# Predicate Logic

Laura Kovács

Recap:

# Boolean Algebra and Propositional Logic

| | |
|---|---|
| 0, 1<br>(other notation: t, f) | |
| boolean variables $a \in \{0,1\}$ | |
| boolean operators $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ | |
| boolean functions:<br>• 0 and 1 are boolean functions;<br>• boolean variables are boolean functions;<br>• if a is a boolean function, then $\neg a$ is a boolean function;<br>• if a and b are boolean functions, then $a \wedge b$, $a \vee b$, $a \Rightarrow b$, $a \Leftrightarrow b$ are boolean functions. | |
| truth value of a boolean function<br>(truth tables) | |

Recap:

# Boolean Algebra and Propositional Logic

| | |
|---|---|
| 0, 1<br>(other notation: t, f) | True, False |
| boolean variables $a \in \{0,1\}$ | atomic formulas (atoms) $p \in \{True, False\}$ |
| boolean operators $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ | logical connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ |
| boolean functions:<br>• 0 and 1 are boolean functions;<br>• boolean variables are boolean functions;<br>• if a is a boolean function, then $\neg a$ is a boolean function;<br>• if a and b are boolean functions, then $a \wedge b$, $a \vee b$, $a \Rightarrow b$, $a \Leftrightarrow b$ are boolean functions. | propositional formulas (propositions, Aussagen):<br>• True and False are propositional formulas;<br>• atomic formulas are propositional formulas;<br>• if a is a propositional formula, then $\neg a$ is a propositional formula;<br>• if a and b are propositional formulas, then $a \wedge b$, $a \vee b$, $a \Rightarrow b$, $a \Leftrightarrow b$ are propositional formulas. |
| truth value of a boolean function<br>(truth tables) | truth value of a propositional formula<br>(truth tables) |

Recap:

# Boolean Algebra and Propositional Logic

| | |
|---|---|
| 0, 1<br>(other notation: t, f) | True, False |
| boolean variables a∈{0,1} | atomic formulas (atoms) p∈{True,False} |
| boolean operators ¬, ∧, ∨, ⇒, ⇔ | logical connectives ¬, ∧, ∨, ⇒, ⇔ |
| boolean functions:<br>• 0 and 1 are boolean functions;<br>• boolean variables are boolean functions;<br>• if a is a boolean function, then ¬a is a boolean function;<br>• if a and b are boolean functions, then a∧b, a∨b, a⇒b, a⇔b are boolean functions. | propositional formulas (propositions, Aussagen):<br>• True and False are propositional formulas;<br>• atomic formulas are propositional formulas;<br>• if a is a propositional formula, then ¬a is a propositional formula;<br>• if a and b are propositional formulas, then a∧b, a∨b, a⇒b, a⇔b are propositional formulas. |
| truth value of a boolean function<br>(truth tables) | truth value of a propositional formula<br>(truth tables) |

Boolean algebra and propositional logic are different ways of interpreting the *same* thing.

Usage: computer processors (circuits, switches).

# Satisfiability Example: Russian Spy Puzzle

There are three persons: Stirlitz, Mueller, and Eismann.

It is known that exactly one of them is Russian, while the other two are Germans. Moreover, every Russian must be a spy.

When Stirlitz meets Mueller in a corridor, he makes the following joke: “you know, Mueller, you are as German as I am Russian”. It is known that Stirlitz always says the truth when he is joking.

We have to establish that Eismann is not a Russian spy.

How can we solve problems of this kind? By Propositonal Logic!
(see the course webpage for solution)

# Propositions as mathematical statements

“10 <u>is greater than 5</u>” has value True

“ 1 <u>is greater than 5</u>” has value False

# Nice, but …

"10 <u>is greater than 5</u>" has value True

" 1 <u>is greater than 5</u>" has value False

" x <u>is greater than 5</u>" has value … depends on the value of variable x

# Nice, but …

"10 is greater than 5" has value True

" 1 is greater than 5" has value False

" x is greater than 5" has value … depends on the value of variable x

Speaks about many simple propositions. It is not a propositional formula!

