Choosing Objects and Relations

How to represent: “Pen #7 is red.”
Choosing Objects and Relations

How to represent: “Pen #7 is red.”

$\text{red}(\text{pen}_7)$. It’s easy to ask “What’s red?”

Can’t ask “what is the color of $\text{pen}_7$?”
Choosing Objects and Relations

How to represent: “Pen #7 is red.”

\texttt{red(\textit{pen_7})}. It’s easy to ask “What’s red?”
Can’t ask “what is the color of \textit{pen_7}?”

\texttt{color(\textit{pen_7}, \texttt{red})}. It’s easy to ask “What’s red?”
It’s easy to ask “What is the color of \textit{pen_7}?”
Can’t ask “What property of \textit{pen_7} has value \texttt{red}?”
Choosing Objects and Relations

How to represent: “Pen #7 is red.”

\[ \text{red} (pen_7). \] It’s easy to ask “What’s red?”
Can’t ask “what is the color of pen_7?”

\[ \text{color} (pen_7, \text{red}). \] It’s easy to ask “What’s red?”
It’s easy to ask “What is the color of pen_7?”
Can’t ask “What property of pen_7 has value \text{red}?”

\[ \text{prop} (pen_7, \text{color}, \text{red}). \] It’s easy to ask all these questions.
Choosing Objects and Relations

How to represent: “Pen #7 is red.”

\[ \text{red(pen}_7\text{).} \]  It’s easy to ask “What’s red?”
Can’t ask “what is the color of pen\text{\textsubscript{7}?}”

\[ \text{color(pen}_7\text{, red).} \]  It’s easy to ask “What’s red?”
It’s easy to ask “What is the color of pen\text{\textsubscript{7}?}”
Can’t ask “What property of pen\text{\textsubscript{7}} has value red?”

\[ \text{prop(pen}_7\text{, color, red).} \]  It’s easy to ask all these questions.

\[ \text{prop(Object, Attribute, Value) is the only relation needed:} \]
\[ \text{object-attribute-value representation} \]
Universality of \textit{prop}

To represent “\(a\) is a parcel”

\begin{itemize}
\item \textit{prop}(\(a\), \textit{is}_\textit{a}, \textit{parcel})\), where \textit{is}_\textit{a} is a special attribute
\item \textit{prop}(\(a\), \textit{parcel}, \textit{true})\), where \textit{parcel} is a Boolean attribute
\end{itemize}
To represent $scheduled(cs422, 2, 1030, cc208)$. “section 2 of course $cs422$ is scheduled at 10:30 in room $cc208$.”

Let $b123$ name the booking:

- $prop(b123, course, cs422)$.
- $prop(b123, section, 2)$.
- $prop(b123, time, 1030)$.
- $prop(b123, room, cc208)$.

We have reified the booking.

Reify means: to make into an object.
Semantics Networks

When you only have one relation, *prop*, it can be omitted without loss of information.

Write

\[ prop(Obj, Att, Value) \]

as

\[ Obj \xrightarrow{Att} Val \]
An Example Semantic Network

- building: comp_sci
- room: r107
- room: r117
- owned_by: craig
- deliver_to: ming
- room: building
- room: building
- model: lemon_laptop_10000
- brand: lemon_computer
- logo: lemon_disc
- packing: cardboard_box
- weight: light
- size: medium
- color: brown
Equivalent Logic Program

\[ \text{prop}(\text{comp}_2347, \text{owned_by}, \text{craig}). \]
\[ \text{prop}(\text{comp}_2347, \text{deliver_to}, \text{ming}). \]
\[ \text{prop}(\text{comp}_2347, \text{model}, \text{lemon_laptop}_{10000}). \]
\[ \text{prop}(\text{comp}_2347, \text{brand}, \text{lemon_computer}). \]
\[ \text{prop}(\text{comp}_2347, \text{logo}, \text{lemon_disc}). \]
\[ \text{prop}(\text{comp}_2347, \text{color}, \text{brown}). \]
\[ \text{prop}(\text{craig}, \text{room}, r_{107}). \]
\[ \text{prop}(r_{107}, \text{building}, \text{comp_sci}). \]
Frames

The properties and values for a single object can be grouped together into a frame.

We can write this as a list of attribute = value or slot = filler.

```
[owned_by = craig,
 deliver_to = ming,
 model = lemon_laptop_10000,
 brand = lemon_computer,
 logo = lemon_disc,
 color = brown,
 ⋮]
```