Knowledge Representation

When we use search to solve a problem we must

Capture the knowledge needed to formalize the problem

Apply a search technique to solve problem

Execute the problem solution

Role of KR

The first step is the role of "knowledge representation" in AI.

Formally,

The intended role of knowledge representation in artificial intelligence is to reduce problems of intelligent action to search problems.

"A good description, developed within the conventions of a good KR, is an open door to problem solving; a bad description, using a bad representation, is a brick wall preventing problem solving."

A Knowledge-Based Agent

We previously talked about applications of search but not about methods of formalizing the problem.

Now we look at extended capabilities to general logical reasoning.

Here is one knowledge representation: logical expressions.

A knowledge-based agent must be able to

- Represent states, actions, etc.
- Incorporate new percepts
- Update internal representations of the world
- Deduce hidden properties about the world
- Deduce appropriate actions

We will

- Describe properties of languages to use for logical reasoning
- Describe techniques for deducing new information from current information
- Apply search to deduce (or learn) specifically needed information

The Wumpus World Environment

Percepts

Percepts

- Percept = [Stench, Breeze, Glitter, Bump, Scream]
  - Stench in wumpus square and adjacent (L, R, U, D) squares
  - Breeze in squares adjacent to pit
  - Glitter in gold square
  - Bump when walk into wall or obstacle
  - Everyone hears scream when wumpus is defeated
  - Agent cannot perceive its own location
- Actions = [goforward, turnleft, turnright, grab, shoot, climb]
- Agent is defeated upon entering room with pit or live wumpus
- Agent's goal is to find gold, return to [1,1], and climb out of cave
- Score (-1) for each action attempted, (-10000) for being defeated, (+1000) for leaving cave with gold
- Enhancements
- Reasoning

Wumpus World Environment Properties

Observable

Partial

Is the world deterministic?

Yes

Episodic

Sequential

Static

Yes (for now), wumpus and pits do not move

Discrete

Yes

Agents

Multi (wumpus, eventually other agents)

Sample Run

Click mouse to advance to next frame.
Now we look at

- How to represent facts / beliefs
  "There is a pit in (2,2) or (3,1)"
- How to make inferences
  "No Breeze in (1,2), so pit in (3,1)"

Representation, Reasoning and Logic

Representation

Sentence

Individual piece of knowledge
English sentence forms one piece of knowledge in English language
Statement in C forms one piece of knowledge in C programming language
Sexp in Lisp forms one piece of knowledge in Lisp programming language

Syntax: form used to represent sentences

Syntax of Lisp indicates legal combinations of symbols
(setf a (+ 2 3)) is legal
(setf a +(2, 3) is not legal)
Syntax alone does not indicate meaning

Semantics: mapping from sentences to facts in the world

They define the truth of a sentence in a world
Add the values of 2 and 3, store them in the memory location indicated by variable a

In the language of arithmetic,
$x + 2 \geq y$ is a sentence

$x2 + y >$ is not a sentence
$x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y
$x + 2 \geq y$ is true in a world where $x = 7, \ y = 1$
$x + 2 \geq y$ is false in a world where $x = 0, \ y = 6$

Entailment

There can exist a relationship between items in the language:

- Sentences "entail" sentences (representation level)
- Facts "follow" from facts (real world)

```
                 Sentences   --------------->  Sentence
                    |           Entails           |
                 s  |                             | s
Representation   e  |                             | e
                 m  |                             | m
                 a  |                             | a
--------------   n  | --------------------------  | n -------------
                 t  |                             | t
World            i  |                             | i
                 c  |                             | c
                 s  |                             | s
                    |           Follows           |
                  Facts  -------------------->  Facts
```

```
Entail / Follow means the new item is true if the old items are true
A collection of sentences, or knowledge base (KB), entail a sentence
   KB |= sentence
KB entails the sentence if and only if the sentence is true in all
worlds where the KB is true
```

Entailment Examples

- KB
  - The Giants won
  - The Reds won

- Entails
  - Either the Giants won or the Reds won
- KB
  - To get a perfect score your program must be turned in today
  - I always get perfect scores
- Entails
  - I turned in my program today
- KB
  - CookLectures $\rightarrow$ TodayIsTuesday $\vee$ TodayIsThursday
  - $\neg$ TodayIsThursday
  - TodayIsSaturday $\rightarrow$ SleepLate
  - Rainy $\rightarrow$ GrassIsWet
  - CookLectures $\vee$ TodayIsSaturday
  - $\neg$ SleepLate
- Which of these are correct entailments?
  - $\neg$ SleepLate
  - GrassIsWet
  - $\neg$ SleepLate $\vee$ GrassIsWet
  - TodayIsTuesday
  - True

