The CLARION Cognitive Architecture: A Tutorial

Part 3 – The Non-Action-Centered Subsystem

Ron Sun, Sébastien Hélie, Nick Wilson, Michael Lynch,
Cognitive Science, Rensselaer Polytechnic Institute

Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Formal properties and related theorems
8. Summary

Outline

1. **Representation**
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Formal properties and related theorems
8. Summary

Representation

Basic ideas of the NACS:

- The co-existence of and difference between explicit and implicit knowledge;
- The simultaneous involvement of implicit and explicit processes in most tasks;
- The redundant representation of explicit and implicit knowledge;
- The integration of the results of explicit and implicit processing;
- The iterative (and possibly bidirectional) processing.

(similar to the ACS, except the last one)

Representation

- Storing general knowledge about the world –‘**semantic’ memory**
- Storing specific experiences in the world -- **episodic memory**
- Performing various kinds of memory retrievals and inferences from such knowledge
- Under the control of the Action-Centered Subsystem usually (through its actions)
- Formed through acquiring and assimilating general knowledge, from external sources or from experiences (e.g., during action decision making, or reasoning)
- Therefore, the NACS includes long-term (and short-term) **semantic memory** and long-term (and short-term) **episodic memory**; the ACS includes long-term (and short-term) **procedural memory**.

Representation

Explicit semantic memory

Explicit episodic Memory

Associative Memory Networks (AMN) – Auto-associative

Associative Memory Networks (AMN) – Hetero-associative

Abstract Episodic Memory (AEM)

Representation

The top level:

- Encodes explicit, non-action-centered knowledge
- Chunk nodes encode concepts, linked to (micro)features at the bottom level – co-occurrences of (micro)features

  → a prototype model of concepts
- Links across chunk nodes encode explicit associations between chunks (concepts) –- unidirectional or bidirectional

  → associative rules

Representation

- Each chunk (concept) is represented by a chunk node in the top level
- Each chunk (concept) is represented by its (micro)feature nodes in the bottom level (distributed representation; more later)
- $Chunk\text{-}id_i \ (dim_{i_1}, val_{i_1})\ (dim_{i_2}, val_{i_2}) \ldots (dim_{i_n}, val_{i_n})$
  - e.g., table-1 *(type, table) (size, large) (color, white) (number-of-legs, 4)*
- *Chunk-id* may be externally given (if presented from external source) or generated (randomly) internally.

(essentially the same as the ACS)

Representation

- Explicit associative rules: Chunk nodes (denoting concepts) are connected at the top level by explicit associative rules
  - The condition of an associative rule contains one or more chunk nodes (different from the ACS).
  - The conclusion of an associative rule contains one chunk node.
- Modularity:
  - Similar to the ACS, the bottom level of the NACS can have multiple networks, each for a different kind of process.
  - (Correspondingly, the top level of the NACS can be divided into multiple rule groups.)

Representation

- Chunks may be activated:
  - As a result of receiving inputs (e.g., from the ACS).
  - By applying an associative rule (within the top level of the NACS).
  - From the result of an associative mapping at the bottom level of the NACS.
  - By similarity-based reasoning (through the bottom-level distributed representation and the top-bottom interaction in the NACS).

- The strength of a chunk in the top level (the chunk node) is determined by:

$$s_k^c = \max_x \left(s_k^{c,x} \right)$$

where s_k^c is the activation of chunk k in the top level and x is a particular activation source.

Representation

- Chunk nodes and associative rules in the top level: base-level activations (as in the ACS).
- Chunk nodes:

$$b_j^c = ib_j^c + c_c \sum_{l=1}^{n} t_l^{-d_c}$$

where ib_j^c is the initial BLA, c_c is the amplitude, d_c is the decay rate, and t_l is the time since the l^{th} use of the chunk.

- Associative rules:

$$b_j^r = ib_j^r + c_r \sum_{l=1}^{n} t_l^{-d_r}$$

where the same symbols are used except for r in place of c.

Representation

Questions?

Representation

The Bottom Level:

- Associative Memory Networks: encode non-action-centered implicit knowledge

  (e.g., BP networks (MLP trained with BP); or Hopfield networks; etc.).

- Each chunk (represented by a top-level chunk node): represented by a set of (micro)feature nodes at the bottom level (1 feature 1 node).

- Bottom-up activation through associative mapping
- Bottom-up activation through similarity-based processes

Representation

- Various possibilities of capturing implicit associations in the bottom level:
  - Hetero-associative:

    one set of nodes are presented as the input and another as the output to create an association between the two.

    E.g., MLPs trained with BP.
  - Auto-associative:

    observed nodes are set as both the input and desired output; for soft constraint satisfaction, pattern completion, etc.

    E.g., Hopfield networks. Some details ……
- These different ways of using the bottom level are implemented as separate modules (use as needed).

Representation

The process of **top-down activation** (Sun, 2003; Helie and Sun, 2010):

- When a chunk node is inferred (activated) in the top level but not in the bottom level, a top-down activation process may activate corresponding (micro)feature nodes in the bottom level.
- The activation of a (micro)feature node (in the bottom level) is set to the strength level of its corresponding chunk node.
- If the (micro)feature node receives activation from several chunk nodes, the maximum is used.

Representation

The process of **bottom-up activation** (Sun, 2003; Helie and Sun, 2010):

- When the result from the bottom level is sent bottom-up, it activates all chunk nodes compatible with it (i.e., with overlapping features). Weights later.
- A *Max* function is used to determine the overall strength of activated chunk nodes from bottom-up activation and from within the top level:

$$s_i^c = \max\left(s_i^{c,TL},\ s_i^{c,BL}\right)$$

where s_i^c is the activation of chunk i, $s_i^{c,TL}$ is the top-level activation of chunk i, and $s_i^{c,BL}$ is the bottom-up activation of chunk i.

Representation

Questions?

Outline

1. Representation
2. **Reasoning**
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Properties and related theorems
8. Summary

Reasoning

- Starts with an input to the NACS (by an ACS action)

  (input to the bottom level, the top level, or both)
- Activation levels of inputs are 1 (full activation).
- Bottom-up and top-down activation can then occur

  which ensures that full activation of all appropriate nodes occurs regardless of type of input provided.

Reasoning

Each round of NACS reasoning:

- At the bottom level, starts with the activated (micro)feature nodes (on the input side of the bottom level, if hetero-associative) for associative mapping. One round of associative mapping activates a set of (micro)feature nodes (on the output side if hetero-associative).
- At the top level, concurrently, an iteration of inference occurs starting from all the currently activated chunk nodes. All applicable associative rules fire simultaneously (there is no competition/selection among associative rules). New chunk nodes are inferred in the top level as a result.

