Towards overcoming the knowledge acquisition bottleneck in declarative logic programming applications: embracing natural language inputs

Chitta Baral
Department of Computer Sc. & Eng.
Arizona State University
Tempe, AZ 85287

Knowledge Representation is important in building intelligent systems.

Knowledge is important in intelligent systems

- Meaning of the word “intelligent”
  - 1 (a) The capacity to acquire and apply knowledge. (b) The faculty of thought and reason. (c) Superior powers of mind.
  - The capacity to acquire and apply knowledge, especially toward a purposeful goal.
  - 1 (a) the ability to learn or understand or to deal with new or trying situations (b) the ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria (as tests)
  - the ability to comprehend; to understand and profit from experience [ant: stupidity]
- In summary: the key features of an intelligent entity
  - <u>it can acquire knowledge</u> through various means such as learning from experience, observations, reading and processing natural language text, etc.,
  - and <u>it can reason with this knowledge</u> to make plans, explain observations, achieve goals, etc.

To learn knowledge and to reason with it

- We need to know how to represent knowledge in a computer comprehensible format.
  - Develop a formal knowledge representation language which a computer can understand easily

Importance of inventing suitable knowledge representation languages

- Michael Gelfond once said:

Development of a suitable knowledge representation language and methodology is as important to AI systems

as

Calculus is to Physics and Engineering.

Research agenda hinted by the analogy

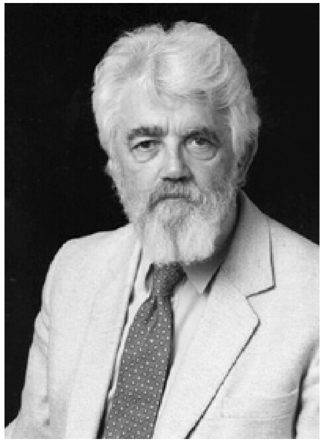
- What is it that makes Calculus widely used?
- It is the building block results that make it useful.
- Similarly, not enough to come up with a knowledge representation language and an interpreter for it.
- Need to develop the support structure, the building block results that will facilitate the use of the language.

Historical perspective

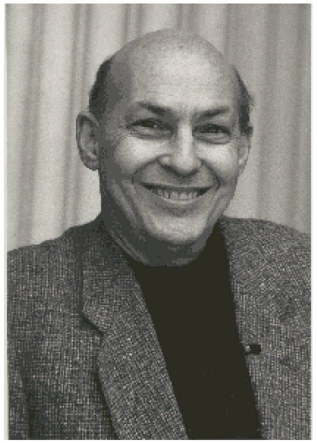
- AI pioneers (especially McCarthy and Minsky) realized the importance of KR to AI.
- McCarthy 1959: *Programs with commonsense*

  (perhaps the first paper on logical AI).

- Minsky 1974: *A framework for representing knowledge*.



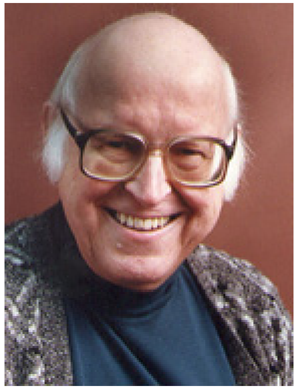
John McCarthy



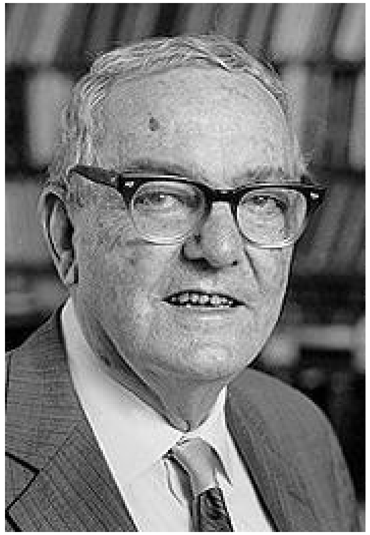
Marvin Minksy

Historical perspective – cont.

- Newell and Simon: Their early focus was more on the architecture of exhibiting intelligence, than on KR per se. Nevertheless, their systems did indeed manipulate various kinds of knowledge.



Allen Newell



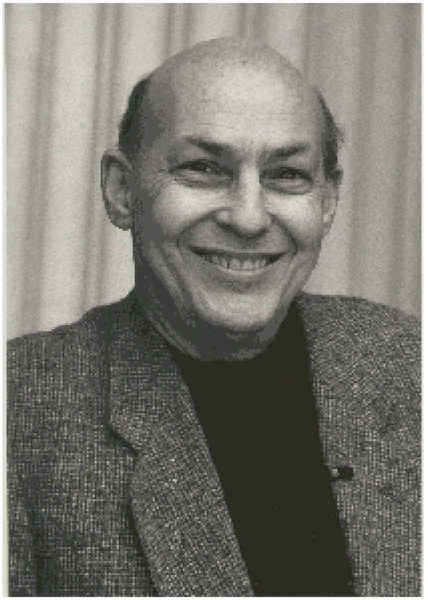
Herb Simon

What are the properties of a good KR language.

