9 Reasoning in Uncertain Situations

Note: the material for Section 9.1 is enhanced

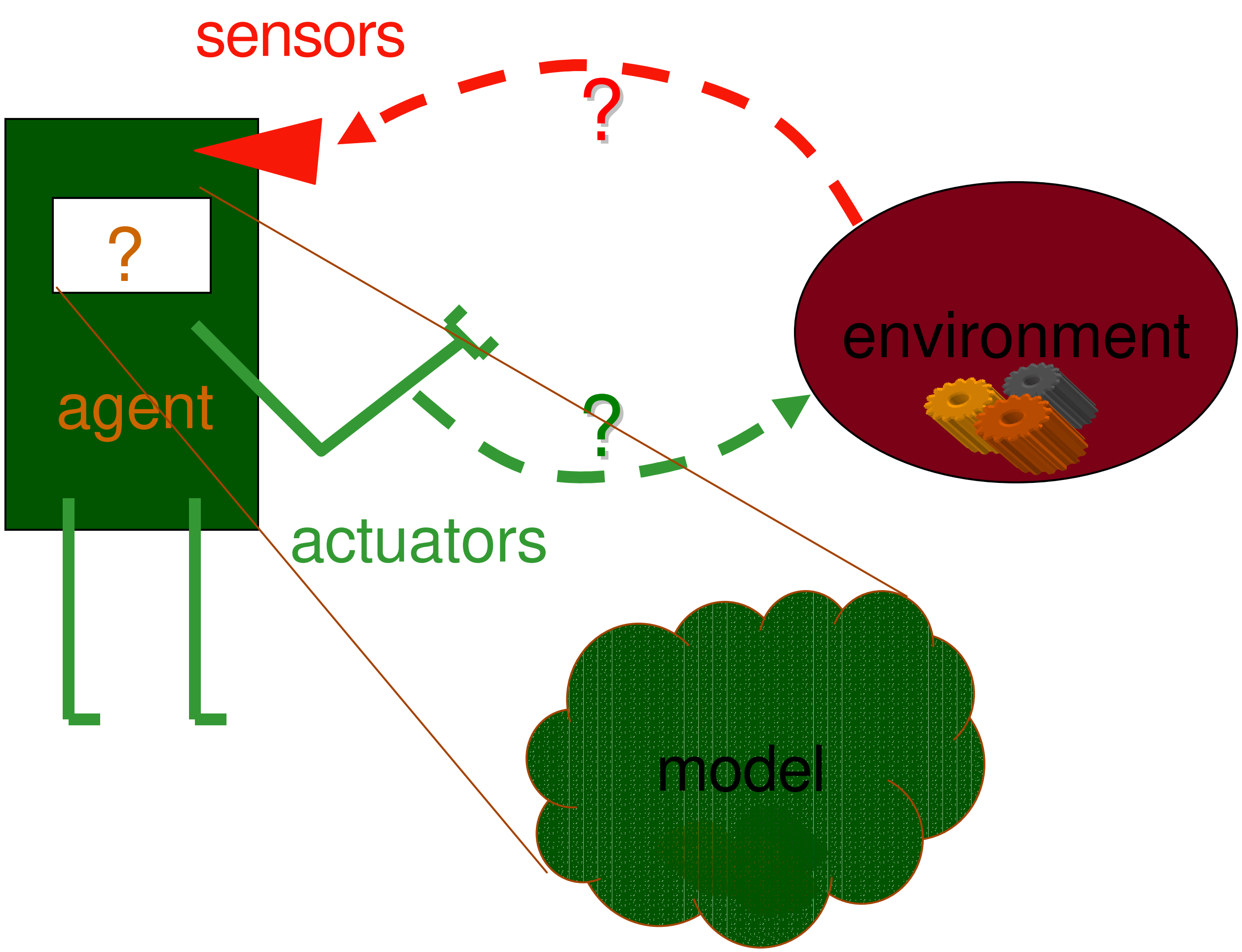
Note: we will only briefly cover fuzzy logic from Section 9.2

Additional references for the slides:
Jean-Claude Latombe's CS121 slides:
robotics.stanford.edu/~latombe/cs121

Chapter Objectives

- Learn about the issues in dynamic knowledge bases
- Learn about adapting logic inference to uncertain worlds
- Learn about probabilistic reasoning
- Learn about alternative theories for reasoning under uncertainty
- The agent model: Can solve problems under uncertainty

Uncertain agent



Types of Uncertainty

- **Uncertainty in prior knowledge**

  E.g., some causes of a disease are unknown and are not represented in the background knowledge of a medical-assistant agent

Types of Uncertainty

- Uncertainty in actions

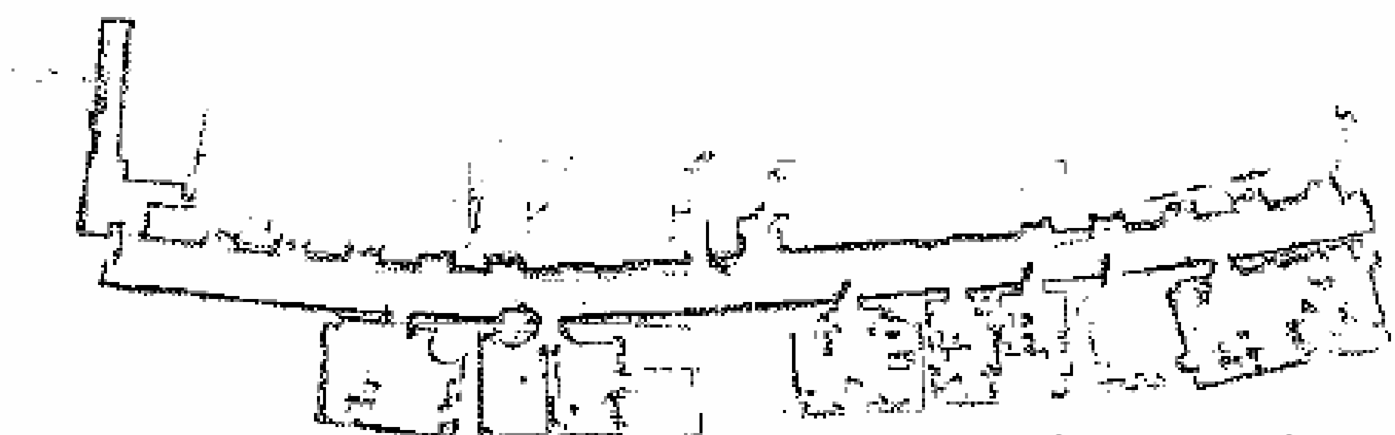
E.g., to deliver this lecture:
I must be able to come to school
the heating system must be working
my computer must be working
the LCD projector must be working
I must not have become paralytic or blind

As we discussed with planning, actions are represented with relatively short lists of preconditions, while these lists are in fact arbitrary long. It is not efficient (or even possible) to list all the possibilities.

Types of Uncertainty

• **Uncertainty in perception**

E.g., sensors do not return exact or complete information about the world; a robot never knows exactly its position.



Courtesy R. Chatila

Sources of uncertainty

- **Laziness (efficiency)**
- **Ignorance**

What we call *uncertainty* is a summary of all that is not explicitly taken into account in the agent's knowledge base (KB).

