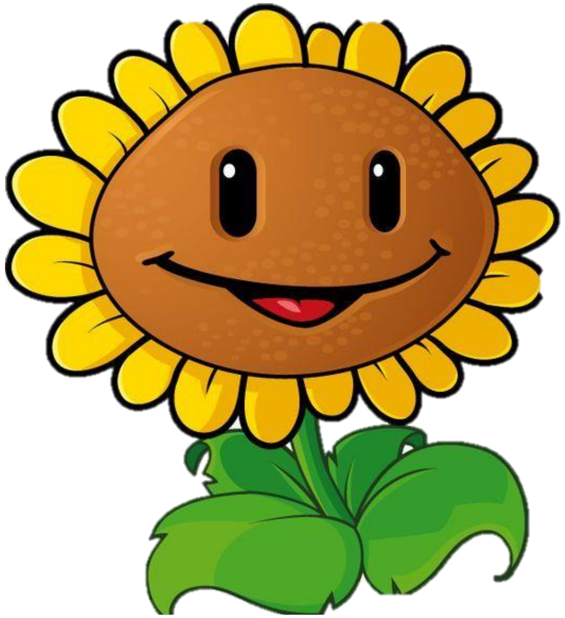
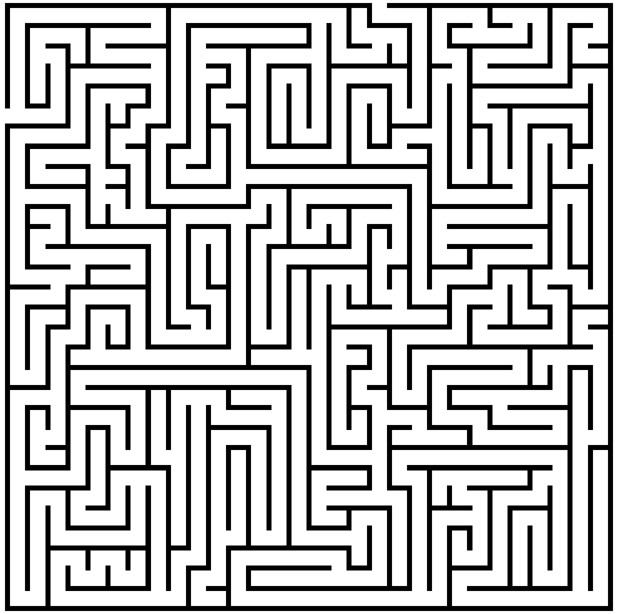
Knowledge Representation and Reasoning

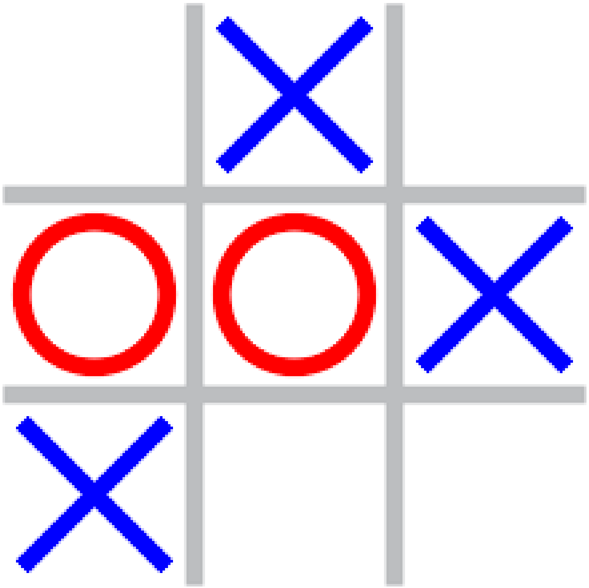


Problems Solved So Far

Process:

1. Implement Abstract Base Class (ABC)
2. Implement game class that extends ABC
3. Implement algorithm that solves all instances of ABC, possibly a heuristic that works for that game





Representations

Our representations store all the necessary state information and rules of the game

It is not particularly hard to represent **these** rules

Knowledge Representation and Reasoning

Human knowledge is complicated!

If it is not cloudy and it is daytime, the sky will be blue

Everyone comes to know such a fact. How?

We know if it is nighttime, the sky will not be blue

How and where is this fact stored?