- **To start with: should be non-monotonic**
  - i.e., allow revision of conclusion in presence of new knowledge.
  - Hayes 1973 *(Computation and Deduction)* mentions monotonicity (calls it “extension property”) and notes that rules of default do not satisfy it.
  - Minsky 1974 *(A framework for representing knowledge)*

    criticizes monotonicity of logistic systems.

Pat Hayes



Marvin Minsky

Have we developed a "calculus" for KR?

Pre-1980 history of non-monotonic logics –from Minker's 93 survey

- THNOT in PLANNER [Hewitt in 1969]
- Prolog [Colmerauer et al. 1973]
- Circumscription [McCarthy 1977]
- Default Reasoning [Reiter 1978]
- Closed World Assumption (CWA) [Reiter 1978]
- Negation as failure [Clark 1978]
- Truth maintenance systems [Doyle 1979]
- **1st NCAI (AAAI 1980) 1st session**
  - **Nonmonotonic logic panel**
- AIJ Volume 13, 1980, a special issue

Last twenty five years …

- Many extensions, variations, and new logics, including
  - Non-monotonic modal logics
  - Auto-epistemic logic
  - Conditional logics
  - Description logics
  - Probabilistic semantics for default reasoning
  - Probability networks (Bayes nets, structural causal models)
  - **Answer Set Prolog** (programming in logic with answer sets) – in short AnsProlog

Have we invented "calculus" of KR yet?

- What basic properties should it have?
  - have a simple and intuitive syntax and semantics;
  - be non-monotonic;
  - have the ability to represent and reason with defaults and their exceptions;
  - allow us to represent and reason with incomplete information; and
  - allow us to express and answer problem solving queries such as planning queries, counterfactual queries, explanation queries and diagnostic queries.

Have we invented “calculus” of KR yet? - continued.

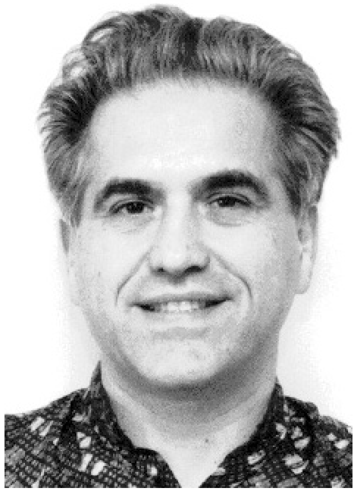
- What properties will make it useful?
  - should have building block results;
  - should have interpreters for reasoning with the language;
  - should have existing applications; and
  - should have systems that can learn knowledge in this language.

Is AnsProlog a good candidate?

- An AnsProlog^{or} program (late 1980s) is a collection of rules of the form:

A_0 or … or $A_l \leftarrow B_1, \ldots, B_m$, not $C_1, \ldots,$ not C_n.

where A_is, B_js and C_ks are literals.



Vladimir Lifschitz

Michael Gelfond

Is AnsProlog a good candidate?

- Its syntax uses the intuitive If-then form.
- It is non-monotonic.
- Can express defaults and their exceptions.
- Can represent and reason with incomplete information.
- Can express and answer problem solving queries.
- Large body of building block results.
- Various implementations: Smodels, DLV, Prolog.
- Many applications built using it.
- Learning systems: Progol.
- Its initial paper among the top 5 AI source documents in terms of citeseer citation.

Applications using Answer Set Prolog.

