Logic

Introduction to Artificial Intelligence

CS/ECE 348

Lecture 11

September 27, 2001

Outline

Last Lecture

- Games Cont.
  - α-β pruning
  - Games with chance, e.g. Backgammon
- Logical Agents and theWumpus World

This Lecture

Introduction to Logic

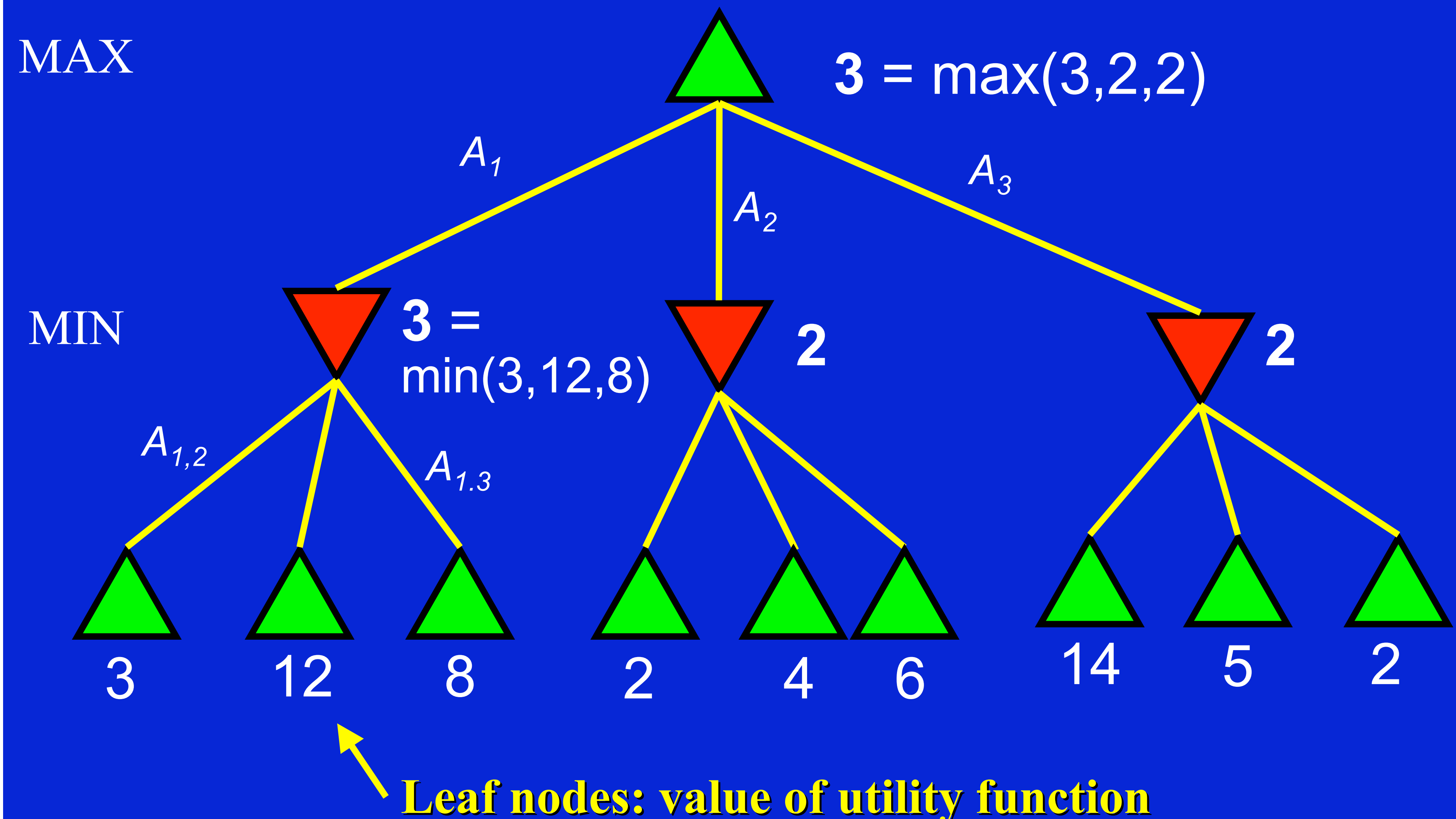
- **MP1 Assigned**

Reading

- Chapter 6

Minimax Example

Game tree for a fictitious 2-ply game



Resource limits

Suppose we have 100 seconds, explore 10^4 nodes/second

→ 10^6 nodes per move

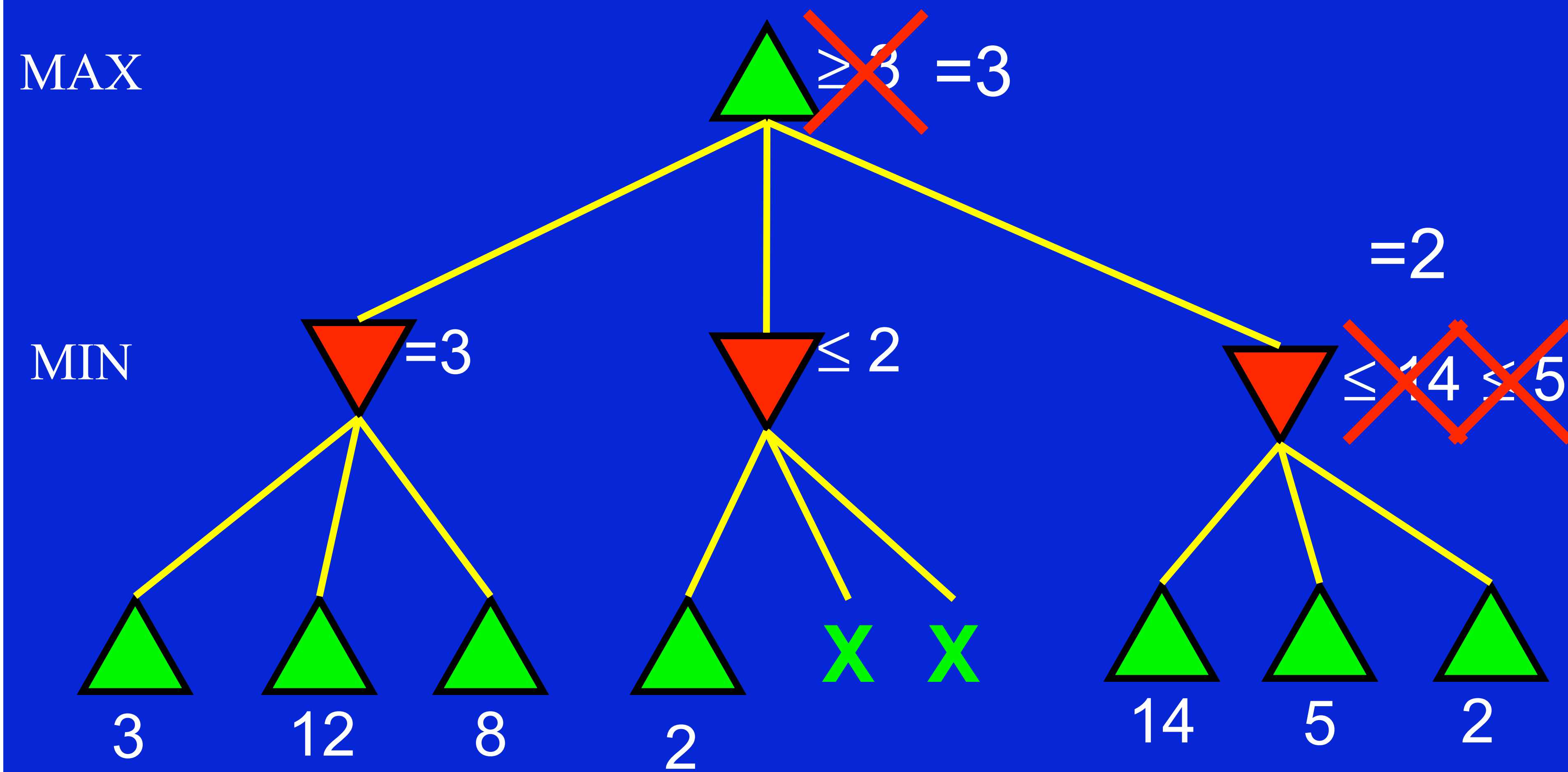
Standard approach:

- *cutoff test*

  e.g., depth limit
- *evaluation function* = estimated desirability of position

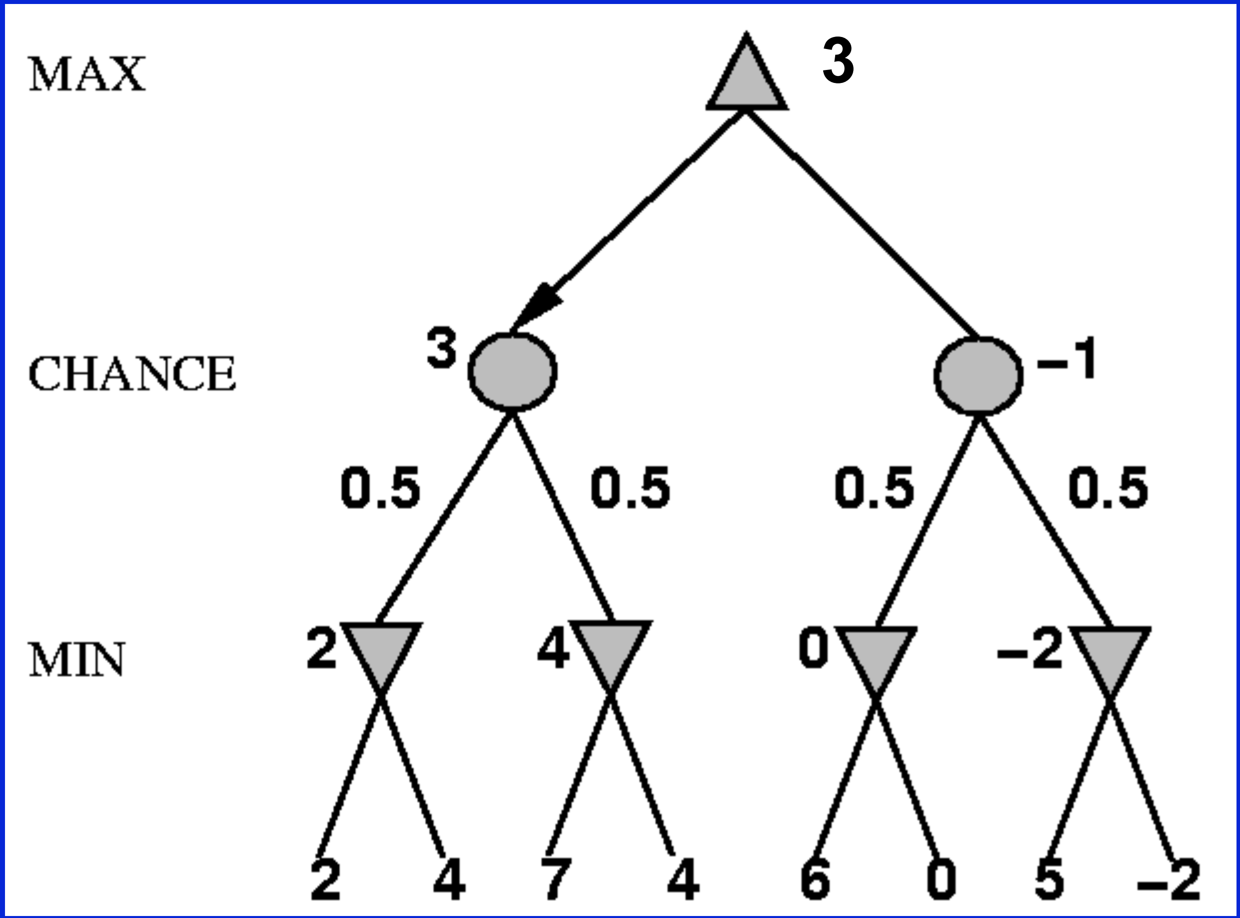
  (an approximation to the utility used in minimax).

