CS3243 Tutorial 7

Eric Han

March 24, 2022

Annoucements

1. Assignment 5 scores are now on Gradebook, please check.
2. Assignment 6 scores are being marked, will appear on tunitin, please check.
3. Midterm scores are now frozen; no more changes are allowed.
4. Wordle Bot on Github - https://github.com/eric-vader/wordle.

Discussion Question from T16

If the opponent plays sub-optimally (not min) and I still play optimally (max); in that case, can I be worse off when compared to the current value (not the true optimal value in this situation as discussed)? Why or why not?

1. If the min player does not play optimally, it would only increase the utility value. It cannot decrease the utility value as it is already the minimum.
2. Hence, `minimax` gives the worse-case utlity v for the max player, ie. $\geq v$.

However,

1. If the min player does not play optimally, the optimal choice is still not taken.
2. In conditions such as not expanding all the branches and all depths, we can end up at very bad states (even losing the game).

If we expand everything with full infomation, the utility value is the worse-case for the max player.

Previously from T06, Q1

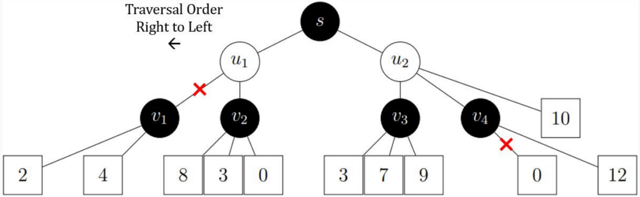

Figure 1: Q1a

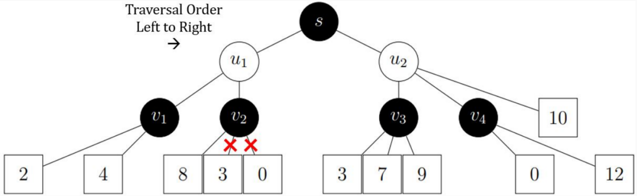

Figure 2: Q1b

Qn: How can we benefit from α-β's efficiency?

Run through the α-β:

1. Right to Left
2. Left to Right

Then determine if the effectiveness of pruning depends on iteration order.

Recap

1. What does α-β do?
2. What kind of efficiency do you gain?
3. What is deep cutoff?

Previously from T06, Q1

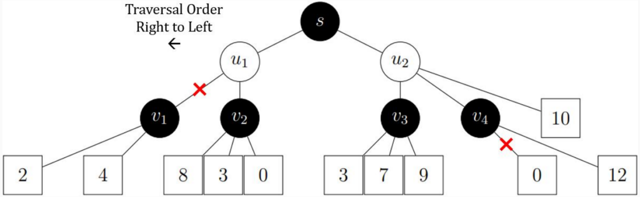

Figure 1: Q1a

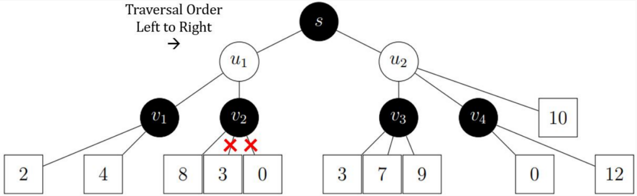

Figure 2: Q1b

Qn: How can we benefit from α-β's efficiency?

Run through the α-β:

1. Right to Left
2. Left to Right

Then determine if the effectiveness of pruning depends on iteration order.

Recap

1. What does α-β do?
2. What kind of efficiency do you gain?
3. What is deep cutoff?

Save on static evaluation and move generation.

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

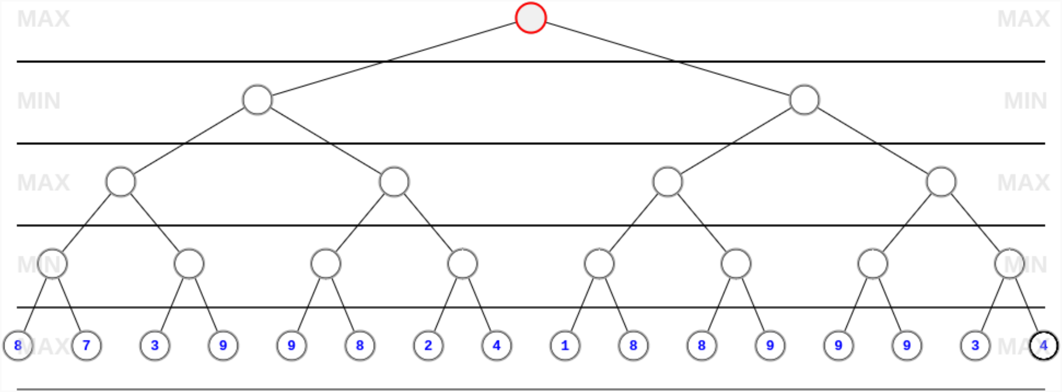

Figure 3: Alpha-Beta Example (Credit MIT)

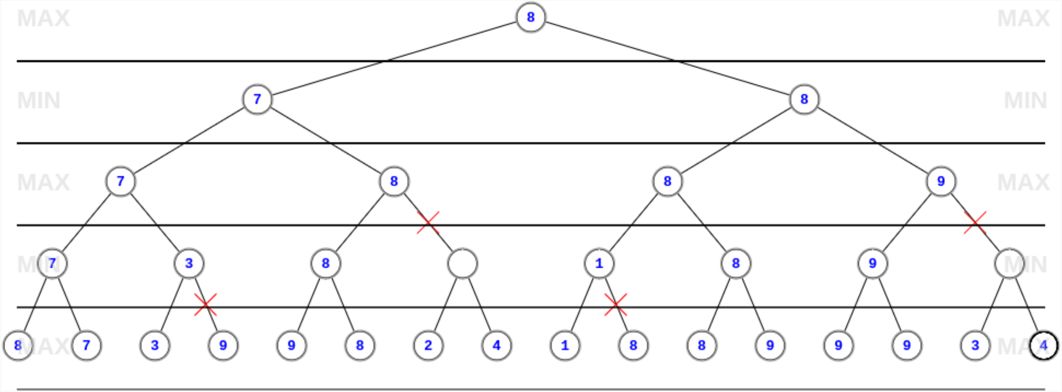

Figure 4: Alpha-Beta Answer (Credit MIT)

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
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 5: 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

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

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

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

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. [Question is updated]

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?

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. [Question is updated]

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.

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
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. $c \implies \neg a \equiv (\neg c \lor \neg a)$
2. $b \lor c \implies a \equiv (b \implies a) \land (c \implies a)$
    2.1 $b \implies a \equiv (\neg b \lor a)$
    2.2 $c \implies a \equiv (\neg c \lor a)$
3. $\neg b \implies c \equiv (b \lor c)$

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.