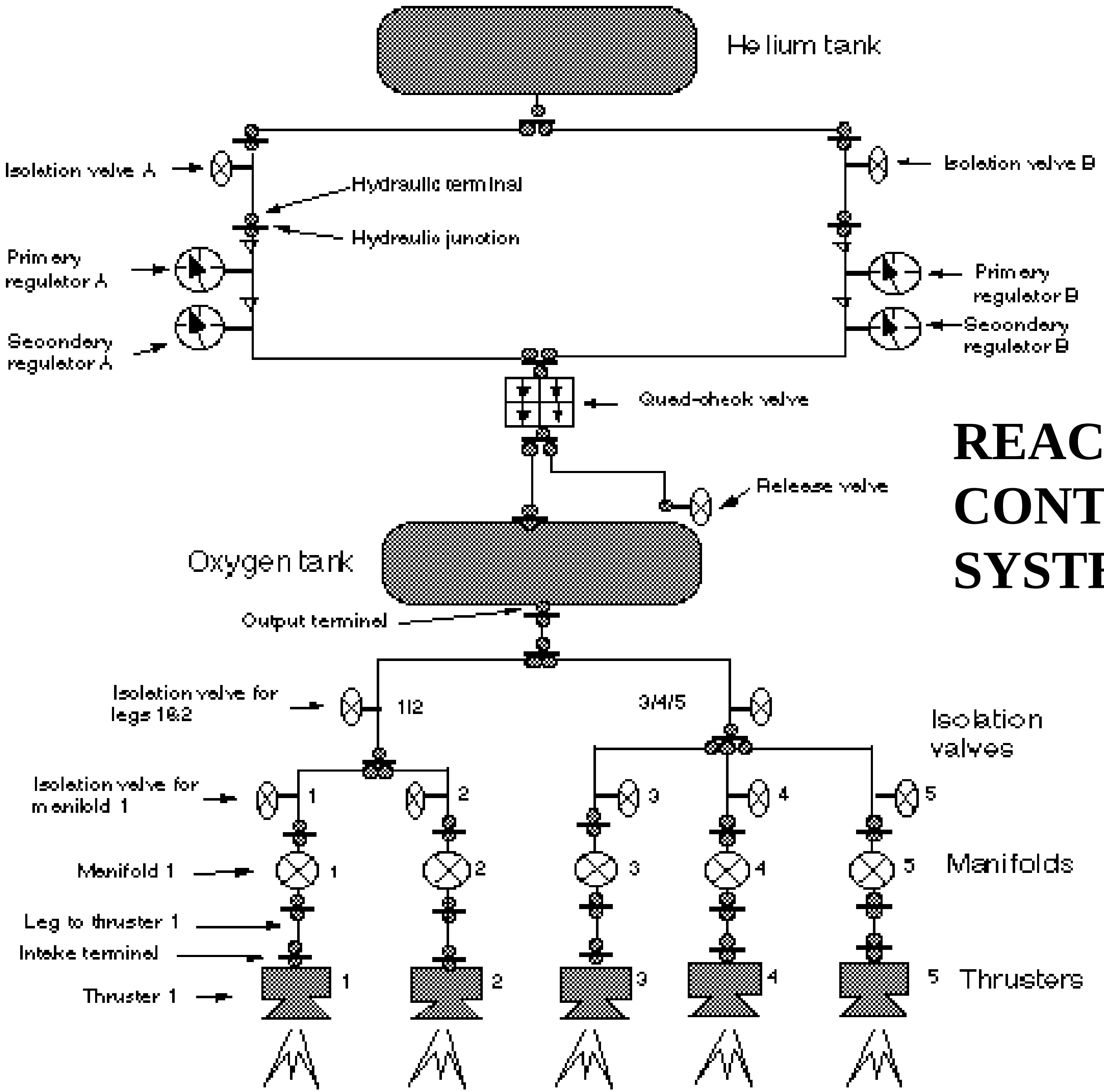
General Applications

- Reasoning
  - Reasoning with incomplete information, default reasoning.
  - Reasoning with preferences and priorities, inheritance hierarchies.
- Declarative problem solving (Answer set programming)
  - Planning, job-shop scheduling, tournament scheduling.
  - Abductive reasoning, explanation generations, diagnosis.
  - Combinatorial graph problems.
  - Combinatorial optimizations, combinatorial auctions.
  - Product configuration.
- Involving both
  - Data integration
  - Decision support systems
  - Question answering

Some specific applications

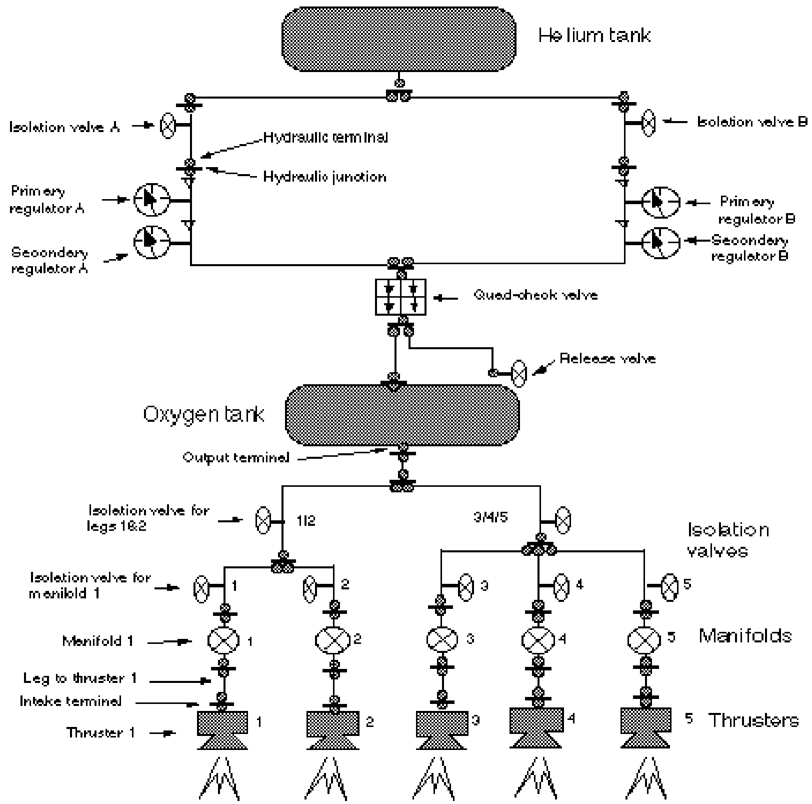
- Phylogeny construction
- Abduction and preferences in linguistics
- Inference of gene relation in micro-array data
- Reaction control system
- Question answering
- Reasoning about cell behavior

