Foundations of Description Logics

Arina Britz

abritz@sun.ac.za

ISAO, Cape Town 2018

Origins

- Logic in the East – Buddhist and Islamic traditions
- Logic in the West – Aristotle (300s BC)
- Logicism – Boole, Frege (late 1800s), Russell (early 1900s)
- Symbolic logic – Gödel and Tarski (1930s)
- Computation and IT – Von Neumann and Turing (1940s)
- Classical logic – Idealist modelling and reasoning
- Non-classical logics
- Logic engineering

Among all the liberal arts, the first is logic – John of Salisbury

Syllogisms

- Logical argument forms
- Statements made about predicates / subjects / categories:
  - All A are B
  - No A are B
  - Some A are B
  - Not all A are B

Syllogisms

- Logical argument forms
- Statements made about predicates / subjects / categories:
  - All A are B
  - No A are B
  - Some A are B
  - Not all A are B
- Let's write $\neg$ whenever we mean 'not':
  - All A are B
  - All A are $\neg B$
  - Some A are B
  - Some A are $\neg B$

Syllogisms

- Logical argument forms
- Statements made about predicates / subjects / categories:
  - All A are B
  - No A are B
  - Some A are B
  - Not all A are B
- Let's write $\neg$ whenever we mean 'not':
  - All A are B
  - All A are $\neg B$
  - Some A are B
  - Some A are $\neg B$
- Let's write $\sqsubseteq$ for 'all ... are', and $\not\sqsubseteq$ instead of 'not all ... are':
  - $A \sqsubseteq B$
  - $A \sqsubseteq \neg B$
  - $A \not\sqsubseteq \neg B$
  - $A \not\sqsubseteq B$

From traditional to modern logic

- Predicates, argument forms, limited negation, concept inclusion, assertions ✓
- Fully symbolic language with compositional semantics ✗

From traditional to modern logic

- Predicates, argument forms, limited negation, concept inclusion, assertions ✓
- Fully symbolic language with compositional semantics ✗
- Boolean connectives:
  - AND: $\wedge$
  - OR: $\vee$
  - NOT: $\neg$
  - ONLY IF: $\rightarrow$
  - … BUT, UNLESS, IF AND ONLY IF, …

From traditional to modern logic

- Predicates, argument forms, limited negation, concept inclusion, assertions ✓
- Fully symbolic language with compositional semantics ✗
- Boolean connectives:
  - AND: $\wedge$
  - OR: $\vee$
  - NOT: $\neg$
  - ONLY IF: $\rightarrow$
  - ... BUT, UNLESS, IF AND ONLY IF, ...
- Objects:
  - Named objects (constant symbols): $a, b, c, \ldots$
  - Placeholders for unnamed objects (variables): $x, y, z, \ldots$
- Predicates: $A, B, \ldots$ (unary, binary, ..., n-ary)

From traditional to modern logic

- Predicates, argument forms, limited negation, concept inclusion, assertions ✓
- Fully symbolic language with compositional semantics ✗
- Boolean connectives:
  - AND: $\wedge$
  - OR: $\vee$
  - NOT: $\neg$
  - ONLY IF: $\rightarrow$
  - … BUT, UNLESS, IF AND ONLY IF, …
- Objects:
  - Named objects (constant symbols): $a, b, c, \ldots$
  - Placeholders for unnamed objects (variables): $x, y, z, \ldots$
- Predicates: $A, B, \ldots$ (unary, binary, ..., n-ary)
- Quantifiers:
  - ALL: $\forall$
  - SOME: $\exists$

Translating sentences to predicate logic

- Simple statements:
  - All A are B: $\forall x(A(x) \rightarrow B(x))$
  - No A are B: $\neg\exists x(A(x) \wedge B(x))$
  - Some A are B: $\exists x(A(x) \wedge B(x))$
  - Not all A are B: $\neg\forall x(A(x) \rightarrow B(x))$

Translating sentences to predicate logic

- Simple statements:
  - All A are B: $\forall x(A(x) \to B(x))$
  - No A are B: $\neg\exists x(A(x) \wedge B(x))$
  - Some A are B: $\exists x(A(x) \wedge B(x))$
  - Not all A are B: $\neg\forall x(A(x) \to B(x))$
- Sentences with multiple quantifiers and binary predicates:
  - Every boy loves a girl:

$$\forall x(B(x) \to \exists y(G(y) \wedge L(x, y)))$$

  - No girl who loves a boy is not loved by some boy:

$$\neg\exists x(G(x) \wedge \exists y(B(y) \wedge L(x, y)) \wedge \neg\exists z(B(z) \wedge L(z, x)))$$

  - There is a cycle of n alternating boys and girls holding hands.

Building complex formulas

Definition (The language of predicate logic[1])

- An atomic formula is an n-ary predicate symbol A followed by n arguments, which can be constants or variables
- An atomic formula is a formula
- If ϕ and ψ are formulas, then so are: $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, and $\phi \rightarrow \psi$
- If ϕ is a formula and x is a variable, then $\forall x(\phi)$ and $\exists x(\phi)$ are formulas
- A sentence is a formula in which there are no free variables

Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt – Wittgenstein

[1] without function symbols

Exercise

Which of the following are predicate formulas?

- $\neg A(c)$
- $\forall x(P(x) \vee R(x, y, z))$
- $R(x, y) \wedge R(x, y, z)$
- $\exists y \forall x(P(x) \vee Q(x, y))$
- $\exists P(P(x))$
- $\exists x(R(x, A(c)) \rightarrow A(x))$

A mind all logic is like a knife all blade; it makes the hand bleed that uses it – Rabindranath Tagore

Exercise

Which of the following are predicate formulas?

