Rational Agents

Paolo Turrini

Department of Computing, Imperial College London

Introduction to Artificial Intelligence
2nd Part

What you have seen

You have seen examples of computational problem-solving:

- Search
- Planning
- Pattern recognition via neural networks

What we will be looking at

An **agent**, a mathematical entity acting in a simple world

What we will be looking at

An **agent**, a mathematical entity acting in a simple world

- Able to reason about the world around
  - True facts (knowledge)
  - Plausible facts (beliefs)

What we will be looking at

An **agent**, a mathematical entity acting in a simple world

- Able to reason about the world around
  - True facts (knowledge)
  - Plausible facts (beliefs)
- Able to take decisions under uncertainty
  - Imperfect and incomplete information
  - Quantifying uncertainty, attaching probabilities
  - Going for uncertain outcomes, calculating expected utility

What we will be looking at

An **agent**, a mathematical entity acting in a simple world

- Able to reason about the world around
  - True facts (knowledge)
  - Plausible facts (beliefs)
- Able to take decisions under uncertainty
  - Imperfect and incomplete information
  - Quantifying uncertainty, attaching probabilities
  - Going for uncertain outcomes, calculating expected utility
- Able to update their beliefs when confronted with new information (learning)

What is rationality?

Robert J. Aumann

Nobel Prize Winner

'A person's behaviour is **rational** *if it is in their best interests, given their information.'*

What is rationality?

Robert J. Aumann
Nobel Prize Winner

'A person's behaviour is **rational** *if it is in their best interests, given their information.'*

Agents (not only humans) can be rational!

The lectures

- The agent and the world
  - Actions and knowledge
  - Inference
- Good decisions
  - Chance
  - Gains
- Good decisions in time
  - Chance and gains in time
  - Patience
  - Finding the best strategy
- Learning from experience
  - Finding a reasonable strategy

The lectures

- The agent and the world (**Knowledge Representation**)
  - Actions and knowledge
  - Inference
- Good decisions
  - Chance
  - Gains
- Good decisions in time
  - Chance and gains in time
  - Patience
  - Finding the best strategy
- Learning from experience
  - Finding a reasonable strategy

The lectures

- The agent and the world (**Knowledge Representation**)
  - Actions and knowledge
  - Inference
- Good decisions (**Risk and Decisions**)
  - Chance
  - Gains
- Good decisions in time
  - Chance and gains in time
  - Patience
  - Finding the best strategy
- Learning from experience
  - Finding a reasonable strategy

The lectures

- The agent and the world (**Knowledge Representation**)
  - Actions and knowledge
  - Inference
- Good decisions (**Risk and Decisions**)
  - Chance
  - Gains
- Good decisions in time (**Markov Decision Processes**)
  - Chance and gains in time
  - Patience
  - Finding the best strategy
- Learning from experience
  - Finding a reasonable strategy

The lectures

- The agent and the world (**Knowledge Representation**)
  - Actions and knowledge
  - Inference
- Good decisions (**Risk and Decisions**)
  - Chance
  - Gains
- Good decisions in time (**Markov Decision Processes**)
  - Chance and gains in time
  - Patience
  - Finding the best strategy
- Learning from experience (**Reinforcement Learning**)
  - Finding a reasonable strategy

The lectures

- The agent and the world (**Knowledge Representation**)
  - Actions and knowledge
  - Inference
- Good decisions (**Risk and Decisions**)
  - Chance
  - Gains
- Good decisions in time (**Markov Decision Processes**)
  - Chance and gains in time
  - Patience
  - Finding the best strategy
- Learning from experience (**Reinforcement Learning**)
  - Finding a reasonable strategy

The tutorials

- Week 1 (today): nothing
- Week 2: Logic
- Week 3: Decision-making
- Week 4 (or 5, I don't know yet): MDPs

The tutorials

- Week 1 (today): nothing
- Week 2: Logic
- Week 3: Decision-making [**last year's exam**]
- Week 4 (or 5, I don't know yet): MDPs

The tutorials

- Week 1 (today): nothing
- Week 2: Logic
- Week 3: Decision-making [**last year's exam**]
- Week 4 (or 5, I don't know yet): MDPs

'Difficult in training, easy in game'
(Mark Dvoretskij, Chess Grandmaster)

The tutorials

- Week 1 (today): nothing
- Week 2: Logic
- Week 3: Decision-making [**last year's exam**]
- Week 4 (or 5, I don't know yet): MDPs
- Last day: Reinforcement Learning Lab

'Difficult in training, easy in game'
(Mark Dvoretskij, Chess Grandmaster)

The book

Stuart Russell and Peter Norvig
Artificial Intelligence: a modern approach.
3rd Edition.