Reasoning

- The outcomes of the bottom and top levels are integrated by bottom-up activation (mentioned before; details later);
- Another round of reasoning as above may take place

  based on (possibly filtered) results from the current iteration (multiple possibilities later). Top-down activation may activate the (micro)feature nodes of the newly activated (possibily filtered) chunk nodes (on the input side of the bottom-level networks)

Reasoning

- Similarity-based reasoning may be employed
  - During reasoning, a known (given or inferred) chunk may be *automatically* compared with another chunk. If the similarity between them is sufficiently high, then the latter chunk is inferred (activated).
- Mixed rule-based and similarity-based reasoning
  - Accounting for a large variety of commonsense reasoning patterns (including "inheritance reasoning" and beyond). See Sun (1994, 1995, 2003) and Sun and Zhang (2006).
  - Examples later

Reasoning

Reasoning modes at the top level of the NACS:

- Forward chaining reasoning:

  For drawing all possible conclusions in a forward direction – from known conditions to new conclusions (Smith, Langston, and Nisbett, 1992)

- Similarity-based forward chaining reasoning:

  For drawing all possible conclusions, using rules as well as similarity-based inferences (Sun, 1994, 1995, 2003; Sun and Zhang, 2006)

- In both cases, there is a threshold that determine whether a conclusion is acceptable or not.
- (By default, rule utility is not used in the NACS.)

Reasoning

- Rule-based reasoning:

$$s_j^a = \sum_i s_i^c \times w_{ij}^r$$

where s_j^a is the activation of rule *j*, s_i^c is the activation of premise chunk node *i*, and w_{ij}^r is the weight of premise chunk node *i* in the rule *j*.

- When several rules activate chunk node *j*, the maximum received activation is used:

$$s_{c_k}^{c,a} = \max_{j \text{Þ} c_k}\left(s_j^a\right)$$

where $s_{c_k}^{c,a}$ is the activation of a chunk node *k*, from RBR.

Reasoning

- Similarity-based reasoning (Tversky, 1977):

$$s_{c_j}^{c,s} = \max_i \left(s_{c_i \sim c_j} \times s_{c_i}^{c} \right)$$

where $s_{c_j}^{c,s}$ is the activation of chunk node c_j from SBR, $s_{c_i \sim c_j}$ is the similarity from chunks c_i to c_j, and $s_{c_i}^{c}$ is the total activation of chunk node c_i (from RBR, SBR, and whatever).

Reasoning

- The default similarity measure is (see Tversky, 1977; Sun, 1995):

$$s_{c_i \sim c_j} = \frac{\sum_{k \in c_i \cap c_j} V_k^{c_j} \times A_k}{f\left(\sum_{k \in c_j} V_k^{c_j} \times D_k\right)}$$

$$s_{c_i \sim c_j} = \frac{n_{c_i \cap c_j}}{f\left(n_{c_j}\right)}$$

(under some simplifying assumptions)

where $n_{c_i \cap c_j}$ is the number of features overlapping between chunks c_i and c_j, n_{cj} is the number of features in chunk c_j,

where A_k is the activation (1) of the k^{th} feature included in $c_i \cap c_j$, V_k^{cj} is the weight of the k^{th} feature, D_k is full activation (1), and $f(\bullet)$ is a slightly supralinear function.

- The similarity measure is bounded in the interval [0, 1); asymmetric

Reasoning: implementation of SBR

- Top-down activation: An activated chunk node at the top level of the NACS activates all its (micro)feature nodes (dimension-value pairs) at the bottom level. The top-down weight from the chunk node to its (micro)feature nodes is uniformly 1. So same activation/strength level.
- Bottom-up activation:

$$S_j^s = \sum_k \frac{v_k^{c_j}}{f\left(\sum_k v_k^{c_j}\right)} \times A_k$$

where c_j is a chunk, , A_k is the activation of its kth (micro)feature node, $\frac{v_k^{c_j}}{f\left(\sum_k v_k^{c_j}\right)}$ is the bottom-up weight of its kth (micro)feature node to its chunk node, and $S^s{}_j$ is the activation of the chunk node for c_j resulting from the bottom-up activation by all its (micro)features.

- A top-down and bottom-up activation cycle implements exactly the similarity measure discussed above.

Reasoning

The **reverse containment** principle (the idealized assumption):

- The (micro)feature representations of the NACS chunks: emulate an "ideal" categorical hierarchy; reasoning without explicit hierarchy
- If chunk *i* represents a category that is a superset (e.g., furniture) of the category represented by chunk *j* (e.g., table), the feature set of chunk *j* is a superset of the feature set of chunk *i* and more (i.e., $n_{c_i \cap c_j} = n_{ci}$).
- The above principle may not hold in less than ideal categories (e.g., some messier natural categories).

Reasoning

- The **reverse containment** principle (the idealized assumption): Will be used later in dealing with (implicit) conceptual hierarchies and inheritance reasoning
- But explicit conceptual hierarchies can be represented as well
- Explicit representation of hierarchies and implicit (feature-based) representations of hierarchies: interaction and synergy

Reasoning

- Mixing RBR and SBR (i.e., similarity-based forward chaining reasoning; Sun and Zhang, 2006):

$$s_{c_i}^{c} = \max\left(\alpha \times s_{c_i}^{c,a}, \beta \times S_{c_i}^{c,s}\right)$$

where $s_{c_i}^{c}$ is the final activation of chunk node c_i, α and β are scaling parameters for RBR and SBR respectively, $s_{c_i}^{c,a}$ is the activation of chunk node c_i from RBR, and $s_{c_i}^{c,s}$ is the activation of chunk node c_i from SBR.

- Such reasoning can be applied iteratively (outcomes from one round used as inputs for a new round)
- Pure RBR or pure SBR are special cases of the above (can also be iterated)

Reasoning

Questions?

Examples? Sloman simulation

Outline

1. Representation
2. Reasoning
3. **Learning**
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Properties and related theorems
8. Summary

Learning

- Learning in the top level
- Learning in the bottom level
- Top-down learning in the NACS
- Bottom-up learning in the NACS

Learning

Learning explicit knowledge:

- Encoding of externally given explicit knowledge (chunks or rules)
  - Under the control of the ACS (serves as a declarative memory)
  - (with a certain encoding probability)
- Extraction of explicit knowledge
  - Extraction from the bottom level of the ACS (like the top level of the ACS)
  - Extraction from the bottom level of the NACS (as in the ACS)
  - All experienced states, actions, and so on lead to chunks

Learning

Learning implicit knowledge (at the bottom level of the NACS)

- Training of the bottom-level networks (e.g., through the control of the ACS)
- Assimilation of explicit knowledge through training bottom-level networks (e.g., using explicit associative rules activated)
- At each step, a subset of items from episodic memory may be used to train the bottom level (with a certain selection probability)

Outline

1. Representation
2. Reasoning
3. Learning
4. **Coordination of the NACS and the ACS**
5. Episodic memory
6. Simulation examples
7. Properties and related theorems
8. Summary

Coordination of the NACS and the ACS

- Usually, the NACS is under control of the ACS

  → action-directed reasoning

- For instance,
  - An ACS action might be to perform a round of reasoning within the NACS
  - The ACS (or alternatively, the MCS) specifies the type of reasoning to be done in the NACS (e.g., RBR, SBR, RBR + SBR, and so on) and (possibly) associated parameters.