Assumptions of reasoning with predicate logic

(1) Predicate descriptions must be sufficient with respect to the application domain.

Each fact is known to be either true or false. But what does lack of information mean?

Closed world assumption, assumption based reasoning:
PROLOG: if a fact cannot be proven to be true, assume that it is false
HUMAN: if a fact cannot be proven to be false, assume it is true

Assumptions of reasoning with predicate logic (cont'd)

(2)The information base must be consistent.

Human reasoning: keep alternative (possibly conflicting) hypotheses. Eliminate as new evidence comes in.

Assumptions of reasoning with predicate logic (cont'd)

(3) Known information grows *monotonically* through the use of inference rules.

Need mechanisms to:

- **add information based on assumptions (nonmonotonic reasoning), and**
- **delete inferences based on these assumptions in case later evidence shows that the assumption was incorrect (truth maintenance).**

Questions

How to represent uncertainty in knowledge?

How to perform inferences with uncertain knowledge?

Which action to choose under uncertainty?

Approaches to handling uncertainty

Default reasoning [Optimistic]
  non-monotonic logic

Worst-case reasoning [Pessimistic]
  adversarial search

Probabilistic reasoning [Realist]
  probability theory

Default Reasoning

Rationale: The world is fairly normal. Abnormalities are rare.

So, an agent assumes normality, until there is evidence of the contrary.

E.g., if an agent sees a bird X, it assumes that X can fly, unless it has evidence that X is a penguin, an ostrich, a dead bird, a bird with broken wings, …

Modifying logic to support nonmonotonic inference

$$p(X) \wedge \text{unless } q(X) \rightarrow r(X)$$

If we

- believe p(X) is true, and
- do not believe q(X) is true (either unknown or believed to be false)

then we

- can infer r(X)
- later if we find out that q(X) is true, r(X) must be retracted

“unless” is a ***modal operator***: deals with belief rather than truth

Modifying logic to support nonmonotonic inference (cont'd)

$p(X) \wedge \text{unless } q(X) \rightarrow r(X)$	in KB
$p(Z)$	in KB
$r(W) \rightarrow s(W)$	in KB
- - - - - -	
$\neg q(X)$??	q(X) is not in KB
$r(X)$	inferred
$s(X)$	inferred

Example

If it is snowing and unless there is an exam tomorrow, I can go skiing.

It is snowing.

Whenever I go skiing, I stop by at the Chalet to drink hot chocolate.

- - - - - -

I did not check my calendar but I don't remember an exam scheduled for tomorrow, conclude: I'll go skiing. Then conclude: I'll drink hot chocolate.

“Abnormality”

$$p(X) \wedge \text{unless ab } p(X) \rightarrow q(X)$$

ab: abnormal

Examples: If X is a bird, it will fly unless it is abnormal.

(abnormal: broken wing, sick, trapped, ostrich, ...)

If X is a car, it will run unless it is abnormal.

(abnormal: flat tire, broken engine, no gas, …)

Another modal operator: M

$p(X) \wedge M\ q(X) \rightarrow r(X)$

If

- **we believe p(X) is true, and**
- **q(X) is *consistent with* everything else,**

then we

- **can infer r(X)**

"M" is a *modal operator* for "is consistent."

Example

$\forall X$ good_student(X) $\wedge$ M study_hard(X) $\rightarrow$ graduates (X)

How to make sure that study_hard(X) is consistent?

Negation as failure proof: Try to prove $\neg$study_hard(X), if not possible assume X does study.

Tried but failed proof: Try to prove study_hard(X), but use a heuristic or a time/memory limit. When the limit expires, if no evidence to the contrary is found, declare as proven.

Potentially conflicting results

$\forall$X good_student (X) $\wedge$ M study_hard (X) $\rightarrow$ graduates (X)

$\forall$X good_student (X) $\wedge$ M $\neg$ study_hard (X) $\rightarrow$ $\neg$ graduates (X)

good_student(peter)

If the KB does not contain information about study_hard(peter), both graduates(peter) and $\neg$graduates (peter) will be inferred!

Solutions: autoepistemic logic, default logic, inheritance search, more rules, ...

$\forall$Y party_person(Y) $\rightarrow$ $\neg$ study_hard (Y)
party_person (peter)

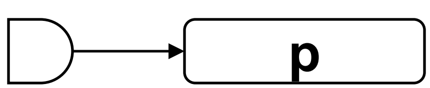
Truth Maintenance Systems

They are also known as ***reason maintenance systems***, or ***justification networks***.

In essence, they are dependency graphs where rounded rectangles denote predicates, and half circles represent facts or "and"s of facts.

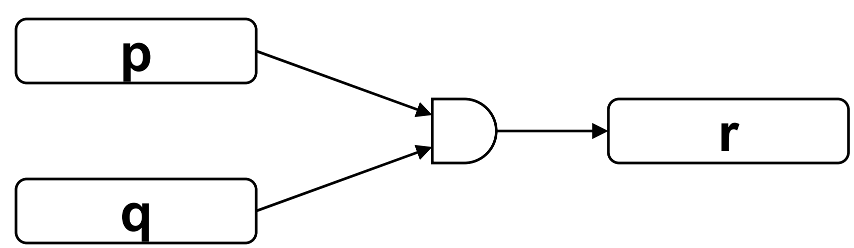
Base (given) facts:

p is in the KB



ANDed facts:

$p \wedge q \rightarrow r$



How to retract inferences

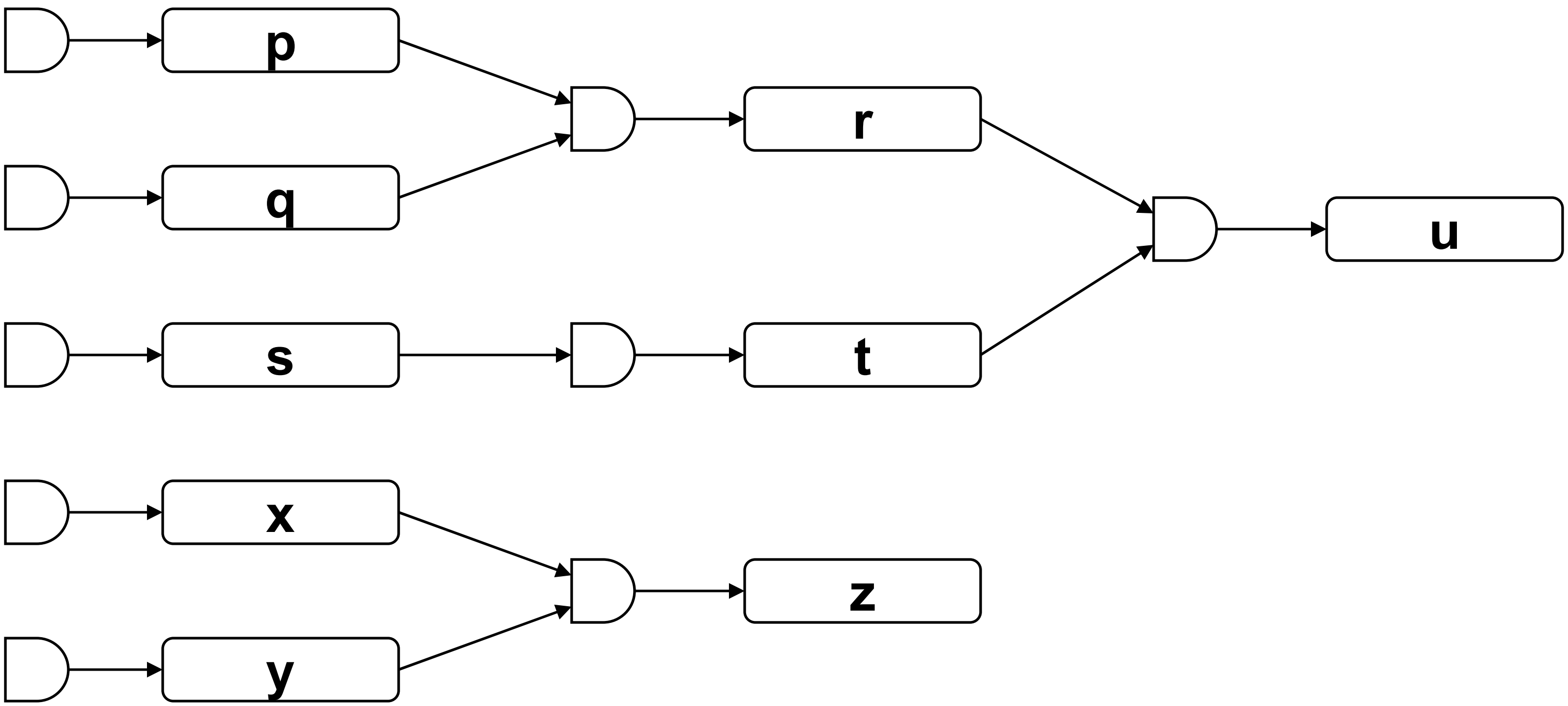
- In traditional logic knowledge bases inferences made by the system might have to be retracted as new (conflicting) information comes in

- In knowledge bases with uncertainty inferences might have to be retracted even with non-conflicting new information

- We need an efficient way to keep track of which inferences must be retracted

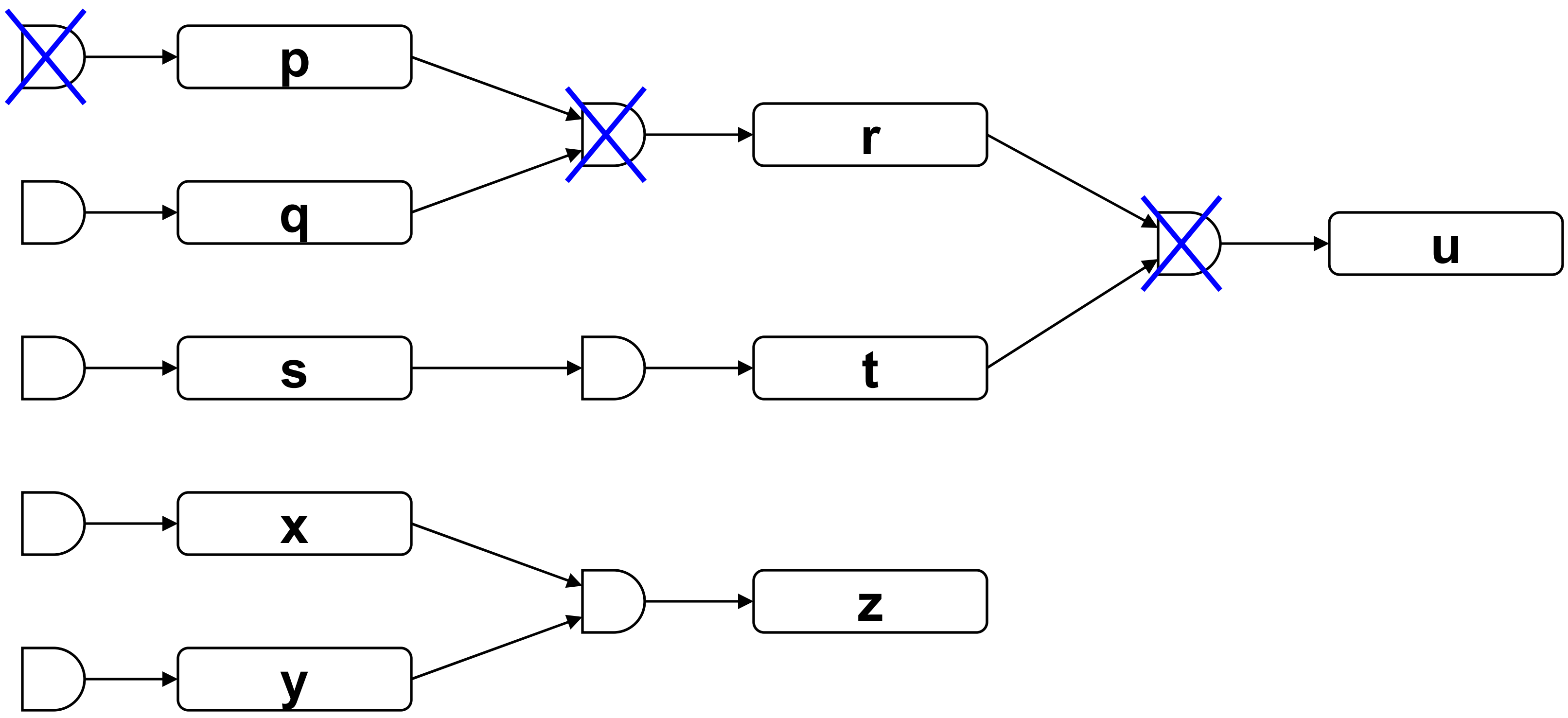
Example

When p, q, s, x, and y are given, all of r, t, z, and u can be inferred.