- $\neg A(c)$ ✓
- $\forall x(P(x) \vee R(x, y, z))$
- $R(x, y) \wedge R(x, y, z)$
- $\exists y \forall x(P(x) \vee Q(x, y))$
- $\exists P(P(x))$
- $\exists x(R(x, A(c)) \rightarrow A(x))$

A mind all logic is like a knife all blade; it makes the hand bleed that uses it – Rabindranath Tagore

Exercise

Which of the following are predicate formulas?

- $\neg A(c)$ ✓
- $\forall x(P(x) \vee R(x, y, z))$ ✓
- $R(x, y) \wedge R(x, y, z)$
- $\exists y \forall x(P(x) \vee Q(x, y))$
- $\exists P(P(x))$
- $\exists x(R(x, A(c)) \rightarrow A(x))$

A mind all logic is like a knife all blade; it makes the hand bleed that uses it – Rabindranath Tagore

Exercise

Which of the following are predicate formulas?

- $\neg A(c)$ ✓
- $\forall x(P(x) \vee R(x, y, z))$ ✓
- $R(x, y) \wedge R(x, y, z)$ ✗
- $\exists y \forall x(P(x) \vee Q(x, y))$
- $\exists P(P(x))$
- $\exists x(R(x, A(c)) \rightarrow A(x))$

A mind all logic is like a knife all blade; it makes the hand bleed that uses it – Rabindranath Tagore

Exercise

Which of the following are predicate formulas?

- $\neg A(c)$ ✓
- $\forall x(P(x) \vee R(x, y, z))$ ✓
- $R(x, y) \wedge R(x, y, z)$ ✗
- $\exists y \forall x(P(x) \vee Q(x, y))$ ✓
- $\exists P(P(x))$
- $\exists x(R(x, A(c)) \rightarrow A(x))$

A mind all logic is like a knife all blade; it makes the hand bleed that uses it – Rabindranath Tagore

Exercise

Which of the following are predicate formulas?

- $\neg A(c)$ ✓
- $\forall x(P(x) \vee R(x, y, z))$ ✓
- $R(x, y) \wedge R(x, y, z)$ ✗
- $\exists y \forall x(P(x) \vee Q(x, y))$ ✓
- $\exists P(P(x))$ ✗
- $\exists x(R(x, A(c)) \to A(x))$

A mind all logic is like a knife all blade; it makes the hand bleed that uses it – Rabindranath Tagore

Exercise

Which of the following are predicate formulas?

- $\neg A(c)$ ✓
- $\forall x(P(x) \vee R(x, y, z))$ ✓
- $R(x, y) \wedge R(x, y, z)$ ✗
- $\exists y \forall x(P(x) \vee Q(x, y))$ ✓
- $\exists P(P(x))$ ✗
- $\exists x(R(x, A(c)) \rightarrow A(x))$ ✗

A mind all logic is like a knife all blade; it makes the hand bleed that uses it – Rabindranath Tagore

Exercise

Translate to predicate logic:

- Every small dog travelling with its owner is happy.
- Tintin owns a small, happy dog.
- Milou is a small dog who travels with Tintin.

Does it follow that Milou is happy?

Exercise

Translate to predicate logic:

- Every small dog travelling with its owner is happy.
- Tintin owns a small, happy dog.
- Milou is a small dog who travels with Tintin.

Does it follow that Milou is happy?

- $\forall x \forall y ((\text{Small}(x) \land \text{Dog}(x) \land \text{TravelsWith}(x, y) \land \text{Owns}(y, x)) \rightarrow \text{Happy}(x))$

Exercise

Translate to predicate logic:

- Every small dog travelling with its owner is happy.
- Tintin owns a small, happy dog.
- Milou is a small dog who travels with Tintin.

Does it follow that Milou is happy?

- $\forall x \forall y((\text{Small}(x) \wedge \text{Dog}(x) \wedge \text{TravelsWith}(x, y) \wedge \text{Owns}(y, x)) \rightarrow \text{Happy}(x))$
- $\exists x(\text{Small}(x) \wedge \text{Dog}(x) \wedge \text{Happy}(x) \wedge \text{Owns}(\text{tintin}, x))$

Exercise

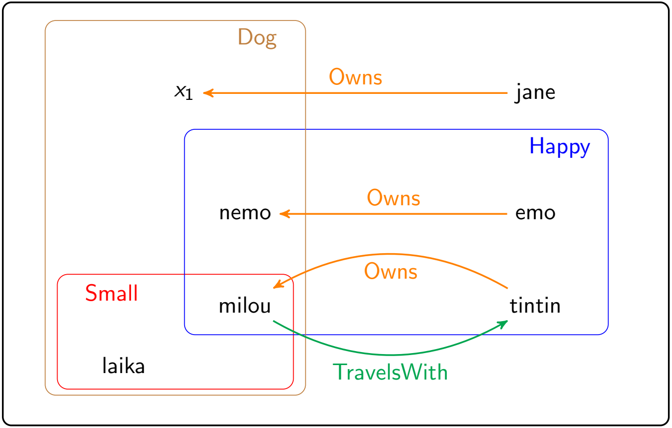
Translate to predicate logic:

- Every small dog travelling with its owner is happy.
- Tintin owns a small, happy dog.
- Milou is a small dog who travels with Tintin.

