Mathematical and Logical Foundations of Computer Science

Lecture 11 - Predicate Logic (Syntax)

Vincent Rahli

(some slides were adapted from Rajesh Chitnis' slides)

University of Birmingham

Where are we?

- Symbolic logic
- Propositional logic
- **Predicate logic**
- Constructive vs. Classical logic
- Type theory

Today

- Syntax of Predicate Logic

Today

- Syntax of Predicate Logic

Further reading:

- Chapter 7 of
  `http://leanprover.github.io/logic_and_proof/`

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number
- 2 is an odd number

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number
- 2 is an odd number
- $\mathcal{P} = \mathcal{NP}$

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number
- 2 is an odd number
- $\mathcal{P} = \mathcal{NP}$
- Mind the gap!

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number
- 2 is an odd number
- $\mathcal{P} = \mathcal{NP}$
- Mind the gap! (not a proposition)

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number
- 2 is an odd number
- $\mathcal{P} = \mathcal{NP}$
- Mind the gap! (not a proposition)

Grammar: $P ::= a \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \neg P$
where a ranges over **atomic propositions**.

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number
- 2 is an odd number
- $\mathcal{P} = \mathcal{NP}$
- Mind the gap! (not a proposition)

Grammar: $P ::= a \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \neg P$
where a ranges over **atomic propositions**.

Two special atoms: $\top$ stands for True, $\bot$ stands for False

Recap: Propositional Logic

Propositions: Facts (that can in principle be true or false)

- 2 is an even number
- 2 is an odd number
- $\mathcal{P} = \mathcal{NP}$
- Mind the gap! (not a proposition)

Grammar: $P ::= a \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \neg P$
where a ranges over **atomic propositions**.

Two special atoms: $\top$ stands for True, $\bot$ stands for False

Four connectives:

- $P \wedge Q$: we have a proof of both P and Q
- $P \vee Q$: we have a proof of at least one of P and Q
- $P \rightarrow Q$: if we have a proof of P then we have a proof of Q
- $\neg P$: stands for $P \rightarrow \bot$

Recap: Proofs

Natural Deduction

Sequent Calculus

Recap: Proofs

Natural Deduction

introduction/elimination rules

Sequent Calculus

right/left rules

Recap: Proofs

Natural Deduction	Sequent Calculus
introduction/elimination rules	right/left rules
natural proofs	amenable to automation

Recap: Proofs

Natural Deduction

introduction/elimination rules

natural proofs

$$\dfrac{\dfrac{}{A}\,1 \\ \vdots \\ B}{A \to B}\;1\;[\to I]$$

Sequent Calculus

right/left rules

amenable to automation

$$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\;[\to R]$$

Expressiveness of Propositional Logic

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

Expressiveness of Propositional Logic

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

Can we express this in propositional logic?

Expressiveness of Propositional Logic

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

Can we express this in propositional logic?

Another example:

- Every even natural number is not odd
- x is even
- x is not odd

Expressiveness of Propositional Logic

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

Can we express this in propositional logic?

Another example:

- Every even natural number is not odd
- x is even
- x is not odd

Can we express this in propositional logic?

Beyond Propositional Logic

Propositional logic allows us to state facts

Beyond Propositional Logic

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** "objects"

Beyond Propositional Logic

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** "objects"
- e.g., the property of numbers of being even, or odd

Beyond Propositional Logic

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** “objects”
- e.g., the property of numbers of being even, or odd

This brings us to a **richer** logic called **predicate logic**

Beyond Propositional Logic

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** "objects"
- e.g., the property of numbers of being even, or odd

This brings us to a **richer** logic called **predicate logic**

- **contains** propositional logic

Beyond Propositional Logic

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** "objects"
- e.g., the property of numbers of being even, or odd

This brings us to a **richer** logic called **predicate logic**

- **contains** propositional logic
- also known as **first-order logic**

Beyond Propositional Logic

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** "objects"
- e.g., the property of numbers of being even, or odd

