Laurea MAGISTRALE in COMPUTER SCIENCE

ARTIFICIAL INTELLIGENCE

Representation / 1
Knowledge Representation and Conceptualization

Knowledge Representation

- The focus in Artificial intelligence is more on *knowledge* than on *data*
  - Facts
  - Procedures
  - Meanings
- Many possible representations
  - Choosing a wrong representation may lead to failure

Knowledge in Machines

- Can a machine have knowledge?
  - Yes, so far as it is provided in a useful form
    - Explicit or implicit
- A program is attributed *knowledge* in terms of:
  - Objects on which it acts
  - Reference domains with respect to which identifying and formulating actions
  - Ability to operate transformation on the input information in order to produce results
  - Indications of good solutions vs. “optimum” solutions, etc.

How to store knowledge?

- In people’s brains
  - Cognitive Psychology is still looking for an answer
- In Computer Science
  - How can we store facts? Can we translate such knowledge into *tractable* structures? Can we define and build operators suitable for processing such structures?
  - We may only use *symbols* to represent and manipulate knowledge
    - Use of strings and numbers to represent a entities in the real world, objects or ideas
      - Internal representation

How to process / manipulate knowledge?

- Still not possible to use natural language for representing knowledge
  - Need to establish at least a *correspondence* between *facts* and *internal representation*
- *Processing knowledge* and defining operators on the knowledge structures =
  ability to make inferences
- Inference
  - Process that from a set of facts – data accurately representing specific properties of the domain – and of rules – formal rules representing the dynamics of the domain – *derives new facts and rules*

How to make knowledge available to a computer?

- Making knowledge available to a computer = endowing it with an automatic mechanism to carry out inferences
- To provide knowledge in computable form one must
  - Define the domain
  - Decide about the vocabulary of predicates, functions and constants
  - Encode the general, shared knowledge on the domain (the ontology)
  - Encode a description of the specific instance of the problem
  - Provide queries to the inference procedure

Knowledge

- Features
  - Difficult to define and formalize
  - Voluminous
  - Changing
- Processing knowledge =
  - Acquiring it
    - Assimilate, classify, recognize, learn, adapt, correlate
  - Storing and Retrieving it
    - Determine what knowledge is relevant for a specific goal, in order to "optimize" storage and retrieval
    - Techniques for
      - *Linking* (connecting) information structures
      - *Lumping* (aggregating) information structures

Knowledge

- Processing knowledge = (cont.)
  - Using and updating it through:
    - Formal reasoning
      - Syntactic manipulation of data structures through predefined inference rules
    - Procedural reasoning
      - Use of simulation to answer questions and solve problems
    - Analogic reasoning
      - Extension of the properties observed in one domain to another domain
    - Generalization / Abstraction
      - Extension to general cases of properties observed in examples, concept synthesis
    - Meta-reasoning
      - Using knowledge about what one knows in order to improve one's reasoning

Knowledge Representation (KR)

- Several perspectives, e.g.:
  - A model that replaces the world, used to define the consequence of acting on the represented entities, by reasoning in the model rather than actually acting on the world
  - A set of (ontological) definitions answering the question "What are we allowed to think about the world?"
  - A fragmented theory of intelligent reasoning expressed in terms of concepts, sets of forbidden inferences, sets of recommended inferences
  - A means for a practically efficient computation
  - The explicitation of the human way of expressing oneself (the language used to talk about worldly matters)

