MATH1005/MATH6005: Discrete Mathematical Models

Adam Piggott

Semester 1, 2021

Section A: The language of mathematics and computer science

Part 1: Logic (continued)

Predicate logic

Predicates

A **predicate** is a sentence containing one or more variables, with the property that, when a value from a specified **domain** is given to each variable, the sentence becomes a statement. The specified domain is the **domain** of the predicate.

Example: Consider the predicate

$$p(x) = \text{“}x \text{ is a bird”},$$

defined over the domain of animals. If $x =$ cockatoo, then $p(x)$ is true. We write $p(\text{cockatoo}) = T$. Further, $p(\text{shark}) = F$ and $p(\text{17@\#})$ is undefined.

More examples

Perhaps the predicate $q(x) =$ "x is allowed to view this file", defined over a list of users, sounds more relevant to computer scientists.

For $r(x, y) =$ "x has been at war with y", with x and y taking values from the set of countries, we have

$$r(\text{Iraq}, \text{USA}) = T$$

and

$$r(\text{Australia}, \text{Indonesia}) = F.$$

Quantifiers

There are two ways to turn a predicate $p(x)$ into a statement:

- specify a value for x; or
- *quantify* x. (“quantify” = “express the ‘quantity’ of”)

The universal quantifier

The **universal quantifier**, $\forall$, is read “for all” (or “for every”, “for each”, “for any” etc.).

The **universal statement** $\forall x\, p(x)$ is read aloud “for all x (in the domain), $p(x)$ is true” or “$p(x)$ is true for all x (in the domain)”.

The existential quantifier

The **existential quantifier**, $\exists$, is read “there exists” (or “for at least one”, etc.)

The **existential statement** $\exists x\, p(x)$ is read aloud “There exists an x (in the domain) such that $p(x)$ is true” or “$p(x)$ is true for at least one x (in the domain)” or “$p(x)$ is true for some x (in the domain)”.

The existential quantifier with uniqueness

The **existential quantifier with uniqueness**, $\exists!$, is read "there exists a unique" (or "for exactly one", etc.)

The **existential statement** $\exists! x\, p(x)$ is read aloud "There exists a unique x (in the domain) such that $p(x)$ is true" or "$p(x)$ is true for exactly one x (in the domain)".

Examples

Q: Let $p(x)$ be the predicate "x is a bird", with x taking values from the domain $\{\text{cockatoo}, \text{parrot}, \text{shark}\}$. For each of the following statements, write out in words a translation of the statements and evaluate it.

1. $\forall x \, p(x)$
2. $\exists x \, p(x)$
3. $\exists ! x \, p(x)$
4. $\exists ! x \, \neg p(x)$

Examples

Recall that $p(x)$ is the predicate "x is a bird", with x taking values from the domain $\{\text{cockatoo}, \text{parrot}, \text{shark}\}$.

1. $\forall x\, p(x)$
   The statement reads: "For all x in the set $\{\text{cockatoo}, \text{parrot}, \text{shark}\}$, x is a bird". This is false because a shark is not a bird ($p(\text{shark}) = F$).

2. $\exists x\, p(x)$
   The statement reads: "There is at least one x in the set $\{\text{cockatoo}, \text{parrot}, \text{shark}\}$ such that x is a bird". This is true because a cockatoo is a bird ($p(\text{cockatoo}) = T$).

Examples

Recall that $p(x)$ is the predicate "x is a bird", with x taking values from the domain $\{\text{cockatoo}, \text{parrot}, \text{shark}\}$.

3. $\exists! x\, p(x)$
   The statement reads: "There is exactly one x in the set $\{\text{cockatoo}, \text{parrot}, \text{shark}\}$ such that x is a bird". This is false because there are two animals in the domain that are birds ($p(\text{cockatoo}) = p(\text{parrot}) = T$).

4. $\exists! x\, \neg p(x)$
   The statement reads: "There is exactly one x in the set $\{\text{cockatoo}, \text{parrot}, \text{shark}\}$ such that x is not a bird". This is true because only one of the animals in the domain (shark) is not a bird.

