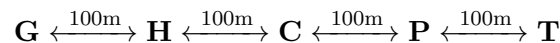In preparation for Occupy Central, one of the organizers – let's call him Bob – has asked the public to donate rotten tomatoes to target unpopular officials. On the morning of the protest the following amounts of rotten tomatoes have been gathered at five collection points around town:

location	amount gathered
Government House	3 crates
HSBC	1 crate
City Hall	7 crates
PLA Headquarters	2 crates
Tamar	1 crate

In anticipation of the number of targets that are expected to show up at each location, Bob would like to redistribute the crates like this:

location	amount needed
Government House	2 crates
HSBC	4 crates
City Hall	1 crate
PLA Headquarters	1 crate
Tamar	6 crates

Moving crates amidst the chaos of Occupy Central is expensive. The distance between each pair of consecutive locations is about 100 meters.

$$\mathbf{G} \xleftrightarrow{100\text{m}} \mathbf{H} \xleftrightarrow{100\text{m}} \mathbf{C} \xleftrightarrow{100\text{m}} \mathbf{P} \xleftrightarrow{100\text{m}} \mathbf{T}$$
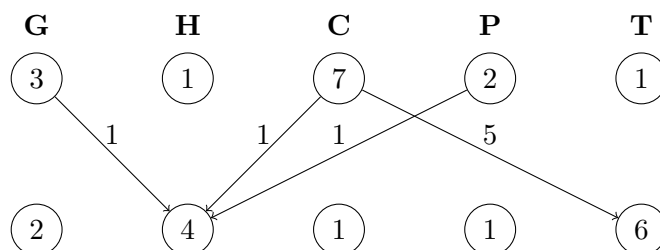
A porter asks for 100 HKD for every crate carried over a distance of 100 meters. If he were to carry, for example, two crates from **C**ity Hall to **T**amar – at a distance of 200 meters – that would cost Bob 400 HKD. Being short on budget, Bob asks Jason, a student at HKU, to come up with an economical way of moving the crates around.
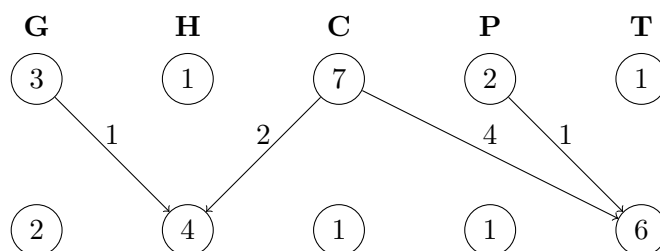
You, a promising young engineer in pursuit of an exciting assignment, overhear Jason shouting these instructions to the porters:

> Move 5 crates from City Hall to Tamar. Then take one crate from each of Government House, City Hall, and the PLA Headquarters and move those to HSBC.

Jason's plan involves moving two crates at a distance of 100 meters and another six crates at a distance of 200 meters, for a total cost of $2 \times 100 + 6 \times 200 = 1,400$ Hong Kong dollars. You take out a piece of paper and make a quick sketch of it:

```
        G           H           C           P           T
       (3)         (1)         (7)         (2)         (1)
              1           1     1           5
       (2)         (4)         (1)         (1)         (6)
```

You notice something fishy: The crates out of City Hall and PLA Headquarters cross paths. This is clearly wasteful, so why not move the crates around like this instead:


```
        G           H           C           P           T
       (3)         (1)         (7)         (2)         (1)
              1           2           4     1
       (2)         (4)         (1)         (1)         (6)
```

Indeed, now we are moving 4 crates at 100 meters and 4 crates at 200 meters for a total cost of 1,200 HKD! Bursting with excitement, you go and show your improvement to Bob. He is happy that you will save him 200 HKD. Then he asks: "Since you are so bright, can you save me another 200 HKD and do the whole operation for 1,000?"

You scratch your head for a few minutes, but you are at a loss; nothing seems to work. However, you are a bit embarassed to admit your failure to Bob. What if your rival Jason impresses Bob with an even better solution?