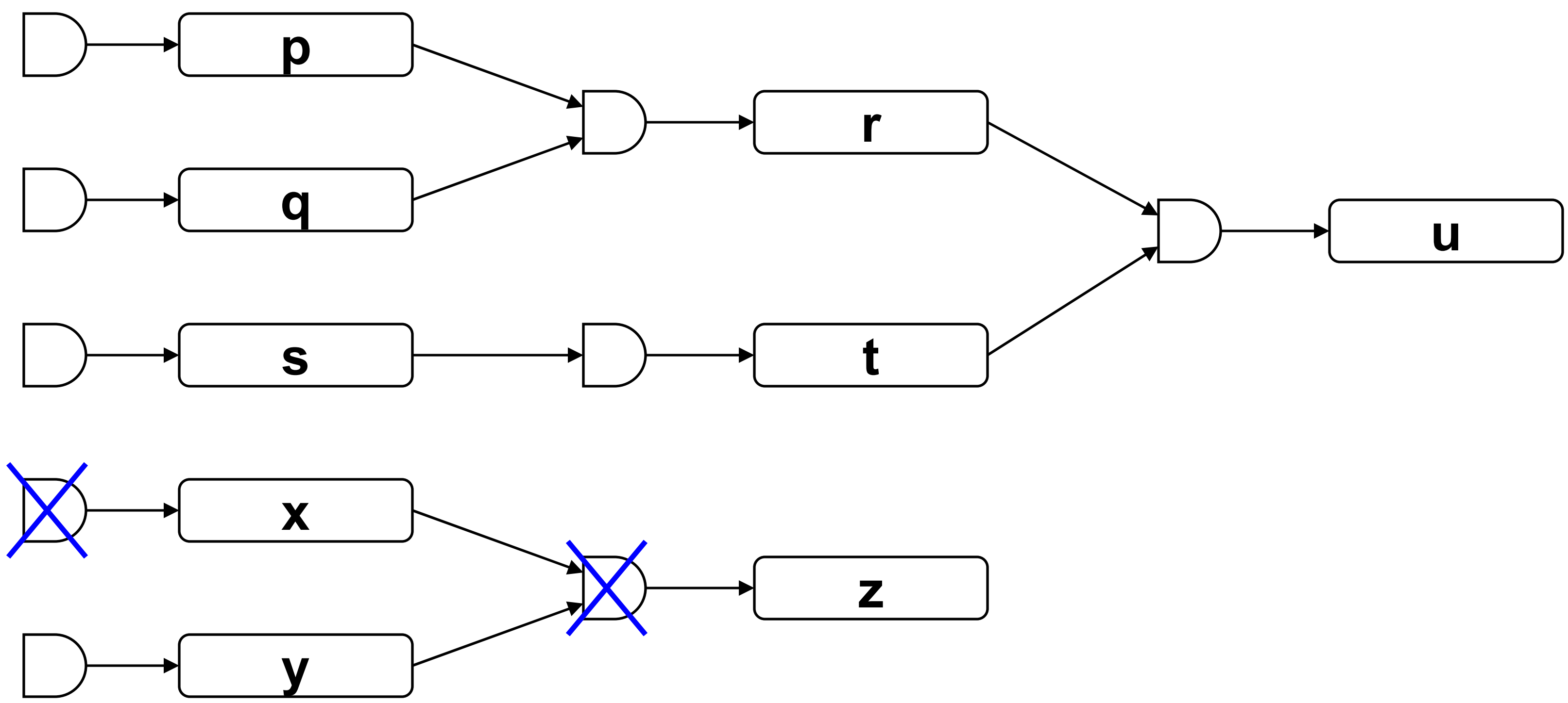
Example (cont'd)

If p is retracted, both r and u must be retracted (Compare this to *chronological backtracking*)



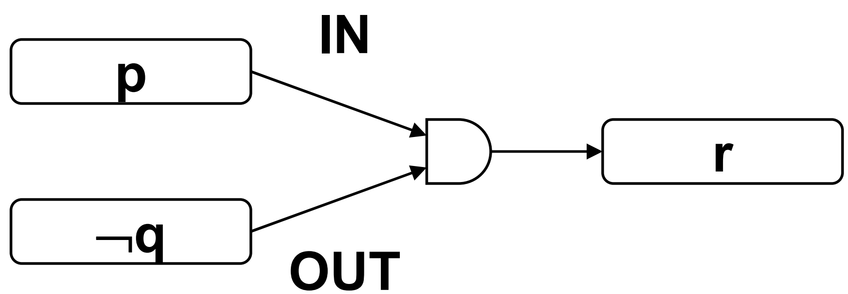
Example (cont'd)

If x is retracted (in the case before the previous slide), z must be retracted.



Nonmonotonic reasoning using TMSs

$p \wedge M\, q \rightarrow r$

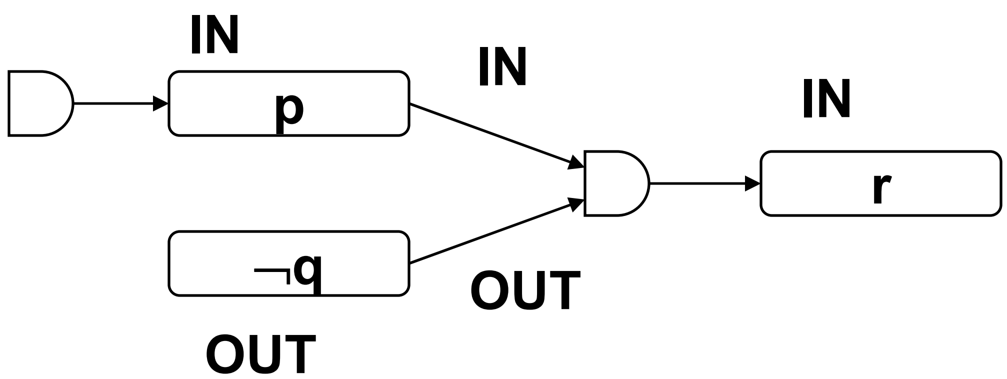


IN means "IN the knowledge base."
OUT means "OUT of the knowledge base."

The conditions that must be IN must be proven.
For the conditions that are in the OUT list,
non-existence in the KB is sufficient.

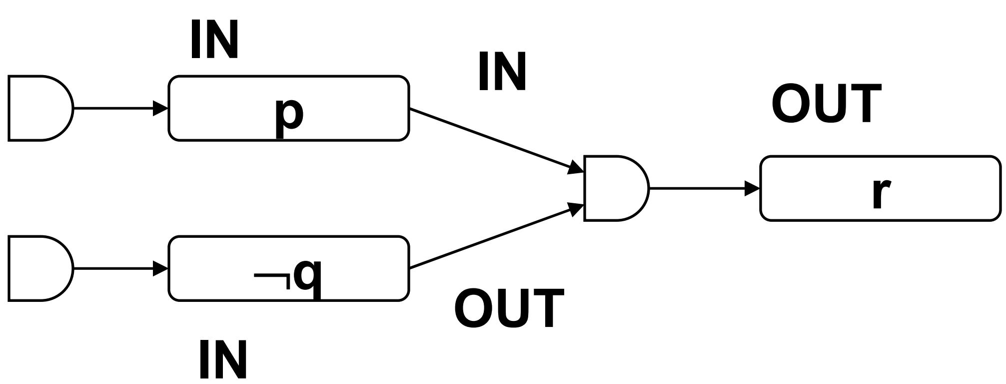
Nonmonotonic reasoning using TMSs

If p is given, i.e., it is IN, then r is also IN.



