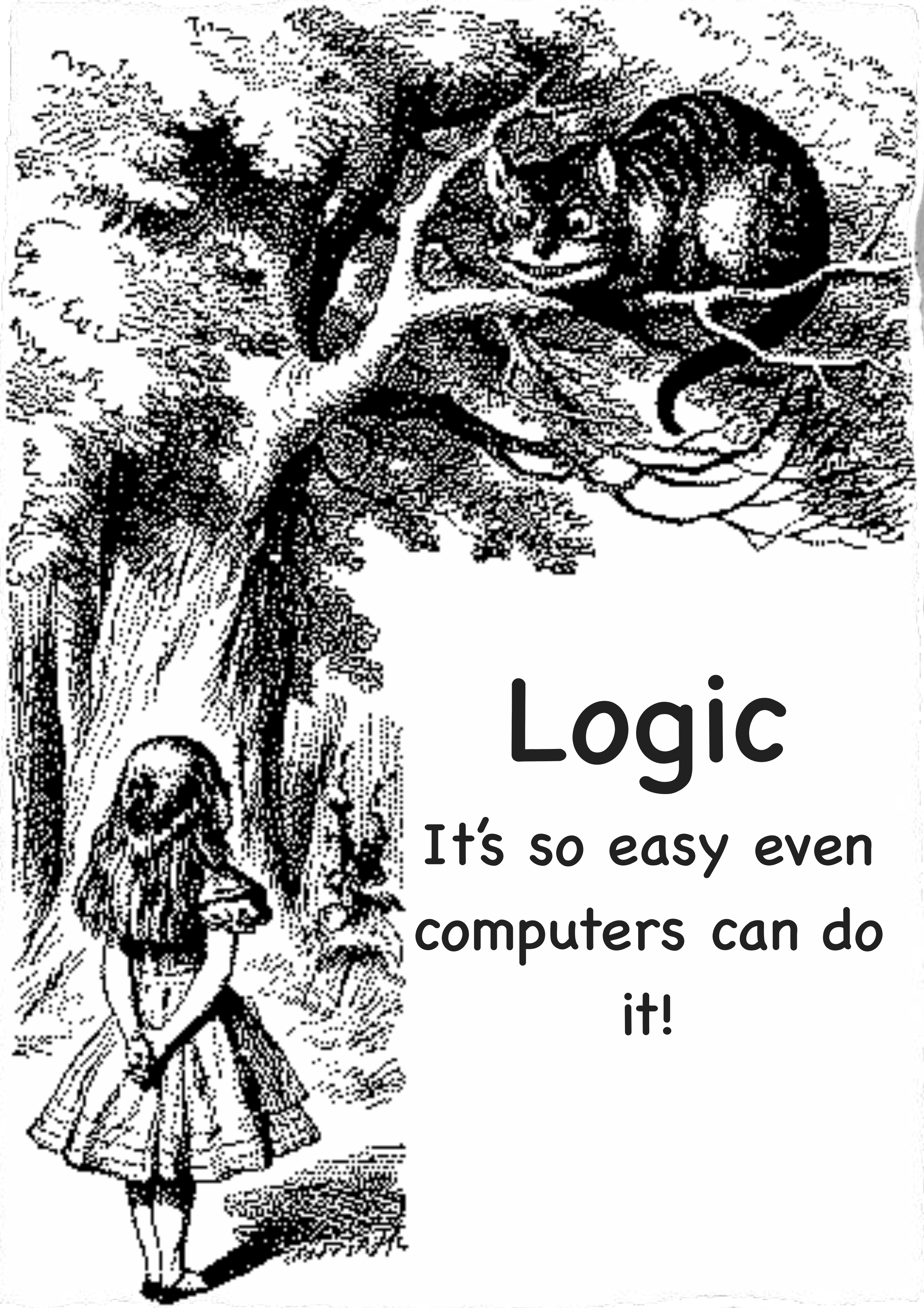
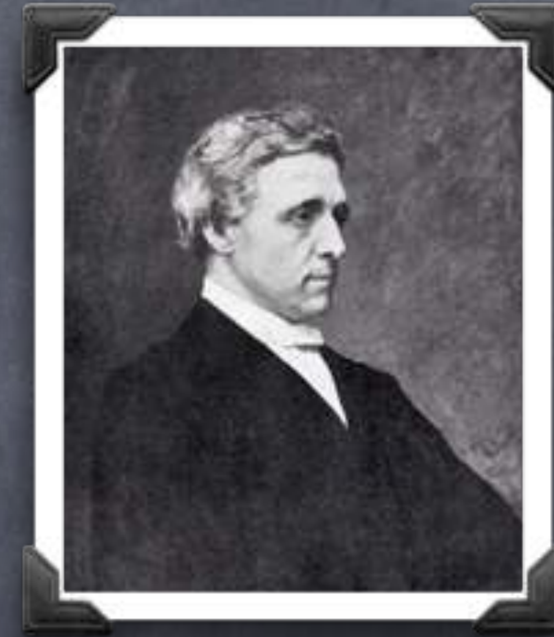


Logic

It's so easy even computers can do it!



