yz709-logic-sup3

Decision Procedures and SMT Solvers

Question 1

1. In Fourier-Motzkin Variable Elimination, any variable not bounded both above and below is deleted from the problem. For example, given the set of constraints:

$$3x \geq y \quad x \geq 0 \quad y \geq z \quad z \leq 1 \quad z \geq 0$$

the variables x and y can be removed (with their constraints), reducing the problem to $z \leq 1 \wedge z \geq 0$. Explain how this happens and why it is correct. (Ex 39)

```
(* eliminate y *)
3x >= y -> y <= 3x
x >= 0  -> y unconstrained
y >= z  -> y >= z
z <= 1  -> y unconstrained
z >= 0  -> y unconstrained

L1 = z, U1 = 3x
-> L1 <= U1
-> z <= 3x
```

the point is that you don't even need to do these steps!

```
(* eliminate x *)
x >= 0  -> x >= 0
z <= 1  -> x unconstrained
z >= 0  -> x unconstrained
z <= 3x -> x >= z/3

no upper bounds for x, so variable x is deleted
hence we have,
z <= 1 ⋀ z >= 0, which is a contradiction
```

- The process is correct because every time we eliminate a variable from the system, we preserve the inequalities that need to be satisfied for that variable in the system. So once we reach a system with only one variable, if there are no contradictions, then we can trivially find suitable values for all variables; otherwise, there are no such set of values that suit all variables, we have refuted the system. This process is linear, and since every step of eliminating a variable preserves correctness, the whole process preserves correctness by induction.

💡 Comments: This considers the elimination process, but doesn't address the issue of no upper/lower bound. Lets discuss!

Question 2

2. Apply Fourier Motskin variable elimination to the set of constraints: (Ex 40).

$$x \geq z \quad y \geq 2z \quad z \geq 0 \quad x + y \leq z$$

```
(* eliminate y *)
x >= z      -> y unconstrained
y >= 2z     -> y >= 2z
z >= 0      -> y unconstrained
x + y <= z  -> y <= z - x

L1 = 2z, U1 = z - x
-> L1 <= U1
-> 2z <= z - x
-> z <= -x

(* eliminate x *)
x >= z      -> x >= z
z >= 0      -> x unconstrained
z <= -x     -> x <= -z

L1 = z, U1 = -z
-> L1 <= U1
-> z <= -z
```

```
-> z <= 0

(* solve the case with one variable z *)
z >= 0 ∧ z <= 0
-> z = 0, x = 0, y = 0
```

At this point you could simply say satisfiable for z=0 (don't give values to x & y as there are many options)

💡 Comments:

Question 3

3. Apply the SMT algorithm sketched in section 9.4/9.5 of the notes (or slides 909 - 912), to the following set of clauses: (Ex 42)

$$\{c = 0, c > 0\} \quad \{a \neq b\} \quad \{c < 0, a = b\}$$

Good note!

```
(* in DPLL, we write a ≠ b as an atomic formula ¬(a = b) *)
{c = 0, c > 0} {¬(a = b)} {c < 0, a = b}
-> {c = 0, c > 0} {c < 0} (* unit propogation using ¬(a = b) *)
-> {c = 0, c > 0} (* unit propagation using (c < 0) *)
-> (* case split on (c = 0) *)
① c = 0: pick a model ¬(a = b) ∧ (c < 0) ∧ (c = 0)
          (* arithmetic decision procedure: finds contradiction,
              - since (c < 0) contradicts with (c = 0)
           *)
          -> return a new clause {(a = b), ¬(c < 0), ¬(c = 0)}
② ¬(c = 0): pick a model ¬(a = b) ∧ (c < 0) ∧ (c > 0)
          (* arithmetic decision procedure: finds contradiction,
              - since (c < 0) contradicts with (c > 0)
           *)
          -> return a new clause {(a = b), ¬(c < 0), ¬(c > 0)}
hence we have refuted the original set of clauses, they are not satisfiable
```

(But can we simplify new clause?)

💡 Comments: Nice, clear presentation.

