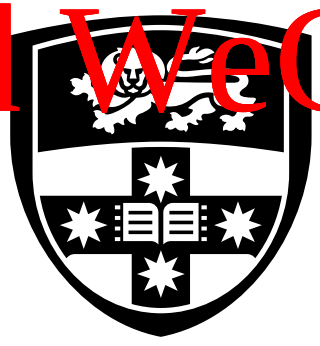
COMP2022: Formal Languages and Logic

2018, Semester 2, Week 9



THE UNIVERSITY OF SYDNEY

Outline

- Revision
-
-
- Logic

FIRST and FOLLOW sets

In order to fill in the entries of the table-drived parser, we need to compute some FIRST and FOLLOW sets.

FIR deriv production of G (i.e. the *right hand side*

$FOLLOW(V)$ is the set of all terminals which w the variable V at any stage of the derivation. Needed whenever V can derive ε.

Why do we need them?

Let the current input symbol be b, and the top of the stack be X.

There are two ways we might try to derive a string starting with b:

1. k for

2. If X can be derived to ε, then we mig string starting with b by using the sy on the stack.
   i.e. If any of the production rules $X \to \alpha$ had $\varepsilon \in FIRST(\alpha)$, then we also look at $FOLLOW(X)$

Another way to calculate FIRST sets

Let $A \to X_1 ... X_n$ be a production of A, where X_i could be a terminal or variable.

We re

- 1:
- X_1
- $FIRST(X_1) \setminus \{\varepsilon\}$
- If X_1 is a variable $\varepsilon \in FIRST(X_1$ also contains $FIRST(X_2 ... X_n)$

Don't forget that $FIRST(\varepsilon) = \{\varepsilon\}$, so if every X_i can generate ε, then rule 3 will (eventually) give us $\varepsilon \in FIRST(X_1 ... X_n)$

Another way to calculate FOLLOW sets

If $\varepsilon \in FIRST(\alpha)$ for some production rule $A \rightarrow \alpha$ then we need to compute $FOLLOW(A)$

Cons *t hand side.*

Let $V \rightarrow Y_1 ... Y_n A Z_1 ... Z_m$ (Y_i, Z_i ca

- If A is the start symbol, then $\$ \in$
- $FIRST(Z_1 ... Z_m) \setminus \{\varepsilon\} \subseteq FOLLOW(A)$
- If $\varepsilon \in FIRST(Z_1 ... Z_m)$ then $FOLLOW(V) \subseteq FOLLOW(A)$

EXAMPLES

Constructing the parse table

Rows: one for each variable of the grammar

Colu

strin

Steps to fill the table T:

1. If there is a rule $R \to \alpha$ with $b \in FIRST(\alpha)$, then put α in $T[R, b]$
2. If there is a rule $R \to \alpha$ with $\varepsilon \in FIRST(\alpha)$ and $b \in FOLLOW(R)$, then put α in $T[R, b]$

EXAMPLE

Left factoring: definition

If a string w appears on the left of several rules for a variable A:

$$A \quad wX_1 \quad \ldots \quad wX_1$$

The $\ldots$:

$$A \to wA'$$
$$A' \to X_1 \ldots$$

Any other rules produced by A are unaffected.

Eliminating left recursion

Let α β be arbitrary strings of terminals and/or variables

Let A be a variable, and R a new variable

If A

It can be replaced with:

$$A \to \beta R$$

$$R \to \alpha R \mid \varepsilon$$

Outline

- Revision
-
-
- Logic

Beyond CFL

An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

$$C$$

$$CB \to E$$

$$aD \to Da$$

$$AD \to AC$$

$$aE \to Ea$$

$$AE \to \varepsilon$$

Beyond CFL

An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

How can we derive $aaaa$?

$$C$$

$$CB \to E$$

$$aD \to Da$$

$$AD \to AC$$

$$aE \to Ea$$

$$AE \to \varepsilon$$

Beyond CFL

An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

How can we derive $aaaa$?

$$C$$

$$CB \to E$$

$$aD \to Da$$

$$AD \to AC$$

$$aE \to Ea$$

$$AE \to \varepsilon$$

Beyond CFL

An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

How can we derive $aaaa$?

$$C$$

$$CB \to E$$

$$aD \to Da$$

$$AD \to AC$$

$$aE \to Ea$$

$$AE \to \varepsilon$$

Beyond CFL

An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

How can we derive $aaaa$?

$$C$$

$$CB \to E$$

$$aD \to Da$$

$$AD \to AC$$

$$aE \to Ea$$

$$AE \to \varepsilon$$

Beyond CFL

An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

How can we derive $aaaa$?

$$C$$

$$CB \to E$$

$$aD \to Da$$

$$AD \to AC$$

$$aE \to Ea$$

$$AE \to \varepsilon$$

$\Rightarrow$

Beyond CFL

An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

How can we derive $aaaa$?