- You can get it for free. I'm not suggesting to download it.
- Lots of useful exercises

How to contact me

- Come and talk to me (After Thursday's class, Huxley 452)
- Send me an email (p.turrini@ic.ac.uk)
- Piazza

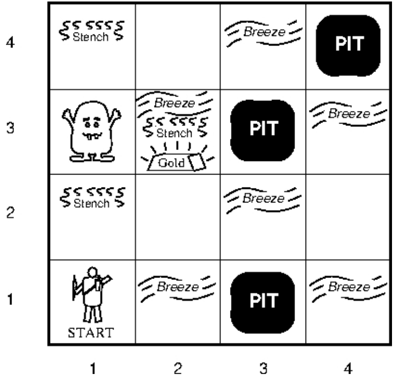
Knowledge Representation

An uncertain world

The main reference

Stuart Russell and Peter Norvig
Artificial Intelligence: a modern approach
Chapters 7-9

The Wumpus World



The Wumpus World

Sensors	Breeze, Glitter, Smell
Actuators	Up, Down, Left, Right, Grab, Release, Shoot, Climb
Rewards	1000 escaping with gold, -1000 dying, -10 using arrow, -1 walking

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

Knowledge base

- A set of sentences representing what the agent thinks about the world.
  - 'I am in [2,1]'
  - 'I am out of arrows'
  - 'I smell Wumpus'
  - 'I'd better not go forward'
- We interpret it as what the agent **knows**, but it works just fine for what the agent **believes**.

Updating the knowledge base

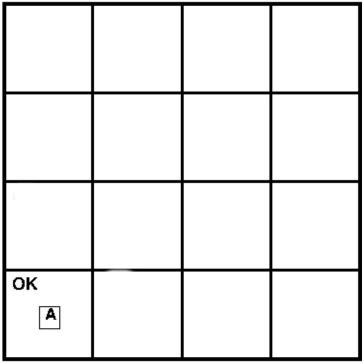
- What we TELL the knowledge base
- What we ASK the knowledge base

```
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
```

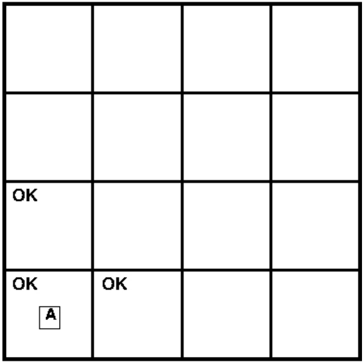
Rational explorations

- The starting state...



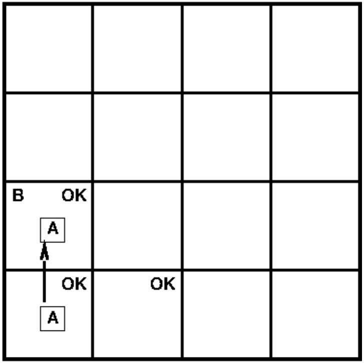
Rational explorations

- and what we know.



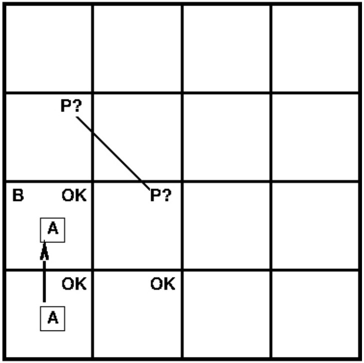
Rational explorations

- B stands for Breeze



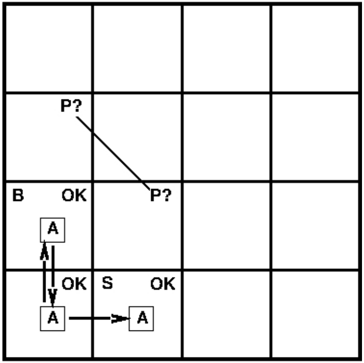
Rational explorations

- Where is the pit?
- We are ruling out one square!