Coordination of the NACS and the ACS

- The outcome of reasoning in the NACS may be sent back to the ACS.
- If only one outcome from the NACS needs to be selected and sent back to the ACS, selection based on a Boltzmann distribution may be used (e.g., based on chunk activations).
- Alternatively, the ACS may choose to receive all outcomes from NACS reasoning or all outcomes of a certain type (e.g., outcomes that were not part of the input).

Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. **Episodic memory**
6. Simulation examples
7. Properties and related theorems
8. Summary

Episodic memory

- **Episodic memory**
  - EM stores specific past events (with time, location, and other episode-specific info) (Tulving, 1983):
    - Action-oriented experience
      - Action rules activated/used
      - State/action experienced,
      - State/action/result experienced
    - Non-action-oriented experience
      - Associative rules activated/used
      - Declarative chunks activated/selected
  - EM: encoding probability (so that not everything is remembered)
  - EM: recency-filtered (using BLA, with thresholding; so there is forgetting)

Episodic memory

- EM chunk: a top-level node; connected to bottom-level (feature-based) distributed representation
- The time stamp: a special feature
- EM may be used to help learning
  - EM stores all the associative rules applied, all the associations given externally, all the associations representing the mapping from the input to the NACS and each of the resulting (inferred) chunks.
  - EM stores all the action rules applied (along with circumstances, and results), etc.
  - Any of those can be selected for off-line training the ACS/NACS (especially the bottom level) ---- "memory rehearsal and consolidation"

Episodic memory

- **Abstract episodic memory** (at the bottom level)
  - AEM summarizes information of past episodes experienced by the ACS, instead of in individuated forms
  - Used to help learning also
  - Used to help with extracting explicit knowledge from the bottom level of the ACS)
  - AEM is constituted by
    - An action frequency network

      "State → Action" frequency distribution
    - A result frequency network

      "State, Action → next state" frequency distribution

      "State, Action → immediate reinforcement" frequency distribution

Episodic memory

- AEM networks may be trained using backpropagation learning
- Training may be based on the content of EM, or actual experiences, or both
- In turn, AEM may be used to help training the ACS (just like the EM may, but with summarized info)

(AEM networks may involve localist representations)

Episodic Memory

Questions?

Outline

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. **Simulation examples**
7. Properties and related theorems
8. Summary

Simulation: Categorical Inference Task

- The categorical inference task (Sloman, 1998; Sun & Zhang, 2006)
  - Premise specificity

    All flowers are susceptible to thrips $\Rightarrow$ All roses are susceptible to thrips

    All plants are susceptible to thrips $\Rightarrow$ All roses are susceptible to thrips
  - Inclusion similarity

    All plants contain bryophytes $\Rightarrow$ All flowers contain bryophytes

    All plants contain bryophytes $\Rightarrow$ All mosses contain bryophytes

- Which one in each pair is stronger?

Simulation

- Which argument is the stronger?
  - Premise specificity: flower $\Rightarrow$ rose (82%) vs. plant $\Rightarrow$ rose (18%)
  - Inclusion similarity: plant $\Rightarrow$ flower (91%) vs. plant $\Rightarrow$ mosses (9%)
- Average likelihood of arguments
  - Premise specificity: 0.86 (flower $\Rightarrow$ rose)
  - Inclusion similarity: 0.89 (plant $\Rightarrow$ flower)
- How do we explain and simulate this?
- These results show the presence of SBR
  - If only RBR was used, relative argument strength ~ 50%
  - Likelihood of arguments ~ 1.

Simulation

Simulation setup:

- Scaling parameters: $\alpha = 0.5$, $\beta = 1.0$ (RBR vs. SBR; based on contexts)
- The top level contains "category inclusion rules" (background knowledge):
  - "Flowers are plants"
  - "Mosses are plants"
  - Etc…
- At the bottom level, the features of the chunks (e. g., "flowers," "mosses") were represented (e.g., "petal," "stem," and other possibly unrecognizable microfeatures)

Simulation

Simulation process:

1. The chunk node represented in the premise of the conclusion statement is activated at the top level;
2. Which in turn activates (micro)features in the bottom level;
3. SBR was performed through the interaction of the top/bottom levels;
4. RBR was performed at the top level;
5. The results of RBR and SBR were integrated (*Max* function)

Simulation

- Simulation results
  - Which argument is the stronger?

    Premise specificity: flower $\Rightarrow$ rose (83%) vs. plant $\Rightarrow$ rose (17%)

    Inclusion similarity: plant $\Rightarrow$ flower (82%) vs. plant $\Rightarrow$ mosses (18%)
  - Average likelihood of arguments

    Premise specificity: 0.87 (flower $\Rightarrow$ rose)

    Inclusion similarity: 0.86 (plant $\Rightarrow$ flower)
- These results provide a good match to the human data.

Simulation

Other experimental conditions:

- Explicitly stating the inclusion argument (human result: 0.99)

  E.g., All plants contain bryophytes. All mosses are plants.

  $\Rightarrow$ All mosses contain bryophytes.

- Having participants make the categorical inclusion judgment before estimating the likelihoods (human result: 0.92):

  E.g., Are all mosses plants?

  All plants contain bryophytes. $\Rightarrow$ All mosses contain bryophytes.

- How do we explain/capture these?

Simulation

- In CLARION, this amounts to manipulating the weight of RBR in knowledge integration (changing relative weighting of RBR/SBS)

- Explicitly stating the inclusion relation in the argument itself:

  $\alpha = \beta = 1.0$

  Simulation result: Mean likelihood = 0.99

- Having participants make the categorical inclusion judgment ahead of the time:

  $\alpha = 0.88, \beta = 1.0$

  Simulation result: Mean likelihood = 0.91

Simulation: Insight in Problem Solving

- Insight problem solving (Hélie & Sun, 2010)

  "A dealer in antique coins got an offer to buy a beautiful bronze coin. The coin had an emperor's head on one side and the date 544 B.C. stamped on the other. The dealer examined the coin, but instead of buying it, he called the police. Why?" (Schooler et al., 1993, p.182)

- Two groups of participants: verbalization vs. no verbalization
- Better performance by the no verbalization group
- How do we capture/explain this?

