CS4211: Formal Methods for Software Engineering

Lecture Notes

Kevin Toh

Introduction to Formal Methods

- Requirements are difficult to define because its written in **Natural Language** which can be imprecise and ambiguous at times.
- As we cannot anticipate the ways a system may be used, written test cases only covers a small subset of use cases.
- We want to verify if a system always satisify a certain property, in possible all cases.
- In Formal Methods, we use **Mathematics** to define the **structure** and **behaviour** of our software because it is **precise** and **unambiguous**
- Eventually, we can use a model checker to automatically verify the software by checking if a certain property holds in all cases.

The Z Specification Language

- Based on set theory and mathematical logic
  - We will be doing a recap on predicates, set theory, functions and relations next.
- Uses schemas to declare object properties
  - Schemas are similar to defining the structure of a class and its properties in an Object-Oriented Programming Language.
- Uses operations to describe state transitions
  - Each object has a state representing the values it's properties hold at a certain moment in time.
  - Operations are similar to methods of a class.
  - Operations modify the state of an object.
  - We use predicates to describe state transitions in an operation.
- We can then proof that a certain property holds manually

Latex Setup

- The Latex package that we will be using is `zed-csp`
- **Setup Code:**

```
% For latex document
\documentstyle[12pt,zed]{article}

% For beamer slides
\usepackage{zed-csp}

\begin{document}
\end{document}
```

- Reference: https://sg.mirrors.cicku.me/ctan/macros/latex/contrib/zed-csp/zed2e.pdf

Recap on Predicates and Logic

Predicate

A statement that is either true or false.

1. There are 365 days in 2024. (**false**)
2. Let $P(x, y)$ be $x + y = 9$
   - $P(4, 5)$ is **true**.
   - $P(3, 7)$ is **false**.

Logic Operators

1. Not ($\neg$) Eg:
2. And ($\wedge$)
3. Or ($\vee$)
4. Implies ($\Rightarrow$)
5. Equivalence ($\Leftrightarrow$)

Recap on Quantifiers

1. Universal Quantifier ($\forall$)
   - Example: All natural numbers are greater than -1.
   - Mathematically, we would write $\forall\, n \in \mathbb{N}, n > -1$
   - In Z Specification, we would write $\forall\, n : \mathbb{N} \bullet n > -1$
   - $\forall\, n : \mathbb{N} \bullet n > 0$
   - In general, $\exists\, x : X \bullet P(x)$ abbreviates $P(a) \wedge P(b) \wedge P(c) \wedge \ldots$
2. Existential Quantifier ($\exists$)
   - Example: There exists a natural number more than 0.
   - In Z Specification, we would write $\exists\, n : \mathbb{N} \bullet n > 0$
   - In general, $\exists\, x : X \bullet P(x)$ abbreviates $P(a) \vee P(b) \vee P(c) \vee \ldots$

Differences between Mathematical Notations and Z Specification

- In Mathemical Notation, : or | means "such that" when used in Set Expressions.
- In Z Specification, : means "belongs to".
  - The difference between : and $\in$ will be explained later.
- In Z Specification, $\bullet$ means "such as" when writing predicates.

Recap on Set Theory

- A set is a collection of elements (or members)
  - Elements are not ordered: $\{a, b, c\} = \{b, a, c\}$
  - Elements are not repeated" $\{a, a, b\} = \{a, b\}$
  - Given Sets
    - $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ (The set of all natural numbers)
    - $\mathbb{N}_1 = \{1, 2, 3, \ldots\}$
    - $\mathbb{Z} = \{0, 1, -1, 2, -2, \ldots\}$ (The set of all integers)
    - $\mathbb{R}$ (The set of all real numbers)
    - $\emptyset$ (Empty Set: The set with no elements)
- Membership: $x \in \mathbb{X}$ is a predicate which is
  - true if x is in the set $\mathbb{X}$. Eg: $a \in \{a, b, c\}$
  - false if x is not in the set $\mathbb{X}$. Eg: $d \in \{a, b, c\}$

Difference between ':' and '$\in$'

Example: $\forall x : \mathbb{Z} \bullet x > 5 \Rightarrow x \in \mathbb{N}$

- $x : \mathbb{Z}$ declares a new variable x of type $\mathbb{Z}$
- $x \in \mathbb{N}$ is a predicate which is either true or false depending on the value of the declared x.