C

$CB \to E$

$aD \to Da$

$AD \to AC$

$aE \to Ea$

$AE \to \varepsilon$

$\Rightarrow$

$\Rightarrow$

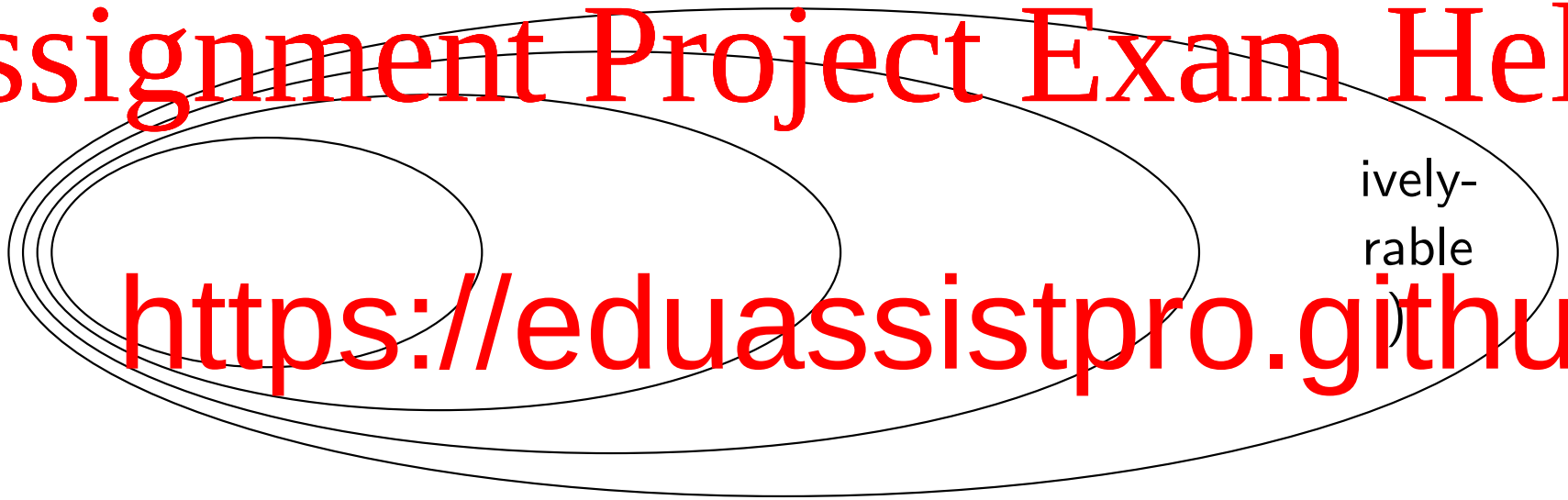
$\Rightarrow$

$\Rightarrow AaaaaE \Rightarrow AaaaEa$

$\Rightarrow AaaEaa \Rightarrow AaEaaa$

$\Rightarrow AEaaaa \Rightarrow aaaa$

Chomsky Hierarchy



Type 0 grammars (unrestricted grammars) ar
describe the set of *recursively enumerable* languages.

