Knowledge-based systems in Bioinformatics, 1MB602

Lecture 5:
Logic and knowledge representation

Lecture overview

- Part I
  - Knowledge bases
  - Propositional logic
  - First order logic
  - Definite Logic Programs
- Part II
  - Ontology-based knowledge representation
  - Ontology types
  - Knowledge representation languages
    - Frames
    - Description logic

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Knowledge bases

- A knowledge base is a set of representations of facts about the world
- Each individual representation is called a sentence
- The sentences are expressed in a knowledge representation (KR) language with defined syntax and semantics
- Requirements:
  - A way of adding new sentences to the KB
  - A way to query the KB what is known
  - When one asks a question to the KB, the answer should follow from what has been told to the KB previously
- Determining what follows from what the KB has been told is the job of the inference mechanism

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Knowledge-based agent

We can describe a KB agent at three levels:

1. Knowledge level (epistemological level)
   - The most abstract
   - We can describe the agent by saying what it knows
2. Logical level
   - The level at which the knowledge is encoded into sentences
3. Implementation level
   - The level that runs on the agent architecture
   - Physical representations of the sentences at the logical level

Epistemology or theory of knowledge is the branch of philosophy that studies the nature, methods, limitations, and validity of knowledge and belief

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

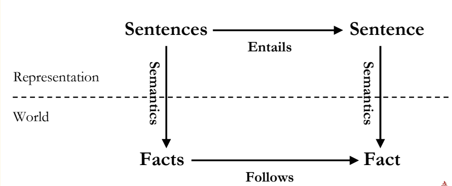
Knowledge-based agent cont.

- The semantics of the language determine the fact to which a given sentence refers (the meaning of the sentence)
- Because sentences are physical configurations of parts of the agent, reasoning must be a process of constructing new physical configurations from old ones
- New configurations should represent facts that actually follow from the facts that the old configurations represent

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Reasoning

The property of one fact following from some other facts is mirrored by the property of one sentence being entailed by some other sentences



THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Logic

A logic consists of:

1. A language with two aspects:
   a) the syntax, which describes how to make sentences
   b) the semantics, which states the systematic constraints on how sentences relate to affairs
2. A syntactic inference method (proof theory) – a set of rules for deducing the entailments of a set of sentences

Semantics

- The meaning of a sentence is what it states about the world
- In order to say what a sentence means, the writer has to provide an interpretation for it
- All representation languages impose a systematic relationship between sentences and facts
- In compositional languages the meaning of a sentence is a function of the meaning of its parts
- A sentence is true under a particular interpretation if the state of affairs it represent is the case
  - Truth depends on the interpretation of the sentence and on the actual state of the world

Examples of logics

Examples of formal languages and their ontological and epistemological commitments:

Language	Ontological commitment (What exists in the world)	Epistemological commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0…1
Fuzzy logic	degree of truth	degree of belief 0…1

Propositional Logic – Syntax

- Atomic propositions:
  - Any letter of the alphabet, e.g., P, or with a numeric subscript, e.g., Q_3
  - Any alphanumeric string, e.g., "Tom is the driver"
- Well-formed propositions (WFPs):
  - Atomic propositions
  - If P is a wfp, then so is ¬P
  - Any combination of two wfps with a binary logical connective:
    - ∧ (conjunction (and))
    - ∨ (disjunction (or))
    - ⇔ (equivalence)
    - ⇒ (implication)

order of precedence (highest to lowest): ¬, ∧ , ∨, ⇒, ⇔

Propositional Logic – Semantics

- The extensional semantics (values or denotations) of the expressions are relative to a particular interpretation, model, or situation (True or False)
- The intensions of the expressions is a statement in e.g. English
- Truth tables for the five connectives (denotations):

P	Q	¬P	P ∧ Q	P ∨ Q	P ⇒ Q	P ⇔ Q
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

$2^{\text{number of propositions}} = 2^2 = 4$ different situations

Propositional Logic - Semantic properties

- A wfp is:
  - satisfiable if it is True in at least one situation
  - contingent if it is True in at least one situation and False in at least one situation
  - valid if it is True in every situation (tautology)
  - contradictory if it is False in every situation