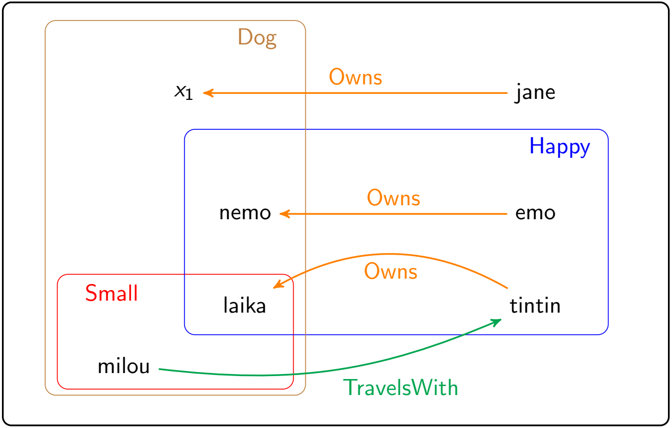
Does it follow that Milou is happy?

- $\forall x \forall y((\text{Small}(x) \wedge \text{Dog}(x) \wedge \text{TravelsWith}(x, y) \wedge \text{Owns}(y, x)) \rightarrow \text{Happy}(x))$
- $\exists x(\text{Small}(x) \wedge \text{Dog}(x) \wedge \text{Happy}(x) \wedge \text{Owns}(\text{tintin}, x))$
- $\text{Dog}(\text{milou}) \wedge \text{Small}(\text{milou}) \wedge \text{TravelsWith}(\text{milou}, \text{tintin})$

Semantic intuition



Semantic intuition



Semantics

- A sentence is either true or false in any given interpretation
- The truth of a complex sentence in an interpretation is determined only by the truth of its components

Semantic entailment

- Formalisation of valid argument forms
- $\phi_1, \ldots, \phi_k \models \psi$:
  - From premises $\phi_1, \ldots, \phi_k$ the conclusion ψ follows
  - In each interpretation where all the premises $\phi_1, \ldots, \phi_k$ are true, so is the conclusion ψ
  - Every model of $\phi_1, \ldots, \phi_k$ is a model of ψ

The sentence 'snow is white' is true if, and only if, snow is white – Alfred Tarski

Validity Problem

- Check if if ϕ is valid, written $\models \phi$
- Look for a counterexample
- Predicate logic is semi-decidable – there is no guaranteed method to test validity, if the answer is 'No' the method may not terminate
- Predicate logic is axiomatisable

Validity Problem

- Check if if ϕ is valid, written $\models \phi$
- Look for a counterexample
- Predicate logic is semi-decidable – there is no guaranteed method to test validity, if the answer is 'No' the method may not terminate
- Predicate logic is axiomatisable
- Predicate logic has many useful decidable fragments:
  - Monadic predicate logic – adding a single binary predicate makes it undecidable
  - Two-variable fragment
  - Guarded fragment – closed under Boolean composition and guarded quantification

Example

- $\forall y(G(x, y) \to C(y))$
- $\exists y(G(x, y) \wedge C(y))$
- $\forall x(G(x) \to C(x))$

Guarded Fragment

Definition (The guarded fragment GF of predicate logic)

- $\top$ and $\bot$ are in GF
- If ϕ is an atomic formula, then ϕ is in GF
- GF is closed under Boolean composition $\wedge$, $\vee$, $\neg$ and $\rightarrow$
- If ϕ is in GF and $A(\bar{x})$ is an atomic formula for which every free variable of ϕ is among the arguments of A, then $\forall\bar{y}(A(\bar{x}) \rightarrow \phi)$ and $\exists\bar{y}(A(\bar{x}) \wedge \phi)$ are in GF for any sequence $\bar{y}$ of variables

Example

- $\forall x \forall y (R(x, y) \rightarrow \neg R(y, x))$ (asymmetry) ✓
- $\forall x \forall y (MWC(x, y) \rightarrow M(x, y) \wedge \exists z (C(z, x) \wedge C(z, y)))$ ✗

Exercise

- Translate the following statements to predicate logic:

  Anyone's aunt is a sister of their parent.
  Jane's aunts are her parents' sisters.

- Which of these are in the 2-variable fragment of predicate logic?
- Which are in the guarded fragment?

Exercise

- Translate the following statements to predicate logic:

  Anyone's aunt is a sister of their parent.
  Jane's aunts are her parents' sisters.

- Which of these are in the 2-variable fragment of predicate logic?
- Which are in the guarded fragment?

$$\forall x \forall y (A(x,y) \rightarrow \exists z (S(x,z) \wedge P(z,y)))$$

Exercise

- Translate the following statements to predicate logic:

  Anyone's aunt is a sister of their parent.
  Jane's aunts are her parents' sisters.

- Which of these are in the 2-variable fragment of predicate logic?
- Which are in the guarded fragment?

$$\forall x \forall y (A(x,y) \rightarrow \exists z (S(x,z) \wedge P(z,y)))$$

$$\forall x (A(x, \text{jane}) \rightarrow \exists z (S(x,z) \wedge P(z, \text{jane})))$$

Exercise

- Translate the following statements to predicate logic:

  Anyone's aunt is a sister of their parent.
  Jane's aunts are her parents' sisters.

