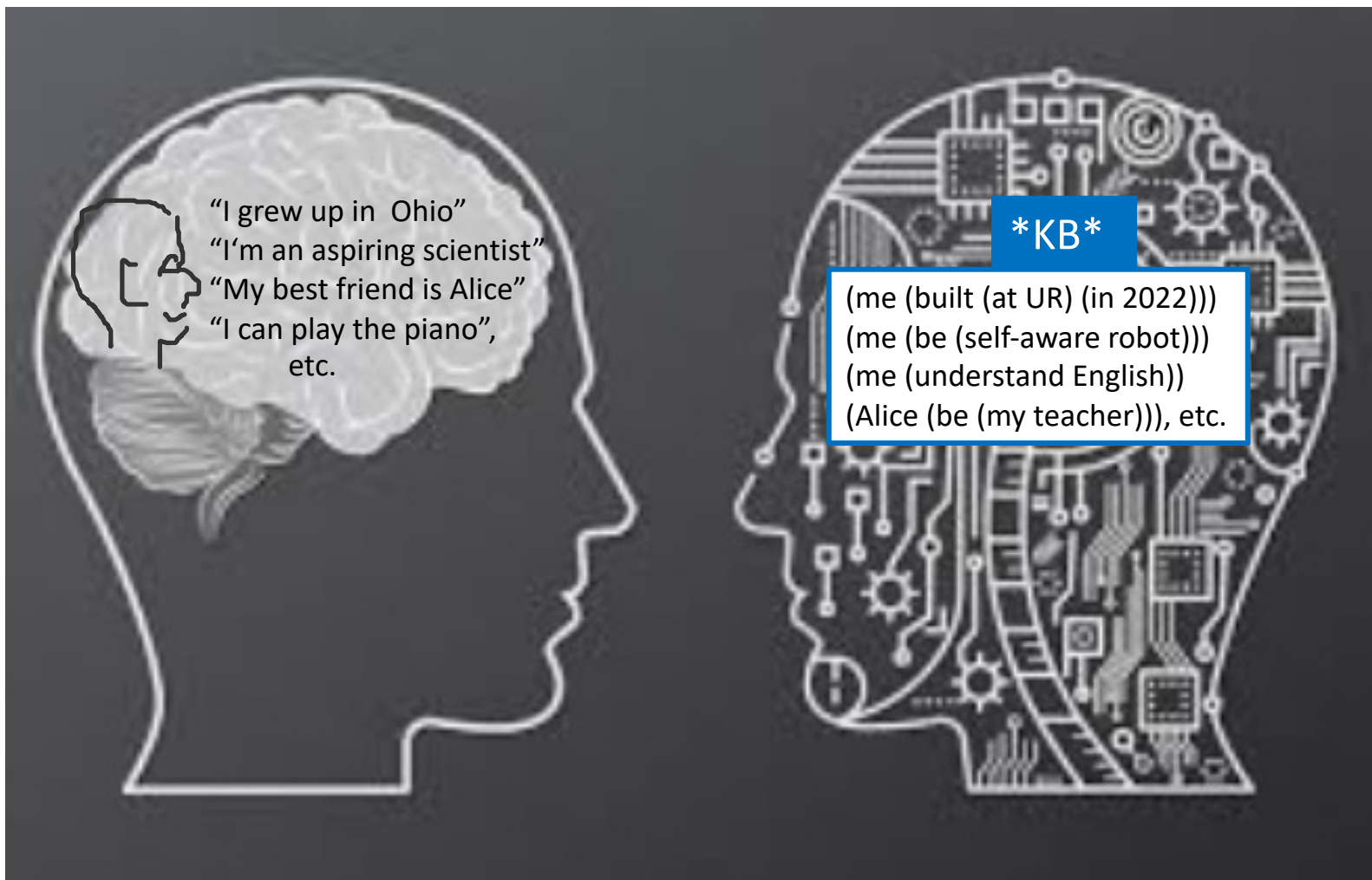


Self-Awareness through Access to Self-Knowledge



Self-model = knowledge about oneself that can be accessed, and can be used together With general (stored) knowledge to reason (& talk) about oneself in relation to the world.

But how can we represent such self-knowledge and general knowledge in a machine?

REPRESENTING KNOWLEDGE IN FIRST-ORDER LOGIC (FOL)

Much of the knowledge that we can informally express in ordinary language can be formally represented in FOL

But if our goal is to capture such "verbalizable" facts, why not use ordinary language (English, Mandarin, Sanskrit, ...) ??

"I SAW THE MOONS OF JUPITER
WITH A TELESCOPE LAST NIGHT"

- AMBIGUOUS
- INDEXICAL



FOL avoids ambiguity, indexicality* by

- using brackets
- using unambiguous symbols
- using formalized versions of and, or, not, implies
- using \exists , \forall , and variables to talk about existing, unspecified individuals, or about all individuals

* In its most general sense, "indexicality" refers to all the contextual factors that influence meaning, including background knowledge.

Consider:

- In the storm, a tree fell on our roof. [cf. Levesque's "winograd schema challenge"]
I hired a crew to ^{repair} ~~remove~~ it.

- Dialogue:

- "I'm studying computer science."
 - "I mostly watch action movies."
- What about you?

We need at least the expressiveness of FOL if we wish to

- refer to individuals :
Nadia, CSC244, Rome, x, ...
- express properties :
Nadia is single, Odie has rabies, ...
- express relationships : The Morning Star is Venus,
Romeo loved Juliet, M. Curie discovered radium,
the population of the world exceeds 7 billion, ...
- express alternatives : (V)
You got to school on foot, on a bicycle,
by bus, or by car.
- express negative facts : (¬)
There is no life on Mars
- express general facts : (A)
All poodles are dogs, All (living)
parents are older than their children

SYNTAX OF FOL

- see inductive definition

- Predication and equality

Dog(Fido)

Loves(Romeo, Juliet)

Morning-Star = Planet-Venus

- Logical connectives

\neg Likes(Cheney, Nader)

Loves(Romeo, Juliet) \wedge Loves(Juliet, Romeo)

At-home(Mary) \vee At-work(Mary)

Poodle(Fifi) \Rightarrow Dog(Fifi)

$(A > B) \Leftrightarrow (B < A)$

- Avoiding ambiguity

$(P(A) \vee \neg Q(A, B)) \wedge R(B, A)$

(officially, conjunctions, disjunctions, quantified sentences, & equalities are always bracketed)

- Functions (prefixed & infix) *

Hot (surface-of (Planet-Venus))

Larger-than (surface-of (Earth), surface-of (Venus))

sum (weight (John), weight (Mary)) = 300

weight (John) + weight (Mary) = 300

weight (John) > weight (Mary)

=, >
infix relations

- Quantification

$(\forall x \text{ Entity}(x))$

$(\forall x (\text{Poodle}(x) \Rightarrow \text{Dog}(x)))$

$(\exists x \text{ Robot}(x))$

$(\forall x (\text{Robot}(x) \Rightarrow (\exists y \text{ Built}(y, x))))$

$(\forall x (\text{Robot}(x) \Rightarrow \neg \text{Intelligent}(x)))$

$(\exists x (\text{Robot}(x) \wedge \text{Intelligent}(x)))$

$\text{Robot}(\text{Daneel}) \wedge (\forall x (\text{Robot}(x) \Rightarrow x = \text{Daneel}))$

* Functions are single-valued relations
and can be partial or total

Syntax of FOL, more formally

$\langle \text{individual constant} \rangle ::= A | B | C | \text{John} | \text{Block1} | \text{Block2} | \dots$

$\langle \text{variable} \rangle ::= x | y | z | x_1 | x_2 | \text{block1} | \dots$

$\langle \text{function constant} \rangle ::= a | b | c | \text{weight} | \text{sum} | \text{mother-of} | \dots$

$\langle \text{term} \rangle ::= \langle \text{individual constant} \rangle | \langle \text{variable} \rangle |$
 $\langle \text{function constant} \rangle (\langle \text{term} \rangle, \dots, \langle \text{term} \rangle)$

$\langle \text{predicate constant} \rangle ::= A | B | C | \text{Dog} | \text{Loves} | \text{Owes} | \dots$

$\langle \text{binary connective} \rangle ::= \wedge | \vee | \Rightarrow | \Leftarrow | \Leftrightarrow$

$\langle \text{formula} \rangle ::= \langle \text{predicate constant} \rangle (\langle \text{term} \rangle, \dots, \langle \text{term} \rangle) |$
 $\langle \text{term} \rangle = \langle \text{term} \rangle | \neg \langle \text{formula} \rangle |$
 $(\langle \text{formula} \rangle \langle \text{binary connective} \rangle \langle \text{formula} \rangle) |$
 $(\forall \langle \text{variable} \rangle \langle \text{formula} \rangle) | (\exists \langle \text{variable} \rangle \langle \text{formula} \rangle)$
 or $\forall \langle \text{var.} \rangle . \langle \text{form.} \rangle$ or $\exists \langle \text{var.} \rangle . \langle \text{form.} \rangle$

Notes:

$\langle \dots \rangle$ are metalinguistic symbols (not part of the language being defined)

function & predicate constants are sorted into 1-place, 2-place, ... and must be used consistently this way

individual constants also include 0, 1, 2, ...

function constants also include infix functions $+$ $-$ $*$ $/$ \uparrow \cup \cap

predicate constants also include infix predicates $<$ $>$ \leq \geq \in

(unlike $=$, these are freely interpretable ...)

REASONING

A robot's (Robbie's) KB:

$\{$ Robot(Robbie), Located-in(Robbie, Rochester, Curr-Time*),
 Dog(Snoopy), Person(Jack),
 $(\forall x) \text{ Person}(x) \Rightarrow \neg \text{Robot}(x)$,
 $(\forall x) \text{ Dog}(x) \Rightarrow \neg \text{Person}(x)$
 $(\forall x) \text{ Dog}(x) \Rightarrow \neg \text{Robot}(x) \}$

clause form
 $\neg \text{Person}(x) \vee \neg \text{Robot}(x)$

Question 1: Are you a person? $\text{Person}(\text{Robbie})?$

Reasoning: $\frac{\neg \text{Person}(x) \vee \neg \text{Robot}(x), \text{Robot}(\text{Robbie})}{\neg \text{Person}(\text{Robbie})}$, i.e., "No"

Question 2: Where are you?

$(\exists x) \text{ Located-in}(\text{Robbie}, x, \text{Curr-Time}^*)?$

Reasoning: Instantiate x to Rochester, i.e., "in Rochester"

Question 3: Is there a dog? $(\exists x) \text{ Dog}(x)?$

Reasoning: Instantiate x to Snoopy; i.e., "Yes, Snoopy"

In general: What can be naturally expressed in words
can be naturally expressed in logic

Types of inference from facts:

• Deduction:

- Mary has a poodle named Fifi
DOES MARY HAVE A DOG?
- Lulu is Fifi's mother (and is alive)
IS LULU YOUNGER OR OLDER THAN FIFI?
- Fifi ate a cookie
WHERE DID THE COOKIE GO?

} Involve
assimilated
general
knowledge

• Uncertain & nonmonotonic inference

- Given (only) that Tweety is a canary,
TWEETY PROBABLY FLIES
- Given that John is sniffing & has a sore throat
but no fever, he probably has a cold

} More gener-
ally, inference
based on
schema
recognition,
analogy

• Abduction

- Given that all the crows I've seen are black,
conjecture that ALL CROWS ARE BLACK

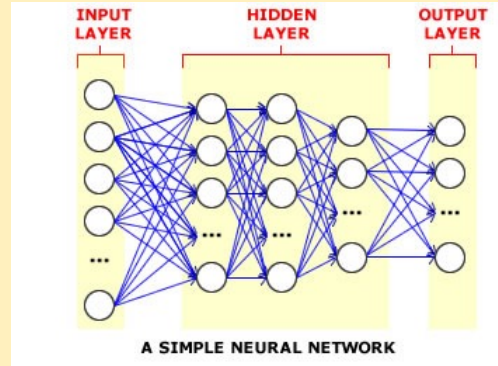
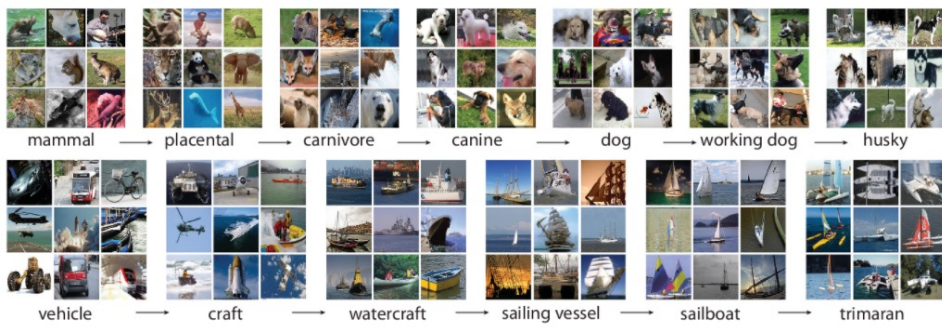
• Planning

- Given my goal to get to **Miami Nov. 2**,
I should book a flight, arrange for TA's
to sub for me, etc.

Advantages of Logic as KR

- versatility
- interpretability
- modularity

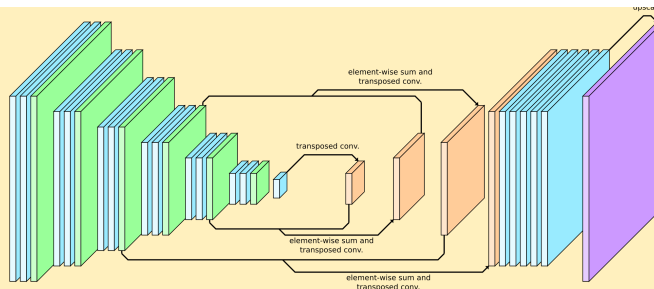
**Not all knowledge is symbolic ...
Some is *analogical* at the feature level**



e.g., new image of a dog → output “dog” (or “beagle”, etc.)

In effect, the NN makes an analogy between the training images & given new ones

A woman fell in love	⇒	The woman was happy;
A man had an accident	⇒	The man was unhappy;
A child received a present	⇒	The child was delighted;
A student was mugged	⇒	The student was furious;
A man tripped and fell	⇒	The man was embarrassed;



Many thousands of examples

e.g., a student graduated → The student was proud

In effect, the NN makes an analogy between the [premise ⇒ conclusion] examples & the conclusion that corresponds to a new given premise (also “learning English”)

Not all knowledge is symbolic ...

Some uses analogical representations

- **Imagistic knowledge** (3-D visualization) [Miller, Kosslyn]

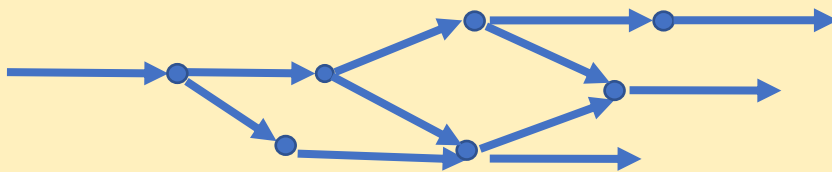


Consider the following situations:

- “Walking alongside the lioness, Joy Adamson stroked its head”.
- “Walking alongside the giraffe, Joy Adamson stroked its head”

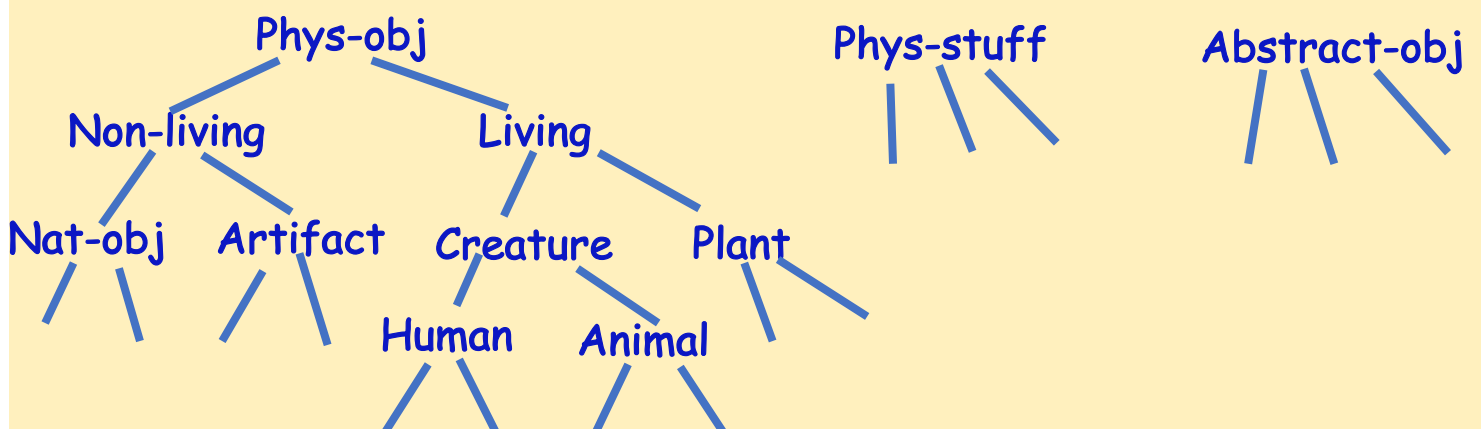
For a computer, use graphics rather than symbols

- **Temporal knowledge** (before/after, when, duration)



The arrow sequences are analogues of event sequences

- **Taxonomic knowledge** (type hierarchies)



What about "how to" knowledge?

- Many skills are hard to express in words, & seem too low-level to be expressed declaratively (it seems better to use mechanisms/procedures to implement or model them):
 - how the tune for "Twinkle, twinkle" goes
 - how to ride a bike, catch a ball, use chopsticks, tie shoelaces, ...
 - how to interpret retinal images
 - how to speak grammatically, ...
 - how to learn
- Procedural knowledge can be just as transparent and "logical" as factual knowledge
 - recipes, assembly instructions, emergency procedures, etc.
 - dynamic logic
 - procedures written at a high level, with all symbols meaningful in the domain (no "extraneous" symbols)