Charles L Dodgson
1832 - 1898

Propositions,
Predicates,
Operators,
Formulas

Logic in action

Expert Systems

- From a repository consisting of "facts" derive answers to questions posed on the fly
- To automate decision making
- e.g., Prolog: a programming language that can be used to implement such a system

```
mother_child(trude, sally).

father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).

sibling(X, Y) ← parent_child(Z, X)
                ∧ parent_child(Z, Y).

parent_child(X, Y) ← father_child(X, Y).
parent_child(X, Y) ← mother_child(X, Y).
```

```
?- sibling(sally, erica).
Yes
```

The Pointless Game

- Alice and Bob sit down to play a new board game, where they take turns to make "moves" that they can choose (no dice/randomness)

- The rules of the game guarantee that
  - The game can't go on for ever
  - There are no ties — Alice or Bob will win when the game terminates
- Alice and Bob (smart as they are) decide that there is no point in playing the game, because they already know who is going to win it!

But how?

Propositions

- Goal: reasoning about whether statements are true or false
  - These statements are called **propositions**
- Propositions refer to things in a "**domain** of discourse" (e.g., characters in Alice in Wonderland)
- A proposition could simply refer to a property of an element in the domain (e.g., Alice doesn't have wings)
  - These properties are formalised as **predicates**

Alice
Jabberwock
Flamingo

Predicates

- Predicate is a function that assigns a value of TRUE or FALSE to each element in the domain of discourse
  - If you apply a predicate to an element you get a proposition
    - A proposition will have truth value True or False
  - More complex propositions can be built from such simple propositions

e.g.: Pink(Flamingo)

	Winged?	Flies?	Pink?
Alice	FALSE	FALSE	FALSE
Jabberwock	TRUE	TRUE	FALSE
Flamingo	TRUE	TRUE	TRUE

Propositional Calculus

cal·cu·lus /ˈkalkyələs/

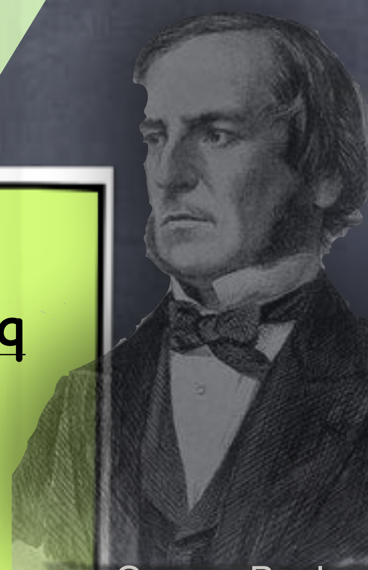
1. The branch of mathematics that deals with the finding and properties ... derivatives and integrals of functions, by methods originally ...
2. A particular method or system of calculation or rea...

Synonyms: stone - calculation - reckoning - comput...

Unary operator

Binary operators

not p	p **or** q	p **and** q	**if** p **then** q
Symbol: ¬p	Symbol: p ∨ q	Symbol: p ∧ q	Symbol: p → q
Negates the truth value.	True if and only if at least one of p and q is true	True if and only if both of p and q are true	(¬p) ∨ (p∧q) Same as ¬p ∨ q
e.g.: ¬Flies(Alice) has value True	e.g.: ¬Flies(Alice) ∨ Pink(Jab'wock) has value True	e.g.: ¬Flies(Alice) ∧ Pink(Jab'wock) has value False	e.g.: Flies(Alice) → Pink(Jab'wock) has value True



George Boole
1815 - 1864

∨	T	F
T	T	T
F	T	F

∧	T	F
T	T	F
F	F	F

→	T	F
T	T	F
F	T	T

Operator Gallery

(T)	T	F
T	T	T
F	T	T

(F)	T	F
T	F	F
F	F	F

(1)	T	F
T	T	T
F	F	F

(2)	T	F
T	T	F
F	T	F

¬(1)	T	F
T	F	F
F	T	T

¬(2)	T	F
T	F	T
F	F	T

XOR

≠

$\oplus$	T	F
T	F	T
F	T	F

IFF

=

$\leftrightarrow$	T	F
T	T	F
F	F	T

OR

$\vee$	T	F
T	T	T
F	T	F

NAND

$\uparrow$	T	F
T	F	T
F	T	T

NOR

$\downarrow$	T	F
T	F	F
F	F	T

AND

$\wedge$	T	F
T	T	F
F	F	F

IMPLIES

$\longrightarrow$	T	F
T	T	F
F	T	T

IMPLIED-BY

$\longleftarrow$	T	F
T	T	T
F	F	T

$\not\rightarrow$	T	F
T	F	T
F	F	F

$\not\leftarrow$	T	F
T	F	F
F	T	F

IMPLIES

$\rightarrow$	T	F
T	T	F
F	T	T

Important: Not a causal relation!

p implies q.

whenever p holds, q holds

if p then q.

q if p.

either not p or (p and q).

p only if q.

if not q then not p.

not p if not q.