Inference

Use two different ways:

1. Generate new sentences that are entailed by KB
2. Determine whether or not sentence is entailed by KB

A sound inference procedure generates only entailed sentences.

```
Modus ponens yes, abduction no
A, A->B             B, A->B
-------             -------
B                   A
```

Definitions

A complete inference procedure can generate all entailed sentences from the knowledge base.

The meaning of a sentence is a mapping onto the "real world".

This mapping is an interpretation (interpretation of Lisp code).

A sentence is valid (necessarily true, tautology) iff true under all possible interpretations.

$$A \vee \neg A$$

A could be:

- Stench at [1,1]
- Today is Monday
- 2+3=5

The above statement is valid. These statements are not valid.

- $A \wedge \neg A$
- $A \vee B$

The last statement is satisfiable , meaning there exists at least one interpretation that makes the statement true. The previous statement is unsatisfiable .

Logics

Logics are formal languages for representing information such that conclusions can be drawn

Logics are characterized by their "primitives" commitments

- Ontological commitment: What exists? Facts? Objects? Time? Beliefs?
- Epistemological commitment: What are the states of knowledge?

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief in 0…1
Fuzzy logic	degree of truth	degree of belief in 0…1

Models

Logicians frequently use models, which are formally structured worlds with respect to which truth can be evaluated.

M is a model of a sentence s if s is true in M.
M(s) is the set of all models of s.
KB entails s if and only if M(KB) $\subseteq$ M(s)
For example, KB = Giants won and Red won, s = Giants won

Examples

Propositional logic Simple logic

Symbols represent entire facts
Boolean connectives `(&, v, ->, <=>, ~)`
Propositions (symbols, facts) are either TRUE or FALSE

First-order logic

Extend propositional logic to include
variables, quantifiers, functions, objects

Propositional Logic

Proposition symbols P, Q, etc., are sentences

The true/false value of propositions and combinations of propositions can be calculated using a truth table

If P and S are sentences, then so are $\neg P,\ P \wedge Q,\ P \vee Q,\ P \rightarrow Q,\ P \leftrightarrow Q$

An interpretation I consists of an assignment of truth values to all proposition symbols I(S)

An interpretation is a logician's word for what is often called a "possible world"
Given 3 proposition symbols P, Q, and R, there are 8 interpretations
Given n proposition symbols, there are 2^n interpretations

Models are then worlds in which a particular sentence is true under at least one interpretation

To determine the truth of a complex statement for I, we can

Substitute I's truth value for every symbol

Use truth tables to reduce the statement to a single truth value.

End result is a single truth value, either True or False.

Propositional Logic

For propositional logic, a row in the truth table is one interpretation

A logic is "monotonic" as long as entailed sentences are preserved as more knowledge is added

P	*Q*	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

Rules of Inference for Propositional Logic

```
    * Modus ponens:   A, A=>B   (antecedent, consequent)
      -------
  B
      All men are mortal (Man => Mortal)
      Socrates is a man  (Man)
      ----------------------------------
      Socrates is mortal (Mortal)


    * And introduction:  A, B
  ----
  A&B

    * Or introduction:   A
  -
  A v B v C v D v ...

    * And elimination:   A&B&C&...&Z
  -----------
  A

    * Double-Negation Elimination:   ~~A
        ---
     A

    * Unit Resolution:  A v B, ~B
 ---------
 A
```

```
Today is either Tuesday or Thursday, Today is not Thursday
----------------------------------------------------------
Today is Tuesday


      * Resolution:  A v B, ~B v C       ~A => B, B => C
     -------------       ---------------
       A v C                ~A => C

        Today is Tuesday or Thursday
Today is not Thursday or tomorrow is Friday
-------------------------------------------
Today is Tuesday or tomorrow is Friday
```

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

Normal Forms

Other approaches to inference use syntactic operations on sentences, often expressed in standardized forms