- Which of these are in the 2-variable fragment of predicate logic?
- Which are in the guarded fragment?

$$\forall x \forall y (A(x, y) \rightarrow \exists z (S(x, z) \wedge P(z, y)))$$

$$\forall x (A(x, \text{jane}) \rightarrow \exists z (S(x, z) \wedge P(z, \text{jane})))$$

Next up – Choosing a fragment of predicate logic to work with

Description Logics

- Family of knowledge representation languages for authoring ontologies
- Designed to represent knowledge about things, categories of things, and relationships between things and categories
- Foundation for the Web Ontology Language (OWL), built on W3C Resource Description Framework (RDF) standard for objects
- Good tradeoff between expressiveness and complexity of reasoning
- Amenable to implementation

From predicate to description logic

Description Logics are syntactic variants of (usually) decidable fragments of predicate logic:

- Only unary and binary predicates
- All variables are hidden from the syntax
- No free variables
- Restricted use of quantifiers, mostly within GF
- Only universal sentences as terminological axioms

Concept building blocks

- Atomic building blocks:
  - Individual names I (objects), e.g. tintin, tibet
  - Concept names C (classes), e.g. Dog, Country
  - Role names R (relations), e.g. owns, travelsTo

Concept building blocks

- Atomic building blocks:
  - Individual names I (objects), e.g. tintin, tibet
  - Concept names C (classes), e.g. Dog, Country
  - Role names R (relations), e.g. owns, travelsTo
- Boolean constructors:
  - conjunction: $\sqcap$ (class intersection)
  - disjunction: $\sqcup$ (class union)
  - negation: $\neg$ (class complement)

Concept building blocks

- Atomic building blocks:
  - Individual names I (objects), e.g. tintin, tibet
  - Concept names C (classes), e.g. Dog, Country
  - Role names R (relations), e.g. owns, travelsTo
- Boolean constructors:
  - conjunction: $\sqcap$ (class intersection)
  - disjunction: $\sqcup$ (class union)
  - negation: $\neg$ (class complement)
- Role restrictions:
  - existential restriction: $\exists$ (some)
  - universal restriction: $\forall$ (only)

Concept building blocks

- Atomic building blocks:
  - Individual names I (objects), e.g. tintin, tibet
  - Concept names C (classes), e.g. Dog, Country
  - Role names R (relations), e.g. owns, travelsTo
- Boolean constructors:
  - conjunction: $\sqcap$ (class intersection)
  - disjunction: $\sqcup$ (class union)
  - negation: $\neg$ (class complement)
- Role restrictions:
  - existential restriction: $\exists$ (some)
  - universal restriction: $\forall$ (only)
- and more: cardinality constraints, inverse roles, role composition, ...

Building complex concepts

Definition (Concept language of $\mathcal{ALC}$)

- Every concept name $A \in \mathsf{C}$ is a concept
- $\top$ and $\bot$ are concepts
- If C and D are concepts, then so are $\neg C$, $C \sqcap D$, $C \sqcup D$
- If $r \in \mathsf{R}$ and C is a concept, then $\forall r.C$ and $\exists r.C$ are concepts

Which of the following are concepts?

- $\top \sqcap r.\top$
- $\exists r.\top$
- $C \sqcup \neg\neg\exists D$
- $\exists r.\forall s.C \sqcap D$
- $\forall r.(C \sqcap \neg D)$

Building complex concepts

Definition (Concept language of $\mathcal{ALC}$)

- Every concept name $A \in \mathsf{C}$ is a concept
- $\top$ and $\bot$ are concepts
- If C and D are concepts, then so are $\neg C$, $C \sqcap D$, $C \sqcup D$
- If $r \in \mathsf{R}$ and C is a concept, then $\forall r.C$ and $\exists r.C$ are concepts

Which of the following are concepts?

- $\top \sqcap r.\top$ ✗
- $\exists r.\top$
- $C \sqcup \neg\neg\exists D$
- $\exists r.\forall s.C \sqcap D$
- $\forall r.(C \sqcap \neg D)$

Building complex concepts

Definition (Concept language of $\mathcal{ALC}$)

- Every concept name $A \in \mathsf{C}$ is a concept
- $\top$ and $\bot$ are concepts
- If C and D are concepts, then so are $\neg C$, $C \sqcap D$, $C \sqcup D$
- If $r \in \mathsf{R}$ and C is a concept, then $\forall r.C$ and $\exists r.C$ are concepts

Which of the following are concepts?

- $\top \sqcap r.\top$ ✗
- $\exists r.\top$ ✓
- $C \sqcup \neg\neg\exists D$
- $\exists r.\forall s.C \sqcap D$
- $\forall r.(C \sqcap \neg D)$

Building complex concepts

Definition (Concept language of $\mathcal{ALC}$)

- Every concept name $A \in \mathsf{C}$ is a concept
- $\top$ and $\bot$ are concepts
- If C and D are concepts, then so are $\neg C$, $C \sqcap D$, $C \sqcup D$
- If $r \in \mathsf{R}$ and C is a concept, then $\forall r.C$ and $\exists r.C$ are concepts

Which of the following are concepts?

- $\top \sqcap r.\top$ ✗
- $\exists r.\top$ ✓
- $C \sqcup \neg\neg\exists D$ ✗
- $\exists r.\forall s.C \sqcap D$
- $\forall r.(C \sqcap \neg D)$

Building complex concepts

Definition (Concept language of $\mathcal{ALC}$)

- Every concept name $A \in \mathsf{C}$ is a concept
- $\top$ and $\bot$ are concepts
- If C and D are concepts, then so are $\neg C$, $C \sqcap D$, $C \sqcup D$
- If $r \in \mathsf{R}$ and C is a concept, then $\forall r.C$ and $\exists r.C$ are concepts

Which of the following are concepts?

- $\top \sqcap r.\top$ ✗
- $\exists r.\top$ ✓
- $C \sqcup \neg\neg\exists D$ ✗
- $\exists r.\forall s.C \sqcap D$ ✓
- $\forall r.(C \sqcap \neg D)$

Building complex concepts

Definition (Concept language of $\mathcal{ALC}$)

- Every concept name $A \in \mathsf{C}$ is a concept
- $\top$ and $\bot$ are concepts
- If C and D are concepts, then so are $\neg C$, $C \sqcap D$, $C \sqcup D$
- If $r \in \mathsf{R}$ and C is a concept, then $\forall r.C$ and $\exists r.C$ are concepts

Which of the following are concepts?