Nonmonotonic reasoning using TMSs

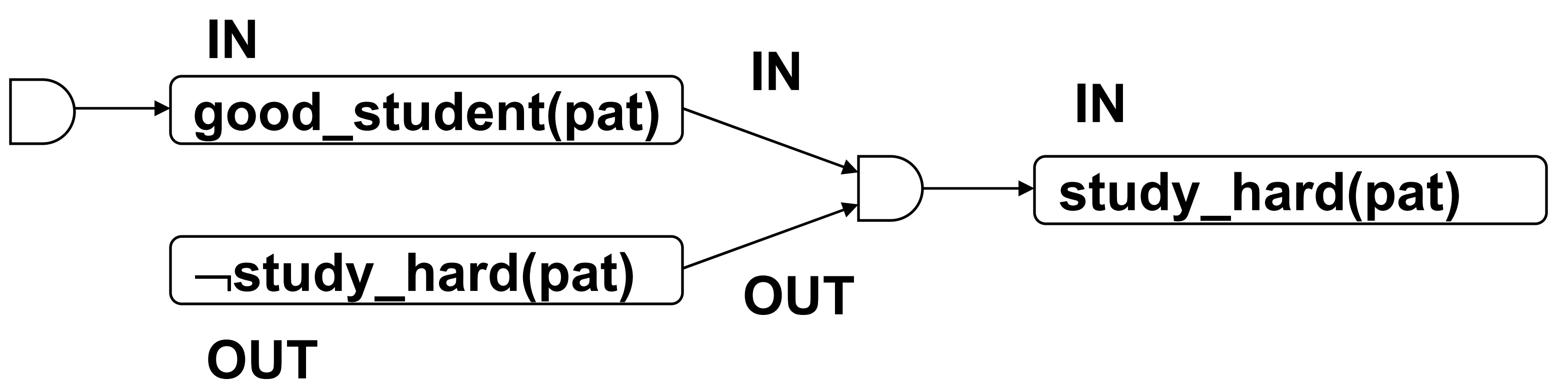
If ¬q is now given, r must be retracted, it becomes OUT. Note that when ¬q is given the knowledge base contains more facts, but the set of inferences shrinks (hence the name nonmonotonic reasoning.)



A justification network to believe that Pat studies hard

$\forall X$ **good_student(X) $\wedge$ M study_hard(X) $\rightarrow$ study_hard (X)**

good_student(pat)

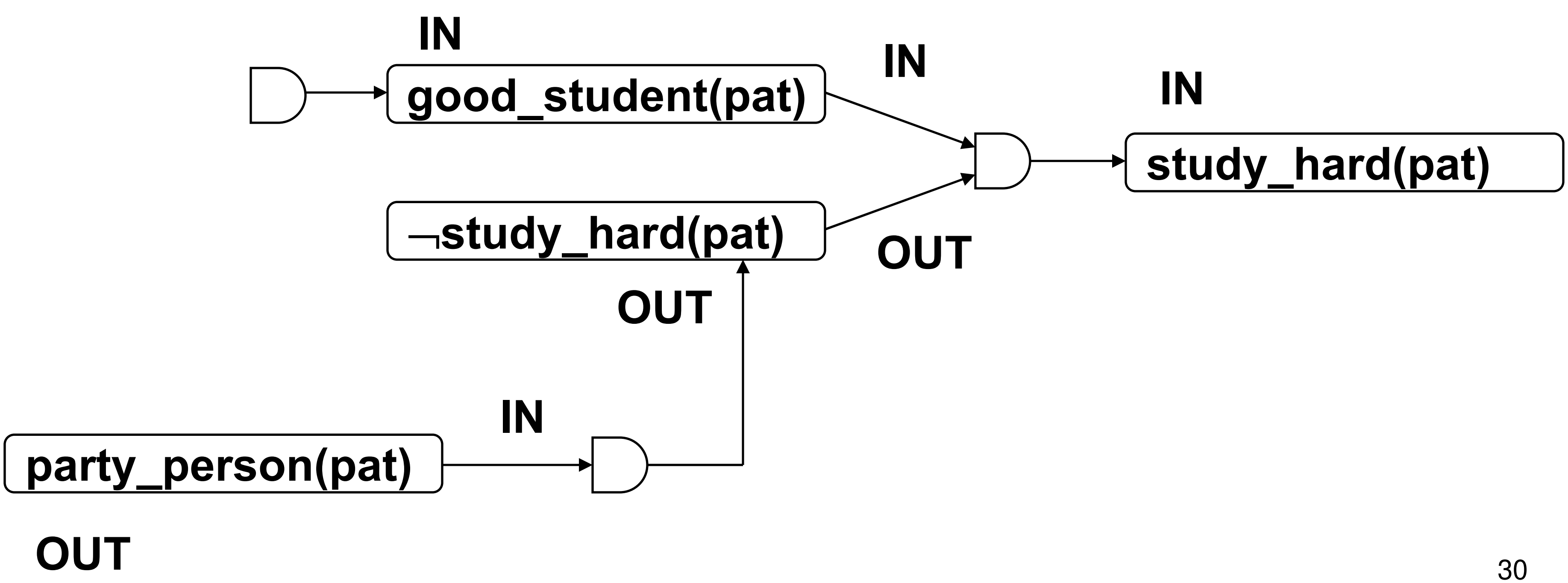


It is still justifiable that Pat studies hard

$\forall X$ good_student(X) $\wedge$ M study_hard(X) $\rightarrow$ study_hard (X)

$\forall Y$ party_person(Y) $\rightarrow \neg$ study_hard (Y)

good_student(pat)



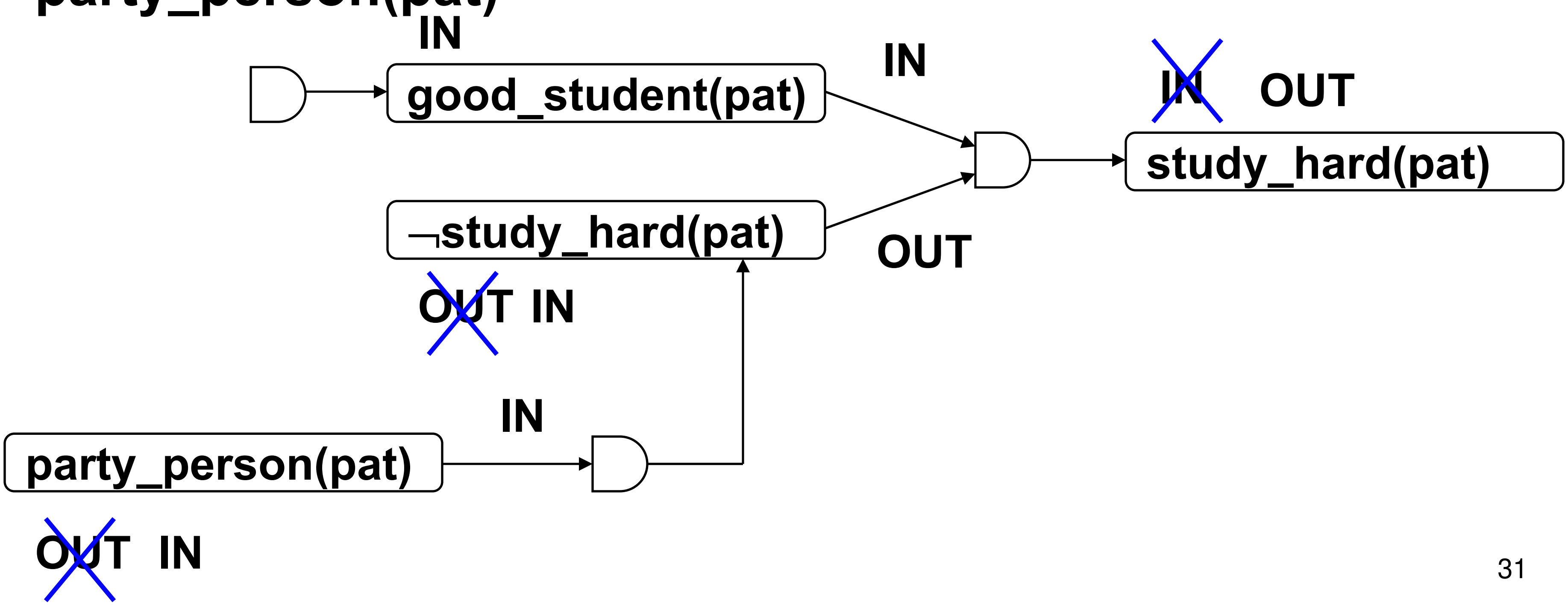
“Pat studies hard” is no more justifiable

