6 KNOWLEDGE REPRESENTATION

6.0 Issues in Knowledge Representation
6.1 A Brief History of AI Representational Systems
6.2 Conceptual Graphs: A Network Language
6.3 Alternatives to Explicit Representation
6.4 Agent Based and Distributed Problem Solving
6.5 Epilogue and References
6.6 Exercises
Figure 6.15: Graph of “Mary gave John the book.”
**Figure 6.16:** Conceptual graph indicating that the dog named emma is brown.

```
   dog:emma  --->  color  -->  brown
```

**Figure 6.17:** Conceptual graph indicating that a particular (but unnamed) dog is brown.

```
   dog:#1352  --->  color  -->  brown
```

**Figure 6.18:** Conceptual graph indicating that a dog named emma is brown.

```
   dog:#1352  --->  color  -->  brown
     \          /  "emma"
      |\      /    |
      |  \    /     |
      |   \  /      |
      |    v       |
      name  --- "emma"
```
Figure 6.19: Conceptual graph of a person with three names.

- Name: "McGill"
- Name: "Nancy"
- Name: "Lil"
- Person: #941765
**Figure 6.20:** Conceptual graph of the sentence “The dog scratches its ear with its paw.”
In Conceptual Graph,

- Each concept node can indicate an individual of a specified type.
- This individual is the *referent* of the concept.
- This reference is indicated either individually or generically.
- If the referent uses an individual marker, the concept is an *individual* concept.
- If the referent uses the generic marker, then the concept is *generic*. 
Type Hierarchy

• If t and s are typed and $t \leq s$, t is said to be a subtype of s
• s is said to be supertype of t
• If s, t, and u are types with $t \leq s$ and $t \leq u$ then t is said to be a common subtype of s and u
• If $s \leq v$ and $u \leq v$ then v is a common supertype of s and u
• every pair of types must have minimal common supertype and maximal common subtype
• v is a minimal common supertype if $s \leq v$, $u \leq v$
• universal type T is a supertype of all types
• absurd type $\bot$ is a subtype of all types
Figure 6.21: A type lattice illustrating subtypes, supertypes, the universal type, and the absurd type. Arcs represent the relationship.
Specialization and Generalization: Operations

- **copy**
  - form a new graph $g$ that is exact copy of $g_1$

- **restrict**
  - allow concept nodes in a graph to be replaced by a node representing their specialization
  - if a concept is labeled with generic marker, the generic marker may be replaced by an individual marker
  - a type label may be replaced by one of its subtypes, if this is consistent with the referent of the concept

- **join**
  - combine two graphs into a single graph
  - specialization rule because the resulting graph is less general than either of its components

- **simplify**
  - If a graph contains two duplicate relations, then one of them may be deleted along with all its arcs
Specialization and Generalization:

- Join and restrict are specialization rules.
- If a graph $g_1$ is a specialization of $g_2$, then $g_2$ is a generalization of $g_1$.
- Generalization hierarchies are important in knowledge representation.
- Generalization hierarchies are used in many learning methods, allowing to construct a generalized assertion from a particular training instance.
Figure 6.22: Examples of restrict, join, and simplify operations.
Figure 6.23: Inheritance in conceptual graphs.

A conceptual graph

Inheritance of a property by a subclass

Inheritance of a property by an individual
Propositional Nodes

• Conceptual graphs include a concept type, proposition, that takes a set of conceptual graphs as its referent and allows us to define relation involving propositions.

• Propositional concepts are indicated as a box that contains another concept graph.

Example
“Tom believes that Jane likes pizza”

experiencer link is used with belief states based on the notation that they are something one experiences rather than does.

• Modal logics are concerned with the various ways propositions are entertained: believed, assert as possible, probably or necessarily true, intended as a result of an action, or counterfactual.
Conceptual Graphs and Logic

- A dog has a color of brown
  \[ \exists X \exists Y (\text{dog}(X) \land \text{color}(X,Y) \land \text{brown}(Y)) \]

- There are no pink dogs
  \[ \forall X \forall Y (\neg (\text{dog}(X) \land \text{color}(X,Y) \land \text{pink}(Y))) \]

- Conceptual graphs are equivalent to predicate calculus in their expressive power
Figure 6.24: Conceptual graph of the statement “Tom believes that Jane likes pizza,” showing the use of a propositional concept.
Figure 6.25: Conceptual graph of the proposition “There are no pink dogs.”
Figure 6.26: The functions of the three-layered subsumption architecture from Brooks (1991a). The layers are described by the AVOID, WANDER, and EXPLORE behaviors.
Figure 6.27: A possible state of the copycat workspace. Several examples of bonds and links between the letters are shown; adopted from Mitchell (1993).
Figure 6.28: A small part of copycat’s slipnet with nodes, links, and label nodes shown; adapted from Mitchell (1993).
Figure 6.29: Two conceptual graphs to be translated into English.
Figure 6.30: Example of an analogy test problem.

Choose one:

A  
B  
C  
D