Logics

Logics - Introduction

- So far we have seen a variety of operational formalisms
  - based on some some notion of state and event/transition
  - model the evolution of the system
- Now instead we will analyze (mathematical) logics, which are instead a typical *descriptive* formalism
  - logics are better suited to describe the *properties* of systems
  - they do not rely on a "built-in" notion of "evolution" of the system
    - they can be used to describe it, also, but it must be "encoded" into them
  - for example, we can use a logic formalism to model properties such as "if the train enters the railroad crossing, the gate must have been closed for at least K instants"
  - it is often the case that the "how" (i.e. the evolution of the system) is modeled through an operational formalism, while the desired properties that must hold for it are described through some sort of logic formalism
    - a "dual-language" approach
- There are many kinds of logics, with varying modeling power and features
  - *propositional* logic
  - *first-order* logic
  - logics with a notion of time
    - i.e. *temporal* logics
      - they can be first-order or propositional...
  - ... many sorts of “descriptive logics”
    - there are logics to describe almost anything you can think of...

Logics in this course

- There is a vast variety of approaches/uses of logics
  - logics have been studied for millennia (since Aristotle!)
  - there are entire courses devoted to the study of logics and their properties
- In this course, we will content ourselves with studying the very basic concepts of logics, those that will then allow us to use logics to effectively model significant properties of dynamic systems
- More precisely, we will analyze the following topics:
  - basic concepts of propositional logic
  - basic concepts of first-order logic
  - how to use (first-order) logic to model (dynamic) systems

- As usual, no textbook covers *precisely* the notions of logics that we will see in this course
  - however, there are many books that provide introductory courses to mathematical logics, for example:
    - E. Mendelson, *Introduction to Mathematical Logic*,
    - P. B. Andrews, *An introduction to mathematical logic and type theory: to truth through proofs*
      - they provide *way more* notions than we will actually see in this course, but are both very good, classic introductory books on mathematical logics
      - in other words, the aforementioned books go beyond the scope of logics as presented in this course, but they are useful references if you are interested in studying logics in greater depth

Propositional logic

- Propositional logic is the simplest kind of logic
  - it has only few key elements, very simple
  - its simplicity allows to perform highly automated analyses
  - it has limited expressive power
    - propositional logic is to logics what finite-state automata are to automata-based formalisms
- Propositional logic has the following key elements:
  - propositions
  - connectives
- Basically, propositions represent basic "facts", which are combined through connectives to form "statements" or *formulas*
  - a proposition (i.e. a fact) can be either true or false (i.e. it can hold or not)
  - a formula (i.e. a statement) can be also true or false, depending on the truth/falsity of the propositions which compose it
- (Informal) Examples of propositions:
  - "Paris is the capital of France"
  - "London is the capital of India"
  - "John is Italian"
  - "Socrates is human"
  - "the light is on"
  - "the window is open"
- (Informal) Examples of formulas:
  - John is Italian **and** the light is on
  - London is the capital of India **implies that** Socrates is human
  - Paris is the capital of France **and** John is Italian, **or** the window is open

Propositional logic: definition

- The elements of propositional logic are the following:
  - a set of *propositional letters*
    - the set of propositional letters is *at most denumerable*
      - In fact, we will always use finite sets
    - Examples of propositional letters:
      A, B, C, A_1, C_4
      - however, we will also use propositional "letters" such as light_on, window_open, gate_up, etc.
  - the following logical connectives:
    ~ (not), $\wedge$ (and), $\vee$ (or), $\rightarrow$ (implies), $\leftrightarrow$ (if and only if)
  - the parentheses: (,)
    - parentheses are so-called *auxiliary symbols*
- A formula is a combination of propositional letters, connectives, and parentheses
  - not all combinations of propositional letters, connectives and parentheses are formulas!
    - e.g. "A ~) B ($\vee$ $\rightarrow$" is *not* a formula!
- *Well-formed formulas* (wff for short) are all the valid combinations of propositional letters, connectives and parentheses:
- The set of wffs is defined *recursively*:
  - a single propositional letter is a wff
  - if $\mathcal{F}$ is a wff, ~$\mathcal{F}$ is also a wff
  - if $\mathcal{F}_1$ and $\mathcal{F}_2$ are wffs, the following are also wffs:

    $(\mathcal{F}_1 \wedge \mathcal{F}_2)$

    $(\mathcal{F}_1 \vee \mathcal{F}_2)$

    $(\mathcal{F}_1 \rightarrow \mathcal{F}_2)$

    $(\mathcal{F}_1 \leftrightarrow \mathcal{F}_2)$
  - nothing else is a wff

Examples of wffs

- Given the following propositional letters,
  A, B, C, F, G, J, window_down, window_up, push, light_on
  the following are wffs:

  $(A \rightarrow (\sim B)) \leftrightarrow (C \wedge A)$

  $(F \vee G) \vee (J \vee J)$

  window_down → ~window_up

  push → light_on

- To simplify writing (well-formed) formulas, we introduce an order of precedence between connectives:

  $\sim, \wedge, \vee, \rightarrow, \leftrightarrow$

  - that is, ~ precedes ∧, which precedes ∨, etc.
  - the connectives are associative to the left
    - that is, $A \rightarrow B \rightarrow C$ corresponds to $(A \rightarrow B) \rightarrow C$

- For example, writing $(A \rightarrow (\sim B)) \leftrightarrow (C \wedge A)$ is the same as
  $A \rightarrow \sim B \leftrightarrow C \wedge A$
  - however, $(A \rightarrow (\sim B \leftrightarrow C) \wedge A)$ is *not* the same!

