Formal Methods in software development

a.y.2017/2018

Prof. Anna Labella

The predicate calculus [H-R ch.2]

- The need for a richer language
- Terms and formulas

- quantifiers

Syntax

- Inductive term definition
- BNF
- t::= x | c | f(t_1,t_2,…,t_n)

Syntax

- Inductive definition of wff
- BNF

$\varphi ::= p(t_1, t_2, \ldots, t_n) \mid t_1 = t_2 \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid$
$(\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid \forall x \varphi \mid \exists x \varphi$

Semantics

syntactical data

- $\mathcal{F}$ *functional symbols- constants*
- $\mathcal{P}$ *predicate symbols*

An interpretation $\mathcal{M}$

- a non empty set (domain) A
- $\mathcal{F}$ ➔ a set of functions $f^{\mathcal{M}}$ on A (A^n)
- $\mathcal{P}$ ➔ a set of relations $P^{\mathcal{M}}$ on A (A^n)

Semantics

A subset is the interpretation of a 1-ary predicate

A n-ary relation is the interpretation of a n-ary predicate

An example: arithmetics

syntactical data

- $\mathcal{F}$: $f_1(-,-)$, $f_2(-,-)$, $f_0(-)$, c
- $\mathcal{P}$: $- = -$, $P_1(-,-)$

An interpretation $\mathcal{M}$

- Natural numbers (domain)
- Functions : - + -, - · -, s(-), 0
- Predicates: - = -, - ≤ -

Semantics

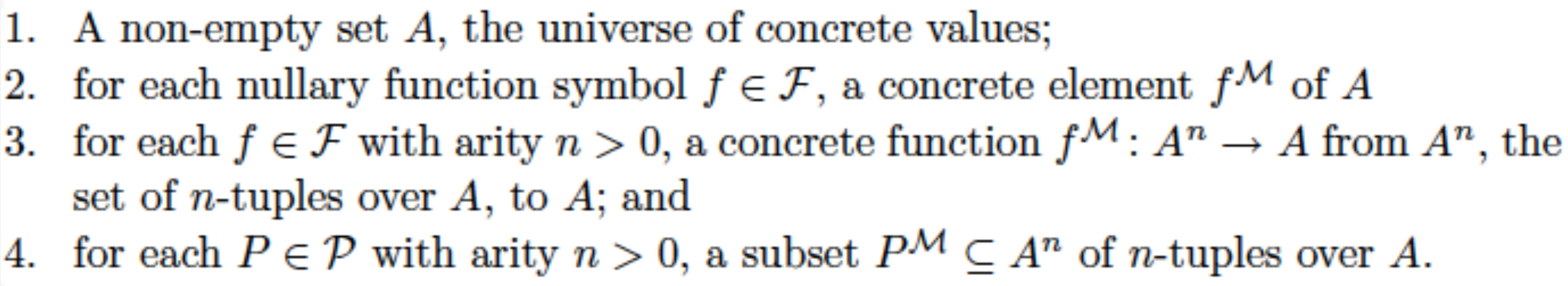
environments

Assigning values

- ℓ: *var* → elements of A

Semantics

Summing up, we are given with



1. A non-empty set A, the universe of concrete values;
2. for each nullary function symbol $f \in \mathcal{F}$, a concrete element $f^{\mathcal{M}}$ of A
3. for each $f \in \mathcal{F}$ with arity $n > 0$, a concrete function $f^{\mathcal{M}} \colon A^n \to A$ from A^n, the set of n-tuples over A, to A; and
4. for each $P \in \mathcal{P}$ with arity $n > 0$, a subset $P^{\mathcal{M}} \subseteq A^n$ of n-tuples over A.

