Knowledge-Based Agents

Dr. Collin F. Lynch

AI Academy: North Carolina State University

AI Academy NC STATE UNIVERSITY

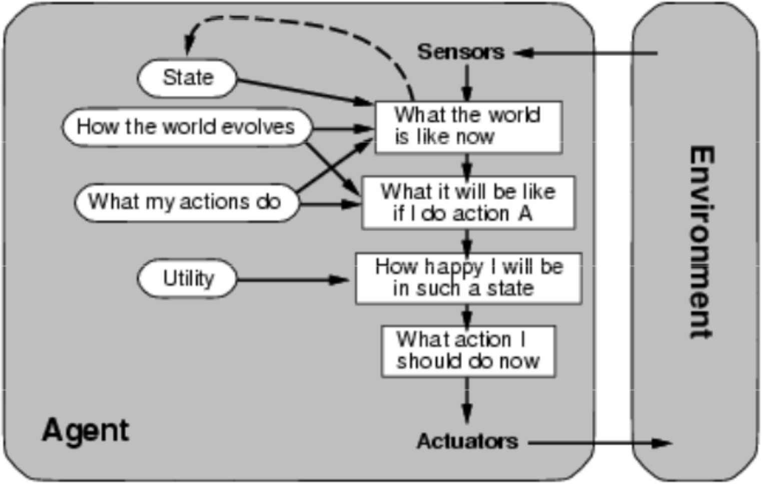
Agenda

Knowledge-Based Agents

Representations

- Atomic (Search)
- Factored (Constraints)
- **Structured** (TBD)

An Agent

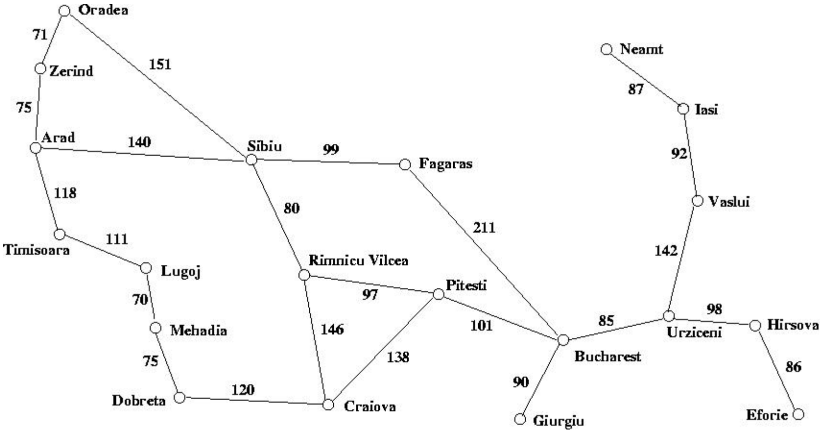


Sensors
State
How the world evolves
What my actions do
Utility
What the world
is like now
What it will be like
if I do action A
How happy I will be
in such a state
What action I
should do now
Agent
Actuators
Environment

The Goal

- Domain-General *Reasoning*,
- Applied to Domain-Specific *Knowledge*
- (In a Domain-General *Representation*)

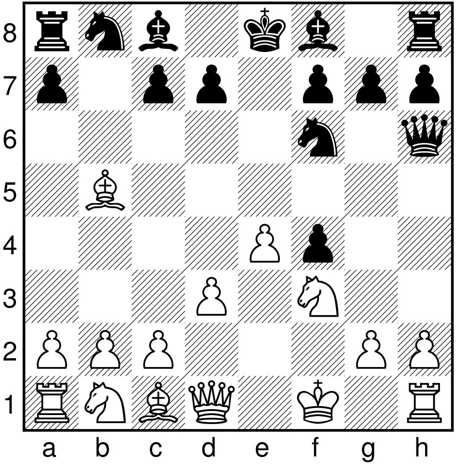
Implicit


Oradea
71
Zerind
151
75
Arad
140
Sibiu
99
Fagaras
118
80
211
Timisoara
111
Lugoj
Rimnicu Vilcea
97
Pitesti
70
Mehadia
146
101
75
138
Dobreta
120
Craiova
Neamt
87
Iasi
92
Vaslui
142
85
Urziceni
98
Hirsova
Bucharest
90
Giurgiu
86
Eforie