# Nice, but …propositional logic is not enough!

"10 is greater than 5" has value True

" 1 is greater than 5" has value False

" x is greater than 5" has value … depends on the value of variable x

We need **Predicate Logic**

Speaks about many simple propositions. It is not a propositional formula!

Usage: software systems in financial services, automotive industry, medicine, …

# Predicate Logic – Predicates (Praedikaten, Eigenschaften)

$$\text{“x } \underbrace{\underline{\text{is greater than 5}}}_{}\text{”}$$

is a *predicate*, denote it by F

describes a property of objects represented by variables from an universe (domain, Bereichsangabe)

# Predicate Logic

$$\underbrace{\text{“x }\underline{\text{is greater than 5}}\text{”}}_{F(x)}$$

F(1) is False

F(4) is False

F(10) is True

# Predicate Logic - example

“x <u>is greater then</u> y” ← G(x,y)

What is

- G(2,0)
- G(7,1)
- G(7,9)

“7 less than 9” can be represented by G(7,9) or G(9,7)?

# Predicate Logic

Predicate with a variable is NOT a proposition!

Example: “$\underbrace{x\ \underline{\text{is greater than 5}}}_{F(x)}$” with variable x over the universe of natural numbers

# Predicate Logic

Predicate with a variable is NOT a proposition!

Example: “$\underbrace{\text{x \underline{is greater than 5}}}_{F(x)}$” with variable x over the universe of natural numbers

Predicates with a variable can be made propositions:

- assign a value to the variable;

  Example: F(1), F(4)

- quantify the variable using quantifiers.

  Example: For all natural numbers x, x is greater than 5

  For some natural number x, x is greater than 5

# Predicate Logic

## Predicate with a variable is NOT a proposition!

Example: "x is greater than 5" with variable x over the universe of natural numbers

$\underbrace{\text{x is greater than 5}}_{F(x)}$

## Predicates with a variable can be made propositions:

- assign a value to the variable;

  Example: F(1), F(4)

- quantify the variable using quantifiers.

  Example: For all natural numbers x, x is greater than 5 ← False

  For some natural number x, x is greater than 5 ← True

# Predicate Logic - Quantifiers

- Universal quantifier: ∀

$$\forall \mathbf{x}\mathbf{:}\ \mathbf{U(x)}\mathbf{:}\ \mathbf{F(x)}$$

"for every object x in the universe U, F(x) holds"
"for all (any) object x in the universe U, F(x) holds"

Example: $\forall \mathbf{x}\mathbf{:}\mathbf{x}\in\mathbb{N}\mathbf{:}\mathbf{x}>\mathbf{5}$

- Existential quantifier: ∃

$$\exists \mathbf{x}\mathbf{:}\ \mathbf{U(x)}\mathbf{:}\ \mathbf{F(x)}$$

"for some object x in the universe U, F(x) holds"
"there exist an object x in the universe U such that F(x) holds"

Example: $\exists\ \mathbf{x}\mathbf{:}\mathbf{x}\in\mathbb{N}\mathbf{:}\mathbf{x}>\mathbf{5}$

# Predicate Logic - Quantifiers

- Universal quantifier: $\forall$

$$\forall \mathbf{x}: \mathbf{U(x)}: \mathbf{F(x)}$$

"for every object x in the universe U, F(x) holds"
"for all (any) object x in the universe U, F(x) holds"

Example: $\forall \mathbf{x}:\mathbf{x} \in \mathbb{N}:\mathbf{x}>\mathbf{5}$

Note: $\forall \mathbf{x}: \mathbf{False}: \mathbf{F(x)}$ is **True**

- Existential quantifier: $\exists$

$$\exists \mathbf{x}: \mathbf{U(x)}: \mathbf{F(x)}$$

"for some object x in the universe U, F(x) holds"
"there exist an object x in the universe U such that F(x) holds"

Example: $\exists\ \mathbf{x}:\mathbf{x} \in \mathbb{N}:\mathbf{x}>\mathbf{5}$

Note: $\exists \mathbf{x}: \mathbf{False}: \mathbf{F(x)}$ is **False**

# Predicate Logic – English with Quantifiers

"All cars have wheels"

"Someone calls you"