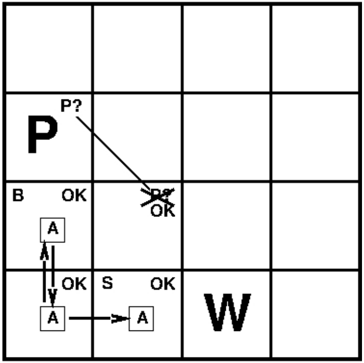
Rational explorations

- S stands for smell
- What do we know?



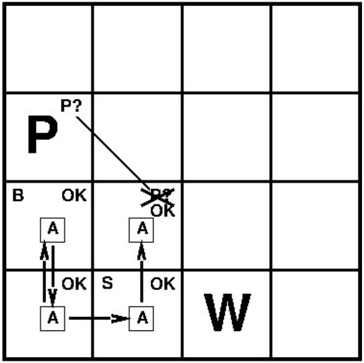
Rational explorations

- Logic is the key!



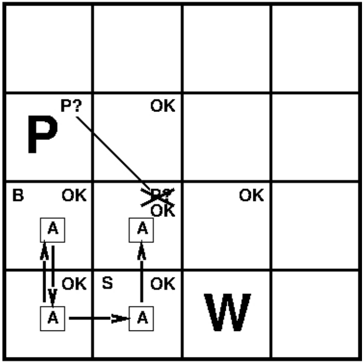
Rational explorations

- The further we go the more we know



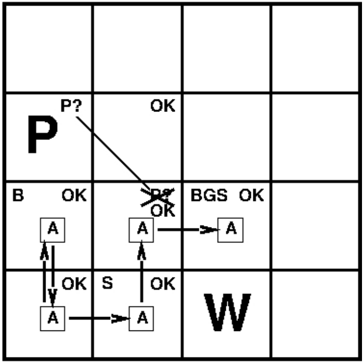
Rational explorations

- The further we go the more we know



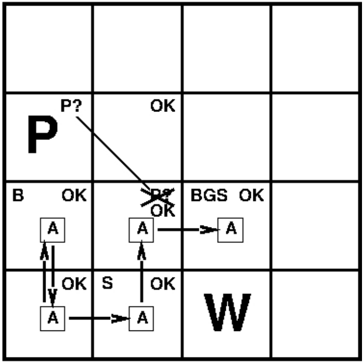
Rational explorations

- Gold!



Rational explorations

- We know the way out
- Game over



Representing the Wumpus World

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 $[i, j]$.

Representing the Wumpus World

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 $[i, j]$.

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

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

$$B_{2,1}$$

Representing the Wumpus World

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"

Representing the Wumpus World

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"

$$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$$
$$B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$$

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

Expressivity: at what cost?

- OK if we were only dealing with finite objects
- But even then we would have to enumerate all the possibilities

Expressivity: at what cost?

- OK if we were only dealing with finite objects
- But even then we would have to enumerate all the possibilities

Propositional Logic lacks expressive power

First order logic

- Massive increase of expressivity
- But there are costs, e.g., decidability
- We will see how to exploit the gains while limiting the costs

KB with FOL

- We can encode the KB at each particular time point using FOL

```
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
```

Representing the Wumpus World

- You already know how to describe the WW in first order logic

Representing the Wumpus World

- You already know how to describe the WW in first order logic
  - Percept (at given time), e.g.,
    $Percept([Stench, Breeze, Glitter], 5)$ or
    $Percept([None, Breeze, None], 3)$

Representing the Wumpus World

- You already know how to describe the WW in first order logic
  - Percept (at given time), e.g., $Percept([Stench, Breeze, Glitter], 5)$ or $Percept([None, Breeze, None], 3)$
  - Starting Knowledge Base, e.g., $\neg AtGold(0)$

Representing the Wumpus World

- You already know how to describe the WW in first order logic
  - Percept (at given time), e.g., $Percept([Stench, Breeze, Glitter], 5)$ or $Percept([None, Breeze, None], 3)$
  - Starting Knowledge Base, e.g., $\neg AtGold(0)$
  - Axioms to generate new knowledge from percepts, e.g., $\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$

Representing the Wumpus World

- You already know how to describe the WW in first order logic
  - Percept (at given time), e.g., $Percept([Stench, Breeze, Glitter], 5)$ or $Percept([None, Breeze, None], 3)$
  - Starting Knowledge Base, e.g., $\neg AtGold(0)$
  - Axioms to generate new knowledge from percepts, e.g., $\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$
  - Axioms to generate actions (plans) from KB, e.g., $\forall\, t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \;\Rightarrow\; Action(Grab, t)$

Representing the Wumpus World

- You already know how to describe the WW in first order logic
  - Percept (at given time), e.g.,
    $Percept([Stench, Breeze, Glitter], 5)$ or
    $Percept([None, Breeze, None], 3)$
  - Starting Knowledge Base, e.g., $\neg AtGold(0)$
  - Axioms to generate new knowledge from percepts, e.g.,
    $\forall s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$
  - Axioms to generate actions (plans) from KB, e.g.,
    $\forall t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \;\Rightarrow\; Action(Grab, t)$
  - Axioms from knowledge to knowledge, e.g.,
    $\forall t \;\; AtGold(t) \wedge Action(Grab, t) \;\Rightarrow\; Holding(Gold, t+1)$

Representing the Wumpus World

Perception $\forall s, b, t \;\; Percept([s, b, Glitter], t) \Rightarrow AtGold(t)$

Representing the Wumpus World

Perception $\forall s, b, t \;\; Percept([s, b, Glitter], t) \Rightarrow AtGold(t)$

Location $At(Agent, s, t)$

Representing the Wumpus World

Perception $\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \Rightarrow AtGold(t)$

Location $At(Agent, s, t)$

Decision-making $\forall\, t \;\; AtGold(t) \Rightarrow Action(Grab, t)$

Representing the Wumpus World

Perception $\forall s, b, t \;\; Percept([s, b, Glitter], t) \Rightarrow AtGold(t)$

Location $At(Agent, s, t)$

Decision-making $\forall t \;\; AtGold(t) \Rightarrow Action(Grab, t)$

Internal reflection $\forall t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \Rightarrow Action(Grab, t)$, do we have gold already? (notice we cannot observe if we are holding gold, we need to track it)

Representing the Wumpus World

Adjacent squares

$$\forall x, y, a, b \;\; Adjacent([x, y], [a, b]) \quad \Leftrightarrow$$
$$(x = a \wedge (y = b - 1 \vee y = b + 1) \vee$$
$$(y = b \wedge (x = a - 1 \vee x = a + 1))$$

Representing the Wumpus World

Adjacent squares

$$\forall x, y, a, b \ \ Adjacent([x, y], [a, b]) \ \Leftrightarrow$$
$$(x = a \wedge (y = b - 1 \vee y = b + 1) \vee$$
$$(y = b \wedge (x = a - 1 \vee x = a + 1))$$

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

$$\forall s \ \ , Breezy(s) \ \Leftrightarrow \ \exists r \, (Adjacent(r, s) \wedge Pit(r))$$

Representing the Wumpus World

- We can go on and describe plans, causal rules, etc.
- But let's do some reasoning now

Facts and knowledge bases

'Joffrey Baratheon is a king'

Facts and knowledge bases

'Jon Snow is a person'

Facts and knowledge bases

'Jon Snow is a king'

Telling and Asking

$Tell(KB, King(Joffrey))$

Telling and Asking

$Tell(KB, King(Joffrey))$

$Tell(KB, Person(Jon))$

Telling and Asking

$Tell(KB, King(Joffrey))$

$Tell(KB, Person(Jon))$

$Tell(KB, \forall x \;\; King(x) \;\Rightarrow\; Person(x))$

Telling and Asking

$Tell(KB, King(Joffrey))$

$Tell(KB, Person(Jon))$

$Tell(KB, \forall x \;\; King(x) \Rightarrow Person(x))$

$Ask(KB, \exists x Person(x))$ is there a person?

Telling and Asking

$Tell(KB, King(Joffrey))$

$Tell(KB, Person(Jon))$

$Tell(KB, \forall x \ \ King(x) \ \Rightarrow \ Person(x))$

$Ask(KB, \exists x Person(x))$ is there a person?

$Askvar(KB, Person(x))$ who is a person?

