COMP2022: Formal Languages and Logic

2017, Semester 1, Week 7

Joseph Godbehere

Adapted from slides by A/Prof Kalina Yacef

April 26, 2017

THE UNIVERSITY OF SYDNEY

Announcements

Tuesday 25th April is a public holiday (if you're reading this in the past!)

- Replacement lecture:
  - 4pm – 6pm on Wednesday 26th April
  - PNR LT1 (i.e. the LT *next* to this one)
- Tuesday tutorials:
  - Join Wednesday / Thursday tutorials *if there is space*
    - See Ed for times/locations
  - Otherwise work through the exercises independently
- No quiz this week
- Quiz week 8 will assess content from weeks 6 and/or 7

Feedback

Detailed feedback next week (when there's more time to spare)

- 94% thought the lectures were helpful
- 92% thought the tutorials were helpful

Comments

- Quizzes are too short $\rightarrow$ quizzes will be longer
- Lecture typos $\rightarrow$ I'll try!
- Desire for chocolate $\rightarrow$ mmm
- Voice is too quiet $\rightarrow$ ask me to turn up the mic!

Outline

- Using a table driven parser to build a Parse Tree
- Applying semantics to a Parse Tree
- Regular Grammars
- Beyond Context Free Languages
- Introduction to Propositional Logic

Descent table-driven LL(1) parser

```
loop
   T = symbol on top of the stack
   c = current input symbol
   if T == c == $ then accept
   else if T is a terminal or T == $ then
       if T == c then pop T and consume c
       else error
   else if P[T,c] = α is defined then
       pop T and push α onto the stack
               //(in reverse order)
   else error
endloop
```

Modifying the algorithm

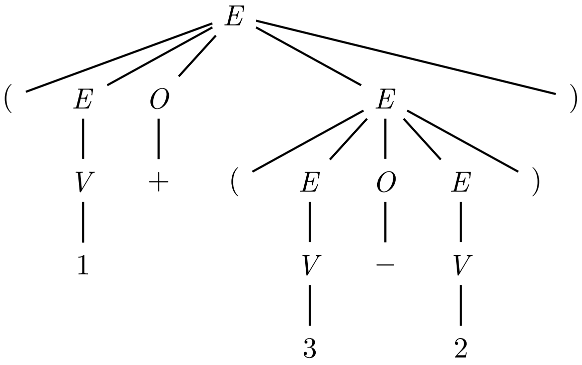
- Add another stack, called the *Expression stack*. This will contain fragments of the tree.
- Whenever (before) we push a production rule onto the parser's stack, we push an *Action Token* onto the stack first.
  - The Action Token remembers which rule we just applied
- Whenever we pop a terminal from the stack, push a single tree node corresponding to that terminal onto the Expression stack
- Whenever an Action Token is on top of the stack, pop it off. Suppose the corresponding rule is $A \to \beta$, then:
  - Pop $|\beta|$ trees from the Expression stack
  - Create a tree node of the appropriate type for $A \to \beta$, using the popped trees as it's children
  - Push the new tree onto the Expression stack

At the end, the parse tree should be the only item left in the stack.

Parse Trees

Example of a parse tree for $(1 + (3 - 2))$ using this simple LL(1) grammar

$$E \to (EOE) \mid V$$
$$O \to + \mid -$$
$$V \to 0 \mid 1 \mid 2 \mid 3$$



Syntax

Syntax describes the *structure* of a string.

- e.g. "$1 + 1$" is a syntactically correct arithmetic expression
- e.g. "Colourless green ideas sleep furiously" is grammatically correct English

Semantics

Semantics describes the *meaning* of a string.

- The meaning might depend on the *Domain of Interpretation*
  - e.g. in base 10 arithmetic "$1+1$" means "add 1 to 1", i.e. "2"
  - e.g. in base 2 arithmetic "$1+1$" means "add 1 to 1", i.e. "10"
- A syntactically correct string might be nonsense:
  - e.g. "Colourless green ideas sleep furiously" means ???

