# Homework (Week 1)

Submission: Due on Friday, 10th of June, 11am Sydney Time. Please submit using the CSE Give System either online or via this command on a CSE terminal:

```
give cs3151 hw1 hw1.pdf
```

Please put all your answers in one PDF file called `hw1.pdf`. Use of LaTeX is encouraged but not required. Please make your answers as concise as possible. This homework should not be more than two pages.

Late submissions are accepted up to five days after the deadline, but at a penalty: 5% off your total mark per day.

## Circularity (1 mark)

What is the answer to this question?

*Hint*: You will find the answer to this question on Ed. If you can't access Ed yet, see the Announcements page for instructions.

## Dining Cryptographers (2 marks)

Assume the setting described in the first lecture for the problem of the Dining Cryptographers. Suppose we modify the protocol so that paying cryptographers now tell the truth about whether the coin tosses are different or equal. Instead, they will lie about whether they got head or tails.

At the end, do we still know if the NSA paid or not? Is confidentiality still preserved? Briefly explain why or why not.

## Safety and Liveness (5 marks)

### Limit closures

Let $s$ be a state, and let $s^\omega$ denote the behaviour $sssssssssss\ldots$ (i.e. infinitely many

repetitions of $s$.)

Give an example of a set $A$ such that $s^\omega \in \overline{A}$, but $s^\omega \notin A$.

## Alpern and Schneider's theorem

1. Let $\Sigma = \{a, b\}$. That is, we assume there are only two states, $a$ and $b$. Consider the property $P = \{\sigma | \sigma \text{ contains exactly one } b\}$. Use Alpern and Schneider's theorem to decompose $P$ into a safety property $P_S$ and a liveness property $P_L$. Simplify them; that is, don't just say $P_L = \Sigma^\omega \backslash (\overline{P} \backslash P)$ but give something that explains what $P_L$ *is*.

2. Assume $P$ is a safety property. Prove that $\Sigma^\omega \backslash (\overline{P} \backslash P) = \Sigma^\omega$ using the algebraic laws of set operations.

3. Is the empty property $\emptyset$ a liveness property? Is it a safety property? Explain.

## Bonus Question

*This question is not for marks, but for a meaningless brownie point*

Assuming there are at least two distinct states $a$ and $b$, prove that any property $P$ is the intersection of two liveness properties.

*Hint*: It is helpful to note that the union of a dense set and any set is itself dense.

# Temporal Logic (5 marks)

## Examples

Define suitable predicate symbols and give LTL formalisations for the following properties:

1. Once the dragon was slain, the princess lived happily ever after.
2. The dragon was never slain, but the princess lived happily until she didn't.
3. The dragon was slain at least twice.
4. The dragon was slain at most once.
5. Whenever the dragon was slain, the princess did not live happily.

## Proof

Prove the following logical statements:

$$\begin{aligned} \Box\Box\varphi &\Leftrightarrow \Box\varphi \\ \Diamond\bigcirc\varphi &\Leftrightarrow \bigcirc\Diamond\varphi \\ \Box\varphi &\Rightarrow \Diamond\varphi \end{aligned}$$

It may help to use these semantic definitions for $\Box$ and $\Diamond$ (derivable from the definition in terms of $\mathcal{U}$):

$$\begin{aligned} \sigma \vDash \Diamond\varphi \quad &\text{iff} \quad \exists i \geq 0.\ (\sigma|_i \vDash \varphi) \\ \sigma \vDash \Box\varphi \quad &\text{iff} \quad \forall i \geq 0.\ (\sigma|_i \vDash \varphi) \end{aligned}$$

You may use previously proven identities (both in this question and in lectures) to prove new ones.

Note that two temporal logic formulas $\phi$ and $\psi$ are logically equivalent, written $\varphi \Leftrightarrow \psi$, iff for all behaviours $\sigma$ it holds that:

$$\sigma \vDash \varphi \text{ if and only if } \sigma \vDash \psi$$

# General remark

For any proofs you submit: feel free to use any well-known laws about propositional logic, predicate logic or set theory without proof (e.g., the de Morgan laws). We're not reinventing that particular wheel in this course. If you do want to reinvent it, you can take COMP2111 :)