Carnegie Mellon University
Language Technologies Institute

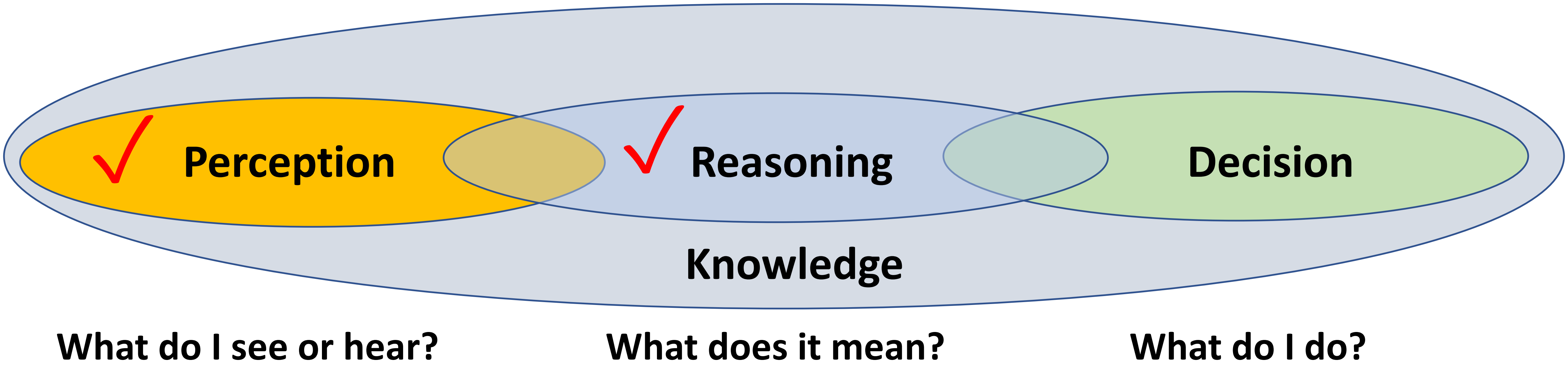
Artificial Intelligence

Class 2

Anatole Gershman

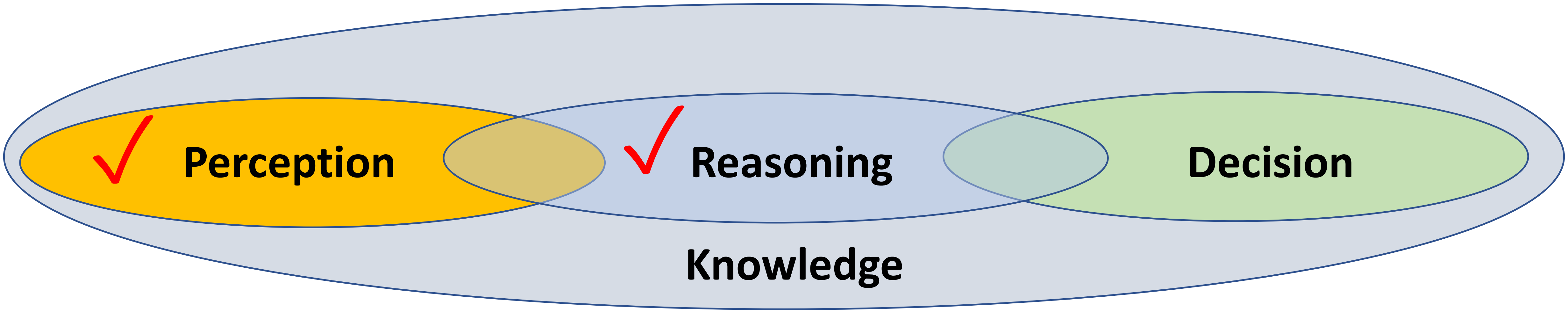
Cognitive Tasks

Carnegie Mellon University
Language Technologies Institute



- Perception and integration of perceived information require reasoning
- Reasoning is application of logic to a model – reasoning is impossible without a model
- Probability theory is an extension of logic

Cognitive Tasks



What do I see or hear?

What does it mean?

What do I do?

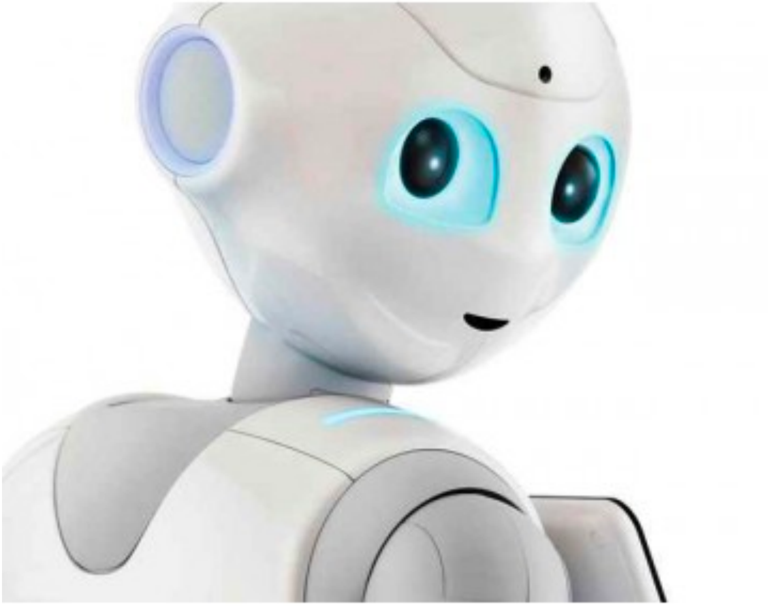
Alexa, who was the director of Avatar?

Why is this person asking this question?

Decision

Carnegie Mellon University
Language Technologies Institute

Having updated its beliefs about reality, the robot is ready to make decisions and perform actions

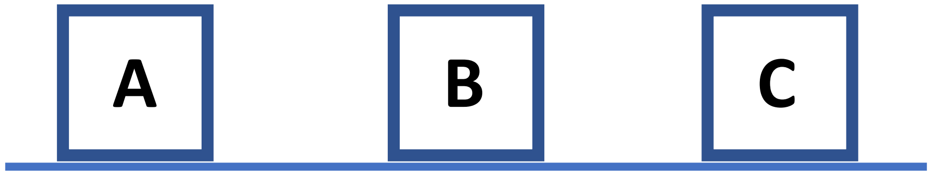


- **The robot needs to decide which action to perform next**
- **Actions are driven by goals**
- **Actions have preconditions**
- **Actions change the state of the world**

Decision

Block World

Current State

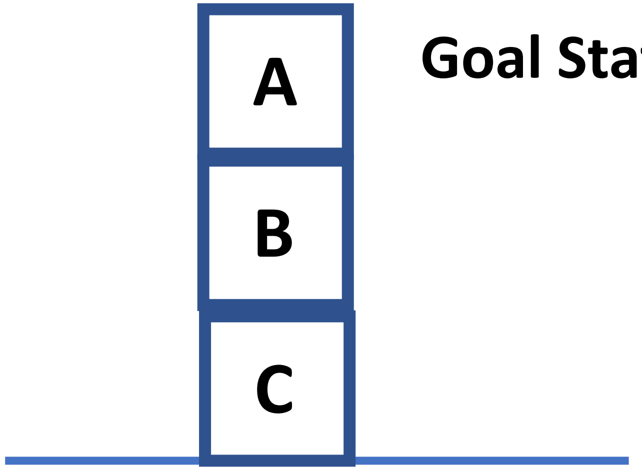


On(A, Table), FreeTop(A)

On(B, Table), FreeTop(B)

On(C, Table), FreeTop(C)

Goal State



On(A, B)

On(B, C)

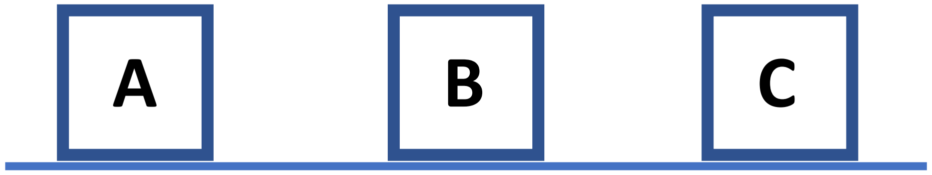
On(C, Table)

Decision

Carnegie Mellon University
Language Technologies Institute

Block World

Current State



On(A, Table), FreeTop(A)

On(B, Table), FreeTop(B)

On(C, Table), FreeTop(C)

Operator

Move(X, Y)