Simulation

Simulation setup:

- Here, each concept (e.g., carving pattern, refusal to buy, bronze material) is represented at the top level by a chunk node;
- Each chunk node is connected to a set of (micro)feature nodes at the bottom level;
- Top-level associative rules represent conceptual links between concepts (i.e., culturally shared "semantic" knowledge);
- Similarity-based associations and soft constraint satisfaction at the bottom level (through feature representations) (after training)
- Exemplars may be coded by the top-level rules (after training)

Simulation

Simulation process:

- Simultaneous bottom-level and top-level processing:
  - A round of soft constraint satisfaction and SBR is initiated in the bottom level;
  - The result is sent bottom-up to activate chunk nodes;
  - RBR occurs at the top level; may (or may not) activate other chunk nodes in the top level;
- A response (combining the two levels) is stochastically chosen to be sent back to the ACS.

Simulation Results

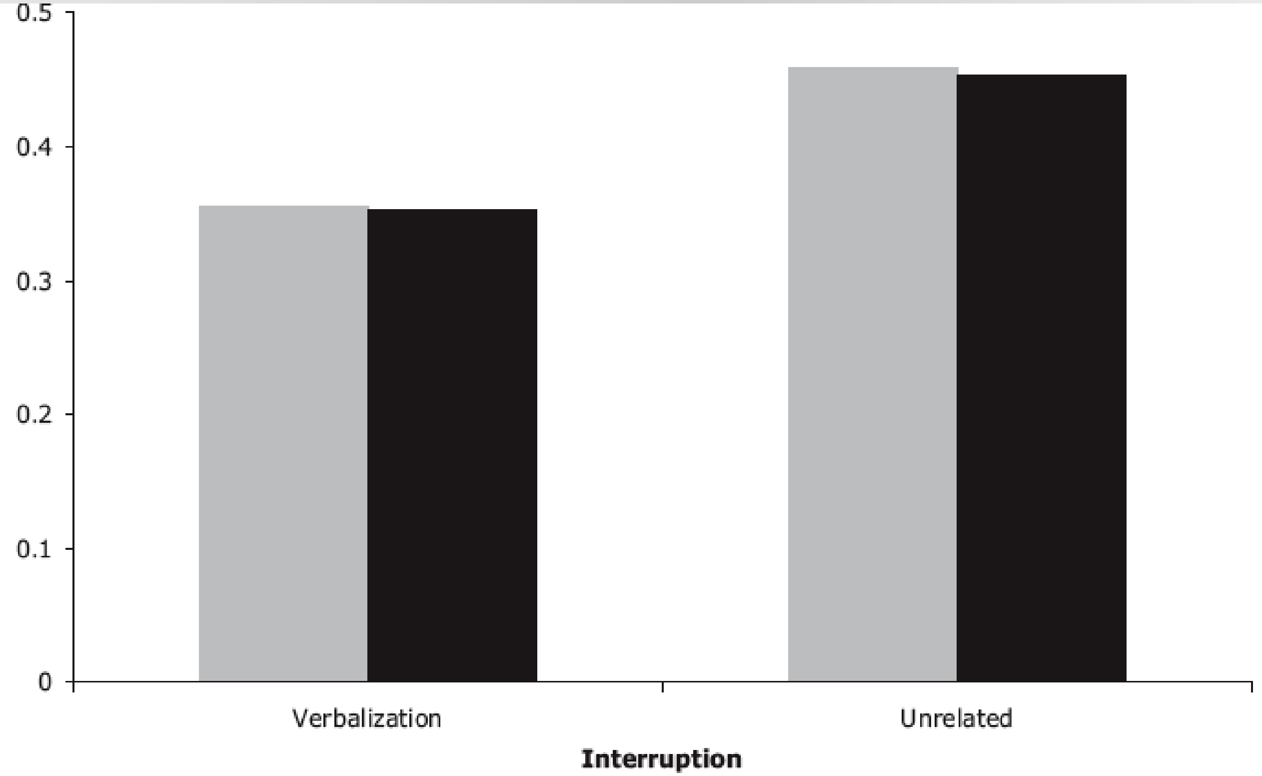


Figure 8. Proportion of correct explanations selected by the participants in Schooler, Ohlsson, and Brooks's (1993) Experiment 1 (gray bars) and by the CLARION model (black bars). The *x*-axis represents the distracting activity during the interruption period.

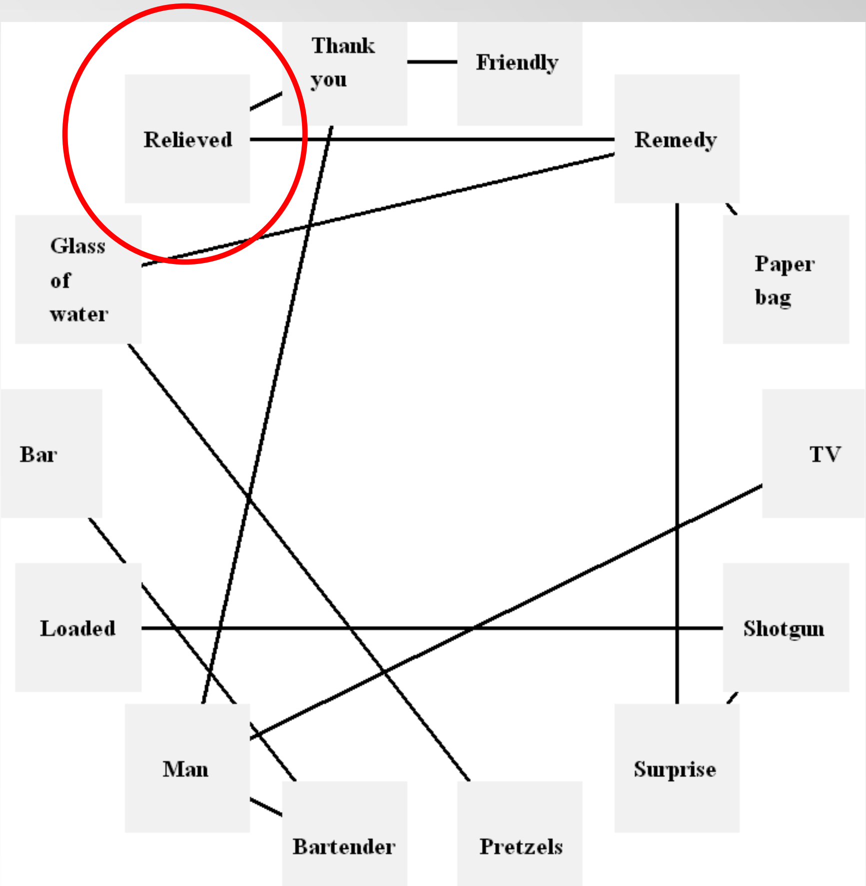
Simulation: Insight in Problem Solving

- Insight problem solving (Durso, Rea, & Dayton, 1994; Hélie & Sun, 2010)