Telling and Asking

$Tell(KB, King(Joffrey))$

$Tell(KB, Person(Jon))$

$Tell(KB, \forall x \;\; King(x) \;\Rightarrow\; Person(x))$

$Ask(KB, \exists x Person(x))$ is there a person?

$Askvar(KB, Person(x))$ who is a person?

Askvar returns a list of **substitutions**: $\{x/Joffrey\}, \{x/Jon\}$

Substitutions

Definition

Given a sentence S and a substitution σ,

Substitutions

Definition

Given a sentence S and a substitution σ,

$S\sigma$ denotes the result of plugging σ into S; e.g.,

Substitutions

Definition

Given a sentence S and a substitution σ,

$S\sigma$ denotes the result of plugging σ into S; e.g.,

$S = Smarter(x, y)$

Substitutions

Definition

Given a sentence S and a substitution σ,

$S\sigma$ denotes the result of plugging σ into S; e.g.,

$S = Smarter(x, y)$

$\sigma = \{x/Tyrion, y/Joffrey\}$

Substitutions

Definition

Given a sentence S and a substitution σ,

$S\sigma$ denotes the result of plugging σ into S; e.g.,

$S = Smarter(x, y)$

$\sigma = \{x/Tyrion, y/Joffrey\}$

$S\sigma = Smarter(Tyrion, Joffrey)$

Substitutions

Definition

Given a sentence S and a substitution σ,

$S\sigma$ denotes the result of plugging σ into S; e.g.,

$S = Smarter(x, y)$

$\sigma = \{x/Tyrion, y/Joffrey\}$

$S\sigma = Smarter(Tyrion, Joffrey)$

$Askvar(KB, S)$ returns some/all σ such that $KB \models S\sigma$

Unification

$$\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)$$

$$King(Joffrey)$$

$$\forall y \;\; Greedy(y)$$

Unification

$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)$$

$$King(Joffrey)$$

$$\forall y \;\; Greedy(y)$$

We can get the inference immediately if we can find a substitution matching the premises of the implication to the known facts.

Unification

$$\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)$$

$$King(Joffrey)$$

$$\forall y \;\; Greedy(y)$$

We can get the inference immediately if we can find a substitution matching the premises of the implication to the known facts.

$\theta = \{x/Joffrey, y/Joffrey\}$ works

Unification

$\textsc{Unify}(\alpha, \beta)$ returns θ if $\alpha\theta = \beta\theta$

Unification

$\text{UNIFY}(\alpha, \beta)$ returns θ if $\alpha\theta = \beta\theta$

p	q	θ
$Knows(Joffrey, x)$	$Knows(Joffrey, Sansa)$	
$Knows(Joffrey, x)$	$Knows(y, Sansa)$	
$Knows(Joffrey, x)$	$Knows(y, Mother(Joffrey))$	
$Knows(Jon, x)$	$Knows(x, Mother(Jon))$	

Unification

$\text{UNIFY}(\alpha, \beta)$ returns θ if $\alpha\theta = \beta\theta$

p	q	θ
$Knows(Joffrey, x)$	$Knows(Joffrey, Sansa)$	$\{x/Sansa\}$
$Knows(Joffrey, x)$	$Knows(y, Sansa)$	
$Knows(Joffrey, x)$	$Knows(y, Mother(Joffrey))$	
$Knows(Jon, x)$	$Knows(x, Mother(Jon))$	

Unification

$\text{UNIFY}(\alpha, \beta)$ returns θ if $\alpha\theta = \beta\theta$

p	q	θ
$Knows(Joffrey, x)$	$Knows(Joffrey, Sansa)$	$\{x/Sansa\}$
$Knows(Joffrey, x)$	$Knows(y, Sansa)$	$\{x/Sansa, y/Joffrey\}$
$Knows(Joffrey, x)$	$Knows(y, Mother(Joffrey))$	
$Knows(Jon, x)$	$Knows(x, Mother(Jon))$	

