Last time: Rules and rule-based systems

- General form of a rule: IF **antecedent** THEN **consequent**
- **Rule-based systems** consist of
  - a knowledge base (memory of rules)
  - a database (memory of facts)
  - an inference engine (for matching and evaluating expressions, and reasoning over rules)
  - optionally an explanation engine
  - User interfaces – for knowledge engineers and end users
- Inference - Forward chaining: What are all known facts?
- Validation - Backward chaining: Can X be inferred?

Today…

Deeper dive into

- General knowledge vs. Facts
- From causal relationships (rules) to an (object-oriented) view on how we could represent „the world“ symbolically
  - Basics of organising general knowledge and facts into „groups of similar things“ - classes/concepts/frames/schemata

3 – Logic: Facts, general knowledge and object-oriented knowledge representation

Institute of Interactive Systems and Data Science

Viktoria Pammer-Schindler

WISSEN
TECHNIK
LEIDENSCHAFT

Learning goals

- Explain and discuss what symbolic knowledge representation means, and assumptions underlying it
- Understand the difference between general knowledge and facts, and be able to give examples for both
- Understand predicate logic, frames, object-oriented programming, entity-relationship modelling and relational databases as different operationalisations of object-oriented knowledge representation
- Understand core modelling constructs in object-oriented knowledge representation - inheritance, generalization, aggregation, association, and type (domain and range) constraints.
- Reason with core modelling constructs

What do the following statements have in common?

- „1+1=2“
- „Singapore is a city“
- „Graz is in Vienna“
- “Alice and Bob were married in Graz on Nov 8, 2015”

They are **facts**. They are statements (assertions) about single, concrete entities.

What do the following logic statements have in common?

- Every human being is intelligent

$$Human(x) \rightarrow intelligent(x)$$

- Every city has inhabitants

$$City(x) \rightarrow \exists y: livesIn(y, x)$$

- No city is part of another city.

$$City(x) \rightarrow \sim\exists y: City(y) \wedge isPartOf(x, y)$$

They express **general knowledge**. They are statements about concepts and relations between the concepts.

Facts vs. general knowledge

Fact = statement about instances, assertional axiom

Set of facts = database, assertional knowledge base

General knowledge statement = statement about concepts/classes/frames, terminological axiom, (production) rule

Set of general knowledge statements **=** knowledge base, terminological knowledge base, set of rules

Propositional Logic

- Propositions (have truth values) A, B, C, …
- Logic operators & (AND), | (OR), ~(NOT), → (IMPLIES)
  - Propositions assert facts (if variables are instantiated)
  - A → B can be written as: If A then B. We used such rules in the last lecture to express general knowledge!

First-order Predicate Logic -> will be used today to illustrate object-oriented KR ideas

- Variables x,y,z, …
- *Variable domains D1, D2, …*
- N-ary predicates A(x), b(x,y), c(x,y,z), …
- Logic operators & (AND), | (OR), ~(NOT), → (IMPLIES)
- Quantors ∃, ∀

Elements of object-oriented knowledge representation

- Instantiations of variables ~ instance,, frame instance, entity
  - Examples: *Graz, Vienna, Italy, Viktoria, Mona Lisa, …*
  - Variables in expressions stand-in for instances in first-order predicate logic

- Unary predicate ~ Class, concept, frame, entity-type
  - *Human(x), City(x)*
  - Set of instances that share characeristics, a group of similar instances

- Binary predicate ~ Member variable, relationship, attribute, slot
  - *Likes(x,y)*

- N-ary predicates ~ difficult translation, mostly represented as class / instance with multiple attributes/slots/member variables
  - *Exam(x, y, z, w)*

Facts and general knowledge in object-oriented knowledge representation

Facts: Assertions about entity type/class or set membership/concept instantiation; and relationships

- *Human(Viktoria), Country(Italy)*
- *Teaches(Viktoria, IDSAI2021)*
- *Exam(Johannes, IDSAI2020, 2021-06-17, 3) (last could be the grade)*

General knowledge: Which concepts/entity types/classes exist, and how are they related to each other?

- *Human(x) -> Alive(x)*
- *Teaches(x,y) -> Human(x) AND Lecture(y)*

Core modelling constructs and reasoning

Reasoning over logic-based knowledge bases – which questions can we ask?

Consistency, satisfiability

- Do all the logic statements (=axioms) fit together? Is there any way (a model) to satisfy all axioms?