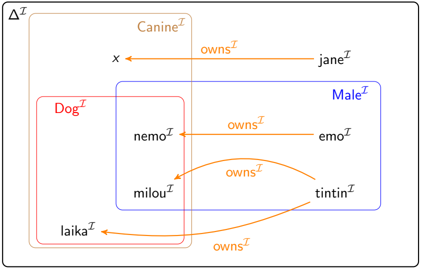
- $\top \sqcap r.\top$ ✗
- $\exists r.\top$ ✓
- $C \sqcup \neg\neg\exists D$ ✗
- $\exists r.\forall s.C \sqcap D$ ✓
- $\forall r.(C \sqcap \neg D)$ ✓

Concept language semantics

Definition (Interpretation)

An $\mathcal{ALC}$ interpretation $\mathcal{I}$ consists of a nonempty domain Δ, and an interpretation function $\cdot^{\mathcal{I}}$ such that:

- for each individual name $a \in \mathsf{I}$, $a^{\mathcal{I}} \in \Delta$
- for each concept name $A \in \mathsf{C}$, $A^{\mathcal{I}} \subseteq \Delta$
- for each role name $r \in \mathsf{R}$, $r^{\mathcal{I}} \subseteq \Delta \times \Delta$



Concept language semantics

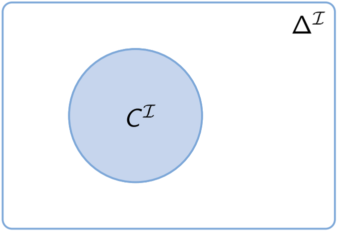
Extend interpretations to complex concept expressions:

- $\top^{\mathcal{I}} = \Delta$
- $\bot^{\mathcal{I}} = \emptyset$
- $(\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $$\begin{aligned}(\exists r.C)^{\mathcal{I}} &= \{x \in \Delta \mid x \text{ is related by } r^{\mathcal{I}} \text{ to some element in } C^{\mathcal{I}}\} \\ &= \{x \in \Delta \mid \exists y (r^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y))\}\end{aligned}$$
- $$\begin{aligned}(\forall r.C)^{\mathcal{I}} &= \{x \in \Delta \mid x \text{ is related by } r^{\mathcal{I}} \text{ only to elements in } C^{\mathcal{I}}\} \\ &= \{x \in \Delta \mid \forall y (r^{\mathcal{I}}(x,y) \rightarrow C^{\mathcal{I}}(y))\}\end{aligned}$$

Concept language semantics

Arbitrary concept

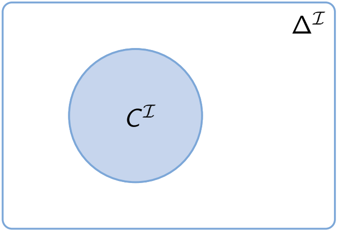
- A class in the domain
- $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$



Concept language semantics

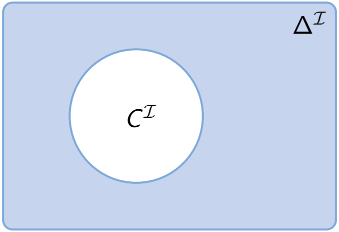
Arbitrary concept

- A class in the domain
- $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$



Concept negation

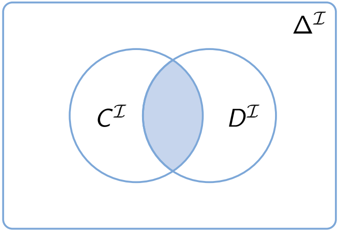
- The complement of a class
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$



Concept language semantics

Concept conjunction

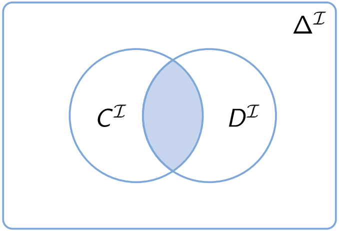
- The intersection of two classes
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$



Concept language semantics

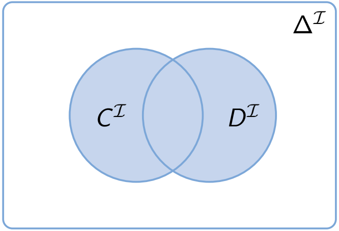
Concept conjunction

- The intersection of two classes
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$



Concept disjunction

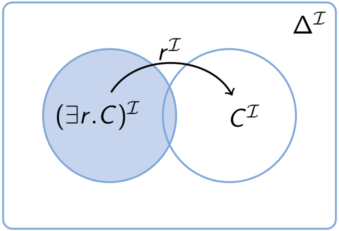
- The union of two classes
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$



Concept language semantics

Existential restriction

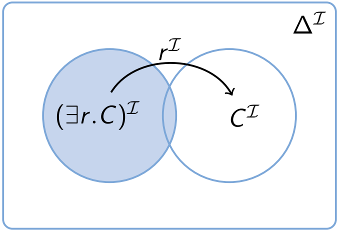
$$(\exists r.C)^{\mathcal{I}} = \{x \in \Delta \mid \exists y(r^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y))\}$$



Concept language semantics

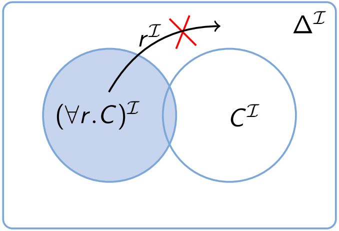
Existential restriction

$$(\exists r.C)^{\mathcal{I}} = \{x \in \Delta \mid \exists y(r^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y))\}$$