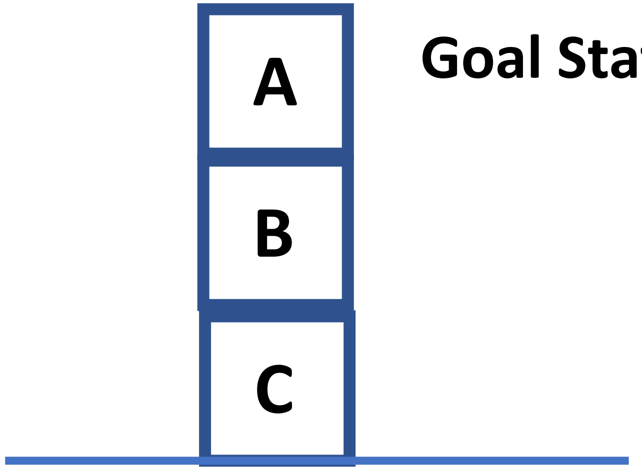
Preconditions:

FreeTop(X)
FreeTop(Y)

Results:

On(X, Y) = True
FreeTop(Y) = False

Goal State



On(A, B)

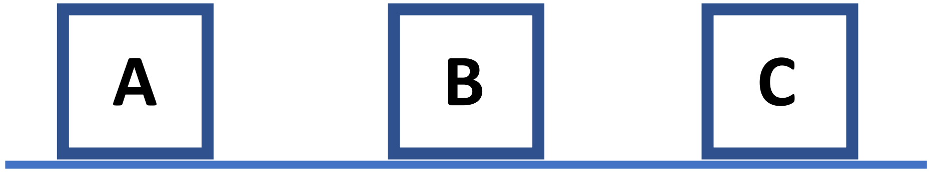
On(B, C)

On(C, Table)

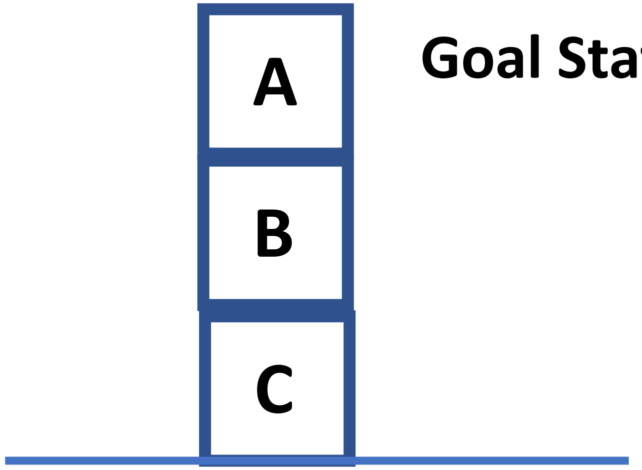
Decision

Block World

Current State



Goal State



It would be great if we could make one move that would satisfy the goal conditions

If not, we can choose the move that satisfies the greatest number of goal conditions

Both move(A, B) and move(B, C) seem equally good

On(A, B)

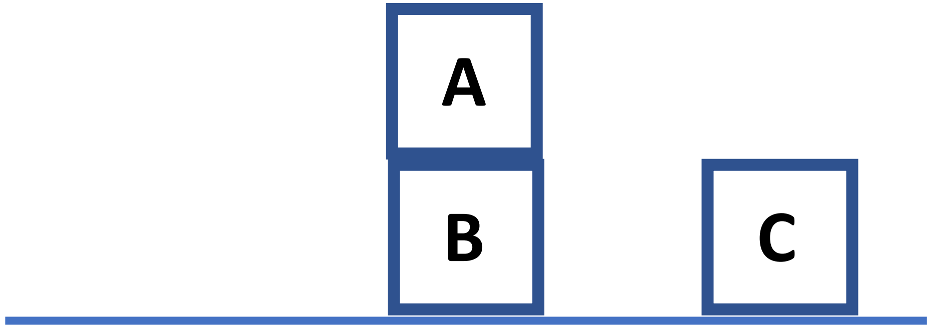
On(B, C)

On(C, Table)

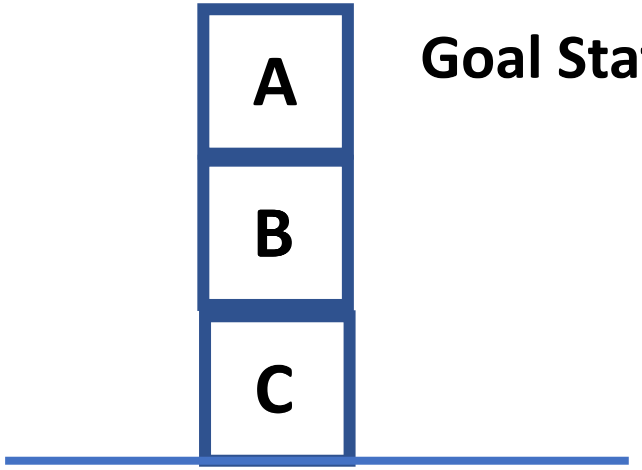
Decision

Block World

Current State



Goal State



On(A, B)

On(B, C)

On(C, Table)

Suppose, the robot chose Move(A, B)

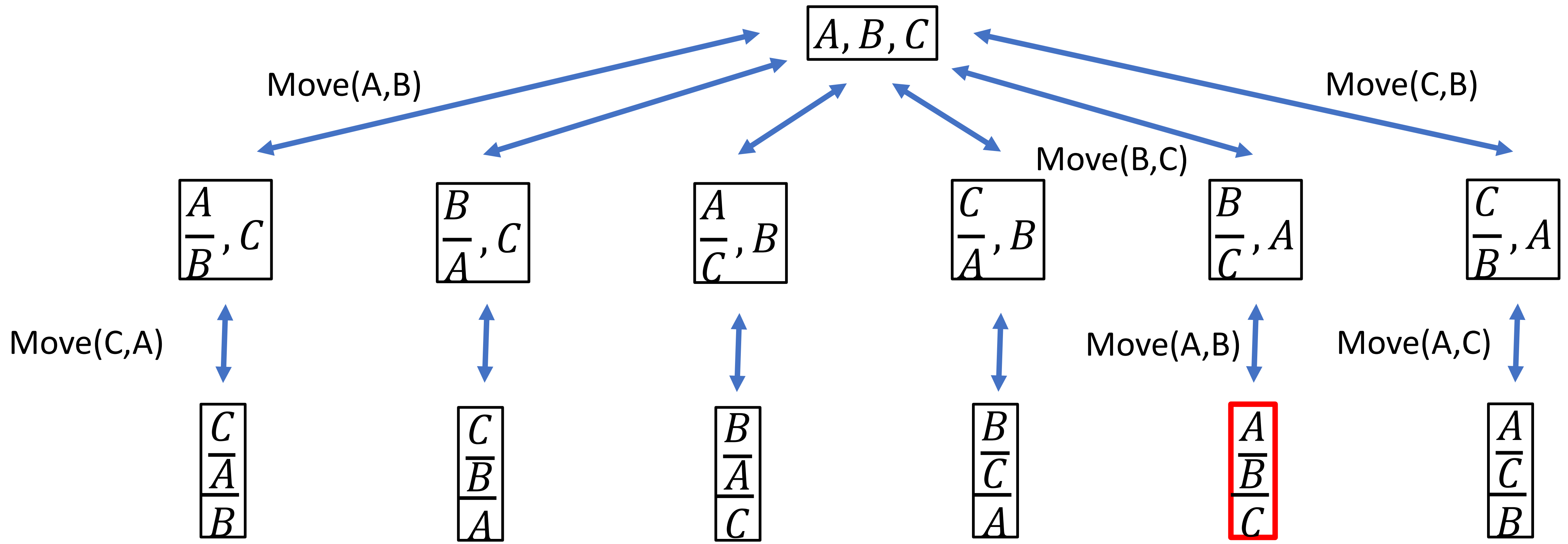
Now it will be stuck – no move leads closer to the goal!

Greedy actions don't always lead to the goal

The robot needs planning!