α–β Pruning Example

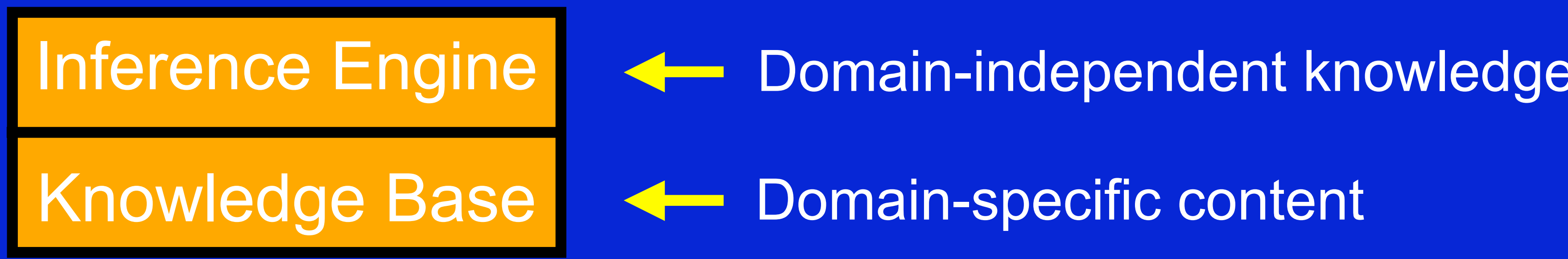


Nondeterministic games

1. For mind node, compute min of children
2. For chance node, compute weighted average of children
3. For max node, compute max of children



Knowledge bases



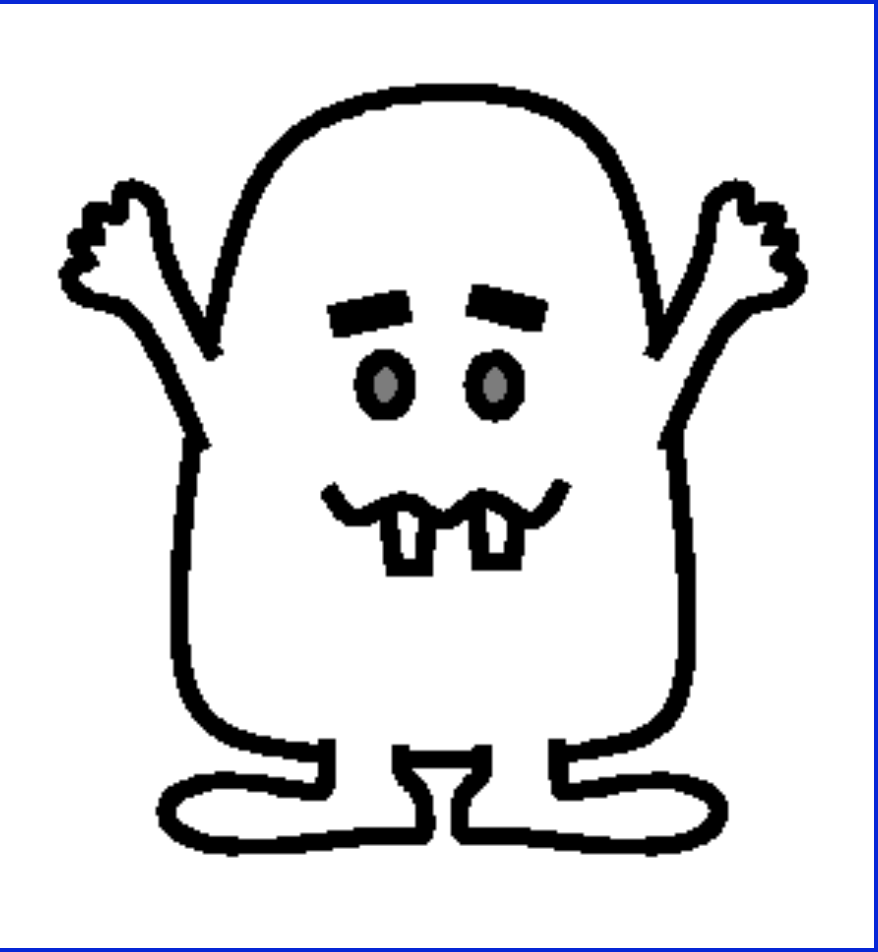
- **Knowledge base** = set of **sentences** in a **formal language**
- **Declarative approach** to building an agent or other system:
  - Tell it what it needs to know
  - Alternative is learning approach, but we'll do this later
- Answers should follow from the KB
- Agents or systems can be viewed at the **knowledge level**

