# Key for Midterm 2

## Wednesday November 17
## 75 minutes == 75 points
## open book and notes

1. *10 points*
   Two sentences in propositional calculus can be shown to be equivalent by proving that one entails the other and viceversa. Show that $\neg(p \wedge q) \equiv \neg p \vee \neg q$ by doing the following steps:

   (a) Prove by contradiction using resolution

   $$\neg(p \wedge q) \models \neg p \vee \neg q$$

   We put $\neg(p \wedge q)$ in the KB, add the negation of the goal $\neg(\neg p \vee \neg q)$, transform them to CNF, and prove a contradiction.
   1. $\neg(p \wedge q)$ becomes $\neg p \vee \neg q$
   2. $\neg(\neg p \vee \neg q)$ becomes $p \wedge q$ which produces
   2a. $p$
   2b. $q$
   We resolve 1. $\neg p \vee \neg q$ with 2a. $p$ and obtain 3. $\neg q$
   We resolve 3. $\neg q$ with 2b. $q$ and produce a contradiction.

   (b) Prove by contradiction using resolution

   $$\neg p \vee \neg q \models \neg(p \wedge q)$$

   We put $\neg p \vee \neg q$ in the KB, add the negation of the goal $\neg(\neg(p \wedge q))$, transform them to CNF, and prove a contradiction.
   1. $\neg p \vee \neg q$ 2. $\neg(\neg(p \wedge q))$ becomes $p \wedge q$ which produces
   2a. $p$
   2b. $q$
   We resolve 1. $\neg p \vee \neg q$ with 2a. $p$ and obtain 3. $\neg q$
   We resolve 3. $\neg q$ with 2b. $q$ and produce a contradiction.

2. *15 points*
   You are given the following sentence "Heads I win, tails you lose."

   (a) Represent it in propositional calculus using the following propositions $Head$, $Tail$, $IWin$, $YouLose$.
   1. $Head \Rightarrow IWin$
   2. $Tail \Rightarrow YouLose$

   (b) Suppose that you are told "Head". Prove, using any method you like, that "You lose". To do the proof you might need to represent additional knowledge.
   We are given

1. $Head \Rightarrow IWin$
2. $Tail \Rightarrow YouLose$
3. $Head$

and the additional knowledge

4. $IWin \Rightarrow YouLose$

We can use Modus Ponens.

From 3 and 1 we infer 5. $Iwin$.
From 5 and 4 we infer 6. $YouLose$

(c) Suppose that you are now told "Tail". Can you prove that "I do not win"? Do you need any additional knowledge? Comment briefly on your choice of additional knowledge.

This cannot be proven. Let's see why. We are given

1. $Head \Rightarrow IWin$
2. $Tail \Rightarrow YouLose$
3. $Tail$

and the additional knowledge

5. $YouLose \Rightarrow IWin$

using Modus Ponens from 3. and 2. we can derive 6. $YouLose$.
Using 6. and 5. we can derive $IWin$.
Since $IWin$ is satisfiable, its negation $\neg IWin$ is unsatisfiable.

3. *10 points*
Convert the following sentences into a form in which all the quantifiers are as far to the left as possible (without changing the meaning of the sentence).

(a) $\forall x[[\exists y Loves(x,y) \vee Loves(y,x)] \Rightarrow Happy(x)]$

We can move the existential quantifier $\exists y$ out of the premise of the implication by replacing it with the universal quantifier $\forall x$ which quantifies across the entire expression.

$\forall x \forall y\, [Loves(x,y) \vee Loves(y,x)] \Rightarrow Happy(x)]$

To understand why this is correct, we can transform to CNF and back:

$\forall x[\neg[\exists y Loves(x,y) \vee Loves(y,x)] \vee Happy(x)]$
$\forall x[\forall y \neg Loves(x,y) \wedge \neg Loves(y,x)] \vee Happy(x)]$
$\forall x \forall y[Loves(x,y) \vee Loves(y,x)] \Rightarrow Happy(x)]$

(b) $\forall x[Happy(x) \Rightarrow [\exists y Loves(x,y)]]$

We can move the existential quantifier $\exists y$ from the conclusion of the implication and move it to quantify over the entire expression.