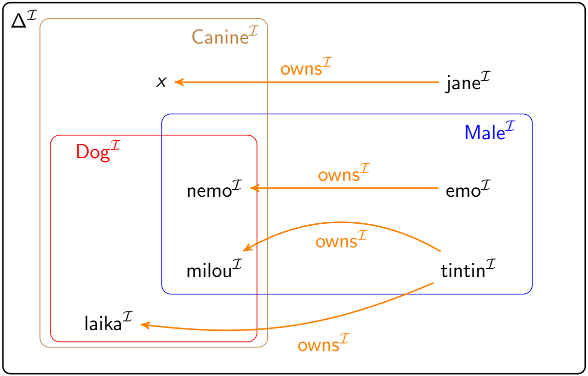


Universal restriction

$$(\forall r.C)^{\mathcal{I}} = \{x \in \Delta \mid \forall y(r^{\mathcal{I}}(x,y) \rightarrow C^{\mathcal{I}}(y))\}$$



Example



- $(\mathsf{Canine} \sqcap \mathsf{Male})^{\mathcal{I}} = \{\ldots\}$
- $(\exists \mathsf{owns}.(\mathsf{Canine} \sqcap \mathsf{Male}))^{\mathcal{I}} = \{\ldots\}$
- $(\neg \mathsf{Canine} \sqcap \forall \mathsf{owns}.\neg(\mathsf{Male} \sqcap \mathsf{Canine}))^{\mathcal{I}} = \{\ldots\}$

Mapping to predicate logic

- Same notion of domain of interpretation
- Individual names mapped to constant symbols
- Concept names mapped to unary atomic formulas
- Role names mapped to binary atomic formulas
- Complex concepts mapped to predicate formulas with a single free variable

Mapping to predicate logic

- Same notion of domain of interpretation
- Individual names mapped to constant symbols
- Concept names mapped to unary atomic formulas
- Role names mapped to binary atomic formulas
- Complex concepts mapped to predicate formulas with a single free variable

Some unanswered questions – how to

- relate or compare complex concepts?
- bind the hidden free variable in complex concepts?
- assert knowledge about individuals?
- determine the veracity of statements?

Making statements

- Terminological axioms (TBox axioms):
  - All owners of a male dog are male
  - In predicate logic:

$$\forall x(\exists y(\text{Owns}(x, y) \wedge \text{Male}(y) \wedge \text{Dog}(y)) \rightarrow \text{Male}(x))$$

  - This is a sentence in predicate logic, so no free variables !

Making statements

- Terminological axioms (TBox axioms):
  - All owners of a male dog are male
  - In predicate logic:

$$\forall x(\exists y(\mathrm{Owns}(x, y) \land \mathrm{Male}(y) \land \mathrm{Dog}(y)) \rightarrow \mathrm{Male}(x))$$

  - This is a sentence in predicate logic, so no free variables !
- Assertions (ABox assertions):
  - assert categories to which individuals or pairs of individuals belong
  - Nemo is a dog
  - Emo owns Nemo
  - In predicate logic:

$$\mathrm{Dog}(\mathrm{nemo}) \land \mathrm{Owns}(\mathrm{emo}, \mathrm{nemo})$$

Axioms

$$C \sqsubseteq D$$

- Concept inclusion / subsumption:
  - C is subsumed by D
  - C is more specific than D
  - D generalises C
  - All C are D
  - Every C is a D
- Formalises the syllogism "All ... are ..."
- Predicate logic translation:
  - Binds the free variable in C and in D
  - $\forall x(C(x) \rightarrow D(x))$

Example

$\exists \text{owns}.(\text{Dog} \sqcap \text{Male}) \sqsubseteq \text{Male}$

Assertions

$$a : C; \qquad (a, b) : r$$

- Concept and role assertions:
  - a is an instance of C
  - a and b are related by r
- Predicate logic translation:
  - No free variables
  - $C(a)$
  - $r(a, b)$

Example

$\text{nemo} : \text{Dog}$

$\text{nemo} : \neg\text{Male}$

$(\text{emo}, \text{nemo}) : \text{owns}$

Truth in an interpretation

Recall – An interpretation $\mathcal{I}$ is a tuple $\langle \Delta, \cdot^{\mathcal{I}} \rangle$ with domain Δ of objects and interpretation function $\cdot^{\mathcal{I}}$ such that:

- for each individual name $a \in \mathsf{I}$, $a^{\mathcal{I}} \in \Delta$
- for each concept name $A \in \mathsf{C}$, $A^{\mathcal{I}} \subseteq \Delta$
- for each role name $r \in \mathsf{R}$, $r^{\mathcal{I}} \subseteq \Delta \times \Delta$

Truth in an interpretation

Recall – An interpretation $\mathcal{I}$ is a tuple $\langle \Delta, \cdot^{\mathcal{I}} \rangle$ with domain Δ of objects and interpretation function $\cdot^{\mathcal{I}}$ such that:

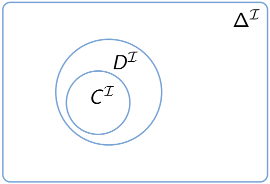
- for each individual name $a \in \mathsf{I}$, $a^{\mathcal{I}} \in \Delta$
- for each concept name $A \in \mathsf{C}$, $A^{\mathcal{I}} \subseteq \Delta$
- for each role name $r \in \mathsf{R}$, $r^{\mathcal{I}} \subseteq \Delta \times \Delta$

Definition (Satisfaction)