P	Q	¬P	Q ⇒ P	P ⇒ (Q ⇒ P)	P ∧ ¬P
True	True	False	True	True	False
True	False	False	True	True	False
False	True	True	False	True	False
False	False	True	True	True	False
		satisfiable, contingent	satisfiable, contingent	satisfiable, valid	contradictory

|= P ⇒ (Q ⇒ P) : Valid

Propositional Logic – Validity

- Check validity of the following complex sentence: $((P \lor H) \land \neg H) \Rightarrow P$
- Valid: True in every situation

P	H	P ∨ H	(P ∨ H) ∧ ¬H	((P ∨ H) ∧ ¬H) ⇒ P
True	True	True	False	True
True	False	True	True	True
False	True	True	False	True
False	False	False	False	True

$\models ((P \lor H) \land \neg H) \Rightarrow P$

Logical equivalences

$P \land (Q \lor R)$	$\Leftrightarrow$	$(P \land Q) \lor (P \land R)$
$P \lor (Q \land R)$	$\Leftrightarrow$	$(P \lor Q) \land (P \lor R)$
$\neg(P \land Q)$	$\Leftrightarrow$	$\neg P \land \neg Q$
$\neg(P \lor Q)$	$\Leftrightarrow$	$\neg P \land \neg Q$
$P \Rightarrow Q$	$\Leftrightarrow$	$\neg Q \Rightarrow \neg P$
$P \Rightarrow Q$	$\Leftrightarrow$	$\neg P \lor Q$
$P \Leftrightarrow Q$	$\Leftrightarrow$	$(P \Rightarrow Q) \land (Q \Rightarrow P)$
$P \Leftrightarrow Q$	$\Leftrightarrow$	$(P \land Q) \lor (\neg P \land \neg Q)$
$P \land \neg P$	$\Leftrightarrow$	False
$P \lor \neg P$	$\Leftrightarrow$	True

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

First Order Predicate Logic (FOPL)

- **Propositional logic has a very limited ontology, making only the commitment that the world consists of facts**
- **In FOPL, the world consists of** objects **with** properties **that distinguish them from other objects**
- **Among objects, various** relations **hold**

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

FOPL - Syntax

- **The language of FOPL consists of the following atomic symbols:**
  - Variables**: alphanumerical identifiers beginning with capital letters, e.g.** X, Y
  - Constants**: numerals or alphanumerical identifiers beginning with lowercase letters, e.g.** x, 17, john,…
  - Function symbols (functors)**: alphanumerical identifiers beginning with lowercase letters and with an associated arity > 0. Functor** f **with arity** n **is written as** f/n, **e.g.,** age(john)
  - Predicate symbols**: written as functors,** p/n, **e.g.** sisters(lisa, mary)
  - Logical connectives**: same as in propositional logic**
  - Quantifiers: $\forall$ **(universal) and** $\exists$ **(existential)**
  - Auxiliary symbols**: for example parentheses and commas**

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

FOPL - Syntax cont.

- **Terms**
  - **Any constant or variable is a term**
  - **If** f/n **is a functor and** $t_1,\ldots,t_n$ **are terms, then** $f(t_1,\ldots,t_n)$ **is a term**
- **Atomic formulas**
  - **If** p/n **is a predicate symbol and** $t_1,\ldots,t_n$ **are terms, then** $p(t_1,\ldots,t_n)$ **is an atomic formula**
- **Well-formed formulas (wff)**
  - **Any atomic formula is a wff**
  - **If** F **and** G **are wffs, then so are** $\neg F$, $F \land G$, $F \lor G$, $F \Rightarrow G$, **and** $F \Leftrightarrow G$
  - **If** F **is a wff and** X **is a variable, then** $\forall X\ F$ **and** $\exists X\ F$ **are wffs**
- **A formula which contains no variables is called** ground

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

FOPL - Semantics of wffs

- **A variable** X **is** free **in** P **if it is out of scope of any quantifier** ($\forall$, $\exists$) **over** X **in** P
- **Substitutions:**
  - **The result of applying a substitution to a wff** P, $P\{t_1/V_1,\ldots,t_n/V_n\}$, **is obtained by simultaneously replacing every occurence of each variable** V_i **in** P **by** t_i
- **The denotation (value) of** $\forall X\ P$ **is True if the denotation of** $P\{t/X\}$ **is true for every ground term** t
- **The denotation of** $\exists X\ P$ **is True if there is some ground term** t **such that the denotation of** $P\{t/X\}$ **is True**