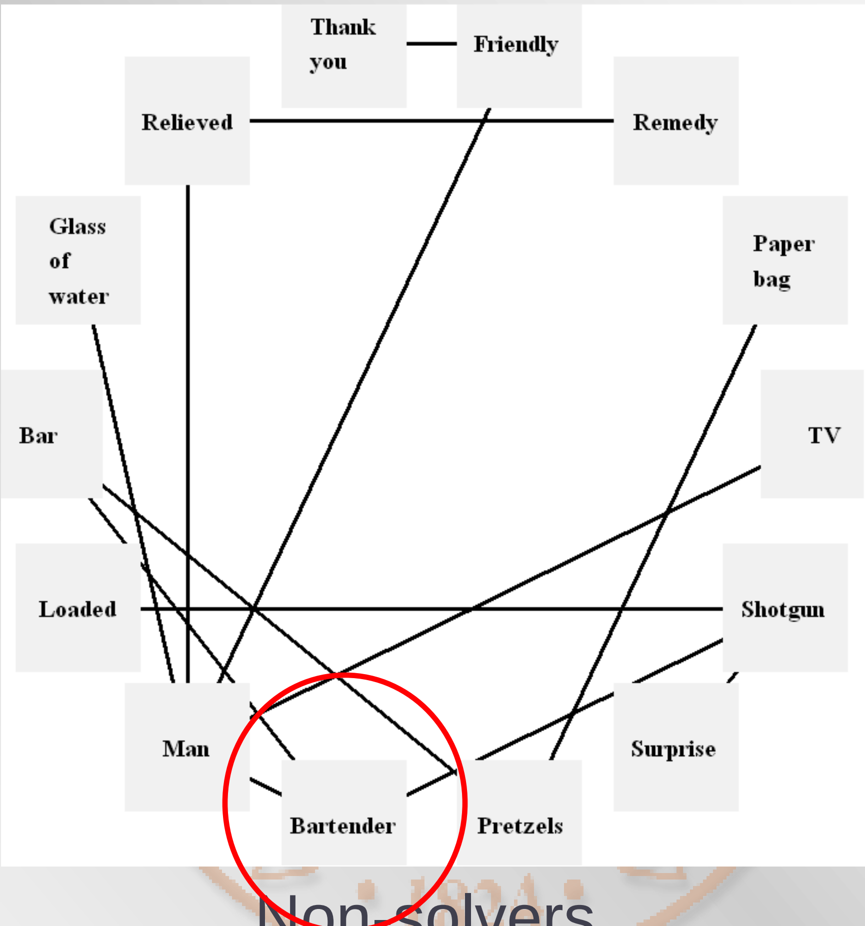
  "A man walks into a bar and asks for a glass of water. The bartender points a shotgun at the man. The man says 'thank you', and walks out. Why?"

- Participants may ask a few questions and get yes/no answers
- How do we capture/explain this?

Simulation



Solvers



Non-solvers

Simulation

Simulation setup:

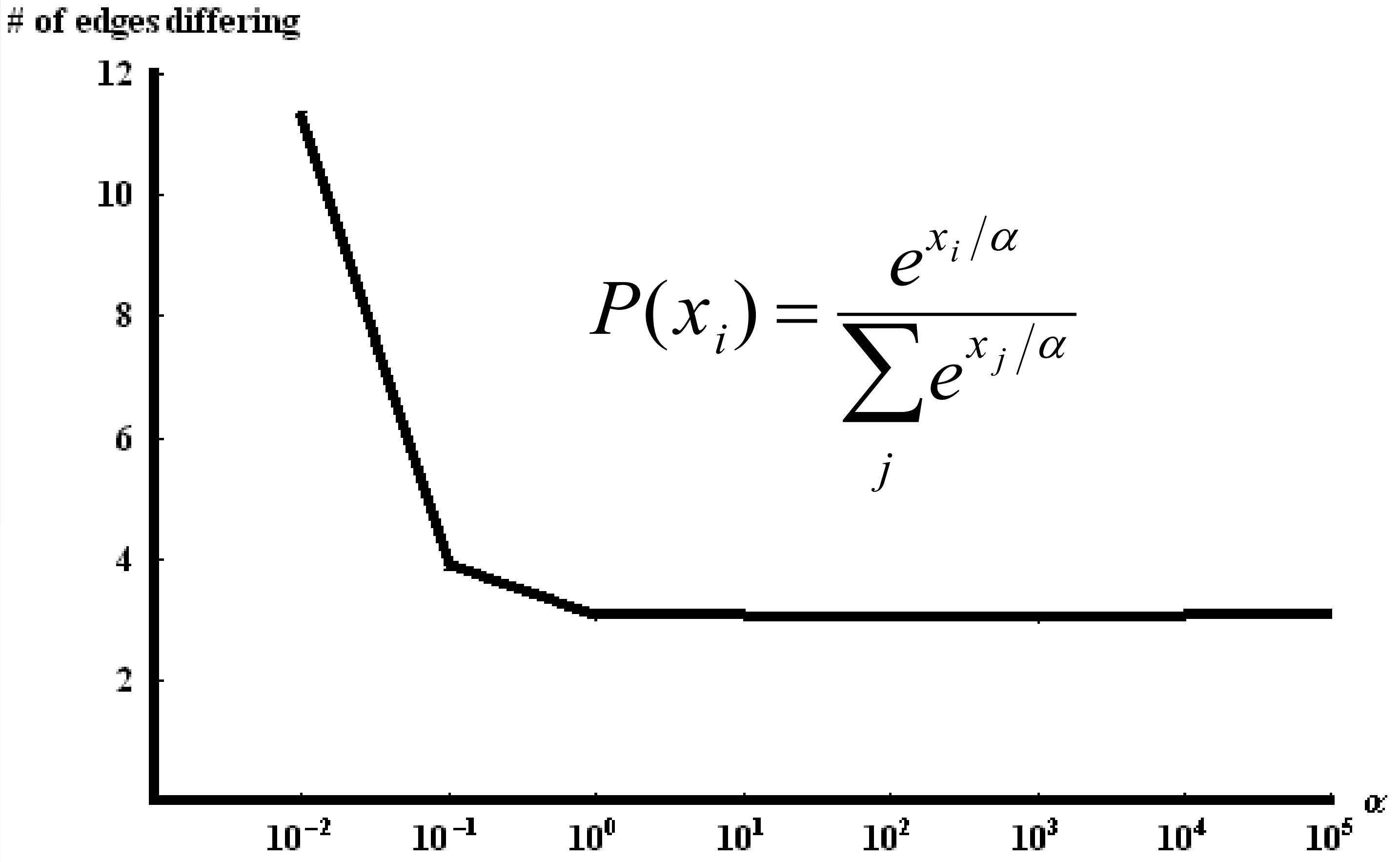
- Each concept (e.g., each node in the graphs) is represented at the top level by a chunk node;
- Each chunk node is connected to a (random) set of (micro)feature nodes at the bottom level;
- Top-level rules represent the conceptual links (e.g., as in the non-solvers' graph) (i.e., culturally shared "semantic" knowledge);
- Exemplars coded by the top-level rules (after training)
- Similarity-based associations and soft constraint satisfaction through the bottom level (after training)

Simulation

Simulation process:

- At the bottom level, a round of implicit processing is initiated (soft constraint satisfaction and SBR);
- The result is sent bottom-up to activate chunk nodes;
- At the top level, RBR is used to activate other chunk nodes (with common, stereotyped knowledge);
- A response (combining SBR and RBR) is stochastically chosen to be sent back to the ACS (which may then serve as questions)
- External answers may be coded at the top level (thus altering explicit knowledge)

Simulation



# of edges differing
12
10
8
6
4
2
$P(x_i) = \frac{e^{x_i/\alpha}}{\sum_j e^{x_j/\alpha}}$
α
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}

Simulations

Questions?

