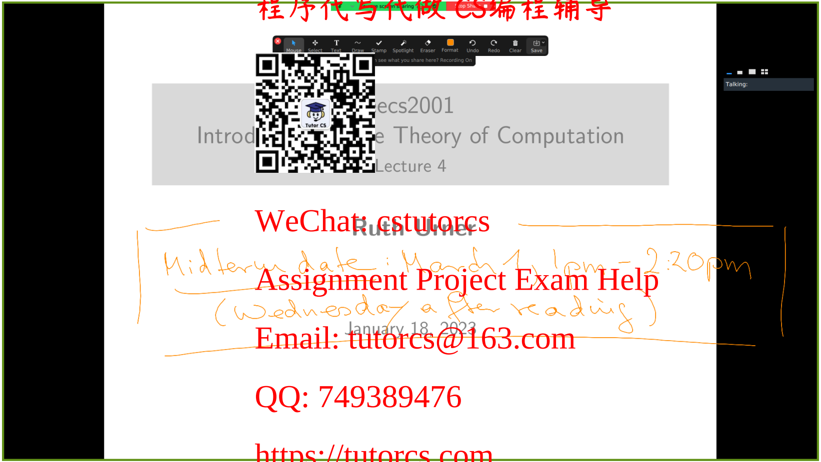
cs2001
Introduction … Theory of Computation

ecture 4

Ruth Urner

January 18, 2023

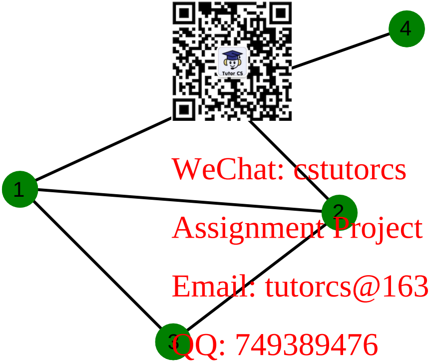

ecs2001
Introd e Theory of Computation
Lecture 4
Ruth Urner
Midterm date: March 1, 1pm - 2:20pm
(Wednesday after reading)
January 18, 2023
Talking:

Graphs

Reading: ITC Section 0.2

Graphs

We illustrate graphs as points and connecting lines:



An undirected graph (or simply a graph) $G = (V, E)$ is defined as a pair of

- a set V of vertices (also called nodes)
- and a set E of edges,

where each edge in E is a (multi-set) subset of V of size 2.

Graphs

We illustrate gr… connecting lines:

Formaly:

$G = (\{1,2,3,4,5\}, \{\{1,5\}, \{5,4\}, \{5,2\}, \{2,3\}, \{1,2\}, \{1,3\}\})$

An undirected graph (or simply a graph) $G = (V, E)$ is defined as a pair of

- a set V of vertices (also called nodes)
- and a set E of edges,

where each edge in E is a (multi-set) subset of V of size 2.

Graphs

Talking: Ruth Urner

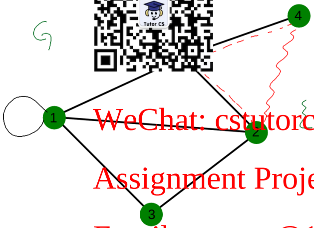
Is G (or E)

• reflexive? no

• transitive? no

• symmetric? yes

We illustrate gr connecting lines:



Formally:

$G = (\{1,2,3,4,5\}, \{\{1,5\}, \{5,4\}, \{5,2\}, \{1,1\}, \{2,3\}, \{1,2\}, \{1,3\}\})$

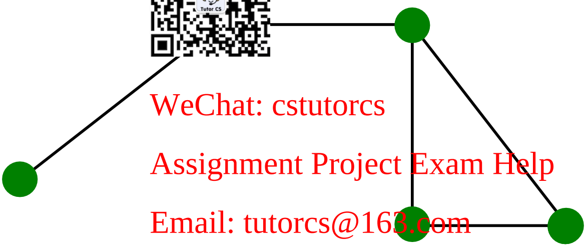
transitive directed graph

An undirected graph (or simply a graph) $G = (V, E)$ is defined as a pair of

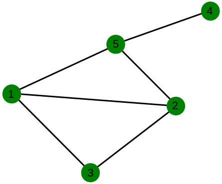
- a set V of vertices (also called nodes)
- and a set E of edges,

where each edge in E is a (multi-set) subset of V of size 2.

