A Representation and Reasoning System (RRS) is made up of:

- **formal language:** specifies the legal sentences
- **semantics:** specifies the meaning of the symbols
- **reasoning theory or proof procedure:** nondeterministic specification of how an answer can be produced.
Implementation of an RRS

An implementation of an RRS consists of

➤ **language parser:** maps sentences of the language into data structures.

➤ **reasoning procedure:** implementation of reasoning theory + search strategy.

Note: the semantics aren’t reflected in the implementation!
Using an RRS

1. Begin with a task domain.

2. Distinguish those things you want to talk about (the ontology).

3. Choose symbols in the computer to denote objects and relations.

4. Tell the system knowledge about the domain.

5. Ask the system questions.
Role of Semantics in an RRS

\[
in(alan, r123).
part_of(r123, cs\_building).
in(X, Y) \leftarrow \quad part_of(Z, Y) \land \quad in(X, Z).
\]

© David Poole, Alan Mackworth, Randy Goebel, and Oxford University Press 1999
Simplifying Assumptions of Initial RRS

An agent’s knowledge can be usefully described in terms of *individuals* and *relations* among individuals.

An agent’s knowledge base consists of *definite* and *positive* statements.

The environment is *static*.

There are only a finite number of individuals of interest in the domain. Each individual can be given a unique name.

⇒ Datalog
Syntax of Datalog

**variable** starts with upper-case letter.

**constant** starts with lower-case letter or is a sequence of digits (numeral).

**predicate symbol** starts with lower-case letter.

**term** is either a variable or a constant.

**atomic symbol** (atom) is of the form $p$ or $p(t_1, \ldots, t_n)$ where $p$ is a predicate symbol and $t_i$ are terms.
Syntax of Datalog (cont)

**definite clause** is either an atomic symbol (a fact) or of the form:

$$a \leftarrow b_1 \land \cdots \land b_m$$

where $a$ and $b_i$ are atomic symbols.

**query** is of the form $\Box b_1 \land \cdots \land b_m$.

**knowledge base** is a set of definite clauses.
Example Knowledge Base

\[ in(alan, R) \leftarrow \]
\[ teaches(alan, cs322) \land \]
\[ in(cs322, R). \]
\[ grandfather(william, X) \leftarrow \]
\[ father(william, Y) \land \]
\[ parent(Y, X). \]
\[ slithy(toves) \leftarrow \]
\[ mimsy \land borogroves \land \]
\[ outgrabe(mome, Raths). \]