$\forall X$ good_student(X) $\wedge$ M study_hard(X) $\rightarrow$ study_hard (X)

$\forall Y$ party_person(Y) $\rightarrow \neg$ study_hard (Y)

good_student(pat)

party_person(pat)



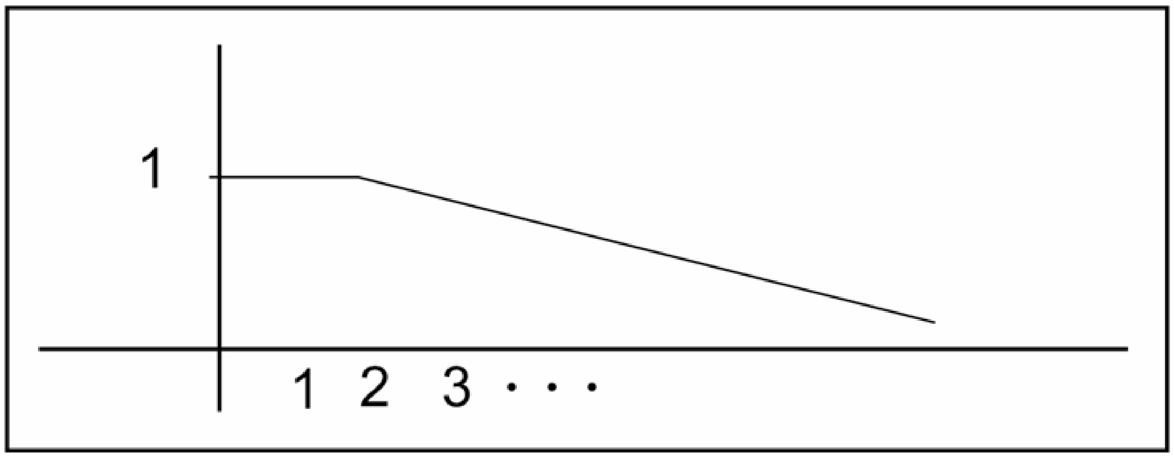
Notes

We looked at JTMSs (Justification Based Truth Maintenance Systems). “Predicate” nodes in JTMSs are pure text, there is even no information about “$\neg$”. With LTMSs (Logic Based Truth Maintenance Systems), “$\neg$” has the same semantics as logic. So what we covered was technically LTMSs.

We will not cover ATMSs (Assumption Based Truth Maintenance Systems).

Did you know that TMSs were first developed for Intelligent Tutoring Systems (ITSs)?

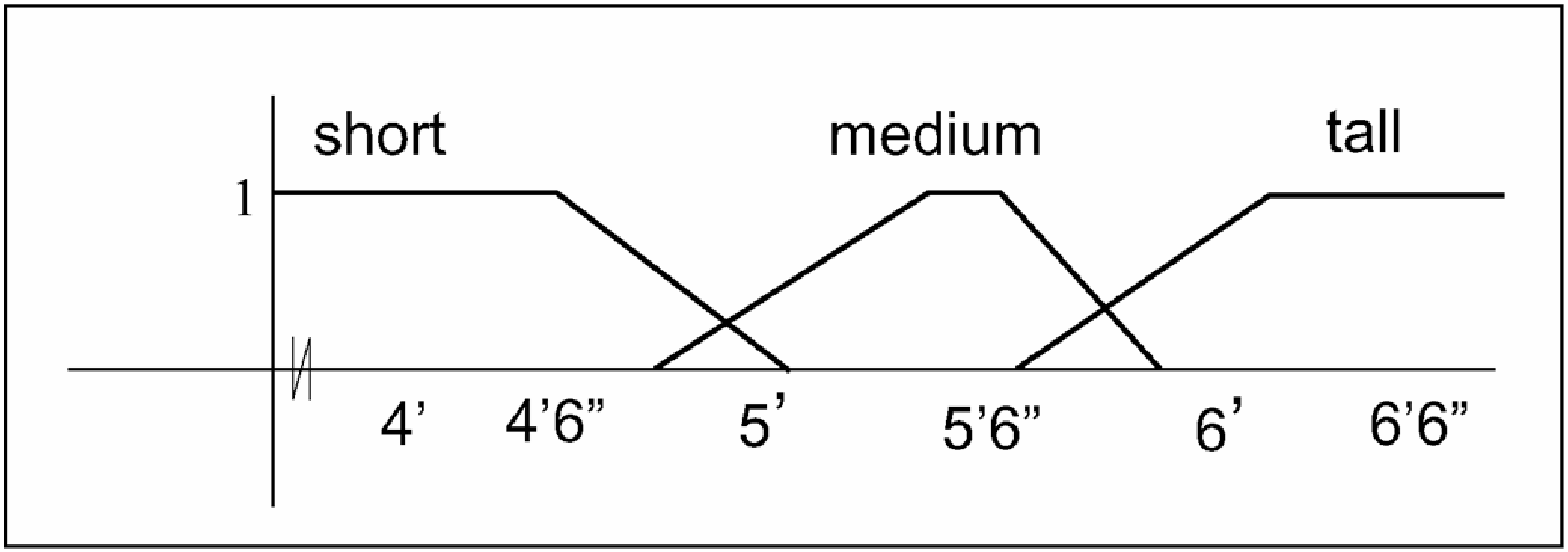
The fuzzy set representation for “small integers”



Reasoning with fuzzy sets

- Lotfi Zadeh's fuzzy set theory
- Violates two basic assumption of set theory
  - For a set S, an element of the universe either belongs to S or the complement of S.
  - For a set S, and element cannot belong to S or the complement S at the same time
- Jack is 5'7". Is he tall? Does he belong to the set of tall people? Does he not belong to the set of tall people?

