Q1

(AND True) False

True = (λx·(λy·x)
AND = (λp·(λq·(p q) p)

((λp·(λq·(p q) p) True) False

((λq·(p q) p)[p := True]) False →β

(λq·(True q) True) False

(λq·((λx·(λy·x) q) True) False →α

(λq·((λx·(λy·x) q) True) False

(λq·((λy·x)[x := q]) True) False →β

(λq·((λy·q) True) False

(λq·((q)[y := True]) False →β

(λq·q) False

q[q := False] →β

False

Q2

(OR True)

True = (λx·(λy·x)
OR = (λp·(λq·(p p) q)

((λp·(λq·(p p) q) True)

((λq·(p p) q)[p := True]) →β

((λq·(True True) q)

((λq·((λx·(λy·x) True) q) →α

((λq·((λx·(λy·x) True) q)

((λq·((λy·x)[x := True]) q) →β

((λq·(λy·True) q)

((λq·(True)[y := q]) →β

(λq·True)

–

A function waiting for a parameter only to return True, ignoring the parameter.

Q3

XOR

a	*b*	*xor*
0	*0*	*0*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

In-fix – *(a || b) & ! (a & b)*

Post-fix – *((&) ((||) a b) ((!) ((&) a b)))*

Deriving NOT;

1)

NOT TRUE $\rightarrow_\beta$ FALSE

TRUE FALSE TRUE $\rightarrow_\beta$ FALSE

2)

NOT FALSE $\rightarrow_\beta$ TRUE

FALSE FALSE TRUE $\rightarrow_\beta$ TRUE

By 1) and 2) we can conclude,

NOT := λp·(p FALSE TRUE)

Now try to implement the post-fix in lambda calculus,

(λx·(λy·(AND (OR x y) (NOT (AND x y)))))

Q4

Summary

$2^n * 2^n$ board (Fig 2) can be covered by just V3 pieces (Fig 1) with a corner removed where, $n \epsilon N$.

Fig 1

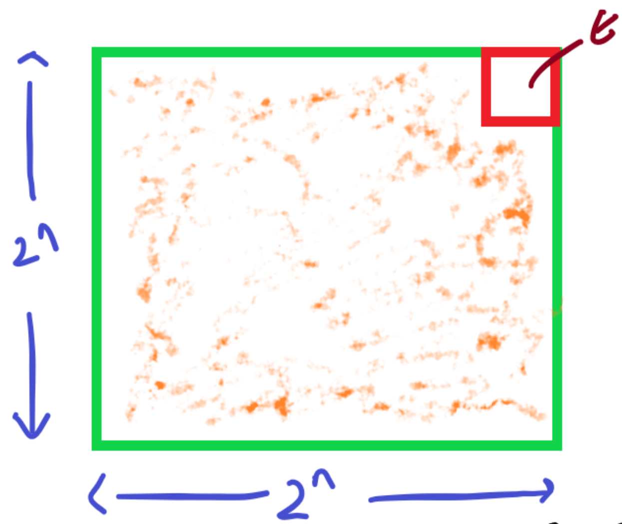

Fig 2

Proof

Base $(n=1)$

We have, $2^n * 2^n = 2^1 * 2^1$. With 1 V3 piece shown in orange.