- Suppose now we wanted to formalize the formulas of slide 4
  - let us introduce the following propositional letters:
    - PcF, for "Paris is the capital of France"
    - LcI for "London is the capital of India"
    - JisI for "John is Italian"
    - SisH for "Socrates is human"
    - light_on for "the light is on"
    - window_open for "the window is open"
  - the formulas become the following:
    - JisI ∧ light_on
    - LcI → SisH
    - PcF ∧ JisI ∨ window_open
      - that is, (PcF ∧ JisI) ∨ window_open

Semantics of propositional logic (1)

- We now study how the truth/falsity of a wff varies with the truth/falsity of the propositional letters it includes
  - this corresponds to defining the *semantics* of propositional logic
    - i.e. when formulas are true or false
- In the following, we represent by T the value "true", and by F the value "false"
  - then, given the propositional letter light_on, its value can be either T or F
- An *assignment s* is a mapping from propositional letters to the set {T, F}
  - for example, given propositional letters A, B, C, the following are possible assignments:

    s_1: {A $\mapsto$ T, B $\mapsto$ T, C $\mapsto$ F}

    s_2: {A $\mapsto$ F, B $\mapsto$ F, C $\mapsto$ F}

    - if s is an assignment and A is a propositional letter, by s(A) we mean the value of A in s
      - e.g. s_1(B) = T, s_2(B) = F
- In general, then, the truth/falsity of a wff depends on the truth/falsity of the propositional letters
  - given a wff formula $\mathcal{G}$, by $\mathcal{V}_s(\mathcal{G})$ we mean the value of $\mathcal{G}$ with respect to assignment s

Semantics of propositional logic (2)

- if A is a propositional letter, while both $\mathcal{G}$ and $\mathcal{H}$ are wffs, then $\mathcal{V}_s$ is defined as follows:
  - $\mathcal{V}_s(\mathrm{A}) = s(\mathrm{A})$
  - $\mathcal{V}_s(\sim\mathcal{G}) = \mathsf{T}$ if and only if $\mathcal{V}_s(\mathcal{G}) = \mathsf{F}$
  - $\mathcal{V}_s(\mathcal{G} \wedge \mathcal{H}) = \mathsf{T}$ if and only if $\mathcal{V}_s(\mathcal{G}) = \mathsf{T}$ and $\mathcal{V}_s(\mathcal{H}) = \mathsf{T}$
- The value of the other connectives ($\vee$, $\rightarrow$, $\leftrightarrow$) are defined from ~, $\wedge$ in the following way:
  - $\mathcal{V}_s(\mathcal{G} \vee \mathcal{H}) = \mathcal{V}_s(\sim(\sim\mathcal{G} \wedge \sim\mathcal{H}))$
    - that is, $\mathcal{V}_s(\mathcal{G} \vee \mathcal{H}) = \mathsf{F}$ if and only if both $\mathcal{V}_s(\mathcal{G}) = \mathsf{F}$ and $\mathcal{V}_s(\mathcal{H}) = \mathsf{F}$
  - $\mathcal{V}_s(\mathcal{G} \rightarrow \mathcal{H}) = \mathcal{V}_s(\sim\mathcal{G} \vee \mathcal{H})$
  - $\mathcal{V}_s(\mathcal{G} \leftrightarrow \mathcal{H}) = \mathcal{V}_s((\mathcal{G} \rightarrow \mathcal{H}) \wedge (\mathcal{H} \rightarrow \mathcal{G}))$
- Examples:
  - given assignment s_1: {A $\mapsto$ T, B $\mapsto$ T, C $\mapsto$ F}, $\mathcal{V}_{s_1}(\mathrm{A} \rightarrow \sim\mathrm{B} \leftrightarrow \mathrm{C} \wedge \mathrm{A}) = \mathsf{T}$
    - instead, given assignment s_2: {A $\mapsto$ F, B $\mapsto$ F, C $\mapsto$ F}, $\mathcal{V}_{s_2}(\mathrm{A} \rightarrow \sim\mathrm{B} \leftrightarrow \mathrm{C} \wedge \mathrm{A}) = \mathsf{F}$
    - notice that $\mathcal{V}_{s_2}(\mathrm{A} \rightarrow (\sim\mathrm{B} \leftrightarrow \mathrm{C}) \wedge \mathrm{A}) = \mathsf{T}$
  - given w: {window_down $\mapsto$ T, window_up $\mapsto$ T}, then $\mathcal{V}_w$(window_down $\rightarrow$ ~window_up) = F
  - given ℓ: {LcI $\mapsto$ F, SisH $\mapsto$ F}, then $\mathcal{V}_\ell$(LcI $\rightarrow$ SisH) = T
    - also, given ℓ: {LcI $\mapsto$ F, SisH $\mapsto$ V}, then $\mathcal{V}_\ell$(LcI $\rightarrow$ SisH) = T
    - that is, if London is not the capital of India, then the implication LcI $\rightarrow$ SisH is true (NB: independent of whether Socrates is human or not)

Truth tables

- A more compact way of representing the truth of a formula depending on the truth of the propositional letters is through *truth tables*
- For example:

A	B	C	$(A \rightarrow \sim B) \leftrightarrow (C \wedge A)$
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	F

- Another example of truth table:

A	B	$A \rightarrow B \vee A$
T	T	T
T	F	T
F	T	T
F	F	T

- Notice that a formula can contain at most a finite number of propositional letters, so the truth table of a formula is always finite
  - it can be very big, if the number of propositional letters is high, but it is always finite
  - in fact, if *n* is the number of propositional letters that appear in a formula $\mathcal{G}$, the size of the truth table of $\mathcal{G}$ is 2^n
    - i.e., it is *exponential* in the number of propositional letters of the formula!