Graphs can be used to il[...]tions between objects.



Some definitions and terminology for graphs



Let $G = (V, E)$ be a gr… define:

- a vertex $v \in V$ is … o edge $e \in E$ if $v \in e$

- the **degree** $d(v)$ of a vertex $v \in V$ is the number of edges v is adjacent to

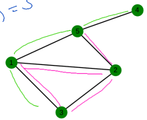
- a **path** in a graph is a sequence $p = (v_1, v_2, \ldots, v_n)$ of vertices such that $\{v_i, v_{i+1}\} \in E$ for all $i \in \{1, 2, \ldots, n\}$

- a **simple path** is a path in which no vertex occurs more than once

Some definitions and terminology for graphs

$d(4) = 1$

$d(1) = 3$



Let $G = (V, E)$ … e define:

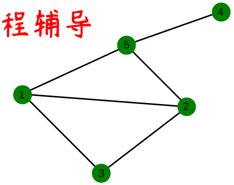
- a vertex v … to edge $e \in E$ if $v \in e$
- the degree $d(v)$ of a vertex $v \in V$ is the number of edges v is adjacent to
- a path in a graph is a sequence $p = (v_1, v_2, \dots, v_n)$ of vertices such that $\{v_i, v_{i+1}\} \in E$ for all $i \in \{1, 2, \dots, n\}$
- a simple path is a path in which no vertex occurs more than once

example of a path $(5, 2, 3, 1, 2)$

not a simple path!

example of a simple path $(4, 5, 1, 3)$

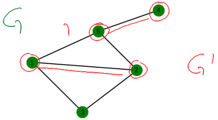
Let $G = (V, E)$ be a graph. We define:



- a subgraph G' of graph G is a a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$
- graph G is connected, if for every two distinct vertices $v, v' \in V$, there is a path from v to v' (that is, a path which has v as it's first and v' as its last vertex)
- we call an edge $e \in E$ a self-loop if $e = \{v, v\}$ for some $v \in V$ (that is, the edge connects some vertex v to itself)

Some definiti… an… for graphs

Let $G = (V,$ … We define:

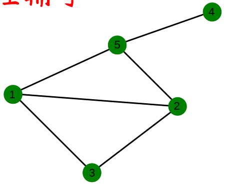


- a subgraph G' of graph G is a a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$
- graph G is connected, if for every two distinct vertices $v, v' \in V$ there is a path from v to v' (that is, a path which has v as it's first and v' as its last vertex)
- we call an edge $e \in E$ a self-loop if $e = \{v, v\}$ for some $v \in V$ (that is, the edge connects some vertex v to itself)

Subgraph $G' = (\{1,5,4,2\}, \{\{4,5\},\{1,2\}\})$

G is a connected graph, but G' is not connected

Some definitions and terminology for graphs

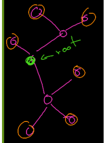


Let $G = (V, E)$ be a graph. We define:

- a cycle in a graph is a path for which the first and last vertex are identical
- a simple cycle in a graph is a cycle that contains at least three different vertices and doesn't contain any vertex more than once (except for the first and last)
- a tree is a graph that is connected and doesn't contain any simple cycles; vertices of degree one in a tree are called leaves; sometimes we declare one vertex in a tree to be special and call it a root

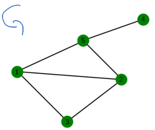
Some definitions and terminology for graphs

tree



leaves of the tree

Let $G = (V,$ … We define:



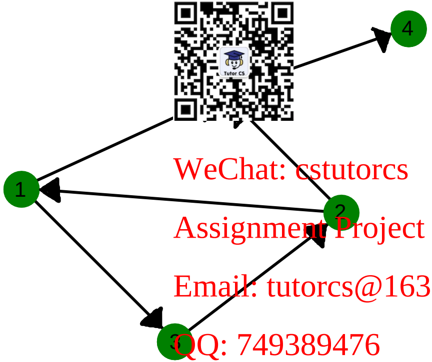
- a cycle in a graph is a path for which the the first and last vertex are identical
- a simple cycle in a graph is a cycle that contains at least three different vertices and doesn't contain any vertex more than once (except for the first and last)
- a tree is a graph that is connected and doesn't contain any simple cycles; vertices of degree one in a tree are called leaves; sometimes we declare one vertex in a tree to be special and call it a root

Cycle in G:
(5,2,3,1,5)
(also a simple cycle)

Cycle that is not simple:
(5,2,3,1,2,5)

Directed graphs

We illustrate directed graphs as points and connecting arrows:



A directed graph $G = (V, E)$ is formally defined as a pair of

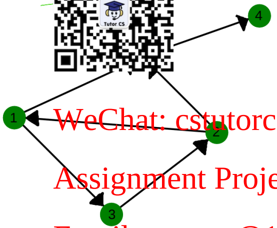
- a set V of vertices
- and a set E of edges,

where each edge in E is a 2-tuple (or pair) of members of V.

Directed graphs

For directed graphs the edge set E is a relation over the vertex set V. In the example graph G, this relation is
- not reflexive
- not transitive
- not symmetric

We illustrate directed graphs using points and connecting arrows:



$G = (\{1,2,3,4,5\}, \{(1,5),(5,4),(2,5),(3,2),(2,1),(1,3)\})$

A directed graph $G = (V, E)$ is formally defined as a pair of
- a set V of vertices
- and a set E of edges,

where each edge in E is a 2-tuple (or pair) of members of V.

Some definitions and terminology for directed graphs

- For an edge $e = (v_1, v_2)$ in directed graph, we call e an incoming edge to v_2 and an outgoing edge for v_1.

- The in-degree $d_{in}(v)$ of a vertex v in a directed graph is the number of its incoming edges. The out-degree $d_{out}(v)$ is the number of outgoing edges from v.

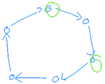
- A directed path in a directed graph is a sequence $p = (v_1, v_2, \ldots, v_n)$ of vertices such that $(v_i, v_{i+1}) \in E$ for all $i \in \{1, 2, \ldots, n\}$.

- A directed graph is strongly connected if every two distinct vertices v and v' in V are connected by a directed path.

- A directed cycle in a directed graph is a directed path for which the first and last vertex coincide.

Some definitions and terminology for directed graphs

- For an e… in directed graph, we call e an incoming… an outgoing edge for v_1.
- The in-d… a vertex v in a directed graph is the number of its incoming edges. The out-degree $d_{out}(v)$ is the number of outgoing edges from v.
- A directed path in a directed graph is a sequence $p = (v_1, v_2, \ldots, v_n)$ of vertices such that $(v_i, v_{i+1}) \in E$ for all $i \in \{1, 2, \ldots, n\}$.
- A directed graph is strongly connected if every two distinct vertices v and v' in V are connected by a directed path.



a strongly connected graph

Strings and Languages

Reading: ITC Section 0.2

An alphabet is a non-empty, finite set of symbols.

Examples:

- $\Sigma = \{0, 1\}$.
- $\Gamma = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$

A string or word over some alphabet Σ is a finite sequence of symbols from Σ.

Examples:

- ruth
- 00101

A language is a set of words (set of strings).

Alphabet, words, languages

An alphabet is a finite set of symbols.

Examples:

- $\Sigma = \{0, 1\}$
- $\Gamma = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$

A string or word over some alphabet Σ is a finite sequence of symbols from Σ.