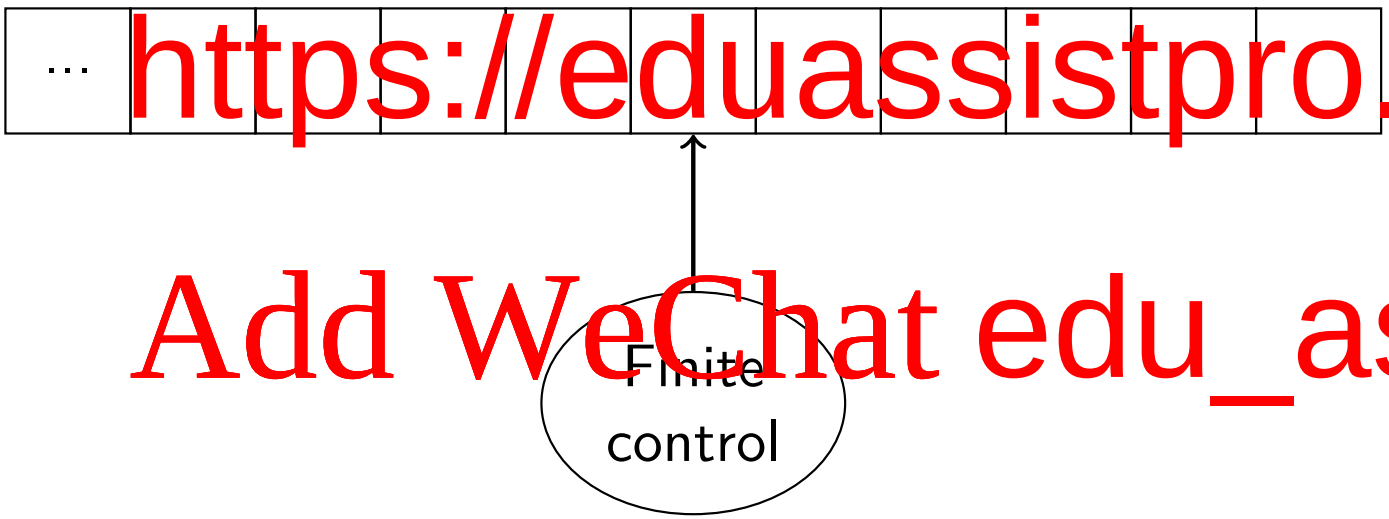
Turing Machines

- Finite automata have a finite memory. They recognise regular languages (regular grammars)
- Push Down Automata: Add a LIFO stack (infinite memory,
- limitation of access) we get Turing machin computers. TM recognise recursively en (unrestricted grammars)
- Alan Turing (1912-1954) mathematician and logician
  - http://www.alanturing.net
  - Turing machines introduced in his 1936 article

Turing Machines: schematic view

Turing machines have infinite memory: a tape.

- Can read and write symbols on it or do nothing
- Can move to the right or to the left along the tape



i.e. it is a finite state automaton attached to an infinite tape

Formal Definition

A Turing machine is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- Q is a finite set of states
-
-
-
-
  - L means 'move left', R means 'm
- $q_0 \in Q$ is the start state
- B is a special symbol of Γ, the *bl*
- $F \subseteq Q$ is the set of accept states

Turing machine for $\{0^n 1^n \mid n > 0\}$

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, X, Y, B\}$
-
-
-
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is giv

	0	1	X	Y	B
q_0	(q_1, X, R)				
q_1	$(q_1, 0, R)$	(q_2, Y, L)		(q_1, Y, R)	
q_2	$(q_2, 0, L)$		(q_0, X, R)	(q_2, Y, L)	
q_3				(q_3, Y, R)	(q_4, B, R)
q_4					

Trying M on 0011

Semi-formal description of M

Assuming:

- The tape initially contains only 0s, 1s and blanks
- We start at the first non-blank symbol on the tape

1.
2.
3.
4. Replace the 1 with a Y
5. Move left until we find the X marker
6. Move right once. If we are on a 0, goto (2)
7. Move right across the tape, skipping X and Y.
   - If a 1 or 0 is scanned, reject
   - If a blank is scanned, accept

Turing machine for a non-CFL: $\{a^n b^n c^n \mid n > 0\}$

- $Q = \{q_0, q_a, q_b, q_c, q_y, q_z, q_f\}$
- $\Sigma = \{a, b, c\}$
- $\Gamma = \{a, b, c, X, Y, Z, B\}$
-
-
-
-
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is giv

	a	b	c	X			
q_0	(q_a, X, R)						
q_a	(q_a, a, R)	(q_b, Y, R)			a		
q_b		(q_b, b, R)	(q_c, Z, L)			(q_b, Z, R)	
q_c	(q_c, a, L)	(q_c, b, L)		(q_0, X, R)	(q_c, Y, L)	(q_c, Z, L)	
q_y					(q_y, Y, R)	(q_z, Z, R)	
q_z						(q_z, Z, R)	(q_f, B, R)
q_f							