Binary Decision Diagrams

Question 1

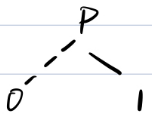
1. Compute the BDD's for the following formulas, taking the variables as alphabetically ordered: (Ex 43)

   (a) $P \wedge Q \rightarrow Q \wedge P$
   (b) $\neg(P \vee Q) \vee P$

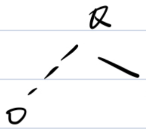
Q1

(a) $P \wedge Q \rightarrow Q \wedge P$

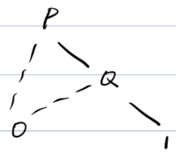
BDD for P:



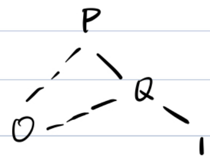
BDD for Q:



BDD for $P \wedge Q$:



BDD for $Q \wedge P$:



BDD for $P \wedge Q \rightarrow Q \wedge P$:

it is a tautology

Which is written as

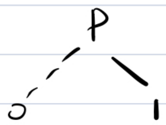
$\boxed{1}$

And as its a tautology (obviously!) $\Rightarrow$ no working required.

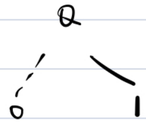
(b)

$\neg(P \vee Q) \vee P$

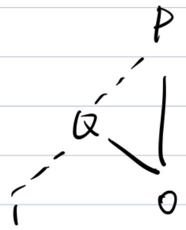
BDD for P :



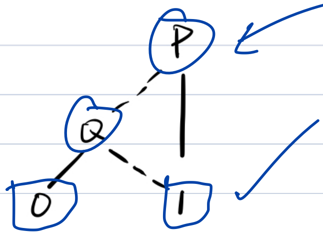
BDD for Q



BDD for $\neg(P \vee Q)$:



BDD for $\neg(P \vee Q) \vee P$:



boxes/circles just makes nodes clearer.

Comments: Excellent, clear working.

Question 2

2. Verify these equivalences using BDD's. (Ex 44/45)

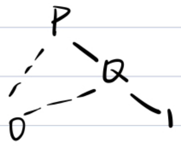
   (a) $(P \wedge Q) \wedge R \simeq P \wedge (Q \wedge R)$

   (b) $(P \vee Q) \rightarrow R \simeq (P \rightarrow R) \wedge (Q \rightarrow R)$

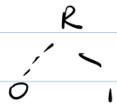
Q2 Under the same alphabetical order

(a) $(P \wedge Q) \wedge R$

BDD for $P \wedge Q$:

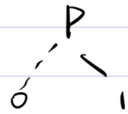


BDD for R:

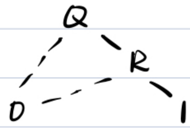


$P \wedge (Q \wedge R)$

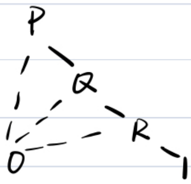
BDD for P:



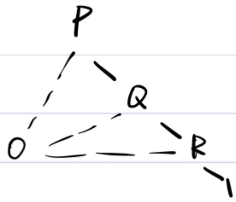
BDD for $Q \wedge R$:



BDD for $(P \wedge Q) \wedge R$:



BDD for $P \wedge (Q \wedge R)$:

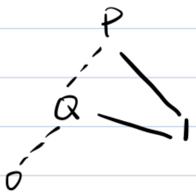


Since they produce the same BDDs, they are equivalent.

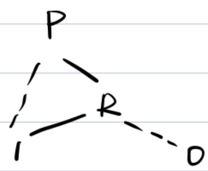
(b) $P \vee Q \to R$

$(P \to R) \wedge (Q \to R)$

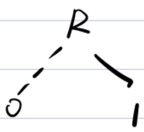
BDD for $P \vee Q$:



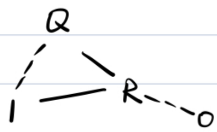
BDD for $P \to R$:



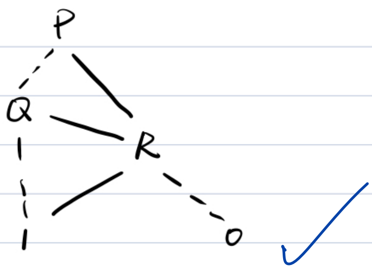
BDD for R:



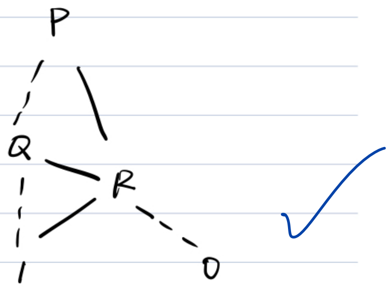
BDD for $Q \to R$:



BDD for $(P \vee Q) \to R$:



BDD for $(P \to R) \wedge (Q \to R)$:



since they produce the same BDDs, they are equivalent.