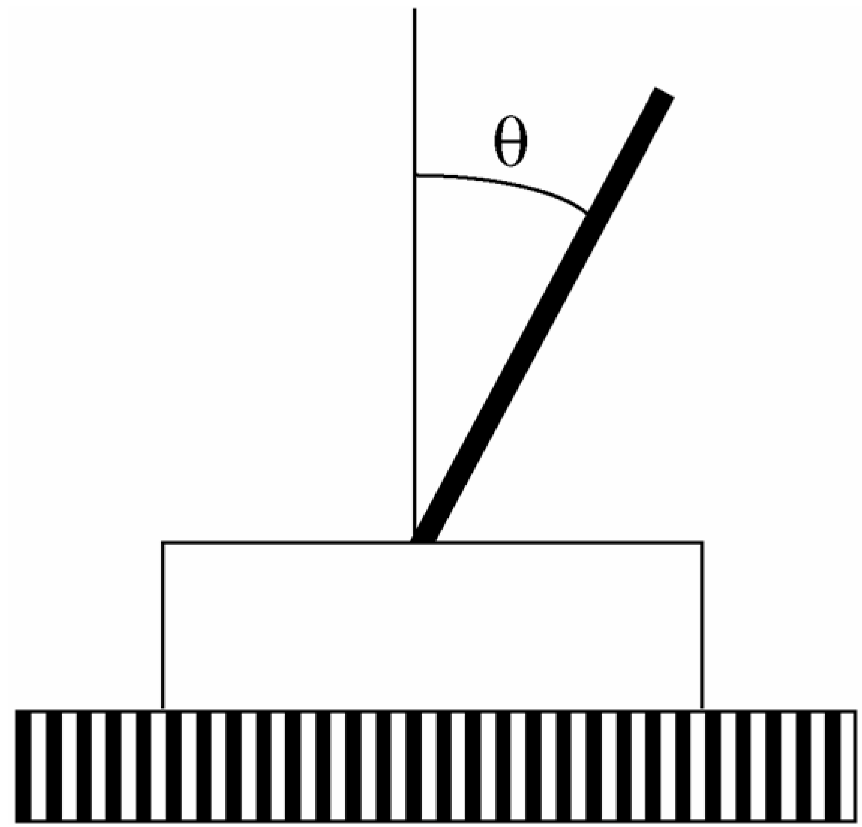
A fuzzy set representation for the sets short, median, and tall males



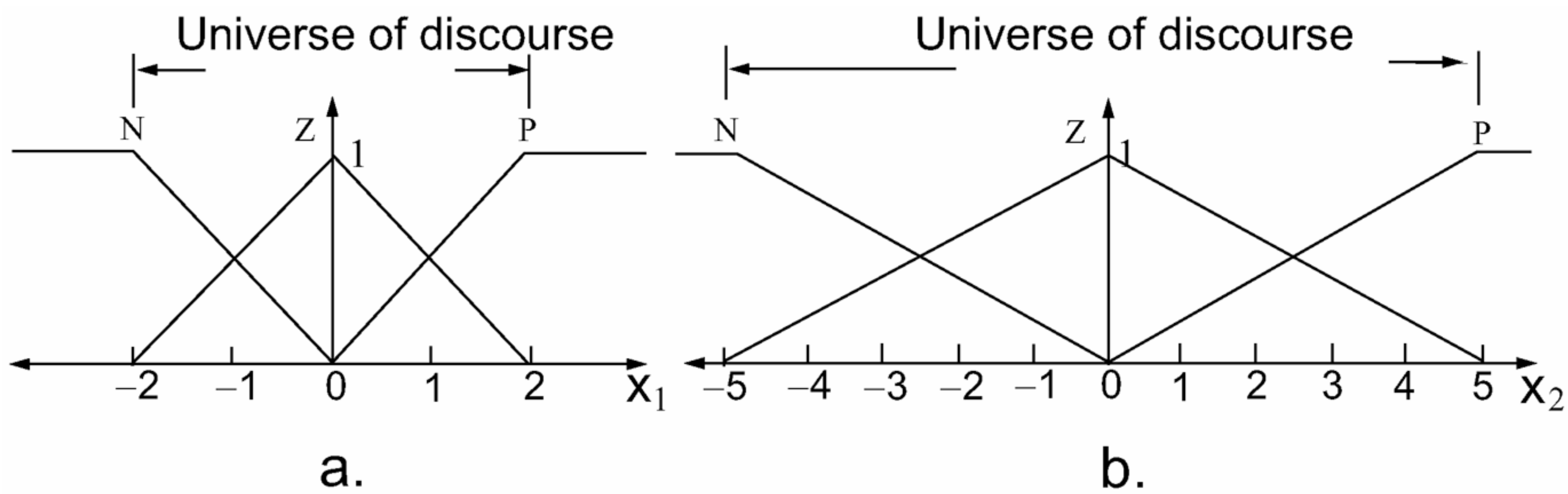
Fuzzy logic

- Provides rules about evaluating a fuzzy truth, T
- The rules are:
  - $T(A \wedge B) = \min(T(A), T(B))$
  - $T(A \vee B) = \max(T(A), T(B))$
  - $T(\neg A) = 1 - T(A)$
- Note that unlike logic $T(A \vee \neg A) \neq T(True)$

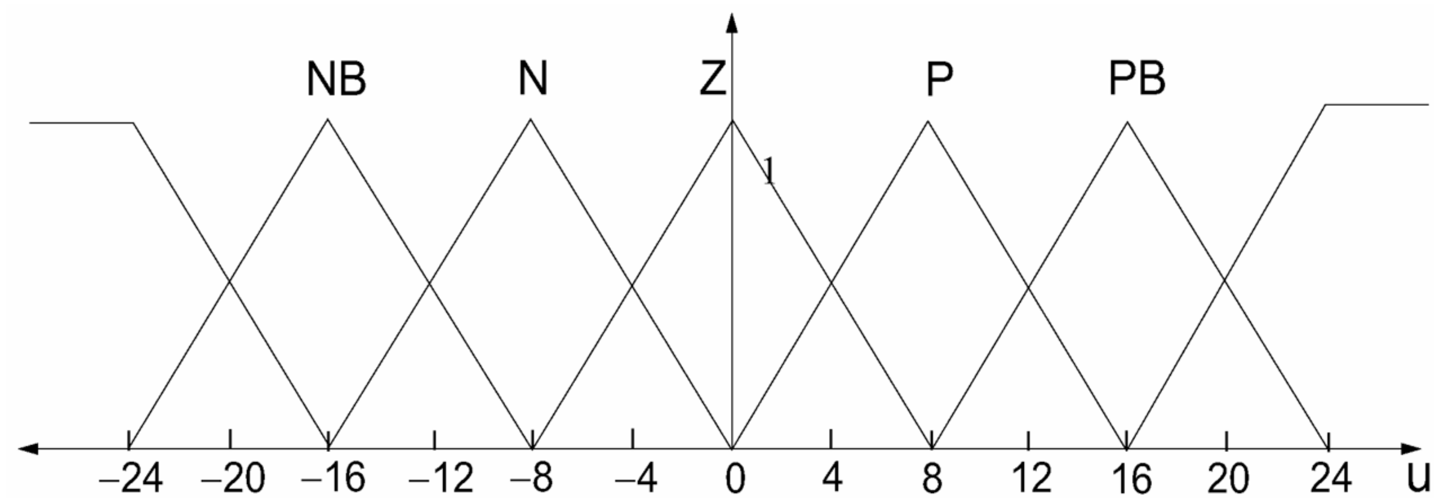
The inverted pendulum and the angle θ and $d\theta/dt$ input values.