We know the sky at noon on a non-cloudy day should not be pink!

If we looked up at the sky and saw it was not blue, we would be able to infer it was cloudy or nighttime outside

Knowledge Representation and Reasoning

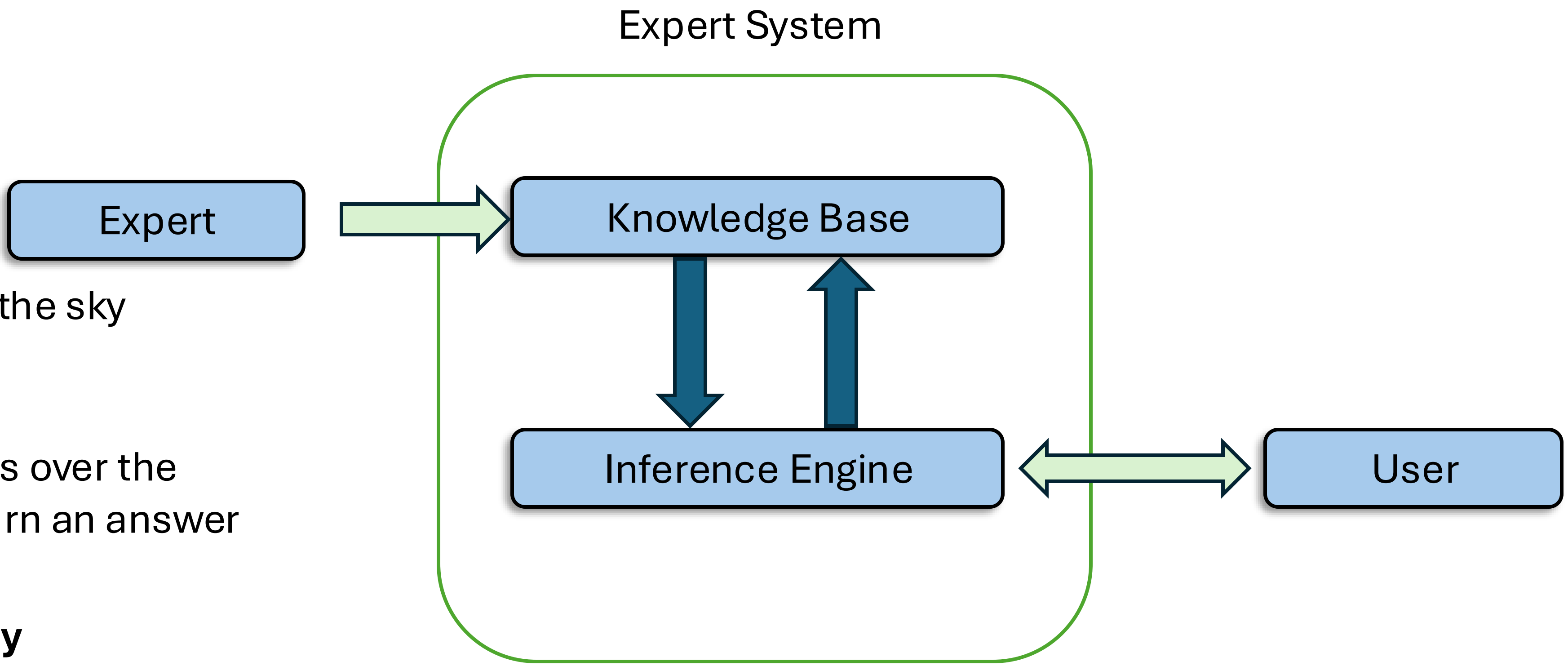
KRR methods seek to store **knowledge** and **reason** about that stored knowledge

Knowledge Base stores facts and rules about the world

Users can ask queries (e.g., the sky is cloudy, is the sky blue?)

The inference engine reasons over the knowledge and query to return an answer

Expert Systems played a **very** important role in the history of AI



How Can We Represent Knowledge?

One possible option: First-Order Logic

Objects: individual entities in the world

Predicates: properties of objects or relationships between objects

Functions: Mapping from object(s) to object(s)

Quantifiers: Universal ($\forall$) and existential ($\exists$) quantification over objects

Logical Connectors: AND ($\wedge$), OR ($\vee$), NOT ($\neg$), IMPLIES ($\Rightarrow$), etc.