Contrapositive

Try an example:

p: you're in the kitchen

q: you're in the house

IMPLIED-BY

$\leftarrow$	T	F
T	T	T
F	F	T

p is implied by q.

q implies p.

p if q.

IFF

=

$\leftrightarrow$	T	F
T	T	F
F	F	T

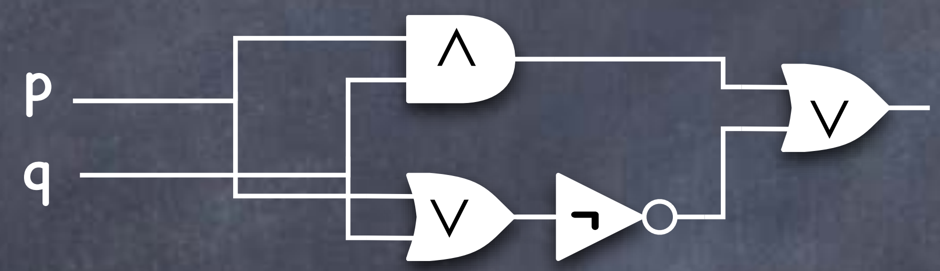
p if and only if q.

p iff q.

p if q and not p if not q.

Formulas

- A recipe for creating a new proposition from given propositions, using operators
  - e.g. $f(p,q) \triangleq (p \wedge q) \vee \neg(p \vee q)$



p	q	f
F	F	T
F	T	F
T	F	F
T	T	T

- Can also use "logic circuits" instead of formulas
- Different formulas can be **equivalent** to each other
  - e.g., $g(p,q) \triangleq \neg(p \oplus q)$. Then $f \equiv g$.
- A formula on two variables is <u>equivalent</u> to a binary operator

Another Example

- $g(p,q,r) \triangleq (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge q \wedge r)$
  - "Majority operator"
- $h(p,q,r) \triangleq (p \wedge (q \vee r)) \vee (q \wedge r)$
- $g \equiv h$

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

p	q	r	g	h
F	F	F	F	F
F	F	T	F	F
F	T	F	F	F
T	F	F	F	F
F	T	T	T	T
T	F	T	T	T
T	T	F	T	T
T	T	T	T	T

More Equivalences [Exercise]

- Conjunction and disjunction with T and F

$T \wedge q \equiv q$	$F \vee q \equiv q$
$F \wedge q \equiv F$	$T \vee q \equiv T$

- Implication involving T and F

$T \rightarrow q \equiv q$	$F \rightarrow q \equiv T$
$q \rightarrow F \equiv \neg q$	$q \rightarrow T \equiv T$

