Propositional Logic

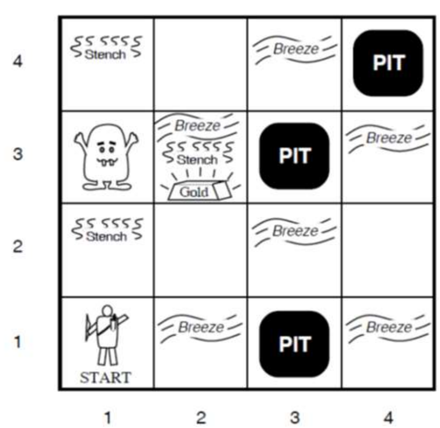
Sven Koenig, USC

Russell and Norvig, 3rd Edition, Sections 7.1-7.5

These slides are new and can contain mistakes and typos.
Please report them to Sven (skoenig@usc.edu).

Knowledge Representation and Reasoning

- Wumpus world, where knowledge representation and reasoning supports planning



Knowledge Representation and Reasoning

- Suppose that liars always speak what is false, and truth-tellers always speak what is true. Further suppose that Amy, Bob, and Cal are each either a liar or truth-teller. Amy says that Bob is a liar. Bob says that Cal is a liar. Cal says that Amy and Bob are liars.
  - Is Amy a truth-teller?
  - Is Bob a truth-teller?
  - Is Cal a truth-teller?
- Heads, I win; Tails, you lose.
  - Do I win?

Knowledge Representation and Reasoning

- Agents are given knowledge about the world.
- Knowledge representation:
  How can facts about the world be represented?
- Reasoning:
  How can an agent infer new facts from the given ones?

Knowledge Representation

- Knowledge representation languages should be
  expressive, concise, unambiguous, context independent and effective.
- Syntax
  Are “x+2=5” and “x+*y><“ well-formed formulas in arithmetic?
- Semantics
  When is “x+2=5” true in arithmetic?

Propositional Logic

- In propositional logic, sentences represent propositions
  (= statements that are either true or false).
  Sentences can refer to other sentences.
- Examples of propositions:
  P ≡ “2 is prime”
  Q ≡ “2 is even”
  R ≡ “2 is prime and 2 is even”

Syntax

- Syntax = what a well-formed sentence is.

- Sentence → AtomicSentence | ComplexSentence
- AtomicSentence → T(RUE) | F(ALSE) | Symbols
- Symbols → P | Q | R | …
- ComplexSentence → (Sentence) | NOT Sentence | Sentence Connective Sentence
- Connective → AND | OR | IMPLIES | EQUIV

Syntax

- Examples of well-formed sentences:
  P
  P IMPLIES (NOT R) ≡ P IMPLIES NOT R
  P AND (Q OR R)

- Precedence of the connectives:
  NOT (negation, write: "$\neg$", read: "not")
  AND (conjunction, write: "$\wedge$", read: "and")
  OR (disjunction, write: "$\vee$", read: "(inclusive) or" but not "either … or")
  IMPLIES (implication, write: "$\Rightarrow$", read: "implies" or "if … then")
  EQUIV (equivalence, write: "$\Leftrightarrow$", read: "is equivalent to" or "if and only if")

Semantics

- Semantics = when a sentence is true (= what it means).

Semantics

- Questions:
  Does “2 is prime” imply that “2 is prime”?
  Does “2 is prime” imply that “2 is even”?
  Does “2 is odd” imply that “2 is even”?

Semantics

- Questions:
  Does “2 is prime” imply that “2 is prime”?
  Does “2 is prime” imply that “2 is even”?
  Does “2 is odd” imply that “2 is even”?

- P ≡ “2 is prime”, Q ≡ “2 is even”, R ≡ “2 is odd”

- Questions:
  P IMPLIES P
  P IMPLIES Q
  R IMPLIES Q

Semantics

- Questions:
  Does “2 is prime” imply that “2 is prime”?
  Does “2 is prime” imply that “2 is even”?
  Does “2 is odd” imply that “2 is even”?

