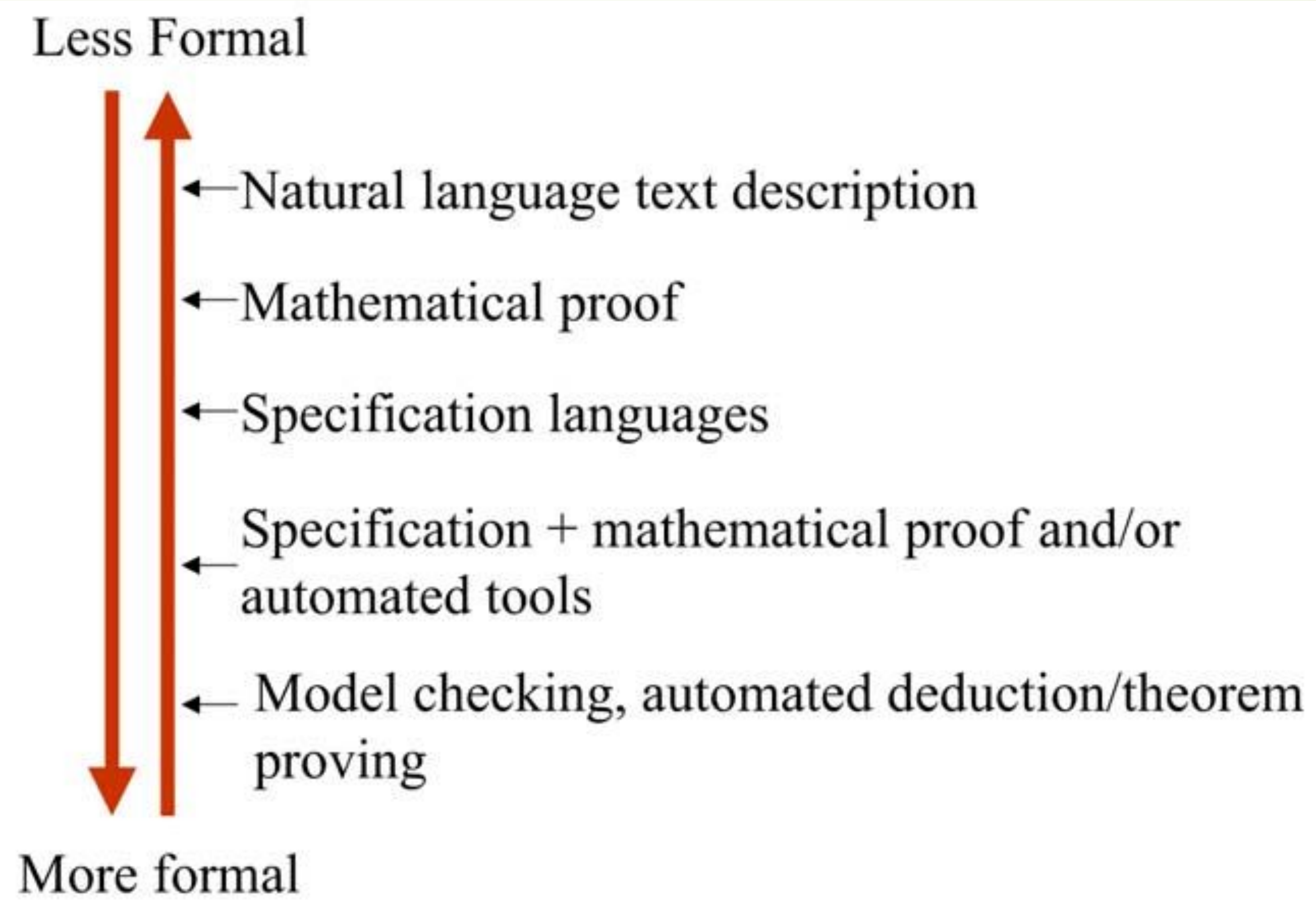
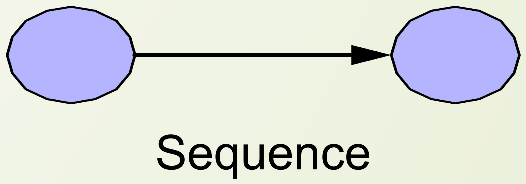
SE4033 Formal Methods for Software Engineering