💡 Comments:

Modal Logics

Question 1

1. Prove the equivalence $\Box(A \wedge B) \simeq \Box A \wedge \Box B$ (Ex 49)

Q1

$\Box(A\wedge B) \Rightarrow \Box A \wedge \Box B$

$$\dfrac{\dfrac{\dfrac{\dfrac{A,B\Rightarrow A}{A\wedge B\Rightarrow A}\,(\wedge l)}{\Box(A\wedge B)\Rightarrow A}\,(\Box l)}{\Box(A\wedge B)\Rightarrow \Box A}\,(\Box r) \qquad \dfrac{\dfrac{\dfrac{A,B\Rightarrow B}{A\wedge B\Rightarrow B}\,(\wedge l)}{\Box(A\wedge B)\Rightarrow B}\,(\Box l)}{\Box(A\wedge B)\Rightarrow \Box B}\,(\Box r)}{\Box(A\wedge B)\Rightarrow \Box A\wedge\Box B}\,(\wedge r)$$

$\Box A\wedge\Box B \Rightarrow \Box(A\wedge B)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A,B\Rightarrow A \qquad A,B\Rightarrow B}{A,B\Rightarrow A\wedge B}\,(\wedge r)}{A,\Box B\Rightarrow A\wedge B}\,(\Box l)}{\Box A,\Box B\Rightarrow A\wedge B}\,(\Box l)}{\Box A,\Box B\Rightarrow \Box(A\wedge B)}\,(\Box r)}{\Box A\wedge\Box B\Rightarrow \Box(A\wedge B)}\,(\wedge l)$$

Comments:

Question 2

2. **Optional**: Prove the sequent: $\Diamond(A \to B), \Box A \Rightarrow \Diamond B$ (Ex 48).

Q2

$$\dfrac{\dfrac{\dfrac{A\Rightarrow A}{\Box A\Rightarrow A\ \Diamond B}\,(\Box l) \qquad \dfrac{\dfrac{B,A\Rightarrow B}{B,A\Rightarrow \Diamond B}\,(\Diamond r)}{B,\Box A\Rightarrow \Diamond B}\,(\Box l)}{A\to B,\Box A\Rightarrow \Diamond B}\,(\to l)}{\Diamond(A\to B),\Box A\Rightarrow \Diamond B}\,(\Diamond l)$$

working issues

no point doesn't help reach basic

Comments:

Question 3

3. Prove: $\Box\Diamond\Box A, \Box\Diamond\Box B \Rightarrow \Box\Diamond\Box(A \wedge B)$. (Ex 50).

Q3

$$A, B \Rightarrow A \qquad A, B \Rightarrow B$$

$$A, B \Rightarrow A \land B \quad (\land r)$$

$$A, \Box B \Rightarrow A \land B \quad (\Box l)$$

$$\Box A, \Box B \Rightarrow A \land B \quad (\Box l)$$

$$\Box A, \Box B \Rightarrow \Box(A \land B) \quad (\Box r)$$

$$\Box A, \Box B \Rightarrow \Diamond\Box(A \land B) \quad (\Diamond r)$$

$$\Box A, \Diamond\Box B \Rightarrow \Diamond\Box(A \land B) \quad (\Diamond l) \; ✗$$

$$\Diamond\Box A, \Diamond\Box B \Rightarrow \Diamond\Box(A \land B) \quad (\Diamond l)$$

$$\Box\Diamond\Box A, \Box\Diamond\Box B \Rightarrow \Diamond\Box(A \land B) \quad (\Box l)$$

$$\Box\Diamond\Box A, \Box\Diamond\Box B \Rightarrow \Box\Diamond\Box(A \land B) \quad (\Box r) \; ✓$$

this would have deleted $\Box\Diamond B$ term

should only remove one $\Box$ i.e. you should still have one of $\Box\Diamond\Box B$ or $\Box\Diamond\Box A$.

