Artificial Intelligence

Lecture 9 – Knowledge-Based Agents and Logic

28 August 2024 1

1

Outline

- Knowledge-based agents
- Wumpus world
- Logic in general - models and entailment
- Propositional (Boolean) logic
- Equivalence, validity, satisfiability
- Inference rules and theorem proving
  - Forward chaining
  - Backward chaining
  - Resolution.

28 August 2024 2

2

Knowledge Base

Inference engine	← domain-independent algorithms
Knowledge base	← domain-specific content

- Knowledge base: Set of sentences in a formal language
- Declarative approach for building an agent
  - TELL it what it needs to know
  - Then it can ASK itself what to do - answers should follow from the KB
- Agents can be viewed at the knowledge level
  - What they know, regardless of implementation
- Or at the implementation level
  - Data structures in KB and algorithms that manipulate them.

28 August 2024 3

3

A Knowledge-based Agent

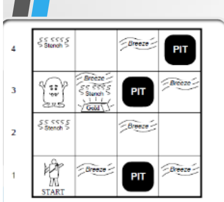
- The agent must be able to:
  - Represent states, actions, etc.
  - Incorporate new percepts
  - Update internal representations of the world
  - Deduce hidden properties of the world
  - Deduce appropriate actions.

```
function KB-Agent(percept) returns an action
  static: KB, a knowledge base
          t, a counter, initially 0, indicating time
  Tell(KB, Make-Percept-Sentence(percept, t))
  action ← Ask(KB, Make-Action-Query(t))
  Tell(KB, Make-Action-Sentence(action, t))
  t ← t + 1
  return action
```

28 August 2024 4

4

Wumpus World



Performance measure:
- Gold +1000, Death -1000, -1 per step, -10 for using the arrow

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

Actuators:
- Left turn, Right turn, Forward, Grab, Release, Shoot

Sensors:
- Breeze, Glitter, Smell.

28 August 2024 5

5

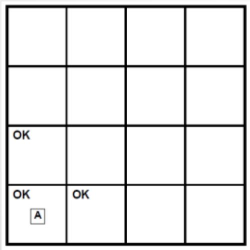
Wumpus World Characterization

- **Observable**? No - only local perception
- **Deterministic**? Yes -outcomes exactly specified
- **Episodic**? No - sequential at the level of actions
- **Static**? Yes -Wumpus and Pits do not move
- **Discrete**? Yes
- **Single-agent**? Yes

28 August 2024 6

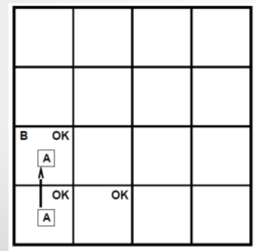
6

Exploring the Wumpus World



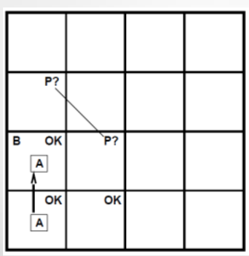
7

Exploring the Wumpus World



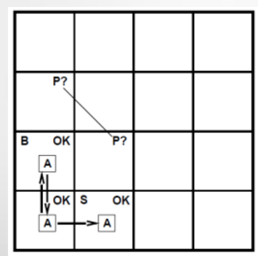
8

Exploring the Wumpus World



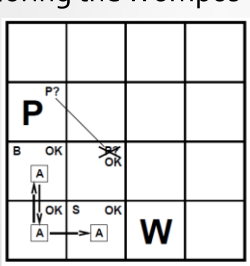
9

Exploring the Wumpus World



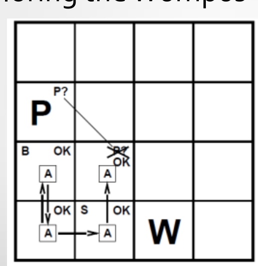
10

Exploring the Wumpus World



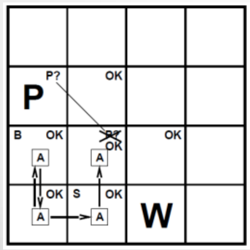
11

Exploring the Wumpus World



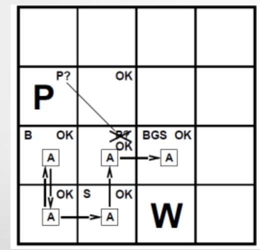
12

Exploring the Wumpus World



13

Exploring the Wumpus World



14

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 ≥ y is a sentence; x2 + y > is not a sentence
- x + 2 ≥ y is true iff the number x + 2 is no less than the number y
- x + 2 ≥ y is true in a world where x=7; y=1
- x + 2 ≥ y is false in a world where x=0; y =6

15

Entailment

Entailment means that one thing **follows** from another

$KB \models \alpha$

Knowledge base KB entails sentence B iff B is true in all worlds where KB is true

E.g., the KB containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"

E.g., x + y =4 entails 4=x + y

Entailment is a relationship between sentences (i.e., syntax) that is based on semantics.