  i.e., what they know, regardless of how implemented
- Or at the **implementation level**

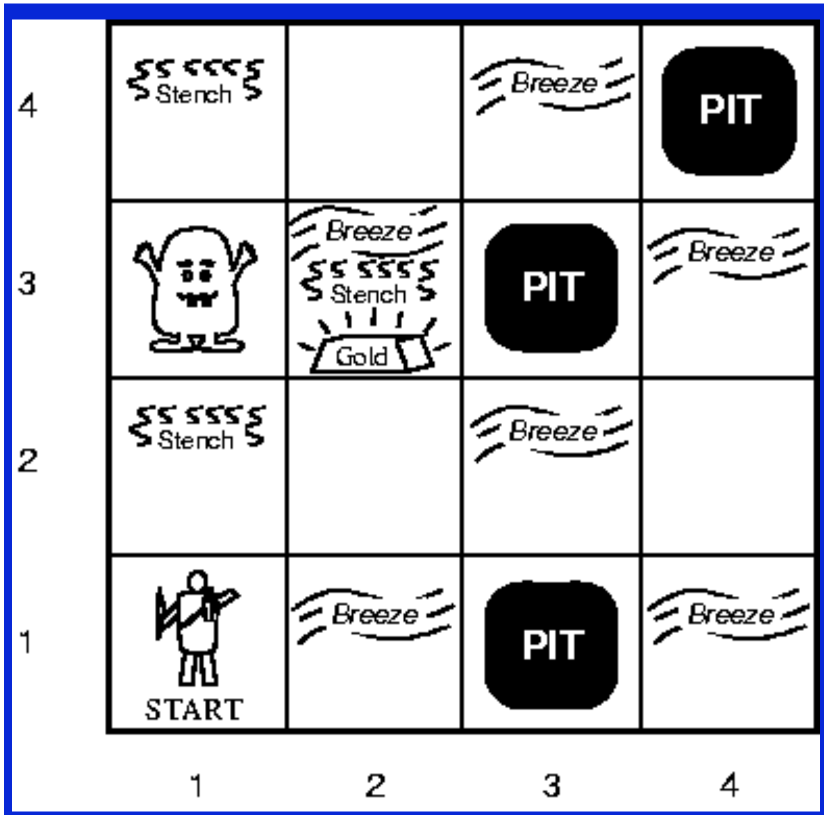
  i.e., data structures in KB and algorithms that manipulate them

Introducing

THE WUMPUS



The Wumpus World



- **Goals:** Get gold back to the start without entering it or wumpus square
- **Percepts:** Breeze, Glitter, Smell
- **Actions:** Left turn, Right turn, Forward, Grab, Release, Shoot