Trying it on $aabbcc$

Language accepted by a Turing Machine

- The language accepted by a Turing machine M is the set of strings such that M started on the left of the input and
-
- input w, then M does not accept
- Languages accepted by a Turing Machin enumerable"
- Unrestricted grammars (type 0)

Turing machine to add 2 numbers

A number n is represented by n 0's (*unary*)

The two numbers are separated by a 1

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
-
-
-
- $F = \{q_3\}$
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is given

	0	1	B
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$	(q_3, B, R)
q_1	$(q_1, 0, R)$		(q_2, B, L)
q_2	(q_3, B, R)		
q_3			

Semi-formal description

1. Move to the right until we reach a 1 (skipping 0s)
2. Rewrite the 1 with a 0
3.
4.

For example, if our tape originally read
reads $\ldots B0000BB\ldots$ i.e. $3 + 2 = 5$

Turing machines can make calculations! Note the similarity with computers.

Left-bounded tapes

Can a Turing Machine with an infinite tape in *both* directions recognise more languages than tape that is only unbounded in one direction?

Does n
one wi
integ

Cou
integer with a positive integer:

1	2	3	4		
0	-1	1	-2	2	

So, a tape unbounded to the left does not have more positions.

This is what we mean when we say a set is *enumerable*.

Left-bounded tapes

Can a Turing Machine with an infinite tape in *both* directions recognise more languages than tape that is only unbounded in one direction?

We co
posit
but th

Add states to the TM such that whenever we want to the left edge of the tape, we simply shift everything o one position to the right instead.

- We will need to place a *marker* at the far right of the input, so we know when to stop shifting.

Multi-tape Turing Machines

Can a multi-tape Turing Machine recognise more languages than a single-tape TM?

Multi-tape Turing Machines

Can a multi-tape Turing Machine recognise more languages than a single-tape TM?

NO

Proo
mult

- Add markers symbolising the start and end
- Add markers symbolising the current TM p
- Add states which simulate switching betw
- Add states which extend these (finite) stretches of tape (by shifting everything beyond it by one space)

Context-free languages

Can a TM recognise every context-free language?

Context-free languages

Can a TM recognise every context-free language?

Yes

Proo

-
- stack
- Repeatedly move between the tape and the e
- i.e. each state in the PDA will have two states in t
  - One for the current input symbol, one for th

Context-free languages

Can a TM recognise every context-free language?

Yes

Proo

-
- stack
- Repeatedly move between the tape and the e
- i.e. each state in the PDA will have two states in t
  - One for the current input symbol, one for th

Note: we will need a *non-deterministic* TM (our PDA are non-deterministic)

Non-deterministic TM

Is a Non-deterministic TM more powerful than a deterministic one?

Non-deterministic TM

Is a Non-deterministic TM more powerful than a deterministic one?

NO

Proo

- Tape 2 simulates a particular computatio
- Tape 3 encodes the path through the tree of po through the NTM

Non-deterministic TM

Is a Non-deterministic TM more powerful than a deterministic one?

NO

Proo

- Tape 2 simulates a particular computatio
- Tape 3 encodes the path through the tree of po through the NTM

This DTM will (very slowly!) explore all the possible paths through the NTM, from shortest to longest.

Universal Turing machine

- It is possible to encode a Turing machine or to write a "program" to describe a Turing machine.
- $\langle M \rangle$
- Exactly what general computers do:
  - They accept a program P
  - and the input to that program
  - and produces the output of P o

- Again, note the similarity between TM and computers

Church-Turing Thesis

"Every effectively calculable function is a computable function."

Esse t to the se

So:

- TM are a very precise model for definition of al
- TM can do *everything* that a compu
- However, even a Turing machine cannot solve all problems:
  - some are beyond theoretical limits of computation

Computability, Decidability, Intractability

- Decidable $\subset$ TM-computable
- (e)
- It is *decidable* if and only if the Turing ma
  halt on any input string (the Halting Proble
-
- *Intractability*: The efficiency problem – can we solve the problem in a reasonable time (e.g. polynomial vs exponential)?