This brings us to a **richer** logic called **predicate logic**

- **contains** propositional logic
- also known as **first-order logic**
- Predicate logic allows us to reason about members of a (non-empty) domain

Beyond Propositional Logic

For example, the argument:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

Beyond Propositional Logic

For example, the argument:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

includes the following components:

- Domain = Men

Beyond Propositional Logic

For example, the argument:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

includes the following components:

- Domain = Men
- Socrates is one member of this domain

Beyond Propositional Logic

For example, the argument:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

includes the following components:

- Domain = Men
- Socrates is one member of this domain
- Predicates are “being a man” and “being mortal”

Beyond Propositional Logic

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

Beyond Propositional Logic

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

These 3 tables can be seen as 3 relations:

- $Student(sid, name)$: predicate $Student$ relates student ids and names
- $Module(mid, name)$: predicate $Module$ relates module ids and names
- $Enroll(sid, mid)$: predicate $Enroll$ relates student and module ids

Beyond Propositional Logic

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

These 3 tables can be seen as 3 relations:

- $Student(sid, name)$: predicate $Student$ relates student ids and names
- $Module(mid, name)$: predicate $Module$ relates module ids and names
- $Enroll(sid, mid)$: predicate $Enroll$ relates student and module ids

Domain $=$ all possible values

Beyond Propositional Logic

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

These 3 tables can be seen as 3 relations:

- $Student(sid, name)$: predicate $Student$ relates student ids and names
- $Module(mid, name)$: predicate $Module$ relates module ids and names
- $Enroll(sid, mid)$: predicate $Enroll$ relates student and module ids

Domain = all possible values

A formula can be seen as a query

Beyond Propositional Logic

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

These 3 tables can be seen as 3 relations:

- $Student(sid, name)$: predicate $Student$ relates student ids and names
- $Module(mid, name)$: predicate $Module$ relates module ids and names
- $Enroll(sid, mid)$: predicate $Enroll$ relates student and module ids

Domain = all possible values

A formula can be seen as a query

For example: find the Students x enrolled in the Math module

Beyond Propositional Logic

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

These 3 tables can be seen as 3 relations:

- $Student(sid, name)$: predicate $Student$ relates student ids and names
- $Module(mid, name)$: predicate $Module$ relates module ids and names
- $Enroll(sid, mid)$: predicate $Enroll$ relates student and module ids

Domain = all possible values

A formula can be seen as a query

For example: find the Students x enrolled in the Math module

- $\exists y.\exists z.Student(y, x) \wedge Module(z, \text{Math}) \wedge Enroll(y, z)$

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

We can write this argument as $\forall x.(p(x) \to q(x)), p(s) \vdash q(s)$

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

We can write this argument as $\forall x.(p(x) \to q(x)), p(s) \vdash q(s)$

- **Predicates**:

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

We can write this argument as $\forall x.(p(x) \rightarrow q(x)), p(s) \vdash q(s)$

- **Predicates**:
  - $p(x)$ which states that x is a man
  - $q(x)$ which states that x is mortal

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

We can write this argument as $\forall x.(p(x) \to q(x)), p(s) \vdash q(s)$

- **Predicates**:
  - $p(x)$ which states that x is a man
  - $q(x)$ which states that x is mortal
- **Quantifier**: The "for all" symbol $\forall$

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

We can write this argument as $\forall x.(p(x) \rightarrow q(x)), p(s) \vdash q(s)$

- **Predicates**:
  - $p(x)$ which states that x is a man
  - $q(x)$ which states that x is mortal
- **Quantifier**: The "for all" symbol $\forall$
- **Variable**: x to denote an element of the domain