Inference

- Which statements follow from a given set of axioms? *Depending on logic, the number of inferences could be infinite.*

Validate

- Is X true, given a set of (assertional and terminological) axioms? (is X true in all models of the knowledge base)

Foundational relationships and reasoning in object-oriented KR formalisms - Overview

Relationship between instances and classes:

- **Instantiation / Inheritance**

Typical relationships between classes (unary predicates):

- **Generalization**
- **Part-Whole** relationships
- **Association** – relationships with a pre-defined name can exist between classes *(general object-oriented word for binary predicates)*
- ***Event, role*** – *conceptually important relationships in object-oriented KR*

Typical relationship between classes (unary predicates) and properties (binary or n-ary predicates):

- **Type** (domain and range) **constraints**.

Foundational relationships and reasoning in predicate logic

Instantiation / Inheritance: „x is of type C“, „x is a C“

- Reasoning: x inherits all characteristics of C (via the generalisation rules, see next slide)!
- It may be convenient to be able to override inherited values

In 1st order predicate logic:

$C(x)$: x is a variable, C is a unary predicate

Example:

$Canary(x)$: x is a Canary; x is of type „Canary“

Generalisation: „a C is a D“, „a C is a special kind of D“, „a C is something that…“

- Creates subsumption hierarchies
- Reasoning: C inherits all characteristics from D, P is true for all things that “are” C

In 1st order predicate logic:

$C(x) \rightarrow D(x)$: Where x is a variable, and C and D are unary predicates.

$C(x) \rightarrow P$: … where P is a logic statement in 1st order logic

Examples:

$Canary(x) \rightarrow Bird(x)$: A Canary is a Bird; A Canary is a special kind of Bird“

$Canary(x) \rightarrow canSing(x)$: A Canary is something that can sing.

Part-Whole Relationship: „Every C is part of a D"; „D consists of C", …

In first order predicate logic, multiple expressions are possible:

$$C(x) \rightarrow \exists y : hasPart(x, y) \wedge P(y)$$

$$C(x) \rightarrow \exists y : isPartOf(x, y) \wedge P(y)$$

Where x is a variable, C is a unary predicate, and P(y) are arbitrarily complex statements in which y occurs

- Aggregation: C consists of entities, entities can exist outside the aggregate
- Composition: C consists of entities, entities cannot exist outside the aggregate.

Part-Whole Relationship: „Every C is part of a D“; „D consists of C“, …

Examples:

$House(x) \rightarrow \exists y: hasRooms(x, y) \wedge Room(y)$: A house exists of rooms, strictly speaking we should add: $Room(x) \rightarrow \exists y: isPartOf(x, y) \wedge Building(y)$ to express that a room cannot exist

$Club(x) \rightarrow \exists y: hasMember(x, y) \wedge Human(y)$: A club has members, and the members of course exist without the club

Knowledge engineering perspective:

- Very widely varying formal implementation of part-whole relationships, often in knowledge engineering the consequences of a particular part-whole relationship need to be specificylly defined (e.g., deletion of dependent entities?, etc.)
- Cardinality is desirable, minimum/maximum values, typical values

Event

In first order predicate logic, multiple expressions are possible:

- Events modelled as n-ary predicates: *Event(id,date,place,list-of-participants, …)*
- or events modelled as objects with the corresponding relationships:

$\mathrm{Event}(x) \rightarrow \exists y: hasDate(x,y) \land Date(y)$

$\mathrm{Event}(x) \rightarrow \exists y: hasPlace(x,y) \land Place(y)$

Knowledge engineering perspective:

- Typical no formal implementatin of the idea of an „event“ exists in object-oriented KR formalisms, even though fundamentally important types of relationships in how humans understand the world (-> schema theory)

Role

Role: In first order predicate logic:

- Role as specific unary predicate: *Teacher(x), President(x), … - this mixes a bit what x intrinsically and unchangeable is, and what temporary roles x takes*
- Role as n-ary predicate: *Teacher(name,start-date,end-date,class, …)*

Knowledge engineering perspective:

- Typical no formal implementatin of the idea of a „role“ exists in object-oriented KR formalisms, even though fundamentally important types of relationships in how humans understand the world (-> schema theory)

Type constraints: p(x,y) implies that x is of type C, and y is of type D

In first-order predicate logic:

- Domain constraint: $p(x,y) \rightarrow P(x)$
- Range constraint: $p(x,y) \rightarrow Q(y)$

Where x,y are variables, p is a binary predicate, and P,Q are arbitrarily complex expressions in which x and y respectively occur as a variable

Examples:

$hasFoot(x,y) \rightarrow LivingBeing(x)$

$hasFoot(x,y) \rightarrow Foot(y)$

How do propositional and predicate logic relate to other knowledge representations?

Object-orientation as fundamental type of knowledge representation

Implemented in

- Propositional and predicate logic
- Frames
- Object-orientation as programming paradigm
- Entity-Relationship modelling for data modelling
- Relational databases for implementing object-oriented knowledge

Underlying assumption: That the world can usefully be represented around objects, ideal concepts, and relationships between those.

