

→ greatest fixpoint

$$[x_2] = \mu Y ((a \wedge c \rightarrow X) \vee (\neg b \wedge c \rightarrow Y)) = \{S_1, S_2, S_3\}$$

$$[y_{20}] = \{\emptyset\}$$

$$[y_{21}] = (a \wedge c \rightarrow x_1) \vee (\neg b \wedge c \rightarrow y_{20})$$

$$= ([a] \wedge \text{PreA}(-, [x_1])) \vee ([\neg b] \wedge \text{PreE}(-, [y_{20}]))$$

$$= \{S_2\} \cap \{S_4\} \cup (\{S_1, S_2, S_3\} \cap \{\emptyset\}) = \{S_2\}$$

$$[y_{22}] = ([a] \wedge \text{PreA}(-, [x_1])) \vee (\neg b \wedge \text{PreE}(-, [y_{21}]))$$

$$= \{S_2\} \cup (\{S_1, S_2, S_3\} \cap \{S_1\}) = \{S_1, S_2\}$$

$$[y_{23}] = ([a] \wedge \text{PreE}(-, [x_1])) \vee (\neg b \wedge \text{PreE}(-, [y_{22}]))$$

$$= \{S_2\} \cup (\{S_1, S_2, S_3\} \cap \{S_1, S_3, S_4\}) = \{S_1, S_2, S_3\}$$

$$[y_{24}] = ([a] \wedge \text{PreE}(-, [x_1])) \vee (\neg b \wedge \text{PreE}(-, [y_{23}]))$$

$$= \{S_2\} \cup (\{S_1, S_2, S_3\} \cap \{S_4\}) = \{S_1, S_2, S_3\}$$

→ least fixpoint

Initial state in solution $\rightarrow T \models \phi$

$$AG(\underline{AF}_d \wedge \underline{EF}_b \wedge \underline{EG}_{\neg b})$$

$$\alpha$$

$$\beta$$

$$\gamma$$

$$\alpha = \mu X \alpha \vee \neg X = S$$

$$[x_0] = \emptyset$$

$$[x_1] = [a] \cup \text{PreA}(-, [x_0])$$

$$= \{S_2\} \cup \emptyset = \{S_2\}$$

$$[x_2] = [a] \cup \text{PreA}(-, [x_1])$$

$$= \{S_2\} \cup \{S_1\} = \{S_1, S_2\}$$

$$[x_3] = [a] \cup \text{PreA}(-, [x_2])$$

$$= \{S_2\} \cup \{S_1, S_4\} = \{S_1, S_2, S_4\}$$

$$[x_4] = [a] \cup \text{PreA}(-, [x_3])$$

$$= \{S_2\} \cup \{S_1, S_3, S_4\} = S \quad \text{least fixpoint}$$

$$\beta = \mu X b \vee \neg \rightarrow X = \{S_4\}$$

$$[x_0] = \emptyset$$

$$[x_1] = [b] \cup \text{PreE}(-, [x_0])$$

$$= \{S_4\} \cup \emptyset = \{S_4\}$$

$$[x_2] = [b] \cup \text{PreE}(-, [x_1])$$

$$= \{S_4\} \cup \{S_3\} = \{S_3, S_4\}$$

$$[x_3] = [b] \cup \text{PreE}(-, [x_2])$$

$$= \{S_4\} \cup \{S_2, S_3\} = \{S_2, S_3, S_4\}$$

$$[x_4] = [b] \cup \text{PreE}(-, [x_3])$$

$$= \{S_4\} \cup \{S_1, S_2, S_3\} = S \rightarrow \text{least fixpoint}$$

$$\gamma = \nu X \gamma b \wedge \neg \rightarrow X$$

$$[x_0] = S$$

$$[x_1] = [\gamma b] \cap \text{PreE}(-, [x_0])$$

$$= \{S_1, S_2, S_3\} \cap \{S_4\} = \{S_1, S_2, S_3\}$$

$$[x_2] = [\gamma b] \cap \text{PreE}(-, [x_1])$$

$$= \{S_1, S_2, S_3\} \cap S = \{S_1, S_2, S_3\}$$

} greatest fixpoint

$$AG(\alpha \wedge \beta \wedge \gamma) = AG(\gamma) = \nu X [\delta] \wedge [-]X$$

$$[x_0] = S$$

$$[x_1] = [\delta] \cap \text{PreA}(-, [x_0])$$

$$= \{S_1, S_2, S_3\} \cap \{S_4\} = \{S_1, S_2, S_3\}$$

$$[x_2] = [\delta] \cap \text{PreA}(-, [x_1])$$

$$= \{S_1, S_2, S_3\} \cap \{S_1, S_2, S_4\} = \{S_1, S_2\}$$

$$[x_3] = [\delta] \cap \text{PreA}(-, [x_2])$$

$$= \{S_1, S_2, S_3\} \cap \{S_1, S_4\} = \{S_1\}$$

$$[x_4] = [\delta] \cap \text{PreA}(-, [x_3])$$

$$= \{S_1, S_2, S_3\} \cap \{\emptyset\} = \{S_1\} \text{ least fixpoint}$$

Initial state not in solution $\rightarrow T \not\models \phi$

4) $q_1 \subseteq q_2$?