Propositional Logic does not have predicates, functions, or quantifiers

How Can We Represent Knowledge?

$$\left(\neg isCloudy(sky) \land isDay(\mathrm{time}) \rightarrow isBlue(sky)\right)$$

Predicate (acts on object, is T or F)

Object

This expresses a **rule** using predicates and objects

A User may query this system with facts: “The sky is not cloudy, the sky is not blue”, the inference engine will determine what other facts can be determined.

Inference Rules in FOL

Universal Instantiation:

Replace Universally quantized variables with specific variable

All humans are mortal. I am a human. Therefore, I am mortal

Modus Ponens:

If we know P (is true), and $\mathrm{P} \Longrightarrow \mathrm{Q}$, we can conclude Q

Modus Tollens:

If we know not Q, and $\mathrm{P} \Longrightarrow \mathrm{Q}$, we can conclude not P

Inference Algorithms

Forward Chaining:

From initial set of facts:

Apply Modus Ponens/Tollens where possible

Add new facts to knowledge base

Repeat

Backward Chaining:

Start from Goal Fact and work backwards

If our goal is to prove that Q is True, and we have the rule $\mathrm{P} \Rightarrow \mathrm{Q}$ and can prove P is true, then we will prove Q is true…

Predicates

- $\text{Book}(x)$ - "x is a book"
- $\text{Author}(x)$ - "x is an author"
- $\text{User}(x)$ - "x is a user"
- $\text{Genre}(x)$ - "x is a genre"
- $\text{WroteBook}(a, b)$ - "author a wrote book b"
- $\text{HasGenre}(b, g)$ - "book b belongs to genre g"
- $\text{CheckedOut}(u, b)$ - "user u has checked out book b"
- $\text{Recommends}(u_1, b, u_2)$ - "user u_1 recommends book b to u
- $\text{Influences}(a_1, a_2)$ - "author a_1 influences author a_2"
- $\text{PopularGenre}(g)$ - "genre g is popular"
- $\text{Bestseller}(b)$ - "book b is a bestseller"

Rules

$$\forall a, b\,(\text{WroteBook}(a, b) \Rightarrow \text{Author}(a) \wedge \text{Book}(b))$$
$$\forall u, b\,(\text{CheckedOut}(u, b) \Rightarrow \text{User}(u) \wedge \text{Book}(b))$$
$$\forall b, g\,(\text{HasGenre}(b, g) \wedge \text{PopularGenre}(g) \Rightarrow \text{Bestseller}(b))$$
$$\forall a_1, a_2, b\,(\text{Influences}(a_1, a_2) \wedge \text{WroteBook}(a_1, b) \wedge \text{Bestseller}(b) \Rightarrow \exists b_2\,(\text{WroteBook}(a_2, b_2) \wedge \text{Bestseller}(b_2)))$$

Facts

$$\text{Author}(\text{Asimov}) \wedge \text{Author}(\text{Clarke}) \wedge \text{Author}(\text{Bradbury})$$
$$\text{User}(\text{Alice}) \wedge \text{User}(\text{Bob}) \wedge \text{User}(\text{Carol})$$
$$\text{Book}(\text{Foundation}) \wedge \text{Book}(\text{Childhood'sEnd}) \wedge \text{Book}(\text{Fahrenheit451})$$
$$\text{Genre}(\text{SciFi}) \wedge \text{Genre}(\text{Mystery}) \wedge \text{Genre}(\text{Fantasy})$$
$$\text{WroteBook}(\text{Asimov}, \text{Foundation})$$
$$\text{WroteBook}(\text{Clarke}, \text{Childhood'sEnd})$$
$$\text{WroteBook}(\text{Bradbury}, \text{Fahrenheit451})$$
$$\text{HasGenre}(\text{Foundation}, \text{SciFi})$$
$$\text{HasGenre}(\text{Childhood'sEnd}, \text{SciFi})$$
$$\text{HasGenre}(\text{Fahrenheit451}, \text{SciFi})$$
$$\text{CheckedOut}(\text{Alice}, \text{Foundation})$$
$$\text{CheckedOut}(\text{Bob}, \text{Childhood'sEnd})$$
$$\text{Influences}(\text{Asimov}, \text{Clarke})$$
$$\text{Bestseller}(\text{Foundation})$$
$$\text{PopularGenre}(\text{SciFi})$$