Key ingredients of Predicate Logic

The key ingredients of predicate logic are

- predicates, quantifiers, variables, functions, and constants

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

We can write this argument as $\forall x.(p(x) \to q(x)), p(s) \vdash q(s)$

- **Predicates**:
  - $p(x)$ which states that x is a man
  - $q(x)$ which states that x is mortal
- **Quantifier**: The "for all" symbol $\forall$
- **Variable**: x to denote an element of the domain
- **Constant**: s which stands for Socrates

Key ingredients of Predicate Logic

Domain (also called universe)

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$
- Similar to variables from programming languages

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$
- Similar to variables from programming languages

Quantifiers

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$
- Similar to variables from programming languages

Quantifiers

- **universal** quantifier
  $\forall x. \cdots$: "for all elements x of the domain"

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$
- Similar to variables from programming languages

Quantifiers

- **universal** quantifier
  $\forall x. \cdots$: "for all elements x of the domain"
- **existential** quantifier
  $\exists x. \cdots$: "there exists an element x of the domain such that"

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$
- Similar to variables from programming languages

Quantifiers

- **universal** quantifier
  $\forall x. \cdots$: "for all elements x of the domain"
- **existential** quantifier
  $\exists x. \cdots$: "there exists an element x of the domain such that"
- quantify over elements of the domain

Key ingredients of Predicate Logic

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$
- Similar to variables from programming languages

Quantifiers

- **universal** quantifier
  $\forall x. \cdots$: "for all elements x of the domain"
- **existential** quantifier
  $\exists x. \cdots$: "there exists an element x of the domain such that"
- quantify over elements of the domain
- **precedence**: lower than the other connectives

Key ingredients of Predicate Logic

Functions

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**
- A function symbol of arity 1 can only be applied to 1 argument, A function symbol of arity 2 can only be applied to 2 arguments, etc.

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**
- A function symbol of arity 1 can only be applied to 1 argument, A function symbol of arity 2 can only be applied to 2 arguments, etc.
- **Notation**: We sometimes write f^k when we want to indicate that the function symbol f has arity k

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**
- A function symbol of arity 1 can only be applied to 1 argument,
  A function symbol of arity 2 can only be applied to 2 arguments, etc.
- **Notation**: We sometimes write f^k when we want to indicate that the function symbol f has arity k

Constants

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**
- A function symbol of arity 1 can only be applied to 1 argument, A function symbol of arity 2 can only be applied to 2 arguments, etc.
- **Notation**: We sometimes write f^k when we want to indicate that the function symbol f has arity k

Constants

- Specific objects in the domain

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**
- A function symbol of arity 1 can only be applied to 1 argument, A function symbol of arity 2 can only be applied to 2 arguments, etc.
- **Notation**: We sometimes write f^k when we want to indicate that the function symbol f has arity k

Constants

- Specific objects in the domain
- Functions of arity 0

Key ingredients of Predicate Logic

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**
- A function symbol of arity 1 can only be applied to 1 argument, A function symbol of arity 2 can only be applied to 2 arguments, etc.
- **Notation**: We sometimes write f^k when we want to indicate that the function symbol f has arity k

Constants

- Specific objects in the domain
- Functions of arity 0
- Usually denoted by $a, b, c, \ldots$

Key ingredients of Predicate Logic

Let the domain be $\mathbb{N}$.

Key ingredients of Predicate Logic

Let the domain be $\mathbb{N}$.

Provide examples of function symbols, along with their arities

Key ingredients of Predicate Logic

Let the domain be $\mathbb{N}$.

Provide examples of function symbols, along with their arities

- $0, 1, 2, \ldots$ are constant symbols (nullary function symbols)

Key ingredients of Predicate Logic

Let the domain be $\mathbb{N}$.

Provide examples of function symbols, along with their arities

- $0, 1, 2, \ldots$ are constant symbols (nullary function symbols)
- `add`: the binary addition function

Key ingredients of Predicate Logic

Let the domain be $\mathbb{N}$.