Decision

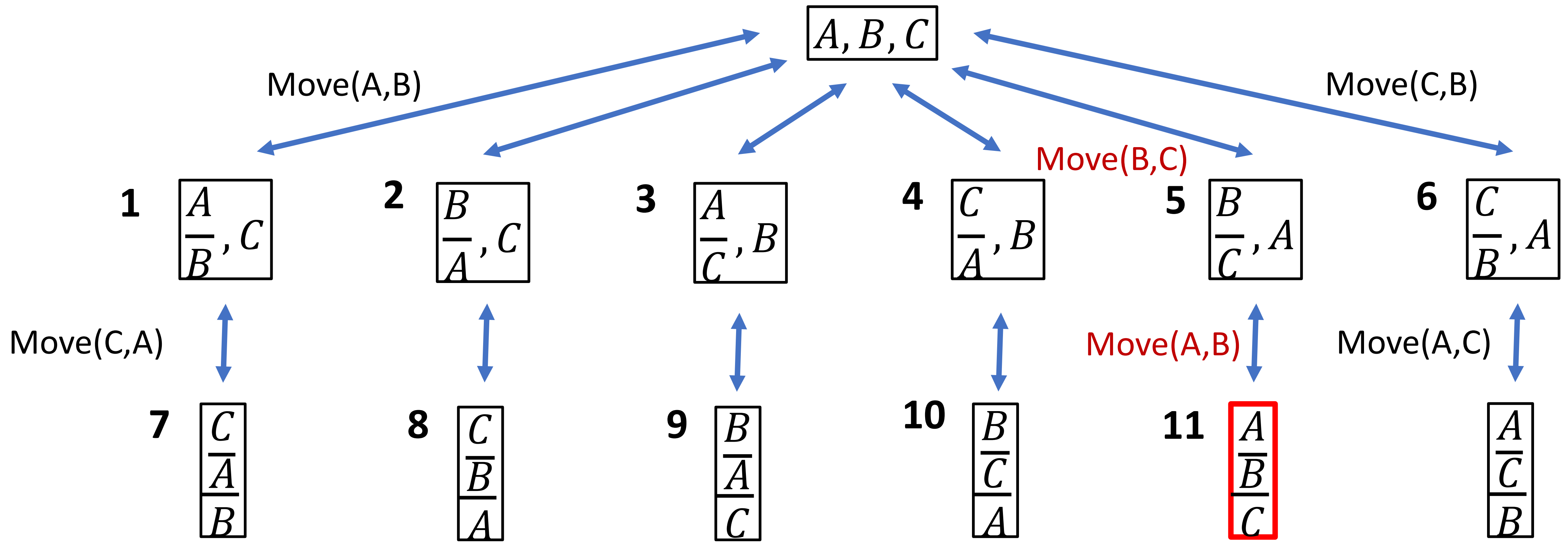
Block World – Planning Space



Carnegie Mellon University
Language Technologies Institute

Decision

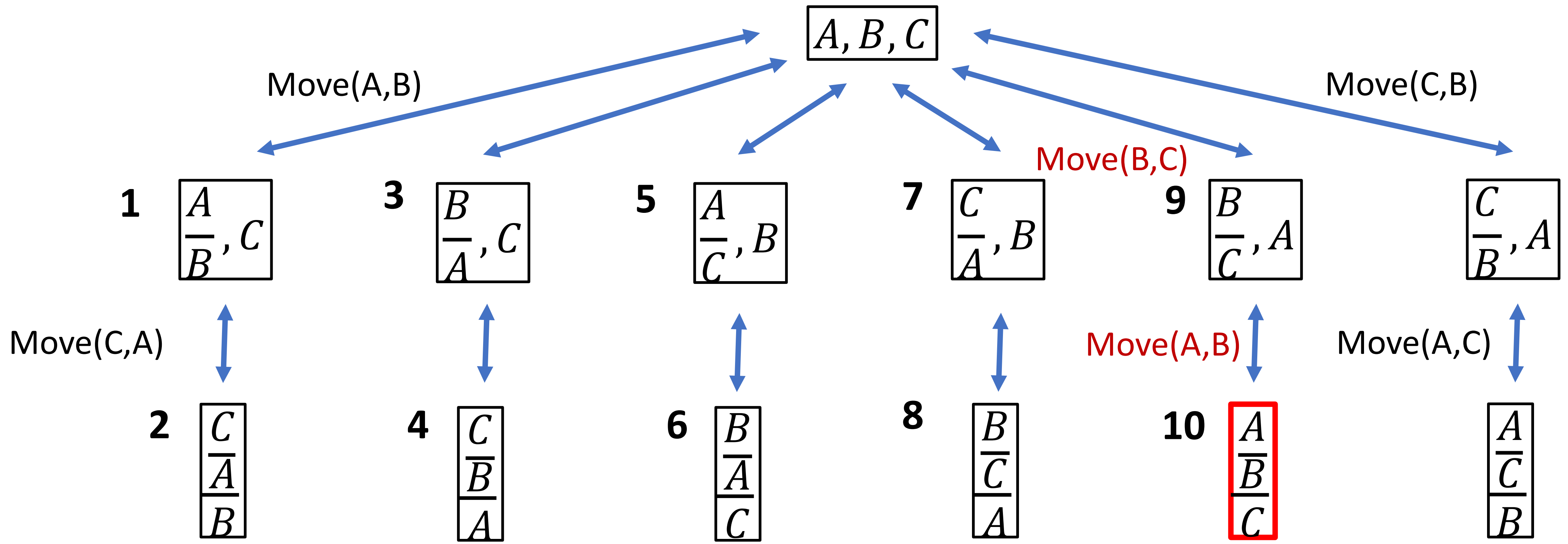
Breadth-first search



Breadth-first search will always find the shortest path to the goal

Decision

Depth-first search

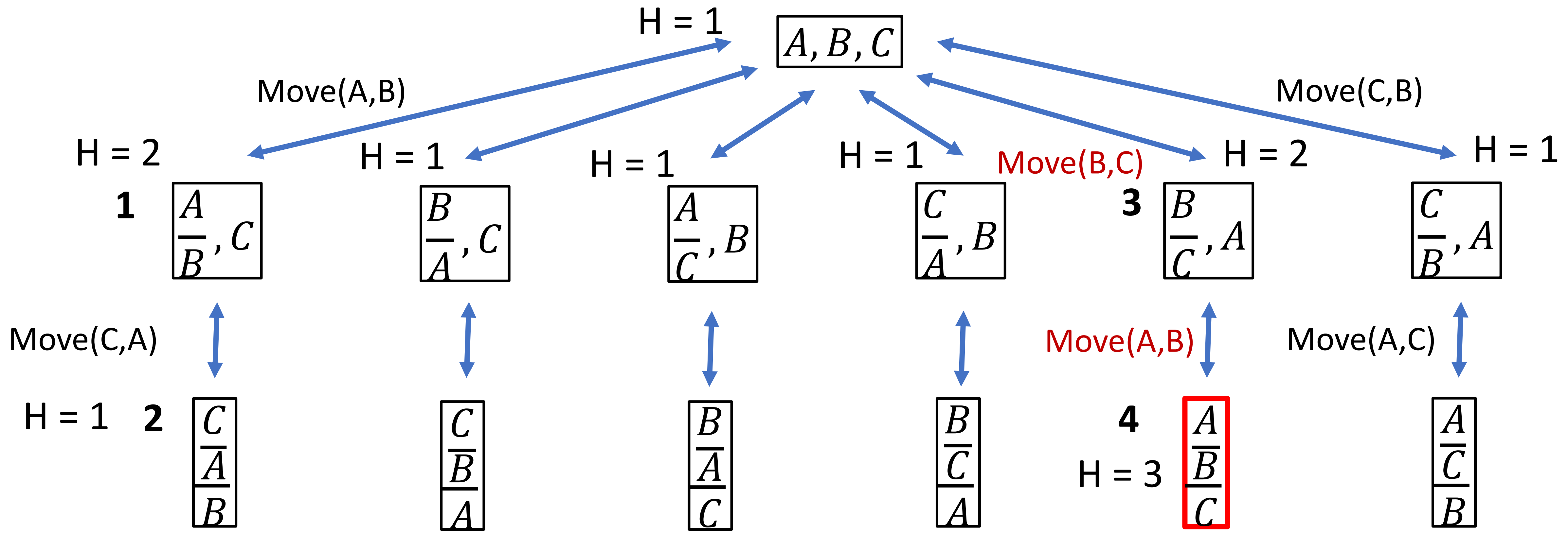


Depth-first search will always find the goal, but not necessarily the shortest path

Decision

Carnegie Mellon University
Language Technologies Institute

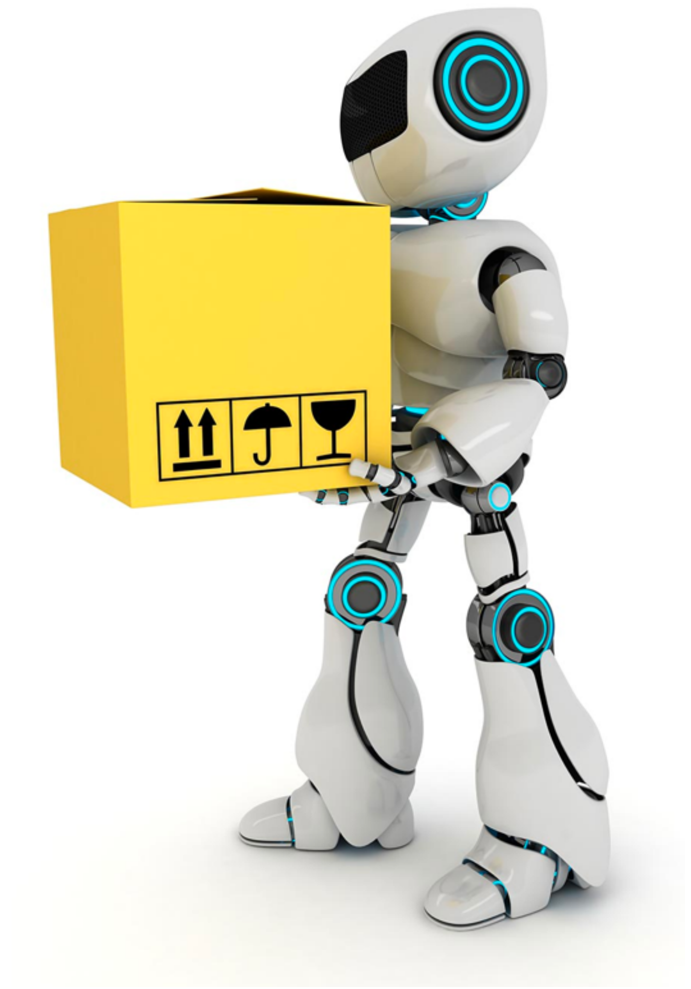
Directed search



In this example, the value of the state is the number H(s) of satisfied goal conditions
You will learn about various search algorithms in this course