$\forall x \exists y\, [Happy(x) \Rightarrow Loves(x,y)]$

To understand why this is correct, we can transform to CNF and back:

$\forall x \neg Happy(x) \vee \exists y Loves(x,y)$
$\forall x \exists y \neg Happy(x) \vee Loves(x,y)$
$\forall x \exists y\, [Happy(x) \Rightarrow Loves(x,y)]$

4. *30 points*

Write the following sentences in predicate calculus. Be consistent in your choice of predicates,

(a) "Every city has a dogcatcher who has been bitten by every dog in town."

$\forall x\, City(x) \Rightarrow [\exists y\, DogCatcher(y) \wedge [\forall z\, Dog(z) \wedge LivesIn(z,x) \Rightarrow BittenBy(y,z)]]$

(b) "All mushrooms are either purple or poisonous but not both." This sentence requires an exclusive or.

$\forall x\, Mushroom(x) \Rightarrow [Purple(x) \vee Poisonous(x)] \wedge \neg[Purple(x) \wedge Poisonous(x)]$

or, equivalently,

$\forall x Mushroom(x) \Rightarrow [Purple(x) \wedge \neg Poisonous(x)] \vee [\neg Purple(x) \wedge Poisonous(x)]$

(c) "All purple mushrooms except one are poisonous."

$\exists x Mushroom(x) \wedge Purple(x) \wedge \neg Poisonous(x) \wedge \forall y [Mushroom(y) \wedge Purple(y) \wedge \neg Poisonous(y) \Rightarrow x = y]$

(d) "Rich people have big houses." This sentence has more than one interpretation. The most obvious is that if rich people have a house it is big.

$\forall x \forall y\, Rich(x) \wedge HasHouse(x,y) \Rightarrow Big(y)$

which could also be written as

$\forall x\, Rich(x) \Rightarrow [\forall y\, HasHouse(x,y) \Rightarrow Big(y)]$

Another possible interpretation is that rich people have at least a big house:

$\forall x\, Rich(x) \Rightarrow \exists y\, HasHouse(x,y) \wedge Big(y)$

(e) "Big houses require work unless they have a house keeper and no garden."

$\forall x House(x) \wedge Big(x) \Rightarrow Work(x) \vee [\exists y Keeper(y,x) \wedge \neg\exists z\, Garden(z,x)]$

or, equivalently,

$\forall x House(x) \wedge Big(x) \wedge [\neg\exists y Keeper(y,x) \vee \exists z\, Garden(z,x)] \Rightarrow Work(x)$

(f) "If Bill does not have a big house, Bill is not rich."

$\neg[\exists x\, HasHouse(Bill,x) \wedge Big(x)] \Rightarrow \neg Rich(Bill)$

or, equivalently

$\forall x\, \neg HasHouse(Bill,x) \vee \neg Big(x)] \Rightarrow \neg Rich(Bill)$

Another possible interpretation: if Bill has a house and the house is not big then Bill is not rich.

$\forall x\, HasHouse(Bill,x) \wedge \neg Big(x)] \Rightarrow \neg Rich(Bill)$

5. *10 points*

Show the backed-up values for all the nodes in the following game tree and show the branches that are pruned by alpha-beta. For each branch pruned, explain briefly why alpha-beta prunes it. Follow the convention used in the textbook to examine the branches in the tree from left to right.

The pruning is shown in the figure. The first pruning can be done because -3 (which is the value for min) is less than 0 (which is the value for Max). The next pruning can be done because 3 (which is the current value for Max) is greater than 0 (the current best choice for the top min node). No additional pruning can be done. After the last node (with a value of -3) is reached, the top min node selects -3.