Description Logics

Structural Description Logics

Enrico Franconi

`franconi@cs.man.ac.uk`

`http://www.cs.man.ac.uk/~franconi`

Department of Computer Science, University of Manchester

Description Logics

- A logical reconstruction and *unifying* formalism for the representation tools
  - Frame-based systems
  - Semantic Networks
  - Object-Oriented representations
  - Semantic data models
  - Ontology languages
  - . . .
- A *structured* fragment of predicate logic
- Provide theories and systems for *expressing* structured information and for *accessing* and *reasoning* with it in a principled way.

Applications

Description logics based systems are currently in use in many applications.

- Configuration
- Conceptual Modeling
- Query Optimization and View Maintenance
- Natural Language Semantics
- I3 (Intelligent Integration of Information)
- Information Access and Intelligent Interfaces
- Terminologies and Ontologies
- Software Management
- Planning

A formalism

- Description Logics formalize many *Object-Oriented* representation approaches.
- As such, their purpose is to disambiguate many imprecise representations.

Frames or Objects

- Identifier
- Class
- Instance
- Slot (attribute)
  - Value
    - Identifier
    - Default
  - Value restriction
    - Type
    - Concrete Domain
    - Cardinality
    - Encapsulated method

Ambiguities: classes and instances

$$\begin{array}{l} \texttt{Person} : \mathtt{AGE} : \mathtt{Number}, \\ \qquad \mathtt{SEX} : M, F, \\ \qquad \mathtt{HEIGHT} : \mathtt{Number}, \\ \qquad \mathtt{WIFE} : \mathtt{Person}. \end{array}$$

$$\begin{array}{l} \textit{john} : \mathtt{AGE} : 29, \\ \qquad \mathtt{SEX} : M, \\ \qquad \mathtt{HEIGHT} : 76, \\ \qquad \mathtt{WIFE} : mary. \end{array}$$

Ambiguities: incomplete information

$$\begin{aligned} \texttt{29'er} : \mathtt{AGE} &: 29, \\ \mathtt{SEX} &: M, \\ \mathtt{HEIGHT} &: \mathtt{Number}, \\ \mathtt{WIFE} &: \mathtt{Person}. \end{aligned}$$

$$\begin{aligned} \textit{john} : \mathtt{AGE} &: 29, \\ \mathtt{SEX} &: M, \\ \mathtt{HEIGHT} &: \mathtt{Number}, \\ \mathtt{WIFE} &: \mathtt{Person}. \end{aligned}$$

Ambiguities: is-a

Sub-class:

$$
\begin{array}{|l|}
\hline
\texttt{Person} : \mathtt{AGE} : \mathtt{Number}, \\
\qquad\qquad \mathtt{SEX} : M, F, \\
\qquad\qquad \mathtt{HEIGHT} : \mathtt{Number}, \\
\qquad\qquad \mathtt{WIFE} : \mathtt{Person}. \\
\hline
\end{array}
$$

$$\Uparrow$$

$$
\begin{array}{|l|}
\hline
\texttt{Male} : \mathtt{AGE} : \mathtt{Number}, \\
\qquad\quad \mathtt{SEX} : M, \\
\qquad\quad \mathtt{HEIGHT} : \mathtt{Number}, \\
\qquad\quad \mathtt{WIFE} : \mathtt{Female}. \\
\hline
\end{array}
$$

Ambiguities: is-a

Instance-of:

$$
\begin{aligned}
\texttt{Male} : \mathtt{AGE} &: \mathtt{Number}, \\
\mathtt{SEX} &: M, \\
\mathtt{HEIGHT} &: \mathtt{Number}, \\
\mathtt{WIFE} &: \mathtt{Female}.
\end{aligned}
$$

$$\Uparrow$$

$$
\begin{aligned}
\textit{john} : \mathtt{AGE} &: 35, \\
\mathtt{SEX} &: M, \\
\mathtt{HEIGHT} &: 76, \\
\mathtt{WIFE} &: mary.
\end{aligned}
$$