- freeze free variables (none)

- Build canonical interpretation I_{q_1} and I_{q_2} .

$$I_{q_1} = \begin{cases} \Delta^{I_{q_1}} = \{r, g, b\} \\ E^{I_{q_1}} = \{(r, g), (g, b), (b, r)\} \end{cases}$$

$$I_{q_2} = \begin{cases} \Delta^{I_{q_2}} = \{x, y, z, v, w\} \\ E^{I_{q_2}} = \{(x, y), (y, z), (z, x), (z, v), (v, w), (w, z)\} \end{cases}$$

- Check if $I_{q_1} \models I_{q_2} \rightarrow$ find homomorphism from I_{q_2} to I_{q_1} ,

- $h(x) = r$ at random

- $h(x, y) = (r, ?) \rightarrow h(y) = g$

- $h(y, z) = (g, ?) \rightarrow h(z) = b$

- $h(z, x) = (b, r) \rightarrow$ ok

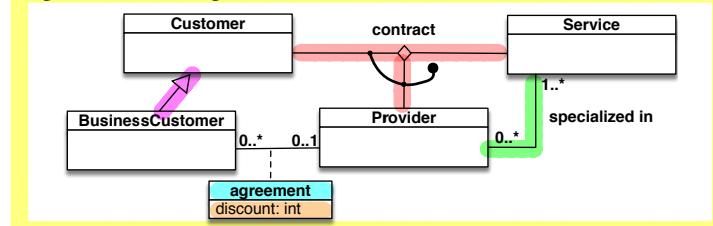
- $h(z, v) = (b, ?) \rightarrow h(v) = r$

- $h(v, w) = (r, ?) \rightarrow h(w) = g$

- $h(w, z) = (g, b) \rightarrow$ ok

Homomorphism exists $\Rightarrow q_1 \subseteq q_2$

Exercise 1. Express the following UML class diagram in FOL:

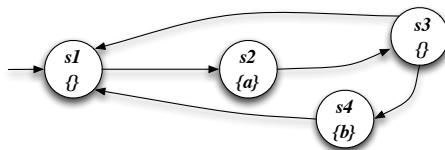


Exercise 2. Consider the above UML class diagram and the following (partial) instantiation:

Customer	BusiCustomer	Provider	agreement/disc	Service	specializedin	contacts
c1 c2	b1 b2	p1 p2	b1 p1 30	s1 s2 s3 s4 s5	p1 s1 p1 s2 p1 s3 p2 s4 p2 s5	c1 p1 s1 c1 p2 s2 c2 p1 s1 b1 p1 s4 b2 p2 s5

- Check whether the above instantiation, once completed, is correct, and explain why it is or it is not.
- Express in FOL the following queries and evaluate them over the completed instantiation:
 - Return those providers that are specialized in at least two services.
 - Return those business customers that have contracts only with providers with whom they have an agreement.
 - Return those business customers that have contracts with all providers with whom have an agreement .
 - Check whether there exists a customer with contracts for all services.

Exercise 3. Model check the Mu-Calculus formula $\nu X.\mu Y.((a \wedge \langle next \rangle X) \vee ([next] \neg b \wedge \langle next \rangle Y))$ and the CTL formula $EG(AFa \wedge (EFb \vee AG \neg b))$ (showing its translation in Mu-Calculus) against the following transition system:



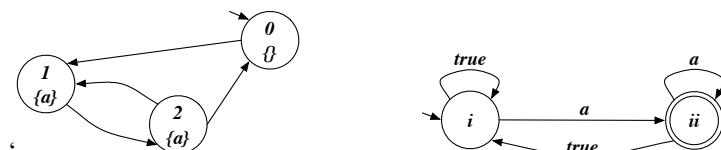
Exercise 4. Check whether the Hoare triple below is correct, by using $(x \geq 0 \wedge y \geq 0 \wedge x + y = 23)$ as invariant:

$$\{x = 23 \wedge y = 0\} \text{ while}(x > 0) \text{ do } (x = x - 1; y := y + 1) \{y = 23\}$$