Semantics

Up to now we have only been concerned with whether the *syntax* of a string matches some language (i.e. is it a member of the set defining the language).

Often we are more interested in the *semantics*, or meaning, of a string. One approach to applying semantics to a grammar is to give each rule of the grammar some sort of interpretation.

For example, we might define the rule $E \to (EOE)$ with the meaning "apply the operation defined by O to the results of evaluating each of the two E expressions"

Evaluating a Parse Tree

In this week's tutorial materials you will look at some sample code implementing a parse tree for arithmetic expressions. It is incomplete and you will need to finish it.

The code is a simple linked implemention of a tree data structure.

Each class is a type of node in the tree, representing a particular production rule from the grammar.

Evaluating a Parse Tree

The *evaluate* method returns a string representing the evaluation of the subtree rooted at that node. e.g.

- *TerminalNode* represents a terminal symbol, so it's *evaluate* method just returns the terminal symbol itself
- *ValueNode* represents the rules $V \to 0 \mid 1 \mid 2$, so it returns the result of evaluating it's child
  - The *ValueExpressionNode*, *OperationNode* are very similar
- *BinaryExpressionNode* represents the rule $E \to (EOE)$. It evaluates the O child to determine which binary operation to apply to the evaluations of the two child expressions.

Evaluating $(1 + (3 - 2))$

- (refer to earlier slide for tree, and to the tutorial code)

The root of the tree is a *BinaryExpressionNode*, which:

1. Evaluates the child node corresponding to the O:
   - An *OperationNode*, which evaluates it's child *TerminalNode*, which returns $+$
2. Evaluates the left expression node, which is a *ValueExpressionNode*, which:
   - evaluates it's child *ValueNode*, which returns it's child *TerminalNode*, which is 1.
3. Evaluates the right expression node, which is a *BinaryExpressionNode*, which (eventually) also evaluates as 1
4. So we perform addition on 1 and 1, to get 2.

Assignment 2

As *part* of Assignment 2 you will need to apply the semantics for a very simple programming language. The language contains

- print statements
- conditional if and if-else blocks
- arithmetic expressions

The tutorial code is similar to the arithmetic expressions part of this grammar. You may use this code as part of your assignment, or you could start from scratch.

Regular Grammars

A grammar is *regular* if and only if all its rules are in the form

$$A \to xB \text{ or } A \to x$$

where x is a string of zero or more *terminal* symbols

They are also known as *right-linear grammars*

Are these regular?

1. $S \to abS \mid \varepsilon$
2. $S \to aSb \mid \varepsilon$
3. $S \to abS \mid Sab \mid \varepsilon$
4. $S \to abAB$
   $A \to aA \mid \varepsilon$
   $B \to bB \mid \varepsilon$

Regular Grammars

Regular grammars generate exactly the class of regular languages

- Every regular language has a regular grammar describing it
- Every regular language generates a regular language

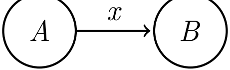
Therefore, regular languages can be equally represented by:

- Finite automata (DFA, NFA)
- Regular expressions
- Regular grammars

Algorithms exist to switch between any of these representations

DFA to regular grammar

1. The set of states Q is the set of variables
2. The alphabet Σ is the set of terminals
3. The initial state gives the start variable
4. The transition function δ and the accept states give the productions



gives $A \to xB$

gives $F \to \varepsilon$

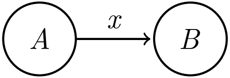
From regular grammars to NFA

1. The set of variables is the set of states Q
2. The set of terminals is the alphabet Σ
3. The start variable gives the initial state
4. The productions give the transition function δ and the accept states

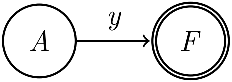
Simple case

Suppose all the strings of terminals in the productions had at most 1 symbol, then we get:

$A \to xB$ gives:



$A \to y$ gives:
(where F is a new state, not in the set of variables)



$A \to \varepsilon$ gives:

Example of simple case

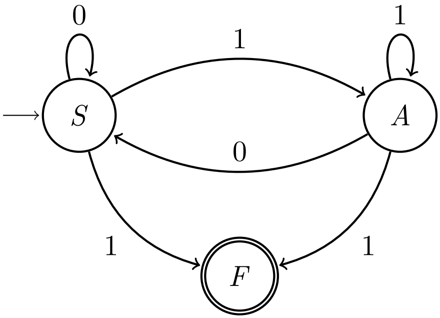
$$S \to 0S \mid 1A \mid 1$$
$$A \to 1A \mid 0S \mid 1$$

We can build the following NFA:

Example of simple case

$$S \to 0S \mid 1A \mid 1$$
$$A \to 1A \mid 0S \mid 1$$

We can build the following NFA:



General case

If a production has more than one terminal, i.e.

$$A \rightarrow x_1 ... x_n B$$

Then transform the grammar so that all the productions have the form $A \rightarrow x$ or $A \rightarrow xB$, where x is a single terminal or ε

Example: $A \rightarrow bcdE$ becomes

$$\begin{aligned} A &\rightarrow bA_1 \\ A_1 &\rightarrow cA_2 \\ A_2 &\rightarrow dE \end{aligned}$$

(where A_1 and A_2 are a new variables.)

Beware!

Consider the language of strings containing the substring abc

This grammar generates that language:

$$\begin{aligned} S &\to TabcT \\ T &\to aT \mid bT \mid cT \mid \varepsilon \end{aligned}$$

Is the grammar regular?

Is the language regular?

Beware!

Consider the language of strings containing the substring abc

This grammar generates that language:

$$S \rightarrow TabcT$$
$$T \rightarrow aT \mid bT \mid cT \mid \varepsilon$$

Is the grammar regular? No

Is the language regular?

Beware!

Consider the language of strings containing the substring abc

This grammar generates that language:

$$S \to TabcT$$
$$T \to aT \mid bT \mid cT \mid \varepsilon$$

Is the grammar regular? No

Is the language regular? Yes!

Beware!

Consider the language of strings containing the substring abc

This grammar generates that language:

$$S \to TabcT$$
$$T \to aT \mid bT \mid cT \mid \varepsilon$$

Is the grammar regular? No

Is the language regular? Yes!

Regular languages can be described by a grammar which is context-free but not regular. If the language is regular, then is a regular grammar which generates the same language.

Many CFL are not regular

For example, we already know that $A = \{0^n 1^n \mid n \geq 1\}$ is not regular.

It is generated by $S \to 0S1 \mid 01$, so it is a CFL

Because it is not regular, it will be impossible to find an equivalent regular grammar

Regular languages $\subset$ Context-free languages

Pumping lemma for context-free languages

Let L be an infinite context-free language over the alphabet Σ. Then there exists an integer $m > 0$ such that for any string $s \in L$ where $|s| \geq m$ there exist strings $u, v, x, y, z \in \Sigma^\star$ such that:

1. $s = uvxyz$
2. $|vy| \geq 1$
3. $|vxy| \leq m$
4. $uv^kxy^kz \in L$ for all $k \geq 0$

If a language does not satisfy this lemma, then it cannot be context-free.

Beyond CFL

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

$$\begin{aligned} S &\to ACaB \\ Ca &\to aaC \\ CB &\to DB \\ CB &\to E \\ aD &\to Da \\ AD &\to AC \\ aE &\to Ea \\ AE &\to \varepsilon \end{aligned}$$

Beyond CFL

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

How can we derive $aaaa$?

$$
\begin{aligned}
S &\to ACaB \\
Ca &\to aaC \\
CB &\to DB \\
CB &\to E \\
aD &\to Da \\
AD &\to AC \\
aE &\to Ea \\
AE &\to \varepsilon
\end{aligned}
$$

