# Predicate Logic

# Propositional Logic Redux

Recall: A *proposition* is a declarative sentence that is either true or false.

A *compound proposition* is a expression formed from propositional variables (e.g., P, Q), values (T and F) and logical operators (e.g., ^, v).

# And elimination / and introduction

*And elimination* lets us infer the truth of either of the conjuncts from the truth of a conjunctive sentence. For instance, $P \wedge Q$ lets us conclude both P and Q are true.

*And introduction* allows us to infer the truth of a conjunction from the truth of its conjuncts. For instance, if P and Q are true, then $P \wedge Q$ is true.

# Tautology and Contradiction

A *tautology* is a compound proposition that is always true.

A *contradiction* is a compound proposition that is always false.

A *contingency* is neither a tautology nor a contradiction.

A compound proposition is *satisfiable* if there is at least one assignment of truth values to the variables that makes the statement true.

# Examples

| $p$ | $\neg p$ | $p \vee \neg p$ | $p \wedge \neg p$ |
|---|---|---|---|
| T | F | T | F |
| F | T | T | F |

# Logical Equivalence

Two compound propositions, p and q, are logically equivalent if $p \leftrightarrow q$ is a tautology.

Notation: $p \equiv q$

De Morgan's Laws:

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

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

# Predicate Logic

Some statements cannot be expressed in propositional logic, such as:

All men are mortal.

Some trees have needles.

$X > 3$.

Predicate logic can express these statements and make inferences on them.

# Subjects and Predicates

In predicate logic,

every atomic sentence consists of

one predicate

and

one or more "subjects"

including subjects, direct objects, indirect objects, etc.

in mathematics "subjects" are called "arguments"

(Shakespeare used the term 'argument' to mean 'subject')

# Example 1

| Subject | Predicate |
| --- | --- |
| Jay | is asleep |
| Kay | is awake |
| Elle | is a dog |

# Example 2

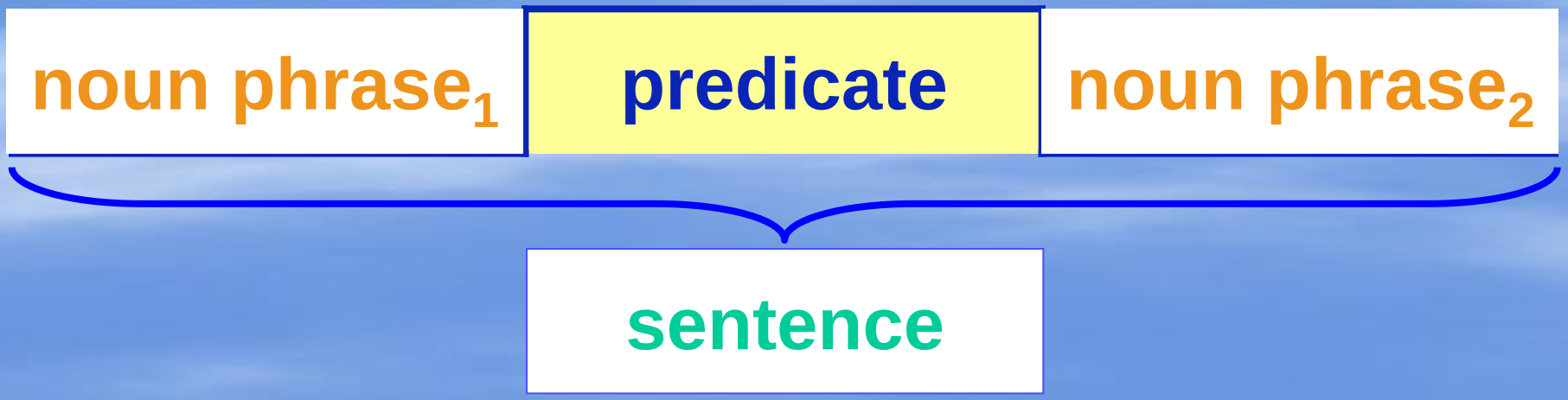
| Subject | Predicate | Object |
|---|---|---|
| Jay | respects | Kay |
| Kay | is next to | Elle |
| Elle | is taller than | Jay |