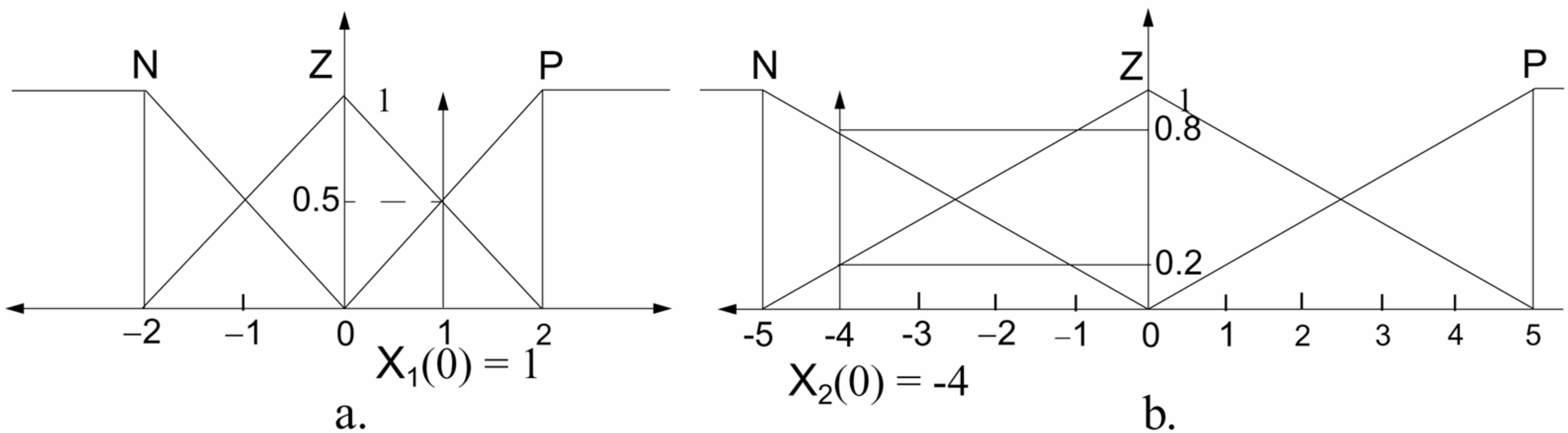
The fuzzy regions for the input values θ (a) and dθ/dt (b)



The fuzzy regions of the output value u, indicating the movement of the pendulum base



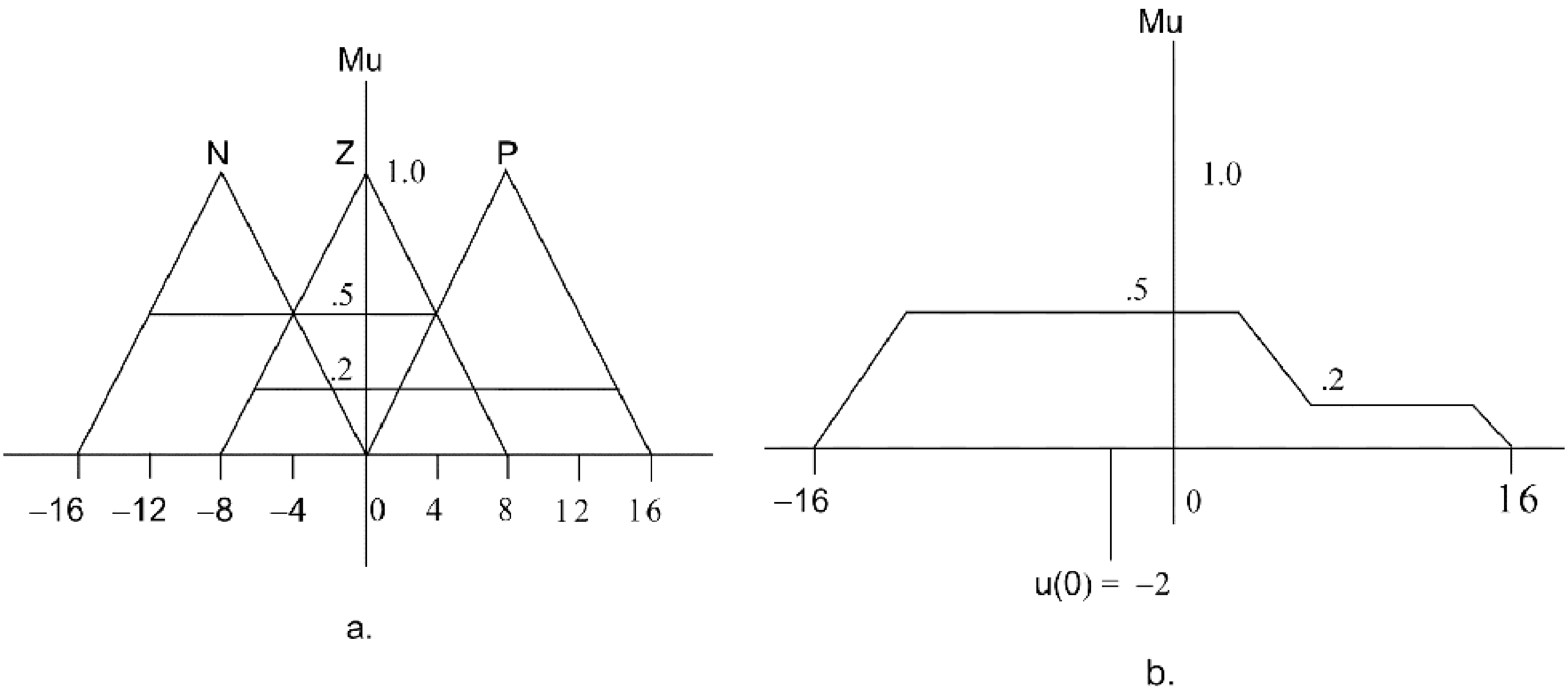
The fuzzification of the input measures x1=1, x2 = -4



The Fuzzy Associative Matrix (FAM) for the pendulum problem

x_1 \ x_2	P	Z	N
P	PB	P	Z
Z	P	Z	N
N	Z	N	NB

The fuzzy consequents (a), and their union (b)



The centroid of the union (-2) is the crisp output.

Minimum of their measures is taken as the measure of the rule result

IF $x_1 = P$ AND $x_2 = Z$ THEN $u = P$

$\min(0.5, 0.2) = 0.2\ P$

IF $x_1 = P$ AND $x_2 = N$ THEN $u = Z$

$\min(0.5, 0.8) = 0.5\ Z$

IF $x_1 = Z$ AND $x_2 = Z$ THEN $u = Z$

$\min(0.5, 0.2) = 0.2\ Z$

IF $x_1 = Z$ AND $x_2 = N$ THEN $u = N$

$\min(0.5, 0.8) = 0.5\ N$

Procedure for control

- Take the crisp output and fuzzify it
- Check the Fuzzy Associative Matrix (FAM) to see which rules fire
(4 rules fire in the example)
- Find the rule results
  - ANDed premises: take minimum
  - ORed premises: take maximum
- Combine the rule results
(union in the example)
- Defuzzify to obtain the crisp output
(centroid in the example)

Comments

- “fuzzy” refers to sets (as opposed to *crisp* sets)
- Fuzzy logic is useful in engineering control where the measurements are imprecise
- It has been successful in commercial control applications: automatic transmissions, trains, video cameras, electric shavers
- useful when there are small rule bases, no chaining of inferences, tunable parameters
- The theory is not concerned about how the rules are created, but how they are combined
- The rules are not chained together, instead all fire and the results are combined