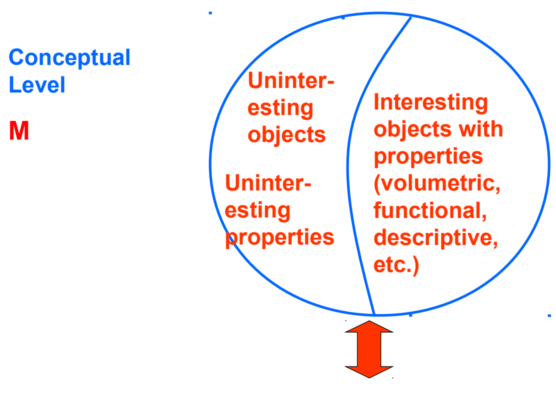
Knowledge Representation

- A good representation must
  - Highlight important objects and relationships
  - Provide effective overall visions
  - Suppress irrelevant details
  - Allow generalizations
  - Be understandable, complete, concise
  - Be usable even if knowlege is incomplete
  - Allow an easy and effective "manipulability" of knowledge
    - Easy information access, storage, retrieval and modification
  - Be COMPUTABLE ⟺ TRACTABLE

Knowledge Representation in Computer Science

- Representation = Knowledge + Access
  - Representation is a *system* able to handle *symbols* that allows to *access* the body of knowledge so as to choose the *actions* to attain some *goals*
- Access function
  - YES: a system for distributing knowledge encoded in a data structure
  - NO: a mere generator of a sequence of knowledge items

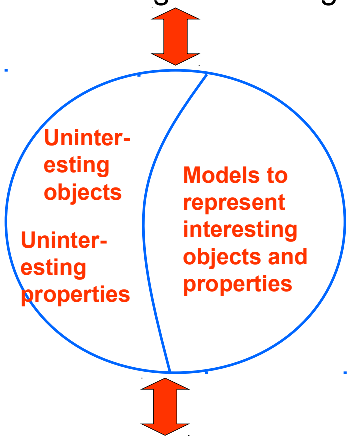
Modeling the World through Knowledge



Modeling the World through Knowledge

Representation Level

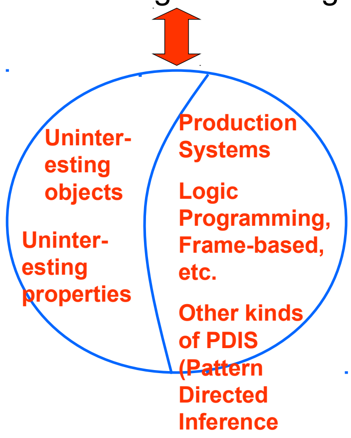
R



Modeling the World through Knowledge

Implementation Level

I



Knowledge Representation Schemes

- Procedural
  - Rule-based Systems
- Network
  - Semantic Networks
  - Conceptual Graphs
- Logical
  - First-order Logics
  - Higher-order Logics
- Structured
  - Frames
  - Scripts

Logics for Knowledge Representation

- Multi-disciplinary Subject
  - Logics
    - Provides the formal structure and inference rules
      - Without it, a knowledge representation is vague, with no clue to determine if claims are redundant or contradictory
  - Ontology
    - Defines the kind of things that exist in the application domain
      - Without it, terms are ill-defined and lead into confusion
  - Computation
    - Distinguishes knowledge representation from pure philosophy
      - Without computable models, logics and ontology could not be implemented in programs

Conceptualization and Representation

- Stored knowledge, descriptions, relationships
  - Consider two entities, X and Y
    - Symbols, numbers, actions or concepts
  - Sentence “attribute r of object X has value Y”
  - Can be modeled in various ways

Conceptualization and Representation

- Function
  - X and r are arguments of a function fY, that returns value of Y
    - Y = fY(X, r)
- Ordered triple
  - (A, O, V) “attribute A of object O has value V”
  - Domains of X and r must be *finite sets*
    - (attribute, object, value)
    - (r, X, Y)

Conceptualization and Representation

- Relationship
  - “X is related to Y through r”
  - Or, graphically
  - $X \xrightarrow{r} Y$
- Representation and manipulation of relationships among entities are fundamental to build knowledge-based systems
- The ability to mathematically characterize relationships and properties of relationships often leads to
  CONCISE and TRACTABLE
  representations