16

Models

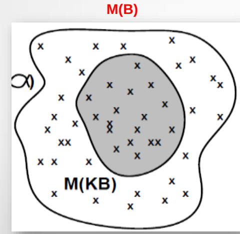
- Logicians typically think in terms of **models**, which are formally structured worlds with respect to which truth can be evaluated
- We say *m* is a model of a sentence B if B is true in *m*
  - M(B) is the set of all models of B
- Then KB |= B if and only if M(KB) ⊆ M(B)
  - E.g. KB = Giants won and Reds won
  - B = Giants won

M(B) could be also true for worlds that are different than the worlds of KB

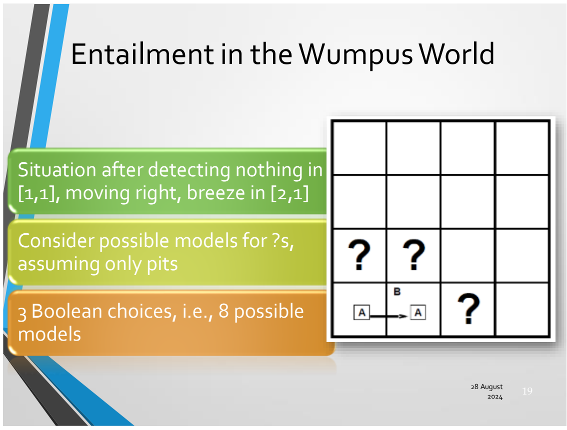
17

Models

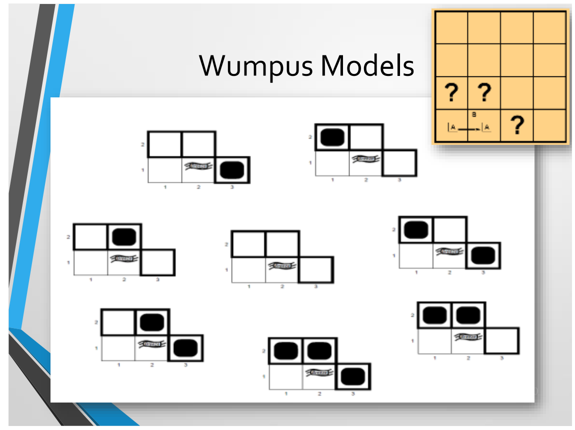
M(B)



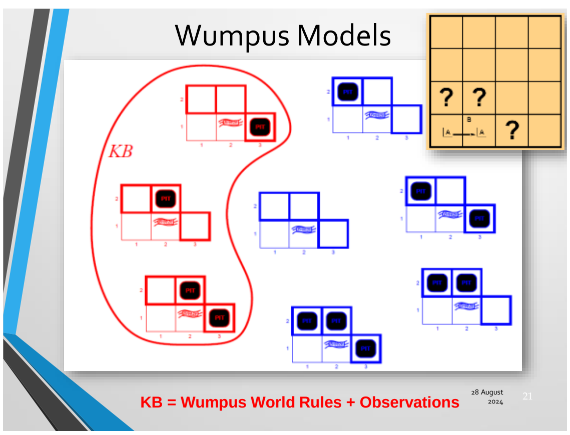
18



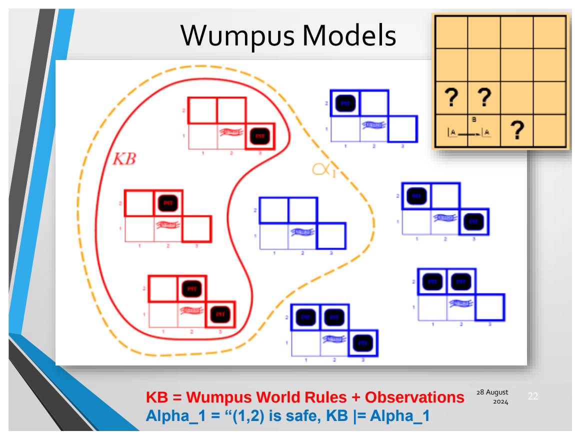
19



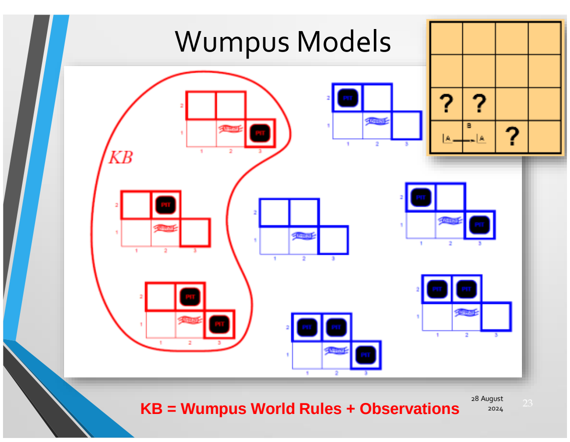
20



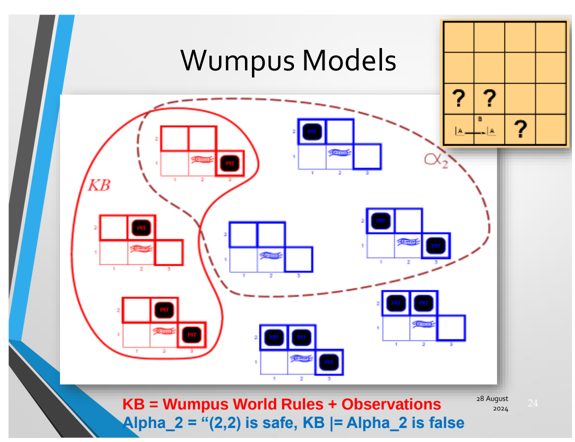
21



22



23



24

Inference

$KB \vdash_i \alpha$ = sentence α can be derived from KB by procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it

Soundness: i is sound if
whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

Completeness: i is complete if
whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

- First Order Logic (FOL): Allows complete and sound inference procedures
  - Will answer any question whose answer follows from what is known by the KB.

28 August 2024 25

25

Propositional Logic

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols P_1, P_2 etc are sentences

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)

If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)

If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Order of Precedence

If and Only If

If S1 then S2

28 August 2024 26

26

Propositional Logic: Semantics

Each model specifies true/false for each proposition symbol

E.g. $P_{1,2}$ $P_{2,2}$ $P_{3,1}$
$true$ $true$ $false$

If S1 is true, then I am claiming that S2 is true

(With these symbols, 8 possible models, can be enumerated automatically.)

Rules for evaluating truth with respect to a model m:

$\neg S$	is true iff	S	is false		
$S_1 \wedge S_2$	is true iff	S_1	is true **and**	S_2	is true
$S_1 \vee S_2$	is true iff	S_1	is true **or**	S_2	is true
$S_1 \Rightarrow S_2$	is true iff	S_1	is false **or**	S_2	is true
i.e.,	is false iff	S_1	is true **and**	S_2	is false
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true **and**	$S_2 \Rightarrow S_1$	is true

Simple recursive process evaluates an arbitrary sentence, e.g.,
$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = true \wedge (false \vee true) = true \wedge true = true$

28 August 2024 27

27

Truth Table for Connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

28 August 2024 28

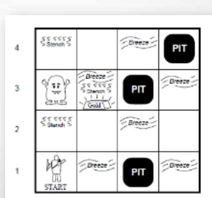
28

Wumpus World Sentences

Let $P_{i,j}$ be true if there is a pit in $[i,j]$.
Let $B_{i,j}$ be true if there is a breeze in $[i,j]$.

$\neg P_{1,1}$
$\neg B_{1,1}$
$B_{2,1}$

"Pits cause breezes in adjacent squares"



28 August 2024 29

29

Wumpus World Sentences

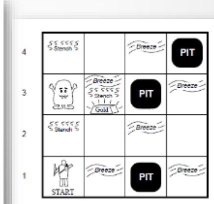
Let $P_{i,j}$ be true if there is a pit in $[i,j]$.
Let $B_{i,j}$ be true if there is a breeze in $[i,j]$.

$\neg P_{1,1}$ R1
$\neg B_{1,1}$ R4
$B_{2,1}$ R5

"Pits cause breezes in adjacent squares"

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$ R2
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$ R3

"A square is breezy **if and only if** there is an adjacent pit"



28 August 2024 30

30

Wumpus World Truth Table

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Enumerate rows (different assignments to symbols),
if KB is true in row, check that α is too

128 possible rows, only 3 make the KB true

$P_{1,2}$ is false for all 3; hence, no pit in $P_{1,2}$

Evaluate the entailed sentences in these rows

$P_{2,2}$ is both false and true; hence, no conclusion can be drawn about the pit being in $P_{2,2}$

31

Inference by Enumeration

Depth-first enumeration of all models is sound and complete

function TT-ENTAILS?(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic
  symbols ← a list of the proposition symbols in KB and α
  return TT-CHECK-ALL(KB, α, *symbols*, [ ])

function TT-CHECK-ALL(KB, α, *symbols*, *model*) **returns** *true* or *false*
  if EMPTY?(*symbols*) **then**
    if PL-TRUE?(KB, *model*) **then return** PL-TRUE?(α, *model*)
    else return *true*
  else do
    P ← FIRST(*symbols*); *rest* ← REST(*symbols*)
    return TT-CHECK-ALL(KB, α, *rest*, EXTEND(P, *true*, *model*)) **and**
      TT-CHECK-ALL(KB, α, *rest*, EXTEND(P, *false*, *model*))

$O(2^n)$ for n symbols; problem is **co-NP-complete**

32

Logical Equivalence

Two sentences are logically equivalent iff true in same models:
$\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) \quad \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) \quad \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

33

Validity and Satisfiability

A sentence is valid if it is true in all models,
  e.g., $True$, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:
  $KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

If KB is true, alpha is always true. Hence, I can say that alpha follows from KB

A sentence is satisfiable if it is true in some model
  e.g., $A \lor B$, C

A sentence is unsatisfiable if it is true in no models
  e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:
  $KB \models \alpha$ if and only if $(KB \land \neg\alpha)$ is unsatisfiable
i.e., prove α by *reductio ad absurdum*

Suppose that KB=true. Then, this will become unsatisfiable only when alpha is true. Hence, I can say that alpha follows from KB

34

Two Famous Inference Rules

- Modus Ponens $\dfrac{P \Rightarrow Q \;,\; P}{Q}$

Given that P implies Q, and I know that P is true, then I can infer Q

- And Rule $P \land Q$

Given that P AND Q is true, I can infer that P is true, and I can also infer that Q is true.

35

Resolution

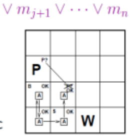
Conjunctive Normal Form (CNF—universal)
  conjunction of $\underbrace{\textbf{disjunctions} \text{ of } \textbf{literals}}_{\textbf{clauses}}$
  E.g., $(A \lor \neg B) \land (B \lor \neg C \lor \neg D)$

Resolution inference rule (for CNF): complete for propositional logic

$$\frac{\ell_1 \lor \cdots \lor \ell_k, \qquad m_1 \lor \cdots \lor m_n}{\ell_1 \lor \cdots \lor \ell_{i-1} \lor \ell_{i+1} \lor \cdots \lor \ell_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n}$$

where ℓ_i and m_j are complementary literals. E.g.,

$$\frac{P_{1,3} \lor P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}}$$



Resolution is sound and complete for propositional logic

36

Conversion to CNF

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and double-negation:

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributivity law ($\vee$ over $\wedge$) and flatten:

   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

37

Resolution Algorithm

- Resolution rule:

$$\frac{\alpha \vee \beta \qquad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

- Resolution refutation:
  - Convert all sentences to CNF
  - Negate the desired conclusion (converted to CNF)
  - Apply resolution rule until either
    - Derive false (a contradiction)
    - Can't apply any more
- Resolution refutation is sound and complete
  - If we derive a contradiction, then the conclusion follows from the axioms
  - If we can't apply any more, then the conclusion cannot be proved from the axioms.

38

Example

Propositional Resolution Example

Prove R

1	P v Q
2	P → R
3	Q → R

false v R

¬ R v false

false v false

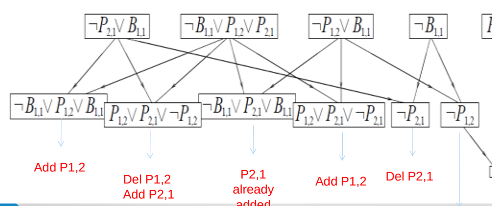
Step	Formula	Derivation
1	P v Q	Given
2	¬ P v R	Given
3	¬ Q v R	Given
4	¬ R	Negated conclusion
5	Q v R	1,2
6	¬ P	2,4
7	¬ Q	3,4
8	R	5,7
9	•	4,8

39

Resolution Example

$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

$KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$ $\alpha = \neg P_{1,2}$



This returns an empty clause. So, alpha is true.

40

Questions

41