Lecture 13
Summary

TDT4136: Introduction to Artificial Intelligence

Xavier F. C. Sánchez Díaz, Ole Christian Eidheim, Pinar Øztürk

Department of Computer Science
Faculty of Information Technology and Electrical Engineering
Norwegian University of Science and Technology

November 14th, 2024

Exam
Information

- 4 hours exam
- Read first the cover/information page
- Multiple choice questions– choose the true/false answer
- Fill-in-the-blank– actually write a number into a box
- 20 questions
- Various weights. Also written in the cover/info page
- From all topics and chapters

Search

Summary

- Search can be **informed** or **uninformed**
  - BFS, DFS, UCS are **uninformed**
  - A* and GBFS are **informed** (they use a heuristic function: a *guess*)
- Remember the difference between graph search and tree search
- Study when the **goal-check** is performed on different algorithms!
- Learn the difference between **admissible** and **consistent** heuristics and what they imply
  - **Admissible**: $h(n) \leq h^*(n)$ for all n
  - **Consistent**: $h(n) \leq c(n, n') + h(n')$ for all n and its children n'
- No need to study iterated deepening search

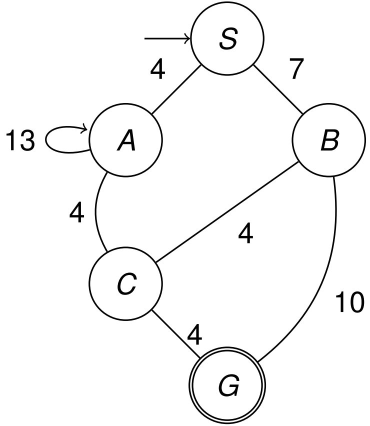
Search

Example question

What is the order of nodes expanded by `<ALGORITHM>` *on the following graph?*

With the following estimated distances to the goal:

- $h(A) = 3$
- $h(B) = 3$
- $h(C) = 3$
- $h(G) = 0$



What happens if $h(B) = x$*?*
What about $h(B) \geq y$*?.*

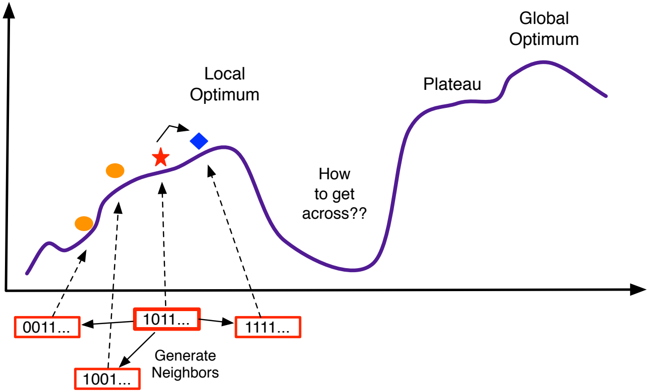
Remember to consider that nodes could be expanded more than once!

Local Search

Summary

Sometimes we only need to find a **good enough** solution, so we can search locally: go to the <u>best</u> spot you see now.

- Assume you are doing **maximisation**
- You then want to **climb** the tallest peak
- This is called **hill-climbing**!



If you are **minimising** instead, then the procedure is called **gradient descent** as we want to move towards the direction where the difference in “height” is largest.

Complex Environments (and a bit of planning)

Summary

- Learn **how to get across** valleys and plateaus in the search space landscape
- Consider what happens to our search (or our plan) when exploring **complex environments**
  - I cannot see (sensorless)
  - I can see but am not sure what will happen (contingency/uncertainty)
  - I need to keep an eye (online/real-time)
- Remember that states could become **belief states** instead!
- Instead of a single action between states, we may end up with a **function** that gives us the probability of reaching different states

Constraint Satisfaction Problems

- Backtracking search
- Inference: reducing domains; forward checking, AC-3
- Heuristics:
  - Selecting variable: choose the variable with the fewest remaining values (fail early)
  - Ordering values: prefer values that rule out the fewest choices for neighboring unassigned variables (keep your options open)
- Backjumping
- Local search: Min-Conflicts algorithm
- Tree-structured CSPs: linear time complexity