- P ≡ “2 is prime”, Q ≡ “2 is even”, R ≡ “2 is odd”

- Answers (“it depends” means that it depends on the interpretation):
  P IMPLIES P – valid (and satisfiable), so the answer is “yes”
  P IMPLIES Q – satisfiable but not valid, so the answer is “it depends”
  R IMPLIES Q – satisfiable but not valid, so the answer is “it depends”

Semantics

- An interpretation (= world = model) assigns each propositional symbol a truth value (namely, either true or false).
- Then, we can determine the truth value of any sentence, as follows:
  - The truth value of T(RUE) is t(rue).
  - The truth value of F(ALSE) is f(alse).
  - The truth value of a sentence can be determined as a function of the truth values of its parts (= compositional semantics), using the following truth tables:

P	Q	NOT P	P AND Q	P OR Q	P IMPLIES Q	P EQUIV Q
true	true	false	true	true	true	true
true	false		false	true	false	false
false	true	true	false	true	true	false
false	false		false	false	true	true

Semantics

- The semantics of NOT, AND, OR, IMPLIES and EQUIV correspond to the English “not”, “and”, “or”, “if … then…” and “if and only if”, respectively.

P	Q	NOT P	P AND Q	P OR Q	P IMPLIES Q	P EQUIV Q
true	true	false	true	true	true	true
true	false		false	true	false	false
false	true	true	false	true	true	false
false	false		false	false	true	true

- For example, …
  - Is the interpretation where P is false and Q is true consistent with the rule “if P (is true) then Q (is true)”? Yes, so “P IMPLIES Q” is true for this interpretation.
  - Is the interpretation where P is true and Q is false consistent with the rule “if P (is true) then Q (is true)”? No, so “P IMPLIES Q” is false for this interpretation.

Semantics

- Questions:
  Does "2 is prime" imply that "2 is prime"?
  Does "2 is prime" imply that "2 is even"?
  Does "2 is odd" imply that "2 is even"?

- P ≡ "2 is prime", Q ≡ "2 is even", R ≡ "2 is odd"

- Answers in our world, where P = Q = true and R = false:
  P IMPLIES P = true IMPLIES true = true – yes
  P IMPLIES Q = true IMPLIES true = true – yes (causality is not important)
  R IMPLIES Q = false IMPLIES true = true – yes (false implies everything)

Semantics

- From a sentence to a truth table (= each row in the truth table corresponds to one interpretation)
- Example: P OR (NOT P IMPLIES Q)

P	Q	NOT P	NOT P IMPLIES Q	P OR (NOT P IMPLIES Q)
true	true	false	true	true
true	false	false	true	true
false	true	true	true	true
false	false	true	false	false

Semantics

- From a truth table to a sentence
- Example: XOR (the exclusive OR, read: "either ... or" – we will not use it)

P	Q	P XOR Q	P OR Q
true	true	false	true
true	false	true	true
false	true	true	true
false	false	false	false

Exclusive OR: Either I go running or (I go) swimming.
Inclusive OR: I go running or (= and/or) (I go) swimmi

Semantics

- From a truth table to a sentence
- Example: XOR

P	Q	P XOR Q	P AND Q	P AND NOT Q	NOT P AND Q	NOT P AND NOT Q
true	true	false	true	false	false	false
true	false	true	false	true	false	false
false	true	true	false	false	true	false
false	false	false	false	false	false	true

- The truth table describes (P AND NOT Q) OR (NOT P AND Q).
- It describes P EQUIV NOT Q (and many other sentences) as well.

Semantics

- A sentence is …
  - valid (= a tautology)
    if and only if it is true for all interpretations
    (if and only if it is true for all rows of the truth table)
  - satisfiable
    if and only if it is true for at least one interpretation
    (if and only if it is true for at least one row of the truth table)
  - unsatisfiable (= a contradiction)
    if and only if it is true for no interpretation
    (if and only if it is false for all rows of the truth table)

Semantics

- Examples:
  TRUE
  FALSE
  P
  P AND NOT P
  P OR NOT P
  P IMPLIES Q

Semantics

- Examples:
  TRUE – valid (and satisfiable)
  FALSE – unsatisfiable
  P – satisfiable
  P AND NOT P – unsatisfiable
  P OR NOT P – valid (and satisfiable)
  P IMPLIES Q – satisfiable

"Same Truth Value" (Meta Equivalence)

- $S \equiv S'$
  if and only if S and S' have the same truth value for all interpretations
  (if and only if their truth values are the same for all rows of the truth table).

- Meta equivalence differs from equivalence in that it is not part of propositional logic, that is, $S \equiv S'$ is not a sentence in propositional logic.