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

FOPL - Semantic properties

- Same as with propositional logic
  - Satisfiability
  - Contingency
  - Validity
  - Contradictoriness

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Examples

- "Tom is the child of Mary"

`child_of(tom,mary)`

- "Mary has a child"
  - There exist an individual X such that X is the child of Mary

`∃X child_of(X,mary)`

- "Every cat is a mammal"
  - For any object X, if X is a cat then X is a mammal

`∀X cat(X) ⇒ mammal(X)`

- "Every parent loves her child"
  - For all individuals X and Y, if X is a parent and Y is the child of X then X loves Y

`∀X ∀Y ((parent(X) ∧ child_of(Y,X)) ⇒ loves(X,Y))`

Prolog

- Acronym for Programming in Logic
- Developed by A. Colmerauer and P. Roussel at the university of Aix-Marseille in 1971
- Designed for natural-language processing but has become one of the most widely used languages for artificial intelligence

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Rules

- "The grandchild of a person is a child of a child of this person"
  - For all X and Y, `grandchild(X,Y)` if there exists a Z such that `child_of(X,Z)` and `child_of(Z,Y)`

```
∀X ∀Y (grandchild(X,Y) ⇐
        ∃Z (child_of(X,Z) ∧ child_of(Z,Y)))
```

⇔ ... ⇔ (equivalence rules)

```
∀X ∀Y ∀Z (grandchild(X,Y) ⇐
        (child_of(X,Z) ∧ child_of(Z,Y)))
```

- General form (definite clause):

$A_0 \Leftarrow A_1 \wedge \ldots \wedge A_n$ or equivalently: $A_0 \vee \neg A_1 \vee \ldots \vee \neg A_n$

  - A_0 is called the head of the clause
  - $A_1 \wedge \ldots \wedge A_n$ is called the body

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Definite clauses

- A clause is a formula $\forall(A_1 \vee \ldots \vee A_n)$ where each A_i is an atomic formula or the negation of an atomic formula
- A definite clause is a clause that contains exactly one non-negative atomic formula

$$\forall(A_0 \vee \neg A_1 \vee \ldots \vee \neg A_n)$$

$A_0 \Leftarrow A_1 \wedge \ldots \wedge A_n$ $(n \geq 0)$ (implicit universal quantifier!)

- Definite program: finite set of definite clauses

```
gchild(X,Y) ⇐ child(X,Z) ∧ child(Z,Y)
child(john,mary) ⇐
child(mary,bob) ⇐
```

(the last two clauses: facts)

- A definite goal is a clause with no head

```
⇐ child(john,X) ∧ child(X,bob)
```

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Ontology-based KR

- Ontology?
  - "the concrete form of a conceptualisation of a community's knowledge of a domain" (Stevens et al. 2000)
  - "include[s] a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related" (Uschold et al., 1998)
  - "the specification of conceptualizations, used to help programs and humans to share knowledge" (Gruber, 1993)
    - Conceptualization: the knowledge about the world in terms of entities (things, the relations they hold and the constraints between them)
    - Specification: The concrete representation of this conceptualization (KR language)

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

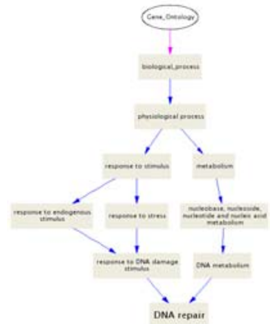
Ontology-based KR cont.

Knowledge base

ontology

- Conceptualization of an ontology
  - Concepts: a set or class of entities within a domain, e.g., `Protein`
    - Primitive concepts (necessary conditions)
      e.g. a globular protein is a kind of protein with a hydrophobic core
    - Defined concepts (necessary and sufficient conditions)
      e.g. eukaryotic cells are kinds of cells that have a nucleus
  - Relations: interactions between concepts
    - Taxonomies (organize concepts into a tree structure):
      'is a kind of'-relationships e.g. `Enzyme isA Protein`
      'part of'-relationships
      e.g. `Protein hasComponent ModificationSite`
    - Associative relationships: (relate concepts across tree structures)
      e.g. `Chromosome hasSubcellularLocation Nucleus`
  - Instances: things associated with a concept
    e.g. a gene annotated to a function