Recap on Set Theory

- Set Expressions
  - We can express a set by listing its elements if the set is finite and small.
    - $\{a, b, c, d\}$ is a finite set.
  - If a set is large or infinite, we can definite a set by giving a predicate which specifies precisely those elements in a set.
    - $\mathbb{N}$ is an infinite set.
    - The set of natural numbers less than 99 is $\{n : \mathbb{N} \mid n < 99\}$
    - In general the set $\{x : \mathbb{X} \mid P(x)\}$ is the set of elements of $\mathbb{X}$ for which predicate P is true.
- Set Examples
  - The set of even integers is $\{z : \mathbb{Z} \mid \exists\, k : \mathbb{Z} \bullet z = 2k\}$
  - The set of natural numbers which when divided by 7 leave a remaineder of 4 is $\{n : \mathbb{N} \mid \exists\, m : \mathbb{N} \bullet n = 7m + 4\}$
  - $\mathbb{N}$ is the set $\{z : \mathbb{Z} \mid z \geq 0\}$
  - $\mathbb{N}_1$ is the set $\{n : \mathbb{N} \mid n \geq 1\}$
  - If a, b are any natural numbers, then $a \mathinner{\ldotp\ldotp} b$ is defined as the set of all natural numbers between a and b inclusive.
    - $a \mathinner{\ldotp\ldotp} b$ is the set $\{n : \mathbb{N} \mid a \leq n \leq b\}$

Recap on Set Theory

- Subset ($\subseteq$): If S and T are sets, $S \subset T$ is a predicate equivalent to $\forall s : S \bullet s \in T$.
  - The following predicates are true:
    - $\{0, 1, 2\} \subseteq \mathbb{N}$
    - $2 \mathinner{..} 3 \subseteq 1 \mathinner{..} 5$
    - $\{a, b\} \subseteq \{a, b, c\}$
    - $\emptyset \subseteq X$ for any set X
    - $\{x\} \subseteq X \Leftrightarrow x \in X$
- Proper Subset ($\subset$): If S and T are sets, $S \subset T$ is a predicate equivalent to $S \subseteq T \wedge S \neq T$.
- Power Set ($\mathbb{P}$): If X is a set, $\mathbb{P}\ X$ (the power set of X) is the set of all subsets of X.
  - $A \in \mathbb{P}\ B = A \subseteq B$
  - The following predicates are true:
    - $\mathbb{P}\{a, b\} = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$
    - $\mathbb{P}\ \emptyset = \{\emptyset\} \neq \emptyset$
    - $1..5 \in \mathbb{P}\ \mathbb{N}$
    - $2..5 \in \mathbb{P}(1..5)$
  - If X has k elements, then $\mathbb{P}\ X$ has 2^k elements.

Recap on Set Theory

- Set Operations
  - Set Union: Suppose $S, T : \mathbb{P}\,X$ or $S \subseteq X, T \subseteq X$, then $S \cup T = \{x : X \mid x \in S \vee x \in T\}$
    - $\{a, b, c\} \cup \{b, g, h\} = \{a, b, c, g, h\}$
    - $A \cup \emptyset = A$ (for any set A)
  - Set Intersection: Suppose $S, T : \mathbb{P}\,X$, then $S \cap T = \{x : X \mid x \in S \wedge x \in T\}$
    - $\{a, b\} \cap \{b, c\} = \{b\}$
    - $\{a, b, c\} \cap \{d, g\} = \emptyset$ (disjoint sets)
    - $A \cap \emptyset = \emptyset$ (for any set A)
  - Set Difference: Suppose $S, T : \mathbb{P}\,X$, then $S - T = \{x : X \mid x \in S \wedge x \notin T\}$
    - $\{a, b, c\} - \{b, g, h\} = \{a, c\}$
    - $\mathbb{N}_1 = \mathbb{N} = \{0\}$
  - Cartesian Product: If A and B are sets, then $A \times B$ is the set of all ordered pairs (a, b) with $a \in A$ and $b \in B$.
    - $\{a, b\} \times \{a, c\} = \{(a, a), (a, c), (b, a), (b, c)\}$
  - Cardinality: $\#X$ is a natural number denoting the cardinality of (number of elements in) a finite set X.
    - $\#\{a, b, c\} = 3$