Example of undecidable problem: The Halting problem

- Halting problem: Is there an algorithm that can decide

-
  - $f(x)\{\ return\ 2x+1;\}$, will f h
  - $h(x)\{\ while(true)\ \{x{+}{+};\}$
  - $g(x)\{\ for(i=x;\ i<10;\ i{-}{-}$

- Answer: NO (Church 1936, Turing 1937)

Proving the Halting Problem is undecidable

Suppose the Halting problem is decidable.

The
$H(a$ at
on inp uld halt

The language L of H is:

$\{a, b \mid a$ represents a TM, b is an inpu $\}$

Proving the Halting Problem is undecidable

$H(a, b)$ accepts iff TM a halts on input b

Let $X(c)$

- If $H(c, c)$ accepts, then loop forever
- If $H(c, c)$ rejects, then halt

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H – i.e. what does X do when given its *own* representation as input.

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H – i.e. what does X do when given its *own* representation as input.

Case 1

beca , X will loop f

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H – i.e. what does X do when given its own representation as input.

Case 1

beca , X will loop f

Case 2: If $H(X, X)$ rejects, then X c
because H is a solution to the halting problem. X will halt because $H(X, X)$ rejected.

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H i.e. what does X do when given its own representation as input.

Case 1
beca , X will loop f

Case 2: If $H(X, X)$ rejects, then X c
because H is a solution to the Halting problem. X will halt because $H(X, X)$ rejected.

Both cases lead to contradictions, so the assumption was incorrect.
i.e. The Halting problem cannot be decidable.

Example of undecidable problem: The Halting problem

- Halting problem: Is there an algorithm that can decide whether the execution of an arbitrary program halts on an

  - $f(x)\{\ return\ 2x+1;\}$, will f h
  - $h(x)\{\ while(true)\ \{x+=1;\}$.
  - $g(x)\{\ for(x=x;i<10;x+=$

- Answer: NO (Church 1936, Turing 1937)

- How can you restrict the halting problem to be decidable?

Computability, Decidability, Intractability

Outline

- Revision
-
-
- **Logic**

Logic: Introductory Example

What do you think of the following argument?

1. If a number is a multiple of 2 3 4, then it is a multiple of

2.

Logic: Introductory Example

What do you think of the following argument?

1. If a number is a multiple of 2 3 4, then it is a multiple of

Expressing this formally:

$$M \to (P \wedge Q$$

$$(P \wedge Q \wedge R) \to M$$

Logic

- Logic is a language used to make some disciplines scientific by providing a way to deduce knowledge from a relatively small number of explicitly stated facts or hypotheses
-
-
- enabling analysis of the argument structu
- Provides a way to *reason* about the c knowledge rigourously. i.e. How to make a ju validity of the argument
- Focus on validity (correctness) of the argument *form*, rather than its *contents*

Logic in real arguments

Argument 1: If I play cricket or I go to work, then I will not be going shopping. Therefore, if I go shopping, then I would neither play cricket nor would I go to work

- Q: I go to work
- R: I go shopping

If P or Q, then not R	$(P \vee Q) \rightarrow \neg R$
Therefore, if R then not P and not Q	$R \rightarrow (\neg P \wedge \neg Q)$

Logic in real arguments

Argument 2: An object remaining stationary or moving at a constant velocity means that there is no net external force acting upon it. Therefore, if there is a net force acting upon the object, then it i

-
- Q: The object is moving at a constant velocity
- R: There is a net external force acting upon the o

If P or Q, then not R	$(P \vee Q) \rightarrow \neg R$
Therefore, if R then not P and not Q	$R \rightarrow (\neg P \wedge \neg Q)$

Arugment, Premises, Deduction

An *argument* is a claim. It is composed of statements

- If P or Q, then not R
-

The

- If P or Q, then not R

The last statement is the *conclusion* o
to be *deduced* from the premises

- Therefore, if R then not P and not Q

Propositional Logic

- Formalise sentences
  - Propositions and connectives
  -

Sem

- Truth tables
- Tautologies

Formal Reasoning

Propositions

A *Proposition* is the underlying meaning of a declarative sentence (a sentence which is either **true** or **false**)

- Mammals are warm-blooded
-
-
-

But these are not propositions

- Can you show me the way to Redfern?
- Pay your bills on time
- Stop talking!