- Implication involving negation

$q \rightarrow \neg q \equiv \neg q$	$\neg q \rightarrow q \equiv q$

- **Contrapositive**

$$p \rightarrow q \equiv (\neg q) \rightarrow (\neg p)$$

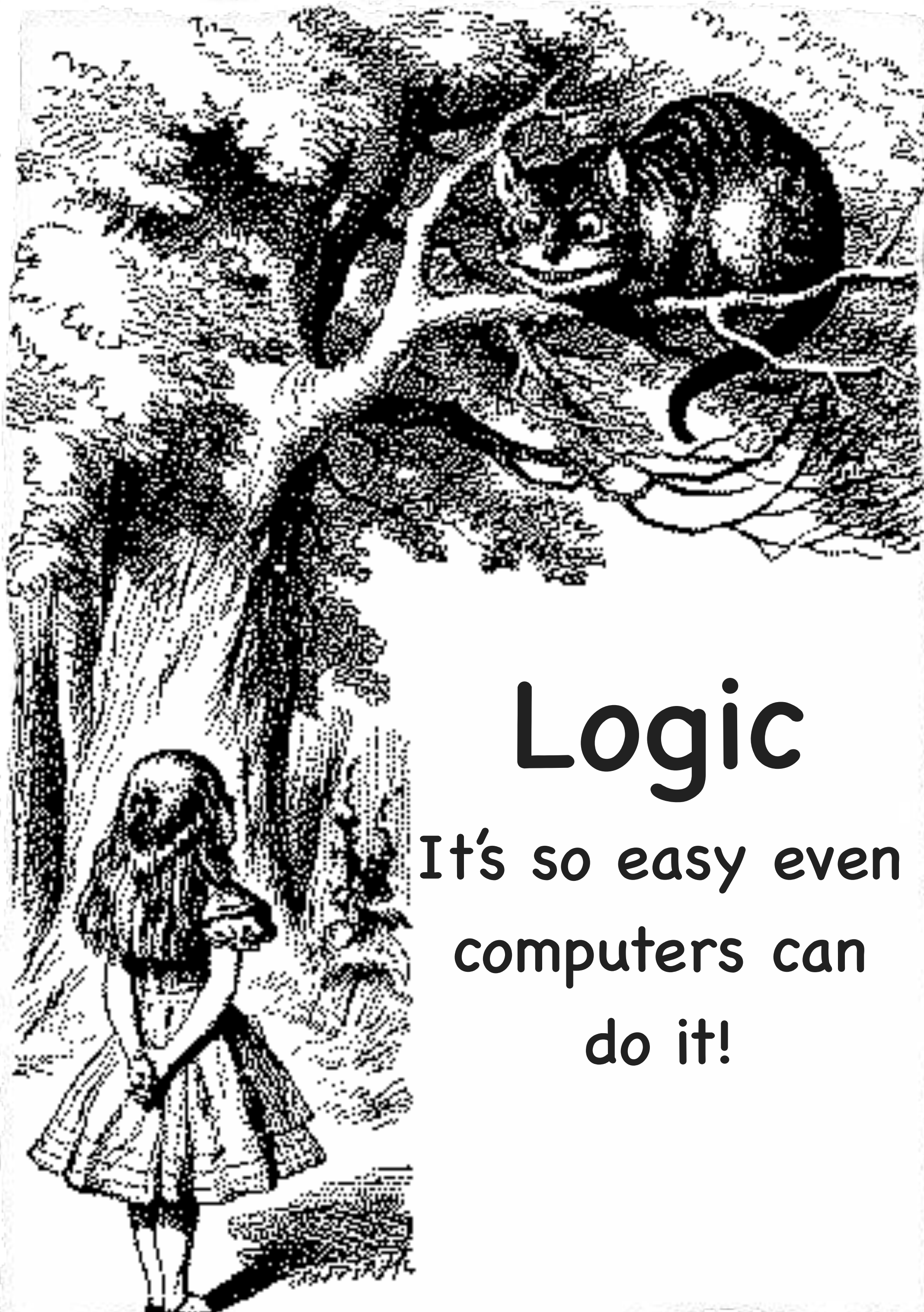
- **Distributive Property**

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

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

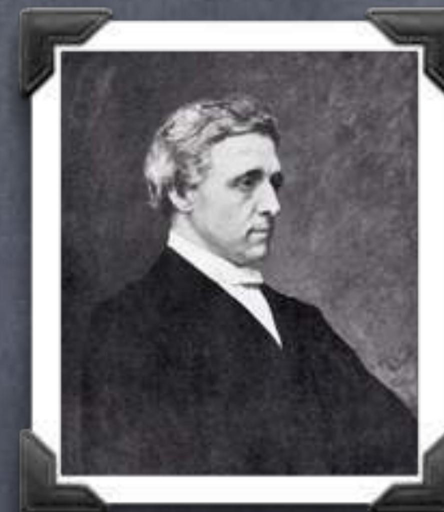
- **De Morgan's Law**

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



Logic

It's so easy even computers can do it!



Charles L Dodgson
1832 - 1898

Quantifiers

Predicates & Propositions

x	Winged(x)	Flies(x)	Pink(x)
Alice	FALSE	FALSE	FALSE
Jabberwock	TRUE	TRUE	FALSE
Flamingo	TRUE	TRUE	TRUE

- A predicate is a column in this table
- A proposition like Winged(Alice) refers to a single cell. Can build more complex propositions using propositional calculus (formulas)
- Next: Propositions involving quantifiers.

Quantified Propositions

(First-Order) Predicate Calculus

x	Winged(x)	Flies(x)	Pink(x)
Alice	FALSE	FALSE	FALSE
Jabberwock	TRUE	TRUE	FALSE
Flamingo	TRUE	TRUE	TRUE

- All characters in AIW are winged. (False!) $\boxed{\forall x_{\in AIW}\ \text{Winged}(x)}$
  - For every character x in AIW, Winged(x) holds
- Some character in AIW is winged. (True) $\boxed{\exists x\ \text{Winged}(x)}$
  - There exists a character x in AIW, such that Winged(x) holds

Quantified Propositions

(First-Order) Predicate Calculus

x	Winged(x)	Flies(x)	Pink(x)
Alice	FALSE	FALSE	FALSE
Jabberwock	TRUE	TRUE	FALSE
Flamingo	TRUE	TRUE	TRUE

- Quantifiers: To what "extent" does a predicate evaluate to TRUE in the domain of discourse
  - Universal quantifier, ∀
  - Existential quantifier, ∃

$\boxed{\forall x \text{ Winged}(x)}$

$\boxed{\exists x \text{ Winged}(x)}$

Quantified Propositions

(First-Order) Predicate Calculus

x	Winged(x)	Flies(x)	Pink(x)
Alice	FALSE	FALSE	FALSE
Jabberwock	TRUE	TRUE	FALSE
Flamingo	TRUE	TRUE	TRUE

- Could write $\forall x$ Winged(x) as:
  Winged(Alice) $\wedge$ Winged(J'wock) $\wedge$ Winged(Flamingo)
- And $\exists x$ Winged(x) as:
  Winged(Alice) $\vee$ Winged(J'wock) $\vee$ Winged(Flamingo)
  - But need to list the entire domain (works only if finite)

Examples

x	Winged(x)	Flies(x)	Pink(x)	Pink(x)→ Flies(x)
Alice	FALSE	FALSE	FALSE	TRUE
Jabberwock	TRUE	TRUE	FALSE	TRUE
Flamingo	TRUE	TRUE	TRUE	TRUE