- Examples:
  P IMPLIES P $\equiv$ TRUE
  P IMPLIES Q $\equiv$ Q IMPLIES P

“Same Truth Value” (Meta Equivalence)

- S ≡ S’
  if and only if S and S’ have the same truth value for all interpretations
  (if and only if their truth values are the same for all rows of the truth table).

- Examples:
  P IMPLIES P ≡ TRUE – yes (that is, P IMPLIES P is valid)
  P IMPLIES Q ≡ Q IMPLIES P – no

“Same Truth Value” (Meta Equivalence)

- First way how to prove S ≡ S’:
  - Truth tables: iterate through all interpretations and check the definition, namely that S and S’ have the same truth value.

- Example: P EQUIV NOT Q ≡ (NOT P OR NOT Q) AND (Q OR P)

P	Q	P EQUIV NOT Q	(NOT P OR NOT Q) AND (Q OR P)
true	true	false	false
true	false	true	true
false	true	true	true
false	false	false	false

“Same Truth Value” (Meta Equivalence)

- First way how to prove $S \equiv S'$:
  - Iterate through all interpretations and check the definition, namely that S and S’ have the same truth value.

- This can be very time-consuming and thus impractical.
- If S and S’ together contain 100 symbols, one needs to check 2^{100} interpretations!

“Same Truth Value” (Meta Equivalence)

- Second way how to prove $S \equiv S'$:
  - Syntactic manipulation of S and S’ via transformation of S to S’ using rewrite rules.

"Same Truth Value" (Meta Equivalence)

- Rewrite rules
  P EQUIV Q ≡ (P IMPLIES Q) AND (Q IMPLIES P)
  P IMPLIES Q ≡ NOT P OR Q
  NOT NOT P ≡ P
  NOT (P AND Q) ≡ NOT P OR NOT Q
  NOT (P OR Q) ≡ NOT P AND NOT Q
  P AND Q OR R ≡ (P OR R) AND (Q OR R)
  P OR Q AND R ≡ (P OR Q) AND (P OR R)
  (P OR Q) AND R ≡ P AND R OR Q AND R
  P AND (Q OR R) ≡ P AND Q OR P AND R
  and many more

"Same Truth Value" (Meta Equivalence)

- If S ≡ S' then one can use S' instead of S in a larger sentence without changing the truth value of the larger sentence for any interpretation.

- Example (applies the rewrite rules from the next slide to obtain the conjunctive normal form of P EQUIV NOT Q):
  P EQUIV NOT Q
  ≡ (P IMPLIES NOT Q) AND (NOT Q IMPLIES P)
  ≡ (NOT P OR NOT Q) AND (NOT NOT Q OR P)
  ≡ $\underbrace{\text{(NOT P OR NOT Q)}}_{\text{Clause of length 2}}$ AND $\underbrace{\text{(Q OR P)}}_{\text{Clause of length 2}}$

“Same Truth Value” (Meta Equivalence)

- We can use rewrite rules to change a sentence into its conjunctive normal form, namely into a conjunction of (disjunctions of propositional symbols or their negations). Each disjunct is called a clause.
  - Eliminate EQUIV with the following rewrite rule:
    P EQUIV Q ≡ (P IMPLIES Q) AND (Q IMPLIES P)
  - Eliminate IMPLIES with the following rewrite rule:
    P IMPLIES Q ≡ NOT P OR Q
  - Move NOT inward with the following rewrite rules:
    NOT NOT P ≡ P
    NOT (P AND Q) ≡ NOT P OR NOT Q
    NOT (P OR Q) ≡ NOT P AND NOT Q
  - Move OR inward with the following rewrite rules:
    P AND Q OR R ≡ (P OR R) AND (Q OR R)
    P OR Q AND R ≡ (P OR Q) AND (P OR R)
  - Flatten nested conjuncts and disjuncts with the following rewrite rules:
    P AND (Q AND R) ≡ P AND Q AND R
    (P AND Q) AND R ≡ P AND Q AND R
    P OR (Q OR R) ≡ P OR Q OR R
    (P OR Q) OR R ≡ P OR Q OR R

Knowledge Representation and Reasoning

- Suppose that liars always speak what is false, and truth-tellers always speak what is true. Further suppose that Amy, Bob, and Cal are each either a liar or truth-teller. Amy says that Bob is a liar. Bob says that Cal is a liar. Cal says that Amy and Bob are liars.
  - Is Amy a truth-teller?
  - Is Bob a truth-teller?
  - Is Cal a truth-teller?
- Heads, I win; Tails, you lose.
  - Do I win?