Models, tautologies, contradictions

- Given a formula $\mathcal{G}$, an assignment s is a *model* for $\mathcal{G}$ if and only if $\mathcal{V}_s(\mathcal{G}) = \mathsf{T}$
  - for example, s_1: $\{A \mapsto \mathsf{T}, B \mapsto \mathsf{T}, C \mapsto \mathsf{F}\}$ is a model for $A \rightarrow \sim B \leftrightarrow C \wedge A$
- A formula $\mathcal{G}$ is *satisfiable* if and only if it has **at least** one model
  - both $A \rightarrow \sim B \leftrightarrow C \wedge A$ and $A \rightarrow B \vee A$ are satisfiable
- A formula $\mathcal{G}$ is a *tautology* if and only if **every** assignment s is a model
  - e.g. $A \rightarrow B \vee A$ is a tautology
  - another example of tautology is $A \vee \sim A$
  - to indicate that formula $\mathcal{A}$ is a tautology we will write $\models \mathcal{A}$
    - e.g. $\models A \rightarrow B \vee A$
- A formula $\mathcal{G}$ is a *contradiction* if and only if **no** assignment s is a model
  - that is, if and only if, for any assignment, its value is F
  - notice that a formula $\mathcal{G}$ is a contradiction if and only if its negation is a tautology
    - in other words, $\mathcal{G}$ is a contradiction if and only if $\models \sim\mathcal{G}$
  - an example of contradiction is $A \wedge \sim A$
    - its negation is $\sim(A \wedge \sim A)$, which is the same as $\sim A \vee A$, which is in fact a tautology!

Logical consequence

- Let us now consider a set Γ of wffs
  - for example, Γ_w = {window_down → ~window_up, window_down ∨ window_up}
- A *model* for a set Γ is an assignment that is a model for **all** wffs in Γ
  - e.g., {window_up ↦ T, window_down ↦ F} is a model for Γ_w
  - {window_up ↦ T, window_down ↦ T}, instead, is not a model for Γ_w
- A set Γ of wffs is *satisfiable* if and only if it has **at least** one model
  - Γ_w is satisfiable
- A set Γ is *contradictory* if and only if it has **no** models

- We say that a formula $\mathcal{G}$ is a *logical consequence* of a set Γ of wffs if and only if every model of Γ is also a model of $\mathcal{G}$
  - basically, if and only if the models of Γ are a subset of the models of $\mathcal{G}$
    - for example, ~(window_down ∧ window_up) is a logical consequence of Γ_w
  - to indicate that $\mathcal{G}$ is a logical consequence of Γ we write $\Gamma \models \mathcal{G}$

Two fundamental theorems

- **Theorem**: Given a set Γ of wffs such that $\Gamma = \Delta \cup \{\mathcal{H}\}$ (i.e., such that Γ contains a wff $\mathcal{H}$, and $\Gamma - \{\mathcal{H}\} = \Delta$), then
  $\Gamma \models \mathcal{G}$ if and only if $\Delta \models \mathcal{H} \rightarrow \mathcal{G}$
  - for example, $\Gamma_w \models$ ~(window_down ∧ window_up), that is,
    {window_down ∨ window_up}⊨
    (window_down → ~window_up) → ~(window_down ∧ window_up)
    - also,
      ⊨ (window_down ∨ window_up) →
      ((window_down → ~window_up) →
      ~(window_down ∧ window_up))
      - that is, the chain of implications is a tautology
- **Theorem**: $\Gamma \models \mathcal{G}$ if and only if $\Gamma \cup \{\sim\mathcal{G}\}$ is contradictory
  - this property is used in some (widely used and highly automated) verification techniques (the so-called SAT-based techniques) to show that a formula is a logical consequence of a set of hypotheses (i.e. of other formulas)

Logical equivalence

- We say that two formulas $\mathcal{G}$ and $\mathcal{H}$ are *logically equivalent* if and only if all models of $\mathcal{H}$ are also models for $\mathcal{H}$ and vice-versa
  - in other words, if and only if they have the same models
    - notice that, if two formulas $\mathcal{G}$ and $\mathcal{H}$ have the same models, they have *the same truth table*
  - in yet other words, if and only if $\mathcal{G}$ is a logical consequence of $\mathcal{H}$ and $\mathcal{H}$ is a logical consequence of $\mathcal{G}$
    - that is, if and only if both $\{\mathcal{H}\} \models \mathcal{G}$ and $\{\mathcal{G}\} \models \mathcal{H}$
      - notice that this corresponds to saying that $\models \mathcal{G} \leftrightarrow \mathcal{H}$
  - to indicate that $\mathcal{G}$ and $\mathcal{H}$ are logically equivalent we write $\mathcal{G} \equiv \mathcal{H}$
- Examples of logical equivalences:

$\sim(\sim A)$	$\equiv$	A			
$A \land A$	$\equiv$	A	$A \lor A$	$\equiv$	A
$A \land B$	$\equiv$	$B \land A$	$A \lor B$	$\equiv$	$B \lor A$
$(A \land B) \land C$	$\equiv$	$A \land (B \land C)$	$(A \lor B) \lor C$	$\equiv$	$A \lor (B \lor C)$
$A \land (A \lor B)$	$\equiv$	A	$A \lor (A \land B)$	$\equiv$	A
$A \land (B \lor C)$	$\equiv$	$(A \land B) \lor (A \land C)$	$A \lor (B \land C)$	$\equiv$	$(A \lor B) \land (A \lor C)$
$\sim(A \land B)$	$\equiv$	$\sim A \lor \sim B$	$\sim(A \lor B)$	$\equiv$	$\sim A \land \sim B$
$A \rightarrow B$	$\equiv$	$\sim A \lor B$	$A \rightarrow B$	$\equiv$	$\sim(A \land \sim B)$
$(\sim A \land A) \lor B$	$\equiv$	B	$(\sim A \lor A) \land B$	$\equiv$	B