- $\mathcal{I} \Vdash C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (read: $\mathcal{I}$ satisfies $C \sqsubseteq D$)
- $\mathcal{I} \Vdash a : C$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $\mathcal{I} \Vdash (a, b) : r$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

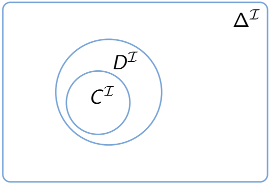
Truth in an interpretation

- $\mathcal{I} \Vdash C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

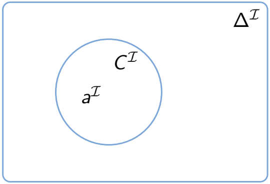


Truth in an interpretation

- $\mathcal{I} \Vdash C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

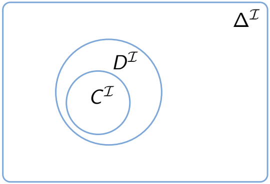


- $\mathcal{I} \Vdash a : C$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$

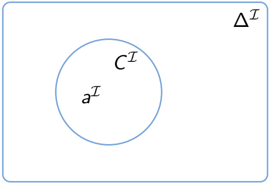


Truth in an interpretation

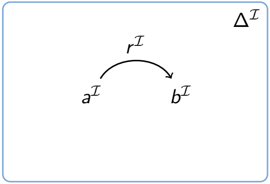
- $\mathcal{I} \Vdash C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$



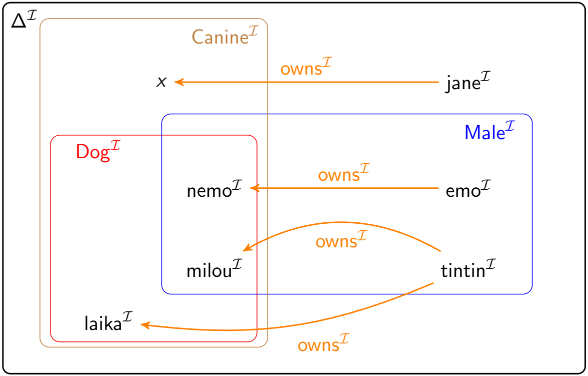
- $\mathcal{I} \Vdash a : C$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$



- $\mathcal{I} \Vdash (a, b) : r$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

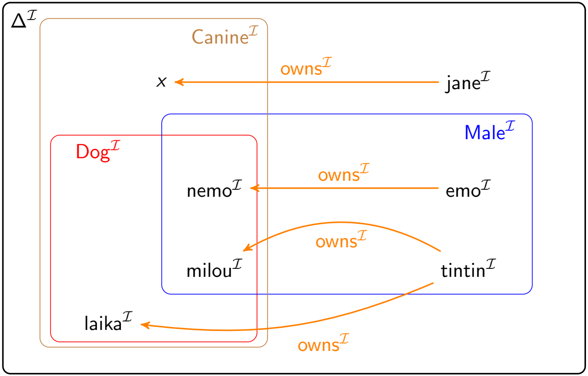


Example



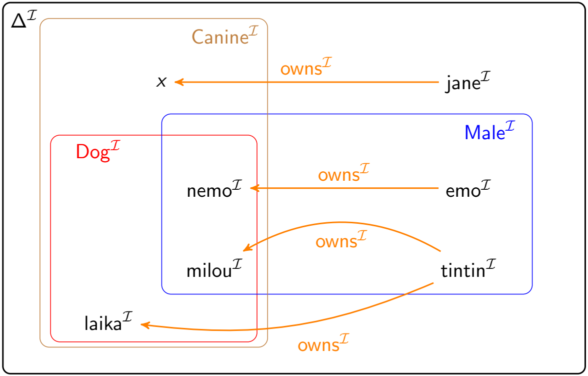
- $\mathcal{I} \models \mathrm{laika} : \mathrm{Canine}$?
- $\mathcal{I} \models \top \sqsubseteq \mathrm{Canine} \sqcup \exists \mathrm{owns}.\mathrm{Canine}$?
- $\mathcal{I} \models \exists \mathrm{owns}.(\mathrm{Male} \sqcap \mathrm{Canine}) \sqsubseteq \mathrm{Male}$?

Example



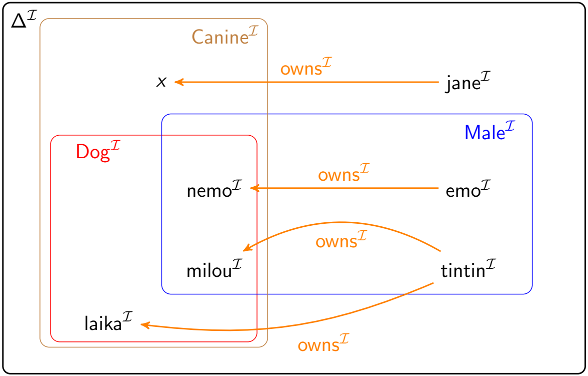
- $\mathcal{I} \models \mathsf{laika} : \mathsf{Canine}$? ✓
- $\mathcal{I} \models \top \sqsubseteq \mathsf{Canine} \sqcup \exists \mathsf{owns}.\mathsf{Canine}$?
- $\mathcal{I} \models \exists \mathsf{owns}.(\mathsf{Male} \sqcap \mathsf{Canine}) \sqsubseteq \mathsf{Male}$?

Example



- $\mathcal{I} \models \mathsf{laika} : \mathsf{Canine}$? ✓
- $\mathcal{I} \models \top \sqsubseteq \mathsf{Canine} \sqcup \exists \mathsf{owns}.\mathsf{Canine}$? ✓
- $\mathcal{I} \models \exists \mathsf{owns}.(\mathsf{Male} \sqcap \mathsf{Canine}) \sqsubseteq \mathsf{Male}$?