Formal properties and related theorems

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. **Formal properties and related theorems**

Theorems

- Many common reasoning *patterns* can be handled by CLARION:
  - Inexact information
  - Incomplete information
  - Similarity matching
  - Superclass to subclass inheritance
  - Subclass to superclass (reverse)“inheritance”
  - Cancellation of inheritance (of both types)
  - Mixed rule-based and similarity-based reasoning
  - Etc.

- Accounts for many everyday commonsense reasoning situations (Sun, 1994)

Theorems

Case 1: Inexact information

Theorem:

Network state: Let c_i and c_j be chunks represented in the top level of the NACS by chunk nodes, w_{ij}^r be a rule in the top level of the NACS linking chunk nodes i and j. Assume that $s_i < 1$.

Result: $s_j = s_i$

(i.e., inexact information leads to inexact conclusions)

Theorems

Case 2: Incomplete information

Theorem:

Network state: Let c_{i1}, c_{i2}, c_{i3}, c_j, be chunks represented in the top level of the NACS, w_{i1j}^r, w_{i2j}^r, w_{i3j}^r, be the rule in the top level of the NACS linking chunk nodes i_k and j. Assume that $s_{i1} = s_{i2} = 1$ and that $s_{i3} = 0$.

Result: $s_j = 2/3$

(i.e., partial information leads to inexact conclusions)

Theorems

Case 3: Similarity matching

Theorem:

Network state: Let c_i, c_j, c_k be chunks represented in the top level of the NACS, $s_{ci\sim cj}$ be the similarity from *I* to *j*, and w_{jk}^r be the rule in the top level of the NACS linking chunk nodes *j* and *k*. Assume that $s_i = 1$.

Result:

$$s_k = s_{ci\sim cj} = \frac{n_{c_i Ç c_j}}{f(n_j)}$$

(i.e., similar conditions lead to inexact conclusions, proportional to the similarity)

Theorems

Case 4: Superclass-to-subclass inheritance

Theorem:

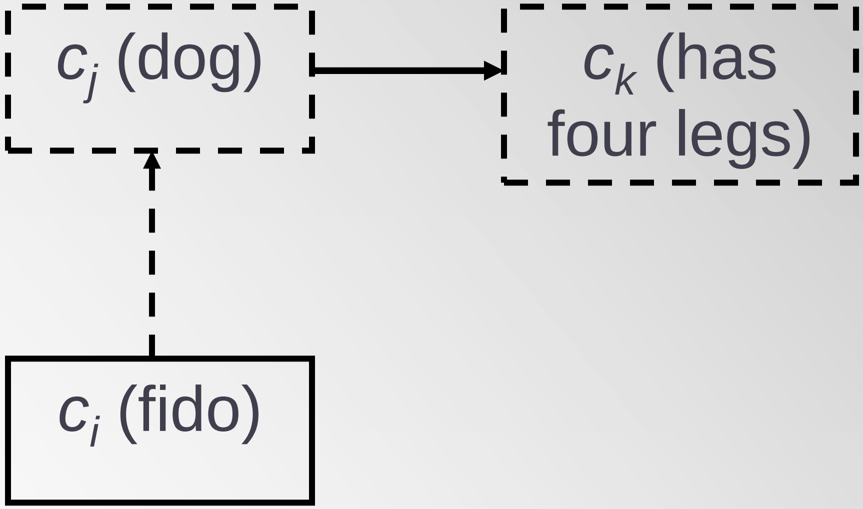
Network state: Let c_i, c_j, c_k be chunks represented in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j, and w_{jk}^r be a rule in the top level of the NACS linking chunks j and k. Assume that $s_i = 1$

Result: $s_k \approx 1$

(i.e. properties of the superclass can be inferred for the subclass)

Theorems: Examples

Case 4: Superclass-to-subclass inheritance



Q

Theorems

Case 5: Subclass-to-superclass reverse “inheritance”

Theorem:

Network state: Let c_i, c_j, c_k be chunks represented in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j, and w_{ik}^r be a rule in the top level of the NACS linking chunks i and k. Assume that $s_j = 1$.

Result: $s_k < 1$

(i.e., properties of the subclass may be plausibly inferred for the superclass –- inexact/uncertain conclusions)

Theorems

Case 6: Cancellation of superclass-to-subclass inheritance

Theorem:

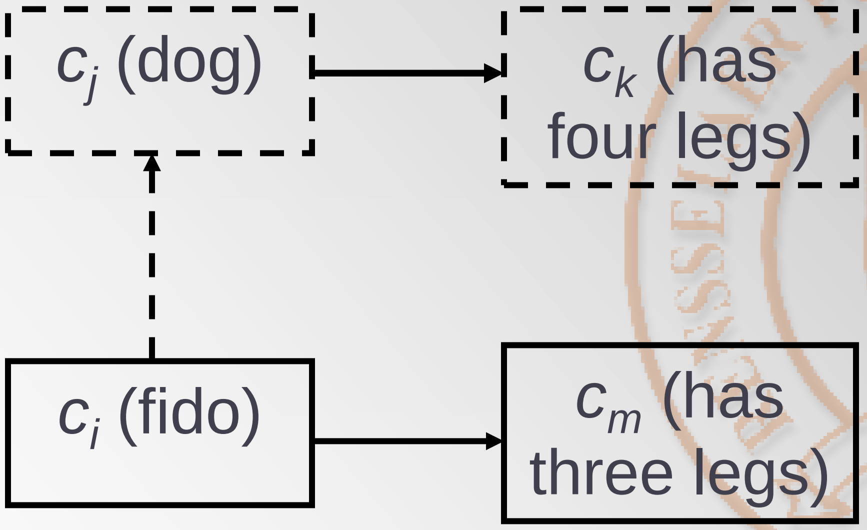
Network state: Let c_i, c_j, c_k, c_m be chunks represented in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j, and w_{jk}^r and w_{im}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result: $s_m = 1 > s_k$