- Notice that, from the equivalences above, we obtain also
  $A \rightarrow B \equiv \sim B \rightarrow \sim A$
- Another interesting equivalence is the following:
  $A \rightarrow (B \rightarrow C) \equiv A \land B \rightarrow C$

Usefulness of logical equivalences

- Logical equivalences are very useful, since they allow us to "transform" wffs without changing their meaning (i.e. without affecting the truth table)
- First, let us introduce the notion of **subformula**
- Intuitively, a subformula of a formula $\mathcal{G}$ is the set of wffs that appear in $\mathcal{G}$
- More precisely, the set SubF($\mathcal{G}$) of subformulas of wff $\mathcal{G}$ is defined (recursively) as follows:
  - if $\mathcal{G}$ is the propositional letter A, then SubF($\mathcal{G}$) = {A}
  - if $\mathcal{G}$ has the form ~$\mathcal{H}$, then SubF($\mathcal{G}$) = {$\mathcal{G}$} $\cup$ SubF{$\mathcal{H}$}
  - if $\mathcal{G}$ has the form $\mathcal{H} \wedge \mathcal{K}$, $\mathcal{H} \vee \mathcal{K}$, $\mathcal{H} \rightarrow \mathcal{K}$, $\mathcal{H} \leftrightarrow \mathcal{K}$, then SubF($\mathcal{G}$) = {$\mathcal{G}$} $\cup$ SubF{$\mathcal{H}$} $\cup$ SubF{$\mathcal{K}$}
- For example, SubF(A $\rightarrow$ ~B $\leftrightarrow$ C $\wedge$ A) = {A $\rightarrow$ ~B $\leftrightarrow$ C $\wedge$ A, A $\rightarrow$ ~B, C $\wedge$ A, A, ~B, B, C, A}
- If $\mathcal{H} \equiv \mathcal{K}$, $\mathcal{H} \in$ SubF($\mathcal{G}$), and $\mathcal{G}'$ is obtained by replacing subformula $\mathcal{H}$ with $\mathcal{K}$, then $\mathcal{G} \equiv \mathcal{G}'$
  - this implies that $\mathcal{K} \in$ SubF($\mathcal{G}'$)
  - for example, since we have
    door_open $\rightarrow$ light_on $\equiv$ ~door_open $\vee$ light_on,
    then also
    power_on $\rightarrow$ (door_open $\rightarrow$ light_on) $\equiv$
    power_on $\rightarrow$ (~door_open $\vee$ light_on)
  - equivalences are useful to rewrite complex formulas in equivalent ones, to better understand their meaning

From propositional to first-order logic

- Propositional logic has limited expressive power
  - as a simple example, let us consider the classic Aristotelian syllogism:
    All humans are mortal
    Socrates is human
    therefore, Socrates is mortal
  - in propositional logic we lack the possibility to state things "for all" elements of a class (e.g. for all elements belonging to the class "humans")
    - we can state things for single elements (e.g. for Socrates), but we cannot generalize (e.g. to "all humans")
  - in other words, in propositional logic we cannot **quantify** over the elements of a certain class/set
- To overcome this limitation, we introduce **first-order logic**
  - the key addition of first-oder logic (FOL) with respect to propositional logic is the possibility to *quantify* over *variables*
    - FOL, then, introduces a few concepts to the core elements of propositional logic
  - for, example, in FOL we can formalize the syllogism above through the following formula:
    $\forall x\ (human(x) \rightarrow mortal(x)) \wedge human(Socrates)$
    $\rightarrow$
    $mortal(Socrates)$

The syntax of FOL

- FOL is based on the following elements
  - a set of individual *constants*
    - often constants are named c_1, c_2, ...
      - sometimes we use K, T, or even '1', '2', etc.
  - a set of individual *variables*
    - often variables are named x_1, x_2, ...
      - however, we will also use y, z, t, etc.
  - a set of *function letters*
    - we will often indicate function letters as f_i^j
      - where i is a natural number (i = 1, 2, ...), and j is also a natural > 0 which indicates the number of arguments of the function
        - » j is called the "arity" of the function letter
  - a set of *predicate letters*
    - we will often call them A_i^j
      - again, i is a natural number (i = 1, 2, ...), and j is the arity of the predicate
  - the quantifiers ∀, ∃
    - ∀ is called the "universal" quantifier, while ∃ is called the "existential" quantifier
  - the same logical connectives of propositinal logic: ~, ∧, ∨, →, ↔
  - the usual auxiliary symbols (,)
    - notice that the set of constants, the set of variables, the set of function letters, and the set of predicate letters are all at most *denumerable*
- For example, in the formula

  ∀x (human(x) → mortal(x)) ∧ human(Socrates)
  →
  mortal(Socrates)

  x is a variable, *Socrates* is a constant, and both *human* and *mortal* are predicate letters that have exactly one argument
  - i.e. in which the arity j is 1