Adversarial Search

- Observable games: turn-taking, perfect information games
  - minimax, expectimax, expectiminimax
  - Alpha-beta pruning: do not visit nodes and branches you do not need to check
  - Heuristic minimax: depth limited search
  - Monte Carlo tree search

- Partially observable games: states not fully observable and/or players act simultaneously
  - Regret minimization
  - (Monte Carlo tree search)
  - Game theory (next slide): exact solutions

Game Theory

- Nash equilibrium
- Pure strategy Nash equilibrium
- Mixed strategy Nash equilibrium
- Repeated games:
  - Finite and known number of rounds
  - Infinite rounds
  - Finite but unknown number of rounds

Propositional Logic

- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Precedence of operators/connectives: $\neg$, $\wedge$, $\vee$, $\implies$, $\iff$.

Propositional Logic

- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Precedence of operators/connectives: $\neg$, $\wedge$, $\vee$, $\implies$, $\iff$.
- Inference: truth of a sentence given a KB, i.e. whether the KB entails the sentence
  - soundness of inference algorithm: produces for a given KB only the sentences entailed by the KB
  - important notions e.g., entailment, validity, satisfiability
  - model checking by truth table
  - Forward, backward chaining - uses Horn clauses
  - Resolution refutation, uses CNF sentences,

Remember propositional logic lacks expressive power.

First Order Logic

- More expressiveness,
  - can express partial information, e.g., "there exist a person in the room" without needing to tell which concrete person/object.
  - objects and relations between them, properties of objects
  - uses quantifications and variables for such expressive.
- Figure 8.3 in the textbook shows what atomic/complex sentence, term, constant, predicate, function, etc means

First Order Logic

- More expressiveness,
  - can express partial information, e.g., "there exist a person in the room" without needing to tell which concrete person/object.
  - objects and relations between them, properties of objects
  - uses quantifications and variables for such expressive.
- Figure 8.3 in the textbook shows what atomic/complex sentence, term, constant, predicate, function, etc means
- Typical wrong use of universial and existential quantifications.
  - Correct use (mostly): Universal/all quantifiers dominantly with $\rightarrow$,
  - Existential quant. with $\wedge$.

First Order Logic

- More expressiveness,
  - can express partial information, e.g., "there exist a person in the room" without needing to tell which concrete person/object.
  - objects and relations between them, properties of objects
  - uses quantifications and variables for such expressive.
- Figure 8.3 in the textbook shows what atomic/complex sentence, term, constant, predicate, function, etc means
- Typical wrong use of universial and existential quantifications.
  - Correct use (mostly): Universal/all quantifiers dominantly with $\rightarrow$,
  - Existential quant. with $\wedge$.
- Relationship between Universal and Existential Quantifiers
- Nested Quantifiers

Inference in First order Logic

- Inference rules for quantifiers
  - Universal Instantiation - substitution of variables with ground terms (i.e, terms without variables)
  - Existential Instantiation - a variable is substituted with a unique(not seen/used before) constant/fn, skolem constant/fn
- The use of Quantifiers and variables require some extra work/syntactical operations (e.g., when converting to CNF (for resolution) and to Definite clausal form (for Backward Chaining) such as unification, skolemization for existential quantifiers, standardization of variable names

Inference in First order Logic

- Inference rules for quantifiers
  - Universal Instantiation - substitution of variables with ground terms (i.e, terms without variables)
  - Existential Instantiation - a variable is substituted with a unique(not seen/used before) constant/fn, skolem constant/fn
- The use of Quantifiers and variables require some extra work/syntactical operations (e.g., when converting to CNF (for resolution) and to Definite clausal form (for Backward Chaining) such as unification, skolemization for existential quantifiers, standardization of variable names
- Inference through reduction to propositional inference - propositionalize and use propositional logic inference. Not efficient.
- Inference without propositionalization. Using "generalized" versions of the inference rules in PropLogic,
  - e.g., generalized/lifted modus ponens. A sound inference.
  - involves Substitution. Advantage of inference without propositionalization: makes only substitution required for particular inferences required for solving the problem, not all possible substitutions

