Semantic Nets

CPSC 433: Artificial Intelligence
Fall 2024

Jonathan Hudson, Ph.D.
Assistant Professor (Teaching)
Department of Computer Science
University of Calgary

August 8, 2024

UNIVERSITY OF CALGARY

Knowledge Representation

- Basis of each AI concept or system!
- Representation without processing makes no sense (therefore we started with knowledge processing)
- Same knowledge can be represented very differently:
  - Spectrum: computer friendly - human friendly
  - Levels of abstraction
  - Different views on problem
  - Different processing techniques

  Note: transformations are possible!

Syntax and Semantics

- Similar to programming languages, in knowledge representation we have to look at syntax and semantics of a representation approach
- Syntax: What symbols, data types, etc. are allowed; sorts, number of arguments (multiplicity) and so on?
  What symbols have special meaning (and therefore have to be used with this meaning in mind)?
- Semantics: What do the symbols mean, what has knowledge processing to accomplish?

☞ we have to look at both

Semantic Nets

Semantic Nets

- Developed to have (partial) graphical representation of predicate logic with special interpreted symbols
- First used to represent sentences in natural language
- Later abstracted to represent just meanings (Conceptual Dependency)

- Many different approaches
- Sometimes used for describing ontologies
- Often also coupled with a logic and the possibility to add formulas to description
- Models classes and instances

Basic data structures

- Nodes:
  describe concepts and instantiations (objects, actions)
- Arcs/links:
  describe dependencies,
  like isa, is-element, greater-than,…
  can be predefined and user-defined
- Modifiers:
  add constraints, roles, etc. to links

Semantics

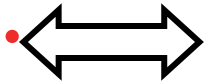
- Provide fixed interpretations for as many links as possible
- Provide fixed interpretations for modifiers
- User defined links require way to define their semantics
  (e.g. axioms in a logic with already defined semantics, or other descriptions)

Conceptual Dependency

Example: Conceptual Dependency (I)

- Actors: name or class name
- Actions (selection; including semantics)
  - ATRANS: Transfer of abstract relationship (give)
  - PTRANS: Transfer of physical location of object (go)
  - MOVE: Movement of body part by owner (kick)
  - INGEST: Ingesting of object by actor (eat)
  - MTRANS: Transfer of mental information (tell)
  - MBUILD: Building new information out of old (decide)

Example: Conceptual Dependency (II)

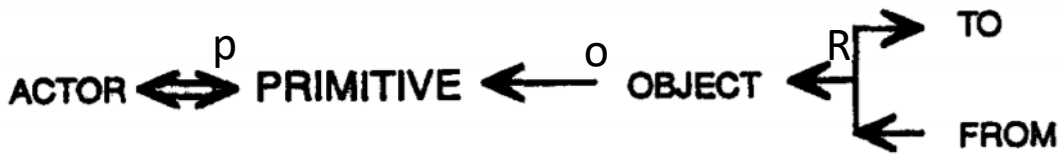
- Links:
  -  relation between actor and action
  -  indicates dependency and direction of it
- Modifiers (selection; including semantics):
  for relations between actor and action:
  - p : past tense
  - f: future
  - nil: present

  for dependencies:
  - o: object of an action
  - R: recipient of object

  + user-defined modifiers

Example: Conceptual Dependency (II)

- **Links**
  - relation between actor and action
  - indicates dependency and direction of it



- **Actions** (PRIMITIVE)
  - *ATRANS*: Transfer of abstract relationship (give) -> recepient
  - *PTRANS*: Transfer of physical location of object (go) -> recepient
  - *MOVE*: Movement of body part by owner (kick)
  - *INGEST*: Ingesting of object by actor (eat)
  - *MTRANS*: Transfer of mental information (tell) -> recepient
  - *MBUILD*: Building new information out of old (decide)

- **Modifiers**
  - **To link to action**
    - p : past tense
    - f: future
    - nil: present for dependencies:
  - o: object of an action
  - R: recipient of object

How to get knowledge into the representation structure

- Knowledge engineer should use as many predefined concepts, links and modifiers as possible in his/her graphs
- Knowledge engineer has to provide semantics (procedural, descriptive) for all user defined concepts, links and modifiers

Discussion

- Semantic nets express structure in a way also understandable by humans
- Easy to combine with other representation concepts
- Easily extendable

– Problem with how to express semantics for user-defined elements
– Some extensions are not decidable
– Often the predefined elements are not what we want for an application

And what about processing data?