Terms and formulas

- Before being able to define which are the well-formed formulas (wffs) of FOL, we first have to introduce the notion of **term** and of **atomic formula**
  - in a nutshell, terms are built from function letters, while atomic formulas are built y applying predicate letters to terms
- As usual, the notion of term is defined in an *inductive* (i.e. recursive) manner:
  - individual constants and variables are terms
  - if f_i^j is a function letter (with arity j) and $t_1, \dots t_j$ are terms, $f_i^j(t_1, \dots t_j)$ is a term
  - nothing else is a term
    - examples of terms: c_3, $f_2^3(c_1, x_5, x_2)$, $f_5^2(x_2, f_1^2(x_3, x_4))$
    - if *sum* and *div* are function letters with arity 2, and *x*, *y*, *z* are variables, the following is also a term: sum(x, div(z, sum(x, y)))
      - we might as well write it as x + z/(x + y) !
- The notion of atomic formula defined from the ones of predicate letter and of term:
  - if A_i^j is a predicate letter (with arity j) and $t_1, \dots t_j$ are terms, then $A_i^j(t_1, \dots t_j)$ is an atomic formula
  - nothing else is an atomic formula
    - form example, *x* is a term, and so is *Socrates*, while *human* and *mortal* are both predicate letters, so human(x) and mortal(Socrates) are atomic formulas
    - if we introduce predicate letters *gt* and *eq*, both with arity 2 (and *x*, *y*, *z* are variables), *eq(y, sum(x, z))* and *gt (div(z, y), sum(x, x))* are both atomic formulas
      - we will often write, more succinctly, $y = x + z$ and $z/y > x+x$

Well-formed formulas

- Wffs are, as usual, defined inductively:
  - atomic formulas are wffs
  - if $\mathcal{F}$ is a wff, $\sim\mathcal{F}$ is a wff
  - if $\mathcal{F}_1$, $\mathcal{F}_2$ are wffs, the following are also wffs:
    $(\mathcal{F}_1 \wedge \mathcal{F}_2)$, $(\mathcal{F}_1 \vee \mathcal{F}_2)$, $(\mathcal{F}_1 \rightarrow \mathcal{F}_2)$, $(\mathcal{F}_1 \leftrightarrow \mathcal{F}_2)$
  - if $\mathcal{F}$ is a wff and x is a variable, the following are also wffs:
    $(\forall x\ \mathcal{F})$,
    $(\exists x\ \mathcal{F})$
  - nothing else is a wff
- Examples of wffs:
  $(\forall x\ A_1^1(x)) \rightarrow (\exists y\ (A_2^2(f_2^1(y), z) \wedge A_1^1(x)))$
  mortal(Socrates)
  $\forall h\ (\text{human}(h) \rightarrow \text{man}(h) \vee \text{woman}(h))$
  $\forall n_1, n_2\ (\text{natural}(n_1) \wedge \text{natural}(n_2)$
  $\rightarrow \exists n_3(\text{natural}(n_3) \wedge \text{gt}(n_3, n_1) \wedge \text{lt}(n_3, n_2)))$

- Just as with propositional logic, an order of precedence for the connectives and quantifiers of FOL is defined, to avoid writing all the parentheses:
  - $\sim$, $\forall$, $\exists$ have the same precedence
  - they precede $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
    - in this order (i.e. $\wedge$ precedes $\vee$, which precedes $\rightarrow$, etc.)
    - the operators are associative to the left, just as in propositional logic

Some definitions

- The notion of subformula is defined also for FOL; more precisely, the set SubF($\mathcal{G}$) of subformulas of wff $\mathcal{G}$ of FOL is (inductively) defined as follows:
  - if $\mathcal{G}$ is an atomic formula then SubF($\mathcal{G}$) = {$\mathcal{G}$}
  - if $\mathcal{G}$ is either ~$\mathcal{H}$, or $\forall$x $\mathcal{F}$, or $\exists$x $\mathcal{F}$ then SubF($\mathcal{G}$) = {$\mathcal{G}$} $\cup$ SubF{$\mathcal{H}$}
  - if $\mathcal{G}$ has the form $\mathcal{H} \wedge \mathcal{K}$, $\mathcal{H} \vee \mathcal{K}$, $\mathcal{H} \rightarrow \mathcal{K}$, $\mathcal{H} \leftrightarrow \mathcal{K}$, then SubF($\mathcal{G}$) = {$\mathcal{G}$} $\cup$ SubF{$\mathcal{H}$} $\cup$ SubF{$\mathcal{K}$}
- Let us introduce some definitions and terminology, which will allow us to better reason about FOL formulas:
  - in a formula of the kind $\forall$x $\mathcal{G}$, subformla $\mathcal{G}$ is called the **scope** of variable x
    - for example, in formula $\forall x\ A_1^1(x) \rightarrow \exists y\ (A_2^2(f_2^1(y), z) \wedge A_1^1(x))$ the scope of $\forall$x is $A_1^1(x)$, while the scope of $\exists$y is $A_2^2(f_2^1(y), z) \wedge A_1^1(x)$
  - we say that an **occurrence** of a variable x is **bound** in a wff $\mathcal{G}$ if and only if it lies in the scope of a quantifier
    - i.e., if it lies in the scope of either a $\forall$x, or of a $\exists$x
    - an occurrence that is not bound is said to be **free**
    - for example, in formula $\forall x\ A_1^1(x) \rightarrow \exists y\ (A_2^2(f_2^1(y), z) \wedge A_1^1(x))$ the first occurrence of x is bound, the last is free
  - we say that a **variable** x is **bound** (free) in a wff $\mathcal{G}$ if and only if it has *at least* one bound (free) occurrence in $\mathcal{G}$
    - notice that we are now talking about the variable in general, not one of its occurrences
  - we say that a wff is **closed** if and only if it has no free variables
    - e.g, formula $\forall$h (human(h) $\rightarrow$ man(h) $\vee$ woman(h)) is closed
  - Given a wff $\mathcal{G}$ with free variables x_1, ... x_n, we say that $\forall x_1, \ldots x_n\ \mathcal{G}$ is its **universal closure**, while $\exists x_1, \ldots x_n\ \mathcal{G}$ is its **existential closure**