An example: arithmetics 2

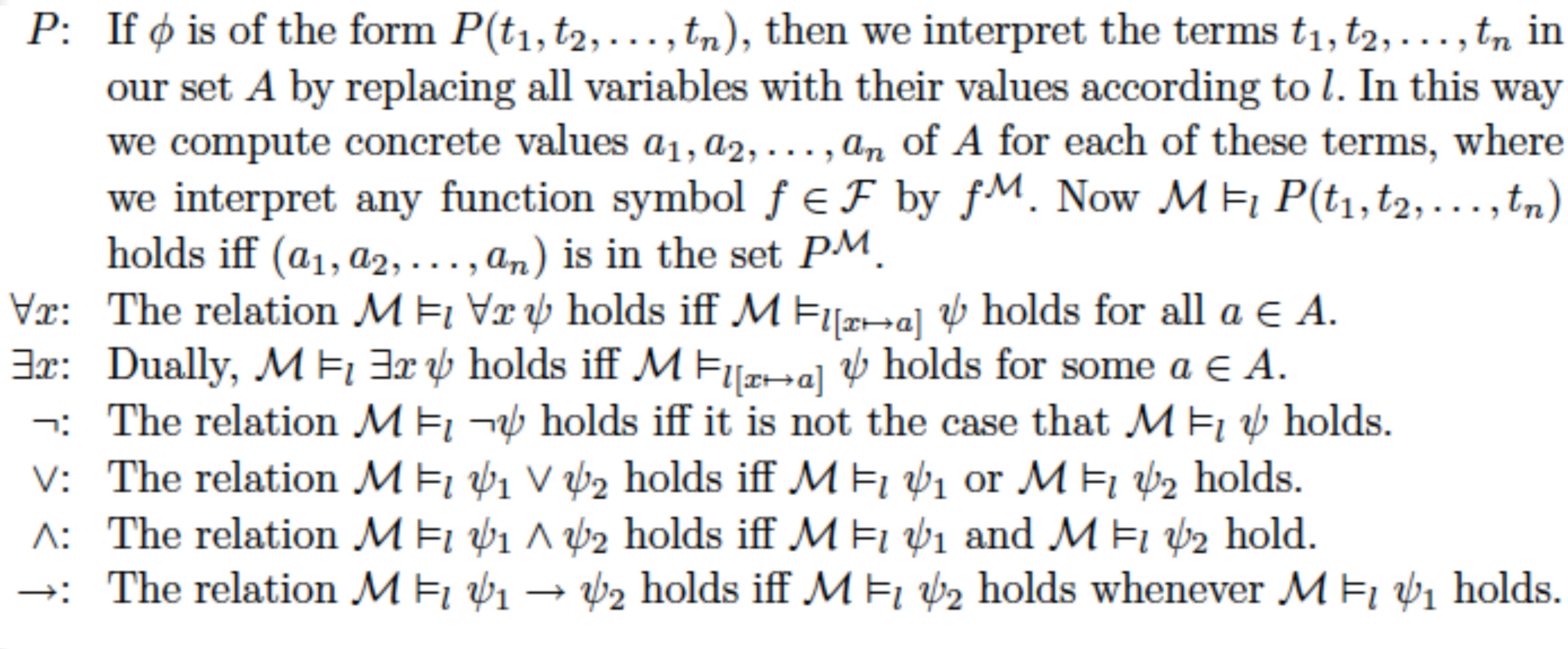
- $f_1(c,x)=x$ *always true in the interpretation*
- $f_2(c,x)=x$ *sometimes true sometimes false in the interpretation depending on the assignment*
- $\forall x(f_1(c,x)=x)$ *true in the interpretation*
- $\exists x(f_2(c,x)=x)$ *true in the interpretation*
- $\forall x(f_2(c,x)=x)$ *false in the interpretation*
- $\forall x(f_1(c,x)=f_1(c,x))$ *valid*

Semantics

An interpretation $\mathcal{M}$

is a model of φ $\mathcal{M} \models_{\mathcal{L}} \varphi$:

Semantics: satisfiability



P: If ϕ is of the form $P(t_1, t_2, \ldots, t_n)$, then we interpret the terms $t_1, t_2, \ldots, t_n$ in our set A by replacing all variables with their values according to l. In this way we compute concrete values $a_1, a_2, \ldots, a_n$ of A for each of these terms, where we interpret any function symbol $f \in \mathcal{F}$ by $f^{\mathcal{M}}$. Now $\mathcal{M} \vDash_l P(t_1, t_2, \ldots, t_n)$ holds iff $(a_1, a_2, \ldots, a_n)$ is in the set $P^{\mathcal{M}}$.