Conceptualization and Representation

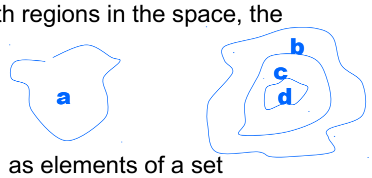
- Relationships and Properties
  - The concept of an entity that owns a property or an attribute is fundamental
    - E.g., we may represent the fact that
      - “Chairs have legs”
    - through relationship “has_legs” and the ordered pair (r,X)
      - (has_legs, chair)
    - or with the generic relationship “has” in which “legs” becomes a value (r, X, Y)
    - (has, chair, legs)

Conceptualization and Representation

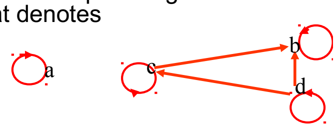
- Relationships among entities may be
  - Positional
  - Temporal
  - Procedural

Conceptualization and Representation

- Example: Spatial relationships
  - If we are dealing with regions in the space, the following regions



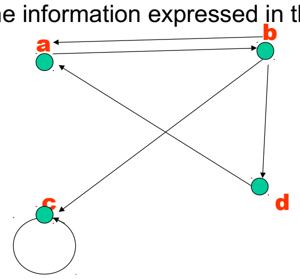
  - Can be represented as elements of a set
    - L = { a,b,c,d }
  - And we can define a relationship among the elements (a subset) that denotes
    - “Included in”
  - By using a *digraph*
    - Directed Graph



Conceptualization and Representation

- Binary and higher-order relationships may be described by *relational tables* that enumerate the relationships as ordered n-tuples in tabular form
  - The table contains the information expressed in the digraph

	a	b	c	d
a		1		
b	1		1	
c			1	
d	1			



Representing knowledge using Logics

- Knowledge formalization starts with a
  CONCEPTUALIZATION
  - Identification of objects that one presumes or assumes exist in the world and of their relationships
- Straightforward way to conceptualize: using natural language sentences
- *Universe of Discourse*
  - The set of objects about which knowledge is to be expressed
    - Objects may be concrete or abstract, primitive or composite

Conceptualization

- The process by which knowledge is expressed through logics
  - 3 kinds of concepts
    - Objects
      - Anything we want to talk about
    - Functions
      - Allow to associate objects to objects
    - Relationships
      - Link objects among them
  - Can be real, abstract or imaginary

Conceptualization

- A formal structure of (a piece of) reality as perceived and organized by an agent, independently of
  - Vocabulary used
  - Occurrence of a specific situation
- Different situations involving the same objects described by different vocabularies may share the same conceptualization
  - E.g., “apple” and “mela” have the same conceptualization
- Formalizing such knowledge allows us to share it with other (natural or artificial) systems

Conceptualization

- Formally, a triple < U, F, R>
  - U universe of discours
  - F set of basic functions
  - R set of basic relationships
    - Some relationships may be empty
      - *Unsupported* relationships

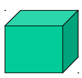
Conceptualization

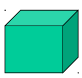
- For a finite universe of discourse there is a limit to the number of possible *n*-ary relationships il limited
  - For a universe of size *b* there are b^n distinct *n*-tuples
  - Each n-ary relationship is a subset of such *n*-tuples
- A function is a kind of relationship between objects in a universe of discourse

Conceptualization

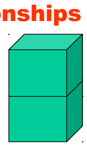
- Real world

Objects





Relationships



“A is above B”

- Logic world

Object symbols

A

B

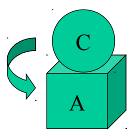
Predicates

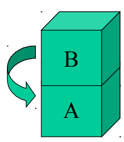
above(A, B)