- $\forall x \ \ \mathrm{Winged}(x) \leftrightarrow \mathrm{Flies}(x)$ is True
- $\exists x \ \ \mathrm{Winged}(x) \rightarrow \neg\mathrm{Flies}(x)$ is True
- $\forall x \ \ \mathrm{Pink}(x) \rightarrow \mathrm{Flies}(x)$ is True

Quantified Propositions

(First-Order) Predicate Calculus

x	Winged(x)	Flies(x)	Pink(x)	¬Winged(x)
Alice	FALSE	FALSE	FALSE	TRUE
Jabberwock	TRUE	TRUE	FALSE	FALSE
Flamingo	TRUE	TRUE	TRUE	FALSE

- $\forall x$ Winged(x) is False
  - Not everyone is winged
  - Same as saying, there is someone who is not winged
    - i.e., $\exists x \neg$Winged(x) is True
- $\boxed{\neg (\forall x\ W(x)) \equiv \exists x\ \neg W(x)}$

$$\neg(W(a) \wedge W(j) \wedge W(f))$$
$$\equiv$$
$$\neg W(a) \vee \neg W(j) \vee \neg W(f)$$

Predicates, again

- A predicate can be defined over any number of elements from the domain
  - e.g., Likes(x,y): "x likes y"

x,y	Likes(x,y)
Alice, Alice	TRUE
Alice, Jabberwock	FALSE
Alice, Flamingo	TRUE
Jabberwock, Alice	FALSE
Jabberwock, Jabberwock	TRUE
Jabberwock, Flamingo	FALSE
Flamingo, Alice	FALSE
Flamingo, Jabberwock	FALSE
Flamingo, Flamingo	TRUE

Two quantifiers

x,y	Likes(x,y)
Alice, Alice	TRUE
Alice, Jabberwock	FALSE
Alice, Flamingo	TRUE
Jabberwock, Alice	FALSE
Jabberwock, Jabberwock	TRUE
Jabberwock, Flamingo	FALSE
Flamingo, Alice	FALSE
Flamingo, Jabberwock	FALSE
Flamingo, Flamingo	TRUE

- And we can quantify all the variables of a predicate
- e.g. $\forall x,y$ Likes(x,y)
    - Everyone likes everyone
    - False!

Two quantifiers

x,y	Likes(x,y)
Alice, Alice	TRUE
Alice, Jabberwock	FALSE
Alice, Flamingo	TRUE
Jabberwock, Alice	FALSE
Jabberwock, Jabberwock	TRUE
Jabberwock, Flamingo	FALSE
Flamingo, Alice	FALSE
Flamingo, Jabberwock	FALSE
Flamingo, Flamingo	TRUE

- $\forall x\ \exists y\ Likes(x,y)$
  - Everyone likes someone (True)
- $\exists y\ \forall x\ Likes(x,y)$
  - Someone is liked by everyone (False)

Order of quantifiers is important!

Two quantifiers

x	y	Likes(x,y)	∃y Likes(x,y) i.e., LikesSomeone(x)
Alice	Alice	TRUE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	
Jabberwock	Alice	FALSE	TRUE
	Jabberwock	TRUE	
	Flamingo	FALSE	
Flamingo	Alice	FALSE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	

- ∀x ∃y Likes(x,y)
  - Everyone likes someone
  - ∀x LikesSomeone(x)
  - True

Moving the Quantifiers

- $\forall x\ \forall y\ P(x,y) \equiv \forall y\ \forall x\ P(x,y)$ for all pairs (x,y), P(x,y) holds
- $\exists x\ \exists y\ P(x,y) \equiv \exists y\ \exists x\ P(x,y)$ for some pair (x,y), P(x,y) holds
- Below R is a proposition not involving x

$$\forall x\ P(x) \lor R \;\equiv\; (\forall x\ P(x)\ ) \lor R$$

- **Scope** of x extends to the end: $\forall x\ (\underline{P(x) \lor R})$
- i.e., if domain is $\{a_1,\ldots,a_N\}$ $(\underline{P(a_1)\lor R}) \land \ldots \land (\underline{P(a_N)\lor R})$

- R evaluates to True or False (indep of x)
- When R is True, both equivalent (to True)
- Also, when R is False, both equivalent
- Hence both equivalent

Moving the Quantifiers

- $\forall x\ \forall y\ P(x,y) \equiv \forall y\ \forall x\ P(x,y)$ for all pairs (x,y), P(x,y) holds
- $\exists x\ \exists y\ P(x,y) \equiv \exists y\ \exists x\ P(x,y)$ for some pair (x,y), P(x,y) holds
- Below R is a proposition not involving x

  $\forall x\ P(x) \vee R \equiv (\forall x\ P(x)) \vee R$ $\quad$ $\exists x\ P(x) \vee R \equiv (\exists x\ P(x)) \vee R$