Exercise 5. Check whether the following FOL formula is valid, by using tableaux:

$$(\forall x.(A(x) \equiv B(x))) \supset ((\forall y.A(y)) \equiv (\forall z.B(z)))$$

Exercise 6 (optional).¹ Model check the LTL formula $\diamond \square \neg a$ against the following transition system, by considering that the Büchi automaton for $\neg(\diamond \square \neg a)$ is the one below:



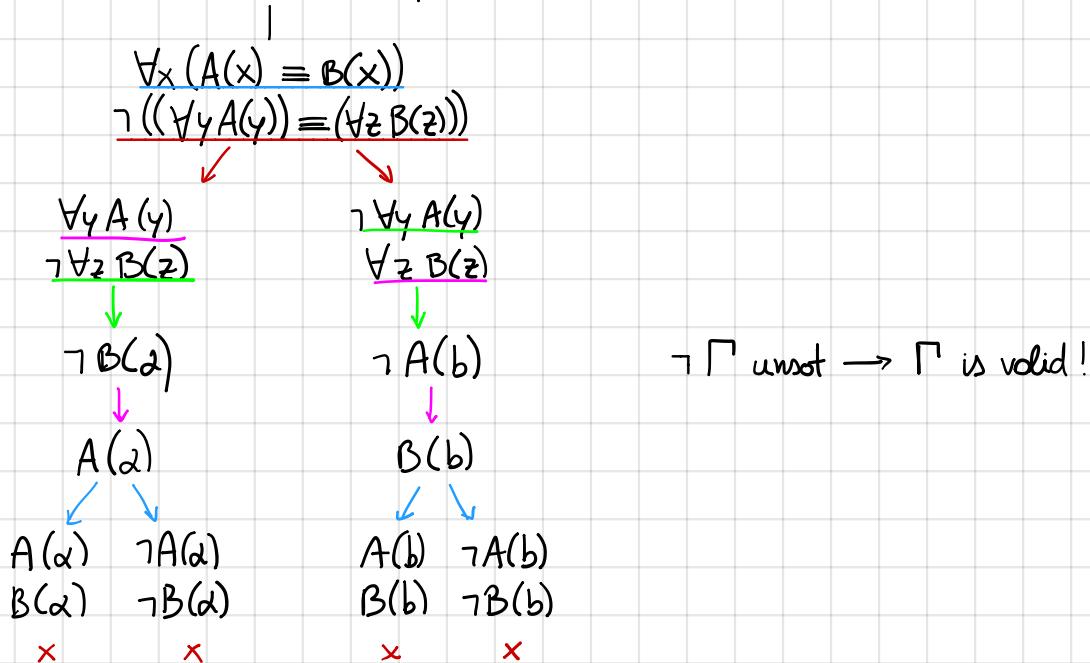
¹The student can get the maximum grade even without doing Exercise 6.

- 3) **Alphabet**: $c(x)$, $b(x)$, $p(x)$, $s(x)$, $\text{cont}_z(x,y,z)$, $\text{ogr}(x,y)$, $\text{discount}(x,y,z)$
- ISA**: $\forall x \ B(x) \supset c(x)$
- SPECIALIZED**: $\forall x, y \ \text{spec}(x,y) \supset p(x) \wedge s(y)$
 $\forall x \ P(x) \supset \{y \mid \text{spec}(x,y)\}$
- CONTRACT**: $\forall x, y, z \ \text{cont}_z(x,y,z) \supset c(x) \wedge p(y) \wedge s(z)$
 $\forall x, y, z, z' \ \text{cont}_z(x,y,z) \wedge \text{cont}_{z'}(x,y,z') \supset z = z'$
- AGREEMENT**: $\forall x, y \ \text{ogr}(x,y) \supset b(x) \wedge p(y)$
 $\forall x \ B(x) \supset \{y \mid \text{ogr}(x,y)\} \leq 1$
 $\forall x, y \ \text{ogr}(x,y) \supset \{z \mid \text{disc}(x,y,z)\} \leq 1$
- DISCOUNT**: $\forall x, y, z \ \text{disc}(x,y,z) \supset \text{ogr}(x,y) \wedge \text{Int}(z)$

2) The completed instantiation (choose ISA) is correct because all axioms of Γ are made true

