COMP11120 Notes

Todd Davies, Chris Williamson

November 26, 2013

Note, extra space has been allocated for the right hand margin to allow for more extensive margin notes. Also, it gives you space to make your own annotations and perhaps try some problems of your own.

Contents

1 Discrete Structures

1.1 Terminology

A *structure* consists of certain *sets*. It also contains *elements* of these sets, *operations* on these sets and *relations* on these sets.

1.2 Number systems to learn

The following number must be learnt:

$\mathbb{N}$	The set of natural numbers (all whole numbers from 0 to ∞)
$\mathbb{Z}$	The set of integers (all whole numbers from $-\infty$ to ∞)
$\mathbb{Q}$	The set of rational numbers (any integer divided by any other integer e.g. $\frac{5}{4} = 1.25$)
$\mathbb{R}$	The set of real numbers (all finite and infinite decimal numbers)

1.2.1 Operations

Each number system has a set of valid operations that can be performed on elements in that system. Number systems only contain operations that will produce an output that is still within the number system.

For example, the number system $\mathbb{N}$ contains the operations of addition and multiplication. This is because the summation of any two positive integers will *always* be a member of $\mathbb{N}$, and the same goes for multiplication.

However, you may be wondering why subtraction and division aren't included in this number system. This is because for some numbers, the result of subtraction or division won't be inside the set $\mathbb{N}$. An example of this would be subtracting 4 from 2. Even though both of the operands are inside $\mathbb{N}$, the answer isn't.

Different sets may have different operations available. For example, you can concentrate any two members of the set $\mathbb{S}tring$ and end up with another $\mathbb{S}tring$.

Types of operation Operations that have two operands either side of them are called *infix* operations. An example of an infix operation is addition. Infix operations are also referred to as *binary* operations since they have *two* operands.

Commutativity If an operation is commutative, the order of the operands doesn't matter. For example, addition is commutative since:

$$a + b = b + a$$

Subtraction however, isn't commutative:

$$a - b \neq b - a$$

Associativity An operation is associative if inserting or changing brackets doesn't change the outcome of the operation. For example, multiplication is associative since:

$$(a \times b) \times c = a \times (b \times c)$$

An operation may only be commutative or associative if it is commutative or associative for *all* elements of the set that the operation supports.

Distinguished elements A set may contain distinguished elements that have strange effects on certain operations in the set. An example is the number 1. If we multiply *something* by 1, then the result will always be the same *something*. The same goes for 0 with addition. Because of this, we refer to 1 and 0 as distinguished elements of the set $\mathbb{Z}$.

1.2.2 Relations

Each of the sets $\mathbb{N}, \mathbb{Q}, \mathbb{Q}, \mathbb{R}$ carries binary comparison relations $\leq$ and $<$. Different sets (such as $\mathbb{S}tring$) may have other relations, such as:

- Is a section of
- Is an initial section of
- Occurs in

All relations return values in the set $\mathbb{B}ool$.

1.3 Bases

Conventionally, we count using base 10. Base 10 includes, you guessed it, ten different symbols from 0 through to 9.

Sometimes however, it is convenient to count using different bases. Popular bases include:

Base n	**Member symbols**	**Name**
$n = 2$	$\mathbb{Z}_2 = \{0, 1\}$	Binary
$n = 8$	$\mathbb{Z}_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$	Octal
$n = 10$	$\mathbb{Z}_{10} = \{0, 1, 2, 3, 4, 5, 6, 7\}$	Decimal/Denary
$n = 16$	$\mathbb{Z}_{16} = \{0, 1, 2, \ldots, 9, A, B, C, D, E, F\}$	Hexadecimal

1.3.1 How to read numbers in any given base

The formula for reading a number in a given base is as follows:

$$\sum_{i=0}^{k} a_i b^i$$

Where the number you're trying to read takes the form $a_k, a_{k-1}, \ldots, a_2, a_1, a_0$ and b is the base you're using.

Example 1 Lets apply the formula to the base 10 number 27385:

$$\begin{aligned}
27385 &= (5 \times 10^0) + (8 \times 10^1) + (3 \times 10^2) + (7 \times 10^3) + (2 \times 10^4) \\
&= (5 \times 1) + (8 \times 10) + (3 \times 100) + (7 \times 1000) + (2 \times 10000) \\
&= 5 + 80 + 300 + 7000 + 20000 \\
&= 27385
\end{aligned}$$

Example 2 Lets apply the formula to the base 16 number $F00BA4$:

$$\begin{aligned}
F00BA4 &= (4 \times 16^0) + (A \times 16^1) + (B \times 16^2) + (0 \times 16^3) + (0 \times 16^4) + (F \times 16^5) \\
&= (4 \times 16^0) + (10 \times 16) + (11 \times 256) + (0 \times 4096) + (0 \times 65536) + (15 \times 1048576) \\
&= 4 + 160 + 2816 + 0 + 0 + 15728640 \\
&= 15731620
\end{aligned}$$