# Predicate Logic – Programs and Quantifiers

$\forall$ i: $0 \leq i < N$: $a_i = 0$

all elements $a_i$ of array a (of length N) are zero

$\forall$ i: $1 \leq i < N$: $a_{i-1} \leq a_i$

elements of array a are monotonically increasing
(array a is increasingly sorted)

$\forall$ i,j: $0 \leq i < j < N$: $a_i \leq a_j$

elements of array a are monotonically increasing
(array a is increasingly sorted)

$\forall$ i,j: $0 \leq i,j < N$: $a_i = a_j$

all elements of array a are equal

$\forall$ i: $0 \leq i < N$: $a_i \leq a_1$ ...

$a_1$ is the greatest element of array a

# Predicate Logic – Programs and Quantifiers

$\forall$ i: $0 \le i < N$: $a_i = 0$

all elements $a_i$ of array a (of length N) are zero

$\forall$ i: $1 \le i < N$: $a_{i-1} \le a_i$

elements of array a are monotonically increasing
(array a is increasingly sorted)

$\forall$ i,j: $0 \le i < j < N$: $a_i \le a_j$

elements of array a are monotonically increasing
(array a is increasingly sorted)

$\forall$ i,j: $0 \le i,j < N$: $a_i = a_j$

all elements of array a are equal

$\forall$ i: $0 \le i < N$: $a_i \le a_1$ ...

$a_1$ is the greatest element of array a

$\exists$ i: $0 \le i < N$: $a_i = 0$

at least one element of array a (of length N) is zero

$\neg(\exists$ i: $1 \le i < N$: $a_{i-1} > a_i)$

elements of array a are monotonically increasing
(array a is increasingly sorted)

$\neg$ ($\exists$ i,j: $0 \le i < j < N$: $a_i > a_j$)

elements of array a are monotonically increasing
(array a is increasingly sorted)

$\exists$ i,j: $0 \le i < j < N$: $a_i = a_j$

at least two elements of array a are equal

$\neg$ ($\exists$ i: $0 \le i < N$: $a_i > a_1$)

$a_1$ is the greatest element of array a

# Predicate Logic - Well-Formed Formulas (WFF)

- True and False are WFFs;
- Propositional variables are WFFs;
- Predicates with variables are WFFs;
- If A and B are WFFs, then $\neg A$, $A \wedge B$, $A \vee B$, $A \Rightarrow B$, $A \Leftrightarrow B$ are also WFFs;
- If x is a variable (representing objects of the universe U) and A is a WFF, then $\forall x: U(x): F(x)$ and $\exists x: U(x): F(x)$ are also WFFs.

# Predicate Logic – Quantifiers and Negation

## De Morgan's Axiom $\neg\exists$ :

$$\neg(\exists\ x: U(x): F(x)) \ \Leftrightarrow \forall\ x: U(x): \neg\ F(x)$$

„There exist no x in the universe U such that F(x) holds“

is the same as (equivalent to)

„For all x in the universe U, F(x) does not hold“

## De Morgan's Axiom $\neg\forall$ :

$$\neg(\forall\ x: U(x): F(x)) \ \Leftrightarrow \exists\ x: U(x): \neg\ F(x)$$

„Not for all x in the universe U, F(x) holds“

is the same as (equivalent to)

„There exist an x in the universe U, such that F(x) does not hold“

# Predicate Logic – Quantifiers and Negation

Examples:

- Rewrite $(\exists\ i: 0 \le i < N: a_i \ne 0)$ into an equivalent WFF with $\forall$
- Rewrite $(\forall\ i: 1 \le i < N: a_{i-1} < a_i)$ into an equivalent WFF with $\exists$
- Rewrite $(\exists\ i,j: 0 \le i < j < N: a_i \ne a_j)$ into an equivalent WFF with $\forall$

# Predicate Logic – Quantifiers and Connectives $\wedge$, $\vee$

- Distributivity of $\forall \wedge$

$$\forall x: U(x): (P(x) \wedge Q(x)) \Leftrightarrow (\forall x: U(x): P(x)) \wedge (\forall x: U(x): Q(x))$$

