CSC384 – Final Exam Review

Summer 2019

General Course Information

Homework / Midterm:

- A3 Marks posted, please submit remark request by Thursday.
- A4 Marks posted to MarkUs by Wednesday, we hope.
- A4 Modules will remain up online for studying.
- Quizzes can be picked up in Bahen 3219 from 10-12 and 2-4pm.

General Exam Information

We'll be updating the CSC384 web page **Test** tab at the top of the page with exam information and also posting information on piazza.

Timing of Exam: Thursday, August 15th, at 2 pm, in Room 200 of the Exam Centre (255 McCaul Street).

Remaining Pre-Exam Help Session:

- Wednesday, August 14, 5 - 6 pm (With Zhewei; Bahen 3201; Topic Reasoning with Uncertainty)

Any changes to help sessions will be posted on piazza and on the "Test" web page

Exam Resources

- Old exams in library
- Problem sets from midterm etc.
- A4 problem sets online
- Brachman & Levesque KR book (see link on piazza "KR Additional Readings" post)
- Russell & Norvig textbook
- Course slides and other posted materials

Tips and Resources

- Let the **lecture slides and the posted tutorial materials** be your guide for studying. If you're unclear about something, augment it with the text or other online materials and/or come to an office hour!
- Make sure you *understand* the material. If there are proofs, work through them so you understand them. Understand the rationale for why things work the way they do.
- Work through some problem sets. Look at the **posted sample problems on the test web page** as well as problems we went through in class on the board, and old exams in the library.
- Know and understand the facts: know the complexity of different algorithms and why. Know the axioms of probability and understand how to apply.

General Exam Information

About the exam

- 3 hours in duration
- no aids permitted
- worth 40% of your course grade
- You must get 40% on the exam to pass the course

Approximate distribution of marks on exam:

- Game Tree and Uninformed & Heuristic Search: 8%
- CSP: 12%
- KR: 40%
- Uncertainty: 40%
- Format will be short answer, followed by problem solving.
- NO local search material on the exam.

Today

→ KR Example

- CSP Example

- BN Example

  See posting on Lectures slides page entitled

  “Solution (and additional d-separation examples)”

  http://www.teach.cs.toronto.edu/~csc384h/summer/Lectures/Bayes-Net-example-solved.pdf

Resolution Example

One of Gordon Novak's problems (posted on our Lectures page)

Consider the following axioms:

1. All hounds howl at night.
2. Anyone who has any cats will not have any mice.
3. Light sleepers do not have anything which howls at night.
4. John has either a cat or a hound.
5. (Conclusion) If John is a light sleeper, then John does not have any mice.

TASK: Prove the conclusion using resolution refutation

Resolution Example

Consider the following axioms:

1. All hounds howl at night.
2. Anyone who has any cats will not have any mice.
3. Light sleepers do not have anything which howls at night.
4. John has either a cat or a hound.
5. (Conclusion) If John is a light sleeper, then John does not have any mice.

The above English sentences can be written as the following first-order logic sentences.

1. $\forall x\,(HOUND(x) \rightarrow HOWL(x))$
2. $\forall x\ \forall y\,(HAVE\,(x,y) \wedge CAT\,(y) \rightarrow \neg \exists z\,(HAVE(x,z) \wedge MOUSE\,(z)))$
3. $\forall x\,(LS(x) \rightarrow \neg \exists y\,(HAVE\,(x,y) \wedge HOWL(y)))$
4. $\exists x\,(HAVE\,(John,x) \wedge (CAT(x) \vee HOUND(x)))$
5. $LS(John) \rightarrow \neg \exists z\,(HAVE(John,z) \wedge MOUSE(z))$

Resolution Example

1. $\forall x\, (HOUND(x) \rightarrow HOWL(x))$

2. $\forall x\, \forall y\, (HAVE\,(x,y) \wedge CAT\,(y) \rightarrow \neg\, \exists z\, (HAVE(x,z) \wedge MOUSE\,(z)))$

3. $\forall x\, (LS(x) \rightarrow \neg\, \exists y\, (HAVE\,(x,y) \wedge HOWL(y)))$