Another example

Q: Let $q(x)$ be the predicate "x is in Australia", with x taking values from the domain $D = \{\text{Brisbane}, \text{Sydney}, \text{Melbourne}, \text{Adelaide}, \text{Perth}\}$.

Evaluate each of the following statements

1. $\forall x\ q(x)$
2. $\exists x\ q(x)$
3. $\exists! x\ q(x)$
4. $\exists! x\ \neg q(x)$

Examples

Recall that $q(x)$ is the predicate "x is in Australia", with x taking values from the domain $D = \{\text{Brisbane}, \text{Sydney}, \text{Melbourne}, \text{Adelaide}, \text{Perth}\}$.

1. $\forall x\, q(x)$ is true because every city in D is in Australia
2. $\exists x\, q(x)$ is true because $q(\text{Brisbane}) = T$.
3. $\exists! x\, q(x)$ is false because more than one city in D is in Australia
4. $\exists! x\, \neg q(x)$ is false because there are no cities in D that are not in Australia.

Example

With domain the set of users (of some online system), express the statements below symbolically, using the following notation:

$o(x)$: x is online.
$c(x)$: x has changed status.
$u(x)$: x has uploaded pictures.

1. All users are online.
2. No user has changed status.
3. All users who have changed status have uploaded pictures.
4. Some users have changed status.
5. Only one user has not uploaded pictures.

Example

With domain the set of users (of some online system), express the statements below symbolically, using the following notation:

$o(x)$: x is online.
$c(x)$: x has changed status.
$u(x)$: x has uploaded pictures.

1. All users are online. $\forall x \; o(x)$
2. No user has changed status. $\forall x \; \neg c(x)$
3. All users who have changed status have uploaded pictures. $\forall x \; c(x) \rightarrow u(x)$
4. Some users have changed status. $\exists x \; c(x)$
5. Only one user has not uploaded pictures.$\exists ! x \; \neg u(x)$.

Notation: $\Rightarrow$ and $\Leftrightarrow$

Let $p(x)$ and $q(x)$ be predicates and suppose the common domain of x is D.

- The notation $p(x) \Rightarrow q(x)$ is short for

$$\forall x\ p(x) \to q(x)$$

- The notation $p(x) \Leftrightarrow q(x)$ is short for

$$\forall x\ p(x) \leftrightarrow q(x)$$

WARNING: It is not uncommon for mathematicians to use $\to$ and $\Rightarrow$ interchangeably, as if they mean the same thing.

Order of precedence and quantification

This is perhaps best explained by example: the expression

$$\forall x\, p(x) \to q(x)$$

means

$$\forall x\, \big(p(x) \to q(x)\big).$$

Example

What is the negation of “all users are online”?
Answer: Not all users are online, *i.e.* at least one user is offline.

Symbolically: $\neg(\forall x\, p(x)) \equiv \exists x\, \neg p(x)$.

What is the negation of “some users have changed status”?
Answer: No user has changed status, *i.e.* all users have not changed status.

Symbolically: $\neg(\exists x\, q(x)) \equiv \forall x\, \neg q(x)$.

Example

Here is a more complicated example from mathematical analysis. The variables x and y take values from the set of real numbers; the variable ϵ and δ take values from the set of positive real numbers.

$$
\begin{aligned}
&\neg\Big(\forall\varepsilon\,\exists\delta\,\forall x\,\forall y \quad |x-y|<\delta \implies |x^2-y^2|<\varepsilon\Big)\\
&\equiv \exists\varepsilon\,\forall\delta\,\exists x\,\exists y\neg\Big(|x-y|<\delta \implies |x^2-y^2|<\varepsilon\Big)\\
&\equiv \exists\varepsilon\,\forall\delta\,\exists x\,\exists y \quad |x-y|<\delta \;\wedge\; |x^2-y^2|\geq\varepsilon
\end{aligned}
$$

The order of quantification matters

$$(\forall x\, \exists y\, p(x,y)) \not\equiv (\exists y\, \forall x\, p(x,y))$$
$$(\exists x\, \forall y\, p(x,y)) \not\equiv (\forall y\, \exists x\, p(x,y))$$

Example:

Domains: a set of people and a set of countries.
$p(x,y) =$ "x is a tall inhabitant of y"

$\forall y\, \exists x\, p(x,y)$ says that each country has at least one tall inhabitant.