$\forall x$: The relation $\mathcal{M} \vDash_l \forall x\, \psi$ holds iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for all $a \in A$.

$\exists x$: Dually, $\mathcal{M} \vDash_l \exists x\, \psi$ holds iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for some $a \in A$.

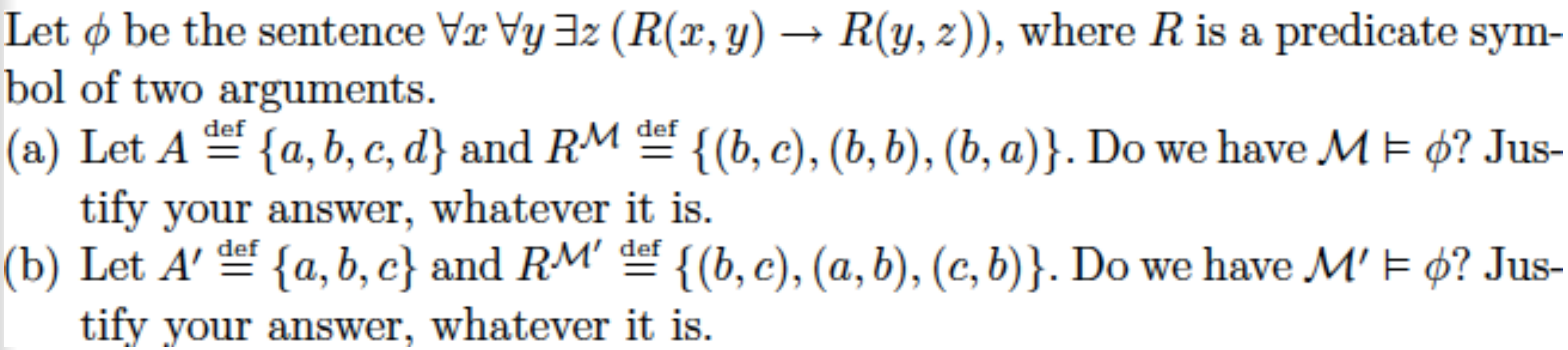
$\neg$: The relation $\mathcal{M} \vDash_l \neg\psi$ holds iff it is not the case that $\mathcal{M} \vDash_l \psi$ holds.

$\vee$: The relation $\mathcal{M} \vDash_l \psi_1 \vee \psi_2$ holds iff $\mathcal{M} \vDash_l \psi_1$ or $\mathcal{M} \vDash_l \psi_2$ holds.

$\wedge$: The relation $\mathcal{M} \vDash_l \psi_1 \wedge \psi_2$ holds iff $\mathcal{M} \vDash_l \psi_1$ and $\mathcal{M} \vDash_l \psi_2$ hold.

$\rightarrow$: The relation $\mathcal{M} \vDash_l \psi_1 \rightarrow \psi_2$ holds iff $\mathcal{M} \vDash_l \psi_2$ holds whenever $\mathcal{M} \vDash_l \psi_1$ holds.

Exercises



Let ϕ be the sentence $\forall x\, \forall y\, \exists z\, (R(x, y) \to R(y, z))$, where R is a predicate symbol of two arguments.

(a) Let $A \stackrel{\text{def}}{=} \{a, b, c, d\}$ and $R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(b, c), (b, b), (b, a)\}$. Do we have $\mathcal{M} \vDash \phi$? Justify your answer, whatever it is.

(b) Let $A' \stackrel{\text{def}}{=} \{a, b, c\}$ and $R^{\mathcal{M}'} \stackrel{\text{def}}{=} \{(b, c), (a, b), (c, b)\}$. Do we have $\mathcal{M}' \vDash \phi$? Justify your answer, whatever it is.