4. $\exists x\, (HAVE\,(John,x) \wedge (CAT(x) \vee HOUND(x)))$

5. $LS(John) \rightarrow \neg\, \exists z\, (HAVE(John,z) \wedge MOUSE(z))$

Convert to **Clausal Form** using the 8 steps we learned in class, yielding the following:

1. (¬Hound(x), Howl(x))
2. (¬Have(x,y), ¬Cat(y), ¬Have(x,z), ¬Mouse(z))
3. (¬LS(x), ¬Have(x,y), ¬Howl(y))
4. (Have(John,a))
5. (Cat(a), Hound(a))
6. (LS(John))
7. (Have(John,b))
8. (Mouse(b))

Clause 1-3 correspond to Formulas 1-3, Clause 4&5 correspond to Formula 4, and Clauses 6-8 correspond to the **negation of Formula 5 (i.e., the negated query)**

Resolution Example

Perform resolution refutation with these clauses to derive the empty clause:

1. $(\neg Hound(x), Howl(x))$
2. $(\neg Have(x,y), \neg Cat(y), \neg Have(x,z), \neg Mouse(z))$
3. $(\neg LS(x), \neg Have(x,y), \neg Howl(y))$
4. $(Have(John,a))$
5. $(Cat(a), Hound(a))$
6. $(LS(John))$
7. $(Have(John,b))$
8. $(Mouse(b))$

General Tips:

- Remember that each of these clauses is universally quantified from the outside using the variables that are contained in the clause. As such the "x" in clause 1 is different from the "x" in clauses, 2, 3, and 4. So if you're resolving two clauses that each have a variable of the same name (e.g., "x") after application of the MGU, you might wish to rename one of the "x"s to "z" or some other variable.
- A rule of thumb is to use the negated query or clauses derived from the negated query in your proof, since that negated query is the source of the inconsistency that will lead to the empty clause.

Resolution Example

Perform resolution refutation with these clauses to derive the empty clause:

1. (¬Hound(x), Howl(x))
2. (¬Have(x,y), ¬Cat(y), ¬Have(x,z), ¬Mouse(z))
3. (¬LS(x), ¬Have(x,y), ¬Howl(y))
4. (Have(John,a))
5. (Cat(a), Hound(a))
6. (LS(John))
7. (Have(John,b))
8. (Mouse(b))

We start the resolution with one of Clauses 6,7,8 which correspond to the negated query.

9.	R(2d,8a){z=b}	(¬Have(x,y), ¬Cat(y), ¬Have(x,b))
10.	R(9c,7a){x=John}	(¬Have(John,y), ¬Cat(y))
11.	R(10b,5a){y=a}	(¬Have(John,a), Hound(a))
12.	R(11b,1a){x=a}	(¬Have(John,a), Howl(a))
13.	R(12b,3c){y=a}	(¬LS(x), ¬Have(x,a), ¬Have(John,a))
14.	R(13a,6){x=John}	(¬Have(John,a))
15.	R(14,4)	() ← empty clause QED

Today

- KR Example

→ CSP Example

- VE Example

See posting on Lectures slides page entitled

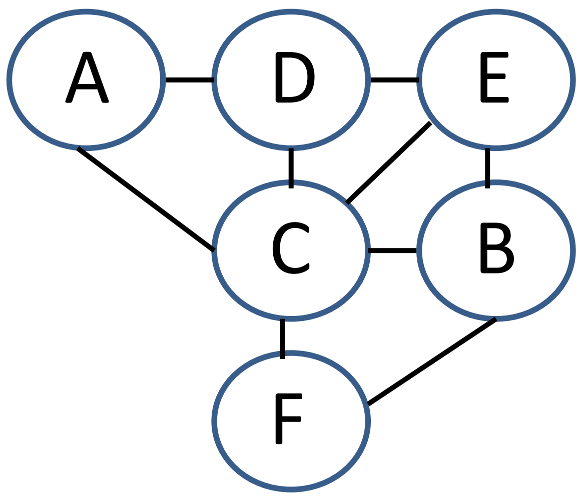
“Solution (and additional d-separation examples)”

http://www.teach.cs.toronto.edu/~csc384h/summer/Lectures/Bayes-Net-example-solved.pdf

CSP Example

CSC384 Test-Colour Assignment Problem

- 6 Students: **A**lice, **B**ob, **C**arol, **D**on, **E**lla, **F**red ….{A,B,C,D,E}
- 3 different colours of tests: pink, green, blue ….{p,g,b}
- Students are sitting in seats in the following configuration:



Problem: Assign coloured tests to students so that students cannot look over and see another student's test of the same colour.