Ambiguities: is-a

Instance-of:

$$\texttt{29'er} : \mathtt{AGE} : 29, \quad \mathtt{SEX} : M, \quad \mathtt{HEIGHT} : \mathtt{Number}, \quad \mathtt{WIFE} : \mathtt{Person}.$$

$$\Uparrow$$

$$\textit{john} : \mathtt{AGE} : 29, \quad \mathtt{SEX} : M, \quad \mathtt{HEIGHT} : \mathtt{Number}, \quad \mathtt{WIFE} : \mathtt{Person}.$$

Ambiguities: relations

Implicit relation:

$$\begin{array}{l} \textit{john} : \texttt{AGE} : 35, \\ \qquad \texttt{SEX} : M, \\ \qquad \texttt{HEIGHT} : 76, \\ \qquad \underline{\texttt{WIFE}} : mary. \end{array}$$

$$\begin{array}{l} \textit{mary} : \texttt{AGE} : 32, \\ \qquad \texttt{SEX} : F, \\ \qquad \texttt{HEIGHT} : 59, \\ \qquad \underline{\texttt{HUSBAND}} : john. \end{array}$$

Ambiguities: relations

Explicit relation:

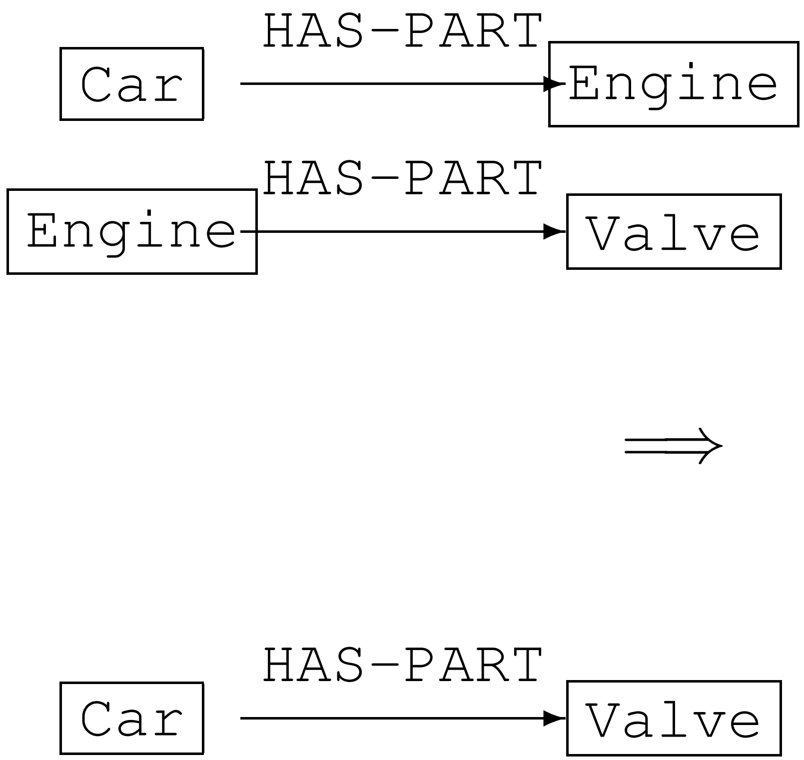
$$\textit{john} : \texttt{AGE} : 35,\quad \texttt{SEX} : M,\quad \texttt{HEIGHT} : 76.$$

$$\textit{mary} : \texttt{AGE} : 32,\quad \texttt{SEX} : F,\quad \texttt{HEIGHT} : 59.$$

$$\textit{m-j-family} : \underline{\texttt{WIFE}} : mary,\quad \underline{\texttt{HUSBAND}} : john.$$