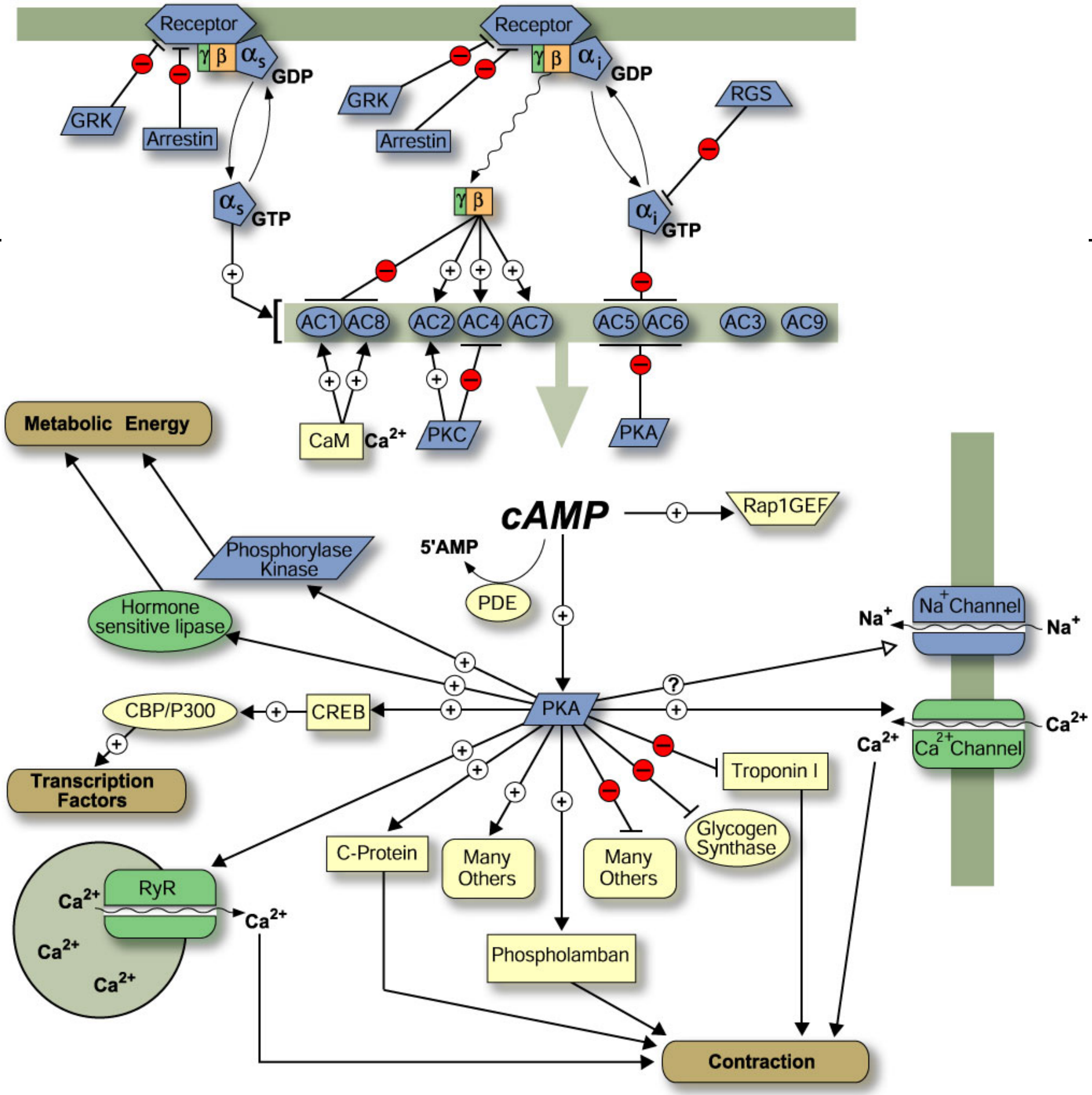
REACTION CONTROL SYSTEM (RCS)



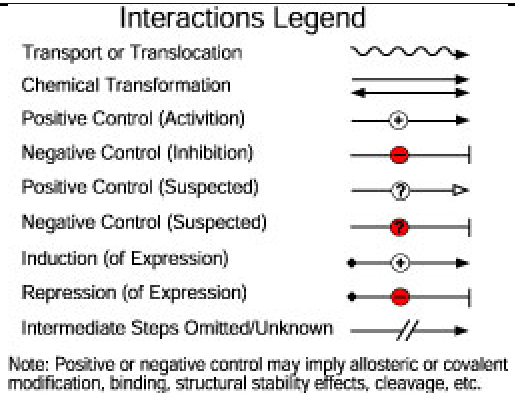
RCS/USA-Advisor (Texas Tech Univ.)

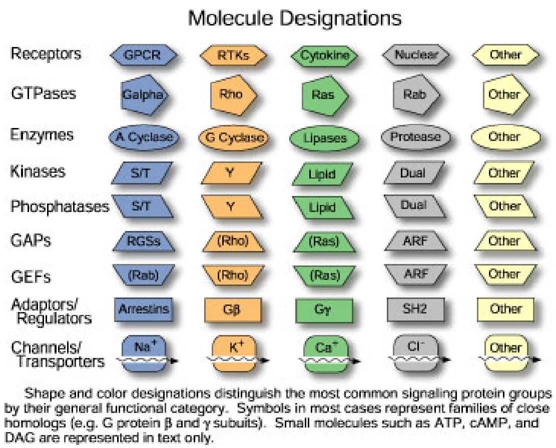
- A decision support system for shuttle controllers.
- Action: Switchon
- Direct effects, plus …
- Effect propagates and affects several objects.
- AnsProlog based planner is well-suited for this.