Well-formed formula (wff) syntax

A well-formed formaula (wff) is an expression with the correct syntax (i.e. it is a string from the language of wff)

Trut

Ato

Complex propositions are built up using con

If P and Q are wff, then (P), $\neg P$, $(P \vee$ $\wedge$, $(P \leftrightarrow Q)$ are all also wff

To make it easier to refer to complex wff, we can set labels for them by writing, for example $Z = ((P \rightarrow Q) \vee Q)$

Semantics (truth tables)

Truth tables define the possible values that a wff can take, depending on the values of the atomic propositions that it contains.

The m
a wff is it

P	Q	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0

Connectives

Negation (not)	$\neg P$ is true iff P is false
Conjunction (and)	$(P \wedge Q)$ is true iff both P and Q are true
Disjunction (or)	$(P \vee Q)$ is true iff P or Q is true
Im	Q is
Eq	th

P	Q	P	$(P \wedge Q)$	$(P \vee Q)$		
1	1					
1	0					
0	1					
0	0					

Connectives

Negation (not)	$\neg P$ is true iff P is false
Conjunction (and)	$(P \wedge Q)$ is true iff both P and Q are true
Disjunction (or)	$(P \vee Q)$ is true iff P or Q is true
Im	Q is
Eq	th

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$		
1	1	0				
1	0	0				
0	1	1				
0	0	1				

Connectives

Negation (not)	$\neg P$ is true iff P is false
Conjunction (and)	$(P \wedge Q)$ is true iff both P and Q are true
Disjunction (or)	$(P \vee Q)$ is true iff P or Q is true
Im	Q is
Eq	th

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$		
1	1	0	1			
1	0	0	0			
0	1	1	0			
0	0	1	0			

Connectives

Negation (not)	$\neg P$ is true iff P is false
Conjunction (and)	$(P \wedge Q)$ is true iff both P and Q are true
Disjunction (or)	$(P \vee Q)$ is true iff P or Q is true
Im	Q is
Eq	th

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$		
1	1	0	1	1		
1	0	0	0	1		
0	1	1	0	1		
0	0	1	0	0		

Connectives

Negation (not)	$\neg P$ is true iff P is false
Conjunction (and)	$(P \wedge Q)$ is true iff both P and Q are true
Disjunction (or)	$(P \vee Q)$ is true iff P or Q is true
Im	Q is
Eq	th

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$		
1	1	0	1	1		
1	0	0	0	1	0	
0	1	1	0	1	1	
0	0	1	0	0	1	

Connectives

Negation (not)	$\neg P$ is true iff P is false
Conjunction (and)	$(P \wedge Q)$ is true iff both P and Q are true
Disjunction (or)	$(P \vee Q)$ is true iff P or Q is true
Im	Q is
Eq	th

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$		
1	1	0	1	1		
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

Example

Construct the truth table for

P	Q	$(P \wedge Q)$	$\neg Q$	$((P \wedge Q) \vee \neg Q$			X
1	1						
1	0						
0	1						
0	0						

Formalising English as logic

On Friday morning Mary went for a walk, in the afternoon she went to work and on Saturday she stayed home while her house was being painted.

Alth

next

If Gromit is not in his kennel, then he is reading the pape

Increased spending overheats the economy

Increased spending coupled with tax cuts overheats the economy

Formalising English as logic

On Friday morning Mary went for a walk, in the afternoon she went to work and on Saturday she stayed home while her house was being painted.

$$(W \wedge J \wedge H \wedge P)$$

Alth

next

If Gromit is not in his kennel, then he is reading the pape

Increased spending overheats the economy

Increased spending coupled with tax cuts overheats the economy

Formalising English as logic

On Friday morning Mary went for a walk, in the afternoon she went to work and on Saturday she stayed home while her house was being painted.

$$(W \wedge J \wedge H \wedge P)$$

Alth

next

If Gromit is not in his kennel, then he is reading the pape

Increased spending overheats the economy

Increased spending coupled with tax cuts overheats the economy

Formalising English as logic

On Friday morning Mary went for a walk, in the afternoon she went to work and on Saturday she stayed home while her house was being painted.

$$(W \wedge J \wedge H \wedge P)$$

Alth

next

If Gromit is not in his kennel, then he is reading the pape

$$(\neg K \rightarrow P)$$

Increased spending overheats the economy

Increased spending coupled with tax cuts overheats the economy

Formalising English as logic

On Friday morning Mary went for a walk, in the afternoon she went to work and on Saturday she stayed home while her house was being painted.