Beyond CFL

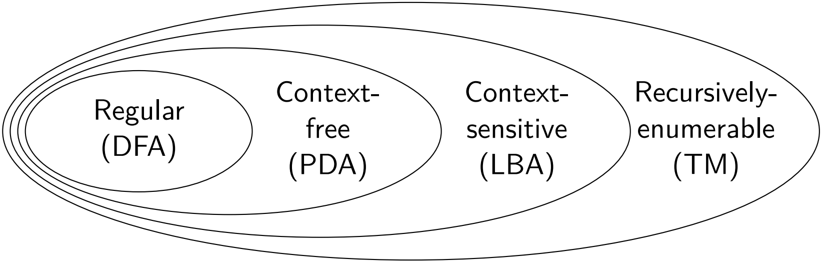
An unrestricted grammar to generate $L = \{a^{2^i} \mid i \geq 1\}$:

How can we derive $aaaa$?

$$
\begin{aligned}
S &\to ACaB \\
Ca &\to aaC \\
CB &\to DB \\
CB &\to E \\
aD &\to Da \\
AD &\to AC \\
aE &\to Ea \\
AE &\to \varepsilon
\end{aligned}
$$

$$
\begin{aligned}
S &\Rightarrow ACaB \Rightarrow AaaCB \\
&\Rightarrow AaaDB \Rightarrow AaDaB \Rightarrow ADaaB \\
&\Rightarrow ACaaB \Rightarrow AaaCaB \Rightarrow AaaaaCB \\
&\Rightarrow AaaaaE \Rightarrow AaaaEa \Rightarrow AaaEaa \\
&\Rightarrow AaEaaa \Rightarrow AEaaaa \\
&\Rightarrow aaaa
\end{aligned}
$$

Chomsky Hierarchy



Unrestricted grammars are powerful enough to describe the set of recursively enumerable languages. We will look at these later in the course, when we study Turing Machines.

Logic: Introductory Example

What do you think of the following argument?

1. If a number is a multiple of $2 \times 3 \times 4$, then it is a multiple of 2 and a multiple of 3 and a multiple of 4
2. Therefore, if a number is a multiple of 2 and a multiple of 3 and a multiple of 4, then it is a multiple of $2 \times 3 \times 4$

Logic: Introductory Example

What do you think of the following argument?

1. If a number is a multiple of $2 \times 3 \times 4$, then it is a multiple of 2 and a multiple of 3 and a multiple of 4
2. Therefore, if a number is a multiple of 2 and a multiple of 3 and a multiple of 4, then it is a multiple of $2 \times 3 \times 4$

Expressing this formally:

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

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
- In CS: specification of requirements, program verification, some databases
- Allows expression of knowledge concisely and precisely, enabling analysis of the argument structure
- Provides a way to *reason* about the consequences of that knowledge rigourously. i.e. How to make a judgements on the validity of the argument
- Focus on validity (correctness) of the argument *form*, rather than it's *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

- P: I play cricket
- 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 is neither stationary nor is it moving at a constant velocity.

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

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

Arugment, Premises, Deduction

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

- If P or Q, then not R
- Therefore, if R then not P and not Q

The *premises* of an arugment are the hypothesis for the argument

- If P or Q, then not R

The last statement is the *conclusion* of the argument, which needs to be *deduced* from the premises

- Therefore, if R then not P and not Q

Propositional Logic

Formalise sentences

- Propositions and connectives
- From english to propositions

Semantics

- 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
- The sun orbits the earth
- $2 + 2 = 4$
- All integers are even

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)

Truth symbols are wff (**true** or **false**)

Atomic propositions are wff ($P, Q, R, ...$)

Complex propositions are built up using connectives:
If P and Q are wff, then (P), $\neg P$, $(P \wedge Q)$, $(P \vee Q)$, $(P \rightarrow Q)$, $(P \leftrightarrow Q)$ 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 meaning of **true** is 1, and **false** if 0, otherwise the meaning of a wff is it's truth table.