CSP Example

- 6 Students: **A**lice, **B**ob, **C**arol, **D**on, **E**lla, **F**red ….{A,B,C,D,E}
- 3 different colours of test: pink, green, blue ….{p,g,b}

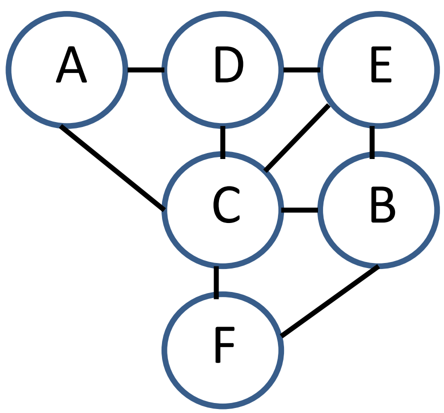
Constraints

$A \neq D$, $A \neq C$

$B \neq E$, $B \neq F$, $B \neq C$

$C \neq D$, $C \neq E$, $C \neq F$

$D \neq E$



Initial Domains of Variables

Dom(A) = {p,g,b}

Dom(B) = {p,g,b}

Dom(C) = {p,g,b}

Dom(D) = {p,g,b}

Dom(E) = {p,g,b}

Dom(F) = {p,g,b}

Solve this CSP using **Forward Checking.**
Use a **FIXED variable order: A,B,C,D,E,F**
And a **FIXED assignment order: p,g,b**

CSP Example

Solve this CSP using **Forward Checking.**

Use a **FIXED variable order: A,B,C,D,E,F**

And a **FIXED assignment order: p,g,b**

Draw the search tree

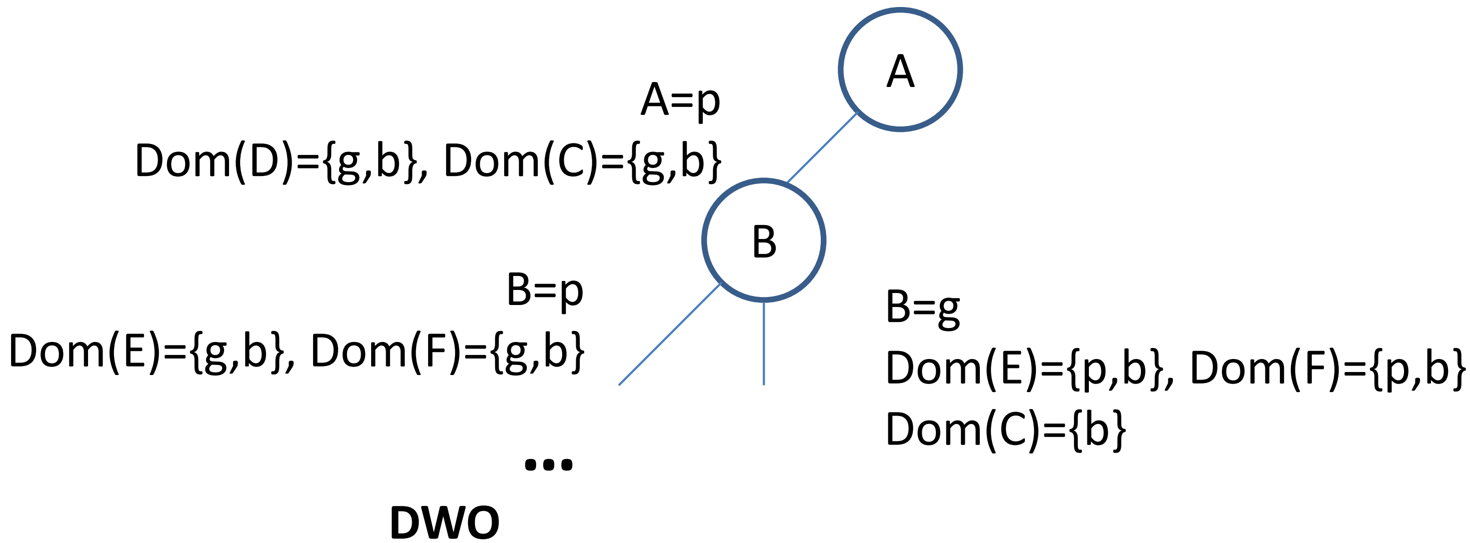
At each node of the search tree indicate

