Intelligent Systems: Reasoning and Recognition

James L. Crowley

ENSIMAG 2 / MoSIG M1 Second Semester 2015/2016

Lesson 18 27 April 2015

Problem Solving, Planning and Structured Knowledge Representations

Bibliography:

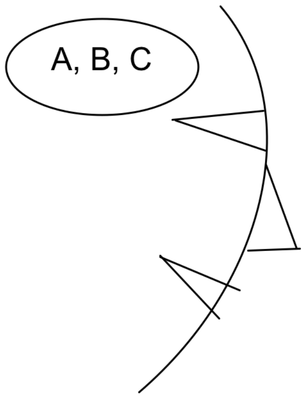
1) Marvin Minsky, A Framework for Representing Knowledge, in: Patrick Henry Winston (ed.), The Psychology of Computer Vision. McGraw-Hill, New York (U.S.A.), 1975.

2) R. Schank and R. Abelson, Scripts, Plans, Goals and Understanding, Erlbaum, 1977.

3) P.N Johnson-Laird, Mental models, MIT Press Cambridge, MA, USA, 1989.

<u>The Intelligent Agent</u>

To provide a formal basis for studying intelligence, in 1970s, Nils Nilsson proposed the Intelligent Agent as a fundamental concept for formalizing intelligence.



The Intelligent Agent is an abstract concept composed of 3 pars: (A, B, C)
A) Actions; The ability to act; A physical body;
B) Goals. (In French "Buts")
C) Knowledge; The ability to choose actions to accomplish goals.

The "Principal of Rationality:
The intelligent agent is assumed to be rational.
The intelligent agent chooses its actions to accomplish its goals.

This was a foundational principal of economics for much of the 20th century.
Nilsson proposed to define intelligence as rationality:

<u>Rational Intelligence</u>: The ability to choose actions to accomplish goals.

An agent is <u>intelligent</u> if it
1) Can act, (embodied)
2) Has goals, (situated) and
3) Can choose its actions to accomplish goals (autonomous).

Rational intelligence leads to a formulation of intelligence as problem solving and planning, formalized using the notion of state space.

<u>Knowledge</u> is whatever enables the agent to choose actions to accomplish goals.

Problem Spaces

Rational intelligence is formalized using a Problem space.

A problem space is defined as

1) A set of states {U} (the Universe),
2) A set of operators for changing states {A} (Actions).

A state is defined using a conjunction of predicates.

A problem is {U}, {A} plus

an initial state $i \in \{U\}$
a set of Goal States $\{G\} \subset \{U\}$

A plan creates a sequence of actions A_1, A_2, A_3, A_4,... that lead from the state S to one of the states $g \in \{G\}$

Thus problem solving is formalized as Planning: The search for a sequence of actions leading to a goal.

The core concept is the notion of "State".

States: A state, s, is a "partial" description of the real universe.

A state is defined as a conjunction of predicates (Truth functions) based on measured (observed) values. The measured values are called "observations". The predicates define relations between observable.s

Examples:

Mobile Robotics: AtPlace (Kitchen)
Blocks World: OnTable (A) ∧ On (A, B) ∧ HandEmpty

The external universe is described as states defined as logical assemblies of predicates concerning observations.

Structured knowledge representations provide a declarative (symbolic) representation for problem spaces.

Blocks World

Blocks world is an abstract, toy world for exploring problems. Blocks world is a "Closed" world. It has a finite number of states.

Blocks world is composed of a finite number of blocks in a finite number of states.

Blocks world is composed of:

- A set of blocks
- An agent that can act on blocks to change their state

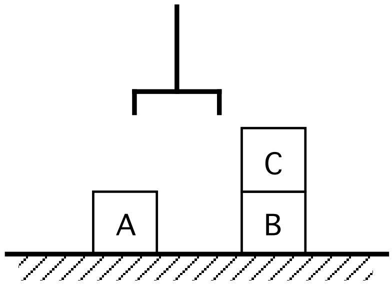
Classic Definitions:
1) A universe composed of a set of cubic blocks and a table
2) Blocks are mobile, the table is immobile
3) The agent is a mobile hand,
4) A block can sit on a table, on another block, or in the hand.
5) There cannot be more than one block on another block
6) The table is large enough for all blocks to be on the table.
7) The hand can move only one block at a time.