- Conjunctive Normal Form (CNF)
  $\textit{conjunction}$ of $\underbrace{\textit{disjunctions} \text{ of } \textit{literals}}_{\textit{clauses}}$
  For example, $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$
- Disjunctive Normal Form (DNF)
  $\textit{disjunction}$ of $\underbrace{\textit{conjunctions} \text{ of } \textit{literals}}_{\textit{terms}}$
  For example, $(A \wedge B) \vee (A \wedge \neg C) \vee (A \wedge \neg D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge \neg D)$
- Horn Form (restricted)
  conjunction of *Horn clauses* (clauses with ≤ 1 positive literal)
  For example, $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$
  Often written as a set of implications:
  $B \rightarrow A$ and $(C \wedge D) \rightarrow B$

Proof Methods

Proof methods divide into (roughly) two kinds:

Model checking

- Truth table enumeration (sound and complete for propositional logic) - show that all interpretations in which the left hand side of the rule is true, the right hand side is also true
- heuristic search in model space (sound but incomplete)

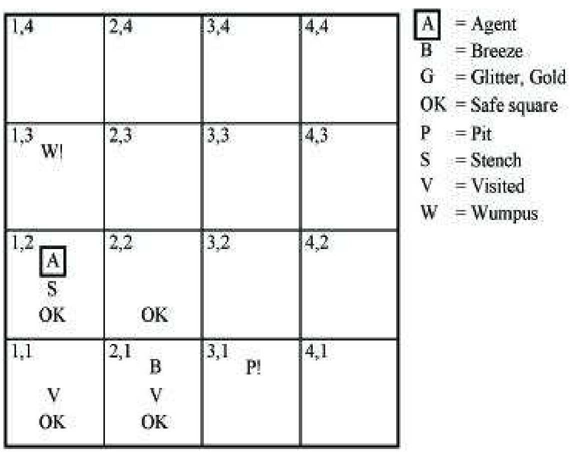
Application of inference rules

- Legitimate (sound) generation of new sentences from old
- Proof = a sequence of inference rule applications
- Can use inference rules as operators in a standard search alg.

An Agent for the Wumpus World

Imagine we are at a stage in the game where we have had the above experiences. What is in our knowledge base?

What can we deduce about the world?




```
    Example: finding the wumpus
      If we are in [1,1] and know
        1) ~S11
2) S12
3) S21
        4) ~S11 => ~W11 & ~W12 & ~W21    ; If no stench, no adj square has wumpus
        5) S12 => W11 v W12 v W13 v W22
        6) S21 => W11 v W21 v W31 v W22

      What can we conclude?
```

Limitations Of Propositional Logic

- Propositional logic cannot express general-purpose knowledge succinctly
- We need 32 sentences to describe the relationship between wumpi and stenches
- We would need another 32 sentences for pits and breezes
- We would need at least 64 sentences to describe the effects of actions
- How would we express the fact that there is only one wumpus?
- Difficulty to identify individuals (Mary, 3)
- Generalizations, patterns, regularities difficulty to represent (all triangles have 3 sides)

First-Order Predicate Calculus

Propositional Logic uses only propositions (symbols representing facts), only possible values are True and False

First-Order Logic includes:

- Objects: peoples, numbers, places, ideas (atoms)
- Relations: relationships between objects (predicates, T/F value)
  Example: father(fred, mary)
- Properties: properties of atoms (predicates, T/F value)
  Example: red(ball)
- Functions: father-of next, (any value in range)

FOPC Models

Example

Express “Socrates is a man” in

Propositional logic: MANSOCRATES – single proposition representing entire idea

First-Order Predicate Calculus: Man(SOCRATES) – predicate representing property of constant SOCRATES

FOPC Syntax

Constant symbols (Capitalized, Constants with no arguments)

Interpretation must map to exactly one object in the world

Predicates (can take arguments, True/False)

Interpretation maps to relationship or property T/F value

Function (can take arguments)

Maps to exactly one object in the world

Definitions

Term

Anything that identifies an object

Function(args)

Constant – function with 0 args

Atomic sentence

Predicate with term arguments

Enemies(WilyCoyote, RoadRunner)

Married(FatherOf(Alex), MotherOf(Alex))

Literals

atomic sentences and negated atomic sentences

Connectives

$\wedge$ `(&)`, $\vee$ `(v)`, $\rightarrow$ `(->)`, $\leftrightarrow$ `(<=>)`, $\neg$ `(~)`

if connected by $\wedge$, conjunction (components are conjuncts)

if connected by $\vee$, disjunction (components are disjuncts)

Quantifiers

Universal Quantifier $\forall$

Existential Quantifier $\exists$

Universal Quantifiers

How do we express "All unicorns speak English" in Propositional Logic?