Inference in First order Logic

- Inference rules for quantifiers
  - Universal Instantiation - substitution of variables with ground terms (i.e, terms without variables)
  - Existential Instantiation - a variable is substituted with a unique(not seen/used before) constant/fn, skolem constant/fn
- The use of Quantifiers and variables require some extra work/syntactical operations (e.g., when converting to CNF (for resolution) and to Definite clausal form (for Backward Chaining) such as unification, skolemization for existential quantifiers, standardization of variable names
- Inference through reduction to propositional inference - propositionalize and use propositional logic inference. Not efficient.
- Inference without propositionalization. Using "generalized" versions of the inference rules in PropLogic,
  - e.g., generalized/lifted modus ponens. A sound inference.
  - involves Substitution. Advantage of inference without propositionalization: makes only substitution required for particular inferences required for solving the problem, not all possible substitutions
- Inference with Backward/Forward chaining
- inference through Resolution Refutation

Example 1

Somebody stole the Christmas lights of Mr. Olsen in Trondheim. Inspector Harry Hole thinks one of the five famous burglars, Arne, Bernt, Cristoffer, Dina, Edna, in the town must have stolen the lights. He interviews each of them to find out the guilty one.

The following is the result of the interviews – two statements from each thief. It is well known that exactly one of the two statements of each thief is a lie:

Arne: *It was not Edna. It was Bernt.*

Bernt: *It was not Cristoffer. It was not Edna.*

Cristoffer: *It was Edna. It was not Arne.*

Derek: *It was Cristoffer. It was Bernt.*

Edna: *It was Dina. It was not Arne.*

Use the following propositional variables:

A=It was Arne. **B**=It was Bernt. **C**=It was Cristoffer. **D**=It was Dina. **E**=It was Edna

Example 2

1. Translate the evidence from each thief (i.e. from the statements of thieves taking also into account that exactly one of two statements of each statement of each thief is a lie) into propostional logic representation . i.e.,

   Arne: ……..

   Bernt:….

   ..

   ..

   Edna: …..

2. It is well known that one thief stole the lights alone. Represent this information as a set of implications.
3. Your data base consists of the statements you wrote in 1) and 2). Convert the statements in the KB to conjunctive normal form.
4. Using this KB, apply resolution refutation in order to infer “It was Cristoffer”. If you cannot, then you must have made a mistake somewhere. Use the following structure to show your proof – add as many lines as you need:

   Resolve …………… and ……………… to produce ………………

   Resolve …………… and ……………… to produce ………………

   …………..

Example 3

1) Translate the evidence (i.e. from the statements of thieves taking also into account that exactly one of the two statements of each thief is a lie.

Arne: $(E \land B) \lor (\neg E \land \neg B)$

Bernt: $(C \land \neg E) \lor (\neg C \land E)$

Cristoffer: $(\neg E \land \neg A) \lor (E \land A)$

Dina: $(\neg C \land B) \lor (C \land \neg B)$

Edna: $(\neg D \land \neg A) \lor (D \land A)$

Example 4

2) Representation of the information that one thief has stolen the lights alone:

$$(A \Rightarrow \neg B \wedge \neg C \wedge \neg D \wedge \neg E)$$
$$(B \Rightarrow \neg A \wedge \neg C \wedge \neg D \wedge \neg E)$$
$$(C \Rightarrow \neg A \wedge \neg B \wedge \neg D \wedge \neg E)$$
$$(D \Rightarrow \neg A \wedge \neg B \wedge \neg C \wedge \neg E)$$
$$(E \Rightarrow \neg A \wedge \neg B \wedge \neg C \wedge \neg D)$$

Example 5

3) The KB:

1. $(E \wedge B) \vee (\neg E \wedge \neg B)$
2. $(C \wedge \neg E) \vee (\neg C \wedge E)$
3. $(\neg E \wedge \neg A) \vee (E \wedge A)$
4. $(\neg C \wedge B) \vee (C \wedge \neg B)$
5. $(\neg D \wedge \neg A) \vee (D \wedge A)$
6. $(A \Rightarrow \neg B \wedge \neg C \wedge \neg D \wedge \neg E)$
7. $(B \Rightarrow \neg A \wedge \neg C \wedge \neg D \wedge \neg E)$
8. $(C \Rightarrow \neg A \wedge \neg B \wedge \neg D \wedge \neg E)$
9. $(D \Rightarrow \neg A \wedge \neg B \wedge \neg C \wedge \neg E)$
10. $(E \Rightarrow \neg A \wedge \neg B \wedge \neg C \wedge \neg D)$

Converted to CNF:

$(E \vee \neg B) (\neg E \vee B)$ (using distributivity of $\vee$ over ^ on 1.)
$(C \vee E) (\neg C \vee \neg E)$
$(\neg E \vee A) (E \vee \neg A)$
$(\neg C \vee \neg B) (C \vee B)$
$(\neg D \vee A) (D \vee \neg A)$
$(\neg A \vee \neg B) (\neg A \vee \neg C)$ (implic. elim. +distributivity for 6-10)
$(\neg A \vee \neg D) (\neg A \vee \neg E)$
$(\neg B \vee \neg C) (\neg B \vee \neg D)$
$(\neg B \vee \neg E) (\neg C \vee \neg D)$
$(\neg C \vee \neg E) (\neg D \vee \neg E)$

Example 6

4) Add ¬C to the KB and obtain { }

Resolve ¬B ∨ ¬E and C ∨ E to produce ¬B ∨ C

Resolve ¬B ∨ C and C ∨ B to produce C ∨ C = C

Resolve C and ¬C to produce { }

t is possible to do it in other ways as well. For example:

Resolve ¬ C and C ∨ E to give E

Resolve ¬ C and C ∨ B to give B

Resolve E and ¬ B ∨ ¬ E to give ¬ B

Resolve B and ¬ B to give {}

Converted to CNF:

(E ∨ ¬B) (¬E ∨ B)
(C ∨ E) (¬C ∨ ¬E)
(¬E ∨ A) (E ∨ ¬A)
(¬C ∨ ¬B) (C ∨ B)
(¬D ∨ A) (D ∨ ¬A)
(¬A ∨ ¬B) (¬A ∨ ¬C)
(¬A ∨ ¬D) (¬A ∨ ¬E)
(¬B ∨ ¬C) (¬B ∨ ¬D)
(¬B ∨ ¬E) (¬C ∨ ¬D)
(¬C ∨ ¬E) (¬D ∨ ¬E)

Planning

Summary

Planning is the process of finding a sequence of actions that transforms the world from an initial state to a goal state.

- Remember how to represent both **actions** and **states** using PDDL

$$
\begin{aligned}
&\mathit{Action}(\mathit{Move}(\mathit{who}, \mathit{from}, \mathit{to})\\
&\qquad \mathit{Precond}: \mathit{At}(\mathit{who}, \mathit{from}) \wedge \mathit{Adj}(\mathit{from}, \mathit{to}) \wedge \neg \mathit{Pit}(\mathit{to})\\
&\qquad \mathit{Effect}: \neg \mathit{At}(\mathit{who}, \mathit{from}) \wedge \mathit{At}(\mathit{who}, \mathit{to}))
\end{aligned}
$$

- *Move* is the action being defined
- *who*, *from* and *to* are variables
- *Precond* describes the state of the world needed for the action *to occur*
- *Effect* describes the *resulting* state after acting

States in PDDL
Summary

- The world is closed—any fluents not mentioned are *False*
- Unique names—different literals refer to different entities (like *cellA*1 and *cellA*2)
- **No** uncertain or **negated** literals

As with search, we also need **starting** and **goal** states.