$$(W \wedge J \wedge H \wedge P)$$

Alth

next

If Gromit is not in his kennel, then he is reading the pape

$$(\neg K \rightarrow P)$$

Increased spending overheats the economy

$$(S \rightarrow E)$$

Increased spending coupled with tax cuts overheats the economy

Formalising English as logic

On Friday morning Mary went for a walk, in the afternoon she went to work and on Saturday she stayed home while her house was being painted.

$$(W \wedge J \wedge H \wedge P)$$

Alth

next

If Gromit is not in his kennel, then he is reading the pape

$$(\neg K \rightarrow P)$$

Increased spending overheats the economy

$$(S \rightarrow E)$$

Increased spending coupled with tax cuts overheats the economy

$$((S \wedge T) \rightarrow E)$$

Possible interpretations in English

$\neg P$	not P P does not hold it is not the case that P P is false
$(P$	P while Q P despite Q P yet Q P although Q
$(P \vee Q)$	P or Q P or Q or both P and/or Q

Possible interpretations in English

$(P \rightarrow Q)$	If P then Q Q if P P implies Q
$(P \leftrightarrow Q)$	P if and only if Q P iff Q P is necessary and sufficien

Be careful with disjunction!

We need to be careful with the definition of disjunction

- Inclusive "or": $(P \quad Q)$
-

Be careful with disjunction!

We need to be careful with the definition of disjunction

- Inclusive "or": $(P \quad Q)$
-

Examples:

You can go to the airport by taxi or bus

You can choose to save your money or your life

The error is in the program or the sensor data

The program or the sensor data are erroneous

Be careful with disjunction!

We need to be careful with the definition of disjunction

- Inclusive "or": $(P \quad Q)$
-

Examples:

You can go to the airport by taxi or bus

You can choose to save your money or your life

The error is in the program or the sensor data

The program or the sensor data are erroneous

Be careful with disjunction!

- We need to be careful with the definition of disjunction
  - Inclusive "or": $(P \quad Q)$
  -

Examples:

You can go to the airport by taxi or bus

You can choose to save your money or your life

The error is in the program or the sensor data

The program or the sensor data are erroneous

Be careful with disjunction!

- We need to be careful with the definition of disjunction
  - Inclusive "or": $(P \quad Q)$
  -

Examples:

You can go to the airport by taxi or bus

You can choose to save your money or your life

The error is in the program or the sensor data

The program or the sensor data are erroneous

Be careful with disjunction!

We need to be careful with the definition of disjunction

- Inclusive "or": $(P \quad Q)$
-

Examples:

You can go to the airport by taxi or bus

You can choose to save your money or your life

The error is in the program or the sensor data

The program or the sensor data are erroneous Inclusive?

Be careful with implication/equivalence!

Sometimes in english the syntax and terms used do not reflect the logical meaning

Be careful with implication/equivalence!

Sometimes in english the syntax and terms used do not reflect the logical meaning

"Eat
worl

- implication (it's unlikely that they are tryin
  destroying rainforests implies eating fast f

Be careful with implication/equivalence!

Sometimes in english the syntax and terms used do not reflect the logical meaning

"Eat
worl

- implication (it's unlikely that they are tryin
  destroying rainforests implies eating fast f

"I will give you a lift to the city, if you are going to the city"

- This is really an equivalence, not the implication that the sentence structure suggests.

More examples

Max is home and Claire is at the library

Max is home or Claire is at the library

Ma

Ma

Ma

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

More examples

Max is home and Claire is at the library $(H \wedge L)$

Max is home or Claire is at the library

Ma

Ma

Ma

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	
Ma	
Ma	
Max is not home nor Claire is at the library	
Max is home although Claire is at the library	
Max is home unless Claire is at the library	

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	)
Ma	
Ma	
Max is not home nor Claire is at the library	
Max is home although Claire is at the library	
Max is home unless Claire is at the library	

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	)
Ma	)
Ma	

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	)
Ma	)
Ma	)

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	)
Ma	)
Ma	)

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	)
Ma	)
Ma	)

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	)
Ma	)
Ma	)

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \quad L)$
Ma	)
Ma	)
Ma	)

Max is not home nor Claire is at the library

Max is home although Claire is at the library

Max is home unless Claire is at the library

Announcements

- Assignment 2

- Week 10 quiz tests concepts from the Week 8 t