Knowledge Representation and Reasoning

- Agents are given knowledge about the world.
- Knowledge representation:
  How can facts about the world be represented?
- Reasoning:
  How can an agent infer new facts from the given ones?

Entailment (Meta Implication)

- Is P IMPLIES Q true?
- It depends on the interpretation.
- From now on, the computer is not provided with the interpretation (because it would be inconvenient to specify a truth value for each symbol).
- Then, it can say “yes” only for valid sentences and “no” only for invalid sentences. In many cases, it cannot say “yes” or “no”.
- We address this issue by providing the computer with some knowledge of the world in terms of a knowledge base of sentences that are known to be true. This restricts the possible interpretations to those that make all sentences in the knowledge base true.

Entailment (Meta Implication)

- User: Is P IMPLIES Q true? (short: P IMPLIES Q?)
- Computer: I don't know. Could be true, could be false. It depends on the interpretation.
- User: Consider only the interpretations where P is false. Is P IMPLIES Q true? (short: NOT P ⊨ P IMPLIES Q)
- Computer: Indeed, it is true: NOT P ⊨ P IMPLIES Q.
- Comment: The user did not specify a unique interpretation but rather a subset of all interpretations which contains two possible interpretations (P is false but Q can be true or false). But this is sufficient for the computer to determine that P IMPLIES Q must be true.

Entailment

- KB $\equiv S_1$ AND S_2 AND … AND S_n
- KB $\equiv S_1, S_2, \ldots, S_n$
- KB

  S_1
  S_2
  S_3
  S_4

- where KB = knowledge base and $S_1, S_2, \ldots, S_n$ = sentences

Entailment (Meta Implication)

- KB ⊨ S (entailment, read: "entails")
  if and only if, whenever KB is true for an interpretation, then S is also true for that interpretation.
- Entailment differs from implication in that it is not part of propositional logic, that is, S ⊨ S' is not a sentence in propositional logic.
- Example:
  "Heads, I win; Tails, you lose." ⊨ "I win."

Entailment (Meta Implication)

- KB ⊨ S (entailment, read: "entails")
  if and only if, whenever KB is true for an interpretation, then S is also true for that interpretation.
- Examples:
  P ⊨ Q
  Q ⊨ P
  P AND Q ⊨ P
  P ⊨ P AND Q
  FALSE ⊨ P
  FALSE ⊨ NOT P

Entailment (Meta Implication)

- KB ⊨ S (entailment, read: "entails")
  if and only if, whenever KB is true for an interpretation, then S is also true for that interpretation.

- Examples:
  P ⊨ Q – no
  Q ⊨ P – no
  P AND Q ⊨ P – yes
  P ⊨ P AND Q – no
  FALSE ⊨ P – yes (an invalid knowledge base entails everything)
  FALSE ⊨ NOT P – yes (an invalid knowledge base entails everything)

These two entailments hold because there are no counter examples, that is, no interpretations that make FALSE true and P (or NOT P) false.

Entailment (Meta Implication)

- KB ⊨ S (entailment, read: "entails")
  if and only if, whenever KB is true for an interpretation, then S is also true for that interpretation.

- Note: KB ⊭ S (read: "does not entail") is not the same as KB ⊨ NOT S.

- Example:
  P ⊨ Q – no (in other words, P ⊭ Q)
  P ⊨ NOT Q – no (in other words, P ⊭ NOT Q)

Entailment (Meta Implication)

- First way how to prove KB ⊨ S:
  - Truth tables: iterate through all interpretations and check the definition, namely that, whenever KB is true for an interpretation, then S is also true for that interpretation.