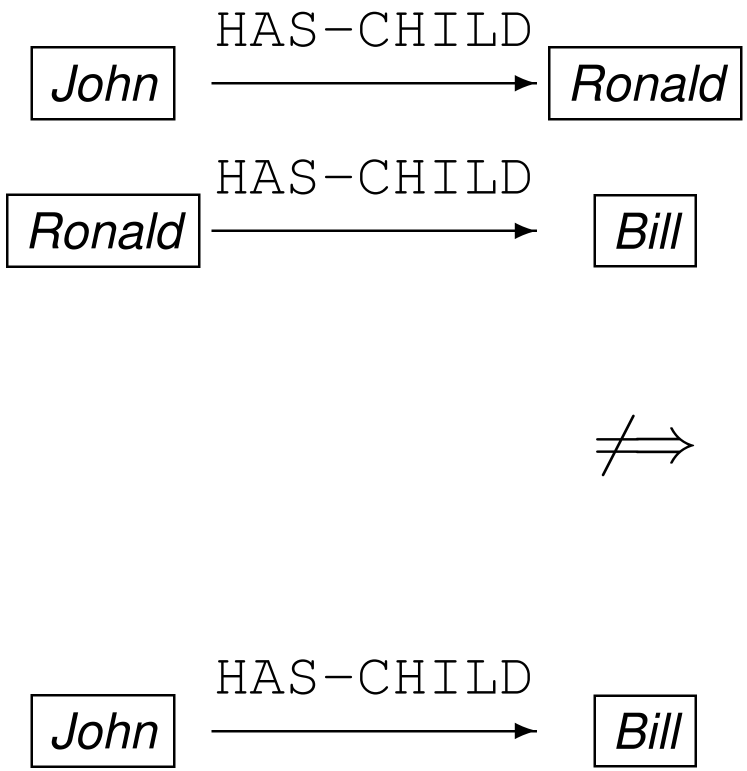
Ambiguities: relations

Special relation:



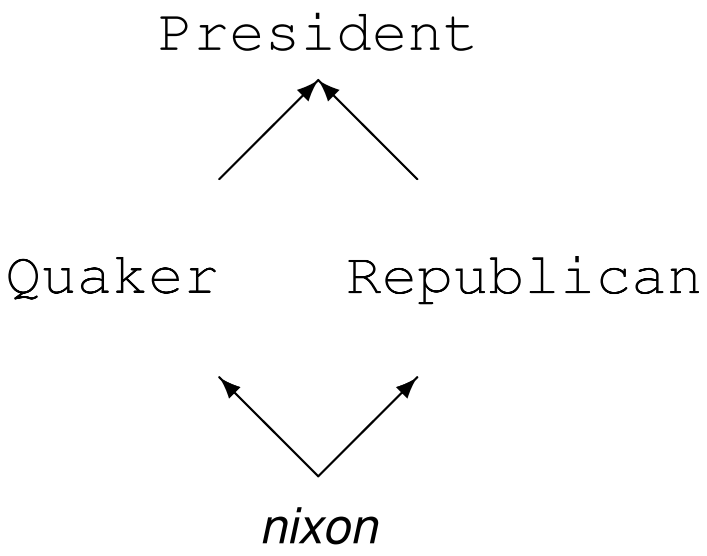
Ambiguities: relations

Normal relation:



Ambiguities: default

The *Nixon* diamond:

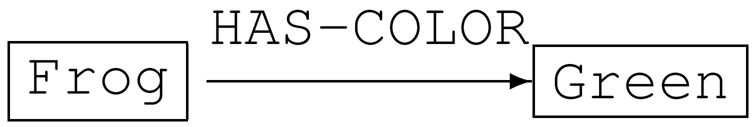


Quakers are pacifist, Republicans are not pacifist.

$\Longrightarrow$ Is Nixon pacifist or not pacifist?