💡 Comments: I think there are some issues applying the "tricky" rules here. Lets go over.

Tableaux-Based Methods

Question 1

1. Use the free variable tableau calculus to prove this formula: $(\exists y\, \forall x\, R(x, y)) \to (\forall x\, \exists y\, R(x, y))$ (Ex 51)

Q1

negate the goal formula & convert into NNF:

$$(\exists y \forall x R(x,y)) \land \neg(\forall x \exists y R(x,y))$$

NNF $$\simeq (\exists y \forall x R(x,y)) \land (\exists x \forall y \neg R(x,y))$$

skolemisation:

$$\simeq \forall x R(x,a) \land \forall y \neg R(b,y)$$

We unify last in Tableaux calc.

$w \mapsto b \;, v \mapsto a$

$$R(\cancel{b}^{w},a), \neg R(b,\cancel{a}^{v}) \Rightarrow$$

$(\forall l)$ with $[a/y]$

$$R(\cancel{b}^{w},a), \forall y \neg R(b,y) \Rightarrow$$

$(\forall l)$ with $[b/x]$

$$\forall x R(x,a), \forall y \neg R(b,y) \Rightarrow$$

$(\land l)$

$$\forall x R(x,a) \land \forall y \neg R(b,y) \Rightarrow$$

Comments: Good reasoning/presentation overall!

Question 2

2. Compare the sequent calculus, the free variable tableau calculus and resolution by using each of them to prove the following formula:

$$(P(a,b) \lor \exists z\, P(z,z)) \to \exists xy\, P(x,y)$$

Q2

Using sequent calculus:

$$(P(a,b) \vee \exists z P(z,z)) \rightarrow \exists xy P(x,y)$$

negate the formula:

$$(P(a,b) \vee \exists z P(z,z)) \wedge \neg(\exists xy P(x,y))$$
$$\simeq (P(a,b) \vee \exists z P(z,z)) \wedge \forall xy \neg P(x,y)$$

don't do this for sequent calculus!

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(a,b) \Rightarrow P(a,b)}}{P(a,b), \neg P(a,b) \Rightarrow}(\neg l)}{P(a,b), \forall y \neg P(a,y) \Rightarrow}(\forall l)}{P(a,b), \forall xy \neg P(x,y) \Rightarrow}(\forall l) \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(z,z) \Rightarrow P(z,z)}}{P(z,z), \neg P(z,z) \Rightarrow}(\neg l)}{P(z,z), \forall y \neg P(z,y) \Rightarrow}(\forall l)}{P(z,z), \forall xy \neg P(x,y) \Rightarrow}(\forall l)}{\exists z P(z,z), \forall xy \neg P(x,y) \Rightarrow}(\exists l)}{\dfrac{(P(a,b) \vee \exists z P(z,z)), \forall xy \neg P(x,y) \Rightarrow}{(P(a,b) \vee \exists z P(z,z)) \wedge \forall xy \neg P(x,y) \Rightarrow}(\wedge l)}(\vee l)$$

Steps are simpler w/ pure sequent calculus

Using free variable tableau calculus

$(p(a,b) \vee \exists z P(z,z)) \to \exists xy P(x,y)$

negate the formula:

$(p(a,b) \vee \exists z p(z,z)) \wedge \neg(\exists xy P(x,y))$

$\simeq (p(a,b) \vee \exists z p(z,z)) \wedge \forall xy \neg P(x,y)$

$\xrightarrow{\text{skolemisation}} (p(a,b) \vee p(c,c)) \wedge \forall xy \neg p(x,y) \Rightarrow$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p(a,b), \neg p(a,b) \Rightarrow}}{p(a,b), \forall y \neg p(a,y) \Rightarrow}(\forall l)}{p(a,b), \forall xy \neg p(x,y) \Rightarrow}(\forall l) \qquad \dfrac{\dfrac{\dfrac{}{p(c,c), \neg p(c,c) \Rightarrow}}{p(c,c), \forall y \neg p(c,c) \Rightarrow}(\forall l)}{p(c,c), \forall xy \neg p(x,y) \Rightarrow}(\forall l)}{(p(a,b) \vee p(c,c)), \forall xy \neg p(x,y) \Rightarrow}(\vee l)}{(p(a,b) \vee p(c,c)) \wedge (\forall xy \neg p(x,y)) \Rightarrow}(\wedge l)$$