Recap on Relations

- A relation R from sets A to B, is declared as $R : A \leftrightarrow B$ is a subset of $A \times B$
- Example: $R = \{(c, x), (c, z), (d, x), (d, y), (d, z)\}$
  - The following predicates are equivalent
    1. $(c, z) \in R$
    2. $c \rightarrow z \in R$
    3. cRz
- **Domain:** $\mathrm{dom}\, R$ is the set $\{a : A \mid \exists\, b : B \bullet aRb\}$
- **Range:** $\mathrm{ran}\, R$ is the set $\{b : B \mid \exists\, a : A \bullet aRb\}$

Types in Z Specification

- Z specification language is **strongly typed**.
- Every expression is given a type.
- Any set can be used as a type.
- The following are equivalent declarations of variables x and y of types A and B respectively.
  - $(x, y) : A \times B$
  - $x : A, y : B$
  - $x, y : A$ (only when $B = A$)

Modelling using Z Specification

- When we write a program, we can write code procedurally, functionally or in an object oriented manner.
- Z Specification can help us model our code using two distinct sections.
  1. Declaration: To define variables.
  2. Predicate: Often used to define behaviours or invariants.
- **Example #1:** We can define the relation **divides** between two natural numbers.

  | divides: $\mathbb{N}_1 \leftrightarrow \mathbb{N}$
  | -----------------------------------
  | $\forall x : \mathbb{N}_1;\ y : \mathbb{N} \bullet x$ <u>divides</u> $y \Leftrightarrow \exists k : \mathbb{N} \bullet x \cdot k = y$

  Usage: 3 <u>divides</u> 6, $\neg$ (3 <u>divides</u> 7)
- **Example #2:** We can define the relation $\leq$ between two natural numbers.

  | _ <= _: $\mathbb{N} \leftrightarrow \mathbb{N}$
  | -----------------------------------
  | $\forall x, y : \mathbb{N} \bullet x <= y \Leftrightarrow \exists k : \mathbb{N} \bullet x + k = y$

  The relation $<=$ is the infinite subset of ordered pairs in $\mathbb{N} \times \mathbb{N}$.
  $\{(0,0),(0,1),(1,1),(0,2)(1,2),(2,2),\ldots\}$

Domain and Range Restriction

- Let A, B, R, S, T be sets.
- A is the domain set, B is the range set and R is the relation set.
- Note that S is a subset of the domain set and T is the subset of the range set.
- Suppose $R : A \leftrightarrow B, S \subseteq A$ and $T \subseteq B$.
  - **Domain Restriction:** $S \lhd R$ is the set $\{(a, b) : R \mid a \in S\}$
  - **Range Restriction:** $R \rhd T$ is the set $\{(a, b) : R \mid b \in T\}$
- Notice that both $S \lhd R \in A \leftrightarrow B$ and $R \rhd T \in A \leftrightarrow B$, meaning that both domain restriction and range restrictions are relations from sets A to B.
- If has_sibling: People $\leftrightarrow$ People then
  - female $\lhd$ has_sibling is the relationship is_sister_of.
  - has_sibling $\rhd$ female is the relationship has_sister.

Domain and Range Subtraction

- Let A, B, R, S, T be sets.
- A is the domain set, B is the range set and R is the relation set.
- Note that S is a subset of the domain set and T is the subset of the range set.
- Suppose $R : A \leftrightarrow B, S \subseteq A$ and $T \subseteq B$.
  - **Domain Subtraction:** $S \trianglelefteq R$ is the set $\{(a, b) : R \mid a \notin S\}$
  - **Range Subtraction:** $R \trianglerighteq T$ is the set $\{(a, b) : R \mid b \notin T\}$
- The following predicates are true.
  - $S \trianglelefteq R = (A - S) \triangleleft R$
  - $R \trianglerighteq T = R \triangleright (B - T)$
  - $S \trianglelefteq R \in A \leftrightarrow B$
  - $R \trianglerighteq T \in A \leftrightarrow B$
- If has_sibling: People $\leftrightarrow$ People then
  - female $\trianglelefteq$ has_sibling is the relationship is_brother_of.
  - has_sibling $\trianglerighteq$ female is the relationship has_brother.

