COMP3400
Assignment 1 Written Solutions

Paul Vrbik

April 6, 2023

Lambda Calculus

We can encode boolean logic in lambda calculus as follows:

$$\mathsf{True} = \lambda x.(\lambda y.x) = \lambda xy.x \qquad \mathsf{False} = \lambda x.(\lambda y.y) = \lambda xy.y$$
$$\mathsf{And} = \lambda p.(\lambda q.(pq)p) = \lambda pq.pqp \qquad \mathsf{Or} = \lambda p.(\lambda q.(pp)q) = \lambda pq.ppq$$

Question 1. [1 MARK]

Give the λ-expression for NOT that takes True to False and vice-versa. Your solution should be in its β-normal form.

Answer:

$$\mathsf{Not} \equiv \lambda p.(p\,\mathsf{False}\,\mathsf{True}) \qquad \text{2 Marks}$$

Question 2. [5 MARKS]

Recall that $\neg(p \wedge q) \equiv \neg p \vee \neg q$ and thereby Or is redundant because

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

Give the λ-expression for $\neg(\neg p \wedge \neg q)$ and show it is equivalent to Or.

Answer: We give the lambda expression and reduce to the beta-normal form.

$$\begin{aligned}
&\lambda pq.\mathsf{Not}\ (\mathsf{And}\ (\mathsf{Not}\ p)\ (\mathsf{Not}\ q)) \\
&\quad = \lambda pq.\mathsf{Not}\ (\mathsf{And}\ (p\,\mathsf{False}\,\mathsf{True})\ (q\,\mathsf{False}\,\mathsf{True})) \\
&\quad = \lambda pq.\mathsf{Not}\ ((p\,\mathsf{False}\,\mathsf{True})\ (q\,\mathsf{False}\,\mathsf{True})\ (p\,\mathsf{False}\,\mathsf{True})) \\
&\quad = \lambda pq.((p\,\mathsf{False}\,\mathsf{True})\ (q\,\mathsf{False}\,\mathsf{True})\ (p\,\mathsf{False}\,\mathsf{True}))\,\mathsf{False}\,\mathsf{True} \qquad \text{1 Mark}
\end{aligned}$$

Let this beta-reduced form be denoted Or' and observe Or' has the same truth table as Or.

$$\begin{aligned}\mathsf{Or}'\ \mathsf{True} &= \lambda q.((\mathsf{True\ False\ True})\ (q\ \mathsf{False\ True})\ (\mathsf{True\ False\ True}))\ \mathsf{False\ True}\\ &= \lambda q.(\mathsf{False}\ (q\ \mathsf{False\ True})\ \mathsf{False})\ \mathsf{False\ True}\\ &= \lambda q.\mathsf{False\ False\ True}\\ &= \lambda q.\mathsf{True}\end{aligned}$$

and thus $\mathsf{Or}'\ \mathsf{True\ False} = \mathsf{Or}'\ \mathsf{True\ True} = \mathsf{True}$. 2 Marks

$$\begin{aligned}\mathsf{Or}'\ \mathsf{False} &= \lambda q.((\mathsf{False\ False\ True})\ (q\ \mathsf{False\ True})\ (\mathsf{False\ False\ True}))\ \mathsf{False\ True}\\ &= \lambda q.(\mathsf{True}\ (q\ \mathsf{False\ True})\ \mathsf{True})\ \mathsf{False\ True}\\ &= \lambda q.(q\ \mathsf{False\ True})\ \mathsf{False\ True}\\ &= \lambda q.(\mathsf{Not}\ q)\ \mathsf{False\ True}\\ &= \lambda q.\mathsf{Not\ Not}\ q\\ &= \lambda q.q\end{aligned}$$

and thus $\mathsf{Or}'\ \mathsf{False\ True} = \mathsf{True}$ and $\mathsf{Or}'\ \mathsf{False\ False} = \mathsf{False}$ 2 Marks.

Thus Or' and Or are equivalent over the domain $\{\mathsf{True}, \mathsf{False}\}$.

Question 3. [4 MARKS]

Reduce the following lambda expression to its β-normal form.

Answer:

$$\begin{aligned}&(\lambda xy.x)(\lambda abc.cab)\,z\,(\lambda z.zz)\\ &= (\lambda y.(\lambda abc.cab))\,z\,(\lambda z.zz) && \text{1 Mark}\\ &= (\lambda abc.cab)\,(\lambda z.zz) && \text{1 Mark}\\ &= \lambda bc.c(\lambda z.zz)b && \text{1 Mark}\end{aligned}$$

Does not reduce any further than this 1 Mark.