$\exists x\, \forall y\, p(x,y)$ says that there is a tall individual who lives in every country.

These two statements are not equivalent!

Logic circuits

A question

Do we have all of the logical connectives we need?

That is, suppose I present you with a truth table but I do not label the right-hand column with a compound statement. Can you construct a compound statement for which the given truth table is correct?

Said another way: Is every compound statement logically equivalent to a compound statement made using only AND, OR and NOT?

An example

Q: Find a compound statement to replace ?

p	q	r	?
T	T	T	F
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

In the truth table table, ? can be replaced by:

An example

Q: Find a compound statement to replace ?

p	q	r	$?$
T	T	T	F
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

In the truth table table, ? can be replaced by:

$$(p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge r)$$

An example

A compound statement like the one we just wrote is said to be **disjunctive normal form**.

Following the plan used in the previous example, we see that every compound statement is logically equivalent to a compound statement made using only parentheses and the logical connectives $\wedge, \vee, \neg$.

We say that the set $\{\wedge, \vee, \neg\}$ is **functionally complete**.

CLAIM: The set $\{\wedge, \neg\}$ is functionally complete.

Q: How can we show this?

An example

A compound statement like the one we just wrote is said to be **disjunctive normal form**.

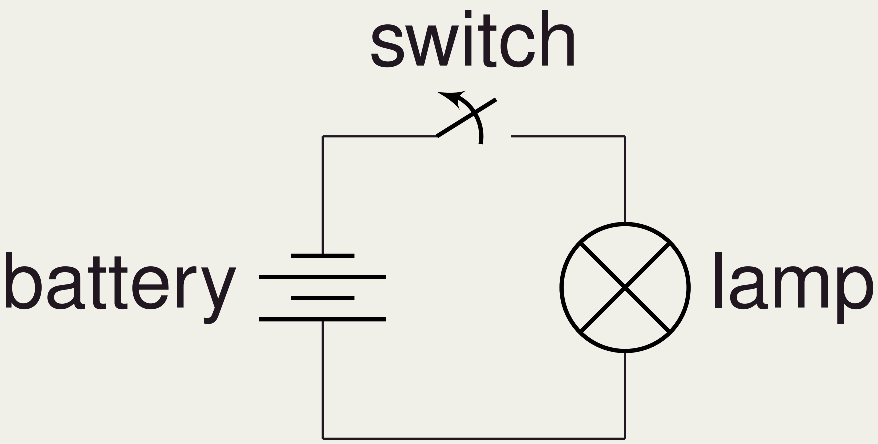
Following the plan used in the previous example, we see that every compound statement is logically equivalent to a compound statement made using only parentheses and the logical connectives $\wedge, \vee, \neg$.

We say that the set $\{\wedge, \vee, \neg\}$ is **functionally complete**.

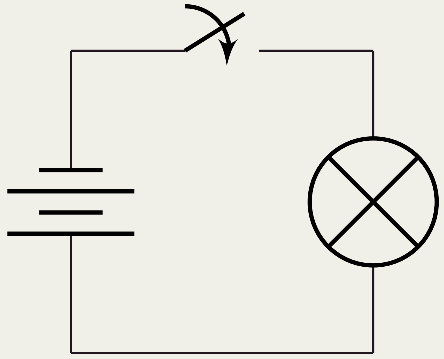
CLAIM: The set $\{\wedge, \neg\}$ is functionally complete.

Q: How can we show this?
A: Show that $p \vee q \equiv \neg(\neg p \wedge \neg q)$

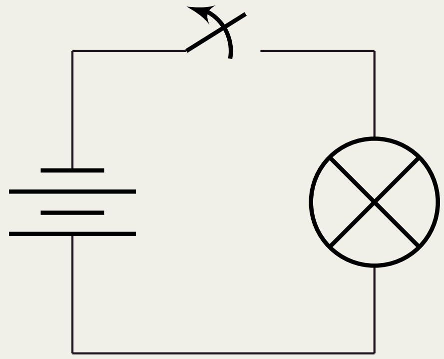
Here is a simple circuit:



It has two possible states:



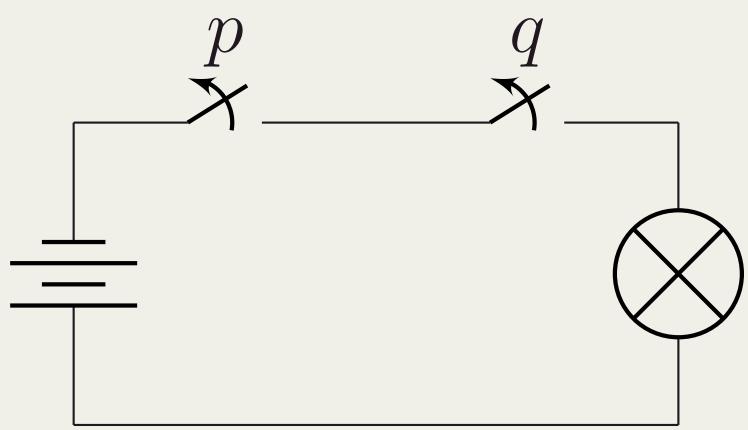
Light ON



Light OFF

The states correspond to the values **True** and **False** of the statement “the light is ON”.

Circuits: AND

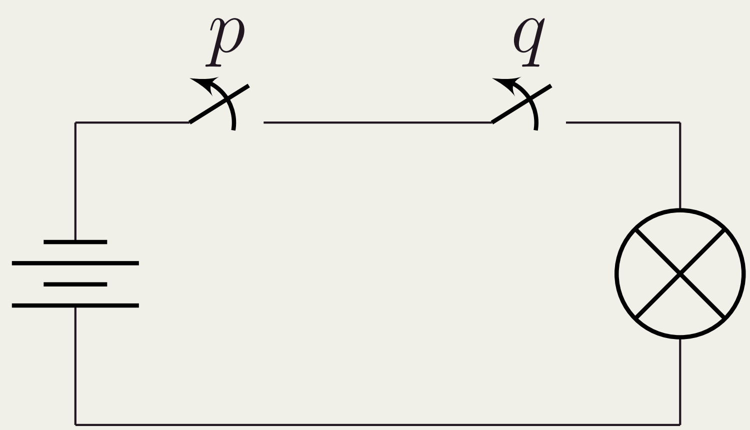


The behaviour of this circuit can be represented by a truth table (which coincides with the truth table for AND):

Switch p is ON	Switch q is ON	Light is ON
T	T	T
T	F	F
F	T	F
F	F	F

The AND gate

The circuit



is called an AND gate.

It takes two inputs:

- the state of switch p
  (denoted by 1 for ON and 0 for OFF)
- the state of switch q

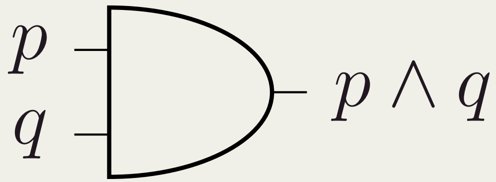
and produces an output:

- the state of the light bulb.

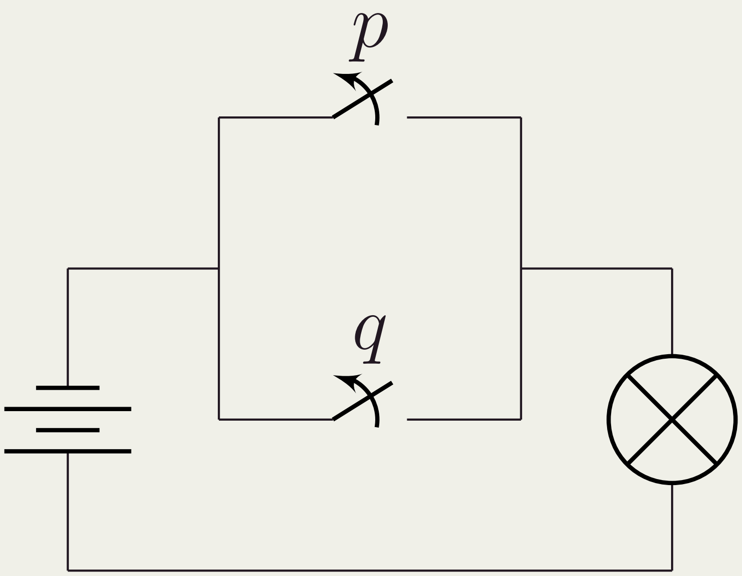
The AND gate

The AND gate is represented by the following input-output table and symbol:

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

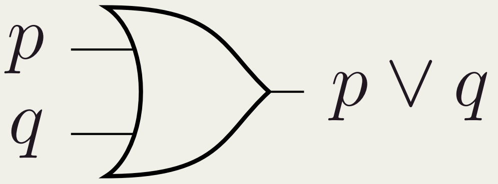


The OR gate

The circuit  gives an OR gate.

It is represented by the following table and symbol:

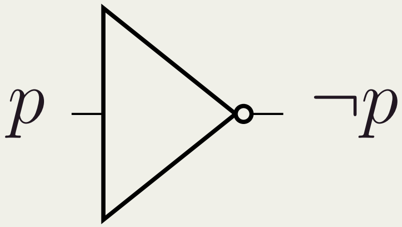
p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0



The NOT gate.

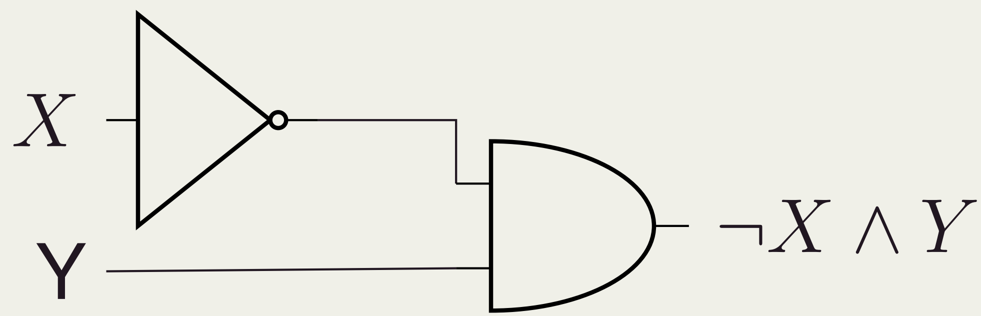
The NOT gate has the following table and symbol:

p	$\neg p$
1	0
0	1



Combining gates

Gates can be combined to create a circuit corresponding to a given compound statement.
Example:

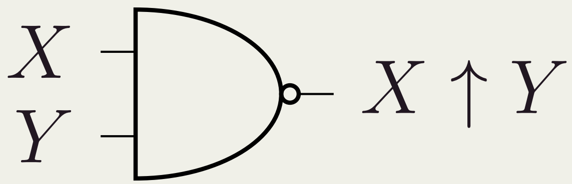


NAND and the NAND Gate

The logical connective **NAND** is a shorthand for “NOT AND”.
p NAND q is denoted by $p \uparrow q$ (sometimes $p|q$ is used instead of $p \uparrow q$). So $p \uparrow q \equiv \neg(p \wedge q)$.

A corresponding **NAND gate** is defined as follows:

X	Y	$X \vert Y$
1	1	0
1	0	1
0	1	1
0	0	1



The functional completeness of NAND

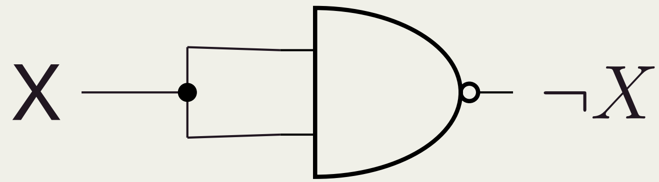
CLAIM: Every gate is equivalent to one that can be constructed by combining NAND gates alone.

How can we establish that this is true?