Environment

- Squares adjacent to wumpus are smelly
- Squares adjacent to pit are breezy
- Glitter if and only if gold is in the same square
- Shooting kills the wumpus if you are facing it
- Shooting uses up the only arrow
- Grabbing picks up the gold if in the same square
- Releasing drops the gold in the same square

Logic in general

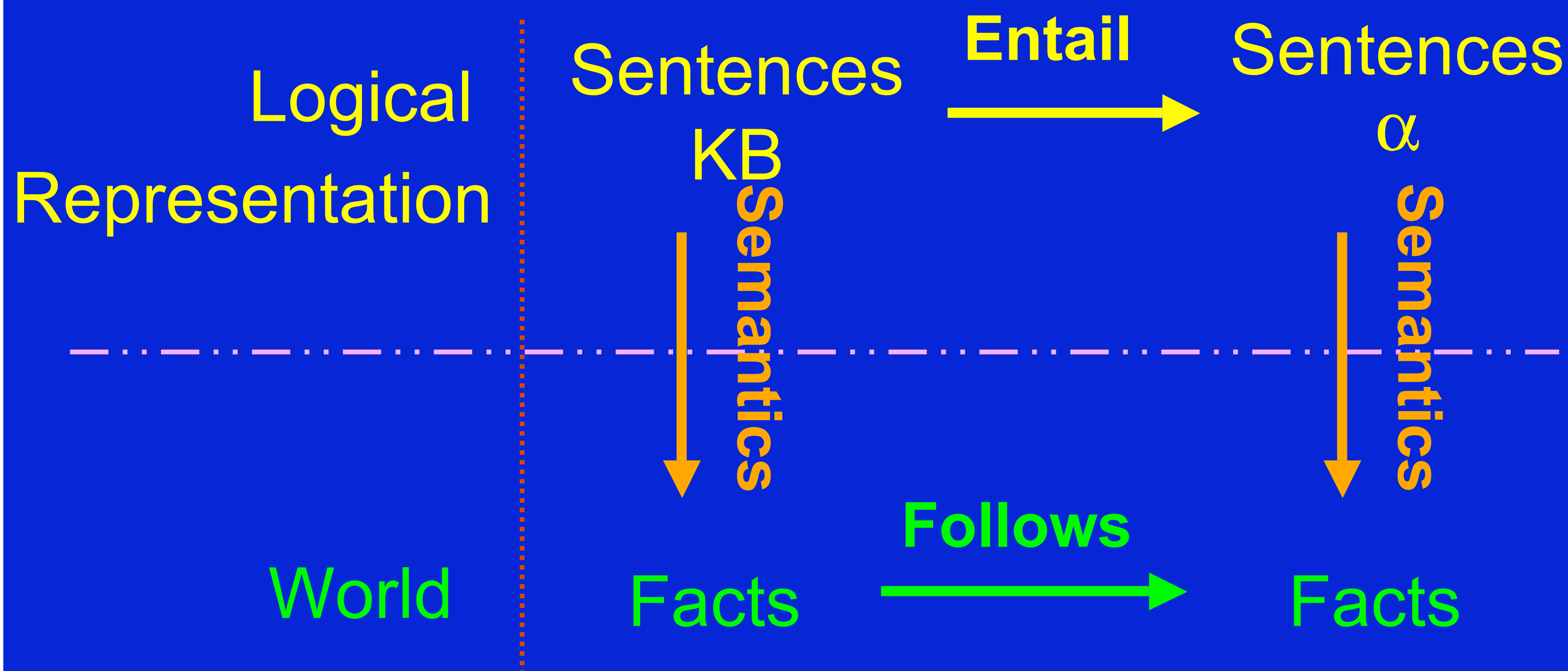
- **Logics** are formal languages for representing information such that conclusions can be drawn
- **Syntax** defines the sentences in the language
- **Semantics** define the ``meaning'' of sentences;
  i.e., define **truth of a sentence** in a world
- E.g., the language of arithmetic
  - $x + 2 \geq y$ is a sentence
  - $x + y >$ is not a sentence
  - $x + 2 \geq y$ is true iff the number $x+2$ is no less than the number y
  - $x + 2 \geq y$ is true in a world where $x = 7, y = 1$
  - $x + 2 \geq y$ is false in a world where $x = 0, y = 6$

Types of logic

- Logics are characterized by what they commit to as ``primitives''
- Ontological commitment: what exists - facts? objects? time? beliefs?
- Epistemological commitment: what states of knowledge?

