CS3243 Tutorial 7

Eric Han

Oct 17, 2022

Annoucements

1. Assignment 6 scores are now on Luminus, please check.
2. Your midterm papers will be release at the end of this lesson. **Remember to collect your papers before you leave.**
    1. Check the math for the scores.
    2. Check that each question have been marked correctly.
    3. Raise any concerns or disputes (if applicable).
    4. Come talk to me or telegram me after class.
    5. Make sure any disputes or concerns are raised before 26 Oct 2359hrs.
3. Midterms statistics will be discussed in lectures; I will not cover it here.

Bonus Question

α-β algorithm is a very interesting algorithm. Draw a large tree to see the full capability:

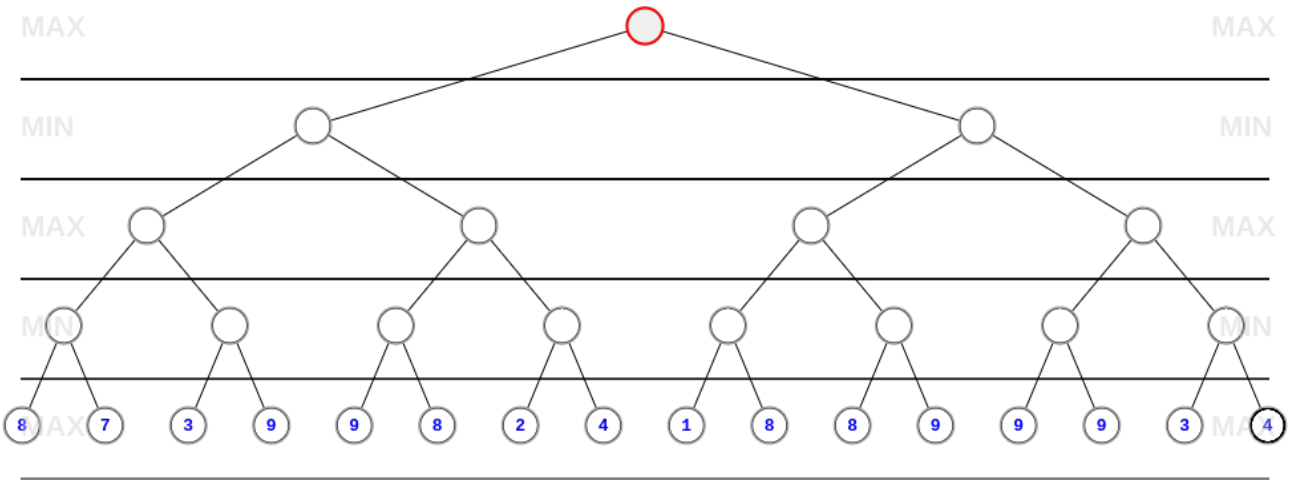

Figure 1: Alpha-Beta Example (Credit MIT)

Question 1

Verify the following logical equivalences. Cite the equivalence law used with each step of your working (refer to Appendix B for a list of these laws).

1. $\neg(p \vee \neg q) \vee (\neg p \wedge \neg q) \equiv \neg p$
2. $(p \wedge \neg(\neg p \vee q)) \vee (p \wedge q) \equiv p$