Unification

$\text{UNIFY}(\alpha, \beta)$ returns θ if $\alpha\theta = \beta\theta$

p	q	θ
$Knows(Joffrey, x)$	$Knows(Joffrey, Sansa)$	$\{x/Sansa\}$
$Knows(Joffrey, x)$	$Knows(y, Sansa)$	$\{x/Sansa, y/Joffrey\}$
$Knows(Joffrey, x)$	$Knows(y, Mother(Joffrey))$	$\{y/Joffrey, x/Mother(Joffrey)\}$
$Knows(Jon, x)$	$Knows(x, Mother(Jon))$	

Unification

$\text{UNIFY}(\alpha, \beta)$ returns θ if $\alpha\theta = \beta\theta$

p	q	θ
$Knows(Joffrey, x)$	$Knows(Joffrey, Sansa)$	$\{x/Sansa\}$
$Knows(Joffrey, x)$	$Knows(y, Sansa)$	$\{x/Sansa, y/Joffrey\}$
$Knows(Joffrey, x)$	$Knows(y, Mother(Joffrey))$	$\{y/Joffrey, x/Mother(Joffrey)\}$
$Knows(Jon, x)$	$Knows(x, Mother(Jon))$	$fail$

Standardising apart

$Knows(Jon, x)$ & $Knows(x, Mother(Jon))$ fails

Standardising apart

$Knows(Jon, x)$ & $Knows(x, Mother(Jon))$ fails

Standardising apart eliminates overlap of variables, e.g., $Knows(z_{17}, Mother(Jon))$

Generalized Modus Ponens (GMP)

Definite clause:
disjunction of literals, **exactly** one of which positive

Generalized Modus Ponens (GMP)

Definite clause:
disjunction of literals, **exactly** one of which positive
e.g., $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)$

Generalized Modus Ponens (GMP)

Definite clause:
disjunction of literals, **exactly** one of which positive
e.g., $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)$

$$\frac{p_1{}',\ \ p_2{}',\ \ldots,\ p_n{}',\ \ (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{q\theta}$$

where $p_i{}'\theta = p_i\theta$ for all i

Generalized Modus Ponens (GMP)

Definite clause:
disjunction of literals, **exactly** one of which positive
e.g., $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)$

$$\frac{p_1{}',\ \ p_2{}',\ \ldots,\ p_n{}',\ \ (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{q\theta} \qquad \text{where } p_i{}'\theta = p_i\theta \text{ for all } i$$

Assuming all variables are universally quantified...

Generalized Modus Ponens (GMP)

Definite clause:
disjunction of literals, **exactly** one of which positive
e.g., $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)$

$$\frac{p_1{}', \;\; p_2{}', \; \ldots, \; p_n{}', \;\; (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{q\theta} \qquad \text{where } p_i{}'\theta = p_i\theta \text{ for all } i$$

Assuming all variables are universally quantified...

$p_1{}'$ is $King(Joffrey)$
$p_2{}'$ is $Greedy(y)$
θ is $\{x/Joffrey, y/Joffrey\}$
$q\theta$ is $Evil(Joffrey)$

p_1 is $King(x)$
p_2 is $Greedy(x)$
q is $Evil(x)$

Soundness of GMP

Need to show that

$$p_1', \ \ldots, \ p_n', \ \ (p_1 \wedge \ldots \wedge p_n \Rightarrow q) \models q\theta$$

provided that $p_i'\theta = p_i\theta$ for all i
Lemma: If φ is definite clause, then $\varphi \models \varphi\theta$ by Universal Instantiation.

1. $(p_1 \wedge \ldots \wedge p_n \Rightarrow q) \models (p_1 \wedge \ldots \wedge p_n \Rightarrow q)\theta = (p_1\theta \wedge \ldots \wedge p_n\theta \Rightarrow q\theta)$
2. $p_1', \ \ldots, \ p_n' \models p_1' \wedge \ldots \wedge p_n' \models p_1'\theta \wedge \ldots \wedge p_n'\theta$
3. From 1 and 2, $q\theta$ follows by ordinary Modus Ponens

Today

- How to describe the world in logic
- Moving as a way to gather new facts
- Generalised modus ponens

Today

- How to describe the world in logic
- Moving as a way to gather new facts
- Generalised modus ponens

Today

- How to describe the world in logic
- Moving as a way to gather new facts
- Generalised modus ponens

Today

- How to describe the world in logic
- Moving as a way to gather new facts
- Generalise

Coming next

- Making sound inferences
- Walking forward from the assumptions
- Walking backwards from the conclusion