We would need to specify a proposition for each unicorn

$\forall$ is used to express facts and relationships that we know to be true for all members of a group (objects in the world)

A variable is used in the place of an object

FORALL x, Unicorn(x) $\rightarrow$ SpeakEnglish(x)

The "domain" of x is the world

The "scope" of x is the statement following FORALL (sometimes in [])

```
Same as specifying
   Unicorn(Uni1) => SpeakEnglish(Uni1) &
   Unicorn(Uni2) => SpeakEnglish(Uni2) &
   Unicorn(Uni3) => SpeakEnglish(Uni3) &
   ...
   Unicorn(Table1) => Table(Table1)    &
   ...

   One statement for each object in the world
```

We will leave variables lower case (sometimes ?x)

Notice that x ranges over all objects, not just unicorns.

A term with no variables is a "ground term"

Existential Quantifier

This makes a statement about some object (not named)

$\exists$ x [Bunny(x) $\wedge$ EatsCarrots(x)]

means there exists some object in the world (at least one) for which the statement is true. Same as disjunction over all objects in the world.

```
(Bunny(Bun1) & EatsCarrots(Bun1))   v
(Bunny(Bun2) & EatsCarrots(Bun2))   v
(Bunny(Bun3) & EatsCarrots(Bun3))   v
(Bunny(Bun4) & EatsCarrots(Bun4))   v
(Bunny(Bun5) & EatsCarrots(Bun5))   v
...
(Bunny(Table1) & EatsCarrots(Table1))   v
...
```

```
What about

EXISTS x, Unicorn(x) => SpeakEnglish(x)
?

Means implication applies to at least one object in the universe
```

DeMorgan Rules

- $\forall x \neg P \leftrightarrow \neg \exists x\ P$
- $\forall x\ P \leftrightarrow \neg \exists x \neg P$
- $\neg \forall x \neg P \leftrightarrow \exists x\ P$
- $\neg \forall x\ P \leftrightarrow \exists x \neg P$

- Example: $\forall x$ LovesWatermelon(x) $\leftrightarrow \neg \exists x \neg$LovesWatermelon(x)

Other Properties

- $(X \rightarrow Y) \leftrightarrow \neg X \vee Y$
  Prove with truth table.
- NOT TRUE:
  $(X \rightarrow Y) \leftrightarrow (Y \rightarrow X)$
  This is a type of inference/learning that is not sound (abduction).

Examples

- All men are mortal
  $\forall$ x [Man(x) $\rightarrow$ Mortal(x)]
- Socrates is a man.
  Man(Socrates)
- Socrates is mortal
  Mortal(Socrates)

- All purple mushrooms are poisonous

  $\forall$ x [(Purple(x) $\wedge$ Mushroom(x)) $\rightarrow$ Poisonous(x)]

- A mushroom is poisonous only if it is purple

  $\forall$ x [(Mushroom(x) $\wedge$ Poisonous(x)) $\rightarrow$ Purple(x)]

- No purple mushroom is poisonous

  $\neg$ ($\exists$ x [Purple(x) $\wedge$ Mushroom(x) $\wedge$ Poisonous(x)])

Examples

- There is exactly one mushroom

  $\exists$ x [Mushroom(x) $\wedge$ ($\forall$ y (NEQ(x,y) $\rightarrow$ $\neg$Mushroom(y)))]

  Because "exactly one" is difficult to express we can use $\exists!$ to denote exactly one of a type of object.

- Every city has a dog catcher who has been bitten by every dog in town.
  Use City(1), DogCatcher(1), Bit(2), Lives(2) – the number in parentheses indicates the number of arguments for the function or predicate

  $\forall$ a, b [City(a) $\rightarrow$ $\exists$ c DogCatcher(c) $\wedge$ (Dog(b) $\wedge$ Lives(b,a) $\rightarrow$ Bit(b,c))]

- No human enjoys golf

  $\forall$ x [Human(x) $\rightarrow$ $\neg$ Enjoys(x, Golf)]

- Some professor that is not a historian writes programs

  $\exists$ x [Professor(x) $\wedge$ $\neg$ Historian(x) $\wedge$ Writes(x, Programs)]

- Every boy owns a dog

  $\forall$ x $\exists$ y [Boy(x) $\rightarrow$ Owns(x, y)]
  $\exists$ y $\forall$ x [Boy(x) $\rightarrow$ Owns(x, y)]

  Do these mean the same thing?