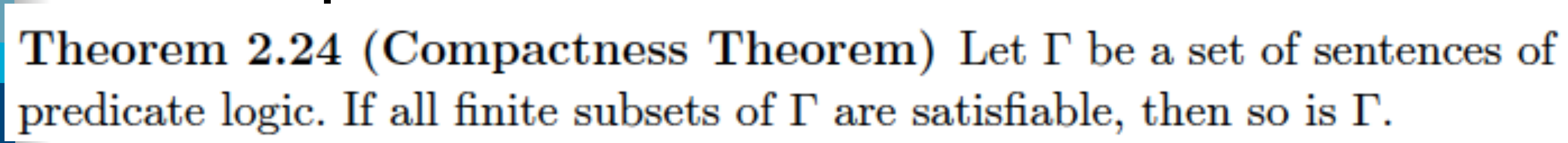
Semantics: validity

φ is valid if for every intepretation and for every environment

$$\mathcal{M} \models \varphi$$

Properties

- Soundness Γ |- φ ➔ $\mathcal{M}$ |= φ
- Completeness $\mathcal{M}$ |= φ ➔ Γ |- φ
- Indecidability
- Compactness



Theorem 2.24 (Compactness Theorem) Let Γ be a set of sentences of predicate logic. If all finite subsets of Γ are satisfiable, then so is Γ.

- Expressivity

Second order logic

- Existential second order logic
- $\exists P \phi$
- Universal second order logic
- Peano's arithmetics

Specification, verification and logics

[H-R ch.3]

Logic provides:

- A framework for modelling systems
- A specification language for describing properties to be verified
- A verification method to ascertain whether the description of the system satisfies the properties

Possibilities of approaching model verification

- Proof-based

Γ |-- φ

Γ is the description while φ is the property to be satisfied

- Model based

$\mathcal{M}$ |= φ

$\mathcal{M}$ is a finite model (only one)

- Degree of automation:

Fully automatic

Manual

- Full behaviour
- Sequential
- Reactive

….

- A priori

- One property
- Concurrent
- Terminating

…..

- A posteriori

We are possibly dealing with

- Γ |-- φ proof theory
- Γ |= φ semantic entailment
- $\mathcal{M}$ |= φ satifiability

We are possibly dealing with

- ~~$\Gamma \vdash \varphi$~~
- $\Gamma \models \varphi$
- $\mathcal{M} \models \varphi$

We are possibly dealing with

- ~~$\Gamma \vdash \varphi$~~
- ~~$\Gamma \models \varphi$~~
- $\mathcal{M} \models \varphi$

Model Checking

- Automatic
- Based on a builded model
- Verifying satisfiability of properties
- A posteriori
- Provides a counterexample

- Concurrent systems
- Reactive systems
- Temporal aspects

A formula can change its truth value

- We build a model $\mathcal{M}$

- We model our system using the description language of the model checker
- We code the property to be verified in the same language and the model checker should say whether $\mathcal{M}$ |= φ or not

- Time could change the truth value of a formula

- $\mathcal{M}$,s |= φ or not for a given state s
- In this last case it is often possible using the model checker to have a counterexample

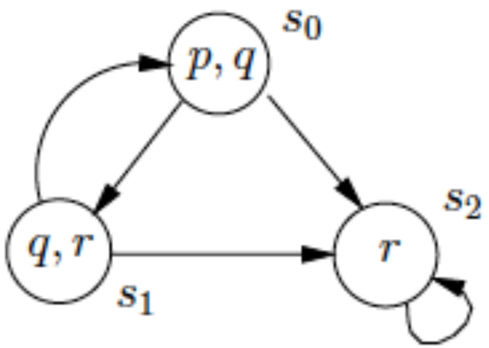
Models and states

- A model $\mathcal{M}$ is an abstraction: it can describe very different things and omits lot of particulars
- A model $\mathcal{M}$ is a transition system
- We have states and and transitions between them. An assignment statement can make the model move from one state to another one
- We can think of a transition system as a set S of states together with a binary relation