Language	Ontological Commitment	Epistemological Commitment
Propositional Logic	Facts	True/false/unknown
First order logic	Facts, objects, relations	True/false/unknown
Temporal logic	Facts, objects, relations, time	True/false/unknown
Probability theory	Facts	Degree of belief 0…1
Fuzzy logic	Degree of truth	Degree of belief 0…1

Entailment



- A knowledge base is a collection of sentences
- An entailed sentence is true, given that the old sentences in the knowledge base are true.

Inference

- A knowledge base is a collection of sentences
- Knowledge base KB entails sentence α if and only if
  α is true whenever KB is true (i.e. all sentences in KB are true)

- An inference procedure $\vdash_i$ can do two things:
  – Given KB, generate new sentence α purported to be entailed by KB
  – Given KB and α, report whether or not α is entailed by KB.

- A sound inference procedure only produces entailed sentences
- Every entailed sentence can be generated by a complete inference procedure

Propositional logic: Syntax

- Propositional logic is the simplest logic

- Logical constants **TRUE** and **FALSE** are sentences
- Proposition symbols P1, P2 etc are sentences

- Symbols P1 and negated symbols $\neg$ P1 are called literals

- If S is a sentence, $\neg$ S is a sentence (NOT)
- If S1 and S2 is a sentence, S1 $\wedge$ S2 is a sentence (AND)
- If S1 and S2 is a sentence, S1 $\vee$ S2 is a sentence (OR)
- If S1 and S2 is a sentence, S1 $\Rightarrow$ S2 is a sentence (Implies)
- If S1 and S2 is a sentence, S1 $\Leftrightarrow$ S2 is a sentence (Equivalent)

Order of Precedence

From highest to lowest:

parenthesis (Sentence)

NOT $\neg$

AND $\wedge$

OR $\vee$

Implies $\Rightarrow$

Equivalent $\Leftrightarrow$

Propositional logic: Semantics

Truth Table

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

Implication: $P \Rightarrow Q$

"If P is true, then Q is true; otherwise I'm making no claims about the truth of Q."

$P \Rightarrow Q$ is equivalent to $\neg P \vee Q$

Propositional Logic: Proof methods

- **Model checking**
  - Truth table enumeration (sound and complete for propositional logic)
- **Application of inference rules**
  - Legitimate (sound) generation of new sentences from old
  - Proof = a sequence of inference rule applications

    Can use inference rules as operators in a standard search algorithm

Propositional Inference: Enumeration Method

- Let **α=A ∨ B** and **KB = (A ∨ C) ∧ (B ∨ ¬C)**
- Is it the case that KB ⊨ α ?
  - I.e., α is true in all worlds where KB is true
- Check all possible models -- α must be true whenever KB is true

A	B	C	**KB** **(A ∨ C) ∧** **(B ∨ ¬C)**	**α** **A ∨ B**
False	False	False		
False	False	True		
False	True	False		
False	True	True		
True	False	False		
True	False	True		
True	True	False		
True	True	True		

Propositional Inference: Enumeration Method

- Let **α=A $\vee$ B** and **KB = (A $\vee$ C) $\wedge$ (B $\vee$ $\neg$C)**
- Is it the case that KB $\models$ α ?
  - I.e., α is true in all worlds where KB is true
- Check all possible models -- α must be true whenever KB is true

A	B	C	**KB (A $\vee$ C) $\wedge$ (B $\vee$ $\neg$C)**	**α A $\vee$ B**
False	False	False	False	
False	False	True	False	
False	True	False	False	
False	True	True	True	
True	False	False	True	
True	False	True	False	
True	True	False	True	
True	True	True	True	