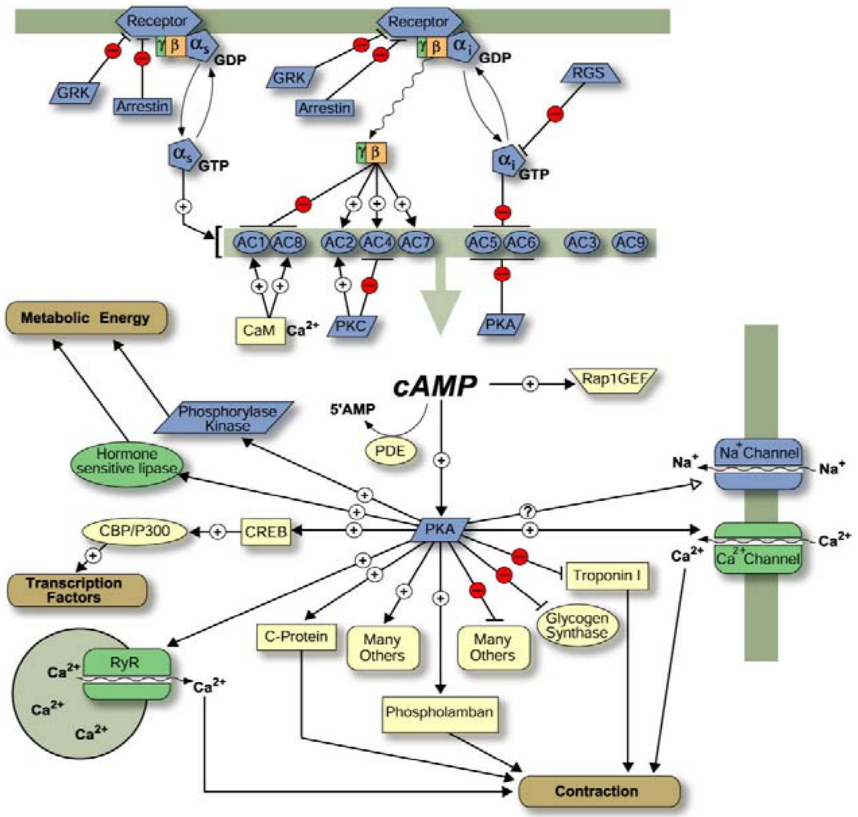



Receptor
GDP
GRK
Arrestin
GTP
Receptor
GDP
RGS
GRK
Arrestin
GTP
AC1
AC8
AC2
AC4
AC7
AC5
AC6
AC3
AC9
CaM
PKC
PKA
Metabolic Energy
cAMP
Rap1GEF
5'AMP
Phosphorylase Kinase
PDE
Hormone sensitive lipase
Na Channel
CBP/P300
CREB
PKA
Troponin I
Ca Channel
Transcription Factors
Glycogen Synthase
C-Protein
Many Others
Many Others
RyR
Phospholamban
Contraction

Signal Pathways
(from http://www.afcs.org/cm2/)







Reasoning about cell behavior: ASU

- Biosignet-RR
  - Hypothetical Reasoning : side effect of drugs
  - Planning: therapy design
  - Explanation of observations: figuring out what is wrong
- Biosignet-RRH
  - Hypothesis generation

Applications that involve NLP.

Question Answering

- Quite old as a problem. But consider the following!
- Text: John took a plane from Phoenix to Pittsburgh.
  - Question: Where is John after that? Where is his laptop which he always carries with him?
  - Answer: Pittsburgh. Pittsburgh.
    - Uses common-sense knowledge.
- Text/Data: On Dec 10th John is at home in Boston and does not have a ticket to Paris yet. On Dec 11th he is in Paris.
  - Query: Explain what might have happened in between.
  - Bought a ticket; gone to the Boston airport; taken a flight to Paris.
- Text/Data: On Dec 10th John is at home in Boston and does not have a ticket to Paris yet.
  - Query: What does John need to do to be in Paris on Dec 11th.
- Puzzles

How we did question answering?

- Extracted facts from natural language input.
- Created a AnsProlog question using the question asked in natural language.
- Wrote domain knowledge, common-sense knowledge.

An illustration

- **Text**
  - John spent Dec 10 in Paris and took a plane to Baghdad the next morning. He was planning to meet Bob who was waiting for him there.
- **Queries**
  - Q_1: Was John in the Middle East in mid-December?
  - Q_2: If so, did he meet Bob in the Middle East in mid-December?

Required background and common-sense knowledge

- Knowledge about geographical objects and their hierarchy. (M1)
  - Baghdad is a city in Iraq. Iraq is a country in the middle east region. …
  - A city in a country in a region is a city in that region.
- Knowledge about travel events. (M2)
  - If someone is in a city then she is in the country where the city is in and so on.
  - Executability conditions and effect of travel events
  - Inertia
  - Duration of flying
- Knowledge about time units. (M3)
  - Relation between various time granularities
- Knowledge about planned events, meeting events. (M4)
  - People normally follow through their plans
  - Executability condition of meeting events

M_1: The geography Module

- List of places
  - is(baghdad,city).
  - is(iraq,country).
  - ...
- Relation between places
  - in(baghdad, iraq).
  - in(iraq,middle_east).
  - in(paris,france).
  - in(france,western_europe).
  - in(western_europe,europe).
  - ...
- Transitive closure

  in(P1,P3) ← in(P1,P2), in(P2,P3).
- Completeness assumption about `in’

  -in(P1,P2) ← not in(P1,P2)

M_2: The traveling module

- Based on theory of dynamic systems
  - Views world as a transition diagram
    - States are labeled by fluents
    - Arcs labeled by actions
- Various types of traveling events
  - instance_of(fly,travel).
  - instance_of(drive,travel). …
- Generic description of John flying to Baghdad
  - event(a1).
  - type(a1,fly).
  - actor(a1,john).
  - destination(a1,baghdad).
- Actual event is recorded as
  - occurs(a1,i)

M_2: The traveling module (cont.)