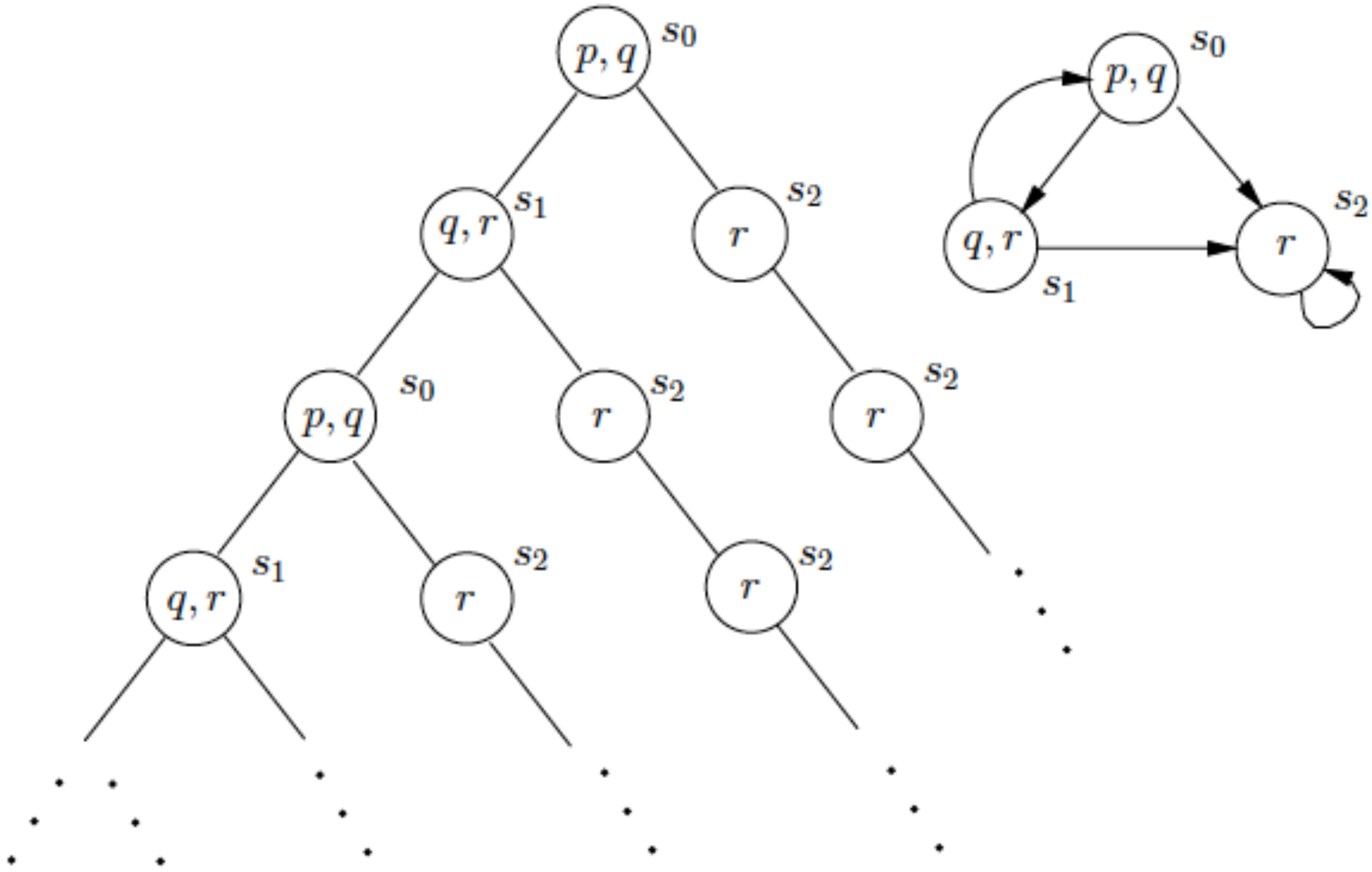
$$\rightarrow \; \subseteq \; S \times S$$