Provide examples of function symbols, along with their arities

- $0, 1, 2, \ldots$ are constant symbols (nullary function symbols)
- `add`: the binary addition function
- `add`(m, n): addition applied to the two expressions m and n

Key ingredients of Predicate Logic

Let the domain be $\mathbb{N}$.

Provide examples of function symbols, along with their arities

- $0, 1, 2, \ldots$ are constant symbols (nullary function symbols)
- `add`: the binary addition function
- `add`(m, n): addition applied to the two expressions m and n
- `square`: the unary square function

Key ingredients of Predicate Logic

Let the domain be $\mathbb{N}$.

Provide examples of function symbols, along with their arities

- $0, 1, 2, \ldots$ are constant symbols (nullary function symbols)
- `add`: the binary addition function
- $\texttt{add}(m, n)$: addition applied to the two expressions m and n
- `square`: the unary square function
- $\texttt{square}(m)$: square applied to the expression m

Key ingredients of Predicate Logic

Predicates

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

- $\forall x.(p(x) \wedge q(x))$

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

- $\forall x.(p(x) \wedge q(x))$
    - for all x it is true that $p(x)$ and $q(x)$

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

- $\forall x.(p(x) \land q(x))$
  - for all x it is true that $p(x)$ and $q(x)$
- $(\forall x.p(x)) \rightarrow \neg\forall x.q(x)$

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

- $\forall x.(p(x) \land q(x))$
  - for all x it is true that $p(x)$ and $q(x)$
- $(\forall x.p(x)) \rightarrow \neg\forall x.q(x)$
  - if $p(x)$ is true for all x, then $q(x)$ is not true for all x

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

- $\forall x.(p(x) \wedge q(x))$
  - for all x it is true that $p(x)$ and $q(x)$
- $(\forall x.p(x)) \rightarrow \neg\forall x.q(x)$
  - if $p(x)$ is true for all x, then $q(x)$ is not true for all x
- $\exists x.(p(x) \vee \neg q(x))$

Key ingredients of Predicate Logic

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

- $\forall x.(p(x) \wedge q(x))$
  - for all x it is true that $p(x)$ and $q(x)$
- $(\forall x.p(x)) \rightarrow \neg\forall x.q(x)$
  - if $p(x)$ is true for all x, then $q(x)$ is not true for all x
- $\exists x.(p(x) \vee \neg q(x))$
  - there is some x for which $p(x)$ is true or $q(x)$ is not true

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast:
- All red cars are fast:
- Some red cars are fast:
- There are no red cars:
- No fast cars are purple:

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x. f(x)$
- All red cars are fast:
- Some red cars are fast:
- There are no red cars:
- No fast cars are purple:

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x.f(x)$
- All red cars are fast: $\forall x.r(x) \rightarrow f(x)$
- Some red cars are fast:
- There are no red cars:
- No fast cars are purple:

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x.f(x)$
- All red cars are fast: $\forall x.r(x) \rightarrow f(x)$
- Some red cars are fast: $\exists x.r(x) \wedge f(x)$
- There are no red cars:
- No fast cars are purple:

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x)$ = "x is fast"
- $r(x)$ = "x is red"
- $p(x)$ = "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x.f(x)$
- All red cars are fast: $\forall x.r(x) \rightarrow f(x)$
- Some red cars are fast: $\exists x.r(x) \wedge f(x)$
  - Wrong answer: $\exists x.r(x) \rightarrow f(x)$
- There are no red cars:
- No fast cars are purple:

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x.f(x)$
- All red cars are fast: $\forall x.r(x) \to f(x)$
- Some red cars are fast: $\exists x.r(x) \wedge f(x)$
  - Wrong answer: $\exists x.r(x) \to f(x)$
- There are no red cars: $\neg\exists x.r(x)$
- No fast cars are purple:

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x. f(x)$
- All red cars are fast: $\forall x. r(x) \rightarrow f(x)$
- Some red cars are fast: $\exists x. r(x) \wedge f(x)$
  - Wrong answer: $\exists x. r(x) \rightarrow f(x)$
- There are no red cars: $\neg \exists x. r(x)$
  - Alternative answer: $\forall x. \neg r(x)$
- No fast cars are purple:

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x.f(x)$
- All red cars are fast: $\forall x.r(x) \rightarrow f(x)$
- Some red cars are fast: $\exists x.r(x) \wedge f(x)$
  - Wrong answer: $\exists x.r(x) \rightarrow f(x)$
- There are no red cars: $\neg\exists x.r(x)$
  - Alternative answer: $\forall x.\neg r(x)$
- No fast cars are purple: $\neg\exists x.f(x) \wedge p(x)$

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols

- $f(x) =$ "x is fast"
- $r(x) =$ "x is red"
- $p(x) =$ "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x. f(x)$
- All red cars are fast: $\forall x. r(x) \rightarrow f(x)$
- Some red cars are fast: $\exists x. r(x) \wedge f(x)$
  - Wrong answer: $\exists x. r(x) \rightarrow f(x)$
- There are no red cars: $\neg \exists x. r(x)$
  - Alternative answer: $\forall x. \neg r(x)$
- No fast cars are purple: $\neg \exists x. f(x) \wedge p(x)$
  - Alternative answer: $\forall x. f(x) \rightarrow \neg p(x)$

Connections between $\exists$ and $\forall$

To disprove a **"for all"** proposition, we need to find an x for which the predicate is false

Connections between $\exists$ and $\forall$

To disprove a **"for all"** proposition, we need to find an x for which the predicate is false

- $\neg(\forall x.p(x))$ is the same as $\exists x.\neg p(x)$

Connections between $\exists$ and $\forall$

To disprove a **"for all"** proposition, we need to find an x for which the predicate is false

- $\neg(\forall x.p(x))$ is the same as $\exists x.\neg p(x)$

To disprove a **"there exists"** proposition, we need to show that the predicate is false for all x

Connections between $\exists$ and $\forall$

To disprove a **"for all"** proposition, we need to find an x for which the predicate is false

- $\neg(\forall x.p(x))$ is the same as $\exists x.\neg p(x)$

To disprove a **"there exists"** proposition, we need to show that the predicate is false for all x

- $\neg(\exists x.p(x))$ is the same as $\forall x.\neg p(x)$

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

- Example: $m(a, b) =$ "a is married to b"

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

- Example: $m(a, b) =$ "a is married to b"
- Doesn't have to be symmetric!

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

- Example: $m(a, b) =$ "a is married to b"
- Doesn't have to be symmetric!
- Example: $l(a, b) =$ "a likes b"

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

- Example: $m(a, b) =$ "a is married to b"
- Doesn't have to be symmetric!
- Example: $l(a, b) =$ "a likes b"

What are **nullary** predicates (arity 0)?

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

- Example: $m(a, b) =$ "a is married to b"
- Doesn't have to be symmetric!
- Example: $l(a, b) =$ "a likes b"

What are **nullary** predicates (arity 0)?

- Atomic propositions!

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x) =$ "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

- Example: $m(a, b) =$ "a is married to b"
- Doesn't have to be symmetric!
- Example: $l(a, b) =$ "a likes b"

What are **nullary** predicates (arity 0)?

- Atomic propositions!

Notation: We sometimes write p^k when we want to indicate that the predicate symbol p has arity k

Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \to P \mid \forall x.P \mid \exists x.P
\end{array}
$$

Syntax

The syntax of predicate logic is defined by the following grammar:

$$\begin{array}{lcl} t & ::= & x \mid f(t, \ldots, t) \\ P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P \end{array}$$

where:

Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables

Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables
- f ranges over function symbols

Syntax

The syntax of predicate logic is defined by the following grammar:

$$\begin{array}{lll} t & ::= & x \mid f(t, \ldots, t) \\ P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P \end{array}$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n

Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \to P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols

Syntax

The syntax of predicate logic is defined by the following grammar:

$$\begin{array}{lcl} t & ::= & x \mid f(t, \ldots, t) \\ P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P \end{array}$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols
- $p(t_1, \ldots, t_n)$ is a well-formed formula only if p has arity n

Syntax

The syntax of predicate logic is defined by the following grammar:

$$\begin{array}{lll} t & ::= & x \mid f(t, \ldots, t) \\ P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P \end{array}$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols
- $p(t_1, \ldots, t_n)$ is a well-formed formula only if p has arity n

The pair of a collection of function symbols, and a collection of predicate symbols, along with their arities, is called a **signature**.

Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols
- $p(t_1, \ldots, t_n)$ is a well-formed formula only if p has arity n

The pair of a collection of function symbols, and a collection of predicate symbols, along with their arities, is called a **signature**.

The scope of a quantifier extends as far right as possible. E.g., $P \wedge \forall x.p(x) \vee q(x)$ is read as $P \wedge \forall x.(p(x) \vee q(x))$

Examples

Consider the following domain and signature:

- Domain: $\mathbb{N}$
- Functions: $0, 1, 2, \ldots$ (arity 0); $+$ (arity 2)
- Predicates: $\texttt{prime}$, $\texttt{even}$, $\texttt{odd}$ (arity 1); $=$, $>$, $\geqslant$ (arity 2)

Examples

Consider the following domain and signature:

- Domain: $\mathbb{N}$
- Functions: $0, 1, 2, \ldots$ (arity 0); $+$ (arity 2)
- Predicates: $\texttt{prime}$, $\texttt{even}$, $\texttt{odd}$ (arity 1); $=$, $>$, $\geqslant$ (arity 2)

Express the following sentences in predicate logic

- All prime numbers are either 2 or odd.
- Every even number is equal to the sum of two primes.
- There is no number greater than all numbers.
- All numbers have a number greater than them.

Examples

Consider the following domain and signature:

- Domain: $\mathbb{N}$
- Functions: $0, 1, 2, \ldots$ (arity 0); $+$ (arity 2)
- Predicates: $\texttt{prime}$, $\texttt{even}$, $\texttt{odd}$ (arity 1); $=$, $>$, $\geqslant$ (arity 2)

Express the following sentences in predicate logic

- All prime numbers are either 2 or odd.
  $\forall x.\texttt{prime}(x) \to x = 2 \vee \texttt{odd}(x)$
- Every even number is equal to the sum of two primes.
  $\forall x.\texttt{even}(x) \to \exists y.\exists z.\texttt{prime}(y) \wedge \texttt{prime}(z) \wedge x = y + z$
- There is no number greater than all numbers.
  $\neg\exists x.\forall y.x > y$
- All numbers have a number greater than them.
  $\forall x.\exists y.y > x$

Natural Deduction rules for ∀ and ∃?

Propositional logic: Each connective has two inference rules

Natural Deduction rules for $\forall$ and $\exists$?

Propositional logic: Each connective has two inference rules

- One for introduction
- One for elimination

Natural Deduction rules for $\forall$ and $\exists$?

Propositional logic: Each connective has two inference rules

- One for introduction
- One for elimination

Introduction and elimination rules for $\forall$ and $\exists$?

Natural Deduction rules for ∀ and ∃?

Propositional logic: Each connective has two inference rules

- One for introduction
- One for elimination

Introduction and elimination rules for ∀ and ∃?

$$\frac{?}{\forall x.P}\ [\forall I] \qquad \frac{\forall y.P}{?}\ [\forall E]$$

$$\frac{?}{\exists x.P}\ [\exists I] \qquad \frac{\exists y.P}{?}\ [\exists E]$$

Conclusion

What did we cover today?

- Predicate logic (syntax)

Conclusion

What did we cover today?

- Predicate logic (syntax)

Next time?

- Predicate logic (Natural Deduction)