Principal Types

Question 4. [2 MARKS]

Define a function `f1` such that

```
f1 :: (a -> b, a) -> b
f1 f a = f a
```

Question 5. [2 MARKS]

```
f2 :: a -> (b, c) -> b
f2 a (b, c) = b
```

Question 6. [2 MARKS]

```
f3 :: (a -> a) -> a -> [a]
f3 f a = [f.f $ a]
```

Question 7. [2 MARKS]

```
f4 :: (b -> r) -> (a -> b) -> (a -> r)
f4 = (.)
```

Question 8. [1 MARK]

```
f5 :: ((a, b, c) -> d) -> a -> b -> c -> d
f5 f a b c = f (a, b, c)
```

Question 9. [1 MARK]

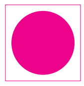
```
f5_inv ::  (a -> b -> c -> d) -> (a, b, c) -> d
f5_inv f (a, b, c) = f a b c
```

Blockus

Let $P(n) : \mathbb{N} \to \mathbb{B}$ and

$$P(n) \iff \text{a } 2^n \times 2^n \text{ board can be tiled with } V3 \text{ pieces so that only the north-west corner is left uncovered}$$

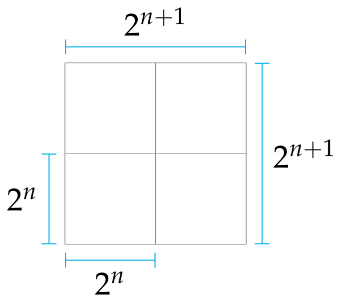
Base Case. We can cover a $2^0 \times 2^0 = 1 \times 1$ board with *zero* tiles:



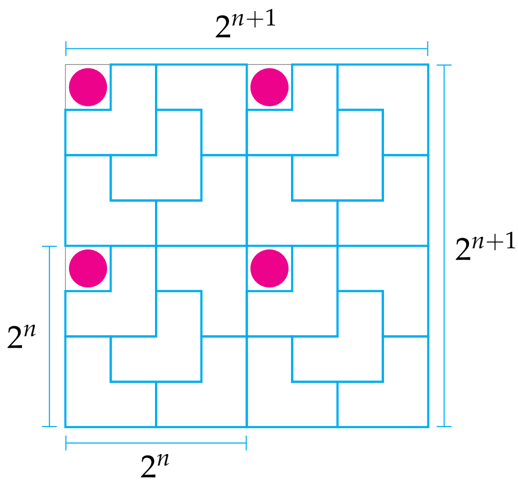
and thus $P(0)$. (Note: the magenta disc denotes the uncovered square.) (1 Mark)

Induction Hypothesis. Assume $P(n)$. (1 Mark)

Induction Notice a $2^{n+1} \times 2^{n+1}$ board can be decomposed into four $2^n \times 2^n$ boards:

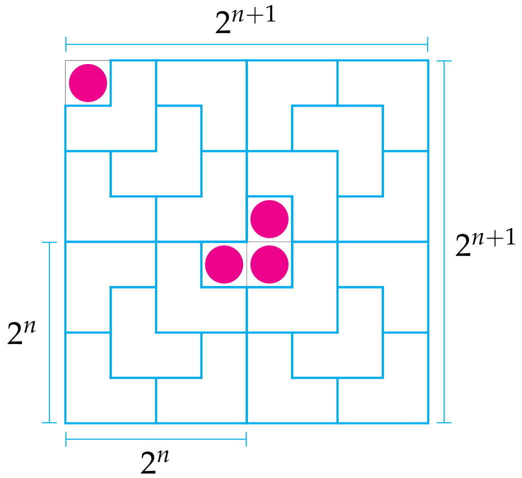


By our induction hypothesis (1 Mark) each sub-board can be tiled so that only the north-west square is left uncovered:

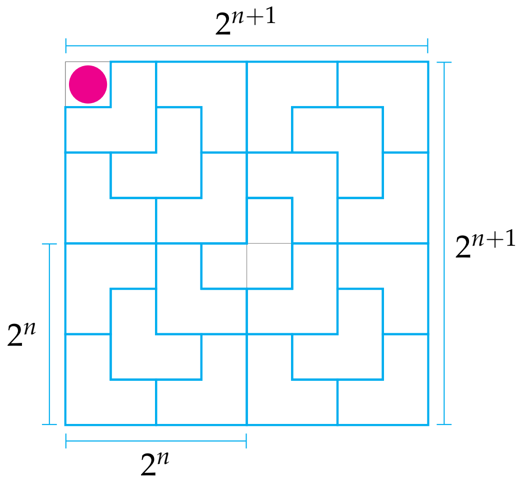


(Note: the magenta discs denote the uncovered squares.)

Notice that *the tiling is invariant to rotation* (1 Mark) and thus we can rotate the sub-boards in quadrant 1 and 3 so that the uncovered squares of quadrant 1, 2, and 3 form a space for a single V3 tile. (1 Mark)



Placing a single $V3$ piece over the uncovered center squares (1 Mark) leaves a tiled $2^{n+1} \times 2^{n+1}$ board with only the north-west corner uncovered.



Thus $P(n) \implies P(n+1)$. Alternatively: Thus $P(n+1)$. (1 Mark)

Conclusion: By the PMI $\forall n \in \mathbb{N}; P(n)$. (1 Mark)

(1 Mark) for style. Is the proof unnecessarily long? Are there unsubstantiated statements? Consistent and appropriate use of symbols and terms?