- Brothers are siblings

- "Sibling" is reflexive

- One's mother is one's female parent

- A first cousin is a child of a parent's sibling

Higher-Order Logic

FOPC quantifies over objects in the universe.

Higher-order logic quantifies over relations and functions as well as objects.

- All functions with a single argument return a value of 1.

  $\forall$ x, y [Equal(x(y), 1)]

- Two objects are equal iff all properties applied to them are equivalent.

  $\forall$ x, y [(x=y) $\leftrightarrow$ ($\forall$ p[p(x) $\leftrightarrow$ p(y)])]

  Note that we use "=" as a shorthand for equal, meaning they are in fact the same object

---

Example Proof

Known:

1. If x is a parent of y, then y is older than x

   $\forall$ x, y [Parent(x,y) $\rightarrow$ Older(x,y)]

2. If x is the mother of y, then x is a parent of y

   $\forall$ x, y [Mother(x,y) $\rightarrow$ Parent(x,y)]

3. Lulu is the mother of Fifi

   Mother(Lulu, Fifi)

   Prove: Lulu is order than Fifi

   Older(Lulu, Fifi)

4. Parent(Lulu, Fifi)
   2, 3, Modus Ponens

5. Older(Lulu, Fifi)
   1, 4, Modus Ponens

Note that we "bind" the variable to a constant as we apply the rule.
In generating a proof, we have to decide in what order to apply rules.
The order shown here is called "forward chaining".

Example Proof

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

Prove: West is a criminal.

Known:

```
1) FORALL x, y, z [(American(x) & Weapon(y) & Nation(z) & Hostile(z) &
    Sells(x,z,y)) -> Criminal(x)]
   It is a crime for an American to sell weapons to hostile nations

2) EXISTS x Owns(Nono, x) & Missile(x)
   Nono has some missiles

3) FORALL x Owns(Nono,x) & Missile(x) -> Sells(West,Nono,x)
   All of its missiles were sold to it by Colonel West

4) FORALL x Missile(x) -> Weapon(x)

5) FORALL x Enemy(x,America) -> Hostile(x)

6) American(West)

7) Nation(Nono)

8) Enemy(Nono,America)

9) Nation(America)
```

Proof

```
10) Owns(Nono,M1) & Missile(M1)
    2 & Existential Elimination

    Existential Elimination
       EXISTS v []
       Substitute k for v anywhere in sentence, if k does not appear in []
       k is a constant (term with no arguments)

    (Skolemization)

11) Owns(Nono,M1)
```

```
    10 & And-Elimination

12) Missile(M1)
    10 & And-Elimination

13) Missile(M1) => Weapon(M1)
    4 & Universal Elimination

    Universal Elimination
       FORALL v []
       If true for universal variable v, true for a ground term
          (term with no variables g)

14) Weapon(M1)
    12, 13, Modus Ponens

15) Owns(Nono,M1) & Missile(M1) => Sells(West,Nono,M1)
    3 & Universal Elimination

16) Sells(West,Nono,M1)
    10, 15, Modus Ponens

17) American(West) & Weapon(M1) & Nation(Nono) & Hostile(Nono) &
    Wells(West,Nono,M1) => Criminal(West)
    1 & Universal Elimination (x West) (y M1) (z Nono)

18) Enemy(Nono,America) => Hostile(Nono)
    5 & Universal Elimination

19) Hostile(Nono)
    8, 18, Modus Ponens

20) American(West) & Weapon(M1) & Nation(Nono) & Hostile(Nono) &
    Sells(West,Nono,M1)
    6, 7, 14, 16, 19, And-Introduction

21) Criminal(West)
    17, 20, Modus Ponens

Another rule:  Existential Introduction
               If [..g..] true (where g is ground term)
               then EXISTS v [..v..] true (v is substituted for g)
```

Inference As Search

- Operators are inference rules
- States are sets of sentences
- Goal test checks state to see if it contains query sentence

