RECAP
The Agent in its Environment

- **Agent has a representation**
  - a sentence in a formal language
- **corresponding to a real-world relationship**
  - via the semantics of the language
- **Agent uses the rep to make decisions**
BASE CASE
Representation

- **Representation is an atomic formula:**
  \[ p(t_1, \ldots, t_n) \]
  - predicate \( p \)
  - constants \( t_1, \ldots, t_n \)
  - constants represent objects in the world
  - predicate represents relation among them
  - written out for I/O with designer; really

BASE CASE
Semantics

- **Designer thinks up and specifies a model**
  - identifies objects and relations in the world needed to solve the problem
  - associates each constant with an object
  - associates each predicate with a relation

- **This determines**
  - what an atomic formula means
  - whether an agent’s rep matches the world
BASE CASE
Formal Semantics

- To study representations, we formalize:
  - Objects: universe or domain \( D \)
  - Consts: map \( \phi \) from const to \( D \)
  - Preds: map \( \pi \) from n-ary pred to \( D^n \rightarrow \{T,F\} \)
    Interpretation: \( \langle D, \phi, \pi \rangle \)
- \( p(t_1,\ldots,t_n) \) is true in interpretation iff
  \[ \pi(p)\langle\phi(t_1),\ldots,\phi(t_n)\rangle = T \]

VARIABLES AND RULES
Representations

- Terms are either constants or variables
  - range over elements in the universe
    (strings beginning with caps or _)
- Extended atoms:
  \[ p(t_1,\ldots,t_n) \]
  predicate terms
  - expression without variables is ground
VARIABLES AND RULES
(Formal) Semantics

• **Variables interpreted by assignment**
  – temporarily links each variable to an object
  – formally, map $\rho$ from var to $D$

\[ \delta(t_1, \rho) = \begin{cases} 
\phi(t_1) & \text{if } t_1 \text{ is a constant} \\
\rho(t_1) & \text{if } t_1 \text{ is a variable} 
\end{cases} \]

• $p(t_1, \ldots, t_n)$ is **true at $\rho$ iff**

\[ \pi(p)(\delta(t_1, \rho), \ldots, \delta(t_n, \rho)) = T \]

VARIABLES AND RULES
Representations

• **Rules** are a kind of compound formula

\[ \underbrace{h \leftarrow b_1 \land \cdots \land b_n}_{\text{head}} \leftarrow \underbrace{b_1 \land \cdots \land b_n}_{\text{body}} \]

  – head and body are atomic formulas
  – meaning: head is true whenever body is
VARIABLES AND RULES
(Formal) Semantics

- \( h \leftarrow b_1 \land \ldots \land b_n \) is true at \( \rho \) iff
  - either \( h \) is true at \( \rho \)
  - or some \( b_i \) is not true at \( \rho \)

- **definite clause** \( f \) is either rule or atom
  - \( f \) is true iff
    - for all \( \rho \), \( f \) is true at \( \rho \)

- **knowledge base** \( K \): set of definite clauses
  - \( K \) is true iff every \( f \in K \) is true

---

USING SEMANTICS
Entailment

- Formalization of information in KB
- \( K \models f \) (read “\( K \) entails \( f \)”) iff
  - every interpretation where \( K \) is true is an interpretation where \( f \) is also true
Proof

- **For us, a proof is a data structure**
  - that describes why a knowledge base entails some fact.
- **To describe the data structure**
  - A substitution $\sigma$ is a finite set of the form
    \[
    \{ V_1 / t_1, \ldots, V_n / t_n \}
    \]
  - The application of $\sigma$ to $e$, $e\sigma$, is $e$ with occurrences of $V_i$ replaced by $t_i$.

Proof

- **A proof is a tree of judgments**
  - A judgment takes the form
    \[
    K \rightarrow f
    \]
    read “$f$ follows from $K$”
  - $K$ is a knowledge base
  - $f$ is a ground clause
- **Leaf is a judgment** $K$, $e \rightarrow e\sigma$
Proof

- **Internal nodes**
  - we can combine proofs together thus:

\[
P_o \quad P_i \quad P_n
\]

\[
\begin{align*}
K & \rightarrow h \leftarrow b_1 \wedge \ldots \wedge b_n \\
K & \rightarrow b_1 \\
& \ldots \\
K & \rightarrow b_n \\
\hline
K & \rightarrow h
\end{align*}
\]

Proof

- When we have

\[
P
\]

\[
K \rightarrow f
\]

we say \( f \) is provable from \( K \), or \( K \vdash f \)
- General result, for ground \( f \):

\[
K \vdash f \iff K \vdash f
\]