Using resolution:

$(p(a,b) \vee \exists z P(z,z)) \to \exists xy P(x,y)$

negate the formula:

$(p(a,b) \vee \exists z p(z,z)) \wedge \neg(\exists xy P(x,y))$

$\simeq (p(a,b) \vee \exists z p(z,z)) \wedge \forall xy \neg P(x,y)$

$\xrightarrow{\text{skolemisation}} (p(a,b) \vee p(c,c)) \wedge \forall xy \neg p(x,y)$

$\xrightarrow{\text{clauses}} \{p(a,b), p(c,c)\} \quad \{\neg p(x,y)\}$

$\{p(c,c)\}\ [a/x, b/y] \quad \{\neg p(c,c)\}\ [c/x, c/y]$

$\square$

Just reuse what you did earlier (no need to repeat working)

Comments: Lets discuss a comparison in class.

Past Exam Questions

Question 1

1. Prof. Blunder was using a SAT solver to solve some propositional logic problems he found in a textbook, presented in clause form. Unfortunately, he typed in the problems incorrectly, making five types of error. In each of the following cases, briefly indicate what can be deduced about the original problem from the SAT solver output for the modified problem. Consider both possible outputs for the SAT solver: reporting "unsatisfiable" and outputting a model.

   (i) Mistyping some occurrences of a propositional symbol so it becomes two different symbols.

   (ii) Mistyping two different propositional symbols such that they become the same symbol

   (iii) Splitting a clause in two, e.g. replacing $\{P, \neg Q, R\}$ by $\{P, \neg Q\}$, and $\{R\}$.

   (iv) Deleting a clause

   (v) Moving a literal from one clause to another.

It would have been acceptable to make this assm.

- (i): It depends on which symbol we have typed for that occurrence, if it is a fresh symbol that is not used anywhere in other clauses, then we have placed fewer constraints; if it is a symbol that occurs somewhere else in other clauses, then that may introduce contradictory scenario for that symbol. Good note.
  - If output unsatisfiable: If the symbol we have typed for that occurrence is a fresh symbol, then the original problem is unsatisfiable because it is not satisfied even under fewer constraints. If the symbol occurs somewhere else in other clauses, then original problem might be satisfiable.
  - If output a model: we need to check whether the two values of the two symbols can be the same under such model, if so then the original problem is satisfiable; otherwise unsatisfiable.

  2/2

  Not necessarily. (Still the best answer I've seen).

- (ii):
  - If output is unsatisfiable: then the original problem might be satisfiable with the condition that these two propositional symbols will not take the same values.
  - If output a model: then the original problem is also satisfiable by simply assigning the same value to the two different propositional symbols.

  2/2

- (iii): suppose we have a clause $\{\underline{A}, \underline{B}\}$, where $\underline{A}, \underline{B}$ each represent a set of atomic formulas, we mistakenly spilt it into $\{\underline{A}\}$ and $\{\underline{B}\}$. It has no effect if we do not need to do case split on the atomic formulas in $\{\underline{A}\}$ and $\{\underline{B}\}$; otherwise we have assumed both of them are true, which is much strict then we need. not convinced by this.
  - If output unsatisfiable: if it is unsatisfiable because of conflicts in atomic formulas in $\{\underline{A}\}$ and $\{\underline{B}\}$, then the original problem is satisfiable; otherwise the original is also unsatisfiable.
  - If output a model: then the original problem is satisfiable using the same model, although there might be alternative models available. $\frac{1.5}{2}$
- (iv): deleting a clause means we have fewer constraints on our system
  - If output unsatisfiable: then the original problem is unsatisfiable because the system is unsatisfiable even without a further constraint.
  - If output a model: we are unsure about the original outcome, if the deleted clause satisfies the model, the original problem is satisfiable; otherwise the original problem is unsatisfiable ← not necessarily $\frac{1.5}{2}$
- (v): suppose we have a clause $\{\underline{A}, \underline{B}\}, \{\underline{C}\}$, where $\underline{A}, \underline{B}, \underline{C}$ each represent a set of atomic formulas, we mistakenly spilt it into $\{\underline{A}\}$ and $\{\underline{B}, \underline{C}\}$. ← We only move 1 literal (i.e. atomic formula)
  - If output unsatisfiable: If the conflicts are caused by atomic formulas in $\{\underline{A}\}$ and $\{\underline{B}\}$, then we may be able to produce a model that satisfies $\{\underline{A}\}$ and $\{\underline{C}\}$, thus the original problem will be satisfiable; otherwise, the original problem is unsatisfiable. What does this mean?
  - If output a model: If the model satisfies $\{\underline{B}\}$ but not $\{\underline{C}\}$, then the original problem is unsatisfiable; if the model satisfies $\{\underline{C}\}$, no matter whether it satisfies $\{\underline{B}\}$, the original problem can be satisfied by the model. $\frac{1}{2}$