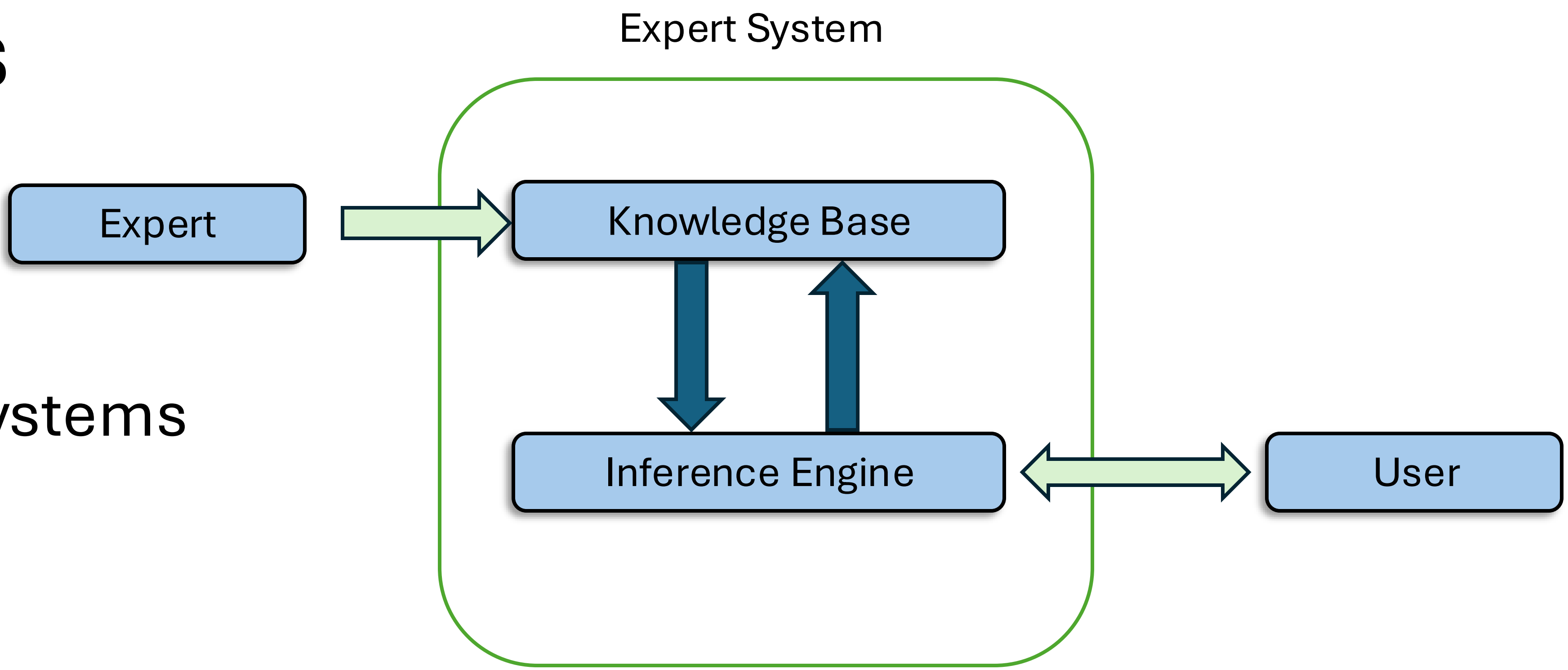
Will Arthur Clarke write a Best Seller?

1. By universal instantiation with b = Childhood'sEnd, g = SciFi:
HasGenre(Childhood'sEnd, SciFi) ∧ PopularGenre(SciFi) ⇒ Bestseller(Childhood'sEnd)

2. We know HasGenre(Childhood's End, SciFi), and PopularGencre(SciFi) from Facts

3. By modus ponens: Bestseller(Childhood'sEnd)

Expert Systems



Knowledge in Expert Systems comes from experts!