- The variable being instantiated, and the value it is being assigned
- A list of the variables that have had at least one of their values pruned by the new assignment and a list of remaining legal values for each of these variables.
- Mark any DWOs

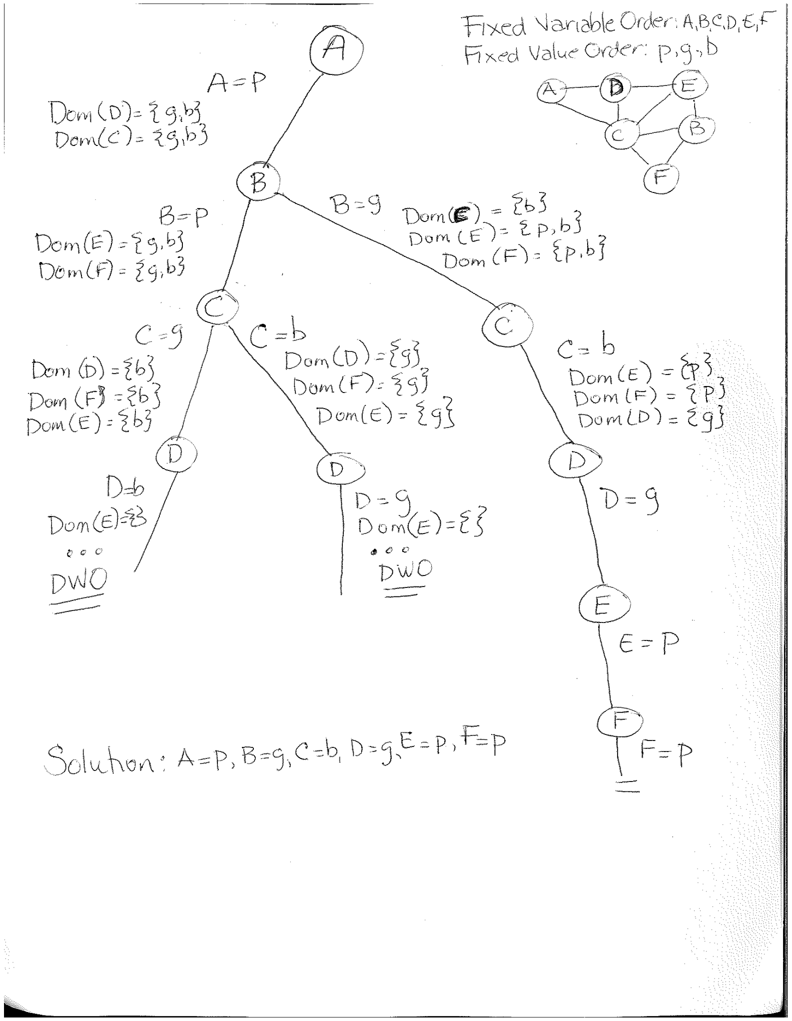
CSP Example

Draw the search tree.

Remember that you're searching in the space of partial assignments to the variables. According to the fixed variable ordering defined in this problem (which we added to make the example illustrate some points), we start with variable A. We can assign A any value in its domain, i.e., any of p,g, or b. We start with p and then apply forward checking and in so doing prune the values in the domain of D and C to remove "p". We proceed following the specified variable ordering.



CSP Example



- Note that as variables are assigned, forward checking prunes the domains of variables.
- At the point of the first DWO, D only had b in its domain. As such the algorithm backtracks to C and tries the assignment C=b, since Dom(C)={g,b}.
- Observe that upon backtracking, the domains of variables are reinstated to their values at the level of backtracking.

Today

- KR Example

- CSP Example

→ VE Example

See posting on Lectures slides page entitled

"Solution (and additional d-separation examples)"

http://www.teach.cs.toronto.edu/~csc384h/summer/Lectures/Bayes-Net-example-solved.pdf