1.3.2 Changing from base 10 to base n

In order to change into base n from base 10, we just repeatedly divide by n and use the remainder as the value for base n. Here are a few examples:

Example 1 Convert 893 into base 2.

$$\begin{array}{rcll}
893 \div 2 & = & 446 & \text{r1} \\
446 \div 2 & = & 223 & \text{r0} \\
223 \div 2 & = & 111 & \text{r1} \\
111 \div 2 & = & 55 & \text{r1} \\
55 \div 2 & = & 27 & \text{r1} \\
27 \div 2 & = & 13 & \text{r1} \\
13 \div 2 & = & 6 & \text{r1} \\
6 \div 2 & = & 3 & \text{r0} \\
3 \div 2 & = & 1 & \text{r1} \\
1 \div 2 & = & 0 & \text{r1}
\end{array}$$

Reading up from the bottom, we can see that the binary (base 2) representation is 1101111101.

Example 2 Convert 893 into base 9.

$$\begin{array}{rcll}
893 \div 9 & = & 99 & \text{r2} \\
99 \div 9 & = & 11 & \text{r0} \\
11 \div 9 & = & 1 & \text{r2} \\
1 \div 9 & = & 0 & \text{r1}
\end{array}$$

Reading up from the bottom, we can see that the nonal (base 9) representation is 1202.

Example 2 Convert 893 into base 16.

$$\begin{array}{rcll}
893 \div 16 & = & 55 & \text{r13} \\
55 \div 16 & = & 3 & \text{r7} \\
3 \div 16 & = & 0 & \text{r3}
\end{array}$$

Reading up from the bottom, we can see that the hexadecimal (base 16) representation is $3, 7, 13$ or $37D$.

1.4 A structure for the integers

The set $\mathbb{Z}$ of all integers $\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$ includes the subset $\mathbb{N}$ of all natural numbers together with the negative integers. We will be using three basic binary operations on the carrier set $\mathbb{Z}$: addition, multiplication and subtraction.

The standard notation for these operations is $+$, $\times$ and $-$ and they are used as infix operations.

Sometimes different notations are used, for example, in programming * is usually used, or there is a convention to write xy as an abbreviation for $x \times y$.

Both + and x are **commutative** but - is not.

1. For all integers x,y both, $x + y = y + x$ and $xy = yx$
2. There are integers x,y with $x - y \neq y - x$.

Commutativity ensures that for most purposes the order in which the two arguments are consumed is irrelevant.

Both + and x are **associative** but - is not.

1. For all integers x,y,z both $(x + y) + z = x + (y + z)$, $(xy)z = x(yz)$.
2. There are integers x,y,z with $(x - y) - z \neq x - (y - z)$.

Associativity ensures that for most purposes brackets are not needed to punctuate an expression. For instance, we can make sense of $x + y + z$ and xyz since it doesn't matter where the brackets are, the resulting values are the same.

Below are the definitions of commutativity and associativity.

N.b. Commutativity and associativity don't always have to go together. An example is the average of two numbers, what is that?

1. $\circledast$ is **commutative** if and only if

$$a1 \circledast a2 = a2 \circledast a1$$

   for all $a1, a2, \in, A$

2. $\circledast$ is **associative** if and only if

$$(a1 \circledast a2) \circledast a3 = a1 \circledast (a2 \circledast a3)$$

   for all $a1, a2, a3, \in, A$

Some numbers have special effects on operations in a set. Some examples are 0 and 1. When 0 is added to a number, the result is still that number, and when multiplying a number by 1, the result is still that number. This means for the set $\mathbb{Z}$, 0 and 1 are neutral elements for the operations addition and multiplication respectively.

1.5 The formal language

We use the formal language to talk about $\mathbb{Z}$. Expressions in the formal language are built up from atoms in a recursive fashion, so an expression may be $(x \circ y)$ where x and y are expressions and

$\circ$ is one of the three operations $(+, -, \times)$. The brackets are important in the formal language, since they ensure that strings can only be parsed in one way.

Strings that contain literals that aren't neutral elements or have invalid bracketing are not valid in the formal language. Examples may include:

$$(x + y + z)$$
$$(x + 2)$$
$$x((y$$

We can use a parse tree to show how an expression is built up from it's identifiers $(a, b, c \ldots)$, constants $(0, 1 \ldots)$ and operations $(+, -, \times \ldots)$.