- Distributivity of $\exists \vee$

$$\exists x: U(x): (P(x) \vee Q(x)) \Leftrightarrow (\exists x: U(x): P(x)) \vee (\exists x: U(x): Q(x))$$

- Non-distributivity of $\forall \vee$

$$(\forall x: U(x): P(x)) \vee (\forall x: U(x): Q(x)) \Rightarrow \forall x: U(x): (P(x) \vee Q(x))$$

- Non-distributivity of $\exists \wedge$

$$\exists x: U(x): (P(x) \wedge Q(x)) \Rightarrow (\exists i: U(x): P(x)) \wedge (\exists x: U(x): Q(x))$$

# Predicate Logic – Bound and Free Variables

An appearance (Auftreten) of variable is bound (gebunden) in a WFF if

- a specific value is assigned to it

or

- it is quantified.

If a variable is not bound in a WFF then it is free (frei).

The extent of the effect of a quantifier is called the scope (Einflussbereich) of the quantifier:
- if not indicated by (), the scope is the smallest WFF following the quantification.

Example:

$\exists$ i: 0<i,j<N: P(i,j) i is bound, j is free. Scope of $\exists$ i is P(i,j).

$\forall$ i: 0<i<N: ( ($\exists$ j: 1<j<N: P(i,j)) $\vee$ Q(i,j)) i and j are bound in P(i,j);
i is bound and j is free in Q(i,j);
Scope of $\exists$ j is P(i,j);
Scope of $\forall$ i is ($\exists$ j: 1<j<N: P(i,j) $\vee$ Q(i,j)).

# Predicate Logic – Sequences of Quantifiers

*Swapping **SAME** Quantifiers in a Sequence of Quantifiers:*

$$\forall x: U_1(x): ( \forall y: U_2(y): Q(x,y)) \Leftrightarrow \forall y: U_2(y): ( \forall x: U_1(x): Q(x,y))$$

$$\exists x: U_1(x): (\exists y: U_2(y): Q(x,y)) \Leftrightarrow \exists y: U_2(y): (\exists x: U_1(x): Q(x,y))$$

# Predicate Logic – Sequences of Quantifiers

*Swapping **SAME** Quantifiers in a Sequence of Quantifiers:*

$$\forall x: U_1(x): (\forall y: U_2(y): Q(x,y)) \Leftrightarrow \forall y: U_2(y): (\forall x: U_1(x): Q(x,y))$$

$$\exists x: U_1(x): (\exists y: U_2(y): Q(x,y)) \Leftrightarrow \exists y: U_2(y): (\exists x: U_1(x): Q(x,y))$$

*But:*

$$\exists x: U_1(x): (\forall y: U_2(y): Q(x,y)) \not\Leftrightarrow \forall y: U_2(y): (\exists x: U_1(x): Q(x,y))$$

# Predicate Logic – Sequences of Quantifiers

*Swapping **SAME** Quantifiers in a Sequence of Quantifiers:*

$$\forall x: U_1(x): (\forall y: U_2(y): Q(x,y)) \Leftrightarrow \forall y: U_2(y): (\forall x: U_1(x): Q(x,y))$$

$$\exists x: U_1(x): (\exists y: U_2(y): Q(x,y)) \Leftrightarrow \exists y: U_2(y): (\exists x: U_1(x): Q(x,y))$$

*But:*

$$\exists x: U_1(x): (\forall y: U_2(y): Q(x,y)) \not\Leftrightarrow \forall y: U_2(y): (\exists x: U_1(x): Q(x,y))$$

$$\exists x: U_1(x): (\forall y: U_2(y): Q(x,y)) \Rightarrow \forall y: U_2(y): (\exists x: U_1(x): Q(x,y))$$

$$\forall y: U_2(y): (\exists x: U_1(x): Q(x,y)) \not\Rightarrow \exists x: U_1(y): (\forall y: U_2(y): Q(x,y))$$

# Predicate Logic – Properties of Quantifiers

*Consider a formula P which does not contain x as a free variable.*

*(Assume that the domain U is nonempty!)*

*Then:*

$$\forall x: U(x): (Q(x) \vee P) \;\Leftrightarrow\; (\forall x: U(x): Q(x)) \vee P$$

$$\forall x: U(x): (Q(x) \wedge P) \;\Leftrightarrow\; (\forall x: U(x): Q(x)) \wedge P$$