- Answering questions:
  match question graph (with holes/variables) against graphs in knowledge base and return substitutions
  ☞ search (for best match)

- Adding to existing knowledge-base (classification):
  match new knowledge against old and add new graph parts (while checking fulfillment of constraints)
  ☞ search (for best fit)

- Other tasks:
  use provided procedures (based on semantics)
  for example: inference rules for conceptual dependency actions

Example

Examples

- Build a conceptual dependency representation for the following sentences:
  - John eats a steak
  - John ate pizza yesterday
- Build the graph for the following question and match it against the knowledge base from above:
  - Who had pizza yesterday?

Examples

- Build a conceptual dependency representation for the following sentences:
  - John eats a steak
  - John ate pizza yesterday
- Build the graph for the following question and match it against the knowledge base from above:
  - Who had pizza yesterday?
- Actors
- Actions
- Links
- Modifiers

Examples

- Build a conceptual dependency representation for the following sentences:
  - **John** eats a **steak**
  - **John** ate **pizza** yesterday
- Build the graph for the following question and match it against the knowledge base from above:
  - Who had **pizza** yesterday?
- Actors -> **John**, Steak, Pizza

Examples

- Build a conceptual dependency representation for the following sentences:
  - John **eats** a steak
  - John **ate** pizza yesterday
- Build the graph for the following question and match it against the knowledge base from above:
  - Who had pizza yesterday?
- Actors -> John, Steak, Pizza
- Actions -> **INGEST**

Examples

- Build a conceptual dependency representation for the following sentences:
  - **John eats a steak**
  - **John ate pizza** yesterday
- Build the graph for the following question and match it against the knowledge base from above:
  - Who had pizza yesterday?
- Actors -> John, Steak, Pizza
- Actions -> INGEST
- Links -> **john linked to object**

Examples

- Build a conceptual dependency representation for the following sentences:
  - John eats a steak
  - John ate pizza **yesterday**
- Build the graph for the following question and match it against the knowledge base from above:
  - Who had pizza yesterday?
- Actors -> John, Steak, Pizza
- Actions -> INGEST
- Links -> john linked to object
- Modifiers -> **yesterday**

Examples

- Build a conceptual dependency representation for the following sentences:
  - **John eats a steak**
  - John ate pizza yesterday
- Actors -> John, Steak, Pizza
- Actions -> INGEST
- Links -> john linked to object
- Modifiers -> yesterday

$$John \Leftrightarrow INGEST \leftarrow^{o} Steak$$

Examples

- Build a conceptual dependency representation for the following sentences:
  - John eats a steak
  - **John ate pizza yesterday**
- Actors -> John, Steak, Pizza
- Actions -> INGEST
- Links -> john linked to object
- Modifiers -> **yesterday**

$$John \Leftrightarrow INGEST \leftarrow^{o} Steak$$
$$John \Leftrightarrow^{\boldsymbol{p}} INGEST \leftarrow^{o} Pizza$$

Examples

- Build a conceptual dependency representation for the following sentences:
  - Who had **pizza** yesterday?
- Actors -> X
- Actions -> INGEST
- Links -> X linked to object
- Modifiers -> yesterday

$$X \Leftrightarrow^{\boldsymbol{p}} INGEST \leftarrow^{o} Pizza$$

Onward to … logic systems

Jonathan Hudson, Ph.D.
jwhudson@ucalgary.ca
https://cspages.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY