NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

CZ3005 Artificial Intelligence

Week 9b – Propositional Logic

Yu Han

han.yu@ntu.edu.sg

Nanyang Assistant Professor
School of Computer Science and Engineering
Nanyang Technological University

Recap

Sentence	$\rightarrow$	AtomicSentence | ComplexSentence
AtomicSentence	$\rightarrow$	LogicalConstant | PropositionalSymbol
ComplexSentence	$\rightarrow$	(Sentence) | Sentence LogicalConnective Sentence | ¬Sentence
LogicalConstant	$\rightarrow$	TRUE | FALSE
PropositionalSymbol	$\rightarrow$	P | Q | R | …
LogicalConnective	$\rightarrow$	Λ | $\vee$ | $\Leftrightarrow$ | $\Rightarrow$ | $\neg$

Precedence (from highest to lowest): $\neg$, Λ, $\vee$, $\Rightarrow$, $\Leftrightarrow$

e.g.: $\neg P \Lambda Q \vee R \Rightarrow S$ (not ambiguous), equal to: $(((\neg P) \Lambda Q) \vee R) \Rightarrow S$

Recap

A	B	C	A ∧ B	$B \Rightarrow C$	$(A \wedge B) \Rightarrow C$	$A \wedge (B \Rightarrow C)$
T	T	T				
T	T	F				
T	F	T				
T	F	F				
F	T	T				
F	T	F				
F	F	T				
F	F	F				

Recap

A	B	C	A ∧ B	B ⇒ C	(A ∧ B) ⇒ C	A ∧ (B ⇒ C)
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	F	T	T	T
T	F	F	F	T	T	T
F	T	T	F	T	T	F
F	T	F	F	F	T	F
F	F	T	F	T	T	F
F	F	F	F	T	T	F

Literal and Clause

- **Literal**: A single proposition or its negation:
  - Example: P, ¬P

- A **clause**: A propositional formula formed from a finite collection of literals and logical connectives:
  - Example: $P \vee Q \vee \neg R$

Rules of Inference

- **Sound inference rules**
  - Pattern of inference, that occur again and again
  - Soundness proven once and for all (truth-table)

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

- **Classic rules of inference**
  - Implication-Elimination, or *Modus Ponens (MP)*
    - $$\frac{\alpha \Rightarrow \beta,\ \alpha}{\beta}$$

e.g., Cloudy Λ Humid $\Rightarrow$ Rain
Cloudy Λ Humid

|= Rain

Rules of Inference

- **Classic rules of inference**
  - And-Elimination
    - $$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

    e.g. Cloudy $\wedge$ Humid $\models$ Cloudy

    Cloudy $\Rightarrow$ NoSunTan
  - And-Introduction
    - $$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

    e.g. Cloudy, Humid

    Cloudy $\wedge$ Humid $\Rightarrow$ Rain
  - Or-Introduction
    - $$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}$$
  - Double-Negation-Elimination
    - $$\frac{\neg\neg\alpha}{\alpha}$$

Rules of Inference

- **Resolution**
  - A technique of inference
  - Suppose *x* is a literal and S1 and S2 are two propositional sentences represented in the clausal form

- If $(x \vee S1) \Lambda (\neg x \vee S2)$. Then, we get $(S1 \vee S2)$
  - Here, $(S1 \vee S2)$ is the resolvent,
  - x is resolved upon

Rules of Inference

- **The resolution rule of inference**
  - Unit Resolution
  - The resolvent is one in which at least one of the parent clauses is a unit clause (i.e., a single literal)
    - $$\frac{\alpha \vee \beta,\ \ \neg\beta}{\alpha}$$

    e.g., Monday $\vee$ Tuesday, $\neg$ Monday |= Tuesday

    same as MP: $\dfrac{P \Rightarrow Q,\ P}{Q}$ i.e. $\dfrac{\neg\beta \Rightarrow \alpha,\ \neg\beta}{\alpha}$

Rules of Inference

- **The resolution rule of inference**
  - Full Resolution
    - $$\frac{\alpha \vee \beta, \quad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

Truth-table for the resolution

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
False	False	False	False	True	False
False	False	True	False	True	True
False	True	False	True	False	False
False	True	True	True	True	True
True	False	False	True	True	True
True	False	True	True	True	True
True	True	False	True	False	True
True	True	True	True	True	True

Equivalence Rules

- **Equivalent notations**
  - e.g., MP:

$$\frac{\alpha \Rightarrow \beta,\ \alpha}{\beta}$$

1) $\alpha \Rightarrow \beta, \alpha \quad |- \quad \beta$

2) $\alpha \Rightarrow \beta, \alpha \quad |= \quad \beta$

3) $$\frac{\begin{array}{c}\alpha \Rightarrow \beta \\ \alpha\end{array}}{\beta}$$

4) $((\alpha \Rightarrow \beta)\ \Lambda\ \alpha) \Rightarrow \beta$

Equivalence Rules

- **Equivalence rules**
  - Associativity: $\alpha \wedge (\beta \wedge \gamma) \Leftrightarrow (\alpha \wedge \beta) \wedge \gamma$
    $\alpha \vee (\beta \vee \gamma) \Leftrightarrow (\alpha \vee \beta) \vee \gamma$
  - Distributivity: $\alpha \wedge (\beta \vee \gamma) \Leftrightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$
    $\alpha \vee (\beta \wedge \gamma) \Leftrightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$
  - De Morgan's Law: $\neg(\alpha \vee \beta) \Leftrightarrow \neg\alpha \wedge \neg\beta$
    $\neg(\alpha \wedge \beta) \Leftrightarrow \neg\alpha \vee \neg\beta$

Complexity of Inference

- **Proof by truth-table**
  - Complete
    - The truth-table can always be written.
  - Exponential time complexity
    - A proof involving ***N*** proposition symbols requires $\mathbf{2^N}$ rows.
    - In practice, a proof may refer only to a small subset of the KB.

- **Monotonicity**
  - Knowledge always increases

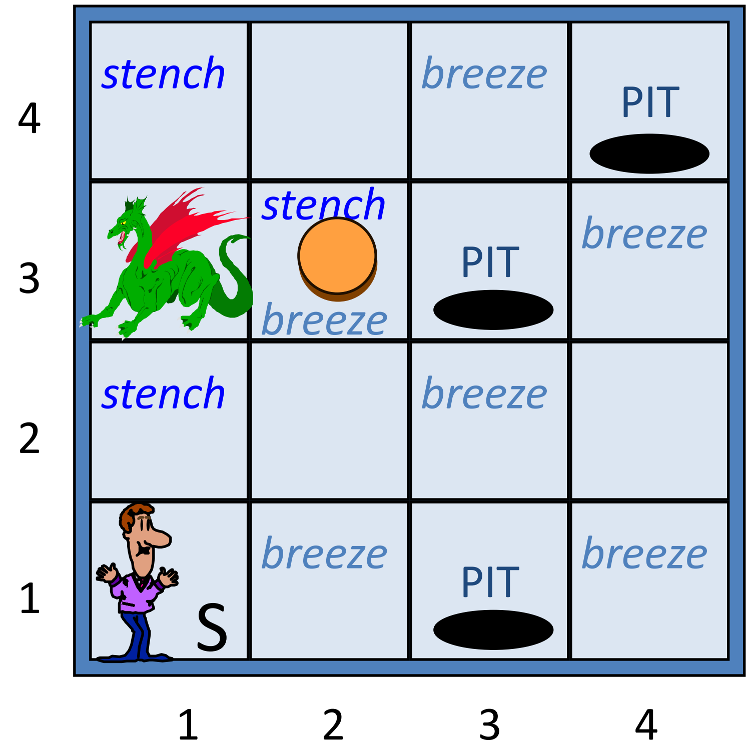
    $$\textbf{if}\ KB_1 \models \alpha\ \ \textbf{then}\ (KB_1 \cup KB_2) \models \alpha$$

    - Allows for local rules,
      e.g., Modus Ponens $\alpha \Rightarrow \beta, \alpha \vdash \beta$

Relooking at the Wumpus World

- **A reasoning agent**
  - Propositional logic as the "programming language"
  - Knowledge base (KB) as problem representation
    - Percepts
    - Knowledge
    - Actions

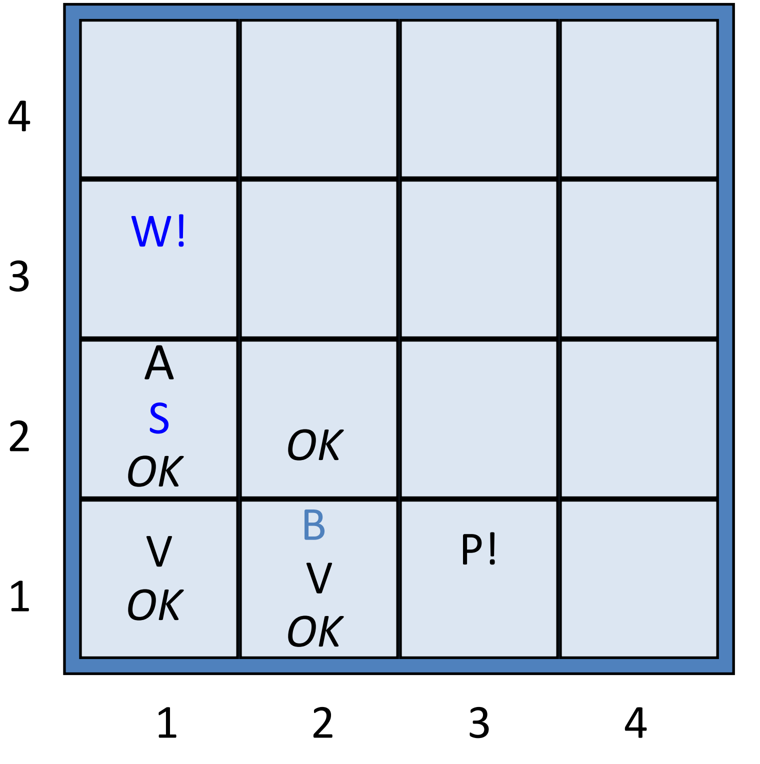
    – sentences
  - Rule of inference (e.g., Modus Ponens) as the algorithm that will find a solution



The Knowledge Base

- **TELLing the KB: percepts**
  - Syntax and semantics
    - Symbol **S11**, meaning "there is stench at [1,1]"
    - Symbol **B12**, meaning "there is breeze at [1,2]"
      …
- **Percept sentences**
  - Partial list:
    - ¬ S11, ¬ B11, ¬ G11, …
      ¬ S21, B21, ¬ G21, …
      S12, ¬ B12, ¬ G12, …
      …

[Stench , nil, nil, nil, nil]



The Knowledge Base

- **TELLing the KB: knowledge**
  - Rules about the environment
    - "All squares adjacent to the wumpus have stench."

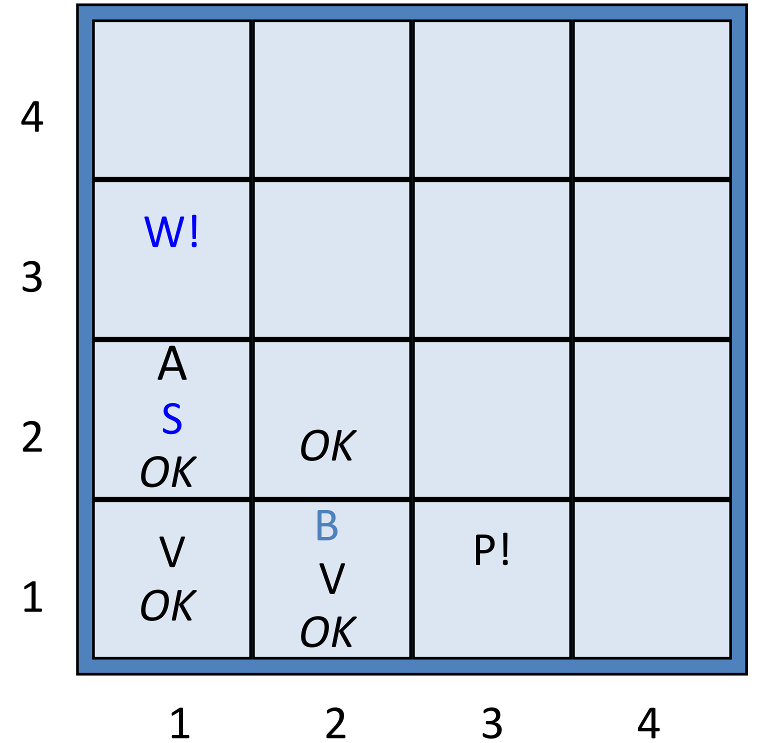
      $S12 \Rightarrow W11 \vee W12 \vee W22 \vee W13$
    - "A square with no stench has no wumpus and adjacent squares have no wumpus either."

      $\neg S11 \Rightarrow \neg W11 \wedge \neg W21 \wedge \neg W12$

      $\neg S21 \Rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$

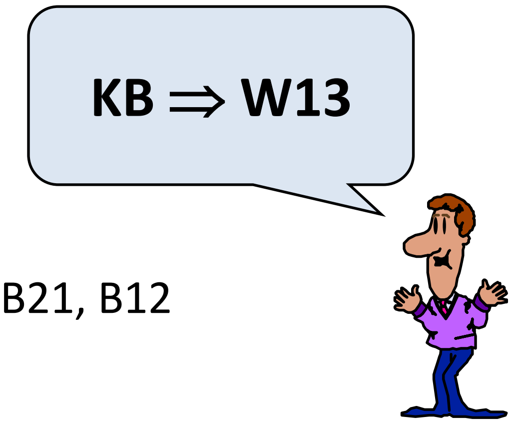
      $\neg S12 \Rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$

[Stench , nil, nil, nil, nil]



Finding the Wumpus

- **Checking the truth-table**
  - Exhaustive check: every row for which KB is true also has W13 true
    - 12 propositional symbols, i.e.
      S11, S21, S12, W11, W21, W12, W22, W13, W31, B11, B21, B12
    - 2^{12} = 4,096 rows
  - > *possible, but lengthy*



- **Reasoning by inference**
  - Application of a sequence of inference rules (proof)
    - Modus Ponens, And-Elimination, and Unit-Resolution

Proof for "KB $\Rightarrow$ W13"

Knowledge Base	Inferences
$\neg$ S11, $\neg$ B11, $\neg$ G11, $\neg$ S21, B21, $\neg$ G21, S12, $\neg$ B12, $\neg$ G12, **R1:** $\neg S11 \Rightarrow \neg W11 \wedge \neg W21 \wedge \neg W12$ **R2:** $\neg S21 \Rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$ **R3:** $\neg S12 \Rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$ **R4:** $S12 \Rightarrow W11 \vee W12 \vee W22 \vee W13$	1. Modus Ponens: $\neg$ S11, **R1** $\vdash$ $\neg W11 \wedge \neg W21 \wedge \neg W12$ 2. And-Elimination: ♦ $\vdash$ $\neg W11, \neg W21, \neg W12$ 3. Modus Ponens: $\neg$ S21, **R2** $\vdash$ $\neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$ 4. And-Elimination: ♦ $\vdash$ $\neg W11, \neg W21, \neg W22, \neg W31$

Proof for “KB $\Rightarrow$ W13”

Knowledge Base	Inferences
$\neg$ S11, $\neg$ B11, $\neg$ G11, $\neg$ S21, B21, $\neg$ G21, S12, $\neg$ B12, $\neg$ G11, **R1:** $\neg S11 \Rightarrow \neg W11 \land \neg W21 \land \neg W12$ **R2:** $\neg S21 \Rightarrow \neg W11 \land \neg W21 \land \neg W22 \land \neg W31$ **R3:** $\neg S12 \Rightarrow \neg W11 \land \neg W12 \land \neg W22 \land \neg W13$ **R4:** $S12 \Rightarrow W11 \lor W12 \lor W22 \lor W13$	KB += $\neg$W11, $\neg$W21, $\neg$W12, $\neg$W22, $\neg$W31 5. Modus Ponens: S12, **R4** $\vdash$ $(W13 \lor W12 \lor W22) \lor W11$ 6. Unit-Resolution: ♦, $\neg$W11 $\vdash$ $(W13 \lor W12) \lor W22$

Proof for “KB $\Rightarrow$ W13”

Knowledge Base	Inferences
$\neg$ S11, $\neg$ B11, $\neg$ G11, $\neg$ S21, B21, $\neg$ G21, S12, $\neg$ B12, $\neg$ G12, **R1:** $\neg S11 \Rightarrow \neg W11 \wedge \neg W21 \wedge \neg W12$ **R2:** $\neg S21 \Rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$ **R3:** $\neg S12 \Rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$ **R4:** $S12 \Rightarrow W11 \vee W12 \vee W22 \vee W13$	KB += $\neg$W11, $\neg$W21, $\neg$W12, $\neg$W22, $\neg$W31, $(W13 \vee W12) \vee W22$ 7. Unit-Resolution: ♦, $\neg$W22 $\vert$– $W13 \vee W12$ 8. Unit-Resolution: ♦, $\neg$W12 $\vert$– W13 **KB $\Rightarrow$ W13**

From Knowledge to Actions

- **TELLing the KB: actions**
  - Additional rules
    - e.g. “if the wumpus is 1 square ahead then do not go forward”

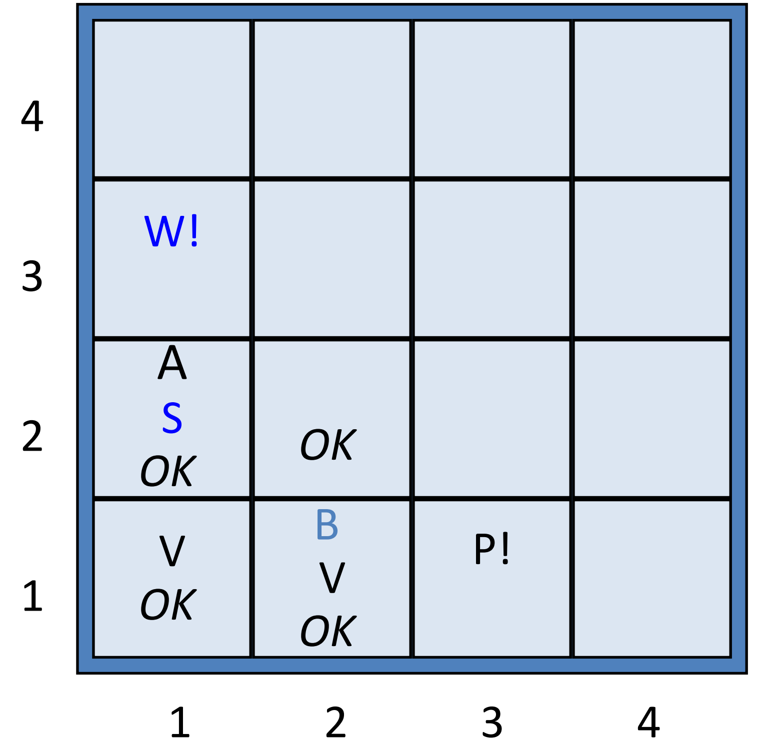
$A12 \wedge East \wedge W22 \Rightarrow \neg Forward$

$A12 \wedge North \wedge W13 \Rightarrow \neg Forward$

…

- **ASKing the KB**
  - Cannot ask “which action?” but “should I go forward?”

[Stench , nil, nil, nil, nil]



A Knowledge-Based Agent Using Propositional Logic

```
function Propositional-KB-Agent (percept) returns action
        static  KB,                         // a knowledge base
                t                           // a time counter, initially 0

        Tell (KB, Make-Percept-Sentence (percept, t))
        foreach action in the list of possible actions
        do
            if Ask (KB, Make-Action-Query (t, action)) then
                Tell (KB, Make-Action-Sentence (action, t))
                t ← t + 1
                return action
        end
```

Limits of Propositional Logic

- **A weak logic**
  - Too many propositions to TELL the KB
    - e.g., the rule "if the wumpus is 1 square ahead then do not go forward" needs 64 sentences (16 squares x 4 orientations)!
    - Result in increased time complexity of inference
  - Handling change is difficult
    - Need time-dependent propositional symbols
      e.g., A11 means "the agent is in square [1,1]" - when?
      at t = 0: A11-0; at t = 1: A21-1;
      at t = 2: A11-2
    - Need to rewrite rules as time-dependent
      e.g., A12-0 $\wedge$ East-0 $\wedge$ W22-0 $\Rightarrow$ ¬Forward-0
      A12-2 $\wedge$ East-2 $\wedge$ W22-2 $\Rightarrow$ ¬Forward-2

Summary

- **Propositional logic …**
  - Commits only to the existence of facts.
  - Has simple syntax and semantics and is therefore limited.

Thank you!