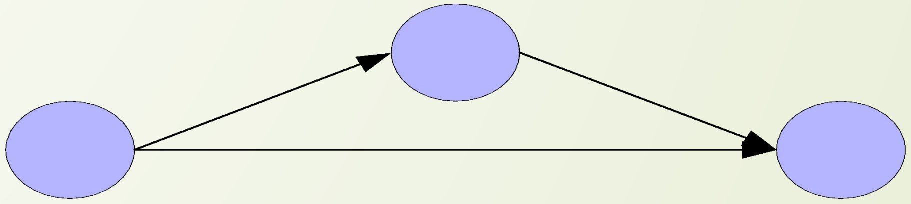
Formal Methods for Software

Engineering

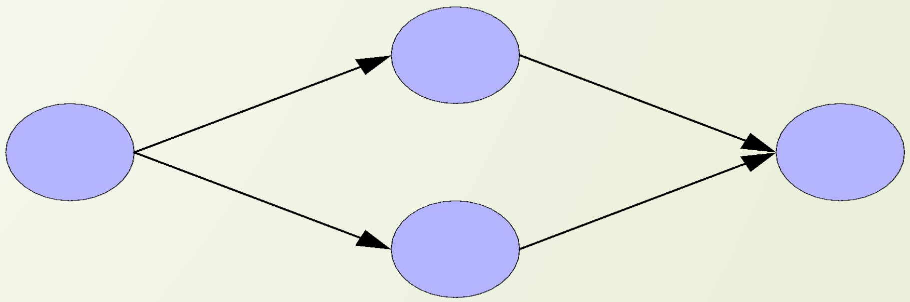
Formalization Spectrum



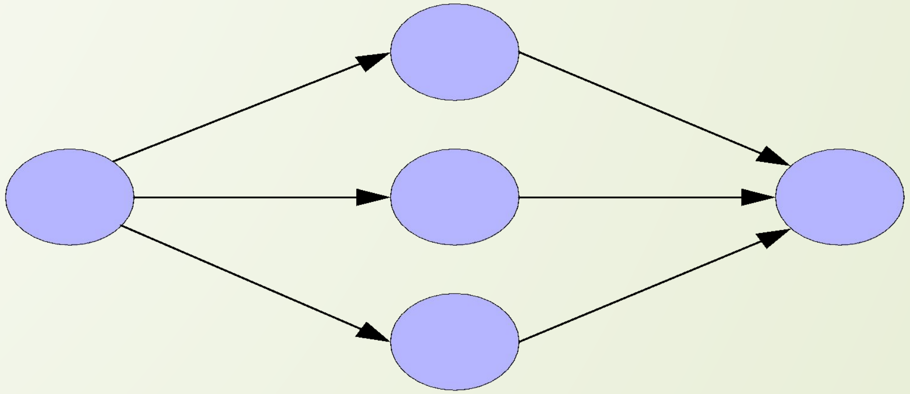

Sequence



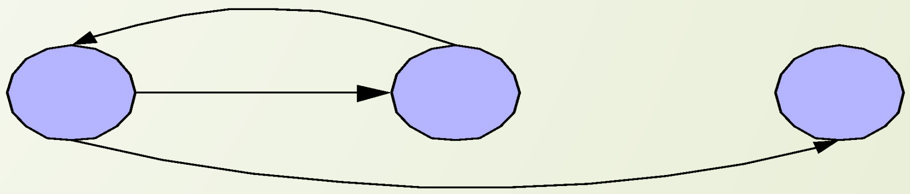
Selection – if statement



Selection – if-else statement



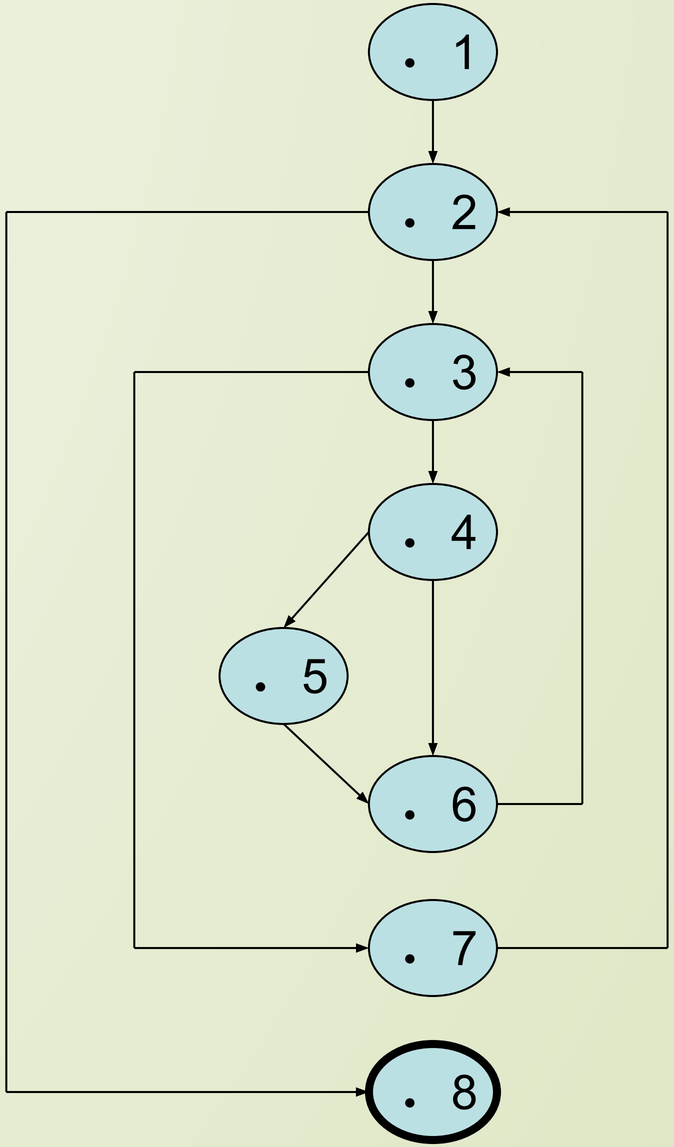
Selection – case statement



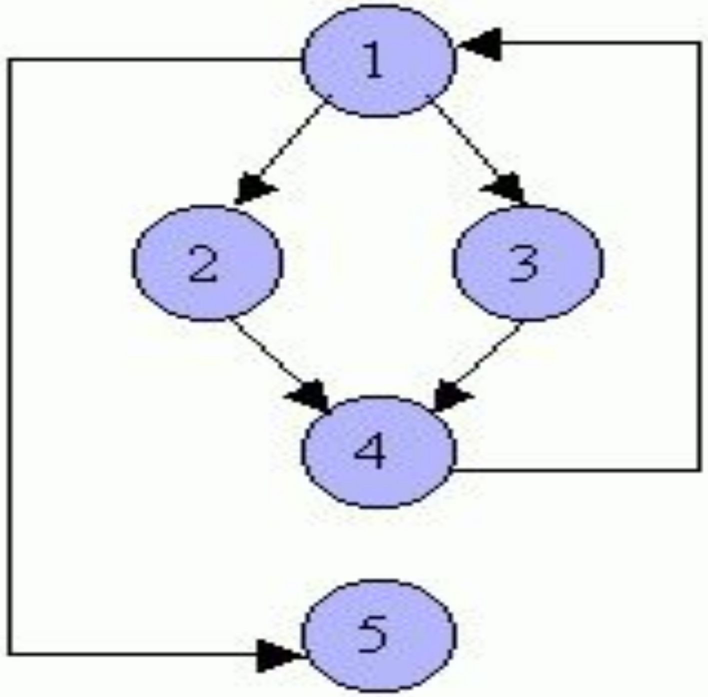
- Loop

Flow graph for bubble sort

```
sorted = false;                // 1
while (!sorted) {              // 2
   sorted = true;
   for (int i = 0; i < SIZE-1; i++) {   // 3
    if (a[i] > a[i+1]) {       // 4
      swap(a[i], a[i+1]);          // 5
      sorted = false;
    }
   }                  //6
}                     //7
                      //8
```



```
for (i = 0; i < N; i++) {          //1
  if (condition1)
        // do something here        //2
  else
        // do something here        //3
  // something here                 //4