  $\forall x\ P(x) \wedge R \equiv (\forall x\ P(x)) \wedge R$ $\quad$ $\exists x\ P(x) \wedge R \equiv (\exists x\ P(x)) \wedge R$
- $\forall x\ R \rightarrow P(x) \equiv R \rightarrow (\forall x\ P(x))$ $\quad$ $\exists x\ R \rightarrow P(x) \equiv R \rightarrow (\exists x\ P(x))$
- $\forall x\ \underline{P(x) \rightarrow R} \equiv (\exists x\ P(x)) \rightarrow R$ $\quad$ $\exists x\ \underline{P(x) \rightarrow R} \equiv (\forall x\ P(x)) \rightarrow R$

$\forall x\ \underline{\neg P(x) \vee R} \equiv (\forall x\ \underline{\neg P(x)}) \vee R \equiv \neg\ (\exists x\ \underline{P(x)}) \vee R$

Moving the Quantifiers

- $\forall x\ \forall y\ P(x,y) \equiv \forall y\ \forall x\ P(x,y)$ for all pairs (x,y), P(x,y) holds
- $\exists x\ \exists y\ P(x,y) \equiv \exists y\ \exists x\ P(x,y)$ for some pair (x,y), P(x,y) holds
- Below R is a proposition not involving x

  $\forall x\ P(x) \lor R \equiv (\forall x\ P(x)) \lor R$ $\exists x\ P(x) \lor R \equiv (\exists x\ P(x)) \lor R$

  $\forall x\ P(x) \land R \equiv (\forall x\ P(x)) \land R$ $\exists x\ P(x) \land R \equiv (\exists x\ P(x)) \land R$
- $\forall x\ R \rightarrow P(x) \equiv R \rightarrow (\forall x\ P(x))$ $\exists x\ R \rightarrow P(x) \equiv R \rightarrow (\exists x\ P(x))$
- $\forall x\ \underline{P(x) \rightarrow R} \equiv (\exists x\ P(x)) \rightarrow R$ $\exists x\ \underline{P(x) \rightarrow R} \equiv (\forall x\ P(x)) \rightarrow R$
- $(\forall x\ P(x)) \land (\forall x\ Q(x)) \equiv \forall x\ (P(x) \land Q(x))$

  $(\exists x\ P(x)) \lor (\exists x\ Q(x)) \equiv \exists x\ (P(x) \lor Q(x))$
- $(\forall x\ P(x)) \lor (\forall x\ Q(x)) \equiv (\forall x\ P(x)) \lor (\forall y\ Q(y))$

  $\equiv \forall x\ (\ P(x) \lor (\forall y\ Q(y))\ )$

  $\equiv \forall x\ (\ \forall y\ (P(x) \lor Q(y))\ )$

  $\equiv \forall x\ \forall y\ (P(x) \lor Q(y))$

Moving the Quantifiers

- $\forall x\ \forall y\ P(x,y) \equiv \forall y\ \forall x\ P(x,y)$ for all pairs (x,y), P(x,y) holds
- $\exists x\ \exists y\ P(x,y) \equiv \exists y\ \exists x\ P(x,y)$ for some pair (x,y), P(x,y) holds
- Below R is a proposition not involving x