AI, UE, MP a common inference pattern, but generate a huge branching factor
We need a single, more powerful inference rule

Generalized Modus Ponens

If we have a rule

`p1(x) & p2(x) & p3(x, y) & p4(y) & p5(x, y) => q(x, y)`

Each p involves universal / existential quantifiers
And we have each antecedent clause in the knowledge base

```
p1(UTA)
p2(UTA)
p3(UTA, Texas)
p4(Texas)
p5(UTA, Texas)
```

And we can find a way to "match" the variables
Then we can infer q(UTA, Texas)

GMP Example

Rule: `Missile(x) & Owns(Nono,x) => Sells(West,Nono,x)`
Knowledge base contains:
Missile(M1)
Owns(Nono,M1)
Must make sure that instantiations of x are the same
This variable matching process is called unification

FOPC and Wumpus World

Perception rules

- $\forall$b,g,t Percept([Smell, b, g], t) $\rightarrow$ Smelled(t)

  Here we are indicating a Percept at time t

- ∀s,b,t Percept([s,b,Glitter], t) → AtGold(t)

We can use FOPC to write rules for selecting actions:
Reflex Agent: ∀t AtGold(t) → Action(Grab, t)
Reflex Agent With Internal State: ∀t AtGold(t) ∧ ¬Holding(Gold, t) → Action(Grab, t)
Holding(Gold, t) cannot be observed, so keeping track of change is essential

---

Deducing Hidden Properties

Properties of locations:
∀l,t At(Agent,l,t) ∧ Smelled(t) → Smelly(l)
∀l,t At(Agent,l,t) ∧ Breeze(t) → Breezy(l)
Squares are breezy near a pit:
Diagnostic rule—infer cause from effect
∀y Breezy(y) → ∃x Pit(x) ∧ Adjacent(x,y)
Causal rule—infer effect from cause
∀x,y Pit(x) ∧ Adjacent(x,y) → Breezy(y)
Neither of these is complete.
For example, the causal rule doesn't say whether squares far away from pits can be breezy
Definition for the *Breezy* predicate:
∀y Breezy(y) ↔ [∃x Pit(x) ∧ Adjacent(x,y)]

Keeping Track Of Change

Facts hold in situations , rather than eternally
E.g., $Holding(Gold, Now)$ rather than just $Holding(Gold)$
Situation calculus is one way to represent change in FOPC:
Adds a situation argument to each time-dependent predicate
E.g., Now in $Holding(Gold, Now)$ denotes a situation
Situations are connected by the *Result* function
$Result(a, s)$ is the situation that results from doing a in s

---

Describing Actions

Effect axiom—describe changes due to action
∀s AtGold(s) → Holding(Gold, Result(Grab, s))
Frame axiom—describe **non-changes** due to action
∀s HaveArrow(s) → HaveArrow(Result(Grab, s))

Frame problem : find an elegant way to handle non-change
(a) Representation—avoid frame axioms
(b) Inference—avoid repeated "copy-overs" to keep track of state

Qualification problem : true descriptions of real actions require endless caveats – what if gold is slippery or nailed down or ...

Ramification problem : real actions have many secondary consequences – what about the dust on the gold, wear and tear on gloves, ...

Describing Actions

Successor-state axioms solve the representational frame problem

Each axiom is "about" a **predicate** (not an action per se):

$$\begin{aligned} \text{P true afterwards} \quad \leftrightarrow & \quad [\text{an action made P true} \\ \lor & \quad \text{P true already and no action made P false}] \end{aligned}$$

For holding the gold:
∀a,s Holding(Gold, Result(a, s)) ↔
((a = Grab ∧ AtGold(s)) ∨ (Holding(Gold, s) ∧ a ≠ Release))

Making Plans

Initial condition in KB:
$At(Agent, [1, 1], S_0)$
$At(Gold, [1, 2], S_0)$

Query: $\text{Ask}(KB, \exists s Holding(Gold, s))$
i.e., in what situation will I be holding the gold?