# Example 3

| Subject | Predicate | Direct Object | | Indirect Object |
|---|---|---|---|---|
| Jay | sold | Elle | to | Kay |
| Kay | bought | Elle | from | Jay |
| Kay | prefers | Elle | to | Jay |

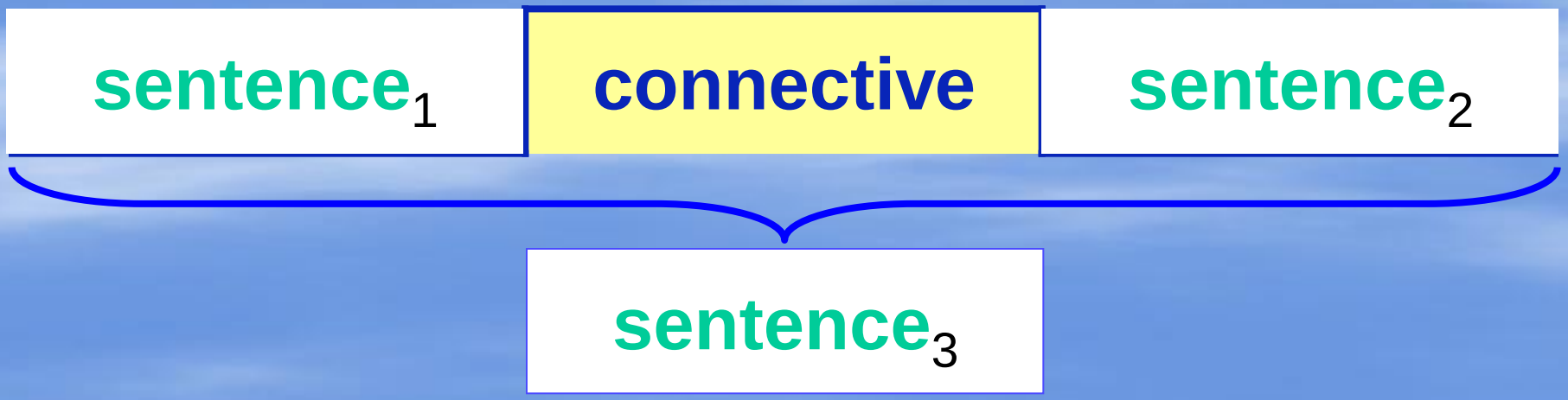
# What is a Predicate?

A **predicate** is an "incomplete" expression –
i.e., an expression with one or more **blanks** –
such that,
whenever the blanks are filled by **noun phrases**,
the resulting expression is a **sentence**.



# Compare with Connective

A **connective** is an "incomplete" expression –
i.e., an expression with one or more **blanks** –
such that,
whenever the blanks are filled by **sentences**,
the resulting expression is a **sentence**.



# Examples









# Symbolization Convention

1. Predicates are symbolized by upper case letters.
2. Subjects are symbolized by lower case letters.
3. Predicates are placed first.
4. Subjects are placed second.

**Pred $sub_1$ $sub_2$ …**

# Examples

| | |
|---|---|
| Jay is tall | Tj |
| Kay is tall | Tk |
| Jay is taller than Kay | Tjk |
| Kay is taller than Elle | Tke |
| Jay recommended Kay to Elle | Rjke |
| Kay recommended Elle to Jay | Rkej |

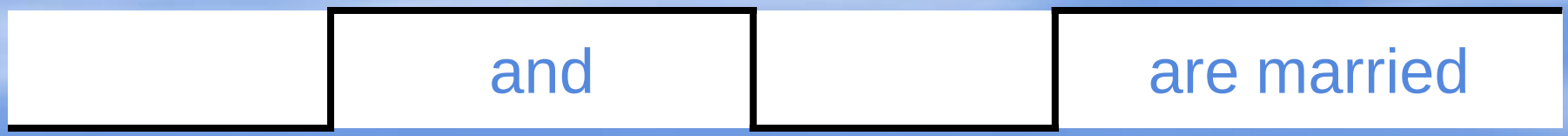
# Compound Sentences - 1

| | |
|---|---|
| Jay is not tall | ~Tj |
| Jay is not taller than Kay | ~Tjk |
| both Jay and Kay are tall | Tj & Tk |
| neither Jay nor Kay is tall | ~Tj & ~Tk |
| Jay is taller than both Kay and Elle | Tjk & Tje |

# Compound Sentences - 2

| | |
|---|---|
| Jay and Kay are married (*individually*)<br>=<br>Jay is married, **and** Kay is married | Mj & Mk |

| | |
|---|---|
| Jay and Kay are married (*to each other*) | Mjk |



# Quantifiers

Quantifiers are linguistic expressions denoting **quantity**.