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
Implication	$(P \rightarrow Q)$ is false iff P is true and Q is false
Equivalence	$(P \leftrightarrow Q)$ is true iff P has the same truth value as Q

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$	$(P \rightarrow Q)$	$(P \leftrightarrow Q)$
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
Implication	$(P \rightarrow Q)$ is false iff P is true and Q is false
Equivalence	$(P \leftrightarrow Q)$ is true iff P has the same truth value as Q

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$	$(P \rightarrow Q)$	$(P \leftrightarrow Q)$
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
Implication	$(P \rightarrow Q)$ is false iff P is true and Q is false
Equivalence	$(P \leftrightarrow Q)$ is true iff P has the same truth value as Q

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$	$(P \rightarrow Q)$	$(P \leftrightarrow Q)$
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
Implication	$(P \rightarrow Q)$ is false iff P is true and Q is false
Equivalence	$(P \leftrightarrow Q)$ is true iff P has the same truth value as Q

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$	$(P \rightarrow Q)$	$(P \leftrightarrow Q)$
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
Implication	$(P \rightarrow Q)$ is false iff P is true and Q is false
Equivalence	$(P \leftrightarrow Q)$ is true iff P has the same truth value as Q

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$	$(P \rightarrow Q)$	$(P \leftrightarrow Q)$
1	1	0	1	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
Implication	$(P \rightarrow Q)$ is false iff P is true and Q is false
Equivalence	$(P \leftrightarrow Q)$ is true iff P has the same truth value as Q

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

$$X = (((P \wedge Q) \vee \neg Q) \rightarrow ((P \vee Q) \wedge P))$$

P	Q	$(P \wedge Q)$	$\neg Q$	$((P \wedge Q) \vee \neg Q$	$(P \vee Q)$	$(P \vee Q) \wedge P$	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.

Although John is not tall, John has a better chance of winning the next match of tennis, despite Mark's experience.

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

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.

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

Although John is not tall, John has a better chance of winning the next match of tennis, despite Mark's experience.

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

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.

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

Although John is not tall, John has a better chance of winning the next match of tennis, despite Mark's experience.

$$(\neg T \wedge W \wedge E)$$

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

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.

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

Although John is not tall, John has a better chance of winning the next match of tennis, despite Mark's experience.

$$(\neg T \wedge W \wedge E)$$

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

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

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.

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

Although John is not tall, John has a better chance of winning the next match of tennis, despite Mark's experience.

$$(\neg T \wedge W \wedge E)$$

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

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

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.

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

Although John is not tall, John has a better chance of winning the next match of tennis, despite Mark's experience.

$$(\neg T \wedge W \wedge E)$$

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

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

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
$(P \wedge Q)$	P and Q P but Q not only P but Q 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 P unless Q

Possible interpretations in English

$(P \rightarrow Q)$	If P then Q Q if P P only if Q Q when P P is sufficient for Q Q is necessary for P P implies Q P materially implies Q
$(P \leftrightarrow Q)$	P if and only if Q P iff Q P is necessary and sufficient for Q P exactly if Q P is materially equivalent to Q

Be careful with disjunction!

We need to be careful with the definition of disjunction:

- Inclusive "or": $(P \lor Q)$
- Exclusive "or": $(P \lor Q) \land \neg(P \land Q)$

Be careful with disjunction!

We need to be careful with the definition of disjunction:

- Inclusive "or": $(P \vee Q)$
- Exclusive "or": $(P \vee Q) \wedge \neg(P \wedge Q)$

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": $(P \vee Q)$
- Exclusive "or": $(P \vee Q) \wedge \neg(P \wedge Q)$

Examples:

You can go to the airport by taxi or bus	Inclusive
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": $(P \vee Q)$
- Exclusive "or": $(P \vee Q) \wedge \neg(P \wedge Q)$

Examples:

You can go to the airport by taxi or bus	Inclusive
You can choose to save your money or your life	Exclusive
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": $(P \vee Q)$
- Exclusive "or": $(P \vee Q) \wedge \neg(P \wedge Q)$

Examples:

You can go to the airport by taxi or bus	Inclusive
You can choose to save your money or your life	Exclusive
The error is in the program or the sensor data	Exclusive?
The program or the sensor data are erroneous	

Be careful with disjunction!