Conceptualization

- Real world

Functions





Correspondence between the domain of what “is above” and that of objects

- Logic world

Functions

Functions

x = logic_function_above(A)

x → B **cube**

x → C **sphere**

Conceptualization

- Human cognitive activity strives to build and predict mental models of the world around us, and thus to organize them
  - This requires an ability to *group objects and ideas in conceptual categories* to which a name is assigned, that share peculiar features in a given context
- *Relational structuring of concepts* is the starting point to build knowledge and our understanding of the world around us

Conceptualization

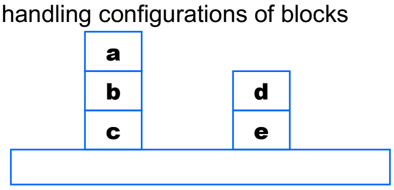
- Sources of categories:
  - Observation
    - Provides knowledge about the physical world
  - Reasoning
    - Gives meaning to observations, generating a set of so-called *meta-physical* abstraction
- Selection of categories determines what can be represented
  - Any incompleteness, distorsion, or restriction in the set of categories limits the generality of any representation of the world we make

Conceptualization

- One usually does not describe the whole world
  - The set of goals determines
    - Limitation to the necessary and sufficient set
    - The description's level of detail
      - Too specific: may make reasoning intricate
      - Too general: may make it impossible
- There is no single conceptualization for something
  - Depending on the chosen one, both tractability and clarity of reasoning may vary

Representation and Conceptualization

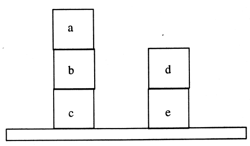
- Example: the "world of blocks"
  - Aim: handling configurations of blocks



  - In this case, the Universe of discourse is finite:
    - { a, b, c, d, e }
  - We may think of infinite universes
    - Sets of integers, of real numbers, ...

Conceptualization

- Example: Configurations of blocks
  - Objects
    - {a,b,c,d,e}
      - Corresponding to blocks
  - Functions
    - on_top/1 = {(b,a), (c,b), (e,d)}
      - Associates each block to the block on top of it
  - Relationships
    - above/2 = {(a,b), (a,c), (b,c), (d,e)}
      - Reports all objects which are above other objects
    - clear/1 = {a, d}
      - Reports the objects not having other objects above them



- < U, F, R > =
  < {a,b,c,d,e}, {on_top/1}, {above/2, clear/1} >

Conceptualization

- Example: Configurations of blocks (cont.)
  - Many other functions and/or relationships on blocks might have been defined
  - Also the table on which blocks are placed might have been considered as an object
- They have been considered as irrelevant for the given goals
  - The same objects might have been described in terms of groups of blocks or molecules that make up blocks
    - The proposed level of detail was considered to be the most appropriate

Conceptualization

- Example: Configurations of blocks (cont.)
  - Set of basic relationships
    - Binary relationship *above* : holds when two blocks are exactly one ontop of the other; in this way we highlighted the sequence of tuples that satisfies this relationship
      - { ‹a,b›, ‹b,c›, ‹d,e› }
    - Unary relationship *clear* : holds if a block has nothing on it; the selected elements are
      - { a, d }

- Example: Configurations of blocks (cont.)
  - Set of basic functions
    - Partial function *on_top* : maps a block onto the block ontop of it (if any)
      - { ‹b,a›, ‹c,b›, ‹e,d›}
    - One may think of purely theoretical functions, such as *rotation*, that maps blocks onto blocks based on the alphabetical ordering of labels
      - { ‹a,b›, ‹b,c›, ‹c,d›, ‹d,e›, ‹e,a› }

Conceptualization

- Some conceptualizations do not permit reasoning about properties of properties
- Solution: *reification*
  - Considering properties as objects, on which functions an relationships may be defined
    - Example
      - Conceptualization involving a set of blocks {a,b,c,d,e} and relationships {red, white, blue} that link each block to its corresponding color
      - Allows to consider colors of various blocks, but not the properties of such colors (if any)
      - Reifying colors = transforming them into objects, associate them to the related blocks by means of a *color* function and define on them the desired relationships (e.g., *warm*)