Entailment Example

- “heads, I win; tails, you lose” ⊨ “I win”
- HE ≡ “heads”, TA ≡ “tails”, IW ≡ “I win”, UL ≡ “you lose”
- KB ≡ HE IMPLIES IW, TA IMPLIES UL ⊨ IW – no

HE	TA	IW	UL	HE IMPLIES IW	TA IMPLIES UL	KB	IW
true	true	true	true	true	true	true	true
true	true	true	false	true	false	false	true
true	true	false	true	false	true	false	false
true	true	false	false	false	false	false	false
true	false	true	true	true	true	true	true
true	false	true	false	true	true	true	true
true	false	false	true	false	true	false	false
true	false	false	false	false	true	false	false
false	true	true	true	true	true	true	true
false	true	true	false	true	false	false	true
false	true	false	true	true	true	true	false
false	true	false	false	true	false	false	false
false	false	true	true	true	true	true	true
false	false	true	false	true	true	true	true
false	false	false	true	true	true	true	false
false	false	false	false	true	true	true	false

Entailment Example

background knowledge is important
and needs to be represented explicitly

- "heads, I win; tails, you lose; either heads or tails; I win if and only if you lose" ⊨ "I win"
- HE ≡ "heads", TA ≡ "tails", IW ≡ "I win", UL ≡ "you lose"
- KB ≡ HE IMPLIES IW, TA IMPLIES UL, HE EQUIV NOT TA, IW EQUIV UL ⊨ IW – yes

HE	TA	IW	UL	HE IMPLIES IW	TA IMPLIES UL	HE EQUIV NOT TA	IW EQUIV UL	KB	IW
true	true	true	true	true	true	false	true	false	true
true	true	true	false	true	false	false	false	false	true
true	true	false	true	false	true	false	false	false	false
true	true	false	false	false	false	false	true	false	false
true	false	true	true	true	true	true	true	true	true
true	false	true	false	true	true	true	false	false	true
true	false	false	true	false	true	true	false	false	false
true	false	false	false	false	true	true	true	false	false
false	true	true	true	true	true	true	true	true	true
false	true	true	false	true	false	true	false	false	true
false	true	false	true	true	true	true	false	false	false
false	true	false	false	true	false	true	true	false	false
false	false	true	true	true	true	false	true	false	true
false	false	true	false	true	true	false	false	false	true
false	false	false	true	true	true	false	false	false	false
false	false	false	false	true	true	false	true	false	false

Entailment (Meta Implication)

- First way how to prove KB ⊨ S:
  - Truth tables: iterate through all interpretations and check the definition, namely that, whenever KB is true for an interpretation, then S is also true for that interpretation.

- Using truth tables to prove entailment shows that the question of entailment is decidable.
- But this can be very time-consuming and thus impractical.
- If KB and S together contain 100 symbols,
  one needs to check 2^{100} interpretations!

Entailment (Meta Implication)

- Second way how to prove KB ⊨ S:
  - Syntactic manipulation of KB and S via inference procedures

Inference Procedures

- KB ⊢ S if and only if the inference procedure can infer S from KB.
- An inference procedure is sound if and only if KB ⊨ S whenever KB ⊢ S.
- An inference procedure is complete if and only if KB ⊢ S whenever KB ⊨ S.

Inference Procedures

- Inference procedures are the repeated application of inference rules.
- KB $\overset{R}{\vdash}$ S (alternative: "$\frac{KB}{S}$") if and only if inference rule R can infer S from KB.
- An inference rule R is sound if and only if KB ⊨ S whenever KB $\overset{R}{\vdash}$ S.
- If KB ⊨ S then KB ⊨ KB AND S according to the definition of entailment.
- Therefore, for a sound inference rule R,
  if KB $\overset{R}{\vdash}$ S and KB AND S $\overset{R}{\vdash}$ S'
  then KB ⊨ S and KB AND S ⊨ S'
  then KB ⊨ KB AND S and KB AND S ⊨ S'
  then KB ⊨ S'.
- In other words, a sentence inferred by a sound inference rule can be put into the KB before the inference rule is used again.
- We will look at two inference rules: resolution and modus ponens.

Resolution

$$\frac{\text{P OR Q, NOT Q OR R}}{\text{P OR R}}$$ (write: "$\overset{\text{Resolution}}{\vdash}$", read: "resolve to")

- P, Q AND R can be arbitrary sentences. Afterwards, we remove duplicates from "P OR R".
- Examples:

P ≡ A OR NOT E, Q ≡ C, R ≡ NOT B OR F

$$\frac{\text{A OR C OR NOT E, NOT B OR NOT C OR F}}{\text{A OR NOT E OR NOT B OR F}} \qquad \frac{\text{A, NOT A}}{\text{EMPTY (= FALSE)}}$$

$$\frac{\text{A OR B, NOT A OR B}}{\text{B}} \qquad \frac{\text{A OR B, NOT A OR NOT B}}{\text{TRUE}}$$