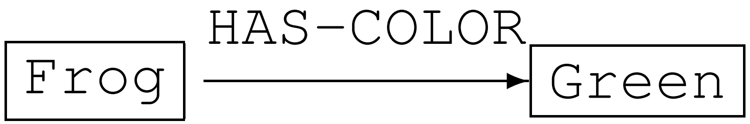
Ambiguities: quantification

What is the exact meaning of:



Ambiguities: quantification

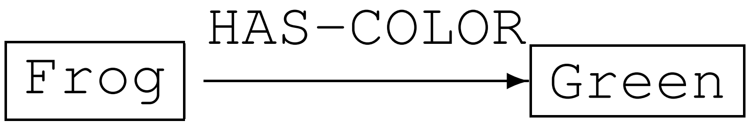
What is the exact meaning of:



- Every frog is just green

Ambiguities: quantification

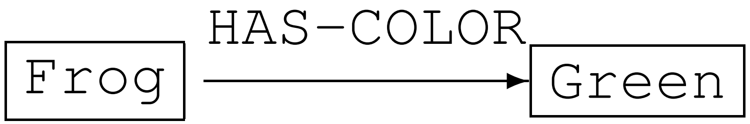
What is the exact meaning of:



- Every frog is just green
- Every frog is also green

Ambiguities: quantification

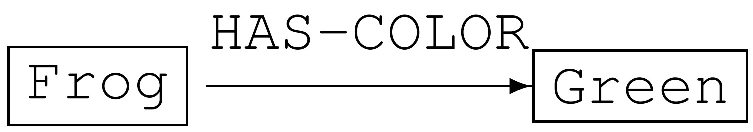
What is the exact meaning of:



- Every frog is just green
- Every frog is also green
- Every frog is of some green

Ambiguities: quantification

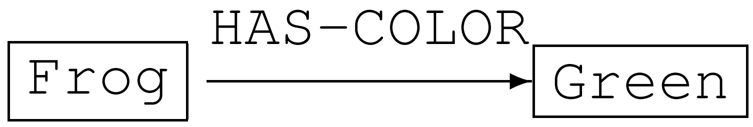
What is the exact meaning of:



- Every frog is just green
- Every frog is also green
- Every frog is of some green
- There is a frog, which is just green
- . . .

Ambiguities: quantification

What is the exact meaning of:



- Every frog is just green
- Every frog is also green
- Every frog is of some green
- There is a frog, which is just green
- . . .
- Frogs are typically green, but there may be exceptions

False friends

- The meaning of object-oriented representations is logically very ambiguous.
- The appeal of the graphical nature of object-oriented representation tools has led to forms of reasoning that do not fall into standard logical categories, and are not yet very well understood.
- It is unfortunately much easier to develop some algorithm that appears to reason over structures of a certain kind, than to *justify* its reasoning by explaining what the structures are saying about the domain.

A structured logic

- Any (basic) Description Logic is a fragment of FOL.
- The representation is at the *predicate level*: no variables are present in the formalism.
- A Description Logic theory is divided in two parts:
  - the definition of predicates (*TBox*)
  - the assertion over constants (*ABox*)
- Any (basic) Description Logic is a subset of $\mathcal{L}_3$, i.e. the function-free FOL using only at most *three* variable names.

Why not FOL

If FOL is directly used without additional restrictions then

- the structure of the knowledge is destroyed, and it can not be exploited for driving the inference;
- the expressive power is too high for obtaining decidable and efficient inference problems;
- the inference power may be too low for expressing interesting, but still decidable theories.

Structured Inheritance Networks: KL-ONE

- Structured Descriptions
  - corresponding to the complex relational structure of objects,
  - built using a restricted set of epistemologically adequate constructs
- distinction between conceptual (*terminological*) and instance (*assertional*) knowledge;
- central role of automatic classification for determining the subsumption – i.e., universal implication – lattice;
- strict reasoning, no defaults.

Types of the TBox Language

- **Concepts** – denote *entities*

    (unary predicates, classes)

  Example: `Student, Married`

  $\{x \mid \texttt{Student}(x)\}$,
  $\{x \mid \texttt{Married}(x)\}$

- **Roles**– denote *properties*

    (binary predicates, relations)

  Example: `FRIEND, LOVES`

  $\{\langle x, y \rangle \mid \texttt{FRIEND}(x, y)\}$,
  $\{\langle x, y \rangle \mid \texttt{LOVES}(x, y)\}$

Concept Expressions

Description Logics organize the information in classes – *concepts* – gathering homogeneous data, according to the relevant common properties among a collection of instances.