Implicit

```
road(oradea,    zerind,     71).
road(zerind,    arad,       75).

city(arad,      46.16667, 21.3).
city(bucharest, 44.41667, 26.1).

twoWayRoad(City1, City2, Distance):-
       road(City1, City2, Distance).
twoWayRoad(City1, City2, Distance):-
      road(City2, City1, Distance).

move(City1, City2):- twoWayRoad(City1, City2, _).

move(City1, City2, Distance):-
      twoWayRoad(City1, City2, Distance).
```

Chess Examples



1. C3.is clear
2. B3.is clear
3. A3.is clear
4. WKn1 in b1
5. Can move WKn1 to C3
6. ¬ can move WQ to C3

Types of Reasoning

- Knowledge Representations:
  - Logic: propositional, predicate, temporal, fuzzy, probabilistic, etc. etc.
  - Networks of all kinds, including so-called Semantic Networks
  - Frames
  - Semantic Web: RDF, OWL, XML, etc.
  - English, Swahili, Urdu, etc. (for humans).
- Reasoning:
  - Logical inference
  - Temporal reasoning
  - Graph/subgraph matching
  - Analogical reasoning
  - Probabilistic inference

Assessment Criteria

- Assessment Criteria
  - **Representational Adequacy**: the ability to represent *sufficient* kinds of knowledge in the domain.
  - **Acquisitional Efficiency**: the ability to *easily* and *quickly* represent new knowledge in the formalism.
- Reasoning Methods
  - **Inferential Adequacy**: the ability to manipulate representational structures to derive sufficient new ones
  - **Inferential Efficiency**: the ability to derive new representational structures quickly

KB-Based Agent: Schema

```
class KB_Agent:
    Knowledge_Base = {...}
    Curr_Time      = 0

    def get_action(Percept):
        Percept_Seq = make_percept_sequence(
            Percept, self.Curr_time)
        tell(self.Knowledge_Base, Percept_Seq)
        Query = make_action_query(self.Curr_Time)
        Act = ask(self.Knowledge_Base, Query)
        Act_Seq = make_action_sequence(
            Act, self.Curr_Time)
        tell(self.Knowledge_Base, Act_Seq)
        self.Curr_Time += 1
        return Act
```

Expert Systems

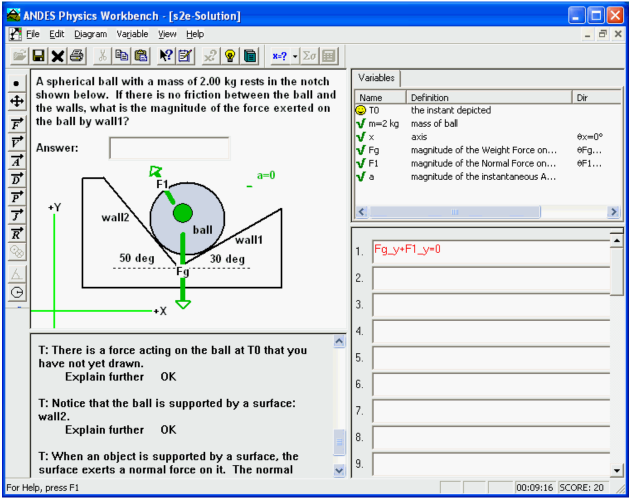


Image Credit: ubiquity.acm.org

Core Concepts

The basics

- *Statements* Individual facts, percepts, or information.
- *Knowledge* The *integration* of those statements to support reasoning.
- *Inference* The *induction* of new knowledge or *decisionmaking*.

Andes

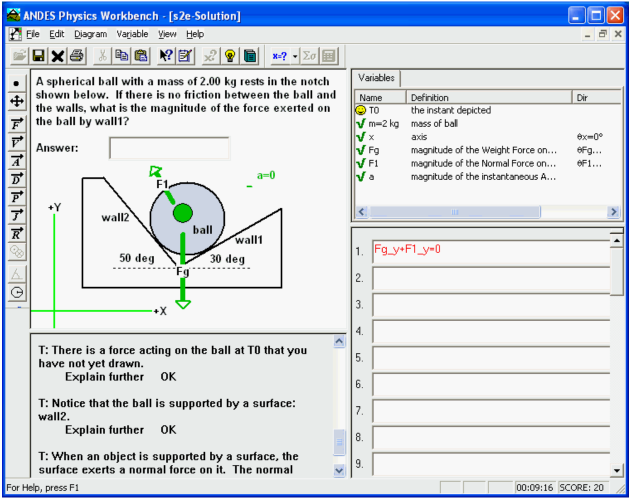


Image Credit: ubiquity.acm.org

Problem

A spherical ball with a mass of 2.00 kg rests in the notch shown below. If there is no friction between the ball and the walls, what is the magnitude of the force exerted on the ball by wall1?