Example: MYCIN

Identify the bacteria causing serious infections

```
(defrule 52
 if (site culture is blood)
  (gram organism is neg)
  (morphology organism is rod)
  (burn patient is serious)
 then .4
  (identity organism is pseudomonas))
Rule 52:
 If
  1) THE SITE OF THE CULTURE IS BLOOD
  2) THE GRAM OF THE ORGANISM IS NEG
  3) THE MORPHOLOGY OF THE ORGANISM IS ROD
  4) THE BURN OF THE PATIENT IS SERIOUS
 Then there is weakly suggestive evidence (0.4) that
  1) THE IDENTITY OF THE ORGANISM IS PSEUDOMONAS
```

Medical Experts write rules and knowledge facts by hand into the expert system

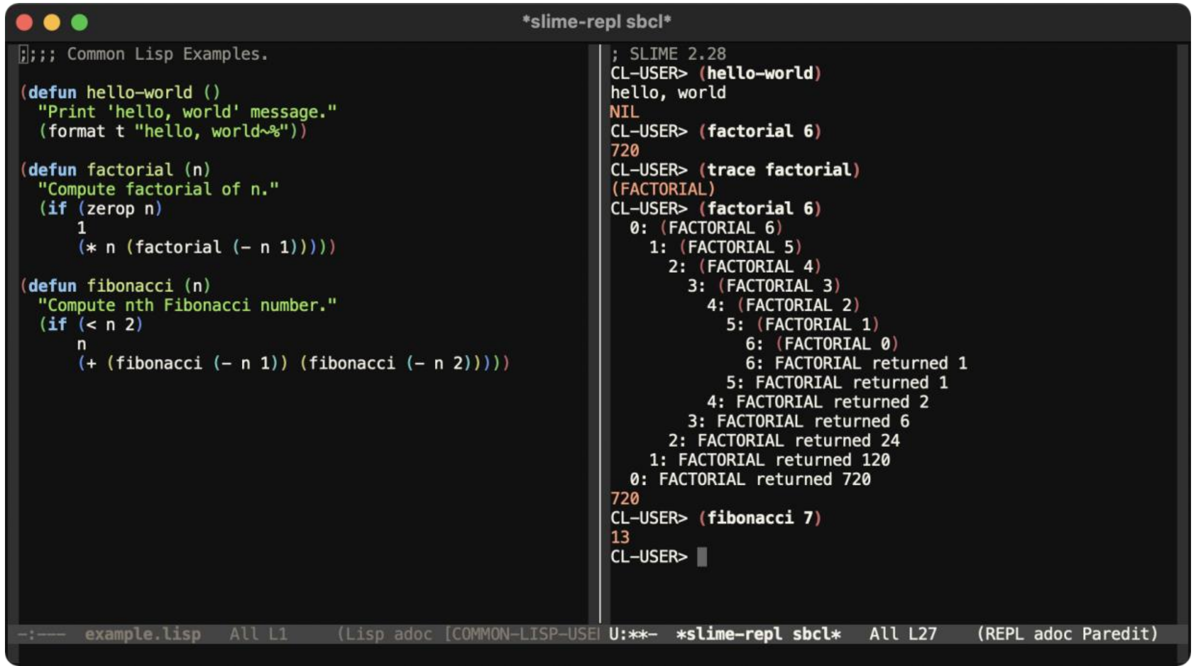
Lisp

Lisp was created by John McCarthy, who also coined the term AI and helped found the field

Lisp is a functional programming language and meant to process symbols and perform symbolic reasoning

Lisp and early work in AI were tightly tied together

```
(defrule 52
 if (site culture is blood)
  (gram organism is neg)
  (morphology organism is rod)
  (burn patient is serious)
 then .4
  (identity organism is pseudomonas))
Rule 52:
 If
  1) THE SITE OF THE CULTURE IS BLOOD
  2) THE GRAM OF THE ORGANISM IS NEG
  3) THE MORPHOLOGY OF THE ORGANISM IS ROD
  4) THE BURN OF THE PATIENT IS SERIOUS
 Then there is weakly suggestive evidence (0.4) that
  1) THE IDENTITY OF THE ORGANISM IS PSEUDOMONAS
```



Lisp Machines

Demand for expert systems was high in industry in the 1970s and 1980s

Custom hardware (Lisp machines) were developed to run expert systems

(Introduced computer mice, GUIs, and ethernet connectivity)

Xerox Daybreak

For more context on Lisp Machines:
A Cult AI Computer's Boom and Bust: https://www.youtube.com/watch?v=sV7C6Ezl35A