The state of the universe is formalized using first order predicates.
Blocks are represented by Capital Letters {A, B, C, …}
Variables (lower case letters) can represent sets of blocks
This are specified by Quantizers: for-all x ($\forall$x:), There-exists x: ($\exists$y:)

Predicates:

O(x,y)	On(x, y)	Block x is on Block y.
OT(x)	OnTable(x)	Block x is on the Table.
H(x)	Held(x)	Block x is in the hand.
F(x)	Free(x)	No block is On x : $\neg\exists$y: (On(y, x)) or $\forall$y: ($\neg$On(y, x))
HF()	HandFree()	The hand is empty, or $\neg\exists$x (H(x))

For example: HF $\wedge$ OT(A) $\wedge$ OT(B) $\wedge$ O(C, B) $\wedge$ F(C) $\wedge$ F(A)



Actions:

Actions are state change operators. Actions are atomic.

Principle: explicitly list all state changes.

Action: Name(Variables)

Precondition: predicates required for the action

Postcondition: predicates rendered true by the action.

In BlocksWorld all precondition predicates are rendered false by execution of an action

Nilsson defined 4 actions for BlocksWorld

Grasp(x):

Precondition: $HF() \wedge F(x) \wedge OT(x)$

Postcondition: $H(x)$

Pose(x):

Precondition: $H(x)$

Postcondition: $HF() \wedge F(x) \wedge OT(x)$

Stack(x, y)

Precondition: $H(x) \wedge F(y)$

Postcondition: $F(x) \wedge O(x, y) \wedge HF()$

Unstack (x, y)

Precondition: $F(x) \wedge O(x, y) \wedge HF()$

Postcondition: $H(x) \wedge F(y)$

A Blocks world is an example of a Context for situation modeling.
Each state in Blocks world is a situation (a conjunction of predicates).
The possible actions are examples of behaviors that can be associated with each situation.

To solve a problem we need to reason among situations to find a path from an initial state to a target state.

Structured Knowledge Representation

Structured knowledge representations, or Schema, are used to represent problem spaces for reasoning, perception and problem solving. In the following we will examine 3 classic structured knowledge representations: Frames, Scripts, and Situation Models. All of these are examples of Schema systems.

Frames

Frame: A Structured Representation to provide context for focusing visual interpretation of scenes.

A frame is composed of:

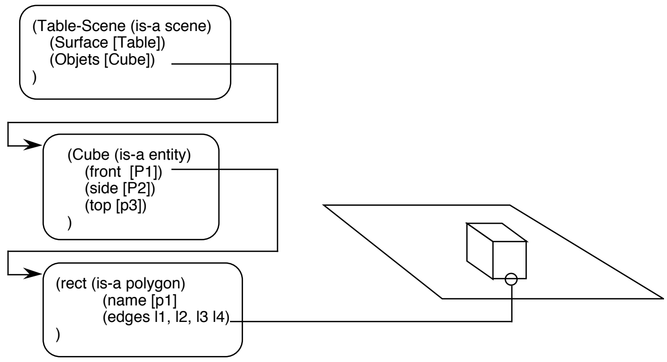
1) A list of Facts or Data (Slots)
    Values (called facets)
2) Procedures (also called procedural attachments)
    IF-NEEDED : deferred evaluation
    IF-ADDED : updates linked information
3) Default Values
    For Data
    For Procedures
4) Other Frames or Subframes (is-a and A-ko relations).

In computer vision, the problem can be stated as transforming a table of numbers into a symbolic description of the scene. This problem can be made easy when you know exactly what to look for.

M. Minsky proposed Frames as a structure to guide scene interpretation in computer vision. A frame guided interpretation in a top down manner, telling the system where to look and what to look for. Minsky's insight was that it is much easier to see if you know what to look for.

Frames provide visual context to guide scene interpretation.

A Frame tells the program what to look for and where to look for it.



A Frame is composed of a set of "slots" and "procedures".

A slot is a named place-holder for a pointer. The slots point to other frames that represent entities that are described (or interpreted) by the frame. Ultimately, some slots point to raw perceptions. When a slot points to an entity it is said to play a "role" in the frame. Frames typically come with methods (procedures) for searching for the entities that can plays roles in the frame.

Frames are formalized as a set of relations between observable entities.

Frames provide a structured representation for reasoning about the world.

Scripts

A script is a data structure used to represent a sequence of events. Scripts are used for interpreting stories. Popular examples have been script driven systems that can interpret and extract facts from Newspaper Stories.

Scripts can be used to represent procedural knowledge.

Scripts have been used to
1) Interpret, understand and reason about stories,
2) Understand and reason about observed events
3) Reason about observed actions
4) Plan actions to accomplish tasks.