For example, the parse tree of $(x \times (y + z))$ is:

$$\dfrac{x \quad \dfrac{y \quad z}{(y+z)}\,(+)}{(x \times (y+z))}\,(\times)$$

Often, especially with large parse trees, it's a pain to write so many identifiers. Because of this, it is a convention to replace identifiers with dots after they've been used once, like so:

$$\dfrac{x \quad \dfrac{y \quad z}{\cdot}\,(+)}{\cdot}\,(\times)$$

In order to parse a parse tree, we must assign an appropriate value to each of the 'leaves' of the tree (i.e. all the identifiers) and let the values trickle down towards the root node of the tree where the evaluated answer will appear.

Lets use the above example again. Let $x = 4$, $y = 6$ and $z = 2$:

$$\dfrac{4 \quad \dfrac{6 \quad 2}{8}\,(+)}{32}\,(\times)$$

At this point, parse trees may seem very pointless, but this is because we're doing very simple arithmetic. However, when we start to define other operators that do unfamiliar things or don't use the *infix* notation, then using parse trees can be a big help!

The infix notation is when an operator is placed between two operands, e.g. $2 + 2$. Other notations include the *prefix* and *postfix* notations.

1.6 The properties of sets

All sets have some properties in common that we can use to manipulate them. Examples include membership, equality, inclusion etc.

1.6.1 Set membership

To indicate that an element is a member of a set, we use the $\in$ symbol. For example, to say that the element *true* is a member of the set $\mathbb{B}ool$, we would write:

$$true \in \mathbb{B}ool$$

Interestingly, we can parse this into English in many ways though. It could mean any of the following things:

- *true* is an element of the set $\mathbb{B}ool$
- *true* is an member of the set $\mathbb{B}ool$
- *true* is contained in $\mathbb{B}ool$
- $\mathbb{B}ool$ contains *true*

Conversely, to indicate that an element is not a member of a set, we use the symbol $\notin$:

$$sheep \notin \mathbb{B}ool$$

Note that there is no concept of *order* or *repetition* in sets. This means that the following sets are all equal:

$$\{1, 2, 3\}$$
$$\{2, 3, 1\}$$
$$\{2, 3, 1, 2\}$$
$$\{3, 3, 3, 3, 3, 2, 1\}$$

1.6.2 Set equality

Sets are equal if they have exactly the same members. Note that as far as sets are concerned, duplicate members are treated as just one member.

The notation for set equality is very easy. To say the set X is equal to the set Y, we write:

$$X = Y$$

However, you must understand that this is only true if for each element a in X that element will also be contained in Y and for each element a in Y, that element will also be contained within X:

$$\text{For each } a, a \in X \leftrightarrow a \in Y$$

Sets with different descriptions can still be equal. Convince yourself that the set of integers where $y^3 < y$ is equal to the set of integers where $x < -1$

1.6.3 Set inclusion

If one set is a subset of another set, all the members of the first set are also found within the second set. In mathematical terms:

$$\text{For each } a, a \in X \rightarrow a \in Y$$

The notation for inclusion is $\subseteq$, so in the above example, we would write:

If $X \subseteq Y$ and $Y \subseteq X$ then what else an we say about the relationship between X and Y?

$$X \subseteq Y$$

1.6.4 The empty set

The empty set is a set that contains no members at all. It's symbol is $\emptyset$.

Because the empty set has no members, it is a subset of all other sets:

This is because otherwise $xin\emptyset$ would be true for some x where $x \notin A$. This is impossible since there are no elements in the $\emptyset$.

$$\emptyset \in A$$

1.6.5 Singleton sets

For any entity a, we can form a set consisting only of a:

$$\{a\}$$

Be aware, a singleton set is not the same as the element contained within the set:

$$a \neq \{a\}$$

1.6.6 Set union

There are several ways of combining multiple sets together to create another set. One such method is set union, the symbol of which is $\cup$. The union of two sets is a set containing all the members of *both* sets. For example:

$$A = \{1, 3, 5, 7, 9\}$$

$$B = \{1, 1, 2, 3, 5, 8, 13\}$$

$$A \cup B = \{1, 2, 3, 5, 7, 8, 9, 13\}$$

We could also define the union of two sets mathematically, like so:

$$x \in A \cup B \leftrightarrow x \in A \text{ or } x \in B$$

1.6.7 Set intersection

Another way of combining sets is intersection. Intersecting two sets will produce a set of elements that belong to both of the sets. The symbol for intersection is $\cap$.