A HIGH-TECH MARKET THAT'S NOT FEELING THE PINCH

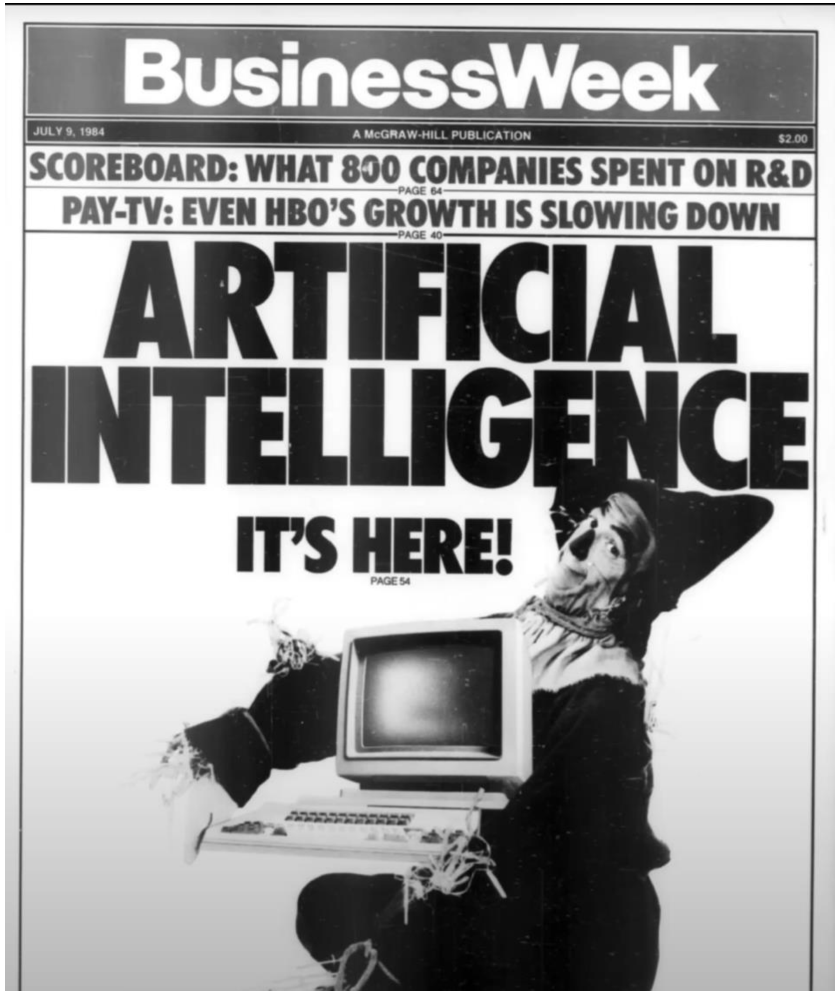
EAGER INVESTORS HAVE CREATED A BOOM IN ARTIFICIAL INTELLIGENCE. BUT CAN ALL THOSE STARTUPS DELIVER?

What's the least likely way to attract investors these days?

Data Inc., a market researcher in Scottsdale, Ariz.

By all rights, AI should be an

Source: Business Week 1985 Vol 1



"It's ironic, three years ago AI was considered flaky. Now it's hot and everyone wants it." Randall Davis. Business Week. July 1984