}                                   //5
```



- 2^N Paths

Implication

P	Q	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Bi-conditional – if and only if

P	Q	P ⇔ Q
T	T	T
T	F	F
F	F	T
F	T	F

P ⇔ Q means P ⇒ Q ∧ Q ⇒ P

- A compound proposition that is always true, irrespective of the truth values of the comprising propositions, is called a tautology.

  $p \lor \neg p$

- The propositions **p** and **q** are called logically equivalent if $\mathbf{p} \Leftrightarrow \mathbf{q}$ is tautology.

- It is written as ,

  $\mathbf{p \equiv q}$

  For example: $\neg\,(\mathbf{p} \vee \mathbf{q}) \equiv \neg\,\mathbf{p} \wedge \neg\,\mathbf{q}$

Some useful equivalences

p or true ≡ true	p or false ≡ p

Some useful equivalences

p or true ≡ true	p or false ≡ p
p and true ≡ p	p and false ≡ false

Some useful equivalences

• p or true ≡ true	• p or false ≡ p
p and true ≡ p	p and false ≡ false
true $\Rightarrow$ p ≡ p	false $\Rightarrow$ p ≡ true
p $\Rightarrow$ true ≡ true	p $\Rightarrow$ false ≡ not p

Some useful equivalences

• p or true ≡ true	• p or false ≡ p
p and true ≡ p	p and false ≡ false
true $\Rightarrow$ p ≡ p	false $\Rightarrow$ p ≡ true
p $\Rightarrow$ true ≡ true	p $\Rightarrow$ false ≡ not p
p or p ≡ p	p and p ≡ p

Some useful equivalences

• p or true ≡ true	• p or false ≡ p
p and true ≡ p	p and false ≡ false
true ⇒ p ≡ p	false ⇒ p ≡ true
p ⇒ true ≡ true	p ⇒ false ≡ not p
p or p ≡ p	p and p ≡ p

not not p ≡ p

Some useful equivalences

• p or true ≡ true	• p or false ≡ p
p and true ≡ p	p and false ≡ false
true $\Rightarrow$ p ≡ p	false $\Rightarrow$ p ≡ true
p $\Rightarrow$ true ≡ true	p $\Rightarrow$ false ≡ not p
p or p ≡ p	p and p ≡ p

not not p ≡ p

p or not p ≡ true	p and not p ≡ false

Some useful equivalences

distributivity of

- and over or
- or over and
- or over $\Rightarrow$
- $\Rightarrow$ over and
- $\Rightarrow$ over or
- $\Rightarrow$ over $\Rightarrow$
- $\Rightarrow$ over $\Leftrightarrow$