Decision

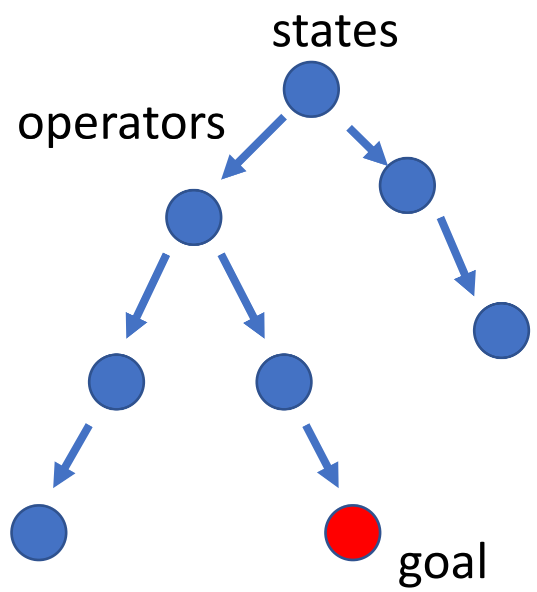
Planning



- Given a **description of the world**
  - e.g., where things are in a floor map
- And a set of **potential actions** (operators)
  - e.g., pick up an object, move to the next room, …
- And a **goal to achieve**
  - e.g., fetch me my iPad
- Find a **sequence of actions** to achieve the goal
  - e.g., move to study, go to desk, pick up iPad, …

Decision

Planning as problem solving

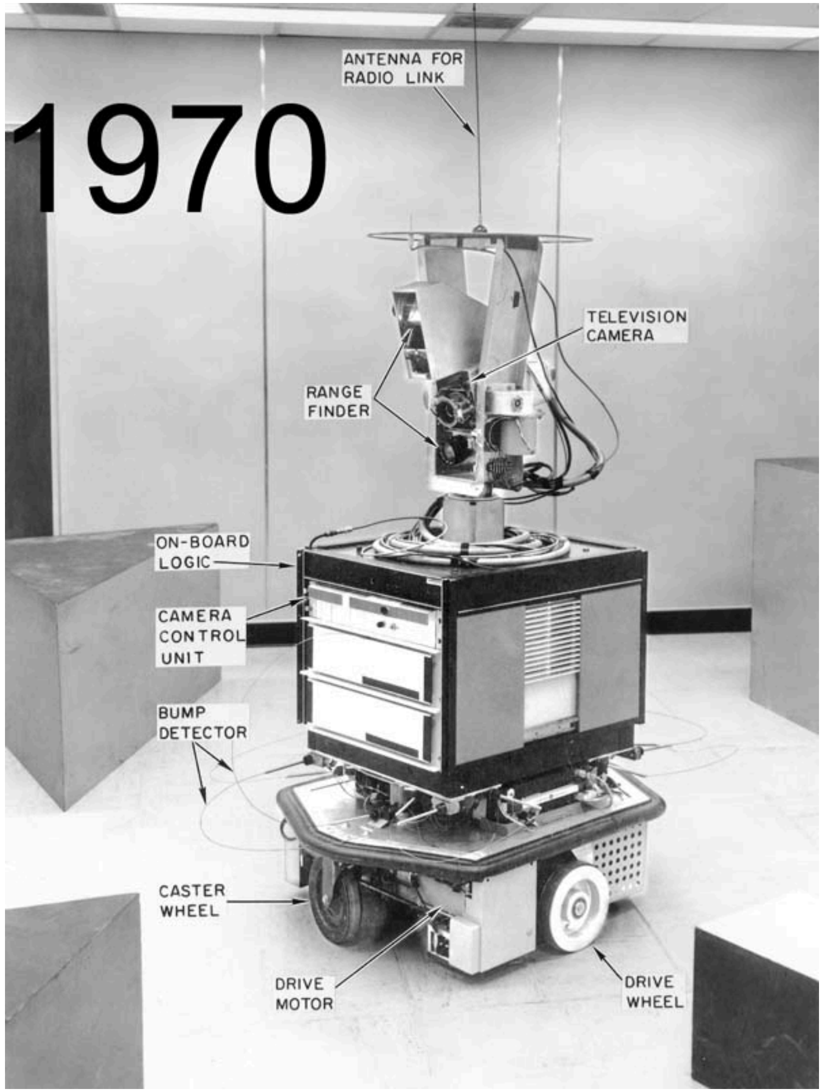


- **The states, including the goal are represented as logical formulas,** e.g., On(A,B) ∧ On(B,C)
- **Operators change states; they have preconditions and cost**
- **A plan is a sequence of operator applications that leads the goal**
- **We want to minimize the cost of the plan**
- **A good estimate of the cost of the plan going through state S is the cost of getting to S from the start plus the size of the difference between the logical representation of the goal and the logical representation of S (means-ends analysis)**

$$h(s) = Cost(Start, s) + Diff(s, Goal)$$

Decision

Shakey the Robot



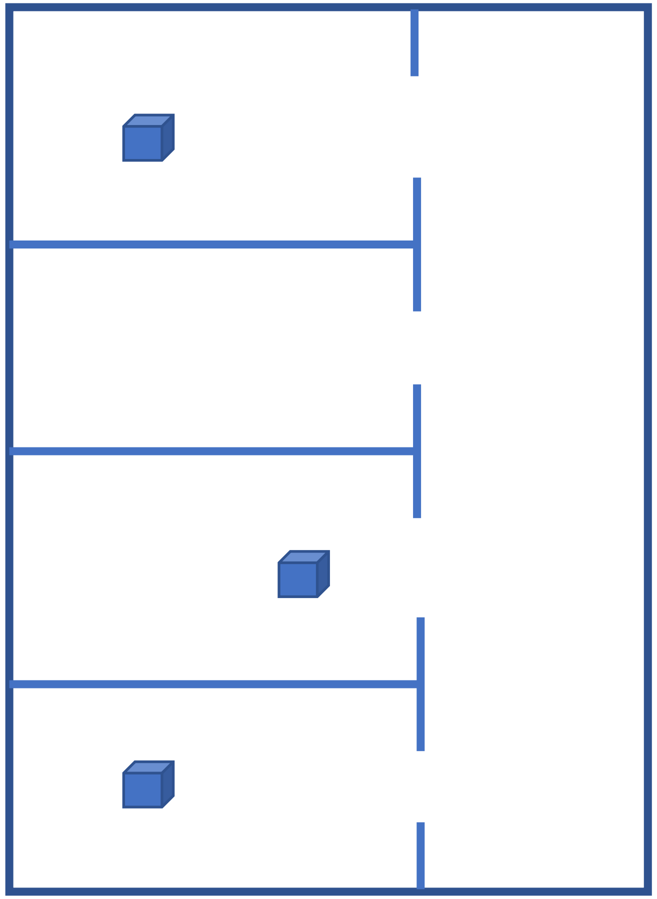
Probably the most influential early work in integrating perception, reasoning and decision was Shakey the Robot based on STRIPS – Stanford Research Institute Problem Solver (1969-72)

Let’s watch the video

Carnegie Mellon University
Language Technologies Institute

Decision

Shakey the Robot and STRIPS



- Shakey pushed boxes between rooms using cameras and range finders for perception
- It represented the state of the world as logical formulas
- It used logic to infer what was true in its world
- For planning, it computed the difference between the goal state and the current state of the world and searched for the operators that would reduce that difference (means-ends analysis)
- Shakey may seem primitive now, but it a direct ancestor of today’s robots and self-driving vehicles


BostonDynamics

Decision

Carnegie Mellon University
Language Technologies Institute

Decisions in an uncertain world

Two lottery tickets

Win $10	Win $20
Price $2	Price $3

Which one would you buy?

Decision

Carnegie Mellon University
Language Technologies Institute

Decisions in an uncertain world

Two lottery tickets

Win \$10	Win \$20
Price \$2	Price \$3
Probability p_1	Probability p_2

Which one would you buy?

It depends on your estimates of the probabilities of winning for each ticket

Ticket value = expected win – expected cost

$$V(t_1) = 10 * p_1 - 2 \qquad V(t_2) = 20 * p_2 - 3$$

p_1	p_2	V_1	V_2
.5	.5	3	7
.1	.1	-1	-1
.25	.15	.5	0

Decision

The Monty Hall Problem

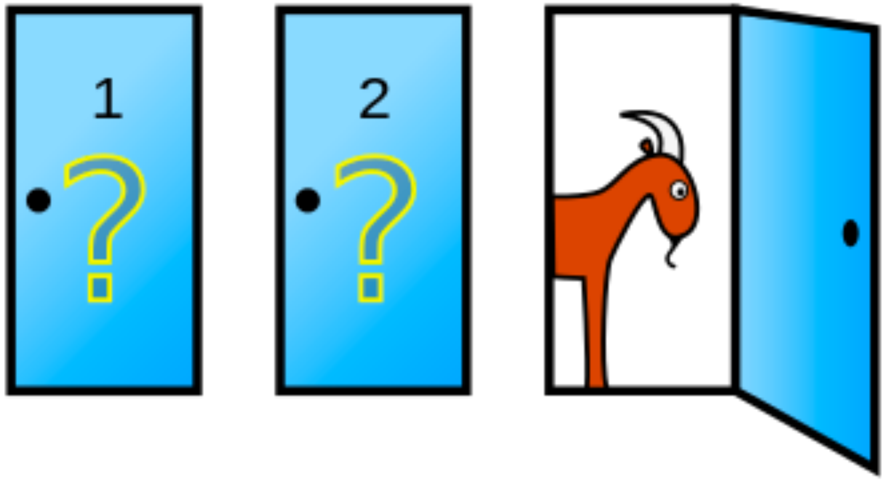
Decisions under uncertainty can be tricky

Suppose you're on a game show, and you're given the choice of three doors:

Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat.

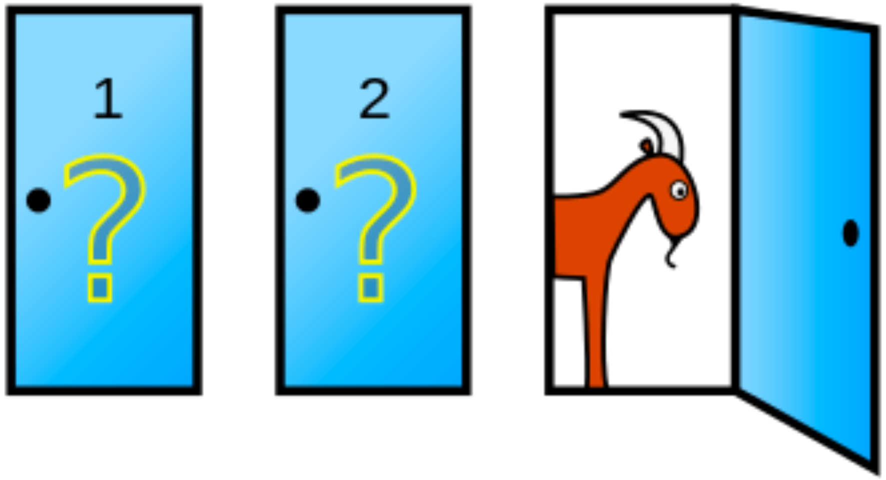
He then says to you, "Do you want to pick door No. 2?”

Is it to your advantage to switch your choice?



Decision

The Monty Hall Problem



C_i – car behind door i; $P(C_1) = P(C_2)$ even after door 3 is opened

X_i – the player chooses door i; $P(X_1) = 1$ – the player chose door 1

H_i – the host opens door i

$q = P(H_3 \mid X_1, C_1)$ – the probability of the host opening door 3 given the player chose door 1 and the car is behind door 1

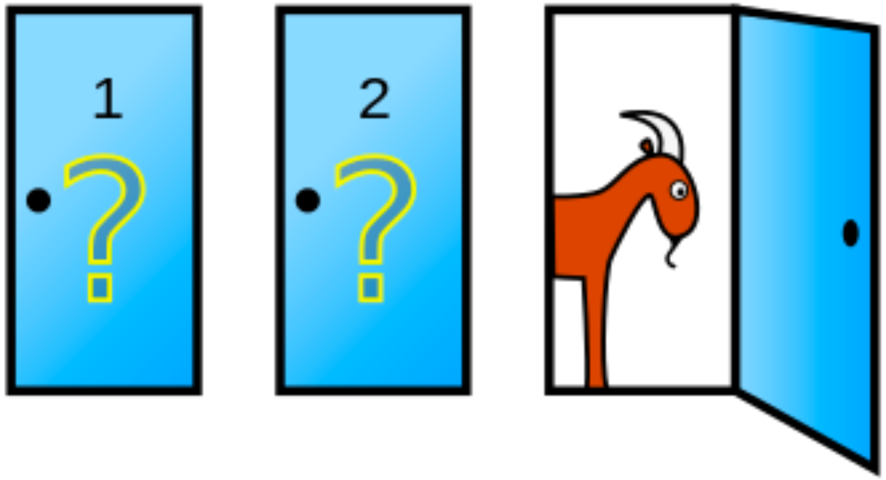
$P(H_3 \mid X_1, C_2) = 1$ – the host has no choice

$$P(C_2|X_1,H_3) = \frac{P(C_2,X_1,H_3)}{P(X_1,H_3)} = \frac{P(H_3|X_1,C_2) * P(X_1) * P(C_2)}{P(X_1,H_3,C_1) + P(X_1,H_3,C_2) + P(X_1,H_3,C_3)} =$$

$$= \frac{P(H_3|X_1,C_2) * P(X_1) * P(C_2)}{P(H_3|X_1,C_1) * P(X_1) * P(C_1) + P(H_3|X_1,C_2) * P(X_1) * P(C_2)} = \frac{1}{q+1}$$

Decision

The Monty Hall Problem



C_i – car behind door i; $P(C_1) = P(C_2)$ even after door 3 is opened

X_i – the player chooses door i; $P(X_1) = 1$ – the player chose door 1

H_i – the host opens door i

$q = P(H_3 \mid X_1, C_1)$ – the probability of the host opening door 3 given the player chose door 1 and the car is behind door 1

$P(H_3 \mid X_1, C_2) = 1$ – the host has no choice

$$P(C_2|X_1,H_3) = \frac{P(C_2,X_1,H_3)}{P(X_1,H_3)} = \frac{P(H_3|X_1,C_2) * P(X_1) * P(C_2)}{P(X_1,H_3,C_1) + P(X_1,H_3,C_2) + P(X_1,H_3,C_3)} =$$

$$= \frac{P(H_3|X_1,C_2) * P(X_1) * P(C_2)}{P(H_3|X_1,C_1) * P(X_1) * P(C_1) + P(H_3|X_1,C_2) * P(X_1) * P(C_2)} = \frac{1}{q+1}$$

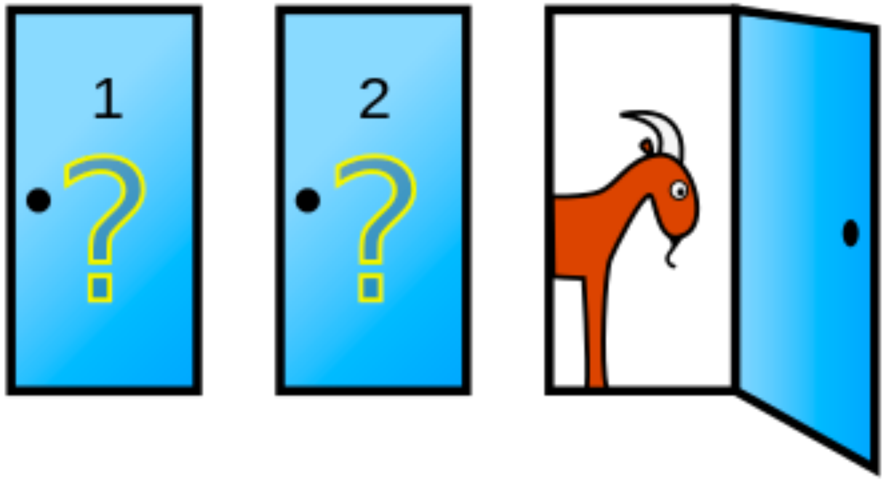
ranges between ½ and 1

You cannot lose by switching

Decision

Carnegie Mellon University
Language Technologies Institute

The Monty Hall Problem



The point of this example is that decisions require models

In this case, we needed to model the host's behavior
We did it with the parameter q – the probability of opening the right door when the host has a choice

It turns out that switching is advantageous at all values of q but this may not be the case if switching had a cost

If we don't know the value of q, we could assume that it is uniformly distributed between 0 and 1 and use it to compute the expected value of switching which is ln(2)≈.693

Decision

Planning in an uncertain world

In the real world, actions do not always lead to the expected results

When a soccer player hits the ball, it may or may not land where intended
The ball may also be unexpectedly intercepted by an opponent

Plans in the real world need to maximize the probability of reaching the goal while minimizing the expected cost

This is complicated because we cannot take into account every contingency and have to accept approximate solutions

These approximations sometimes lead to errors which are unavoidable

Decision

Plans as Schemas

If we had to perform means-ends analysis every moment of our lives, we would be in real trouble:

I need to go to New York City, where should I put my right foot in my next step?

When we need to go from Pittsburgh to New York, first, we decide if we will fly or drive

If we decide to fly, we decide how to get to the airport: to take the bus, to drive or to ask a friend for a ride

We don't need much planning to get to the bus stop – we know where it is and how to get there

Decision

Scripts

Situations such as restaurants are even more stereotypical and require almost no reasoning

When we walk into a restaurant, we don't plan how we are going to get the food from the kitchen

We know the restaurant script: sitting, getting menus, ordering, waiting, eating, and paying

We don't try to figure out why a smiling woman with a glossy booklet is approaching our table. We know it is a waitress bringing the menu.

Decision

We have schemas for most common events

Plane-travel(Traveler=X, Origin=L1, Destination=L2)

Preconditions:	Airport(L1)
	Airport(L2)
	Distance(L1, L2) > 250km
Goal:	At(X,L2)
Events:	**Select-flight**(Flight=F, Origin=L1, Destination=L2)
	Buy-ticket(Traveler=X, Flight=F)
	Travel(Traveler=X, Destination=L1)
	Get-gate-info(Airport=L1, Flight=F, Gate=G)
	Security-check(Traveler=X, Airport=L1)
	Travel(Traveler=X, Destination=G)
	…

The sequence of events is not always linear – it may be a graph; some events are optional

Decision

Carnegie Mellon University
Language Technologies Institute

How do we apply schemas?

Suppose, our robot needs to go from CMU to Rockefeller Center in New York City

It may have millions operators/schemas at various levels of abstractions

It seems reasonable to index schemas by their goals

The first goal is to plan the trip

The Plan-Travel(Origin=X, Destination=Y) schema will select the best mode of transportation between X and Y and invoke a more specific schema such as Plane-travel.

Experienced travelers between Pittsburgh and New York may have very specific schemas – more specific schemas for the goal are tried first, before the more general ones

Many elements of the plan such as how to get from LaGuardia to Rockefeller Center may be left un-instantiated until execution time when more information is available

Decision

We use schemas to understand events

We may hear:

Bob bought a Delta ticket to LaGuardia. He arrived in New York in time for dinner.

Did Bob fly?

Decision

We use schemas to understand events

We may hear:

Bob bought a Delta ticket to LaGuardia. He arrived in New York in time for dinner.

Did Bob fly?

Of course, he did. But how do we know it?

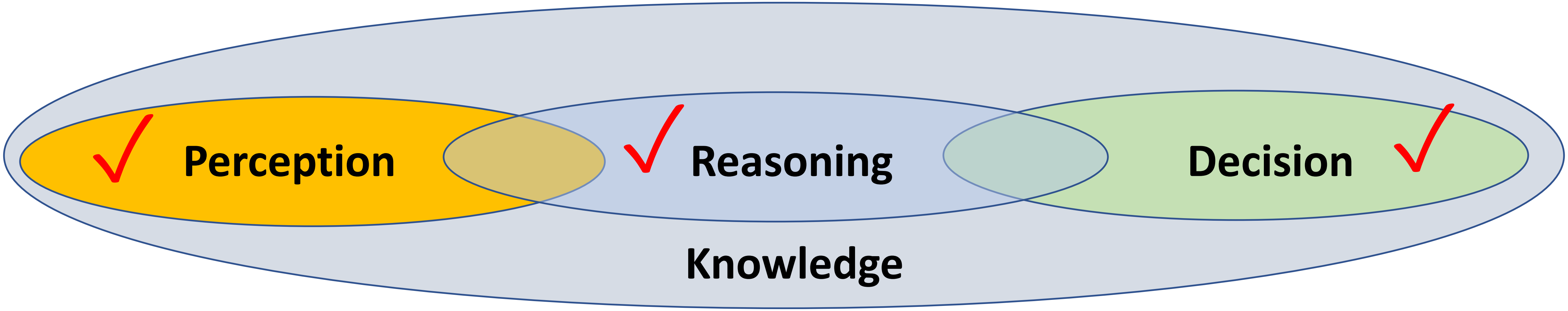
Delta is an airline; LaGuardia is an airport in New York; Bob bought an airline ticket

This fits the Plane-travel schema

We instantiate the variables and assume that the other events in the schema also happened

Cognitive Tasks Require a lot of Knowledge

Carnegie Mellon University
Language Technologies Institute



What kinds of knowledge does a robot need to perform cognitive tasks?

Knowledge includes models and facts

that a child is younger than its parents, but not by 100 years

that cows have no wings and tend to weigh over 400lb

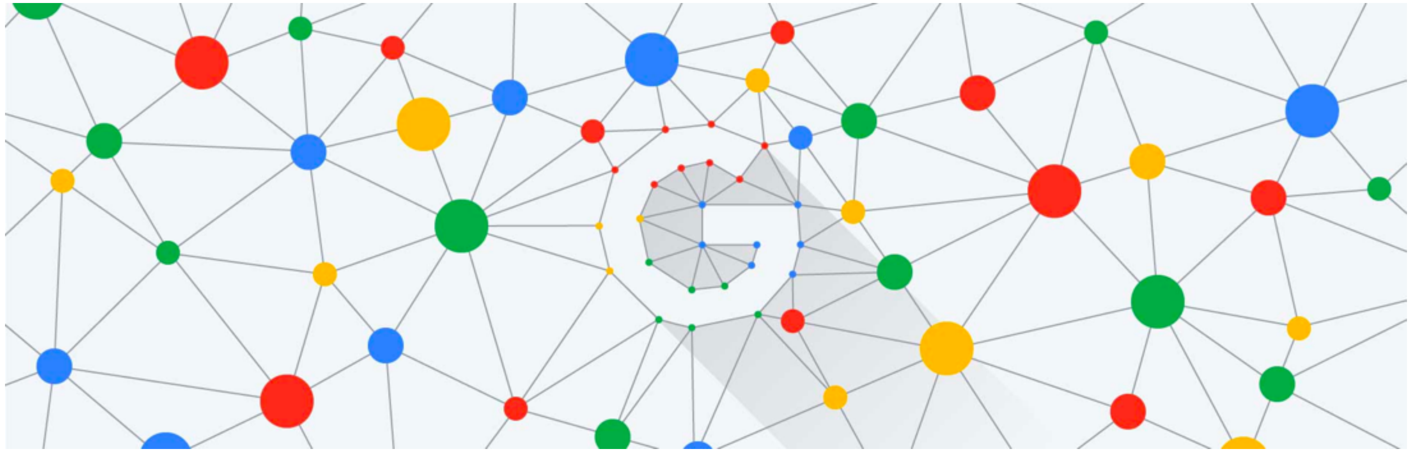
that President Xi is about 60 and was born somewhere in China

and many other things most of which are approximate and uncertain

Knowledge

Knowledge Graphs

- **Knowledge representation was the central concern of AI in 1970-80**
- **The success of statistical Machine Learning eclipsed this concern for a while**
- **But now it is coming back, most frequently in the form of Knowledge Graphs**

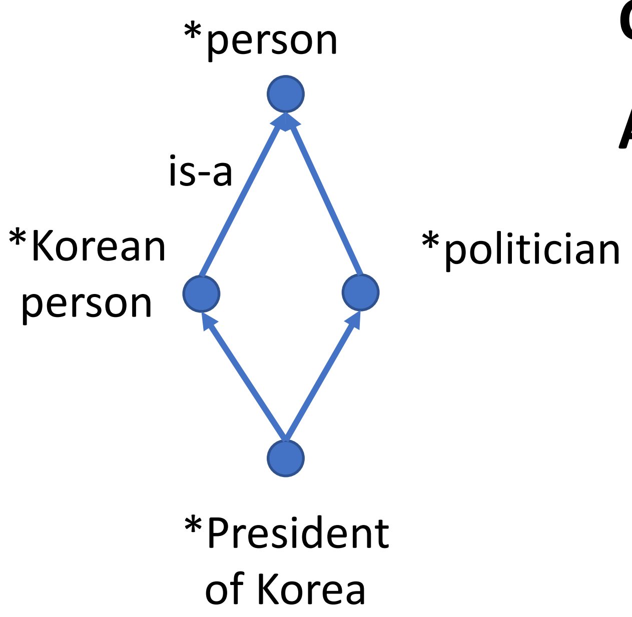


Ontology defines and connects concepts, their properties and relations (e.g., *person)

Instances (e.g., my friend Bob)

Knowledge

Ontologies describe concepts



Concepts have parents (possibly multiple) and attributes

Attributes are either other concepts or literals (e.g., strings, numbers)

*person	
age:	*number
first-name:	*string
last-name:	*string
nationality:	*country
spouse:	*person
….	

*Plane-travel	
Traveler:	*person
Origin:	*airport
Destination:	*airport
Flight:	*flight

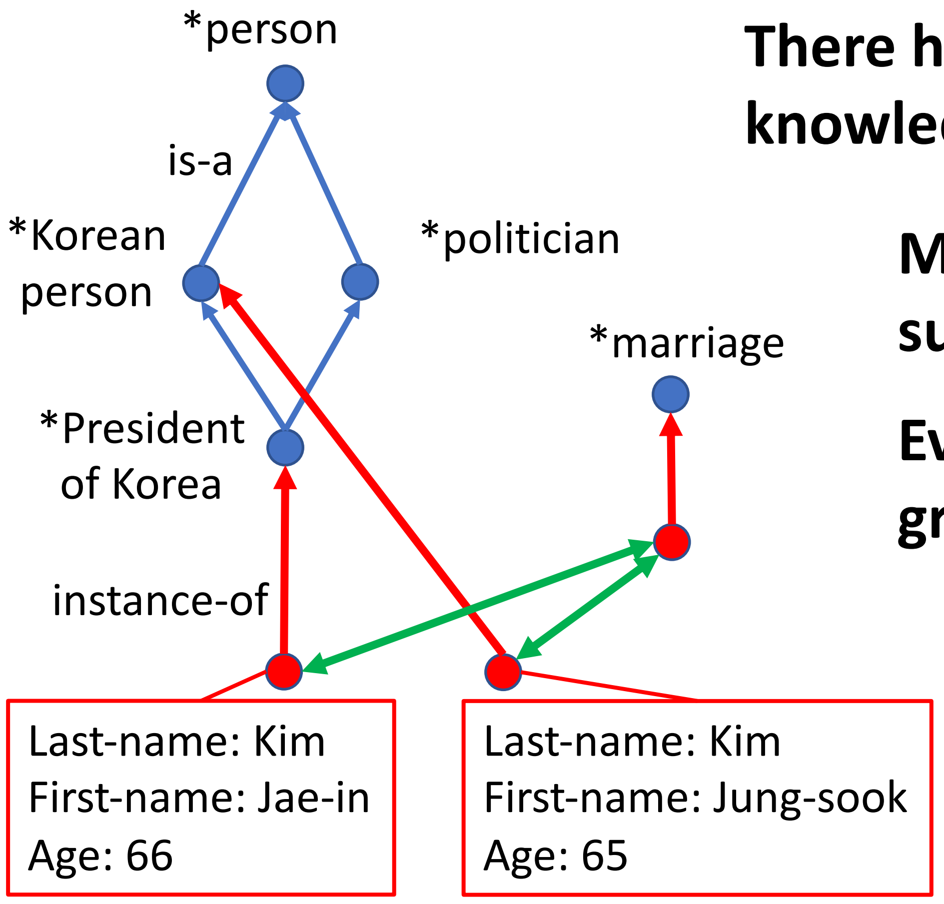
Attributes specify the allowed values for concept instances

(e.g., nationality must be an instance of a country)

Knowledge

Carnegie Mellon University
Language Technologies Institute

Instances represent real objects and facts



There have been many attempts to build extensive universal knowledge graphs, none of them universally accepted

More successful are domain-specific knowledge graphs such as medical terms

Every project ends up developing its own knowledge graph, partially based on some existing ones

Knowledge

There are two main problems with existing Knowledge Graphs:

- **Representation of uncertainty**
- **Representation of dependencies among knowledge elements**

How old is Vladimir Putin? *Probably in his mid-60s*

Where was he born? *Most likely in St. Petersburg but maybe somewhere else in Russia*

Most of what we know is to various degrees uncertain, yet we can function rather well

Knowledge

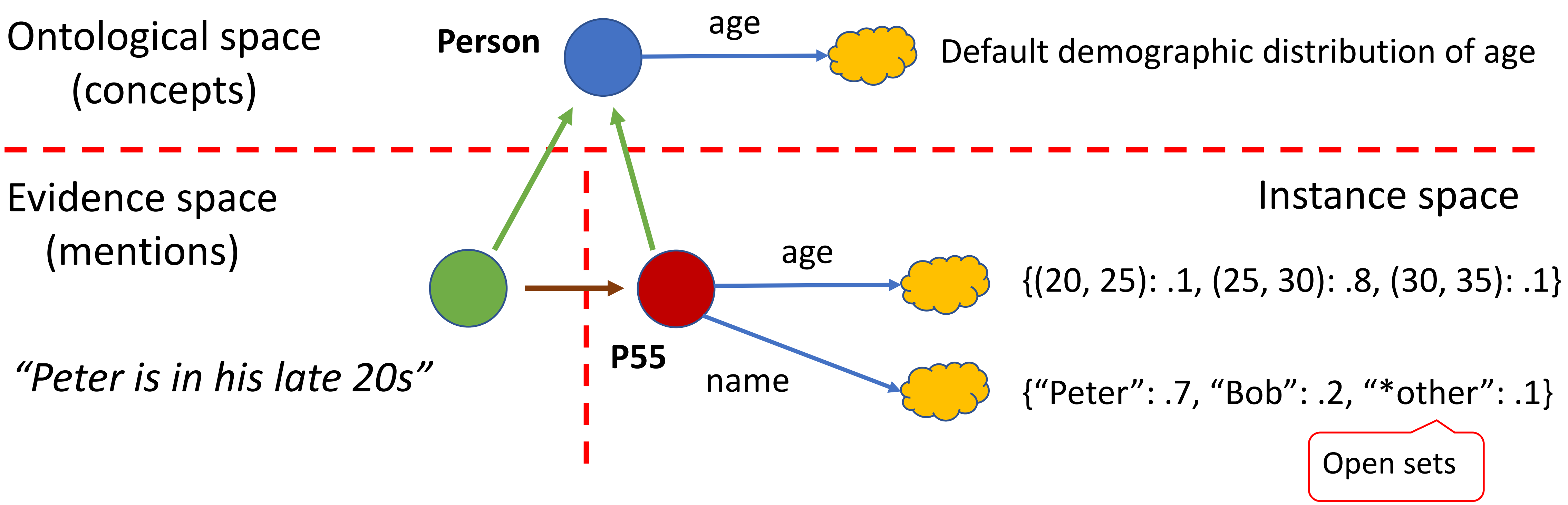
Belief Graphs

Carnegie Mellon University
Language Technologies Institute

Belief Graphs represent what the robot (the system) believes to be true in the world

Uncertain knowledge about node attributes is represented as probability distributions

Belief Graphs consist of 3 interconnected spaces: Ontological, Instance and Evidence