Reification

- Reifying functions and relationships as objects in the universe of discourse provides the advantage of considering properties of properties
  - The conceptualization proposed in the world of blocks allows us to consider colors of blocks but not properties
    - < {a,b,c,d,e}, { }, {ROSSO, BIANCO, BLU} >
  - We may handle this by reifying various relationships about colors, adding a partial function *color* to relate blocks to colors
  - We may add relationships characterizing the colors that now are objects in a universe of discourse
    - <{a,b,c,d,e, ROSSO, BIANCO, BLU}, {COLORE}, {CHIARO} >

Rules

- Can be used to explicitly represent knowledge about procedures and facts
  - Representing *what* and *how*
  - Antecedent (condition) ⟶ Consequent (action)
- In the consequent part of a production rule, rather than just an action, it will suffice entering
- IF (C) THEN (A)

(C)	(A)
A set of conditions	**A hypothesis to be checked, A set of conditions other than C (deducible), an inference**

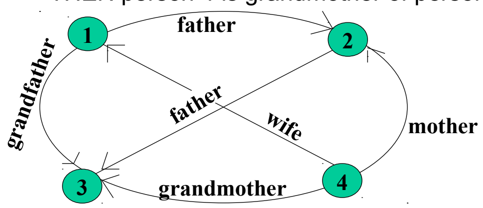
  - This allows us to represent reasoning through concatenation

Rules

- Typical rules
  - 1. IF a person 1 is father of person 2 and person 2 is father of person 3
    THEN person 1 is grandfather of person 3
  - 2. IF patient has diastolic pressure >90 and sistolic pressure >150
    THEN patient is hypertensive
  - 3. IF the first addend is number 2 and the second addend is number 3
    THEN the sum of the two addends is 5
  - 4. IF animal has wings and flies
    THEN is a bird
  - 5. IF animal is a mammal
    THEN (typically) it
    breaths air
    breastfeeds
    breeds by direct reproduction
    …

Rules

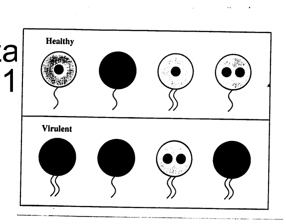
- Similar rules can be chained in a reasoning
  - IF a person 4 is wife of person 1
    AND person 1 is father of person 2
    THEN person 4 is mother of person 2
  - IF person 2 is father of person 3
    AND person 4 is mother of person 2
    THEN person 4 is grandmother of person 3



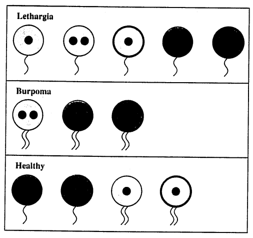
Conceptualization

- Other knowledge representation formalisms (other than rules)
  - Ontologies
  - Semantic nets
  - Frames
  - Scripts
  - Conceptual graphs (Sowa)
  - ...

Conceptualiza
Exercise 1



- Domain: biology
- Goal: distinguishing cells belonging to patients with various diseases
  - Healthy vs Virulent
  - Healthy vs Lethargia vs Burpoma



Conceptualization
Exercise 1

- Analysis
  - Different (types of) cells
  - Each cell characterized by 4 *features*:
    - Number of nuclei (1 or 2)
    - Number of tails (1 or 2)
    - Color (dark or light)
    - Wall (thick or thin)
  - Need to express the *class*
- Definition of vocabulary
  - Constants to identify cells
    - {c1, c2, c3, ...}

Conceptualization
Exercise 1

- Definition of vocabulary
  - Predicates to express features
    - Option 1 (no reification)
      - Predicates
        - nuclei(X, Y) = cell X has Y nuclei
        - tails(X, Y) = cell X has Y tails
        - dark(X) = color of cell X is dark
        - light(X) = color of cell X is light
        - thick(X) = wall of cell X is thick
        - thin(X) = wall of cell X is thin
        - healthy(X) = cell X is of class ‘healthy’
        - virulent(X) = cell X is of class ‘virulent’
          - lethargia(X) = cell X is of class ‘lethargia’
          - burpoma(X) = cell X is of class ‘burpoma’
      - Constants
        - {1, 2}

