Object-Oriented Representation

- Knowledge as structured and organized in terms of what the knowledge is about, the objects of knowledge.
- Objects with parts, constraints
- Frame (Minsky 1975)

Frames

- Individual Frames – to represent single objects
- Generic Frames – to represent categories of objects.
- slots, fillers

```
(Frame-name
  <slot-name1 filler1>
  <slot-name2 filler2>
....)
```

Individual Frames: Example

```
(tripLeg123
  <:INSTANCE-OF  TripLeg>
  <:Destination toronto>..)

(toronto
  <:INSTANCE-OF  CanadianCity>
  <:Province ontario>
  <:Population 4.5M>..)
```

Generic Frames: Example

```
(CanadianCity
  <:IS-A  City>
  <:Province CanadianProvince>
  <:County canada>..)

(Table
  <:Clearance [IF-NEEDED  ComputeClearanceFromLegs
  ...)

(Lecture
  <:DayOfWeek  WeekDay>
  <:Date [IF-ADDED  ComputeDayofWeek]>
  ..)
```

Inheritance

```
(Table
  <:Clearance [IF-NEEDED  ComputeClearanceFromLeg
  ...)

(CoffeeTable
  <:IS-A  Table>...)

(MahoganyCoffeeTable
  <:IS-A CoffeeTable>..)
```

Inheritance (defaults)

```
(Elephant
  <:IS-A  Mammal>
  <:EarSize large>
  <:Color gray>...)

(raja
  <:INSTANCE-OF Elephant>
  <:EarSize small>..)

(RoyalElephant
  <:IS-A  Elephant>
  <:Color white>...)

(clyde
  <:INSTANCE-OF RoyalElephant>
  ..)
```

Reasoning with Frames

1. a user or external system declares that an object exists, thereby instantiating some generic frame;

2. any slot fillers that are not provided explicitly, but can be inherited, are computd;

3. for each slit with a filler, any **IF-ADDED** procedure that can be inherited is run, possibly causing new slots to be filled, or new frames to be instantiated, and the cycle repeats.

Description Logics

- Objects fall into classes.
- Some classes are more general than others.
- Objects have parts.
- *concepts*, *roles constants*

A Description Language DL

Logical Symbols

1. *punctuation:* "[", "]", "(", ")";
2. *positive integers:* 1, 2, 3, etc.
3. *concept-forming operators:*

   "**ALL**", "**EXISTS**", "**FILLS**", "**AND**";
4. *connectives:* "$\sqsubseteq$", "$\doteq$", "$\rightarrow$".

A Description Language DL (cont)

Non-Logical Symbols

1. *atomic concepts:* written in capitalized mized case, e.g., `Person`, `WhiteWine`, `FatherOfOnlyDaughters`; and a special atomic concept `Thing`.
2. *roles:* written like atomic concepts, but preceded by ":", e.g., `:Child`, `:Height`, `:Employer`, `:Arm`.
3. *constants:* writtein in uncapitalized mixed case, e.g. `table13`, `maryAnnJones`.

Syntactic expressions

There are four types of syntactic expressions:

1. *constants*
2. *roles*
3. *concepts*
4. *sentences*

Concepts

The set of concepts of $\mathcal{DL}$ is the least set satisfying:

- every atomic concept is a concept;
- if r is a role and d is a concept, then $[\mathbf{ALL}\ r\ d\ ]$ is a concept;
- if r is a role and n is a positive integer, then $[\mathbf{EXISTS}\ n\ r]$ is a concept;
- if r is a role and c is a constant then, $[\mathbf{FILLS}\ r\ c]$ is a concept;
- if $d_1 \ldots d_n$ are concepts, then $[\mathbf{AND}\ d_1 \ldots d_n]$ is a concept;

Sentences

- if d_1 and d_2 are concepts, then $[d_1 \sqsubseteq d_2]$ is a sentence;
- if d_1 and d_2 are concepts, then $[d_1 \doteq d_2]$ is a sentence;
- if c is a constant and d a concept, then $[c \rightarrow d]$ is a sentence.

Examples: Concepts

$$[\textbf{EXISTS}\ n\ r]$$

$$[\textbf{EXISTS}\ 1\ \texttt{:Child}]$$

$$[\textbf{FILLS}\ r\ c]$$

$$[\textbf{EXISTS}\ \texttt{:Cousin}\ \texttt{vinny}]$$

$$[\textbf{ALL}\ r\ d]$$

$$[\textbf{ALL}\ \texttt{:Employee}\ \texttt{UnionMember}]$$

Examples (cont)

[**AND**`Wine`

[**FILLS** `:Color red`]

[**EXISTS** 2 `:GrapeType`]

Examples: Sentences

$$(d_1 \sqsubseteq d_2)$$

$$(\texttt{Surgeon} \sqsubseteq \texttt{Doctor})$$

$$(d_1 \doteq d_2)$$

$$(d_1 \rightarrow d_2)$$

$$\begin{aligned}
&(\texttt{ProgressiveCompany} \doteq \\
&\qquad [\textbf{AND}\texttt{Company} \\
&\qquad\quad [\textbf{EXISTS}\ 7 : \texttt{Director}] \\
&\qquad\quad [\textbf{ALL}\ 7 : \texttt{Manager}[\textbf{AND}\ \texttt{Woman} \\
&\qquad\qquad [\textbf{FILLS} : \texttt{Degree phD}]]] \\
&\qquad\quad [\textbf{FILLS}\text{: } \texttt{MinSalary}\ \$24.00/hour]])
\end{aligned}$$

Semantics

An interpretation $\Im$ for DL is a pair

$$\langle \mathcal{D}, \mathcal{I} \rangle$$

where D is any set of objects called the *domain* of the interpretation

and

I is a mapping called the *interpretation mapping* from the non-logical symbols of DL to elements and relations over D.

Semantics (cont)

where

1. for every constant symbol c, $\mathcal{I}[c] \in \mathcal{D}$;
2. for every atomic concept a, $\mathcal{I}[a] \subseteq \mathcal{D}$;
3. for every role symbol r, $\mathcal{I}[c] \subseteq \mathcal{D} \times \mathcal{D}$;

Extending I

- $\mathcal{I}[\texttt{thing}]$
- $\mathcal{I}[\textbf{ALL}\ r\ d]$
- $\mathcal{I}[\textbf{EXISTS}\ n\ r]$
- $\mathcal{I}[\textbf{FILLS}\ r\ c]$
- $\mathcal{I}[\textbf{AND} d_1 \ldots d_n]$

Truth in an Interpretation

Given an interpretation $\Im$, we say that α is true in $\Im$, or $\Im \models \alpha$ according to the following rules.

1. $\Im \models (c \rightarrow d)$ iff $\mathcal{I}[c] \in \mathcal{I}[d]$;
2. $\Im \models (d \sqsubseteq d')$ iff $\mathcal{I}[d] \subseteq \mathcal{I}[d']$;
3. $\Im \models (d \doteq d')$ iff $\mathcal{I}[d] = \mathcal{I}[d']$;

Assuming that d and d' are concepts, and c is a constant.