FOL - semantics

- Semantics are a means to defining the truth/falsity of wffs
  - we will define when $\mathcal{V}(\mathcal{G})$ is true (T), or false (F), with $\mathcal{G}$ a wff of FOL
  - we will again introduce a notion of assignment (to *variables*, in this case)
  - however, we first need to introduce some concept that define how function and predicate letters (and individual constants, too) must be "interpreted"
    - that is, we first need to give a meaning to function and predicate letters
  - in addition, before we introduce the notion of assignment for variables, we need to be able to say *what* values can be assigned to variables
    - in propositional logic propositional letters could only be assigned either value T, or value F; however, variables in FOL can be assigned also different values, but which ones?
- As the core of the semantics of FOL is the notion of **interpretation**
- An **interpretation** $\mathcal{I}$ is a pair $<\mathcal{D}, \mathcal{J}>$, where $\mathcal{D}$ is a (nonempty) *domain* (i.e. a set), and $\mathcal{J}$ a *mapping*, which must be as follows:
  - $\mathcal{J}$ maps every constant c_i onto a value d of $\mathcal{D}$ (i.e. $d \in \mathcal{D}$)
  - $\mathcal{J}$ maps every function letter f_i^j onto a function $\mathcal{D}^j \rightarrow \mathcal{D}$
    - $\mathcal{D}^j$ is a shortcut for $\mathcal{D} \times \mathcal{D} \ldots \times \mathcal{D}$, j times
    - that is, $\mathcal{J}$ maps every function letter to a function that takes j arguments
  - $\mathcal{J}$ maps every predicate letter A_i^j onto a *relation* on $\mathcal{D}$ with arity j
    - that is, it maps A_i^j onto a relation *r* such that $r \subseteq \mathcal{D}^j$
- Notice that, when we write a formula, we always intend it to be interpreted in a certain way
  - when we write human(h), or man(h) we do have a certain interpretation in mind

Examples of interpretations

- Let us consider (atomic) formula $A_1^2(f_1^2(x, y), f_2^2(a, x))$, where *a* is a constant, and both *x* and *y* are variables
  - if we interpret A_1^2 as equality, f_1^2 as multiplication, f_2^2 as sum, and *a* as the number 1, the formula means "x times y is equal to 1 plus x", or, in other words, x*y = 1 + x
- Notice that we often give function and predicate letters names that indicate the intended interpretation
  - e.g. div(x, y), to be interpreted as *x / y*, or gt(z, t), to be interpreted as *z > t*
    - notice that, in principle, we could interpret div(x, y) as *x – y* !
  - for practical uses, we will use for function and predicate letters the same symbols as their intended interpretation
    - i.e. we will more succinctly write $x + 2 > y \rightarrow z < 3$ rather than something like $A_2^2(x, b) \rightarrow A_3^2(z, c)$ to be interpreted accordingly
- Notice that, given (atomic) formula x*y = 1 + x, its truth/falsity depends on the values associated with x and y
  - for example, if x is 1 and y is 2, then the formula is true, as 2*1 = 1 + 1
  - however, if x is 3 and y is 2, then the formula is false, as 3*2 is different from 1 + 3
- Then, we need to define how the truth/falsity of a formula changes if the values associated with variables
  - to this end, we introduce the notion of *assignment* of values to variables

Assignment

- An assignment *s* into an interpretation $\mathcal{I}$ is a mapping that associates every variable x_i with a value in $\mathcal{D}$
  - for example, if x, y, z are variables, and the domain $\mathcal{D}$ is $\mathbb{R}$, a possible assignment might be
    s: $\{x \mapsto 3.67, B \mapsto \pi, C \mapsto 1/3\}$
- Then, we introduce the mapping s*, which associates every term with a value in $\mathcal{D}$
  - if the term is made of just an individual constant (e.g. c), then $s^*(c) = \mathcal{J}(c)$
    - i.e. the value of the term corresponds to the interpretation if c
  - if the term is made of just an individual variable (e.g. x), then $s^*(x) = s(x)$
  - finally, for a generic term $f_i^j(t_1, \ldots t_j)$, where $t_1, \ldots t_j$ are in turn terms, we have
    $s^*(f_i^j(t_1, \ldots t_j)) = \mathcal{J}(f_i^j)(s^*(t_1), s^*(t_2), \ldots s^*(t_j))$
    - s* is defined inductively...