Examples:

- ruth   word over Γ
- 00101   word over Σ

abc45 is not a word over Σ or Γ (but a word over $\Sigma \cup \Gamma$)

A language is a set of words (set of strings).

{ruth}   {ruth, abc, chair, dog}

Alphabet, words, languages

$\Omega = \{\ ,\ ,\ \}$

An alphabet i[illegible] finite set of symbols.

Examples:

- $\Sigma = \{0, 1\}$
- $\Gamma = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$

A string or word over some alphabet Σ is a finite sequence of symbols from Σ.

Examples:

- ruth
- 00101

A language is a set of words (set of strings).

Talking: Ruth Urner

Some definitions and terminology for strings and languages

- the length of a string **w** is the number of symbols in **w**
- the empty string ϵ is a string of length 0 (that is, without any symbols)

- the reverse $\mathbf{w}^{\mathcal{R}}$ of a string **w** is the number of symbols in **w**

- the concatenation of strings $\mathbf{w} = s_1 s_2 \ldots s_n$ and $\mathbf{z} = t_1 t_2 \ldots t_m$ is the string $\mathbf{wz} = s_1 s_2 \ldots s_n t_1 t_2 \ldots t_m$

Some definitions and terminology for strings and languages

- the lengt [illegible] is the number of symbols in **w**

- the empty string ϵ is a string of length 0 (that is, without any symbols)

- the reverse $\mathbf{w}^{\mathcal{R}}$ of a string **w** is the number of symbols in **w**

  $\text{ruth}^R = \text{htur}$, $\text{ocean}^R = \text{naeco}$

- the concatenation of strings $\mathbf{w} = s_1 s_2 \ldots s_n$ and $\mathbf{z} = t_1 t_2 \ldots t_m$ is the string $\mathbf{wz} = s_1 s_2 \ldots s_n t_1 t_2 \ldots t_m$

  $w = 01011$, $z = 1101$, $wz = 010111101$

Some definitions and terminology for strings and languages

- a substring $\mathbf{z}$ of string $\mathbf{w}$ is a consecutive subsequence of symbols from $\mathbf{w}$

- a prefix $\mathbf{x}$ of string $\mathbf{w}$ is a substring that starts at the first position in $\mathbf{w}$; that is, $\mathbf{x}$ is a prefix of $\mathbf{w}$ if there exists a string $\mathbf{z}$ such that $\mathbf{w} = \mathbf{xz}$

- $\mathbf{x}$ is a proper prefix of $\mathbf{w}$ if $\mathbf{x}$ is a prefix of $\mathbf{w}$ and $\mathbf{w} \neq \mathbf{x}$

- a prefix-free language over some alphabet Σ is a language where no word in the language is a proper prefix of some other word


Some de
languages
a substri
is a consecutive subsequence of
symbols
= can
not a substring of w
= cea
is a substring of w

Some definitions and terminology for strings and languages

- a substri[illegible] is a consecutive subsequence of symbols
- a prefix **x** of string **w** is a substring that starts at the first position in **w**; that is, **x** is a prefix of **w** if there exists a string **z** such that **w** = **xz**

a proper prefix of w

w = ocean , x = oc is a prefix of w

- **x** is a proper prefix of **w** if **x** is a prefix of **w** and **w** ≠ **x**

w = ocean , x = ocean is a prefix, but not a proper prefix of w

Talking: Ruth Urner


Some definitions and terminology for strings and languages
a substri… is a consecutive subsequence of symbols
a prefix x of string w is a substring that starts at the first position in w; that is, x is a prefix of w if there exists a string z such that w = xz
x is a proper prefix of w if x is a prefix of w and w ≠ x
a prefix-free language over some alphabet Σ is a language where no word in the language is a proper prefix of some other word
Σ = {0,1}
{00, 11, 01, 0011, 0101}
not a prefix-free language since 00 is a prefix of 0011
{00, 11, 0101} is a prefix-free language
Talking:

Given an ordering of symbols in some alphabet Σ, we can define orderings of the words over Σ:

- Lexicographic order

  Definition: We have $\mathbf{w} \leq_{lex} \mathbf{z}$ in lexicographic order for strings $\mathbf{w} = s_1 s_2 \dots s_n$ and $\mathbf{z} = t_1 t_2 \dots t_m$ if $\mathbf{w}$ is a proper prefix of $\mathbf{z}$ or if there exists an index $j < \min\{n, m\}$ such that $s_i = t_i$ for all $i \in \{1, 2, \dots, j\}$ and $s_{j+1} < t_{j+1}$.

- Short-lex order (or string order)

  Definition: We have $\mathbf{w} \leq_{str} \mathbf{z}$ in string order for strings $\mathbf{w} = s_1 s_2 \dots s_n$ and $\mathbf{z} = t_1 t_2 \dots t_m$ if $\mathbf{w}$ is shorter than $\mathbf{z}$ or if $\mathbf{w}$ and $\mathbf{z}$ have the same length and $\mathbf{w} \leq_{lex} \mathbf{z}$ in lexicographic order.

Ordering words

ruth
ocean

apple
apartment

Given an orde… …bols in some alphabet Σ, we can define orderin… …words over Σ:

- Lexicographic order

  Definition: We have $\mathbf{w} \leq_{lex} \mathbf{z}$ in lexicographic order for strings $\mathbf{w} = s_1 s_2 \ldots s_n$ and $\mathbf{z} = t_1 t_2 \ldots t_m$ if $\mathbf{w}$ is a proper prefix of $\mathbf{z}$ or if there exists an index $j < \min\{n, m\}$ such that $s_i = t_i$ for all $i \in \{1, 2, \ldots j\}$ and $s_{j+1} < t_{j+1}$.

- Short-lex order (or string order)

  Definition: We have $\mathbf{w} \leq_{str} \mathbf{z}$ in string order for strings $\mathbf{w} = s_1 s_2 \ldots s_n$ and $\mathbf{z} = t_1 t_2 \ldots t_m$ if $\mathbf{w}$ is shorter than $\mathbf{z}$ or if $\mathbf{w}$ and $\mathbf{z}$ have the same length and $\mathbf{w} \leq_{lex} \mathbf{z}$ in lexicographic order.

cat $\leq_{str}$ dog

ruth $\leq_{str}$ ocean

apple $\leq_{str}$ apartment

Logic Reading: ITC Section 0.2

Propositional logic m[illegible] way we reason about the truth value of statements ([illegible]tions).

A propositional variable is a variable (that stands for a statement) that can take on one of two values:

- true (we also use T or 1 for true)
- false (we also use F or 0 for false)

A truth assignment is a function from a set P of propositional variables to the set of truth values $\{0, 1\}$ (or $\{T, F\}$).

Boolean operations

Given basic propositional statements p and q we can use boolean operations to derive more complex statements (and their truth values):

- $\neg$
- $\wedge$
- $\vee$
- $\rightarrow$
- $\leftrightarrow$
- $\oplus$

We define the truth values of composed propositional statements by the following truth table:

α	β	$(\neg\alpha)$	$(\alpha \vee \beta)$	$(\alpha \wedge \beta)$	$(\alpha \rightarrow \beta)$	$(\alpha \leftrightarrow \beta)$	$(\alpha \oplus \beta)$
T	T	F	T	T	T	T	F
T	F	F	T	F	F	F	T
F	T	T	T	F	T	F	T
F	F	T	F	F	T	T	F

Extending truth assignments to WFF – examples

Say, we have propositional variables p and q, and a formula $\alpha = ((p \to q) \vee (q$ [illegible] a truth assignment v that sets p to T and q to F.

To figure out the truth value of α under this assignment, we build a truthtable with one column for every element in a construction sequence of α as follows:

	p	q	$(p \to q)$	$(q \to p)$	$((p \to q) \vee (q \to p))$
v	T	F	F	T	T

We start with filling the first columns using the truth assignment v and then successively fill the other columns using the truth table for the connectives.