💡 Comments: Great job overall! A few answers were a little long (the inner dot points usually had main part of answer). $\frac{8}{10}$

Question 2

2. This part is concerned with Binary Decision Diagrams. Use the variable ordering P, Q, R.

   (i) Write down the BDD's for $P \wedge Q \wedge R$ and $(\neg R \wedge Q) \to P$. There is no need to show your work.

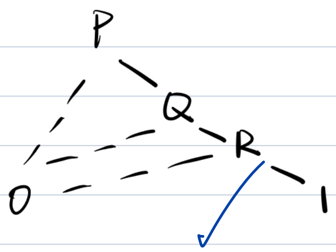
   (ii) Use the results above to obtain the BDD of:

$$[P \wedge Q \wedge R] \leftrightarrow [(\neg R \wedge Q) \to P] \leftrightarrow P$$

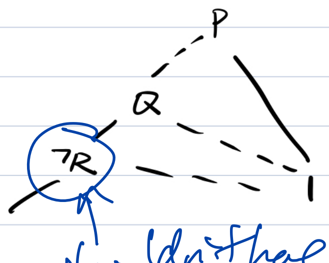
   showing your working

Hint: In $A \leftrightarrow B \leftrightarrow C$ the order of the operands is insignificant.

(i) BDD for $P \wedge Q \wedge R$:

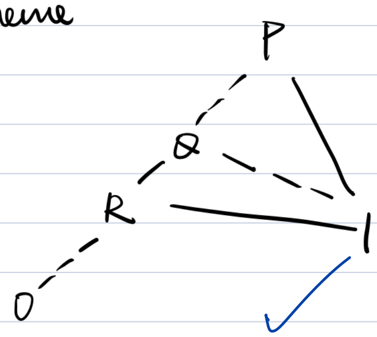


BDD for $(\neg R \wedge Q) \to P$



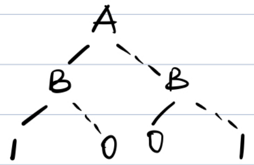
shouldn't have nots in tree!

hence



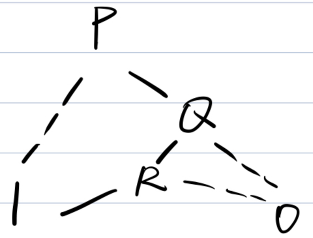
2/2

vi) suppose $A \Leftrightarrow B$, we have BDD

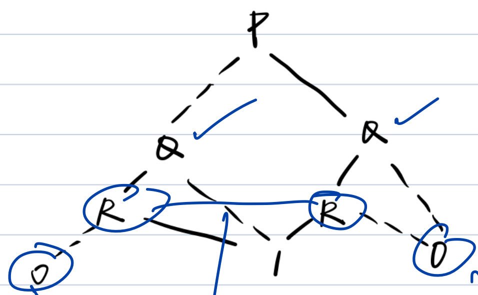


since in $A \Leftrightarrow B \Leftrightarrow C$, the order of the operands is insignificant,

→ BDD for $P \Leftrightarrow (P \wedge Q \wedge R)$:



→ BDD for $(P \Leftrightarrow (P \wedge Q \wedge R)) \Leftrightarrow ((\neg R \wedge Q) \rightarrow P)$



$\frac{6}{8}$

Comments:

Could combine R

should only have a single 0 & 1 leaf.