# Predicate Logic – Properties of Quantifiers

*Consider a formula P which does not contain x as a free variable.*

*(Assume that the domain U is nonempty!)*

*Then:*

$$\forall x: U(x): (Q(x) \lor P) \iff (\forall x: U(x): Q(x)) \lor P$$

$$\forall x: U(x): (Q(x) \land P) \iff (\forall x: U(x): Q(x)) \land P$$

$$\exists x: U(x): (Q(x) \land P) \iff (\exists x: U(x): Q(x)) \land P$$

$$\exists x: U(x): (Q(x) \land P) \iff (\exists x: U(x): Q(x)) \land P$$

# Predicate Logic - Example

Programming example:

| |
|---|
| Given: an array a[0..N-1] of length N≥3 |
| Find: i, j, k such that: |
| - i is the index of the greatest element of a; |
| - j is the index of the $2^{nd}$ greatest element of a; |
| - k is the index of the $3^{rd}$ greatest element of a. |

***Formulate the wanted property in predicate logic!***

# Predicate Logic - Example

Programming example:

| |
|---|
| Given: an array a[0..N-1] of length $N \geq 3$ |
| Find: i, j, k such that: |
| - i is the index of the greatest element of a; |
| - j is the index of the 2nd greatest element of a; |
| - k is the index of the 3rd greatest element of a. |

***Formulate the wanted property in predicate logic!***

$$\exists\, i,j,k: (0 \leq i,j,k < N) \wedge (i \neq j) \wedge (i \neq k) \wedge (j \neq k): \\ \quad ( (a_i \geq a_j \geq a_k) \wedge \\ \quad (\forall\, m: (0 \leq m < N) \wedge (m \neq i) \wedge (m \neq j) \wedge (m \neq k): a_k \geq a_m) )$$

# Example of Ambiguity of Spoken English

„All elements of an array a of length N are either zero or one "

could be interpreted as

$$\forall i: 0 \le i < N: ( a_i = 0 \lor a_i = 1 )$$

or

$$(\forall i: 0 \le i < N: a_i = 0) \lor (\forall i: 0 \le i < N: a_i = 1)$$

# Examples of Ambiguity of Spoken English

„The elements of arrays a and b are the same (both arrays of length N).“

can be interpreted as

$\forall i: 0 \le i < N: (a_i = b_i)$ — ex: a=(2,3,5), b=(2,3,5)

or

$\forall i: 1 \le i < N: ((a_i = a_{i-1}) \wedge (b_i = b_{i-1}))$ — ex: a=(2,2,2), b=(3,3,3)

or

$(\forall i: 1 \le i < N: (a_i = a_{i-1})) \wedge (\forall i: 0 \le i < N: (a_i = b_i))$ — ex: a=(2,2,2), b=(2,2,2)

or

$\forall i: 0 \le i < N: (\exists\, j: 0 \le j < N: (a_i = b_j))$ — ex: a=(2,3,5), b=(5,2,3)

a=(2,3,3), b=(2,2,3)

# Further Quantifiers for Math Reasoning

(not the standard quantifiers of predicate logic)

## Anz-Quantifier:

$$\text{Anz } x\text{: } U(x)\text{: } F(x)$$

gives the „number of all those objects x from the universe U for which F(x) holds“

Note: Anz x: False: F(x)  is 0.

# Further Quantifiers for Math Reasoning

## Anz-Quantifier:

Anz x: U(x): F(x)

gives the „number of all those objects x from the universe U for which F(x) holds“

Note: Anz x: False: F(x) is 0.

$$((\text{Anz } x: U(x): F(x)) = 0) \Leftrightarrow \forall x: ???$$

$$((\text{Anz } x: U(x): F(x)) \geq 0) \Leftrightarrow \exists x: ???$$

# Further Quantifiers for Math Reasoning

## Anz-Quantifier:

Anz x: U(x): F(x)

gives the „number of all those objects x from the universe U for which F(x) holds“

