COMP30026 Models of Computation

Sets

Harald Søndergaard

Lecture Week 6 Part 1

Semester 2, 2020

This Lecture is Being Recorded

Assignment 1

Assignment 1 was released on 1 September; it is due on 20 September. Solutions are submitted through Grok.

Matt has set up some supporting tools at `https://comp30026.far.in.net/puzzle`.

The "wires puzzle playground" will give you an appreciation of Challenge 3.

There is a leader-board for solutions to Challenge 4. That will give you a partial answer to the question "how few gates will do?"

Submission to the leader-board is optional and not related to submission for assessment (which must still happen on Grok).

Submission is also anonymous, and you can re-submit if you find a better design.

Set Theory

"Definition": (Georg Cantor) A set is a collection into a whole of definite, distinct objects of our intuition or of our thought. The objects are called the elements (members) of the set.

Notation: We write $a \in A$ to express that a is a member of set A.

Examples: $42 \in \mathbb{N}$ and $\pi \notin \mathbb{Q}$.

Principle of Extensionality: For all sets A and B we have

$$A = B \Leftrightarrow \forall x \; (x \in A \Leftrightarrow x \in B)$$

Set Notation

Small sets can be specified completely: $\{-2, -1, 0, 1, 2\}$, $\{\text{Huey}, \text{Dewey}, \text{Louie}\}$, $\{\}$. We often write the last one as $\emptyset$.

Note that, by the Principle of Extensionality, order and repetition are irrelevant, for example,

$$\{\{1, 2, 2\}, \{1\}, \{2, 1\}\} = \{\{1\}, \{1, 2\}\}$$

For large sets, including infinite sets, we have set abstraction:

If P is a property of objects x then the abstraction

$$\{x \mid P(x)\}$$

denotes the set of things x that have the property P. Hence $a \in \{x \mid P(x)\}$ is equivalent to $P(a)$.

Set Notation and Haskell's List Notation

Haskell's list notation is clearly inspired by set notation:

<table>
<tr><th>Haskell</th><th>Set notation</th></tr>
<tr><td><code>[]</code></td><td>$\{\}$</td></tr>
<tr><td><code>[1,2,3]</code></td><td>$\{1, 2, 3\}$</td></tr>
<tr><td><code>[n | n <- nats, even n]</code></td><td>$\{n \in \mathbb{N} \mid \mathit{even}(n)\}$</td></tr>
<tr><td><code>[f n | n <- nats]</code></td><td>$\{f(n) \mid n \in \mathbb{N}\}$</td></tr>
<tr><td><code>[1,3..]</code></td><td>$\{1, 3, \ldots\}$</td></tr>
</table>

The dot-dot notation here assumes some systematic way of generating all elements (an enumeration).

Well-Foundedness

Unfettered set abstraction is treacherous: There are sets for which $E = \{x \mid E(x)\}$ does not hold. Call a set S well-founded if there is no infinite sequence $S = S_0 \ni S_1 \ni S_2 \ni \cdots$, and consider the set W of all well-founded sets.

If $W \in W$ then $W \ni W \ni W \cdots$, and therefore $W \notin W$.

If $W \notin W$ then there is some infinite sequence $W = W_0 \ni W_1 \ni W_2 \cdots$. Since $W_1 \ni W_2 \ni W_3 \cdots$, W_1 is not well-founded, that is, $W_1 \notin W$. This contradicts $W = W_0 \ni W_1$.

Bertrand Russell's famous "barber paradox" similarly considers a set property $\mathrm{R} = \{x \mid x \notin x\}$ which leads to an inconsistent set theory:

$$R \in R \Leftrightarrow R \notin R$$

Sets and Types

One way (a crude way) to curb set theory so as to obtain consistency is to impose a system of types. In fact this was Russell's solution.

The purpose of the type discipline is to rule "$S \in S$" inadmissible, by insisting that S cannot inhabit type "t" and also "set of t".

Russell's type concept is the root of type disciplines used in many programming languages.

The Subset Relation

A is a subset of B iff $\forall x\ (x \in A \Rightarrow x \in B)$.

We write this as $A \subseteq B$.

If $A \subseteq B$ and $A \neq B$, we say that A is a proper subset of B, and write this $A \subset B$.

Do not confuse $\subseteq$ with $\in$. We have $\{1\} \subseteq \{1, 2\}$, but $\{1\} \notin \{1, 2\}$.

The Subset Relation Is a Partial Ordering

For all sets A, B, and C, we have

- $A \subseteq A$ (reflexivity)
- $A \subseteq B \wedge B \subseteq A \Rightarrow A = B$ (antisymmetry)
- $A \subseteq B \wedge B \subseteq C \Rightarrow A \subseteq C$ (transitivity)

These laws are easy to prove from the definition of $\subseteq$.

The three laws together state that $\subseteq$ is a partial ordering.

Special Sets

The empty set satisfies $\emptyset \subseteq A$ for every set A.

A set with just a single element is a singleton.

For example, $\{\{1,2\}\}$ is a singleton (its only element is a set).

The set $\{a\}$ should not be confused with its element a.

A set with two elements is a pair.

Ordinarily, and in programming languages, we refer to `(1,2)` as a pair, but in set theory we would call that an ordered pair.