Value of a formula

- We can now define the evaluation function $\mathcal{V}$, which determines whether a wff is true (T) or false (F)
  - in fact, since to evaluate a wff $\mathcal{F}$ we need an interpretation $\mathcal{I}$ and an assignment *s* to variables, we will actually write $\mathcal{V}^{(\mathcal{I},s)}$ to highlight this dependency
- Given an interpretation $\mathcal{I}$, and an assignment *s*, the value $\mathcal{V}^{(\mathcal{I},s)}$ (which can be either true, or false) of a wff formula $\mathcal{F}$ in assignment *s* is defined as follows
  - $\mathcal{V}^{(\mathcal{I},s)}(A_i^j(t_1, \ldots t_j))$=T if and only if $(s^*(t_1), s^*(t_2), \ldots s^*(t_j)) \in \mathcal{J}(A_i^j)$
    - where $A_i^j(t_1, \ldots t_j)$ is, of course, an atomic formula
  - $\mathcal{V}^{(\mathcal{I},s)}(\sim\mathcal{G})$=T if and only if $\mathcal{V}^{(\mathcal{I},s)}(\mathcal{G})$=F
  - $\mathcal{V}^{(\mathcal{I},s)}(\mathcal{G} \wedge \mathcal{H})$=T if and only if $\mathcal{V}^{(\mathcal{I},s)}(\mathcal{G})$=T and $\mathcal{V}^{(\mathcal{I},s)}(\mathcal{H})$=T
  - $\mathcal{V}^{(\mathcal{I},s)}(\mathcal{G} \vee \mathcal{H}) = \mathcal{V}^{(\mathcal{I},s)}(\sim(\sim\mathcal{G} \wedge \sim\mathcal{H}))$
  - $\mathcal{V}^{(\mathcal{I},s)}(\mathcal{G} \rightarrow \mathcal{H}) = \mathcal{V}^{(\mathcal{I},s)}(\sim\mathcal{G} \vee \mathcal{H})$
  - $\mathcal{V}^{(\mathcal{I},s)}(\mathcal{G} \leftrightarrow \mathcal{H}) = \mathcal{V}^{(\mathcal{I},s)}(\mathcal{G} \rightarrow \mathcal{H}) \wedge \mathcal{V}^{(\mathcal{I},s)}(\mathcal{H} \rightarrow \mathcal{G})$
  - $\mathcal{V}^{(\mathcal{I},s)}(\forall x\ \mathcal{G})$=T if and only if, **for every assignment** *s*' that differs from *s* **at most** for the value assigned to variable *x*, $\mathcal{V}^{(\mathcal{I},s')}(\mathcal{G})$=T
    - basically, this says that if we keep the values assigned to variables different from x as in *s*, but "try" all possible values for variable x, formula $\mathcal{G}$ must remain true
  - $\mathcal{V}^{(\mathcal{I},s)}(\exists x\ \mathcal{G})$=T if and only if there is **at least one assignment** *s*', which differs from *s* **at most** for the value assigned to variable x, such that $\mathcal{V}^{(\mathcal{I},s')}(\mathcal{G})$=T
    - in other words, this says that we can find a suitable assignment for variable x such that $\mathcal{G}$ is true (provided all other variables keep the same values that was assigned to them in *s*)

Satisfaction and satisfiability

- We say that assignment *s* satisfies wff $\mathcal{G}$ if and only if $\mathcal{V}^{(\mathcal{I},\, s)}(\mathcal{G})$=T
  - to represent that *s* satisfies $\mathcal{G}$ we will write $(\mathcal{I}, s) \vDash \mathcal{G}$
- Given an interpretation $\mathcal{I}$, we say that:
  - a wff $\mathcal{G}$ is **satisfiable** in $\mathcal{I}$ if and only if there is an assignment *s* that satisfies $\mathcal{G}$ (i.e. such that $(\mathcal{I}, s) \vDash \mathcal{G}$)
  - a wff $\mathcal{G}$ is **true** in $\mathcal{I}$ if and only if *every assignment s* satisfies $\mathcal{G}$
    - in this case we will write $\mathcal{I} \vDash \mathcal{G}$
  - a wff $\mathcal{G}$ is **false** in $\mathcal{I}$ if and only if *no assignment s* satisfies $\mathcal{G}$
    - that is, $\mathcal{G}$ is *contradictory* in $\mathcal{I}$
- If a wff $\mathcal{G}$ is true in interpretation $\mathcal{I}$, then $\mathcal{I}$ is a **model** for $\mathcal{G}$
- We say that a wff $\mathcal{G}$ is **satisfiable** if and only if there is (at least) an interpretation in which $\mathcal{G}$ is satisfiable
  - notice that in this case we say that $\mathcal{G}$ is satisfiable *tout-court*, without mentioning the interpretation
- We say that a wff $\mathcal{G}$ is **logically valid** if and only if it is true *in every interpretation*
  - in this case we will write $\vDash \mathcal{G}$
- Conversely, we say that a wff $\mathcal{G}$ is **contradictory** if and only if it is *false in every interpretation*
  - i.e. if and only if $\vDash \sim\mathcal{G}$
- In a given interpretation $\mathcal{I}$, a wff $\mathcal{G}$ that is closed is either true or false (that is, it cannot be satisfiable but not true) (in other words, its truth does not depend on the value assigned to the -quantified- variables)

Logical consequence and equivalence in FOL

- Some concepts and definitions that remain the same in FOL as in propositional logic:
  - if Γ is a set of wff, an interpretation $\mathcal{I}$ that is a model for **all** wffs in Γ is a **model** for Γ
  - a formula $\mathcal{F}$ is a **logical consequence** of Γ if and only if in every interpretation $\mathcal{I}$, an assignment that satisfies every wff in Γ also satisfies $\mathcal{F}$
    - as usual, we will write $\Gamma \models \mathcal{F}$ to indicate that $\mathcal{F}$ is a logical consequence of Γ
  - **Theorem**: given a set Γ of wffs such that $\Gamma = \Delta \cup \{\mathcal{H}\}$ (i.e., such that Γ contains a wff $\mathcal{H}$, and Γ - $\{\mathcal{H}\} = \Delta$), then
    $\Gamma \models \mathcal{G}$ if and only if $\Delta \models \mathcal{H} \rightarrow \mathcal{G}$
    - it is exactly the same as for propositional logic!
  - a wff $\mathcal{F}$ is **logically equivalent** to another wff $\mathcal{G}$ if and only if both $\{\mathcal{F}\} \models \mathcal{G}$ and $\{\mathcal{G}\} \models \mathcal{F}$
    - that is, if and only if $\models \mathcal{F} \leftrightarrow \mathcal{G}$
    - if $\mathcal{F}$ and $\mathcal{G}$ are logically equivalent, we will write $\mathcal{F} \equiv \mathcal{G}$
    - again, this is the same as for propositional logic