Example:

$$\texttt{Student} \sqcap \exists \texttt{FRIEND}.\texttt{Married}$$

$$\{x \mid \texttt{Student}(x) \wedge \exists y.\, \texttt{FRIEND}(x, y) \wedge \texttt{Married}(y)\}$$

A note on λ's

In general, λ is an explicit way of forming *names* of functions:

$\boxed{\lambda x.\ f(x)}$ is the function that, given input x, returns the value $f(x)$

The λ-conversion rule says that:

$$(\lambda x.\ f(x))(a) = f(a)$$

Thus, $\boxed{\lambda x.\ (x^2 + 3x - 1)}$ is the function that applied to 2 gives 9:

$$(\lambda x.\ (x^2 + 3x - 1))(2) = 9$$

We can give a name to this function, so that:

$$f_{231} \doteq \lambda x.\ (x^2 + 3x - 1)$$
$$f_{231}(2) = 9$$

λ to define predicates

Predicates are special case of functions: they are *truth* functions. So, if we think of a formula $P(x)$ as denoting a truth value which may vary as the value of x varies, we have:

$\lambda x.\ P(x)$ denotes a function from domain individuals to truth values.

In this way, as we have learned from FOL, P denotes exactly the set of individuals for which it is true. So, $P(a)$ means that the individual a makes the predicate P true, or, in other words, that a is in the extension of P.

For example, we can write for the *unary* predicate `Person`:

$$\mathtt{Person} \doteq \lambda x.\ \mathtt{Person}(x)$$

which is equivalent to say that `Person` denotes the *set* of persons:

$$\mathtt{Person} \leadsto \{x \mid \mathtt{Person}(x)\}$$
$$\mathtt{Person}^{\mathcal{I}} = \{x \mid \mathtt{Person}(x)\}$$
$$\mathtt{Person}(john) \quad \text{IFF} \quad john^{\mathcal{I}} \in \mathtt{Person}^{\mathcal{I}}$$

In the same way for the *binary* predicate `FRIEND`:

$$\mathtt{FRIEND} \doteq \lambda x, y.\ \mathtt{FRIEND}(x, y)$$
$$\mathtt{FRIEND}^{\mathcal{I}} = \{\langle x, y\rangle \mid \mathtt{FRIEND}(x, y)\}$$

The functions we are defining with the λ operator may be parametric:

$$\texttt{Student} \sqcap \texttt{Worker} = \lambda x.\ (\texttt{Student}(x) \wedge \texttt{Worker}(x))$$

$$(\texttt{Student} \sqcap \texttt{Worker})^{\mathcal{I}} = \{x \mid\ (\texttt{Student}(x) \wedge \texttt{Worker}(x)\}$$

$$(\texttt{Student} \sqcap \texttt{Worker})^{\mathcal{I}} = \texttt{Student}^{\mathcal{I}} \cap \texttt{Worker}^{\mathcal{I}}$$

(Verify as exercise)

Concept Expressions

$(\texttt{Student} \sqcap \exists\texttt{FRIEND.Married})^{\mathcal{I}}$

=

$(\texttt{Student})^{\mathcal{I}} \cap (\exists\texttt{FRIEND.Married})^{\mathcal{I}}$

=

$\{x \mid \texttt{Student}(x)\} \cap$
$\{x \mid \exists y.\texttt{FRIEND}(x, y) \wedge \texttt{Married}(y)\}$

=

$\{x \mid \texttt{Student}(x) \wedge$
$\exists y.\texttt{FRIEND}(x, y) \wedge \texttt{Married}(y)\}$