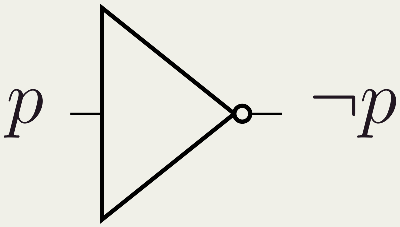
NOT from NAND

Example:

$$\neg X \equiv X \uparrow X$$

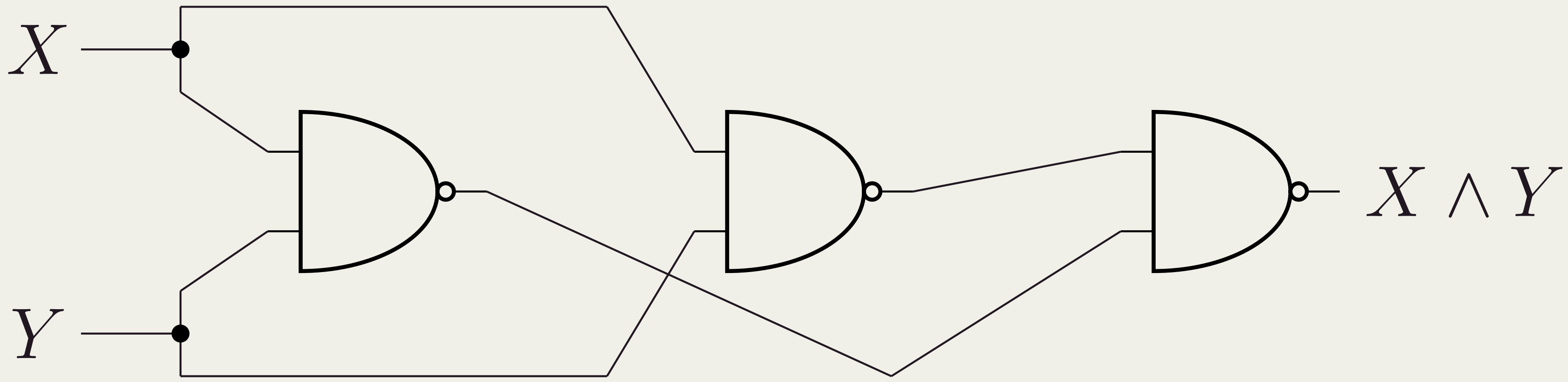


$\equiv$

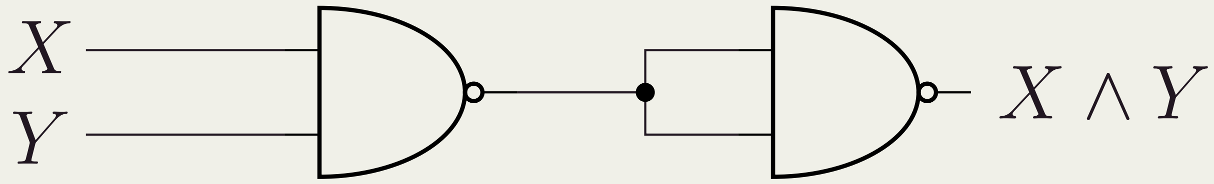


AND from NAND

$$X \wedge Y \equiv \neg(X \uparrow Y) \equiv (X \uparrow Y) \uparrow (X \uparrow Y).$$