We need to be careful with the definition of disjunction:

- Inclusive "or": $(P \vee Q)$
- Exclusive "or": $(P \vee Q) \wedge \neg(P \wedge Q)$

Examples:

You can go to the airport by taxi or bus	Inclusive
You can choose to save your money or your life	Exclusive
The error is in the program or the sensor data	Exclusive?
The program or the sensor data are erroneous	Inclusive?

Be careful with implication/equivalence!

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

Be careful with implication/equivalence!

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

"Eating fast-food is equivalent to aiding the destruction of the world's rainforests"

- This looks like an equivalence, but the speaker actually means implication (it's unlikely that they are trying to claim that destroying rainforests implies eating fast food.)

Be careful with implication/equivalence!

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

“Eating fast-food is equivalent to aiding the destruction of the world’s rainforests”

- This looks like an equivalence, but the speaker actually means implication (it’s unlikely that they are trying to claim that destroying rainforests implies eating fast food.)

“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
Max is home if Claire is at the library
Max is home only if Claire is at the library
Max is home if and only if Claire is at the library
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	
Max is home if Claire is at the library	
Max is home only if Claire is at the library	
Max is home if and only if Claire is at the library	
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	$(H \vee L)$
Max is home if Claire is at the library	
Max is home only if Claire is at the library	
Max is home if and only if Claire is at the library	
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	$(H \vee L)$
Max is home if Claire is at the library	$(L \rightarrow H)$
Max is home only if Claire is at the library	
Max is home if and only if Claire is at the library	
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	$(H \vee L)$
Max is home if Claire is at the library	$(L \rightarrow H)$
Max is home only if Claire is at the library	$(H \rightarrow L)$
Max is home if and only if Claire is at the library	
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	$(H \vee L)$
Max is home if Claire is at the library	$(L \rightarrow H)$
Max is home only if Claire is at the library	$(H \rightarrow L)$
Max is home if and only if Claire is at the library	$(H \leftrightarrow L)$
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	$(H \vee L)$
Max is home if Claire is at the library	$(L \rightarrow H)$
Max is home only if Claire is at the library	$(H \rightarrow L)$
Max is home if and only if Claire is at the library	$(H \leftrightarrow L)$
Max is not home nor Claire is at the library	$(\neg H \wedge \neg L)$
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	$(H \vee L)$
Max is home if Claire is at the library	$(L \rightarrow H)$
Max is home only if Claire is at the library	$(H \rightarrow L)$
Max is home if and only if Claire is at the library	$(H \leftrightarrow L)$
Max is not home nor Claire is at the library	$(\neg H \wedge \neg L)$
Max is home although Claire is at the library	$(H \wedge L)$
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	$(H \vee L)$
Max is home if Claire is at the library	$(L \rightarrow H)$
Max is home only if Claire is at the library	$(H \rightarrow L)$
Max is home if and only if Claire is at the library	$(H \leftrightarrow L)$
Max is not home nor Claire is at the library	$(\neg H \wedge \neg L)$
Max is home although Claire is at the library	$(H \wedge L)$
Max is home unless Claire is at the library	$(L \rightarrow \neg H)$

More examples

Max is home and Claire is at the library	$(H \wedge L)$
Max is home or Claire is at the library	$(H \vee L)$
Max is home if Claire is at the library	$(L \rightarrow H)$
Max is home only if Claire is at the library	$(H \rightarrow L)$
Max is home if and only if Claire is at the library	$(H \leftrightarrow L)$
Max is not home nor Claire is at the library	$(\neg H \wedge \neg L)$
Max is home although Claire is at the library	$(H \wedge L)$
Max is home unless Claire is at the library	$(L \rightarrow \neg H)$
	$(H \vee L)$