- Representation of transition Diagram
  - State Constraints

    loc(P2,X,T) ← loc(P1,X,T), in(P2,P1).

    disjoint(P1,P2) ← -in(P1,P2), -in(P2,P1), neq(P1,P2).

    -loc(P2,X,T) ← loc(P1,X,T),disjoint(P1,P2).
  - Causal Laws

    loc(P,X,T+1) ← occurs(E,T), type(E,travel), actor(E,X), destination(E,P), -interference(E,T).

    -interference(E,T) ← not interference(E,T).
  - Executability Conditions

    -occurs(E,T) ← cond(T).
  - Inertia Rules (frame axioms)

    loc(P,X,T+1) ← loc(P,X,T), not -loc(P,X,T+1).

    -loc(P,X,T+1) ← -loc(P,X,T), not loc(P,X,T+1).

Reasoning with M1 and M2

- Given
  - loc(paris,john,0).
  - loc(baghdad,bob,0).
  - occurs(a1,0).
- And with M_1 and M_2 AnsProlog can conclude
  - loc(baghdad,john,1), loc(baghdad,bob,1),
  - loc(middle_east,john,1), -loc(paris,john,1)

M_3: Time and durations

- Duration of actions (additional ones needed for month etc.)

  time(T+1,day,D) ← occurs(E,T), type(E,fly),
  time(T,day,D), not -time(T+1,day,D).

- Basic measuring units
  - day(1..31). month(1..12). part(start). part(end). part(middle).
- Rules translating between one granularity to another

  time(T,part,middle) ← time(T,d,D), 10 < D < 20.

  time(T,season,summer) ← time(T,month,M), 5 < M < 9.

- Missing elements from the module
  - next(date(10,12,03),date(11,12,03)).
  - next(date(31,12,03),date(1,1,04)).

Reasoning with M_1, M_2 and M_3

- Given information about John's flight
  - loc(paris,john,0).
  - loc(baghdad,bob,0).
  - occurs(a1,0).
  - time(0,day,11).
  - time(0,month,12).
- The query Q1
  - ? loc(middle_east,john,T), time(T,month,12),
    time(T,part,middle).
- AnsProlog gives the correct answer: yes with T = 1.

M_4: planning to meet and meeting

- Describing the event meet
  - event(a2). type(a2,meet).
  - actor(a2,john). actor(a2,bob).
  - place(a2,baghdad).
- Executability conditions of the meeting event
  -occurs(E,T) ← type(E,meet), actor(E,X), place(E,P), -loc(P,X,T).
- Planned meeting: planned(a2,1).
- Planned actions and their occurrence: ``People normally follow their plans''
  occurs(E,T) ← planned(E,T), not -occurs(E).
- People persist with their plans until it happens
  planned(E,T+1) ← planned(E,T), -occurs(E,T).
- Second query
  - ? occurs(E,T), type(E,meet), actor(E,john), actor(E,bob), loc(middle_east,john,T), time(T,month,12), time(T,part,middle).
- Answer: Yes.

Puzzle solving and its role in intelligent analysis

- The domain of the data gives many possibilities.
- Evidences rule out most of the possibilities.
- Some definite conclusions could be made with respect to the remaining conclusions.
- Puzzle Example: Who owns the zebra?
  - There are five houses.
  - Each house has its own unique color.
  - All house owners are of different nationalities.
  - They all have different pets.
  - They all drink different drinks.
  - They all smoke different cigarettes.
  - The English man lives in the red house.
  - The Swede has a dog.
  - …

RTE (Recognizing Textual Entailment) examples

Action and indirection reference to the action

- **Text**: The drug that slows down or halts Alzheimer's disease is expensive.
- **Hypothesis**: Alzheimer's disease is treated using drugs.
- **Answer**: Yes
- **Problem specific analysis**:
  - Connecting "drug that slows downs or halts *X*" with "drug treats *X*".
- **Generalization**:
  - Connecting an action and an indirect reference to that action by mentioning its effect.
  - "treat" is an action; its effects are "slows down or halt."

Connecting the multiple effects of an action

- **Text**: Yoko Ono unveiled a bronze statue of her late husband John Lennon.
- **Hypothesis**: Yoko Ono is John Lennon's widow.
- **Answer**: Yes
- **Problem specific analysis**:
  - connecting "late husband" with "widow".
  - The action dying, when married to Yoko Ono, makes John Lennon a late husband.
  - The same action makes Yoko Ono the widow of John Lennon.
- **Generalization**:
  - An action *a* may have effects *f* and *g*.
  - If *f* is observed and we can explain by saying that *a* happened, then we should be able to conclude that *g* is also true.