Relational Image

- Suppose the relation $R : A \leftrightarrow B$ and $S \leftrightarrow A$
- $R(\!| S |\!) = \{b : B \mid \exists\, a : S \bullet aRb\}$
- $R(\!| S |\!) \leftrightarrow B$
- Example
  - $divides(\!| \{8, 9\} |\!) = \{x : \mathbb{N} \mid \exists\, k : \mathbb{N} \bullet x = 8 \cdot k \vee 9 \cdot k\} = \{0, 8, 9, 16, 18, \ldots\}$
  - $<= (\!| 3, 7, 21 |\!) = \{x : \mathbb{N} \mid x >= 3\}$
- In summary, the relational image returns the set of all elements $b \in B$ such that there exists an $a \in S$ with $(a, b) \in R$.
- The difference between relational image and range restriction is that range restriction returns the subset of R which are ordered pairs of (a, b) where $a \in A$ and $b \in B$ and the first element a of the ordered pair is in S. The relational image simply just returns the set of all second elements b.

Inverse and Relational Composition

- **Inverse:** R^{-1} is the set $\{(b, a) : B \times A \mid aRb\}$ or $R^{-1} \in B \leftrightarrow A$
  - $has\_sibling^{-1} = has\_sibling$
  - $divisor^{-1} = has\_divisor$
- Example: $succ^{-1} = \text{pred}$

```
| succ:   ℕ ↔ ℕ
|-----------------------------------
| ∀x, y : ℕ • x succ y ↔ x + 1 = y
```

- **Relational Composition ($⨾$)**
  - Suppose $R : A \leftrightarrow B$ and $S : B \leftrightarrow C$ are two relations.
  - $R \mathbin{⨾} S = \{(a, c) : A \times C \mid \exists\, b : B \mid aRb \wedge bSc\}$
  - $R \mathbin{⨾} S \in A \leftrightarrow C$
- Examples
  - $is\_parent\_of \mathbin{⨾} is\_parent\_of = is\_grandparent\_of$
  - $R^0 = id[A]$
  - $R^1 = R$
  - $R^2 = R \mathbin{⨾} R$
  - $R^3 = R \mathbin{⨾} R \mathbin{⨾} R$

Recap on Functions

- A (partial) function from a set A to a set B, denoted by $f : A \pfun B$ is a subset f of $A \times B$ with the property that for each $a \in A$, there is **at most one** $b \in B$ with $(a, b) \in f$.
- $\mathrm{dom}\, f$ is the set $\{a : A \mid \exists\, b : B \bullet (a, b) \in f\}$
- $\mathrm{ran}\, f$ is the set $\{b : B \mid \exists\, a : A \bullet (a, b) \in f\}$
- Suppose $f : A \pfun B$ and $a \in \mathrm{dom}\, f$, then $f(a)$ denotes the unique image $b \in B$ that a is mapped to by f.
- $(a, b) \in f$ is equivalent to $f(a) = b$
- **Total Function:** If the function $f : A \pfun B$ is a total function, then $f : A \rightarrow B$ if and only if $\mathrm{dom}\, f = A$

Function Overriding

- Suppose $f, g : A \nrightarrow B$, then $f \oplus g$ is the function $(\mathrm{dom}\, g \lhd f) \cup g$.
- The following predicates are true:
  1. $\mathrm{dom}\, f \oplus g = \mathrm{dom}\, f \cup \mathrm{dom}\, g$
  2. $a : \mathrm{dom}\, g \bullet (f \oplus g)(a) = g(a)$
  3. $\forall\, a : \mathrm{dom}\, f - \mathrm{dom}\, g \bullet (f \oplus g)(a) = f(a)$
  4. $\mathrm{f} \oplus \mathrm{g} \in a \nrightarrow b$
- Examples
  1. $\{a \rightarrow x, b \rightarrow y, c \rightarrow x\} \oplus \{a \rightarrow y\} = \{a \rightarrow y, b \rightarrow y, c \rightarrow x\}$
  2. double $\oplus$ root $= \{(0,0),(1,1),(2,4),(3,6),(4,2),\ldots\}$ (Note: $(4,8)$ was replaced with $(4,2)$ as the domain 4 is both in f and g, so the range was replaced with 2 that was in g)

Specifying Functions

1. Using a look-up table
   - If a function $f : A \nrightarrow B$ is finite (and not too large), we can specify the function explicity by listing all pairs (a, b) in the subset $A \times B$.
   - Example: PassportNo $\rightarrow$ Address