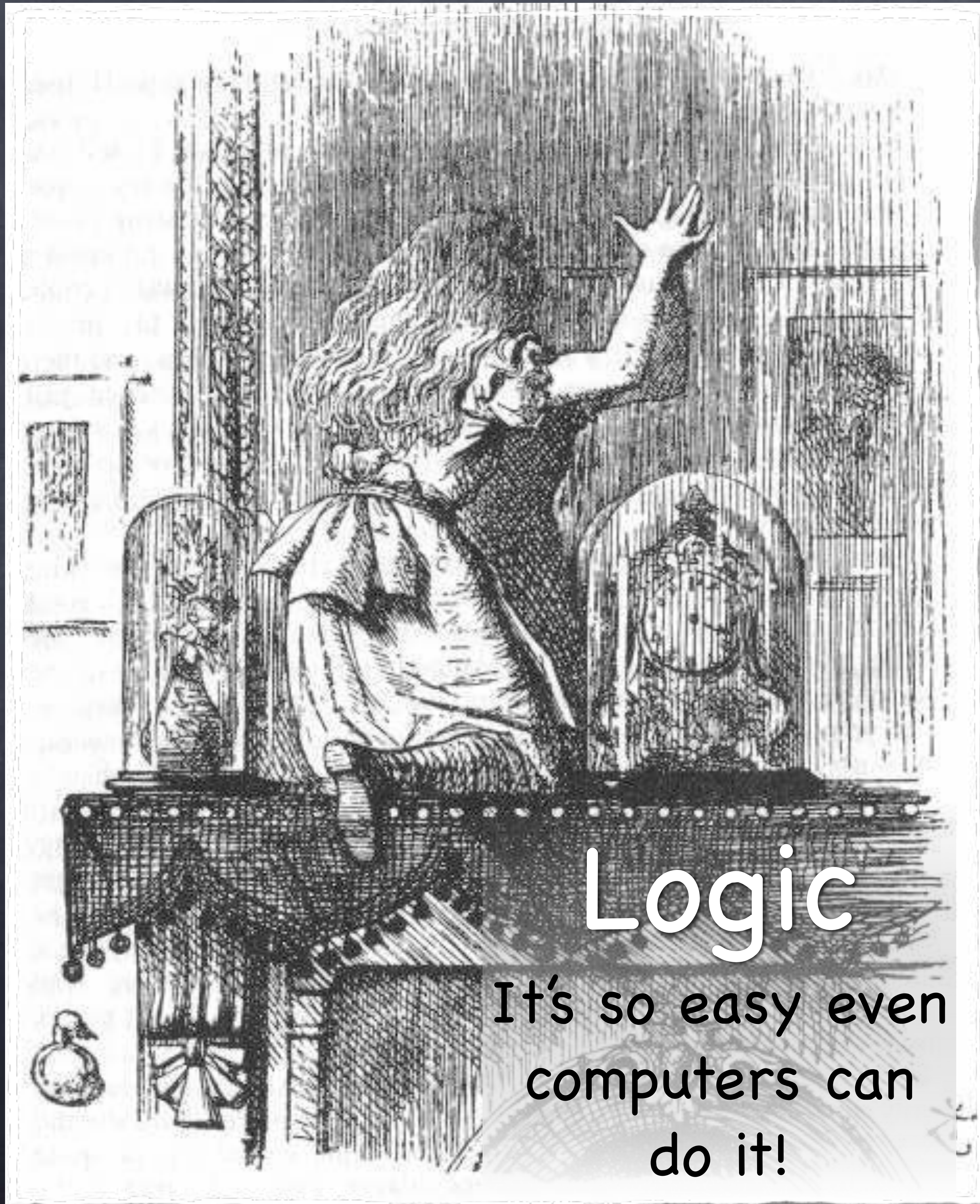
  $\forall x\ P(x) \vee R \equiv (\forall x\ P(x)) \vee R$ $\qquad$ $\exists x\ P(x) \vee R \equiv (\exists x\ P(x)) \vee R$

  $\forall x\ P(x) \wedge R \equiv (\forall x\ P(x)) \wedge R$ $\qquad$ $\exists x\ P(x) \wedge R \equiv (\exists x\ P(x)) \wedge R$
- $\forall x\ R \rightarrow P(x) \equiv R \rightarrow (\forall x\ P(x))$ $\qquad$ $\exists x\ R \rightarrow P(x) \equiv R \rightarrow (\exists x\ P(x))$
- $\forall x\ \underline{P(x) \rightarrow R} \equiv (\exists x\ P(x)) \rightarrow R$ $\qquad$ $\exists x\ \underline{P(x) \rightarrow R} \equiv (\forall x\ P(x)) \rightarrow R$
- $(\forall x\ P(x)) \wedge (\forall x\ Q(x)) \equiv \forall x\ (P(x) \wedge Q(x))$

  $(\exists x\ P(x)) \vee (\exists x\ Q(x)) \equiv \exists x\ (P(x) \vee Q(x))$
- $(\forall x\ P(x)) \vee (\forall x\ Q(x)) \equiv \forall x\ \forall y\ P(x) \vee Q(y)$

  $(\exists x\ P(x)) \wedge (\exists x\ Q(x)) \equiv \exists x\ \exists y\ P(x) \wedge Q(y)$
- $\neg(\forall x\ P(x)) \equiv \exists x\ \neg P(x)$ $\qquad$ $\neg(\exists x\ P(x)) \equiv \forall x\ \neg P(x)$

  $\nexists x\ P(x)$



Logic

It's so easy even computers can do it!

Through the Looking Glass

The Looking Glass

- A mirror which shows the negation of every proposition
- Reflection changes T & F to F & T (resp.)
  - ∨ & ∧ are reflected as ∧ & ∨ (resp.)

Flies(Alice) is False

¬ Flies(Alice) is True

Flies(Alice) ∨ Flies(J'wock) is True

∨	T	F
T	T	T
F	T	F

?	F	T
F	F	F
T	F	T

¬Flies(Alice) ? ¬Flies(J'wock) is False

∧	F	T
F	F	F
T	F	T

∨	T	F
T	T	T
F	T	F

The Looking Glass

- A mirror which shows the negation of every ~~proposition~~ wire
- Reflection changes T & F to F & T (resp.)
  - $\vee$ & $\wedge$ are reflected as $\wedge$ & $\vee$ (resp.)