Reasoning about intentions

- **Text**: After graduating in 1977, Gallager chose to accept a full scholarship to play football for Temple University.
- **Hypothesis**: Gallager attended Temple University.
- **Answer**: Yes
- **Problem specific analysis**:
  - Connecting the actions "*P* accepting a full scholarship to play football" to "attend university"
  - "accept a scholarship to play football" shows intention of attending the university
  - Intentions are normally executed.
- **Generalization**:
  - Intentions are normally executed.

Major difficulty we faced in many of these applications

- Writing the background knowledge!

Two ways to address that

- Build knowledge bases collaboratively.
  - Need to build knowledge libraries.
- Learn knowledge from reading natural language text.

Project Halo!

- An exciting KR effort!
- Halo pilot: structured around a challenge involving 71 pages of an advanced placement (AP) inorganic chemistry syllabus. (2003-04)
- Three participants in the Pilot: SRI-UT Austin; CYCORP, Ontoprise
- Second phase focused on knowledge acquisition.

From natural language text to formal knowledge

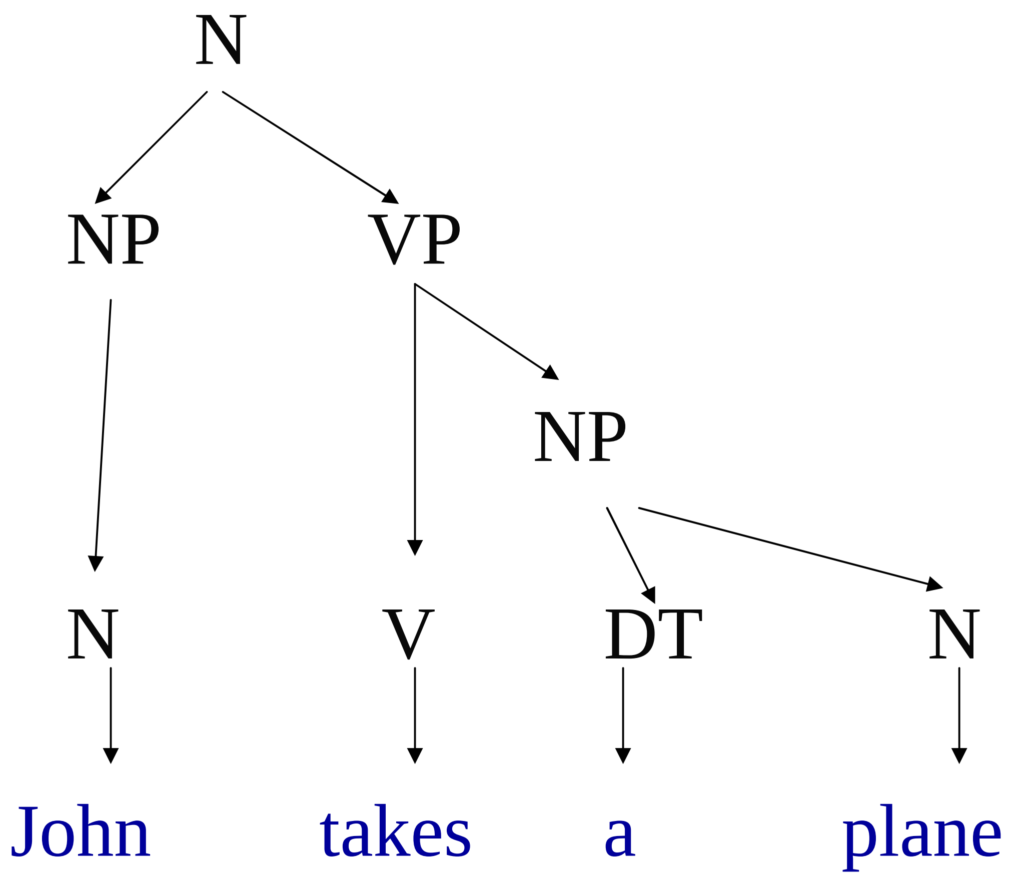
- Direct use of natural language will necessitate reasoning mechanism that can deal with natural language.
- Why not automatically translate natural language input to AnsProlog programs?
  - “programs” with rules, not just facts!

Crazy Idea!

- Not quite!
- Vast literature on Natural language semantics.
  - Translating specific subclasses of natural language to first order logic
  - Specific formalizations of difficult natural language constructs.
    - Often not implemented.

An illustration of translating NL to first order logic using λ-calculus

- John takes a plane



CCG parsing

<: backward functional application

$X \quad Y\backslash X \longrightarrow Y$

>: forward functional application

$X/Y \quad Y \longrightarrow X$

- John (Noun) takes (Trans. verb) a (Determiner) plane (Noun)

John	takes	a	plane
(Noun)	(Trans. verb)	(Determiner)	(Noun)
N	(S[dcl]\NP)/NP	NP[nb]/N	N

N ⟶ NP (lex)

NP[nb]/N N ⟶ NP[nb] (>)

(S[dcl]\NP)/NP NP[nb] ⟶ S[dcl]\NP (>)

NP S[dcl]\NP ⟶ S[dcl] (<)

S[dcl]

dcl – sentence is declarative

nb – non-bare noun, e.g. includes a determiner

The goal

- ‘John takes a plane.’

- *exists y.(plane*(y) Λ *take(John,y))*
  - There exists some y such that y is a plane and John takes it.