A script is composed of
1) A scene: A sequence of Events linked by experience (Condition-Action)
2) Props (objects manipulated in the script)
3) The actors (agents that can change the state of the world).
4) A set of actions by the actors.

The script can be represented as a tree or network of states, driven by events.

As with Frames, scripts drive interpretation by telling the system what to look for and where to look next. The script can predict events.

Example of a script
The classic example is the restaurant script:

Scene: A restaurant with an entrance and tables.
Actors: The diners, servers, chef and Maitre d'Hotel.
Props: The table setting, menu, table, chair.
Acts: Entry, Seating, Ordering a meal, Serving a meal, Eating the meal, requesting the check, paying, leaving.

As with Frames, Scripts are formalized as sequence of states, where each state is a conjunction of relations between entities.
The states can be structured as a network joined by actions.

Scripts can be used to represent plans for solving problems.

Situation Models

P. Johnson-Laird 1983 - Mental Models.

Situations models are used in cognitive psychology to express the mental models that people use to understand and reason.

Properties: Sensed values
Entities: Anything that can be named or designated; People, things, etc.
(entites can be defined as a correlation of properties)
Relations: An N-ary predicate (N=1,2,3 …) over entities.
(relations are defined as tests on the properties of entities).
Situation: A set of relations between entities

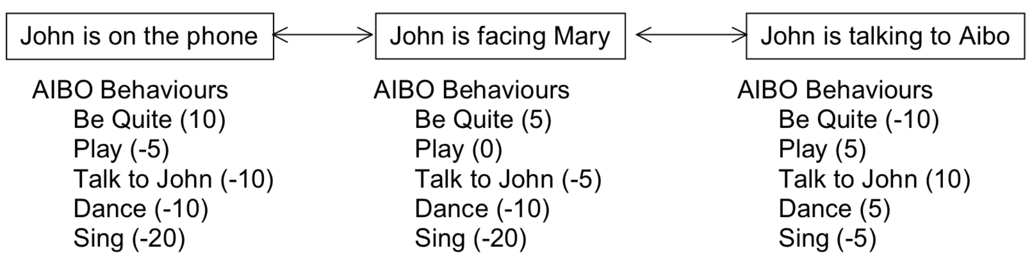
Example: John is facing Mary. John is talking to Mary. Mary is not listening.

facingP(John, Mary) ^ talkingP(John,Mary)^ not listeningP(Mar, John))

Situations can be organized into a state space referred to as a situation network. Each situation (or state) corresponds to a specific configuration of relations between entities. A change in relation results in a change in situation (or state).

The situation graph, along with the set of entities and relations is called a Context.

Each situation can prescribe and proscribe behaviours.
1) Behaviors: List of actions and reactions that are allowed or forbidden for each situation. Behaviors are commonly encoded as Condition-Action rules.
2) Attention: entities and relations for the system to observe, with methods to observe the entities
3) Default values: Expectations for entities, relations, and properties
4) Possible situations: Adjacent neighbors in the situation graph.



Each situation indicates:

Transition probabilities for next situations

The appropriateness or inappropriateness of behaviors

Behaviors include

1) methods for sensing and perception, and
2) appropriateness of actions
3) changes in state in reaction to events.

The sets of entities, relations, behaviors, and situations define a "Context".

Situation models are used to construct context aware systems.

A "Context" is defined as

1) A set of entities, with their properties.
2) A set of relations between entities
3) A network of situations, such that each situation specifies
   - A list of adjacent situations, possibly with transition probabilities.
   -A list of system behaviors that are allowed or forbidden,

possibly with preferences (appropriateness) for the situation.

Fundamental problems with Schema Systems.

Frames, Scripts and Situation Models are examples of Schema systems.

Some Fundamental Problem with all schema systems.

1) Knowledge Acquisition: Learning a schema system is long, tedious, and ad hoc process. There is a temptation to overload with useless information.

Research Challenge: Learning, Development and Adaptation.

2) Context Recognition (The Frame problem): Many problems are easily solved when the context and situation are known. Recognizing the correct context can be very difficult.

Research Challenge: Context recognition.

3) Ontology: Two schema systems describing the same context, may not use the same symbols for entities and relations. Communication requires a shared ontology.

Research Challenge: Semantic Alignment.

Alignment is the problem of determining equivalences in symbol expressions.

Solution these can be obtained from probabilistic techniques for learning and recognition.