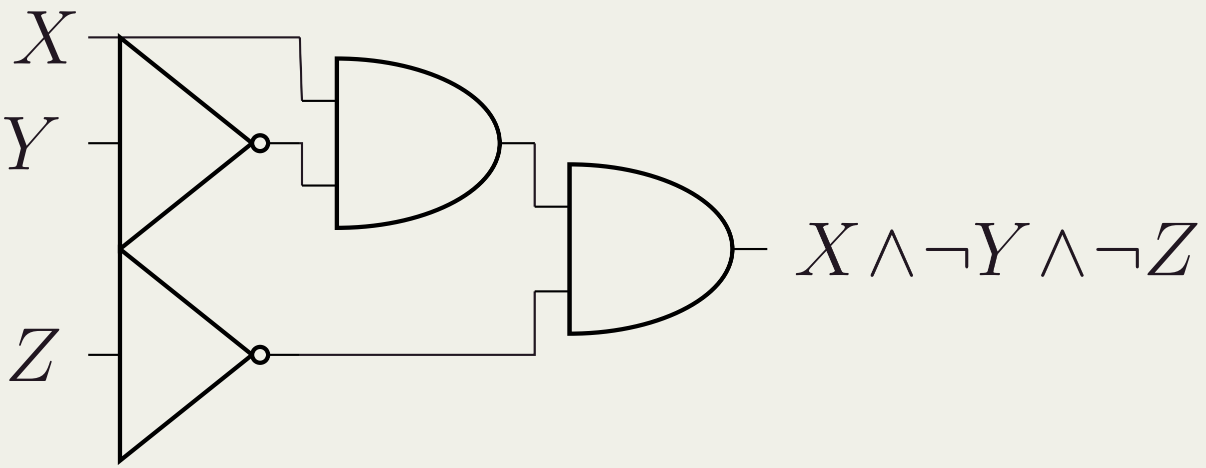


is equivalent to an AND gate.
The circuit can be simplified to use one less NAND gate:



Recogniser Circuits

Consider the circuit and corresponding table below:



X	Y	Z	$X \wedge \neg Y \wedge \neg Z$
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

We say that the circuit recognises the input combination $(X, Y, Z) = (1, 0, 0)$ because that's the only input combination that generates an output of 1.

Similarly a circuit for $\neg X \wedge Y \wedge Z$ would recognise the input combination $(X, Y, Z) = (0, 1, 1)$.

Recogniser circuits

Definition: A circuit that outputs 1 for only one input combination is called a recogniser for that input combination.

Q: Design a circuit to output 1 only for

$$(X, Y, Z) = (1, 1, 1) \;\&\; (1, 0, 0)$$

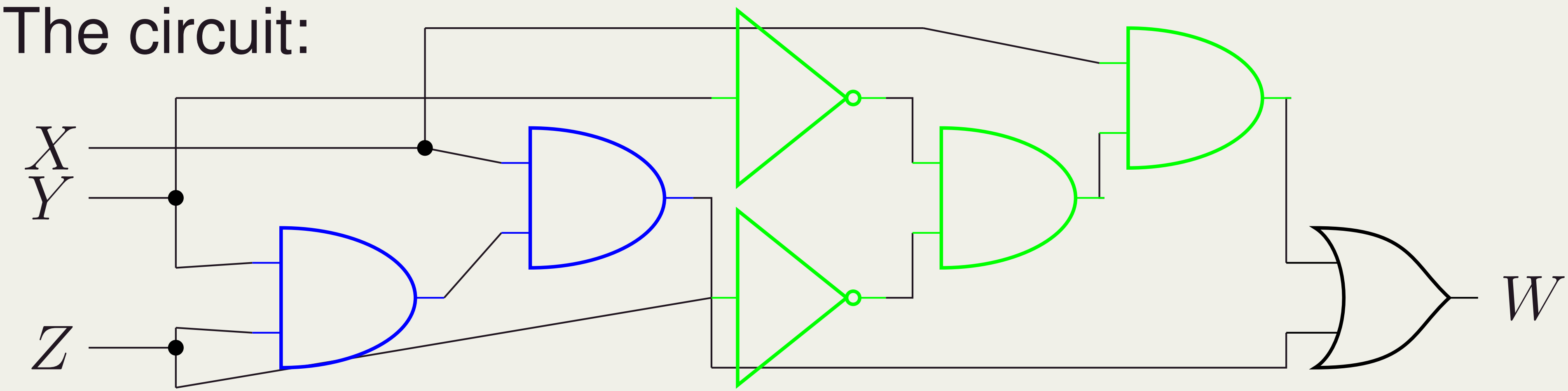
Method: Apply a method like the construction of a compound statement in disjunctive normal form. For outputs equal to 1 express inputs with AND.
Join the resulting expressions with OR.

Example: From input-output tables to circuits

The table:

X	Y	Z	output	$X \wedge Y \wedge Z$	$X \wedge \neg Y \wedge \neg Z$	$W = (X \wedge Y \wedge Z) \vee (X \wedge \neg Y \wedge \neg Z)$
1	1	1	1	1	0	1
1	1	0	0	0	0	0
1	0	1	0	0	0	0
1	0	0	1	0	1	1
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0

The circuit:



Thus ends Section A1