Example: Gene Ontology




```
goterm(GO:0006281, "DNA repair").
isa(GO:0006281, GO:0006259).
isa(GO:0006281, GO:0006974).
goterm(GO:0006974, "response to DNA damage stimulus").
isa(GO:0006974, GO:0006950).
isa(GO:0006974, GO:0009719).
goterm(GO:0006259, "DNA metabolism").
isa(GO:0006259, GO:0006139).
goterm(GO:0006950, "response to stress").
isa(GO:0006950, GO:0050896).
goterm(GO:0009719, "response to endogenous stimulus").
isa(GO:0009719, GO:0050896).
goterm(GO:0006139, "nucleobase, nucleoside, nucleotide
                    and nucleic acid metabolism").
isa(GO:0006139, GO:0008152).
goterm(GO:0050896, "response to stimulus").
isa(GO:0050896, GO:0007582).
goterm(GO:0008152, "metabolism").
isa(GO:0008152, GO:0007582).
goterm(GO:0007582, "physiological process").
isa(GO:0007582, GO:0008150).
goterm(GO:0008150, "biological_process").
isa(GO:0008150, top).
```

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Ontology use

- Domain-oriented
  - Domain specific (e.g. *E. Coli*)
  - Domain generalization (e.g. gene function)
- Task-oriented
  - Task specific (e.g. annotation analysis)
  - Task generalization (e.g. problem solving)
- Generic
  - Captures common high level concepts (`Structure` and `Substance`)

- Most bio-ontologies have a mixture of all three types

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Choosing a KR language

- Major considerations:
  - Expressivity:
    - A measure of the range of constructs that can be formally, flexibly, explicitly and accurately used to describe the components of an ontology
    - Trade-off between expressivity and complexity
  - Rigour:
    - A measure of the satisfiability and consistency of the representation within the ontology
    - Maintained computationally via logic-based systems
  - Semantics:
    - What the language mean
    - Clearly defined and well-understood semantics are essential

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Choosing a KR language cont.

- Three kinds of languages:
  - Vocabularies based on natural language
    - Very expressive
    - Simple tree-like inheritance structures
    - Difficulties with maintenance or preserving consistency
    - Example: language used in the Gene Ontology (GO)
  - Frame-based systems
    - Object-based (object-oriented)
    - Example: EcoCyc and RiboWeb
  - Description logic
    - Based on predicates expressed in logic
    - Trade-off between expressivity and tractability
    - Example: TaO

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Frames

- A frame usually represent a concept (or a class, collections of instances) and is defined by an identifier, and a number of elements called slots (or attributes)
  - Each slot correspond to an attribute that members of the class can have: values or other frames
  - Slots can be unfilled
  - Unfilled slots can be filled through inference (inheritance)
- Frames (can) include:
  - Frame identification information
  - Relationship of this frame to other frames
  - Descriptors of requirements for a frame
  - Procedural information on use of the structure described
  - Frame default information
  - New instance information
- Frames share a lot of properties with object-oriented modeling

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Description Logic

- Describe knowledge in terms of concepts and relations
- Concepts and relations are used to automatically derive taxonomies (i.e. classify concepts)
- Concepts are defined using other relations and concepts
- Example: Definition of the concept `Enzyme`
  - Concept: `Protein`
  - Concept: `Reaction`
  - Relation: `Catalyses`
  - Composite concept: `Protein which catalyses reactions`
- Enables dynamic ontology's with automatic consistency check

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

References

- S. Russell, P. Norvig, Artificial intelligence: a modern approach, Prentice-Hall, Upper Saddle River, New Jersey, 1995
- R. Stevens, C. A. Goble, S. Bechhofer, Ontology-based knowledge representation for bioinformatics, Briefings in Bioinformatics 2000 1(4): 398-414
- S. C. Shapiro, Propositional, First-Order And Higher-Order Logics: Basic Definitions, Rules of Inference, and Examples, in L. L.M. Iwanska, S. C. Shapiro, Eds., Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language, AAAI Press/The MIT Press, Menlo Park, CA, 2000, 379-395
- Francesco Donini, et al, Reasoning in Description Logics, in G. Brewka, Ed., Principles of Knowledge Representation and Reasoning, Studies in Logic, Language and Information, CLSI Publications, 1996

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se