Exists – existential quantifier
Λ – logical and

Construction by composition

- *a* : $\lambda w.\lambda z.$ *exists y.(w @ y ∧ z @ y)*
- plane: *λx.plane(x)*
- a plane
  - *λw.λz. exists y.(w @ y ∧ z @ y)* @ *λx.plane(x)* =
  - *λz. exists y.(λx.plane(x) @ y ∧ z @ y)* =
  - *λz. exists y.(plane(y) ∧ z @ y*
- takes: *λw.λz. (w @ λx take(z,x))*
- takes a plane
  - *λw.λz. (w @ λx take(z,x))* @ *λw. exists y.(plane(y) ∧ w @ y)* =
  - *λz. (λw. exists y.(plane(y) ∧ w @ y) @ λx take(z,x))*
  - *λz. (exists y.(plane(y) ∧ λx take(z,x) @ y))*
  - *λz. (exists y.(plane(y) ∧ take(z,y)))*

Construction by composition (cont.)

- John: λu. (u @ John)
- John takes a plane.
  - λu. (u @ John) @ *λz. (exists y.(plane*(y) Λ *take(z,y)))*
  - *λz. (exists y.(plane*(y) Λ *take(z,y))) @ John =*
  - *exists y.(plane*(y) Λ *take(John,y))*

Bos’s RTE System

- Uses CCG parser and his own system to translate natural language to classical logic.
  - Available.
- Finds a subset of relevant Wordnet terms.
- Does model finding with respect to the above.

Glimpse of the future – Solving the knowledge acquisition bottleneck

- Current Status
  - Simple facts can be extracted
  - Certain first order logical formulas can be constructed.
- What is missing
  - Commonsense quantifiers (most; normally; usually)
  - Representations that can not be done in first order
    - Inductive definitions
  - Reasoning mechanisms that can not be done in first order
- Goal: Acquire the above mentioned kind of knowledge from text.

Reasoning with normative statements

- Normally Birds fly; Tweety is a bird.
  - Tweety flies
- Normally Birds fly; Tweety is a bird. Penguins are birds that do not fly. Tweety is a Penguin.
  - Tweety does not fly.

Inductive definition

- Parents are ancestors.
- Parents of ancestors are ancestors.
- Nothing else are anecstors.
- anc(X,Y) ← par(X,Y).
- anc(X,Y) ←par(X,Z), anc(Z,Y).
- ~anc(X,Y) ← not anc(X,Y).

Dynamic domain

- The property of an object in the world is referred to as a fluent.
- A state of a world tells us about the values of fluents in the world.
- The value of a fluent normally remains unchanged. Exceptions are fluents which are directly or indirectly affected by an action.

Deep reasoning terms

- A plan is a sequence of actions which when executed achieves a goal.
- An initial state explanation is a set of properties about the world which when assumed explains the observations.

Approaches we are taking

- Collecting a corpus of statements
- Developing Lambda calculus with AnsProlog functions.
- Developing translations using Discourse representation structures.

Normal birds fly: a simple glimpse

- normal:
  - λu. λv (v @ X ← u @ X, not ab(X).)
- birds: λy. bird(y).
- normal birds:
  - λv (v @ X ← bird(X), not ab(X).)
- fly: λz. fly(z).
- normal birds fly:
  - fly(X) ← bird(X), not ab(X).

Summing Up

- Knowledge representation is key to AI.
- Having suitable KR languages as well as building-block results around them is crucial for building AI systems.
- AnsProlog is a good candidate to be the “Calculus” of KR and Intelligent system building.
- However, writing knowledge is still a bottleneck.
- Devising ways to converting natural language to a formal language is a worthwhile long term goal.
  - The timing is right!
  - Good parsers and initial systems available.
  - Good progress in knowledge representation.
  - AnsProlog implementations are available.

Acknowledgements

- Michael Gelfond, Richard Scherl, Steve Maiorano, Vladimir Lifschitz, Richard Watson, Juraj Dzifcak, Luis Tari, Marcello Balduccini, Yulia Lierler and many others.
- Funding agencies: NSF, NASA, DTO, USA.

Acknowledgements for pictures obtained from the web.

- John McCarthy: www.kurzweilai.net/ bios/bio0008.html
- Marvin Minsky: www.almaden.ibm.com/ cs/NPUC95/panelists.html
- Pat Hayes: http://www.kuenstliche-intelligenz.de/Artikel/InterviewwithPatrickHayes.htm
- Allen Newell: diva.library.cmu.edu/ Newell/
- Herb Simon: http://istpub.berkeley.edu:4201/bcc/Spring2001/cio.simon.html
- Vladimir Lifschitz: http://www.cs.utexas.edu/users/vl/
- Michael Gelfond: http://www.cs.ttu.edu/~mgelfond/
- RCS: http://www.ksl.stanford.edu/htw/dme/rcs.html
- cAMP and pathway keys: http://www.signaling-gateway.org/

Thanks for the opportunity!

- Thanks to my students, teachers, colleagues and sponsors!
- Its been fun pursuing one (plus epsilon) of the 4 great questions (as mentioned in Simon’s memoir of Newell)
  - the nature of matter,
  - the origins of the universe,
  - the nature of life, and
  - the workings of mind (simulating “intelligence” artificially).
- and looking forward to the continuing journey.
- I wish you all the best of this journey too.

The End!

- THANKS