Resolution

- Resolution uses two clauses "P OR Q" and "NOT Q OR R" to produce a new clause "P OR R".
- Resolution is just the transitivity of implications.

$$\frac{\text{P OR Q, NOT Q OR R}}{\text{P OR R}} \qquad \frac{\text{NOT P} \Rightarrow \text{Q, Q} \Rightarrow \text{R}}{\text{NOT P} \Rightarrow \text{R}}$$

Resolution

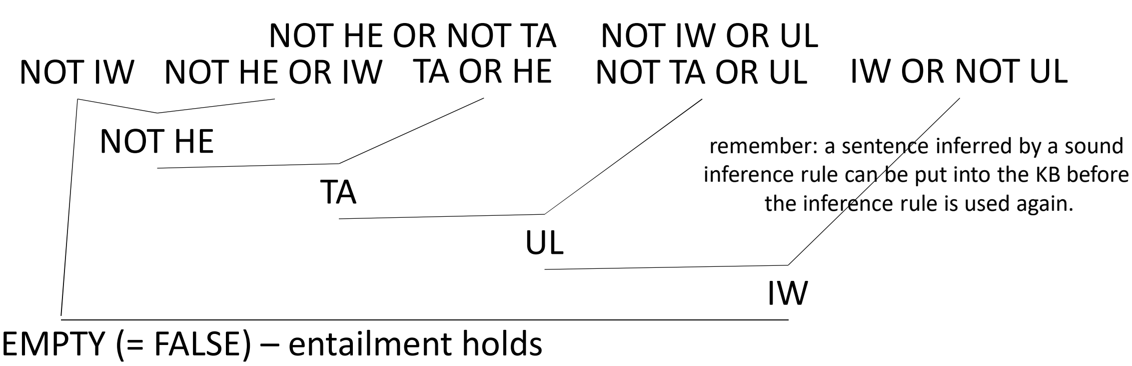
- Using KB $\overset{\text{Resolution}}{\vdash}$ S to show KB $\vDash$ S is sound but not complete.
- Example:
  P $\vDash$ Q OR NOT Q but not P $\overset{\text{Resolution}}{\vdash}$ Q OR NOT Q
- Using KB AND NOT S $\overset{\text{Resolution}}{\vdash}$ FALSE to show KB $\vDash$ S is sound and complete.
- Therefore, we will always use the following scheme:
  - Transform KB AND NOT S into conjunctive normal form.
  - Apply resolution to derive EMPTY (= FALSE).
- This is a proof by contradiction.

Resolution

- “heads, I win; tails, you lose” entails “I win”.
- In other words, whenever “heads, I win; tails, you lose” is true for an interpretation, then “I win” is also true for that interpretation.
- In other words, there is no interpretation that makes “heads, I win; tails, you lose” and NOT(“I win”) both true.
- In other words, “heads, I win; tails, I lose” AND NOT(“I win”) entails FALSE.

Resolution Example 1

- “heads, I win; tails, you lose; $\overbrace{\text{either heads or tails; I win if and only if you lose}}^{\text{background knowledge is important and needs to be represented explicitly}}$” $\overset{?}{\models}$ “I win”
- HE ≡ “heads”, TA ≡ “tails”, IW ≡ “I win”, UL ≡ “you lose”
- HE IMPLIES IW, TA IMPLIES UL, HE EQUIV NOT TA, IW EQUIV UL, NOT IW



EMPTY (= FALSE) – entailment holds

Resolution

- Resolution always terminates for propositional logic.
- If KB $\models$ S, then resolution will eventually produce EMPTY (= FALSE).
- A good search strategy helps to produce EMPTY quickly, e.g.
  - resolve with clauses of size 1 whenever possible
    because a clause of length n and a clause of size 1 resolve to a clause of size n-1.
  - resolve with clauses derived from NOT S
    because FALSE cannot be produced from a satisfiable KB alone.
- If KB $\not\models$ S, then resolution will not produce EMPTY and will eventually not be able to produce new clauses.
- This requires an exhaustive enumeration of all obtainable clauses and can thus take a long time.

Resolution Example 2

- P OR Q $\overset{?}{\models}$ Q
- P OR Q, NOT Q

P OR Q   NOT Q

P

- Resolution produces no additional clauses.
- It does not produce EMPTY (= FALSE) – entailment does not hold.