Note: Anz x: False: F(x) is 0.

$$((\text{Anz } x: U(x): F(x)) = 0) \Leftrightarrow \forall x: U(x): \neg F(x)$$

$$((\text{Anz } x: U(x): F(x)) \geq 0) \Leftrightarrow \exists x: U(x): F(x)$$

# Further Quantifiers for Math Reasoning

## Anz-Quantifier Examples:

*Given arrays a and b. Both arrays are of length N. The elements of both arrays are natural numbers.*

$(\text{Anz } i: 0 \leq i < N: a_i = 0) = 0$ ... all elements $a_i$ of a are different than 0

$(\text{Anz } i: 0 \leq i < N: a_i = x) > 0$ ... at least one element $a_i$ has the value x ($x \in \mathbb{N}$)

$(\text{Anz } i: 0 \leq i < N: a_i < x) = (\text{Anz } i: 0 \leq i < N: a_i > x)$ ... x is the median value of a

$\forall x :: ( (\text{Anz } i: 0 \leq i < N: a_i = x) = (\text{Anz } i: 0 \leq i < N: b_i = x) )$ ... a is a permutation of b

# Further Quantifiers for Math Reasoning

**Sum-Quantifier:**

Sum x: U(x): T(x)

gives the „sum of the terms T(x) for all objects x from the universe U“

Note: Sum x: False: T(x) is 0.

(Anz x: U(x): T(x)) $\Leftrightarrow$

# Further Quantifiers for Math Reasoning

## Sum-Quantifier:

Sum x: U(x): T(x)

gives the „sum of the terms T(x) for all objects x from the universe U"

Note: Sum x: False: T(x) is 0.

(Anz x: U(x): T(x)) $\Leftrightarrow$ Sum x: (U(x) $\wedge$ T(x)): 1

# Further Quantifiers for Math Reasoning

## Sum-Quantifier Examples:

(Sum i: 1≤i≤n: i) = n(n+1)/2 ... Sum of the first n natural numbers

(Sum i: 0≤i<n: 2i+1) = $n^2$ ... Sum of odd (ungerade) naturals numbers smaller than 2n

(Sum i: 0≤i<n: $2^i$) = $2^n$-1 ... Sum of the first n powers of 2

# Further Quantifiers for Math Reasoning

## Min-Quantifier:

$$\text{Min } x\text{: } U(x)\text{: } T(x)$$

gives the „smallest value of terms T(x) for all objects x from the universe U“

Note: Min x: False: T(x) is $+\infty$.

Example: *Given an array a of length N. The elements of a are naturals.*

*The fact that the smallest element of a is m can be equivalently expressed with Min or ∃ ∀ as:*

$$( m = (\text{Min } i\text{: } 0 \le i < N\text{: } a[i]) ) \iff$$

# Further Quantifiers for Math Reasoning

## Min-Quantifier:

$$\text{Min } x: U(x): T(x)$$

gives the „smallest value of terms T(x) for all objects x from the universe U“

Note: Min x: False: T(x) is $+\infty$.

Example: *Given an array a of length N. The elements of a are naturals.*

*The fact that the smallest element of a is m can be equivalently expressed with Min or ∃ ∀ as:*

$$(\ m = (\text{Min } i: 0 \le i < N: a[i])\ ) \Leftrightarrow ((\exists i: 0 \le i < N: m = a[i]) \wedge (\forall\ i: 0 \le i < N: m \le a[i]))$$

# Further Quantifiers for Math Reasoning

## Max-Quantifier:

Max x: U(x): T(x)

gives the „greatest value of terms T(x) for all objects x from the universe U“

Note: Max x: False: T(x) is $-\infty$.

Max x: U(x): T(x) = - (Min x: U(x): - T(x))

Min x: U(x): T(x) = - (Max x: U(x): -T(x))

# Predicate Logic is Useful and Fun!

- One application area: software verification

  Example: <u>if</u> x>0 <u>then</u> x:=x+1; y:=y+1 <u>else</u> x:=x-1; y:=y-1

- It is fun and competitive – see the theorem proving competition:

  http://www.cs.miami.edu/~tptp/CASC/22/