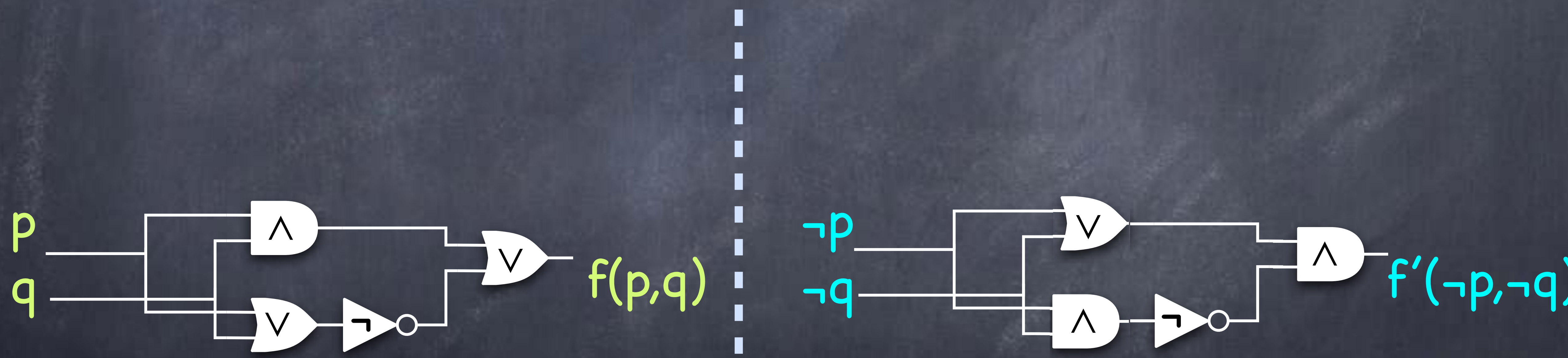
De Morgan's Law

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

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

The Looking Glass

- A mirror which shows the negation of every ~~proposition~~ wire
- Reflection changes T & F to F & T (resp.)
  - $\vee$ & $\wedge$ are reflected as $\wedge$ & $\vee$ (resp.)



$$\boxed{\neg f(p,q) \equiv f'(\neg p, \neg q)}$$

The Looking Glass

- Reflection changes T & F to F & T (resp.)
  - $\vee$ & $\wedge$ are reflected as $\wedge$ & $\vee$ (resp.)
  - $\boxed{\forall \text{ \& } \exists \text{ are reflected as } \exists \text{ \& } \forall \text{ (resp.)}}$

p, q  $p \wedge q$	$\neg p$, $\neg q$  $\neg p \vee \neg q$
p, q  $p \vee q$	$\neg p$, $\neg q$  $\neg p \wedge \neg q$
$\forall x\ Pred(x)$	$\exists x\ \neg Pred(x)$
$\exists x\ Pred(x)$	$\forall x\ \neg Pred(x)$

Two quantifiers

x	y	Likes(x,y)	∃y Likes(x,y) i.e., LikesSomeone(x)
Alice	Alice	TRUE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	
Jabberwock	Alice	FALSE	TRUE
	Jabberwock	TRUE	
	Flamingo	FALSE	
Flamingo	Alice	FALSE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	

- $\forall x\ \exists y\ Likes(x,y)$
  - Everyone likes someone
  - $\forall x\ LikesSomeone(x)$
  - True

Two quantifiers

x	y	Likes(x,y)	∃y Likes(x,y) i.e., LikesSomeone(x)
Alice	Alice	TRUE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	
Jabberwock	Alice	FALSE	TRUE
	Jabberwock	TRUE	
	Flamingo	FALSE	
Flamingo	Alice	FALSE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	

- ∀x ∃y Likes(x,y)
  - Everyone likes someone
  - ∀x LikesSomeone(x)
  - True

- ∃x ¬(∃y Likes(x,y))

Two quantifiers

x	y	Likes(x,y)	∃y Likes(x,y) i.e., LikesSomeone(x)
Alice	Alice	TRUE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	
Jabberwock	Alice	FALSE	TRUE
	Jabberwock	TRUE	
	Flamingo	FALSE	
Flamingo	Alice	FALSE	TRUE
	Jabberwock	FALSE	
	Flamingo	TRUE	

- $\forall x\ \exists y\ Likes(x,y)$
  - Everyone likes someone
  - $\forall x\ LikesSomeone(x)$
  - True

- $\exists x\ \forall y\ \neg Likes(x,y)$
  - Someone doesn't like anyone
  - $\exists x\ DoesntLikeAnyone(x)$
  - False

Two quantifiers

x	y	Likes(x,y)
Alice	Alice	TRUE
	Jabberwock	FALSE
	Flamingo	TRUE
Jabberwock	Alice	FALSE
	Jabberwock	TRUE
	Flamingo	FALSE
Flamingo	Alice	FALSE
	Jabberwock	FALSE
	Flamingo	TRUE

- $\exists y\ \underline{\forall x\ \text{Likes}(x,y)}$

Two quantifiers

x	y	Likes(x,y)	∀x Likes(x,y) i.e., EveryoneLikes(y)
Alice	Alice	TRUE	FALSE
Jabberwock		FALSE	
Flamingo		FALSE	
Alice	Jabberwock	FALSE	FALSE
Jabberwock		TRUE	
Flamingo		FALSE	
Alice	Flamingo	TRUE	FALSE
Jabberwock		FALSE	
Flamingo		TRUE	

- ∃y ∀x Likes(x,y)
  - Someone is liked by everyone
  - False

- ∀y ∃x ¬Likes(x,y)
  - Everyone is disliked by someone
  - True