- $\exists y y' \ \text{spec}(x,y) \wedge \text{spec}(x,y') \wedge y \neq y'$
 $\Rightarrow \{p_1, p_2\}$
- $\forall y \ \exists z \ \text{cont}_z(x,y,z) \supset \text{ogr}(x,y)$
 $\Rightarrow \{\emptyset\}$
- $\forall y \ \text{ogr}(x,y) \supset \exists z \ \text{cont}_z(x,y,z)$
 $\Rightarrow \{b\}$
- $\exists y \ \forall z \ s(z) \supset \text{cont}_z(x,y,z) \quad \forall z \ s(z) \supset \exists y \ \text{cont}_z(x,y,z)$
 $\Rightarrow \{\emptyset\}$

5) $\neg ((\forall x (A(x) \equiv B(x))) \supset ((\forall y A(y)) \equiv (\forall z B(z))))$



$$3) \vee X \mu X ((a \wedge \neg\neg X) \vee (\neg \neg b \wedge \neg\neg Y))$$

$$[X_0] = \{s_1, s_2, s_3, s_4\}$$

$$[X_1] = \mu Y ((a \wedge \neg\neg X) \vee (\neg \neg b \wedge \neg\neg Y)) = \{s_1, s_2, s_4\}$$

$$[Y_{10}] = \{\emptyset\}$$

$$[Y_{11}] = ((a \wedge \neg\neg X_0) \vee (\neg \neg b \wedge \neg\neg Y_{10})) = \\ (([a] \cap \text{PreE}(-, [X_0])) \cup (\text{PreA}(-, \neg b) \cap \text{PreE}(-, [Y_{10}])))$$

$$(\{s_2\} \cap \{s_1, s_2, s_3, s_4\}) \cup (\{s_1, s_2, s_4\} \cap \{\emptyset\}) = \{s_1\}$$

$$[Y_{12}] = (([a] \cap \text{PreE}(-, [X_0])) \cup (\text{PreA}(-, \neg b) \cap \text{PreE}(-, [Y_{11}]))) = \\ (\{s_2\} \cap \{s_1, s_2, s_3, s_4\}) \cup (\{s_1, s_2\} \cap \{s_1\}) = \{s_1, s_2\}$$

$$[Y_{13}] = (([a] \cap \text{PreE}(-, [X_0])) \cup (\text{PreA}(-, \neg b) \cap \text{PreE}(-, [Y_{12}]))) = \\ (\{s_2\} \cap \{s_1, s_2, s_3, s_4\}) \cup (\{s_1, s_2, s_4\} \cap \{s_1, s_3, s_4\}) = \{s_1, s_2, s_4\}$$

$$[Y_{14}] = (([a] \cap \text{PreE}(-, [X_0])) \cup (\text{PreA}(-, \neg b) \cap \text{PreE}(-, [Y_{13}]))) = \\ (\{s_2\} \cap \{s_1, s_2, s_3, s_4\}) \cup (\{s_1, s_2, s_4\} \cap \{s_1, s_3, s_4\}) = \{s_1, s_2, s_4\}$$

$$[X_2] = \mu Y ((a \wedge \neg\neg X) \vee (\neg \neg b \wedge \neg\neg Y)) = \{\emptyset\}$$

$$[Y_{20}] = \{\emptyset\}$$

$$[Y_{21}] = ((a \wedge \neg\neg X_1) \vee (\neg \neg b \wedge \neg\neg Y_{20})) = \\ (([a] \cap \text{PreE}(-, [X_1])) \cup (\text{PreA}(-, \neg b) \cap \text{PreE}(-, [Y_{20}]))) = \\ (\{s_2\} \cap \{s_1, s_3, s_4\}) \cup (\{s_1, s_2, s_4\} \cap \{\emptyset\}) = \{\emptyset\}$$

$$[X_3] = \mu Y ((a \wedge \neg\neg X) \vee (\neg \neg b \wedge \neg\neg Y)) = \{\emptyset\}$$

$$[Y_{30}] = \{\emptyset\}$$

$$[Y_{31}] = ((a \wedge \neg\neg X_2) \vee (\neg \neg b \wedge \neg\neg Y_{30})) = \\ (([a] \cap \text{PreE}(-, [X_2])) \cup (\text{PreA}(-, \neg b) \cap \text{PreE}(-, [Y_{30}]))) = \\ (\{s_2\} \cap \{\emptyset\}) \cup (\{s_1, s_2, s_4\} \cap \{\emptyset\}) = \{\emptyset\}$$