Recap

1. de Morgan's law
2. distributive law

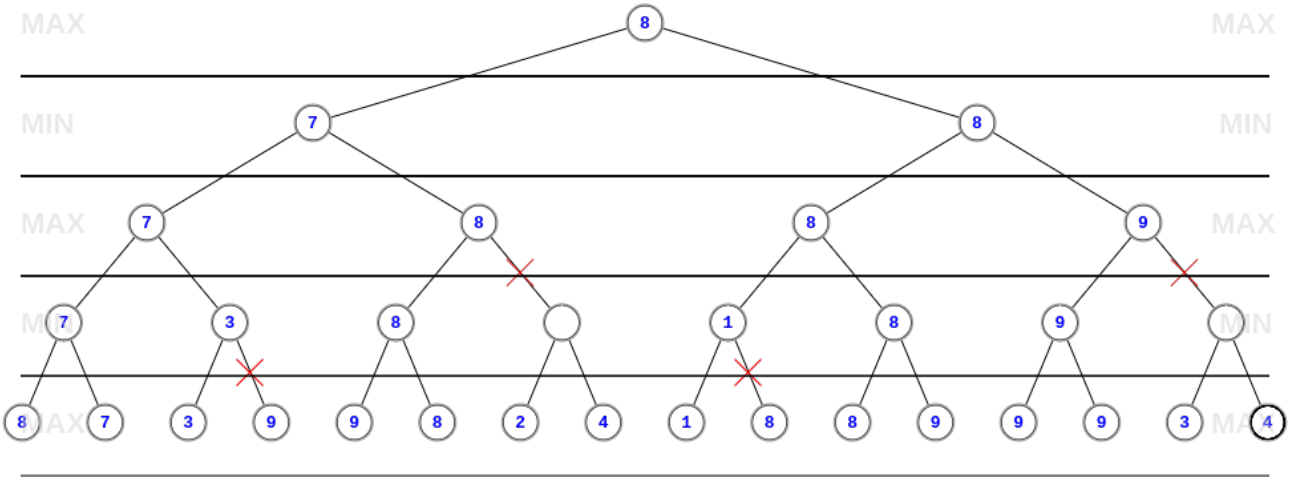

Figure 2: Alpha-Beta Answer (Credit MIT)

3. complement law
4. identity law
5. associative law
6. idempotent law

CS1231...

Appendix B: Propositional Logic Laws

De Morgan's Laws	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Idempotent laws	$p \vee p \equiv p$	$p \wedge p \equiv p$
Associative laws	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Commutative laws	$p \vee q \equiv q \vee p$	$p \wedge q \equiv q \wedge p$
Distributive laws	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Identity laws	$p \vee False \equiv p$	$p \wedge True \equiv p$
Domination laws	$p \wedge False \equiv False$	$p \vee True \equiv True$
Double negation law	$\neg\neg p \equiv p$	
Complement laws	$p \wedge \neg p \equiv False \wedge \neg True \equiv False$	$p \vee \neg p \equiv True \vee \neg False \equiv True$
Absorption laws	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional identities	$p \Rightarrow q \equiv \neg p \vee q$	$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$

Figure 3: Appendix B

Q1a Answer

$$\begin{aligned}
\neg(p \vee \neg q) \vee (\neg p \wedge \neg q) &\equiv (\neg p \wedge q) \vee (\neg p \wedge \neg q) && \because \text{de Morgan's law} \\
&\equiv \neg p \wedge (q \vee \neg q) && \because \text{distributive law} \\
&\equiv \neg p \wedge 1 && \because \text{complement law} \\
&\equiv \neg p && \because \text{identity law}
\end{aligned}$$

Q1b Answer

$$\begin{aligned}
(p \wedge \neg(\neg p \vee q)) \vee (p \wedge q) &\equiv (p \wedge (p \wedge \neg q)) \vee (p \wedge q) && \because \text{de Morgan's law} \\
&\equiv ((p \wedge p) \wedge \neg q) \vee (p \wedge q) && \because \text{associative law} \\
&\equiv (p \wedge \neg q) \vee (p \wedge q) && \because \text{indempotent law} \\
&\equiv p \wedge (\neg q \vee q) && \because \text{distributive law} \\
&\equiv p \wedge 1 && \because \text{complement law} \\
&\equiv p && \because \text{identity law}
\end{aligned}$$

. . .

Tip: Stuck? Truth Table!

Question 2

Victor would like to invite three friends, Alice, Ben, and Cindy to a party, but must satisfy the following constraints:

a. Cindy comes only if Alice does not come.
b. Alice comes if either Ben or Cindy (or both) comes.
c. Cindy comes if Ben does not come.

Victor would like to know who will come to the party, and who will not. Help Victor by expressing each of the above three constraints in propositional logic, and then, using these constraints, determine who will attend his party.

Recap

1. How to formulate this as a Knowledge Base problem?

. . .

You need 2 items: Variables, Constraints

Recall that in formal logic, your expressions are used as follows:

1. A if B means that B implies A
2. A only if B means that A implies B
3. A if and only if B means that A is equivalent to B.

. . .

Answer

Variables: Boolean variables and what they represent.

1. a represent Alice coming
2. b represent Ben coming
3. c represent Cindy coming

Constraints:

Cindy comes only if Alice does not come: $c \implies \neg a$

a	c	$\neg a$	$c \implies \neg a$
0	1	1	1
1	1	0	0
0	0	1	1
1	0	0	1

Alice comes if either Ben or Cindy (or both) comes: $b \vee c \implies a$

a	b	c	$b \vee c$	$b \vee c \implies a$
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	0	1
0	1	1	1	0
0	1	0	1	0

a	b	c	$b \vee c$	$b \vee c \implies a$
0	0	1	1	0
0	0	0	0	1

Cindy comes if Ben does not come: $\neg b \implies c$

b	c	$\neg b$	$\neg b \implies c$
0	1	1	1
0	0	1	0
1	1	0	1
1	0	0	1

In summary, we need to solve all, using implication law, to CNF:

1. $c \implies \neg a \equiv (\neg c \vee \neg a)$
2. $b \vee c \implies a \equiv (b \implies a) \wedge (c \implies a)$
    1. $b \implies a \equiv (\neg b \vee a)$
    2. $c \implies a \equiv (\neg c \vee a)$
3. $\neg b \implies c \equiv (b \vee c)$

Solve, the following:

$$
\begin{aligned}
&(\neg c \vee \neg a) \wedge (\neg c \vee a) \wedge (\neg b \vee a) \wedge (b \vee c) && \\
&\equiv (\neg c \vee (\neg a \wedge a)) \wedge (\neg b \vee a) \wedge (b \vee c) && \because \text{distributive law} \\
&\equiv (\neg c \vee 0) \wedge (\neg b \vee a) \wedge (b \vee c) && \because \text{complement laws} \\
&\equiv (\neg c) \wedge (\neg b \vee a) \wedge (b \vee c) && \because \text{identity laws} \\
&\equiv (\neg b \vee a) \wedge ((\neg c \wedge b) \vee (\neg c \wedge c)) && \because \text{distributive laws} \\
&\equiv (\neg b \vee a) \wedge ((\neg c \wedge b) \vee 0) && \because \text{complement laws} \\
&\equiv (\neg b \vee a) \wedge (\neg c \wedge b) && \because \text{identity laws} \\
&\equiv (\neg b \wedge \neg c \wedge b) \vee (a \wedge \neg c \wedge b) && \because \text{distributive laws} \\
&\equiv (0 \wedge \neg c) \vee (a \wedge \neg c \wedge b) && \because \text{associative,complement laws} \\
&\equiv (a \wedge b \wedge \neg c) && \because \text{domination,identity laws}
\end{aligned}
$$

Alice and Ben will come to Victor's party, but not Cindy.

Question 3

Knowledge Base:

- All firetrucks are red
- All firetrucks are cars
- All cars have four wheels

Recap

1. What is completeness / soundness?

. . .

The inference algorithm is complete if it can derive any sentence that is entailed by KB. The inference algorithm is sound if it derives only sentences that are entailed by KB.

---

[1] https://steemit.com/software/@cpuu/the-difference-between-soundness-and-completeness

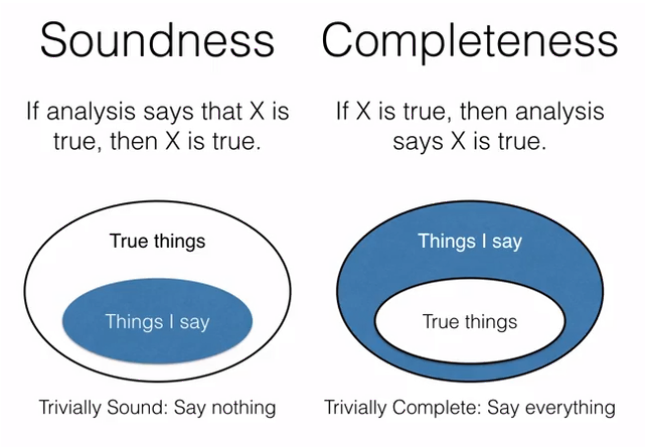

Figure 4: Completeness and Soundness [1]

Q3a Answer

Assume that an inference algorithm, A_1, that takes the query sentence 'a ferrari is a red car' and infers 'a ferrari is a firetruck'. Determine which of the following properties does not apply to A_1:

1. Complete
2. Sound
3. Both of the above

. . .

Soundness does not apply to A_1; The inferred statement is not entailed by the KB. We do not know enough information if the algorithm is complete.

Q3b Answer

Assume that an inference algorithm, A_2 is given the query sentence 'a ferrari is a red car'. Determine which of the following properties would guarantee that A_2 would infer the sentence 'a ferrari has four wheels':

1. Complete
2. Sound
3. Both of the above

. . .

Completeness ensures that everything that is entailed will be inferred; this property gauarantees the inference.

Q3c Answer

> Two agents with the same knowledge base and different inference engines, both of which are complete and sound, always behave in the same way.

Determine if the statement is True or False. Justify your answer.

. . .

False. The behavior of an agent is the interaction with the environment. Although they will result with the same inferencesm they may have different objectives even with the same knowledge.

For example,

1. Achieve efficiency
2. Achieve performance

Question 4

Assignment Question; we will go through this question next week.

Bonus Question - Work for Snack

1. Solve Q4 using code to double check your working.
2. No boilerplate code is given, work from scratch; You can commit it to the repository forked from https://github.com/eric-vader/CS3243-2223s1-bonus/.
3. Your code should tell me for both 4a and 4b if $KB \models \alpha$ or not as the result and also printing the truth table.

Using code, you should be able to solve this in under 5 mins; But you should still trace manually.