KB

- `(DRAW-AXES 40)`
- `(EQN (= |Yc_Fn_BALL_WALL2_1_40| 0))`
- `(IMPLICIT-EQN (= |OFn_BALL_WALL2_1| (DNUM 40 |deg|)))`
- `(VECTOR (FORCE BALL WALL2 NORMAL :TIME 1) (DNUM 40 |deg|))`
- `(DEFINE-VAR (MASS BALL))`
- `(EQN (= |Yc_Fw_BALL_EARTH_1_40| (* |Fw_BALL_EARTH_1| (SIN (- (DNUM 270 |deg|) (DNUM 40 |deg|))))))`

The basics

- **Sentences** (*statements*) in the knowledge base.
- **Syntax**: rules for well formed sentences.
- **Semantics**: rules for the meaning of the sentence or *truth in each possible world*.
- Sentences must be true *XOR* false in each possible world.
- **Model**: A fixed assignment of values to all possible sentences (possible world).
- A sentence S **satisfies** a model if it is true in it.
- M **is a model of S** if S satisfies it.

Propositional Logic

Basic Elements

- Set P of *propositional symbols*: p, q, r, etc.
- Two *truth values*: true, false
- *Logical Symbols*: $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\Box$
- *Punctuation*: ( ), [ ], { }.

Propositions

- A symbol: a
- a negated proposition: $\neg P$
- Conjunction or Disjunction: $P_1 \wedge P_2$, $P_1 \vee P_2$
- An implication: $P_1 \Rightarrow P_2$
- A logical equivalence: $P_1 \Leftrightarrow P_2$.
- Note: Implication is not causation!

Semantics

- An *interpretation* is a specification of values (t—f) for all symbols:
$$m_i = \{s_0 = t, s_1 = f, \ldots\} \quad (1)$$
- Sentence values are constructed *recursively* from symbols up.
- Or stored in truth-tables.
- A model for a proposition (p) is an interpretation in which p is true.
- A proposition p is *valid* if for any interpretation I, $I \models p$ i.e., every interpretation is a model.

Model Checking

- p is *satisfiable* if ∃ an I s.t. $I \models p$.
- p is *unsatisfiable* no valid model can be found.
- Theorem proving can be handled via DFS or rule algorithms.
- What is wrong with that?

Theorem Proving

- *Logical Equivalence* $a \equiv b$ iff $(a \models b) \wedge (b \models a)$.
- *Valid* (Tautology) $\forall m : m \models p$
- *Deduction Theorem*: $(a \models b)$ iff $(a \rightarrow b)$.
- Thus: $(a \models b)$ iff $(a \wedge \neg b)$ is unsatisfiable.

Resolution

- Convert all Knowledge to CNF $((a \vee b \vee \ldots) \wedge (c \vee \ldots) \wedge \ldots)$
- *Resolve* conflicting clauses to yield new information

$$\frac{(a \vee b \vee \neg c) \wedge (f \vee c \vee \neg q)}{(a \vee b \vee f \vee \neg q)}$$

- This process is sound and complete.

Definite & Horn Clauses

- Resolution is powerful but unnecessary with *Horn Clauses*
- At most one variable is positive: $(x \vee \neg y \vee \ldots)$
- These can be handled with implications $(p \wedge q) \rightarrow r$
- Can do Forward-chaining *data-driven* reasoning.
- And backward-chaining *goal-directed* reasoning.

Example: Chemistry

$$
\begin{aligned}
MgO + H2 &\rightarrow Mg + H2O &(2)\\
C + O2 &\rightarrow CO2 &(3)\\
CO2 + H2O &\rightarrow H2CO3 &(4)\\
& &(5)
\end{aligned}
$$

Assuming sufficient quantities of MgO, H2, C, and O2, prove that we can synthesize H2CO3.