Example



- $\mathcal{I} \models \text{laika} : \text{Canine}$? ✓
- $\mathcal{I} \models \top \sqsubseteq \text{Canine} \sqcup \exists\text{owns}.\text{Canine}$? ✓
- $\mathcal{I} \models \exists\text{owns}.(\text{Male} \sqcap \text{Canine}) \sqsubseteq \text{Male}$? ✓

Satisfiability

- A knowledge base is a tuple $\mathcal{K} := \langle \mathcal{T}, \mathcal{A} \rangle$, for some TBox $\mathcal{T}$ and ABox $\mathcal{A}$
- Captures both structural knowledge about the domain and explicit knowledge about individuals

Definition (Satisfaction extended)

- $\mathcal{I} \Vdash \mathcal{T}$ if $\mathcal{I} \Vdash C \sqsubseteq D$ for each $C \sqsubseteq D \in \mathcal{T}$
- $\mathcal{I} \Vdash \mathcal{A}$ if $\mathcal{I} \Vdash a : C$ for each $a : C \in \mathcal{A}$ and $\mathcal{I} \Vdash (a, b) : r$ for each $(a, b) : r \in \mathcal{A}$

- If $\mathcal{I} \Vdash \mathcal{T} \cup \mathcal{A}$, we say $\mathcal{I}$ is a model of $\mathcal{K} := \langle \mathcal{T}, \mathcal{A} \rangle$
- $\mathcal{K}$ is consistent if it has a model

Reasoning

- Each TBox axiom $C \sqsubseteq D$ is either true or false in $\mathcal{I}$
- Each assertion $a : C$ and $(a, b) : r$ is either true or false in $\mathcal{I}$
- Each $\mathcal{I}$ is a complete description of what is true and false in $\mathcal{I}$

But ...

See first, think later, then test. But always see first. Otherwise you will only see what you were expecting – Douglas Adams

Reasoning

- Each TBox axiom $C \sqsubseteq D$ is either true or false in $\mathcal{I}$
- Each assertion $a : C$ and $(a, b) : r$ is either true or false in $\mathcal{I}$
- Each $\mathcal{I}$ is a complete description of what is true and false in $\mathcal{I}$

But ...

- There are infinitely many different interpretations
- The domain of interpretation $\Delta^{\mathcal{I}}$ can itself be infinite

See first, think later, then test. But always see first. Otherwise you will only see what you were expecting – Douglas Adams

Reasoning

- Open World Assumption
  - "lack of knowledge to the contrary does not imply falsity"
  - reasoning across all interpretations of a knowledge base

Definition (Entailment)

An axiom or assertion α follows from a knowledge base $\mathcal{K}$, written $\mathcal{K} \models \alpha$, if every model of $\mathcal{K}$ is also a model of α.

- Consistency – does $\mathcal{K}$ have a model?
- Entailment – does the axiom α follow from $\mathcal{K}$?
- Instance checking – does the assertion α follow from $\mathcal{K}$?

Entailment

Example

What follows from $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$[2]?

$$\mathcal{T} = \left\{ \begin{array}{l} \mathsf{EmployedStudent} \equiv \mathsf{Student} \sqcap \mathsf{Employee}, \\ \mathsf{Student} \sqcap \neg\mathsf{Employee} \sqsubseteq \neg\exists \mathsf{pays}.\mathsf{Tax}, \\ \mathsf{EmployedStudent} \sqcap \neg\mathsf{Parent} \sqsubseteq \exists \mathsf{pays}.\mathsf{Tax}, \\ \mathsf{EmployedStudent} \sqcap \mathsf{Parent} \sqsubseteq \neg\exists \mathsf{pays}.\mathsf{Tax}, \\ \exists \mathsf{worksFor}.\mathsf{Company} \sqsubseteq \mathsf{Employee} \end{array} \right\} \mathcal{A} = \left\{ \begin{array}{l} \mathsf{ibm} : \mathsf{Company}, \\ \mathsf{mary} : \mathsf{Parent}, \\ \mathsf{john} : \mathsf{Student}, \\ (\mathsf{john}, \mathsf{ibm}) : \mathsf{worksFor} \end{array} \right\}$$

- $\mathcal{T} \models \mathsf{Student} \sqcap \exists \mathsf{worksFor}.\mathsf{Company} \sqsubseteq \mathsf{EmployedStudent}$
- $\mathcal{T} \not\models \mathsf{Employee} \sqsubseteq \exists \mathsf{worksFor}.\mathsf{Company}$
- $\mathcal{K} \models \mathsf{john} : \mathsf{EmployedStudent}$
- $\mathcal{K} \not\models \mathsf{mary} : \neg\exists \mathsf{pays}.\mathsf{Tax}$

---

[2] Example adapted from ESLLI 2018 tutorial by Ivan Varzinczak

Query Answering

- For which individuals does the query answer α follow from $\mathcal{K}$?

$$q(x) := \mathsf{EmployedStudent}(x) \land \neg \mathsf{Parent}(x)$$

- Conjunctive queries

$$q(x,y) := \mathsf{EmployedStudent}(x) \land \mathsf{worksFor}(x,y) \land \mathsf{Company}(y)$$

$$q(x) := \exists y(\mathsf{EmployedStudent}(x) \land \mathsf{worksFor}(x,y) \land \mathsf{Company}(y))$$

- First-order queries

$$q(x,y) := \mathsf{EmployedStudent}(x) \land \mathsf{worksFor}(x,y) \land \mathsf{Company}(y)$$