Objects: classes

Student

Person	
name:	[String]
address:	[String]
enrolled:	[Course]

$$\begin{aligned}\{x \mid \texttt{Student}(x)\} = \{x \mid\ & \texttt{Person}(x) \wedge \\ & (\exists y.\ \texttt{NAME}(x,y) \wedge \texttt{String}(y)) \wedge \\ & (\exists z.\ \texttt{ADDRESS}(x,z) \wedge \texttt{String}(z)) \wedge \\ & (\exists w.\ \texttt{ENROLLED}(x,w) \wedge \texttt{Course}(w))\ \}\end{aligned}$$

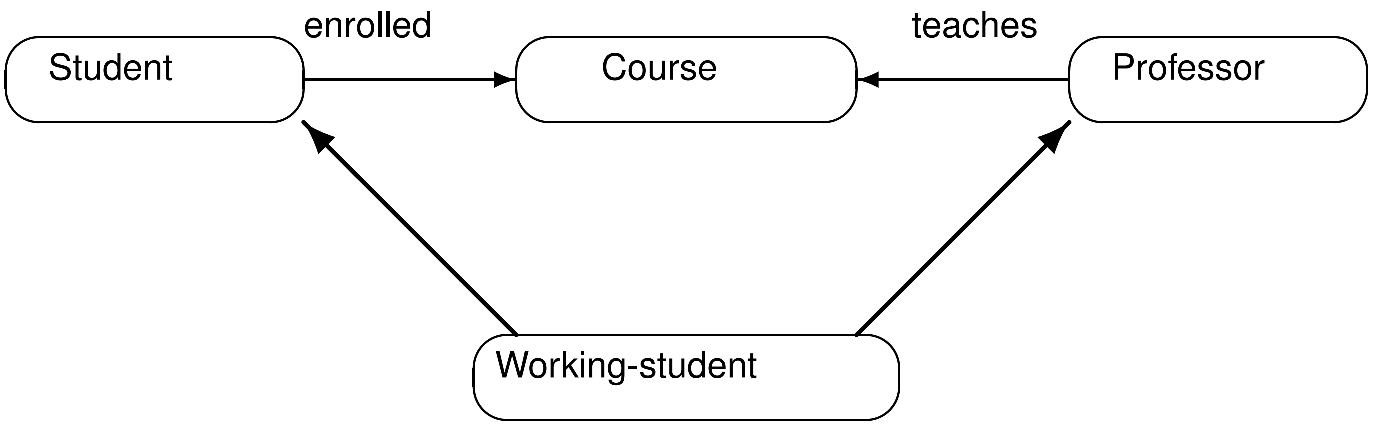
$$\begin{aligned}\texttt{Student} \doteq\ & \texttt{Person} \sqcap \\ & \exists \texttt{NAME}.\texttt{String} \sqcap \\ & \exists \texttt{ADDRESS}.\texttt{String} \sqcap \\ & \exists \texttt{ENROLLED}.\texttt{Course}\end{aligned}$$

Objects: instances

s1: Student	
name:	"John"
address:	"Abbey Road. . ."
enrolled:	cs415

$$\texttt{Student}(\texttt{s1}) \wedge$$
$$\texttt{NAME}(\texttt{s1}, \text{``}\texttt{john}\text{''}) \wedge \texttt{String}(\text{``}\texttt{john}\text{''}) \wedge$$
$$\texttt{ADDRESS}(\texttt{s1}, \text{``}\texttt{abbey-road}\text{''}) \wedge \texttt{String}(\text{``}\texttt{abbey-road}\text{''}) \wedge$$
$$\texttt{ENROLLED}(\texttt{s1}, \texttt{cs415}) \wedge \texttt{Course}(\texttt{cs415})$$

Semantic Networks



$\forall x.\ \texttt{Student}(x) \rightarrow$
$\quad \exists y.\ \texttt{ENROLLED}(x, y) \wedge \texttt{Course}(y)$

$\forall x.\ \texttt{Professor}(x) \rightarrow$
$\quad \exists y.\ \texttt{TEACHES}(x, y) \wedge \texttt{Course}(y)$

$\forall x.\ \texttt{Working-student}(x) \rightarrow$
$\quad \texttt{Student}(x) \wedge \texttt{Professor}(x)$