Initial state not in solution ($\{\emptyset\}$) $\rightarrow \tau \not\models \phi$ ($\phi = \mu$ -calculus formula)

$$\text{CTL: } EG(AF a \wedge \underbrace{(EF b \vee AG \neg b)}_{\beta})$$

$$\alpha = \vee X \neg b \wedge \neg X = \{\emptyset\}$$

$$[X_0] = \{s_1, s_2, s_3, s_4\}$$

$$[X_1] = \neg b \wedge \neg X_0 = [\neg b] \cap \text{PreA}(-, [X_0]) = \{s_1, s_2, s_3\} \cap \{s_1, s_2, s_3, s_4\} = \{s_1, s_2, s_3\}$$

$$[X_2] = [\neg b] \cap \text{PreA}(-, [X_1]) = \{s_1, s_2, s_3\} \cap \{s_1, s_2, s_4\} = \{s_1, s_2\}$$

$$[X_3] = [\neg b] \cap \text{PreA}(-, [X_2]) = \{s_1, s_2, s_3\} \cap \{s_1, s_4\} = \{s_1\}$$

$$[X_4] = [\neg b] \cap \text{PreA}(-, [X_3]) = \{s_1, s_2, s_3\} \cap \{s_1\} = \emptyset \text{ fixpoint! } (X_5 \text{ will be } \emptyset)$$

$$\beta = \mu X b \vee \neg\neg X = \{s_1, s_2, s_3, s_4\}$$

$$[X_0] = \emptyset$$

$$[X_1] = [b] \cup \text{PreE}(-, [X_0]) = \{s_4\} \cup \{\emptyset\} = \{s_4\}$$

$$[X_2] = [b] \cup \text{PreE}(-, [X_1]) = \{s_4\} \cup \{s_1\} = \{s_3, s_4\}$$

$$[X_3] = [b] \cup \text{PreE}(-, [X_2]) = \{s_4\} \cup \{s_2, s_3\} = \{s_2, s_3, s_4\}$$

$$[X_4] = [b] \cup \text{PreE}(-, [X_3]) = \{s_4\} \cup \{s_1, s_2, s_3\} = \{s_1, s_2, s_3, s_4\} \text{ fixpoint! } (X_5 \text{ will be equal } / S)$$

$$EG(AF a \wedge (S \vee \phi)) = EG(AF a \wedge S) = EG(AF a) \text{ because } \wedge S \rightarrow \wedge S \rightarrow \text{intersection with all states is meaningless.}$$

$$AF_d = \mu X_d \vee [-]X = FS$$

$$[X_0] = \emptyset$$

$$[X_1] = [d] \cup \text{PreA}(-, X_0) = \{S_2\}$$

$$[X_2] = [d] \cup \text{PreA}(-, X_1) = \{S_2\} \cup \{S_1\} = \{S_1, S_2\}$$

$$[X_3] = [d] \cup \text{PreA}(-, X_2) = \{S_2\} \cup \{S_1, S_4\} = \{S_1, S_2, S_4\}$$

$$[X_4] = [d] \cup \text{PreA}(-, X_3) = \{S_2\} \cup \{S_1, S_3, S_4\} = \{S\} \text{ fixpoint}$$

$$EG(AF_d) = EG(S) = \forall X S \wedge \leftarrow X = \{S\}$$

$$[X_0] = S$$

$$[X_1] = \{S\} \cup \text{Pre}(-, [X_0]) = \{S\} \text{ fixpoint}$$

Initial state in solution $\rightarrow T \not\models \phi$

4) $I = \{x \geq 0 \wedge y \geq 0 \wedge x+y=23\}$

$$P = \{x=23 \wedge y=0\}$$

$$Q = \{y=23\}$$

$$\delta = \{x = x-1; y = y+1\}$$

$$G = \{x > 0\}$$

- check $P \supset I$

$$x = 23 \wedge y = 0 \supset x \geq 0 \wedge y \geq 0 \wedge x+y=23 \quad \checkmark$$

- check $\neg g \wedge I \supset Q$

$$x \leq 0 \wedge x \geq 0 \wedge y \geq 0 \wedge x+y=23 \supset y=23 \quad \checkmark$$

$x=0$

- check $\{g \wedge I\} \delta \{I\} = g \wedge I \supset w_p(\delta, I)$

$$x \geq 0 \wedge x \geq 0 \wedge y \geq 0 \wedge x+y=23 \supset w_p(\delta, I)$$

$\hookrightarrow x \geq 0 \wedge x \geq 0 \wedge y \geq 0 \wedge x+y=23 \supset x-1 \geq 0 \wedge y+1 \geq 0 \wedge x+y=23$

$$\{x-1 \geq 0 \wedge y+1 \geq 0 \wedge x+y=23\}$$

$$x = x-1$$

$$\{x \geq 0 \wedge y+1 \geq 0 \wedge x+y+1=23\}$$

$$y = y+1$$

$$I \rightsquigarrow \{x \geq 0 \wedge y \geq 0 \wedge x+y=23\}$$

I is an invariant so the Hoare triple is correct!