(i.e. superclass-to-subclass inheritance inferences *may be cancelled by contradictory information)*

Theorems: Examples

Case 6: Cancellation of superclass-to-subclass inheritance



Theorems

Case 7: Cancellation of subclass-to-superclass reverse "inheritance"

Theorem:

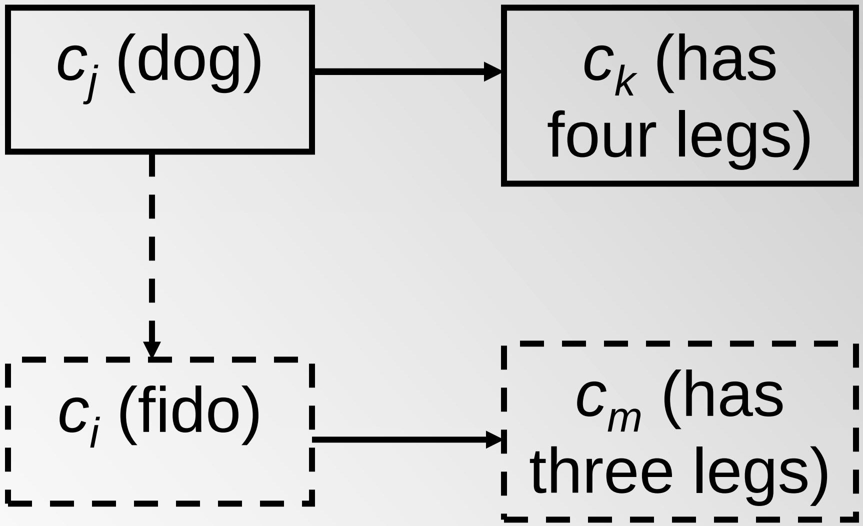
Network state: Let c_i, c_j, c_k, c_m be chunks represented in the top level of the NACS, the category represented by c_j is a proper subset of the category represented by c_i, and $w_{jk}{}^r$ and $w_{im}{}^r$ be rules in the top level of the NACS. Assume that $s_i = 1$.

Result: $s_m = 1 > s_k$

(i.e., subclass-to-superclass "inheritance" may be cancelled by contradictory information)

Theorems: Examples

Case 7: Cancellation of subclass-to-superclass reverse “inheritance” (i.e., cancellation of induction)



Theorems

Case 8: Mixed rules and similarities

Theorem 1:

State of the network: Let c_i, c_j, c_k, be chunks represented in the top level of the NACS, $s_{cj\sim ck}$ be the similarity between chunks j and k, and w_{ij}^r be a rule in the top level of the NACS between chunks i and j. Assume that $s_i = 1$.

Result:

$$s_k = \frac{n_{c_j \cap c_k}}{f(n_k)}$$

(i.e., conclusions similar to the one inferred may be plausible, with the plausibility proportional to the similarity)

Theorems

Case 8: Mixed rules and similarities

Theorem 2:

State of the network: Let c_i, c_j, c_k, be chunks represented in the top level of the NACS, and w_{ij}^r, w_{jk}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result: $s_k = 1$

(i.e., rule chaining)

Theorems

Case 8: Mixed rules and similarities

Theorem 3:

State of the network: Let c_i, c_j, c_k, c_m be chunks represented in the top level of the NACS, $s_{ci\sim cj}$ is the similarity between chunks i and j, and w_{jk}^r and w_{km}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result:

$$s_m = \frac{n_{c_i \cap c_j}}{f\left(n_j\right)}$$

(i.e. similarity matching then rule chaining)

Theorems

Case 8: Mixed rules and similarities

Theorem 4:

State of the network: Let c_i, c_j, c_k, c_m be chunks represented in the top level of the NACS, $s_{cj\sim ck}$ is the similarity between chunks j and k, and w_{ij}^r and w_{km}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result:

$$s_m = \frac{n_{c_j \cap c_k}}{f(n_k)}$$

(i.e., rule, then similarity, then rule)

Theorems

Case 8: Mixed rules and similarities

Theorem 5:

State of the network: Let c_i, c_j, c_k, c_m be chunks represented in the top level of the NACS, $s_{ck \sim cm}$ is the similarity between chunks k and m, and w_{ij}^r and w_{jk}^r be rules in the top level of the NACS. Assume that $s_i = 1$.

Result:

$$s_m = \frac{n_{c_k \cap c_m}}{f(n_m)}$$

(i.e., rule chaining then similarity)

Theorems

Case 8: Mixed rules and similarities

Theorem 6:

State of the network: Let c_i, c_j, c_k, c_m be chunks represented in the top level of the NACS, $s_{ci\sim cj}$ and $s_{ck\sim cm}$ are similarity measures between the chunks, and w_{jk}^{r} is a rule in the top level of the NACS between chunks j and k. Assume that $s_i = 1$.

Result:

$$s_m = \frac{n_{c_i \cap c_j}}{f(n_j)} \times \frac{n_{c_k \cap c_m}}{f(n_m)}$$

(i.e., similarity, rule, then similarity)

Theorems: Analyzing Some Details

Case 4: Superclass to subclass inheritance

Network state:

Let c_i, c_j, c_k be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j, and w_{jk}^r be a rule in the top level of the NACS linking chunks *j* and *k*. Assume that $s_i = 1$.

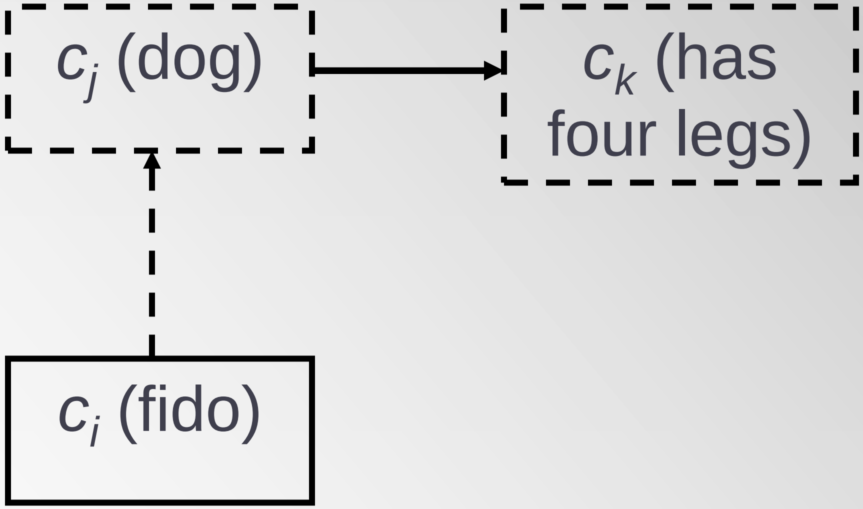
Derivation:

$$s_k = s_i \times s_{c_i \sim c_j} \times w_{jk}^r = \frac{n_{c_i \cap c_j}}{f(n_j)} = \frac{n_j}{f(n_j)} \approx 1$$

In words, chunk *k* is activated because chunk *i* fully activates chunk *j* (up to the slight non-linearity of *f()*, which is negligible). Chunk *j* has a top-level rule that transmits its activation to chunk *k*.