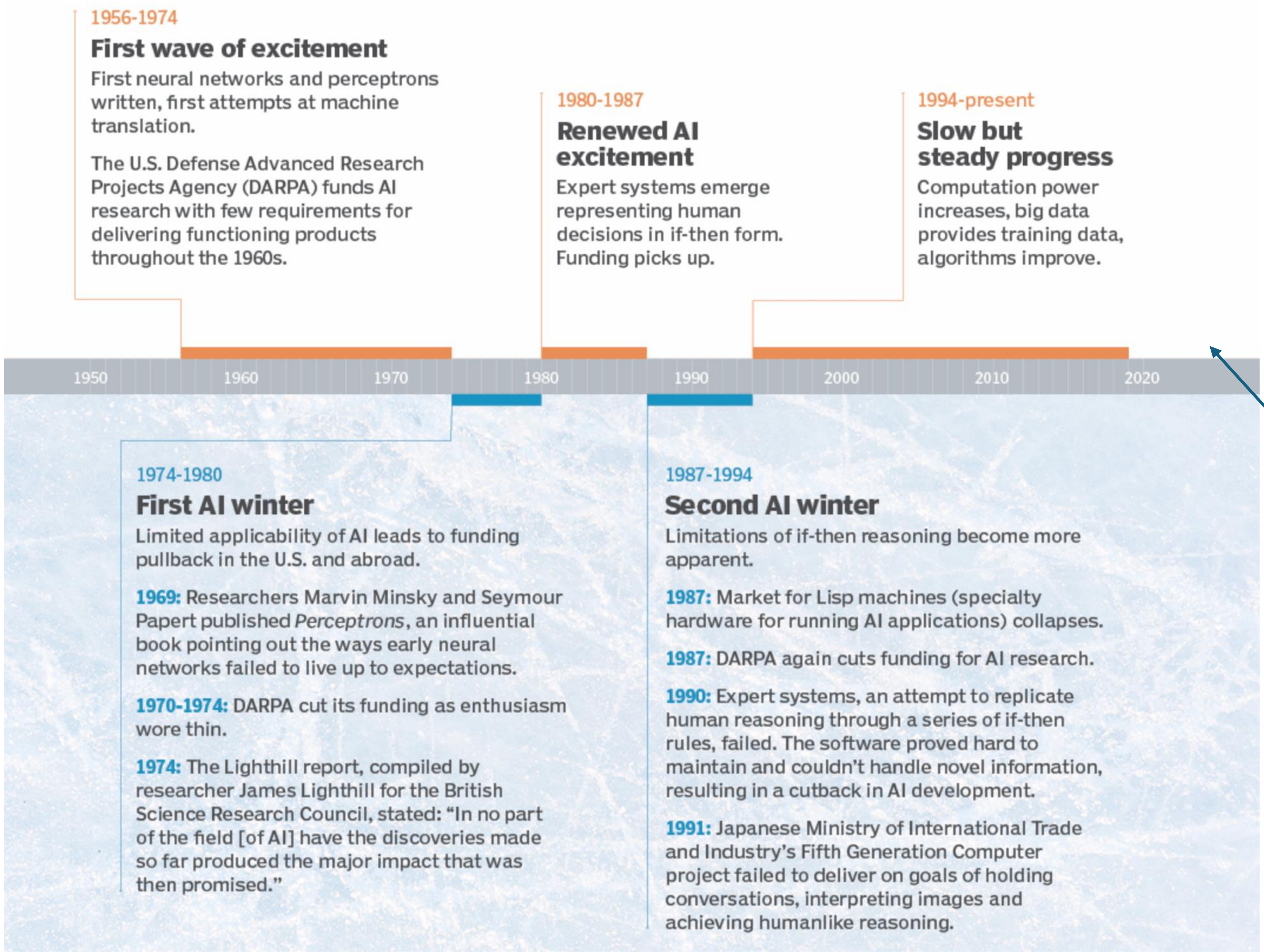
Limitations of Expert Systems

- The information bottleneck: Building expert systems was difficult and expensive. Required programmers to work with experts to translate knowledge into if-then rules
- Inability to handle novel situations: What happens when you encounter new situations that your knowledge base was unprepared for?
- Real world knowledge is “fuzzy” and handling uncertainty in expert systems is hard

The rise of Personal Computers

General Purpose machines (i.e., not just for running Lisp)

Faster than Lisp machines at running Lisp…



Generative AI Boom/Bubble

Image Source: https://www.techtarget.com/searchenterpriseai/definition/AI-winter

LLMs and KRR

LLMs appear to solve many of the problems at the heart of KRR

- Store Facts
- Reason About Facts
- Answer User Queries

AI on Trial: Legal Models Hallucinate in 1 out of 6 (or More) Benchmarking Queries

Magesh et al. *AI on Trial: Legal Models Hallucinate in 1 out of 6 (or More) Benchmarking Queries*, STAN. U.: HUM.-CENTERED A.I. (May 23, 2024),

LLMs and KRR

- KRR is perhaps the most important remaining challenge for LLMs
- Language models don't store facts or reason over them explicitly (which is why they scale well)
- But… It also causes hallucinations in LLMs