If we defined intersection mathematically, we would do so like this:

$$x \in A \cap B \leftrightarrow x \in A \text{ and } x \in B$$

1.6.8 Relative complement

The relative complement of two sets is all the members in the first set that aren't members of the second set. The symbol for the relative complement is $-$. Defined mathematically, we get:

$$x \in A - B \leftrightarrow x \in A \text{ and } x \notin B$$

1.6.9 The universal set

The universal set contains *all of the possible elements*. The notation we use to describe the universal set is S. We can define the De Morgan's laws using the universal set:

$$X' = S - X$$

1.7 Boolean algebra of sets

If we have a universal set, S, and consider only subsets of S. Then these are all the things that we have:

- Two distinguished subsets $\emptyset$ and S
- A unary operation $(.)'$ on such subsets
- Two binary operations; $\cap$ and $\cup$ on such subsets

This structure of operations and sets is known as **Boolean algebra**. Boolean algebra has many basic properties that we can exploit in order to manipulate expression into other forms. Here is a list to be learnt.

There are flashcards on these too.

$X \cup S = S$	extreme	$X \cap \emptyset = \emptyset$
$X \cup \emptyset = X$	neutral	$X \cap S = X$
$X \cup Y = Y \cup X$	commutative	$X \cap Y = Y \cap X$
$X \cup (Y \cup Z) = (X \cup Y) \cup Z$	associative	$X \cap (Y \cap Z) = (X \cap Y) \cap Z$
$X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$	distributive	$X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$
$X \cup X = X$	idempotent	$X \cap X = X$
$X \cup (X \cap Y) = X$	absorbsion	$X \cap (X \cup Y) = X$
$(X \cup Y)' = X' \cap Y'$	De Morgan	$(X \cap Y)' = X' \cup Y'$
$X \cup X' = S$	complement	$X \cap X' = \emptyset$
$X'' = X$	involution	$X = X''$

2 Propositional Logic

2.1 The logical connectives

Most of these are very similar to the properties of sets that we came across in the boolean algebra section. Learn these:

We use it inclusive or in propositional logic, not *exclusive* or.

Iff is an abbreviation for *if and only iff*.

Name	Symbol	Meaning
Negation	$\neg$	not
Conjunction	$\wedge$	and
Disjunction	$\vee$	or
Implication	$\implies$	implies
Bi-implication	$\iff$	iff

All of these symbols are used as infixes, and so go in between two parameters. An exception to this however, is negation, which is a unary symbol, and is placed as a prefix before it's argument.

If a connective takes one parameter, then it's arity is 1, if it takes two then it has an arity of 2 etc etc

2.1.1 The truth tables of the logical connectives

Using a truth table, we can fully describe the behaviour of the connectives we have just described in the previous section:

p_1	p_2	$\neg p_1$	$p_1 \wedge p_2$	$p_1 \vee p_2$	$p_1 \implies p_2$	$p_1 \iff p_2$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

A note about implication: In order to understand why the truth table for implication is as stated above, consider this example:

$$x > 3 \text{ implies } x > 1$$

Now take a look at the truth table for this expression:

x	$x > 3$	$x > 1$	$x > 3 \implies x > 1$
4	T	T	T
2	F	T	T
0	F	F	T

As you can see, there is no way to make the first expression ($x > 3$) true, but the second expression ($x > 1$) false. Henceforth, the first expression implies the second expression.

2.2 The formulae of propositional logic

In order to build up a piece of propositional logic, we must construct an expression from *atomic formulae* and connectives.

An example of an atomic formula is p, or maybe r_2. It's any single boolean variable.

There are three rules we can use to generate a formulae in PL:

1. Every atomic formula is a formula of PL

2. If A is a formula, then so is $\neg A$
3. If A, B are formulae, then so are $(A \wedge B)$, $(A \vee B)$, $(A \implies B)$, $(A \iff B)$

Syntactic conventions are of course in widespread use.

For example, when you have written an expression, it is usual to leave off the outermost parentheses, so $(p \wedge (r \vee q))$ would become $p \wedge (r \vee q)$.

This second convention is adopted for all associative operations, not just conjunction and disjunction.

It is also common to leave out parentheses from repeated uses of conjunction or disjunction, so $p \wedge (q \wedge r)$ would become $p \wedge q \wedge r$.

2.3 Truth valuations

A truth valuation is a list of allocations that define the values of variables in an expression, where:

$$\{p_1 = x_1, \ldots, p_n = x_n\}$$

An example truth valuation for the expression $p \wedge r$ might be $\{p = T, r = F\}$, for which the end value would be F.

In order to determine the end value of the PL formula, you must first construct a truth valuation of it's component atoms, and only then can the formula be evaluated.

2.4 Tautologies and contradictions

If the outcome of a formula is always T for every possible truth valuation, then the formula is said to be a **tautology**. An example of such an expression is $p \vee \neg p$.

In contrast, a formula that will yield an outcome of F for every possible truth valuation is named a **contradiction**. Such an example may be $p \wedge \neg p$.

If an expression is not a tautology and isn't a contradiction either, then it is named **satisfiable**.