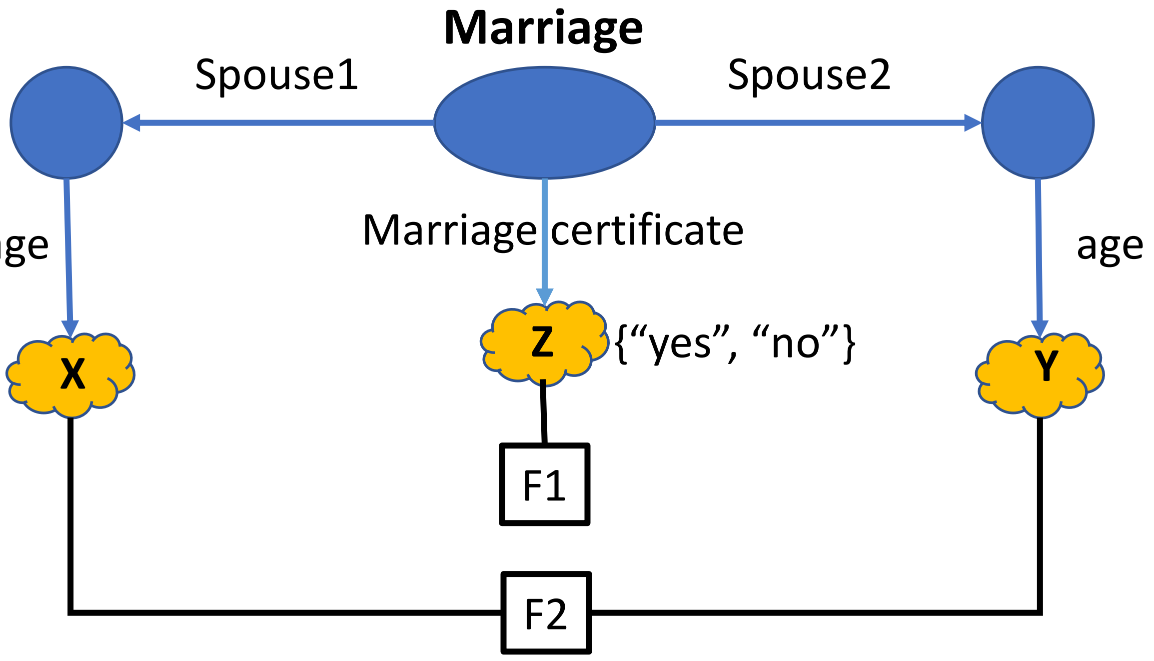
Knowledge

Carnegie Mellon University
Language Technologies Institute

Dependencies among knowledge elements are captured by factors

A factor computes the likelihood that a relation is true

Naïve Factorization Assumption



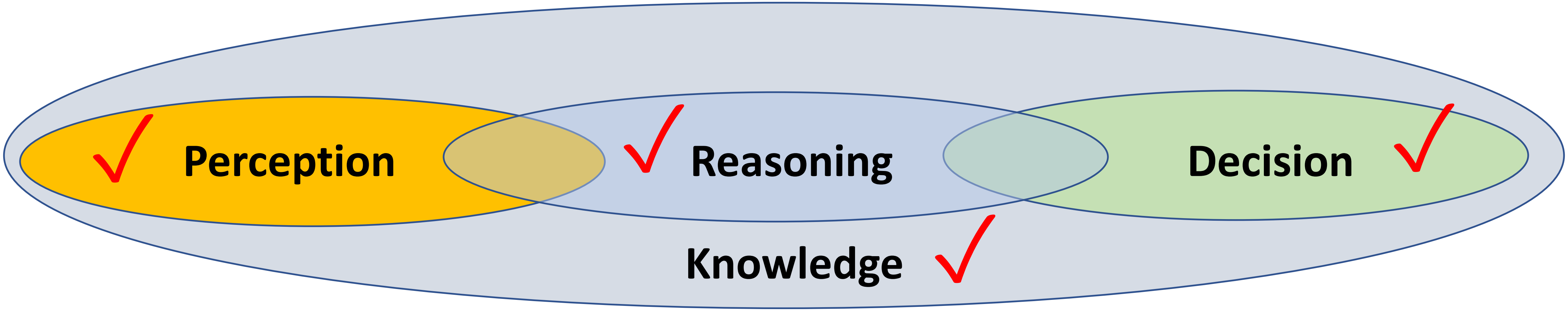
P(Marriage = True | x, y, z) ∝ $F1(z) * F2(x, y)$

$$F1(z) = \begin{cases} .9, if\ z = \text{"yes"} \\ .1, if\ z = \text{"no"} \end{cases}$$

$$F2(x, y) = \begin{cases} .80, if \quad |x - y| \leq 10 \\ .15, if\ 10 < |x - y| \leq 20 \\ .05, if\ \ 20 < |x - y| \end{cases}$$

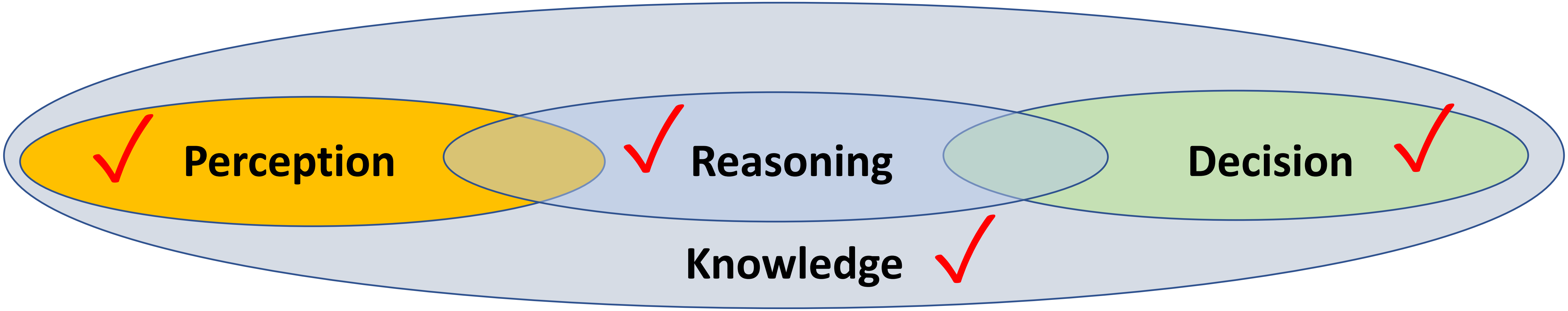
Factor graphs for belief propagation are constructed from relation factors

Cognitive Tasks



What about learning? How do we acquire the necessary knowledge?

Learning



As humans, we learn from a combination of personal experience and communication with other humans

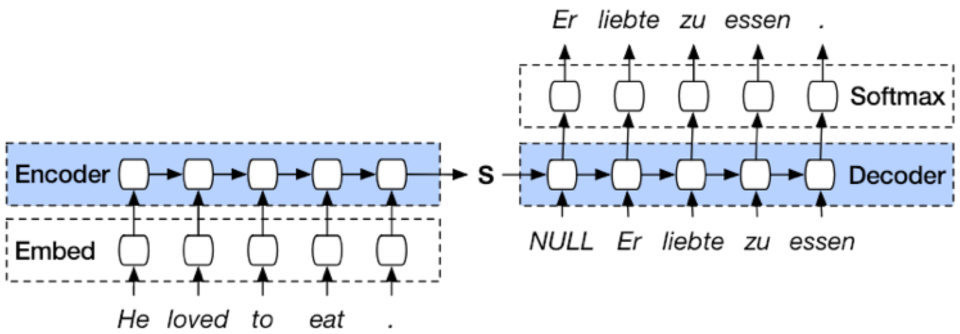
Both are necessary – without communicating to other humans we would be smart monkeys

Natural language is our key instrument for knowledge acquisition

Over time, we build elaborate long-term knowledge structures and modify them as we acquire new information

Compare this to the "End-to-End" approach

Carnegie Mellon University
Language Technologies Institute



With enough inputs and outputs, a neural net can learn the mapping – no models or knowledge needed!

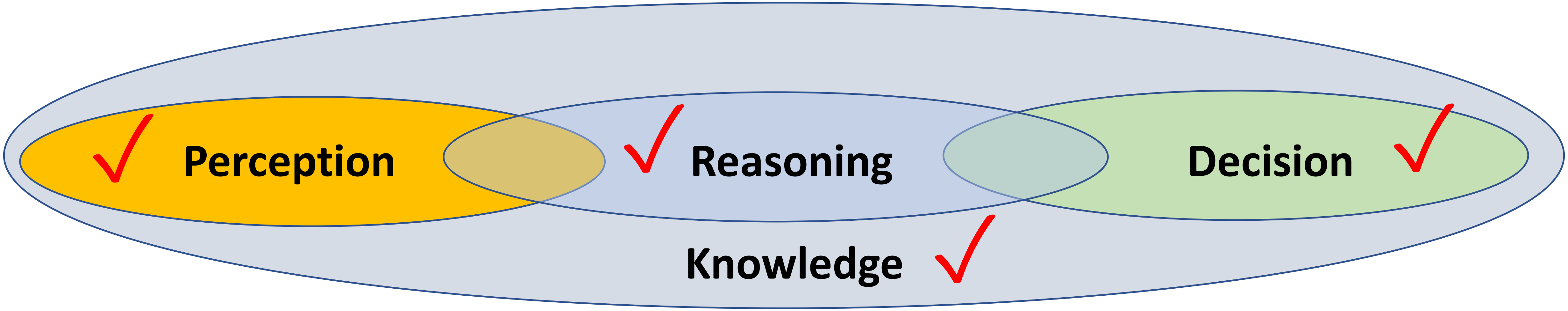
- **The neural net architecture is an implicit model but not a causal one**
- **Training such a model (optimizing parameters) requires astronomical amounts of data and computing power**
- **These systems are completely opaque, involve no reasoning and cannot be used for explanation or causality**

A dog has an elaborate neural net and can be trained to perform very useful tasks

What do we learn as a result?

Statistical ML is a great tool

Carnegie Mellon University
Language Technologies Institute



- Statistical ML is a great tool to refine the parameters of our models
- It is not a substitute for a model
- A neural net cannot explain its decisions which is not acceptable in most applications
- Try to give advice to a neural net – you cannot!
- The AI community is beginning to shift from purely statistical ML to more knowledge-based methods