15 Feb '25

→ Checked HW on where greedy approach fails in giving min. no. (number) of coins in change

HW: Give 5 more examples where total amount for change > 50.

Eg: Amount = 71

Coins = 11c, 10c, 1c

Greedy solution = 11 coins (6→11c ; 5→1c)

Min. = 8 coins (7→10c ; 1→1c)

→ Checked HW on truth tables for boolean expressions. We also saw how to evaluate boolean expressions using different properties.

→ AND and OR operators satisfy closure, commutative, associative & distributive properties. (See AI generated notes).

→ AND & OR are somewhat like * and + but

AND distributes over OR
and also
OR distributes over AND
but for *, +
* distributes over +
but + doesn't distribute over *

→ **HW:** For each of the 4 properties (for both AND & OR), write truth tables and verify if the properties are satisfied. (8 different problems)

Eg: Associative property for AND

x	y	z	x AND y	(x AND y) AND z (A)	y AND z	x AND (y AND z) (B)
true	true	true	true	T	T	T
true	true	false	true	F	F	F
true	false	true	false	F	F	F
true	false	false	false	F	F	F
false	true	true	false	F	T	F
false	true	false	false	F	F	F
false	false	true	false	F	F	F
false	false	false	false	F	F	F

We see that (A) & (B) are always equal.

$\therefore$ (x AND y) AND z = x AND (y AND z)
(therefore)

which means AND satisfies associative property.

→ We also looked at how to simplify boolean expressions (without Truth tables)

→ x and ~x = false

x or ~x = true

x and true = x

x and false = false

x or true = true

x or false = x

We can use these properties & closure/associativity etc. to simplify boolean expressions.

Eg: (See notes on HW examples from last class on truth tables).

Eg1: (A and B) or (A and ~B)

= A and (B or ~B) (distribution of and over or)

= A and true

= A

Eg 2: A or (A and B)

= (A or A) and (A or B)

= A and (A or B)

= (A and A) or (A and B)

= A or (A and B)

$$\begin{tikzcd} p \arrow[d] \\ q \arrow[d] \\ p \end{tikzcd}$$

This is just going in cycles between p and q.

So, let's see if we can solve it in a different way.

A or (A and B) = P

If A = true, P is true

If A = false, P is false

So, A & P are always equal.

So, A or (A and B) = A.

This is called Absorption property. See AI generated notes for more on these properties & examples.

HW: Write truth tables for these properties also & check if these properties are true. (13 examples).

Eg: A and (A or B) = A

A	B	A or B	A and (A or B)
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F
		(X)	(Y)

(X) = (Y) always.

So this absorption property is true.

HW: Simplify these expressions

(1) A or ~A and (B or false)

(2) (A and B) or (A and ~B)

(3) A or (false and B)

(4) (A and B) or ~A or (A and ~B)

(5) A and ~(A or B)

(6) (A or B) and (not A or B)

(7) (A and B and C) or (A and B and ~C)

(8) A or (B and ~A)