PassportNo	Address
A001017	77 Sunset Strip
...	...
G707165	19 Mail Street

2. Declaring Axioms: A function can be specified by giving a **predicate** determining which pairs (a, b) are in the function.
   - **Example: The root function that calculates the square root of a natural number**

```
| root ℕ ⇸ ℕ
|-----------------------------------
| dom root = {n : ℕ | ∃ m : ℕ • m² = n}
| ∀ n : dom root • (root(n))² = n
```

Specifying Functions

3. Using Recursion: For functions defined recursively in terms of itself.

```
| fact ℕ₁ ⇸ ℕ
|---------------------------------
| fact(1) = 1
| ∀ n : ℕ₁ − {1} • fact(n) = n ∗ fact(n − 1)
```

$$fact : \mathbb{N}_1 \nrightarrow \mathbb{N}$$
$$fact(1) = 1$$
$$\forall n : \mathbb{N}_1 - \{1\} \bullet fact(n) = n * fact(n-1)$$

4. Giving an Algorithm: A function $f : A \nrightarrow B$ is specified by an algorithm such that given any element a in the domain of f, the element $f(a)$ can be computed using the algorithm.

```
input n : N
var x, y: integer;
begin
    x := n; y:= 0;
    while x != 0 do
    begin
        x := x - 1; y:= y + 2
    end;
write(y)
end
```

```
| double:  ℕ → ℕ
| ------------------
| ∀ n : ℕ • double(n) = 2n
```

Sequences

- A sequence s of elements of a set A, denoted $s : \mathrm{seq}\, A$, is a function $s : \mathbb{N} \nrightarrow A$ where $\mathrm{dom}\, s = 1 \mathrel{..} n$ for some natural number n.
- **Example**
  - $< b, c, a, b >$ denotes the sequence (function) $\{1 \rightarrow b, 2 \rightarrow c, 3 \rightarrow a, 4 \rightarrow b\}$
  - The empty sequence is denoted by $<>$
- The set of all sequences of elements from A is denoted as $\mathrm{seq}\, A$ and is defined to be $\mathrm{seq}\, A = \{s : \mathbb{N} \nrightarrow A \mid \exists\, n : \mathbb{N} \bullet \mathrm{dom}\, s = 1 \mathrel{..} n\}$
- $\mathrm{seq}_1\, A = \mathrm{seq}\, A - \{<>\}$ is defined as the set of non-empty sequences.
- Since sequences are ordered mapping, $< a, b, a > \neq < a, a, b > \neq < a, b >$

Special Functions for Sequence

1. Concatenation
   - $< a, b > \frown < b, a, c > = < a, b, b, a, c >$
2. Head
   - ```
     | head:   seq₁ A → A
     |-----------------------------------
     |∀ s : seq₁ A • head(s) = s(1)
     ```
   - $head < c, b, b > = c$
3. Tail
   - ```
     | tail:   seq₁ A → seq A
     |-----------------------------------
     |∀ s : seq₁ A• < head(s) > ⁀ tail(s) = s
     ```
   - $tail < c, b, b > = < b, b >$
4. Filter
   - $< a, b, c, d, e, d, c, b, a > \upharpoonright \{a, d\} = < a, d, d, a >$
   - Filter only keeps the element in the specified set, preserves order in the original sequence and outputs a new sequence.

Z Specification

- As explained in an earlier slide, we can write Z Specification to formally specify requirements in **two distinct sections**
  1. Declaration: To define variables.
  2. Predicate: Often used to define behaviours or invariants.
- When we were specifying functions earlier, we were formally specifying them in the form of **axioms** in the **axiom environment**.
- Later we will introduce how to formally specify **schemas** which are used to specify relationships between variable values.
- There are two main type of schemas in the **schema environment**
  - State Schema
  - Operation Schema
- The **axiom environment** and **schema environment** are both available on the `zed-csp` package that was used to make these slides.
- This package allows us to render axioms and schemas using Z Specification in LATEX.

The zed-csp Package

We can use the `zed-csp` package to formally specify requirements in Z Specification.

1. Axiom Environment

$$\begin{array}{|l} limit : \mathbb{N} \\ \hline limit \leq 65535 \end{array}$$

```
limit: \nat
\where
limit \leq 65535
```

The zed-csp Package

2. Schema Environment

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{PhoneDB}} \\
\hline
known : \mathbb{P}\, NAME \\
phone : NAME \pfun PHONE \\
\hline
known = \operatorname{dom} phone \\
\hline
\end{array}
$$

```
known: \power NAME \\ phone: NAME \pfun PHONE
\where
known = \dom phone
```