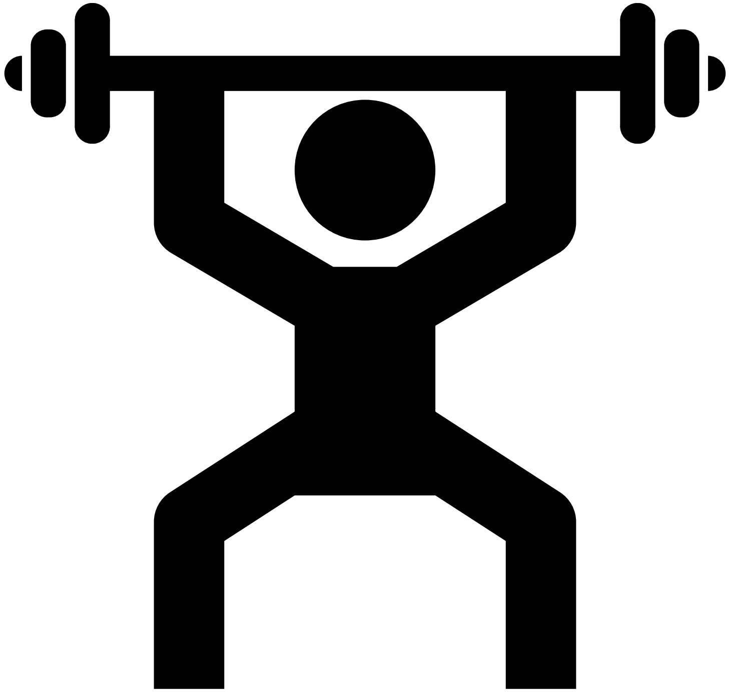
Discussion: Logic

- Logic is one of the oldest knowledge representation formalisms (several thousands years old)
- … and it has known mechanisms of producing valid inferences or chains of argumentation.
- Within AI, logic-based knowledge representation was the first, major line of attempts.
- Problems (see also lecture 2)
  - Limited expressive power for many real-world cases
  - Knowledge engineering may require large effort
  - ➢ What types of problems is logic well-suited to?

Discussion 2: Object-oriented KR

- Underlying assumptions
  - The cognition we are aware of as humans is symbolic; symbolic KR is therefore natural to humans (-> schema theory)
  - Goal is to develop rational (intelligent) systems
- Problems: Not all kinds of knowledge are well suited to be represented as objects (think procedures, mathematical models)
  - Shared agreement: Object-orientation is useful, but not for everything.
  - **Modern AI-based systems are often hybrid!**

Exercise 3



Exercise

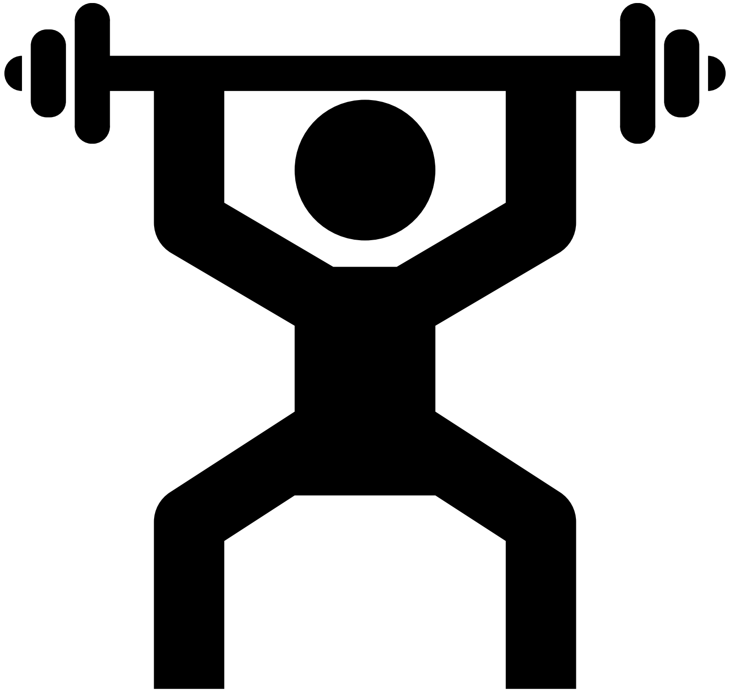
How can you express in predicate logic?

- Fitti is a Canary.
- Alice is the owner of Fitti.
- All canaries can fly.
- Some canaries have owners.

How would you express the following predicate logic statements in English?