and a labeling function L: $S \rightarrow \mathcal{P}(\text{atoms})$

A transition system 1

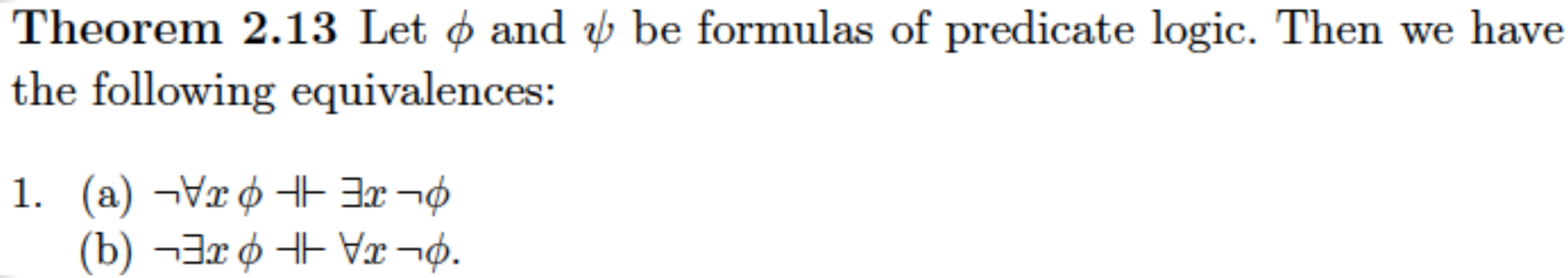


A transition system 2



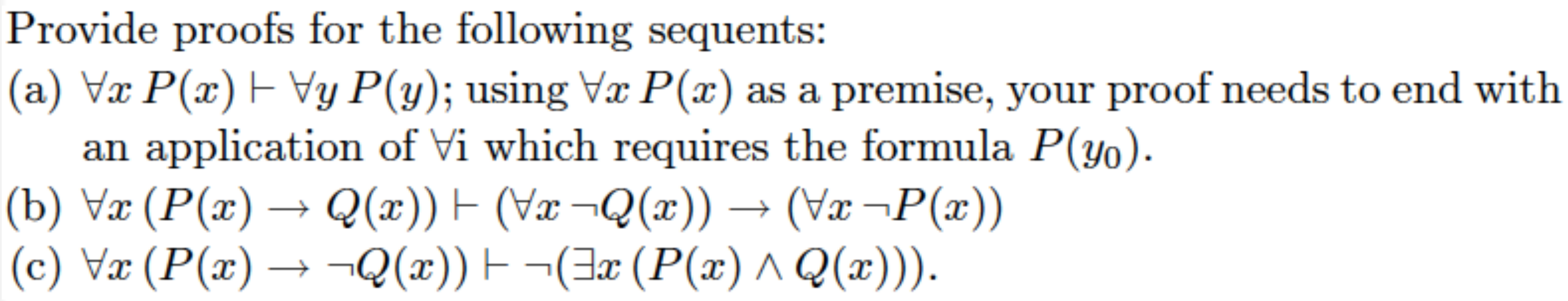
Linear and branching time

Exercises



Theorem 2.13 Let ϕ and ψ be formulas of predicate logic. Then we have the following equivalences:

1. (a) $\neg\forall x\,\phi \dashv\vdash \exists x\,\neg\phi$
   (b) $\neg\exists x\,\phi \dashv\vdash \forall x\,\neg\phi$.



Provide proofs for the following sequents:

(a) $\forall x\,P(x) \vdash \forall y\,P(y)$; using $\forall x\,P(x)$ as a premise, your proof needs to end with an application of $\forall$i which requires the formula $P(y_0)$.

(b) $\forall x\,(P(x) \to Q(x)) \vdash (\forall x\,\neg Q(x)) \to (\forall x\,\neg P(x))$

(c) $\forall x\,(P(x) \to \neg Q(x)) \vdash \neg(\exists x\,(P(x) \wedge Q(x)))$.