Algebra of Sets

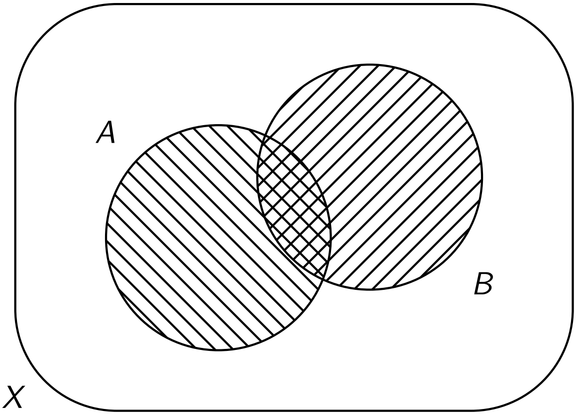
Let A and B be sets. Then

- $A \cap B = \{x \mid x \in A \wedge x \in B\}$ is the intersection of A and B;
- $A \cup B = \{x \mid x \in A \vee x \in B\}$ is their union;
- $A \setminus B = \{x \mid x \in A \wedge x \notin B\}$ is their difference; and
- $A \oplus B = (A \setminus B) \cup (B \setminus A)$ is their symmetric difference.

In the presence of a set X of which all sets are considered subsets, we also define

- $A^c = X \setminus A$ is the complement of A.

Venn Diagrams



Some Laws

Absorption: $A \cap A = A$
$A \cup A = A$

Commutativity: $A \cap B = B \cap A$
$A \cup B = B \cup A$

Associativity: $A \cap (B \cap C) = (A \cap B) \cap C$
$A \cup (B \cup C) = (A \cup B) \cup C$

Distributivity: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

More Laws

Double complement: $A = (A^c)^c$

De Morgan: $(A \cap B)^c = A^c \cup B^c$
$(A \cup B)^c = A^c \cap B^c$

Duality: $X^c = \emptyset$ and $\emptyset^c = X$

Identity: $A \cup \emptyset = A$ and $A \cap X = A$

Dominance: $A \cap \emptyset = \emptyset$ and $A \cup X = X$

Complementation: $A \cap A^c = \emptyset$ and $A \cup A^c = X$

Subset Equivalences

Subset characterisation: $A \subseteq B \equiv A = A \cap B \equiv B = A \cup B$

Contraposition:

$$A^c \subseteq B^c \equiv B \subseteq A$$
$$A \subseteq B^c \equiv B \subseteq A^c$$
$$A^c \subseteq B \equiv B^c \subseteq A$$

Subset Equivalences

Subset characterisation: $A \subseteq B \equiv A = A \cap B \equiv B = A \cup B$

Contraposition:

$$A^c \subseteq B^c \equiv B \subseteq A$$
$$A \subseteq B^c \equiv B \subseteq A^c$$
$$A^c \subseteq B \equiv B^c \subseteq A$$

All very similar to the equivalences we saw for propositional logic—just substitute $\neg$ for complement, $\wedge$ for $\cap$, $\vee$ for $\cup$, $\Rightarrow$ for $\subseteq$, $\bot$ for $\emptyset$, and $\top$ for X.

Powersets

The powerset $\mathcal{P}(X)$ of the set X is the set $\{A \mid A \subseteq X\}$ of all subsets of X.

In particular $\emptyset$ and X are elements of $\mathcal{P}(X)$.

If X is finite, of cardinality n, then $\mathcal{P}(X)$ is of cardinality 2^n.

Generalised Union and Intersection

Suppose we have a collection of sets A_i, one for each i in some (index) set I. For example, I may be $\{1..99\}$, or I may be infinite.

The union of the collection is

$$\bigcup_{i \in I} A_i = \{x \mid \exists i \ (i \in I \ \wedge x \in A_i)\}$$

The intersection of the sets is

$$\bigcap_{i \in I} A_i = \{x \mid \forall i \ (i \in I \ \Rightarrow x \in A_i)\}$$

Ordered Pairs

Can we capture the notion of ordered pairs (a, b) with set-theoretic notions? We want this to hold:

$$(a, b) = (c, d) \Leftrightarrow a = c \wedge b = d$$

We can achieve this by defining

$$(a, b) = \{\{a\}, \{a, b\}\}$$

Hence we can freely use the notation (a, b) with the intuitive meaning.

Cartesian Product and Tuples

The Cartesian product of A and B is defined

$$A \times B = \{(a, b) \mid a \in A \land b \in B\}$$

We define the set A^n of n-tuples over A as follows:

$$\begin{array}{lcl} A^0 & = & \{\emptyset\} \\ A^{n+1} & = & A \times A^n \end{array}$$

Of course we shall write (a, b, c) rather than $(a, (b, (c, \emptyset)))$.

Some Laws Involving Cartesian Product

$$
\begin{aligned}
(A \times B) \cap (C \times D) &= (A \times D) \cap (C \times B) \\
(A \cap B) \times C &= (A \times C) \cap (B \times C) \\
(A \cup B) \times C &= (A \times C) \cup (B \times C) \\
(A \cap B) \times (C \cap D) &= (A \times C) \cap (B \times D) \\
(A \cup B) \times (C \cup D) &= (A \times C) \cup (A \times D) \cup (B \times C) \cup (B \times D)
\end{aligned}
$$

Relations

An n-ary relation is a set of n-tuples.

$$\left\{ \begin{array}{l} (\mathit{MY255}, \mathit{Lagos}, \mathit{Lusaka}, 1755), \\ (\mathit{ZA942}, \mathit{Lima}, \mathit{London}, 1015), \\ (\mathit{BB114}, \mathit{Lyon}, \mathit{Lodz}, 2220) \end{array} \right\}$$

That is, the relation is a subset of some Cartesian product $A_1 \times A_2 \times \cdots \times A_n$.

Or equivalently, we can think of a relation as a function from $A_1 \times A_2 \times \cdots \times A_n$ to $\{0, 1\}$.

After the Break

We take a closer look at binary relations, and a special variant of these, namely functions.