Extending truth assignments to WFF – examples

Say, we have [QR code]riables p and q, and a formula $\alpha = ((p \to q)$[QR code]nd a truth assignment v that sets p to T and q to F[QR code]

p	q	$(p \to q)$	$(q \to p)$	$((p \to q) \vee (q \to p))$
T	F	F		T

$v(\alpha) = T$

Extending truth assignments to WFF – examples

Say, we have ... riables p and q, and a formula $\alpha = ((p \to q)$ ... nd a truth assignment v that sets p to T and q to F

p	q	$(p \to q)$	$(q \to p)$	$((p \to q) \vee (q \to p))$
T	F	F		T
T	T	T	T	T
F	T	T	F	T
F	F	T	T	T

$v(\alpha) = T$

α evaluates to T under every truth assignment, α is a tautology

Logical equivalence

Definition: We call two propositional formulas (statements) α and β logically equivalent if they have the same truth table (that is, they evaluate to the same truth value under all truth assignments).

Show that the following pairs of formulas α and β are logically equivalent:

- $\alpha = p$ and $\beta = (\neg(\neg p))$
- $\alpha = (p \rightarrow q)$ and $\beta = ((\neg q) \rightarrow (\neg p))$
- $\alpha = (\neg(p \wedge q))$ and $\beta = ((\neg p) \vee (\neg q))$
- $\alpha = (\neg(p \vee q))$ and $\beta = ((\neg p) \wedge (\neg q))$

Exercises

Show that the [illegible] of formulas α and β are logically equivalent:

- $\alpha = p$ and $\beta = (\neg(\neg p))$
- $\alpha = (p \to q)$ and $\beta = ((\neg q) \to (\neg p))$

If it rains then the street is wet.

If the street is not wet, I can conclude that it's not raining.

p	q	$(p \to q)$	$(\neg p)$	$(\neg q)$	$((\neg q) \to (\neg p))$
T	T	[illegible]	F	T	F
T	F	F	[illegible]	[illegible]	[illegible]
F	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
F	F	T	T	T	T

Exercises

Show that th… s of formulas α and β are logically equivalent:

- $\alpha = p$ and $\beta = (\neg(\neg p))$
- $\alpha = (p \to q)$ and $\beta = ((\neg q) \to (\neg p))$
- $\alpha = (\neg(p \wedge q))$ and $\beta = ((\neg p) \vee (\neg q))$
- $\alpha = (\neg(p \vee q))$ and $\beta = ((\neg p) \wedge (\neg q))$

Exercise!

prove these!

Talking: Ruth Urner

Show that the following pairs of formulas α and β are logically equivalent:

- $\alpha = (p \vee q)$ and $\beta = (\neg((\neg p) \wedge (\neg q)))$
- $\alpha = (p \rightarrow q)$ and $\beta = ((\neg p) \vee q)$
- $\alpha = (\neg(p \rightarrow q))$ and $\beta = (p \wedge (\neg q))$
- $\alpha = (p \leftrightarrow q)$ and $\beta = ((p \rightarrow q) \wedge (q \rightarrow p))$
- $\alpha = (p \oplus q)$ and $\beta = (\neg(p \leftrightarrow q))$

More exercises

Show that the pairs of formulas α and β are logically equivalent:

- $\alpha = (p \vee q)$ and $\beta = (\neg((\neg p) \wedge (\neg q)))$
- $\alpha = (p \rightarrow q)$ and $\beta = ((\neg p) \vee q)$
- $\alpha = (\neg(p \rightarrow q))$ and $\beta = (p \wedge (\neg q))$
- $\alpha = (p \leftrightarrow q)$ and $\beta = ((p \rightarrow q) \wedge (q \rightarrow p))$
- $\alpha = (p \oplus q)$ and $\beta = (\neg(p \leftrightarrow q))$

Exercise: prove these equivalences.