associativity of

$\vee$, $\wedge$, and $\Leftrightarrow$

Commutativity of

$\vee$, $\wedge$, and $\Leftrightarrow$

- Demorgan's law

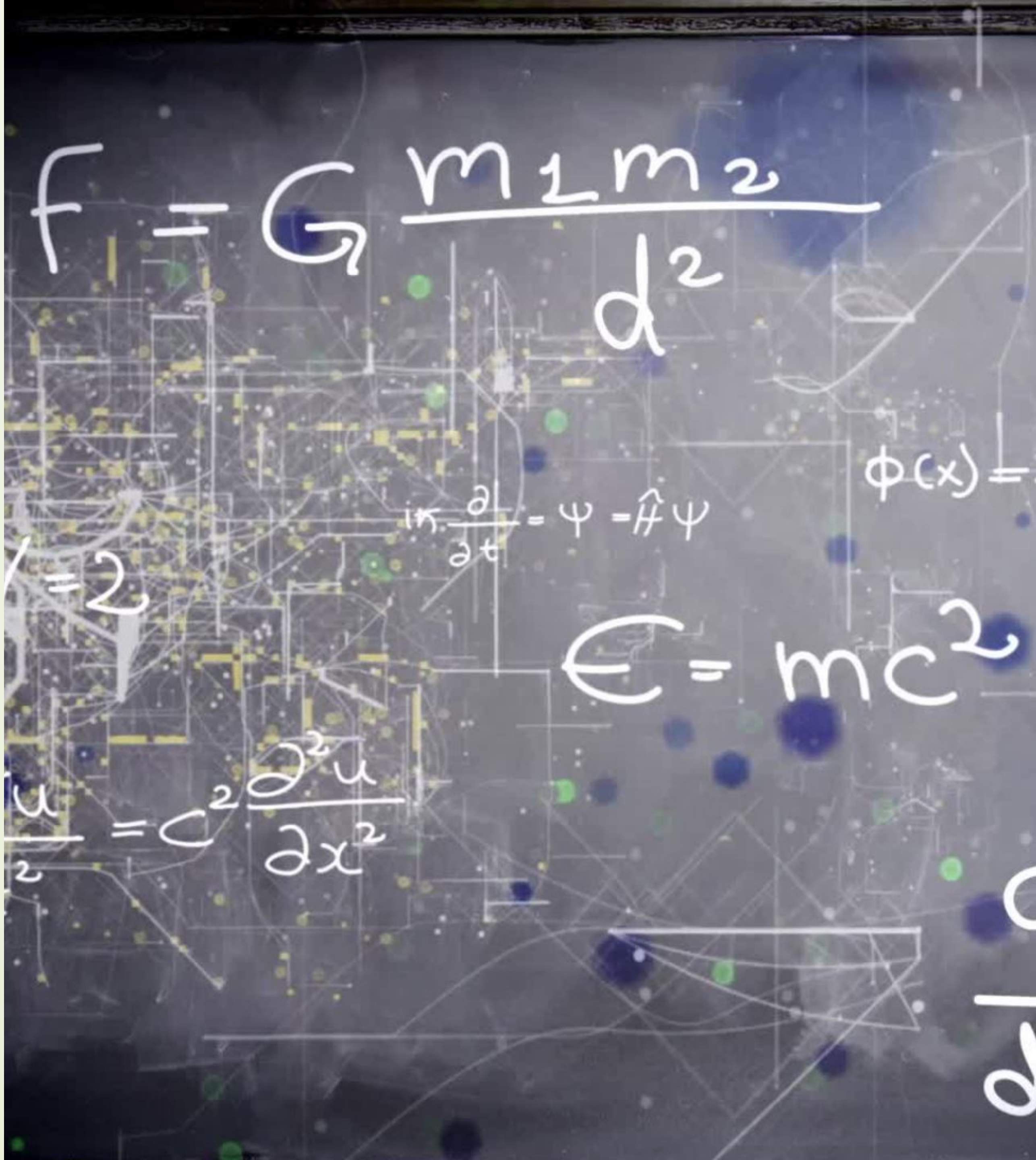
- Implication
- if and only if

Logic problem for the day

Someone asks person A, “Are you a knight?” He replies, “If I am a knight then I’ll eat my hat”. Prove that A has to eat his hat.

Logic

- Logic or propositional calculus is based on statements, which have truth values (true or false).
- Symbolic Statements
- p V q stands for p or q
- P => q stands for p logically implies q
- P <=> q stands for p is logically equivalent to q



Logic

Symbol	Meaning
$\vee$	or
$\wedge$	and
$\neg$	not
$\Rightarrow$	logically implies
$\Leftrightarrow$	logically equivalent
$\forall$	for all
$\exists$	there exists