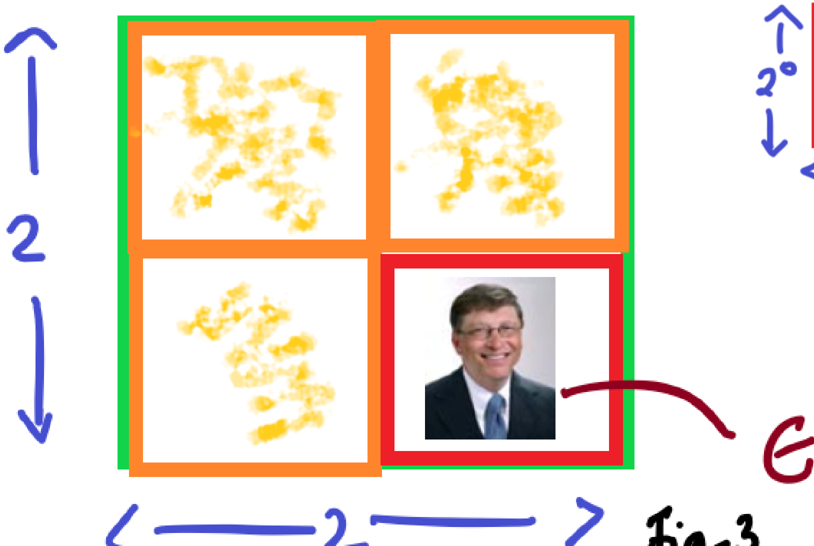

Fig 3

Where $(n=0)$, $2^0 * 2^0 = 1 * 1$ is covered naturally.

Hypothesis $(n = k)$

We have, $2^k * 2^k$ assumed to be true.

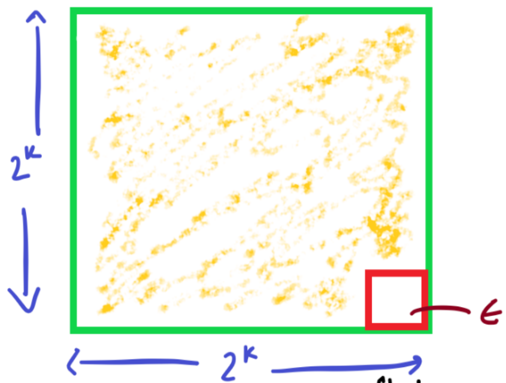

fig 4

Inductive Step $(n = k+1)$

By *Hypothesis* we have,

$$2^{k+1} * 2^{k+1} = 2^k * 2 * 2^k * 2 = 2 * (2^k * 2^k)$$

Shown true, by inserting 1, V3 piece in the center of 4 blocks of the k case.

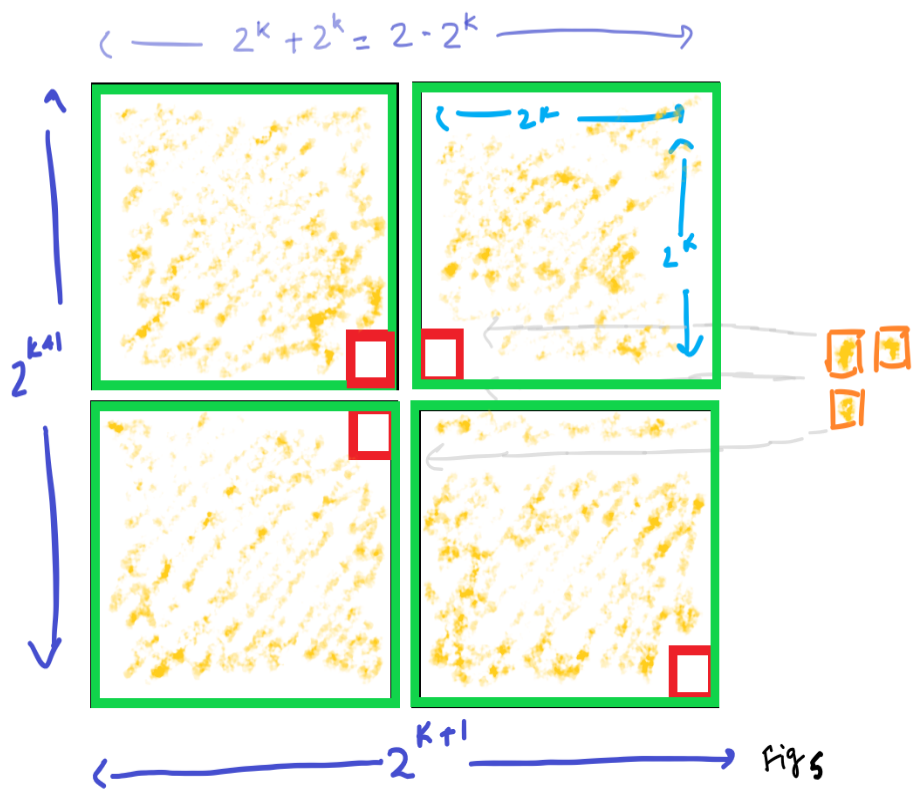

fig 5