Example: Chemistry (Propositional)

$$
\begin{aligned}
MgO \wedge H2 &\Rightarrow Mg \wedge H2O && (6)\\
C \wedge O2 &\Rightarrow CO2 && (7)\\
CO2 \wedge H2O &\Rightarrow H2CO3 && (8)\\
MgO & && (9)\\
H2 & && (10)\\
C & && (11)\\
O2 & && (12)\\
\texttt{Conclusion} & \quad H2CO3 && (13)
\end{aligned}
$$

Example: Chemistry (CNF)

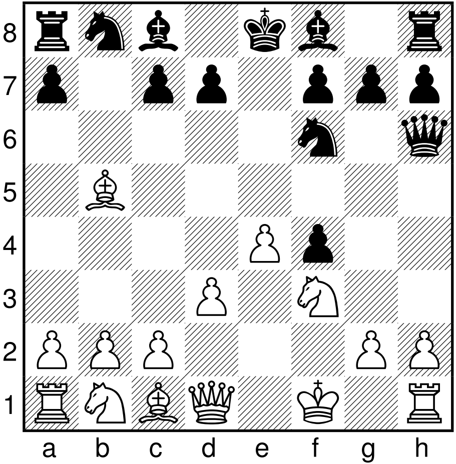
1. $\neg MgO \vee \neg H2 \vee Mg$
2. $\neg MgO \vee \neg H2 \vee H2O$
3. $\neg C \vee \neg O2 \vee CO2$
4. $\neg CO2 \vee \neg H2O \vee H2CO3$
5. MgO
6. $H2$
7. C
8. $O2$
9. $\neg H2CO3$ `Negated conclusion`

Example: Chemistry (CNF)

10.	$\neg CO2 \vee \neg H2O$	$4+9$
11.	$\neg C \vee \neg O2 \vee \neg H2O$	$3+10$
12.	$\neg O2 \vee \neg H2O$	$7+11$
13.	$\neg H2O$	$8+12$
14.	$\neg MgO \vee \neg H2$	$2+13$
15.	$\neg H2$	$5+14$
16.	$\square$	$6+15$

Planning & Frames

We can even use this to do planning (sort of)



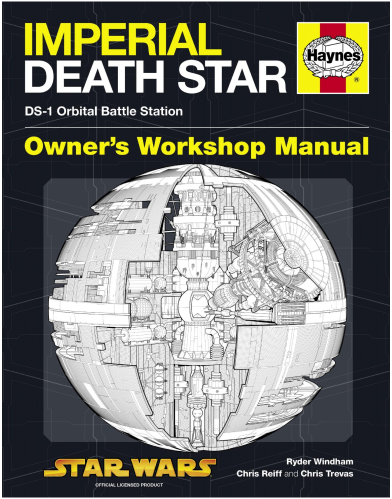
1. $clear^{t1}_{C3}$
2. $clear^{t1}_{B3}$
3. $clear^{t1}_{A3}$
4. $WKn1^{t1}_{b1}$
5. $Move_{Wkn1\,toc3} \Leftrightarrow (WKn1^{t1}_{b1} \wedge clear^{t1}_{C3})$

Debugging

Real Problem: Sherlock

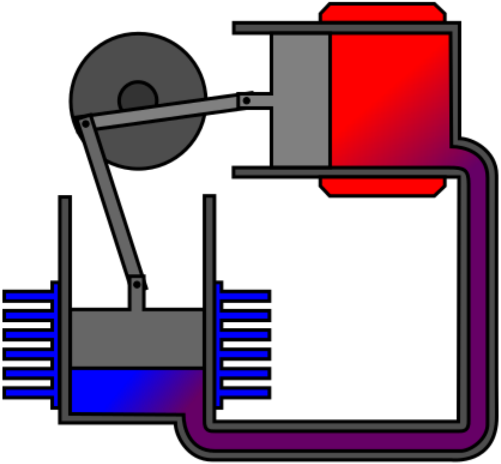
- The Sherlock system: Lesgold & Katz et al.
- Teaches basic diagnostic skills through modeling.
- Uses a logical model of student and system.
- Saved \$$>$500k in one instance.

Service Debugging



Source: http://www.fine1steditions.co.uk

Basic Piston engine.



Source: https://en.wikipedia.org/wiki/File:Alpha_Stirling_frame_12.svg