Propositional Inference: Enumeration Method

- Let **α=A ∨ B** and **KB = (A ∨ C) ∧ (B ∨ ¬C)**
- Is it the case that KB ⊨ α ?
- Check all possible models -- α must be true whenever KB is true

A	B	C	**KB** **(A ∨ C) ∧ (B ∨ ¬C)**	**α** **A ∨ B**
False	False	False	False	False
False	False	True	False	False
False	True	False	False	True
False	True	True	True	True
True	False	False	True	True
True	False	True	False	True
True	True	False	True	True
True	True	True	True	True

Propositional Inference: Enumeration Method

- Let **α=A ∨ B** and **KB = (A ∨ C) ∧ (B ∨ ¬C)**
- Is it the case that KB ⊨ α ?
- Check all possible models -- α must be true whenever KB is true

A	B	C	**KB** **(A ∨ C) ∧ (B ∨ ¬C)**	**α** **A ∨ B**
False	False	False	False	False
False	False	True	False	False
False	True	False	False	True
False	True	True	True	True
True	False	False	True	True
True	False	True	False	True
True	True	False	True	True
True	True	True	True	True

Propositional Inference: Enumeration Method

- Let **α=A ∨ B** and **KB = (A ∨ C) ∧ (B ∨ ¬C)**
- Is it the case that KB $\models$ α ?
- Check all possible models -- α must be true whenever KB is true

A	B	C	**KB** **(A ∨ C) ∧ (B ∨ ¬C)**	**α** **A ∨ B**
False	False	False	False	False
False	False	True	False	False
False	True	False	False	True
False	[illegible]	[illegible]	[illegible]	True
True	[illegible]	[illegible]	[illegible]	True
True	[illegible]	[illegible]	[illegible]	True
True	True	False	True	True
True	True	True	True	True

KB $\models$ α

Proof methods

- **Model checking**
  - Truth table enumeration (sound and complete for propositional logic)
- **Application of inference rules**
  - Legitimate (sound) generation of new sentences from old
  - Proof = a sequence of inference rule applications

Can use inference rules as operators in a standard search algorithm

Normal forms

Approaches to inference use syntactic operations on sentences, often expressed in standardized forms

- Conjunctive Normal Form (CNF – universal)

    Conjunction of disjunctions of literals

    e.g. $(A \vee B) \wedge (B \vee \neg E \vee \neg H)$

- Disjunctive Normal Form (DNF – universal)

    Disjunction of conjunction of literals

    e.g. $A \vee (A \wedge B) \vee (H \wedge \neg M \wedge N \wedge A) \vee (A \wedge \neg B \wedge H)$

- Horn Form (restricted)

    Conjunction of Horn clauses (clauses w/ ≤ 1 positive literal)

    e.g. $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

    Often written as a set of implications

    $B \Rightarrow A$ and $(C \wedge D) \Rightarrow B$

Inference rules

An inference rule is sound if the conclusion is true in all cases where the premises are true

α	Premise
β	Conclusion

$$\frac{\alpha}{\beta}$$

An Inference Rule: Modus Ponens

- From an implication and the premise of the implication, you can infer the conclusion

$$\frac{\alpha \Rightarrow \beta, \ \alpha}{\beta} \qquad \begin{array}{l}\text{Premise} \\ \text{Conclusion}\end{array}$$

- An inference rule is sound if the conclusion is true in all cases where the premises are true

An Inference Rule: And - Elimination

- From a conjunction, you can infer any of conjuncts

$\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n$	Premise
α_i	Conclusion

- An inference rule is sound if the conclusion is true in all cases where the premises are true

And-Introduction & Double Negation

- **And-Introduction**

$\alpha_1, \alpha_2, \ldots, \alpha_n$	Premise
$\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n$	Conclusion

- **Double Negation**

$\neg\neg\alpha$	Premise
α	Conclusion