A set-theoretic interpretation of boolean connectives

- An intuition for the semantics of logic assertions
- Logical formulas may be considered as ***intensive*** representations of ***sets of models***
  - Denote as
    - $\mathcal{S}(\mathcal{G})$ the set of models for formula $\mathcal{G}$
    - **Compl,** $\cup$, $\cap$, the set theoretic operations of complement, union, and intersection
- Then boolean connectives have a set-theoretical interpretation
  - Conjunction stands for intersection $\mathcal{S}(\mathcal{G} \wedge \mathcal{H}) = \mathcal{S}(\mathcal{G}) \cap \mathcal{S}(\mathcal{H})$
  - Disjunction stands for union $\mathcal{S}(\mathcal{G} \vee \mathcal{H}) = \mathcal{S}(\mathcal{G}) \cup \mathcal{S}(\mathcal{H})$
  - Negation stands for complement $\mathcal{S}(\sim\mathcal{G})$ = **compl**($\mathcal{S}(\mathcal{G})$)
  - Implication: union of complement of the premise and consequence $\mathcal{S}(\mathcal{G} \rightarrow \mathcal{H})$ = **compl**($\mathcal{S}(\mathcal{G})$) $\cup\ \mathcal{S}(\mathcal{H})$
- It is easily seen on propositional logic:
  - Example: C means "it is cloudy", R "it rains", W "ground is wet"
  - The "universe set" is the set of all assignments to propositions

C	R	W	~R	C ∧ R	C ∨ R	R → W
F	F	F	T	F	F	T
F	F	T	T	F	F	T
F	T	F	F	F	T	F
F	T	T	F	F	T	T
T	F	F	T	F	T	T
T	F	T	T	F	T	T
T	T	F	F	T	T	F
T	T	T	F	T	T	T

On the strength or weakness of statements

- Again, an intuition for the semantics of logics
  - **STRONG**: more restrictive, less easy to satisfy, satisfied "in fewer cases"(hence having a smaller set of models)
  - **WEAK**: less restrictive, easier to satisfy, satisfied "in more cases"(hence having a larger set of models)
- How is the strength of a statement formalized in logic?
  - Logical statements are ordered (though not totally) from the stronger ones (having fewer models) to the weakest ones (having more models)
    - At one extreme **FALSE** is the strongest formula of all
    - At the other extreme **TRUE** is the weakest formula of all
  - The implication statement $\mathcal{G} \rightarrow \mathcal{H}$ asserts that $\mathcal{G}$ is stronger than $\mathcal{H}$
  - The implication operator acts as a sort of relational operator "<=" that compares the "degree of truth" of statements
  - It corresponds to the set-theoretic "inclusion"operator for the sets of models of the related formulas
    - $\mathcal{G} \rightarrow \mathcal{H}$ if and only if $\mathcal{S}(\mathcal{G}) \subseteq \mathcal{S}(\mathcal{H})$
    - Ex.: if $\mathcal{G}$ is x>15 and $\mathcal{H}$ is x>7 then $\mathcal{G} \rightarrow \mathcal{H}$ and the following set inclusion holds $\{x \mid x>15\} \subseteq \{x \mid x>7\}$
- Effect of boolean connectives on the strength of statements
  - Disjunction ($\vee$) weakens
    - Formula $\mathcal{G} \vee \mathcal{H}$ is weaker than formula $\mathcal{H}$ because it may be true in some more cases than $\mathcal{H}$: those where formula $\mathcal{G}$ is true and $\mathcal{H}$ is not
    - Ex: $\{1<x<10\} \vee \{20<x<30\}$ is weaker than $\{20<x<30\}$
  - Conjunction ($\wedge$) strengthens
    - Formula $\mathcal{G} \wedge \mathcal{H}$ is stronger than formula $\mathcal{H}$ because it may be true in some less cases than $\mathcal{H}$: those where formula $\mathcal{G}$ is false and $\mathcal{H}$ is true
    - Ex: odd(x) $\wedge$ $\{1<x<10\}$ is stronger than $\{1<x<10\}$
  - Implication (adding a premise to a given statement) weakens
    - $\mathcal{G} \rightarrow \mathcal{H}$ is weaker than $\mathcal{H}$ because it may be true in some more cases than $\mathcal{H}$: those where formula $\mathcal{G}$ is false (in these cases $\mathcal{H}$ needs not be true)
    - Ex: (you give me your car) $\rightarrow$ {I give you 100 EURO} is weaker than {I give you 100 EURO}
    - Also notice that an implication in strengthened if the premise is weakened, it is strengthened if so is the consequence (Ex.:...).

Quantifications and {→, ∧} connectives

- Universal quantification ∀ goes hand in hand with implication →
- Existential quantification ∃ goes hand in hand with conjunction ∧
- Ex.: G(x)="x is good", O(x)="x is an orange"
  - "Sensible" statements
    - $\forall x\ (O(x) \rightarrow G(x))$: "all oranges are good"
    - $\exists x\ (O(x) \wedge G(x))$: "some oranges are good"
  - "Wild" statements
    - $\forall x\ (O(x) \wedge G(x))$: "everything is an orange and is good"
      - A very strong statement
      - "impossible to satisfy" if there is something that is not an orange or is not good
    - $\exists x\ (O(x) \rightarrow G(x))$: "there is a thing which, if it is an orange, is good"
      - A very weak statement
      - "trivial" to satisfy: it suffices to find something (good or not) that is not an orange or anything (be it an orange or not) that is good