Conceptual Graphs

Harry Delugach Dan Rochowiak
Department of Computer Science
Intelligent Systems Lab
University of Alabama in Huntsville
Huntsville, AL 35899 USA
delugach@ cs.uah.edu drochowi@cs.uah.edu

Conceptual Graphs

- Introduced by John Sowa in 1984
- Focus of seven workshops and 13 international conferences since 1986
- Studied by researchers in 11 countries
- Included in work on hundreds of research papers published in five languages
- One dialect of ISO standard (Common Logic) being finalized 2006.
- Will be Annex B of ISO 24707

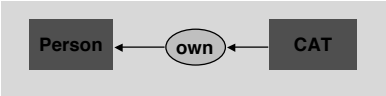
Concept

- Any distinguishable idea (not just object!)
- Shown as type-labeled rectangle

PERSON	GIVING	Proposition
A person	*A giving*	*A proposition*
Some person	*An instance of giving*	
(∃x)(person(x))	(∃x)(giving(x))	(∃x)(proposition(x))
(exists (x) (person(x))	(exists (y) (giving(y))	(exists (y) (proposition(y))

Relation

- Relation
  - Relationship between two or more concepts
  - Shown as oval or circle
  - The owner of a CAT is a Person.

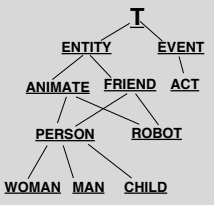


$(\exists x)(person(x) \wedge cat(y) \wedge own(x, y))$

```
(exists (x y)
        (person(x) and cat(y) and own(x, y) )
```


Type Hierarchy

- **Captures "is-a-kind-of" relationship**



ANIMATE < ENTITY.

ANIMATE is-a-kind-of ENTITY.

PERSON is-a-kind-of ANIMATE.

PERSON is-a-kind-of FRIEND.

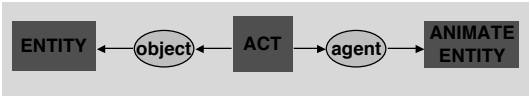
$(\forall x)(person(x) \supset animate(x))$

```
(forall (x )
        (implies person(x) animate(x))
```


Canonical Graph

- **Specifies semantic constraints**
- **Constrains types, relations between concepts**

"The object of an act is an entity and the agent of an act is an animate entity."



$(\forall x)(act(x) \supset (\exists y\ \exists z)(entity(y) \wedge animateentity(z) \wedge object(x, y) \wedge agent(x,z)))$

Semantic Checking

- **Pre-defined semantics can be checked**
- **Can detect a "badly-formed" graph**

"STONE" is not an animate entity (or one of its subtypes) so the graph is illegal.

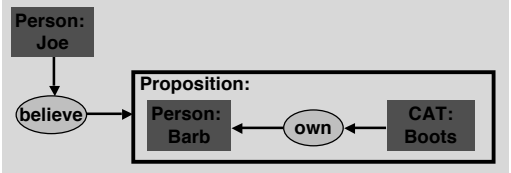
Referent

- Follows the type with a colon ":"
- Denotes individuals and/or sets
- Denotes instances (occurrences) of a type

PERSON: Harry	CAT: {*}@4	CAT: {Amelia, Albert}
The person identified as "Harry"	A set of 4 cats	The set of cats with members "Amelia" and "Albert"

Context

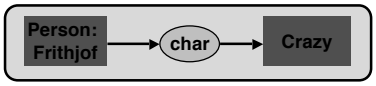
- **A concept that encloses an entire proposition**



Person "Joe" believes (the proposition) that the owner of the cat "Boots" is Person "Barb"

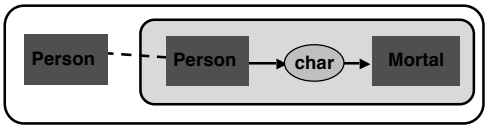
Negation

- There does not exist a person named Joe

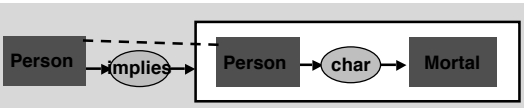


- Frithjof is not crazy

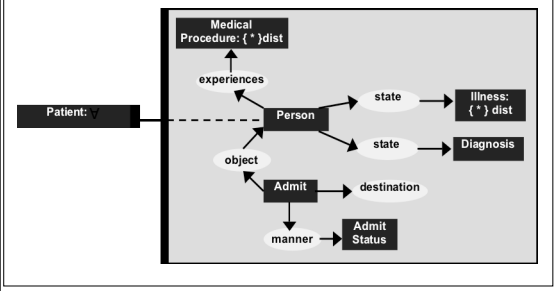
Implication



person(x) ⊃ mortal(y) ∧ char(x,y))

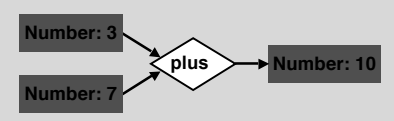


Definitions



Actor

- Specifying a function



- Output referent is changed based on the input referent