**Examples**

every, all, any, each, both, either

some, most, many, several, few

no, neither

at least one, at least two, etc.

at most one, at most two, etc.

exactly one, exactly two, etc.

# Quantifiers – 2

Quantifiers combine
common nouns and verb phrases
to form sentences.

Examples

**every** senior is happy

**no** freshman is happy

**at least one** junior is happy

**few** sophomores are happy

**most** graduates are happy

predicate logic treats
both common nouns
and verb phrases
as predicates

# The Two Special Quantifiers of Predicate Logic

| official name | English expressions | symbol |
|---|---|---|
| universal quantifier | every, any | $\forall$ |
| existential quantifier | some, at least one | $\exists$ |

# Names of Symbols

∀ upside-down ‘A’

∃ backwards ‘E’

Actually, they are both upside-down.

A E

# How Traditional Logic Does Quantifiers

Quantifier Phrases are Simply Noun Phrases

| | |
|---|---|
| every one | is happy |
| some one | is happy |
| Jay | is happy |
| Kay | is happy |
| subject | predicate |

# How Modern Logic Does Quantifiers

Quantifier Phrases are

**Sentential Adverbs**

# Scope of Quantifiers

The part of a logical expression to which a quantifier is applied is called the scope of this quantifier.

e.g., $(\forall x\, P(x)) \wedge (\exists y\, Q(y))$

e.g., $(\forall x\, P(x)) \wedge (\exists x\, Q(x))$

# Existential Quantifier

| | |
|---|---|
| some one | is happy |
| there is some one | who is happy |
| there is some one such that | he/she is happy |
| there is some x such that | x is happy |
| ∃x | Hx |

pronunciation

there is an x (such that) H x

# Universal Quantifier

| | |
|---|---|
| every one | is happy |
| every one is such that | he/she is happy |
| whoever you are | you are happy |
| no matter who you are | you are happy |
| no matter who x is | x is happy |
| $\forall x$ | $Hx$ |

pronunciation

for any x H x

# Negating Quantifiers

modern logic takes '∃' to mean — at least one

which means — one or more

which means — one, or two, or three, or …

if a (counting) number is — not one or more

it must be — zero

thus, the — negation of 'at least one'

is — 'not at least one'

which is — 'none'

# Negative-Existential Quantifier

| | |
|---|---|
| no one | is happy |
| there is no one | who is happy |
| there is no one such that | he/she is happy |
| there is no x such that | x is happy |
| there is not some x such that | x is happy |
| $\sim\exists x$ | $Hx$ |

pronunciation

there is no x (such that) H x

# Negative-Universal Quantifier

| | |
|---|---|
| not every one | is happy |
| not every one is such that | he/she is H |
| it is not true that whoever you are | you are H |
| it is not true that no matter who you are | you are H |
| it is not true that no matter who x is | x is H |
| $\sim\forall x$ | Hx |

pronunciation

not for any x H x

# Quantifying Negations - 1

| | |
|---|---|
| suppose | not everyone is happy |
| then | there is someone |
| | who is |
| | not happy |
| i.e., | there is some x : |
| | x is not happy |

$\sim \forall x Hx$

II

$\exists x \sim Hx$

the converse argument is also valid

# Quantifying Negations - 2

| | |
|---|---|
| suppose | no one is happy |
| then | no matter who you are |
| | you are |
| | not happy |
| i.e. | no matter who x is |
| | x is not happy |

$\sim\exists xHx$

II

$\forall x\sim Hx$

the converse argument is also valid

# Why is Predicate Logic Useful?

Verifying Program Correctness

- Given

```
if (x<0) x = -x;
```

- What is true before? (called *precondition*)
  - ??
- What is true after? (called *postcondition*)
  - greaterThan(x,0)

Systems Specifications

- E.g., Every mail message larger than one megabyte will be compressed.