- $Canary(x) \rightarrow Bird(x)$
- $eats(x, y) \rightarrow Food(y)$

Exercise 4a



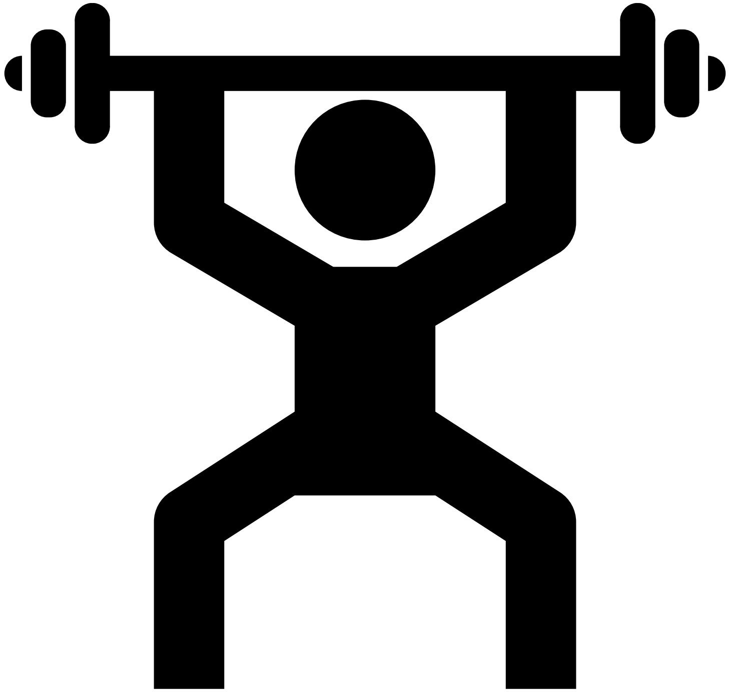
Exercise

Given is the knowledge base:

$$Canary(x) \rightarrow canFly(x) \wedge yellow(x) \wedge \exists y \in D: owns(y, x)$$
$$Canary(Fitti)$$

- Which statements can you infer?
- Is *yellow(Fitti)* true given the above knowledge base?
- Is *yellow(Pipsi)* true given the above knowledge base?

Exercise 4b



Exercise

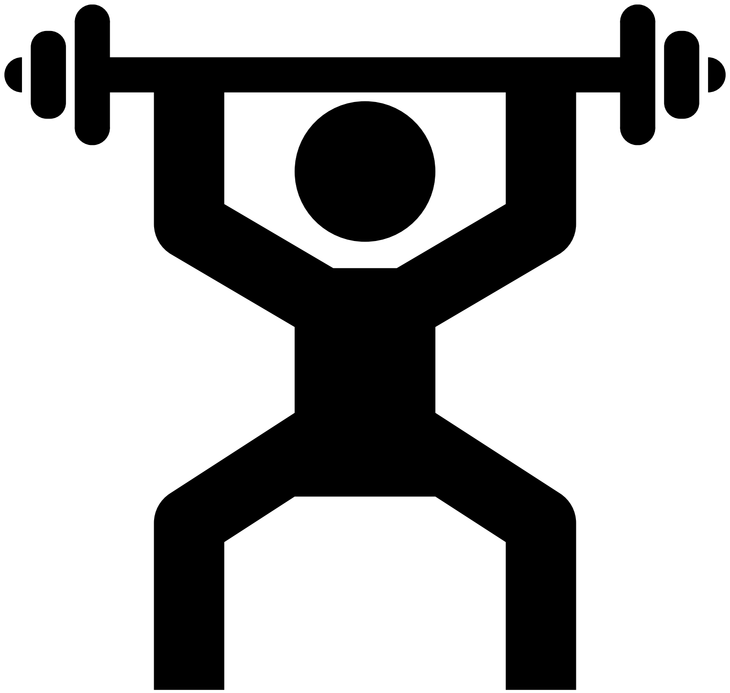
Example knowledge base:

$Canary(x) \rightarrow Bird(x)$

$Canary(Fitti)$

- Which statements can you infer?
- Is *Bird(Fitti)* true given the above knowledge base?
- Is *canFly(Fitti)* true given the above knowledge base?

Exercise 5



Exercise

Given is the following knowledge base:

$$Canary(x) \rightarrow \exists y{:}\, hasFoot(x, y) \wedge Foot(y)$$

$$Canary(Fitti)$$

$$hasFoot(Fitti, Foot2113789)$$

- Which statements can you infer?
- Can *Foot(Foot2113789)* be inferred from the above knowledge base?
- What statement could we add to the given knowledge base such that Foot(Foot2113789) can be inferred?

References

- Propositional Logic and Predicate Logic: http://www1.spms.ntu.edu.sg/~frederique/Teaching.html (scroll down to Discrete Mathematics, Chapters 2 and 3 of this course).
- Michael Negnevitsky. Artificial Intelligence: A Guide to Intelligent Systems. 2004.