Theorems: Analyzing Some Details

Case 4: Superclass to subclass inheritance



Theorems: Analyzing Some Details

Case 6: Cancellation of superclass to subclass inheritance

Network state:

Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j, and $w_{jk}{}^r$ and $w_{im}{}^r$ be rules in the top level of the NACS. Assume that $s_i = 1$.

Derivation:

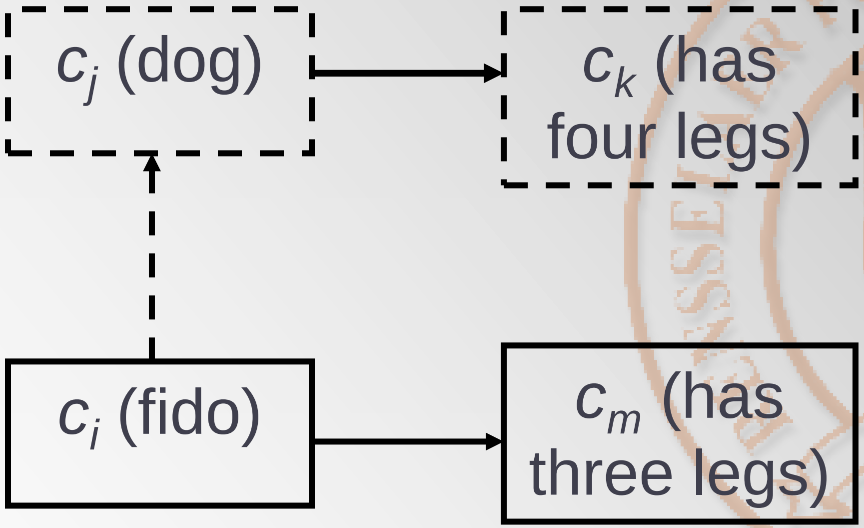
$$\begin{aligned} s_k &= s_i \times s_{c_i \sim c_j} \times w^r_{jk} \\ &= \frac{n_{c_i \cap c_j}}{f(n_j)} \\ &= \frac{n_j}{f(n_j)} \\ &< 1 \end{aligned}$$

$$\begin{aligned} s_m &= s_i \times w^r_{im} \\ &= 1 \end{aligned}$$

Hence, $s_m > s_k$. In words, chunk k is almost fully activated, but the denominator is slightly bigger than the numerator in its derivation [because *f()* is super-linear]. In contrast, chunk m is fully activated, because top-level rules are exact. This shows the superiority of rule-based reasoning over similarity-based reasoning.

Theorems: Analyzing Some Details

Case 6: Cancellation of superclass to subclass inheritance



Q

Theorems: Analyzing Some Details

Case 7: Cancellation of subclass to superclass "inheritance" (i.e., reversing induction)

Network state:

Let c_i, c_j, c_k, c_m be chunks in the top level of the NACS, the category represented by c_i is a proper subset of the category represented by c_j, and w_{im}^r and w_{jk}^r be rules in the top level of the NACS. Assume that $s_j = 1$.

Derivation:

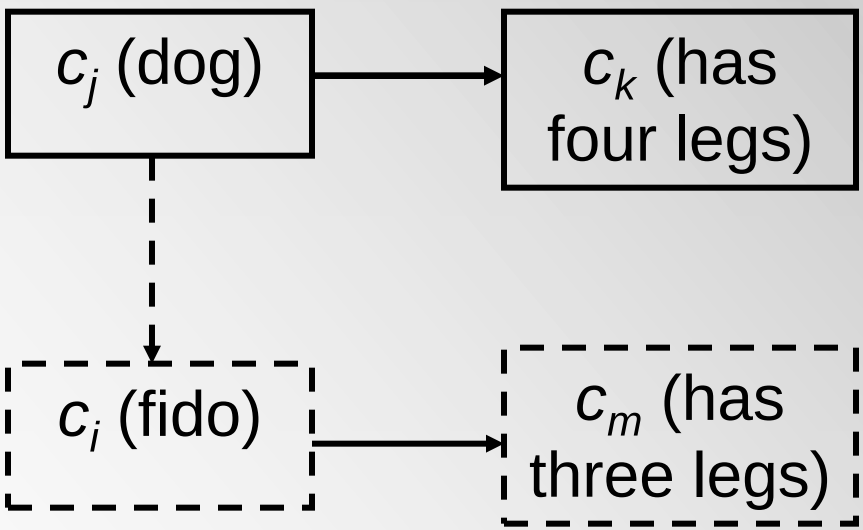
$$\begin{aligned} s_m &= s_j \times s_{c_j \sim c_i} \times w^r_{im} \\ &= \frac{n_{c_j \cap c_i}}{f(n_i)} \\ &= \frac{n_j}{f(n_i)} \\ &< 1 \end{aligned}$$

$$\begin{aligned} s_k &= s_j \times w^r_{jk} \\ &= 1 \end{aligned}$$

Hence, $s_k > s_m$. In words, chunk *m* is partially activated, because chunk *i* has more features than chunk *j* (remember that chunk *i* represents a proper subset of chunk *j*). On the other hand, chunk *k* is fully activated, because top-level rules are exact.

Theorems: Analyzing Some Details

Case 7: Cancellation of subclass to superclass "inheritance" (i.e., reversing induction)



Formal properties and related theorems

Questions?

Summary

1. Representation
2. Reasoning
3. Learning
4. Coordination of the NACS and the ACS
5. Episodic memory
6. Simulation examples
7. Formal properties and related theorems
8. **Summary**

Summary

- The Non-Action-Centered Subsystem (NACS) of CLARION: divided into the top level and the bottom level (same as other subsystems).
  - The top level: explicit declarative knowledge in the form of rules and chunk nodes (concepts).
  - The bottom level: implicit declarative knowledge in the form of (micro)feature nodes,

    with feature similarity matching, associative mapping, and soft constraint satisfaction.

Summary

- RBR is captured by the top level
- SBR is captured through the interaction of the bottom and the top level (the activation flows between the two levels, modulated by top-down weights and bottom-up weights, which capture similarity measures)
- The outcomes of the two levels are integrated (with *Max* at the top level)
- CLARION can account for a variety of common reasoning patterns through varying amounts of SBR and RBR, as well as associative mapping and soft constraint satisfaction.

Summary

- Explicit learning: from external sources or extracted from the bottom level of the ACS or the NACS
- Implicit learning: by assimilation of explicit knowledge given or by learning, for example, using the contents of EM.

Summary

Questions about the NACS?