Show that the follow[illegible]f formulas α and β are not logically equivalent:

- $\alpha = (p \vee q)$ and $\beta = (\neg(p \wedge q))$

- $\alpha = (p \rightarrow q)$ and $\beta = ((\neg p) \rightarrow (\neg q))$

- $\alpha = (p \oplus q)$ and $\beta = (p \vee q)$

More exercises: Show the distributive laws

Show that the follow[illegible] of formulas α and β are logically equivalent:

- $\alpha = (p \wedge (q \vee r))$ and $\beta = ((p \wedge q) \vee (p \wedge r))$

- $\alpha = (p \vee (q \wedge r))$ and $\beta = ((p \vee q) \wedge (p \vee r))$

These two logical equivalences are called distributive laws of propositional logic.

More exercises: Show the distributive laws

Show that the pairs of formulas α and β are logically equivalent:

have identical evaluations for all truth assignments

- $\alpha = (p \wedge (q \vee r))$ and $\beta = ((p \wedge q) \vee (p \wedge r))$

$2^3 = 8$ possible truth assignments for 3 variables

p	q	r	$(q \vee r)$		$(p \wedge q)$	$(p \wedge r)$	β
T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T
T	F	T	T	T	F	T	T
T	F	F	F	F	F	F	F
F	T	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	F	T	T	F	F	F	F
F	F	F	F	F	F	F	F

Tautologies and contradictions

We call a propositional formula a tautology if it evaluates to true under every truth assignment.

We call a propositional formula a contradiction if it evaluates to false under every truth assignment.

Tautologies and contradictions

We call a pro... ...ula a tautology if it evaluates to true under every t...

Example: ... $(\neg p))$

We call a propositional formula a contradiction if it evaluates to false under every truth assignment.

Example: $(p \wedge (\neg p))$

Talking: Ruth Urner

Extending truth assignments to WFF – examples

Say, we have two variables p and q, and a formula $\alpha = ((p \to q) \vee (q \to p))$, and a truth assignment v that sets p to T and q to F.

p	q	$(p \to q)$	$(q \to p)$	$((p \to q) \vee (q \to p))$
T	F	F	T	T
T	T	T	T	T
F	T	T	F	T
F	F	T	T	T

$v(\alpha) = T$

α evaluates to T under every truth assignment, α is a tautology

First order/Predicate logic–motivation

Assumptions:

- All people are mortal
- I am a person

(Sad) **Conclusion:**

- I am mortal

Question: Can propositional logic explain the pattern used in this example of reasoning?

Answer: **No.** Propositional logic can only relate truths or falsehood of statements as a whole. It does not provide as a means of reasoning about objects and properties that these objects may have.

First order/Predicate logic–motivation

Assumptions:
- All people are mortal
- I am a person

(Sad) **Conclusion:**
- I am mortal

To explain this example of reasoning, we need means to refer to the inner structure of these statements (not just the statements as a whole). First order logic allows us to

- refer to objects (for example people)
- state that objects have certain properties (for example being mortal)
- make statements about relationships between objects
- quantify over objects (for example state that something holds for all (all people are mortal) objects or that there exists an object that has a certain property)

In mathematical reas[illegible] make statements that quantify over objects, for example:

- **Definition:** A g[illegible] (V, E) is connected, if for every pair of vertices $v_1, v_2 \in V$ there exists a path from v_1 to v_2.

- To express that there are infinitely prime numbers, we say: For every prime number p, there exists a prime number p' with $p' > p$.

- If I want to show that some propositional formula is not a tautology, I need to show that there exists truth assignment to its variables, for which α evaluates to false.

First order/Predicate logic–quantifiers

In general, I recomm… you spell out mathematical statements and proo… nglish language.

Sometimes, it is useful to have some shorthand for statements. We use

- the symbol $\forall$ to stand for for all, and
- the symbol $\exists$ to stand for there exists.

First order/Predicate logic–negating statements with quantifiers

It is important to rea[illegible] when we negate statements in first order logic the two q[illegible] $\forall$ and $\exists$ "change roles":

- $\neg(\forall x\ P(x))$ is equivalent to $\exists x\ (\neg P(x))$

- $\neg(\exists x\ P(x))$ is equivalent to $\forall x\ (\neg P(x))$

First order/Predicate logic–negating statements with quantifiers

Talking: Ruth Urner

It is importan… t, when we negate statements in first order logic th… s $\forall$ and $\exists$ "change roles":

- $\neg(\forall x\ P(x))$ is equivalent to $\exists x\ (\neg P(x))$

Example: … cities having population larger than 1 000 000.

It's not that all cities have population larger than 1000 000.

There exist a city with less than 1000 000 inhabitants.

First order/Predicate logic–negating statements with quantifiers

It is important ... t, when we negate statements in first order logic th... s $\forall$ and $\exists$ "change roles":

- $\neg(\forall x\ P(x))$ is equivalent to $\exists x\ (\neg P(x))$

- $\boxed{\neg(\exists x\ P(x))}$ is equivalent to $\boxed{\forall x\ (\neg P(x))}$

There does not exist a student in my class that is confused.

All students in my class are not confused.

First order/Predicate logic–negating statements with quantifiers

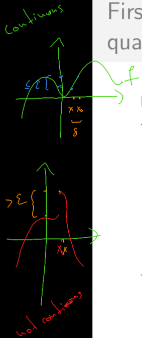
Example: The definition [illegible]ty for a function $f : \mathbb{R} \to \mathbb{R}$ at a point $x_0 \in \mathbb{R}$ can be compactly [illegible]ollows:

$$\forall \epsilon > 0 \; \exists \delta > 0 \; [illegible] - x_0| < \delta \to |f(x) - f(x_0)| < \epsilon)$$

To prove that function $f : \mathbb{R} \to \mathbb{R}$ is not continuous in x_0, we need to show:

$$\exists \epsilon > 0 \; \forall \delta > 0 \; \exists x \in \mathbb{R} \; (|x - x_0| < \delta \land |f(x) - f(x_0)| \geq \epsilon)$$

First order/Predicate logic–negating statements with quantifiers



Example: The [obscured]uity for a function $f : \mathbb{R} \to \mathbb{R}$ at a point $x_0 \in \mathbb{R}$ can be c[obscured]s follows:

$$\forall \epsilon > \ldots (|x - x_0| < \delta \to |f(x) - f(x_0)| < \epsilon)$$

To prove that function $f : \mathbb{R} \to \mathbb{R}$ is not continuous in x_0, we need to show:

$$\exists \epsilon > 0 \forall \delta > 0 \exists x \in \mathbb{R}\ (|x - x_0| < \delta \wedge |f(x) - f(x_0)| \geq \epsilon)$$

Mathematical statements and proofs

Reading: ITC Section 0.3 andd 0.4

Basic elements of mathematical text

Definition

A statement that cle[illegible]es an object/structure/concept based on previously d[illegible]ms.

Theorem

A statement that has been proven to be true.

Proof

A clean, deductive argument for why a statement is true.

Lemma

A "helper theorem", typically only stated as a step in a proof of some theorem.

Examples of what not to do..

Examples of what not to do..

Mouse Select Text Draw Stamp Spotlig Save

Who can see what you

screen sharing Stop Share

Talking: Ruth Urner

Definition
A statement ... nes an object/structure/concept based on prev ... erms.

Theorem
A statement that has been proven to be true.

Proof
A clean, deductive argument for why a statement is true.

what not to do: "proof by example"

Example: " ... natural numbers are prime"

~~"Proof" (by example): 1, 3, 5, 7, ...~~

To refute a statement ... ing, a counter-example is a sufficient proof.

Counter-example to the claim: 9 is odd and 9 = 3·3 is not prime. Thus the claim is false.