Conceptualization
Exercise 1

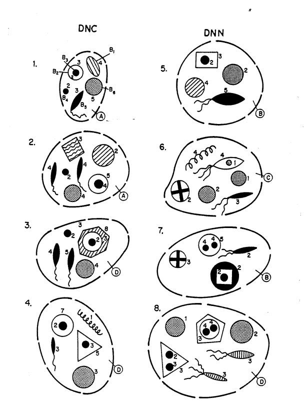
- Definition of vocabulary
  - Predicates to express features
    - Option 2 (reification)
      - Predicates
        - nuclei(X, Y) = cell X has Y nuclei
        - tails(X, Y) = cell X has Y tails
        - color(X, Y) = color of cell X is Y
        - wall(X, Y) = wall of cell X is of type Y
        - class(X, Y) = cell X is of class Y
      - Constants
        - { 1, 2,
          dark, light,
          thick, thin,
          healthy, virulent
          lethargia, burpoma }
    - Other options
      - Some properties reified, some not reified

Conceptualization Exercise 1

- Use of vocabulary to represent the world (option 2)
  - Sample cell description: cell c1 
    - nuclei(c1, 2).
    - tails(c1, 1).
    - color(c1, dark).
    - wall(c1, thick).
    - class(c1, burpoma).
  - Rule
    - class(X, burpoma) :-
      nuclei(X, 2), tails(X, 2), color(X, Y), wall(X, Z).
      - "a cell is of class burpoma if it has two nuclei and two tails"

Conceptualization Exercise 2

- A more complex case
  - Cancerous and non-cancerous cells



Conceptualization Exercise 2

- A possible formalization (with reification)
  - Predicates
    - class(X, Y) = cell X belongs to class Y
      - Constants: {dnc, dnn}
    - circ(X, Y) = wall of cell X consists of Y segments
      - Constants: {1, 2, ...}
    - pplasm(X, Y) = cell X has protoplasm of type Y
      - Constants: {A,B,C,D}
    - contains(X, Y) = cell X contains body Y
    - shape(X, Y) = body X has shape Y
      - Constants: { regular,
        oval,
        circle, ellipse,
        polygon,
        triangle, square, heptagon,
        irregular,
        spring, boat }

Conceptualization Exercise 2

- A possible formalization (with reification)
  - Predicates (cont.)
    - texture(X, Y) = body X has texture Y
      - Constants: {blank, shaded, solid_black, solid_grey, stripes, crossed, wavy}
    - weight(X, Y) = weight of body X is Y
      - Constants: {1,2, ...,5}
    - orient(X, Y) = body X has orientation Y
      - Constants: {N, NE, E, SE, S, SW, W, NW}
    - tail(X, Y) = body X has a tail Y

Conceptualization Exercise 2

- A possible formalization (with reification)
    - Note 1: The domain of shapes of bodies is structured
      - The user may specify just the specific values (the leaves of the tree) and the superclasses may be inferred automatically
  - Rules
    - shape(X, regular) :- shape(X, oval).
    - shape(X, regular) :- shape(X, polygon).
    - shape(X, oval) :- shape(X, circle).
    - …
  - Further notes:
    - Only boat-shaped bodies may have tails and orientation
    - A body may have many tails
    - Property ‘contains’ is transitive, which may be expressed by rule
      - contains(X, Y) :- contains (X, Z), contains(Z, Y).

Conceptualization Exercises

- Conceptualize the following domain
  - Computers (HW/SW) and other electronic devices
    - From single electronic components to composite systems
  - Publications
    - Multimedia documents and their contents