Answer: $\{s/Result(Grab, Result(Forward, S_0))\}$
i.e., go forward and then grab the gold

This assumes that the agent is interested in plans starting at S_0 and that S_0 is the only situation described in the KB

Generating Plans: A Better Way

Represent **plans** as action sequences $[a_1, a_2, \ldots, a_n]$

$PlanResult(p, s)$ is the result of executing p in s

Then the query $\text{Ask}(KB, \exists p Holding(Gold, PlanResult(p, S_0)))$
has the solution $\{p/[Forward, Grab]\}$

Definition of $PlanResult$ in terms of $Result$:
∀s PlanResult([], s) = s
∀a,p,s PlanResult([a—p], s) = PlanResult(p, Result(a, s))

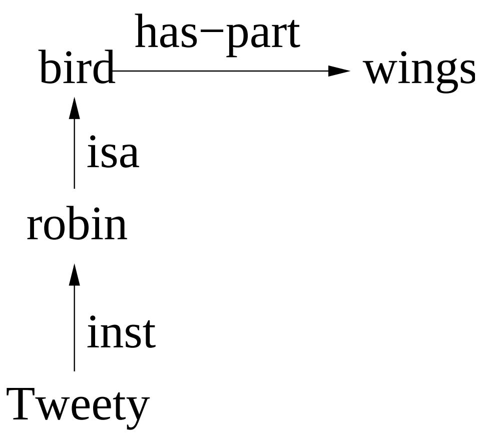
Planning systems are special-purpose reasoners designed to do this type of inference more efficiently than a general-purpose reasoner

Other Common Knowledge Representations

Many problem solving tasks can be characterized as
Knowledge Representation + Search

The choice of KR can dramatically affect the ease of solving the problem.
Propositional logic and FOPC are two common KRs.
Now let's look at others.

Semantic Networks



- A type of OO representation
- Nodes represent objects or concepts in the network
- Edges represent relations between concepts
- The **isa** link is a subset relationship
- The **inst** link is a member relationship
- Property inheritance occurs from being a member or through
- Example: birds have wings, robins are birds, so robins have wings
- Basic inference mechanism: spreading activation
  Inference through network connections between "active" nodes subclass navigation
-

Problems With Semantic Networks

- No formal semantics
- No formal definition of what nodes or arcs represent
- No formal rules for inheritance
- Uniqueness of a concept not easily expressed
- Does not scale well

Frames

Minsky, 1975

- Initially used for visual perception and natural language
- Useful for grouping / chunking information

Frames

A frame is a structure with slots
Slots have facets

- value (declarative)
- range (declarative)
- if-added (procedural)
- if-needed (procedural)

If-added and if-needed slots provide inferential capabilities
Many of the same challenges as semantic networks

Frames and Inheritance

- Frames can be arranged in generalization/specialization hierarchies
- A frame inherits slot faces from more general frames in the hierarchy
- Frames are instantiated for specific instances
- Slots can be filled through different mechanisms
  - Inheritance
  - Defaults
  - Procedural knowledge
  - Explicit knowledge

Frame Example

House
- Address: value: address-frame
- Style: value: A-house-style
- Stores: range: (1-4)
- Rooms: range: (3-20)
- FairPrice: range: (1-5,000,000)
- If=needed: Procedure collects prices of neighboring houses, computes price
- Asking-price: range: (1-5,000,000)
- Age: range: (0-150)
- Condition: value: A-condition
- Lot-size: range: (0-50)

SherwoodForestHouse
- Isa: House
- Style: value: Brick-ranch
- Stores: value: 1
- Rooms: range: (3-8)
- Fair-price: range: (200,000-350,000)
- Age: range: (30-50)
- Condition: value: run-down or remodeled
- Lot-size: value: 1/4

Scripts

Schank and Abelson, 1977

Like frames, but used to represent event sequences

Represent stereotypical situations

Typical slots include roles, header, body, scenes

Conceptual Dependencies are similar to Scripts, used for natural language processing

---

Script Example

Jack went to a restaurant.
He ordered a hamburger.
He paid and left.

What did John eat?

Too many inferences involved, but since we know what typically happens in restaurants, we can use this to control inferences.

Other examples: subway, restaurant, museum, ambulance, hospital, fire department, hurricanes, trainwrecks, earthquakes, trips, weddings, parties

SAM, 1977, used scripts to understand stores, summarize, answer questions

---

Script Difficulties

- How do we select the right script at the right time?
  John went into McDonalds and asked the cashier if he could break a twenty dollar bill.
  Should this use the restaurant script?

- Competition between knowledge sources
  John went with Mary to a restaurant. He asked her for the salt. Then he asked her for her hand.

- Other sources of knowledge
  John needed money. He got a gun.