Conceptual Graphs

- Graph Structure
  - Finite, connected, bipartite
  - Arcs are not labeled
  - Conceptual relation nodes are introduced between concepts
  - The bipartite nature of the graph means concepts can only link to conceptual relations and vice versa
  - In drawings, concepts are shown in boxes and conceptual relations in ellipses

- Concepts may be concrete (dog, child, etc.) or abstract (love, beauty, etc.)

Arity of Relations

- Examples of 1-ary, 2-ary, and 3-ary relations

Flies is a 1-ary relation.

Color is a 2-ary relation.

Parents is a 3-ary relation.
Graph of a Sentence

• “Mary gave John the book”

- As in conceptual dependency, the verb plays a central role in the structure
- The verb “give” in this sentence has an agent, an object, and a recipient

Group Work

• What does the following conceptual graph represent
Types and Individuals

- In the first case, the type is dog and the individual is “emma”
- A specific but unnamed dog is given a unique number (#)
- An alternative representation is to use a dog specified by a # and add a conceptual relation for a name

Three Names

- “Her name was McGill and she called herself Lil, but everyone knew her as Nancy” (song lyric)

- Who was the artist? What was the name of the song?
Itchy Dog

- What is the English sentence for this structure?

- If the same, unspecified individual is present in two or more nodes, a variable can be introduced that may eventually be bound to the same value

Type Lattice

- Concepts often form a lattice of types, such as a class golden retriever a type of dog, a type of carnivore, a type of animal, and so forth

- ♦ is a supertype of all types, ◻ is the absent type

- Answering queries about a pair of concepts may involve finding the minimum common supertype
Generalization and Specialization

- A concept node can be replaced with a restriction

Join of Concepts

- If two graphs contain an identical node, they can be joined together by having only one copy of the identical node
- Join is a form of restriction since the resultant graph is more specific than the original graphs
Simplification

- A join may result in duplicate information
- The simply operation allows the removal of duplication information

Inheritance

- Inheritance is a form of generalization
- Generalization does not guarantee that the resultant graph is true even if the original graphs are true
Propositional Nodes

• "Tom believes that Jane likes pizza"

```
  person:Tom  →  experience  →  believe
                ↓                ↓
                  object
```

• "There are no pink dogs"

```
  person:Jane  →  agent  →  likes
                ↓    ↓
                  pizza
```

• The verb believes takes a propositional node as its object

• In some cases a propositional node may stand alone, as seen here:

```
  proposition:
    dog  →  color  →  pink
          ↓
          neg
```

• This is similar to modal logics that introduce a level of believability, such as necessary, probably, possible, or other levels, such as negative shown here
Group Work

- What does the following conceptual graph represent

```
belief \rightarrow \text{experiencer} \rightarrow \text{kate}
```

```
object \rightarrow \text{propoision:}
```

```
\text{neg} \rightarrow \text{experiencer} \rightarrow \text{john}
```

```
\text{object} \rightarrow \text{pizza}
```

Conceptual Graphs and Logic

- Conceptual graphs are equivalent to predicate calculus in expressive power
  
  \[ \forall X \forall Y (\neg (\text{dog}(X) \land \text{color}(X,Y) \land \text{pink}(Y))). \]

- Here is an algorithm to change a conceptual graph into a predicate calculus expression

1. Assign a unique variable, \(x_1, x_2, \ldots, x_n\), to each of the \(n\) generic concepts in \(g\).
2. Assign a unique constant to each individual concept in \(g\). This constant may simply be the name or marker used to indicate the referent of the concept.
3. Represent each concept node by a unary predicate with the same name as the type of that node and whose argument is the variable or constant assigned to that node.
4. Represent each \(n\)-ary conceptual relation in \(g\) as an \(n\)-ary predicate whose name is the same as the relation. Let each argument of the predicate be the variable or constant assigned to the corresponding concept node linked to that relation.
5. Take the conjunction of all atomic sentences formed under 3 and 4. This is the body of the predicate calculus expression. All the variables in the expression are existentially quantified.