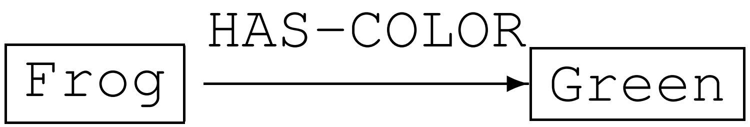
$\texttt{Student} \sqsubseteq \exists\texttt{ENROLLED.Course}$

$\texttt{Professor} \sqsubseteq \exists\texttt{TEACHES.Course}$

$\texttt{Working-student} \sqsubseteq \texttt{Student}$

$\texttt{Working-student} \sqsubseteq \texttt{Professor}$

Quantification



- $\mathtt{Frog} \sqsubseteq \exists \mathtt{HAS{-}COLOR.Green}$:

  Every frog is also green

- $\mathtt{Frog} \sqsubseteq \forall \mathtt{HAS{-}COLOR.Green}$:

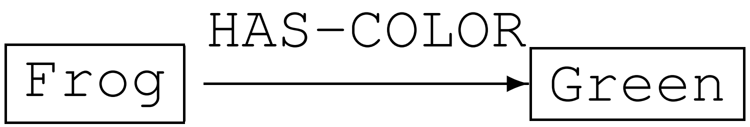
  Every frog is just green

- $\mathtt{Frog} \sqsubseteq \forall \mathtt{HAS{-}COLOR.Green}$

  $\mathtt{Frog}(x),\ \mathtt{HAS{-}COLOR}(x, y)$:

  There is a frog, which is just green

Quantification: existential



Every frog is also green

$$\texttt{Frog} \sqsubseteq \exists \texttt{HAS−COLOR.Green}$$

$$\forall x.\ \texttt{Frog}(x) \rightarrow$$
$$\exists y.\ (\texttt{HAS−COLOR}(x, y) \wedge \texttt{Green}(y))$$

Exercise: is this a model?

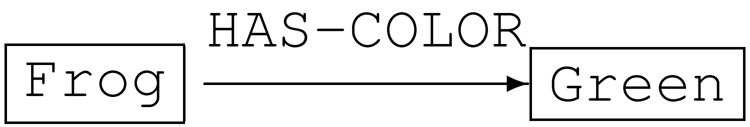
Frog(*oscar*), Green(*green*),

HAS−COLOR(*oscar*,*green*),

Red(*red*),

HAS−COLOR(*oscar*,*red*).

Quantification: universal



Every frog is only green

$\mathtt{Frog} \sqsubseteq \forall \mathtt{HAS{-}COLOR}.\mathtt{Green}$

$\forall x.\ \mathtt{Frog}(x) \rightarrow$
$\quad \forall y.\ (\mathtt{HAS{-}COLOR}(x, y) \rightarrow \mathtt{Green}(y))$

Exercise: is this a model?

Frog(*oscar*), Green(*green*),

HAS−COLOR(*oscar*,*green*),

Red(*red*),

HAS−COLOR(*oscar*,*red*).

and this?

Frog(*sing*),

AGENT(*sing*,*oscar*).

Analytic reasoning (intuition)

Person

subsumes

(Person **with every** male friend **is a** doctor)

subsumes

(Person **with every** friend **is a**

(Doctor **with a** specialty **is** surgery))

Analytic reasoning (intuition)

Person

subsumes

(Person **with every** male friend **is a** doctor)

subsumes

(Person **with every** friend **is a**

(Doctor **with a** specialty **is** surgery))

(Person **with** $\geq$ 2 children)

subsumes

(Person **with** $\geq$ 3 male children)

Analytic reasoning (intuition)

Person

subsumes

(Person **with every** male friend **is a** doctor)

subsumes

(Person **with every** friend **is a**

(Doctor **with a** specialty **is** surgery))

(Person **with** $\geq$ 2 children)

subsumes

(Person **with** $\geq$ 3 male children)

(Person **with** $\geq$ 3 young children)

disjoint

(Person **with** $\leq$ 2 children)