Proofs

Much of the mathematics you study in school is about *calculating* things. In first grade you learn how to add single digit numbers. Later you move on to larger numbers and more complicated calculations, like multiplications, divisions, and square roots. In high school, you may be calculating roots of quadratic equations, sines and cosines, and doing some complex number algebra. At university, you take derivatives and compute integrals, solve systems of linear equations and differential equations.

In order to be a good engineer, you certainly need to be a master at various calculations that come up routinely in your discipline. But calculating is not enough. You will need to learn to set up and solve problems in a confident manner.

In Bob's crate transportation task, a mediocre engineer (Jason) might be satisfied with a solution that "feels" good to him. A great engineer, like you, wants more: You want to be sure that your solution is the best possible one. For this, calculations are not particularly helpful; you need to reason things out.

After thinking for a while, you are quite sure that it is impossible to spend less than 1,200 HKD on moving the crates around. But how do you explain this to Bob?

Here is how. You notice that at the Government House, there are initially three crates and you need to end up with two; so *no matter how the crates are rearranged*, at least one crate will have to be carried out of Government House:

G	H	C	P	T
3	1	7	2	1

1 →

2	4	1	1	6
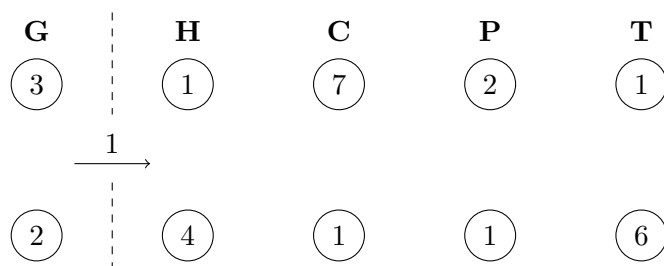
Next, if you add the number of crates at Government House and HSBC, there are 4 available but 6 are needed; so no matter how the crates are rearranged, at least two will have to be brought in from City Hall or beyond:

G	H	C	P	T
3	1	7	2	1

1 →   ← 2

2	4	1	1	6
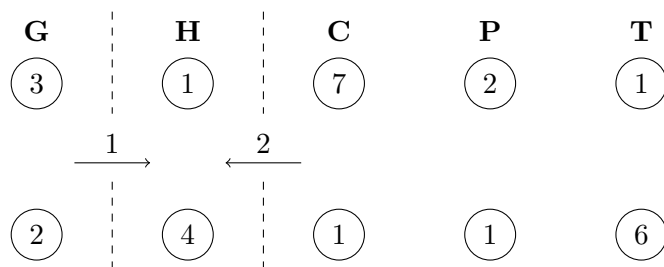
Continuing your reasoning in this way, you come up with the following picture:

G	H	C	P	T
3	1	7	2	1

1 →   ← 2   4 →   5 →

2	4	1	1	6
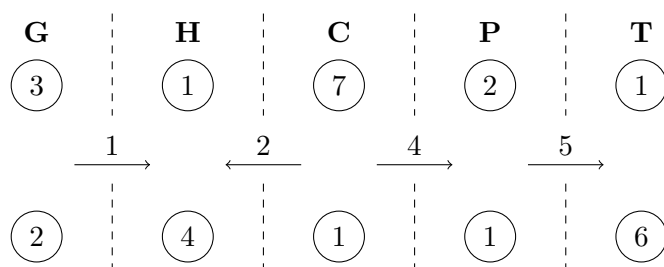
It is now clear that Bob must spend at least 100 HKD moving crates between Government House and HSBC, at least 200 HKD between HSBC and City Hall, and so on. Adding all the expenses together amounts to 1,200 HKD. Your solution was indeed the best possible.

What we just saw is an example of a *proof*: A deduction of an interesting *proposition* ("No matter how the crates are rearranged, it will cost Bob at least 1,200 HKD") by a sequence of clear, rigorous

logical deductions. The ability to come up with proofs and present them clearly is important for computer science and other engineering disciplines. In the next few weeks, we will talk about various types of proofs in some detail.

1 Propositions

A *proposition* is a statement that is either true or false. Here are two examples of propositions.

> $1 + 1 = 2$.
> Tuesday is the day after Tuesday.

The first proposition is about numbers; the second one is about days of the week. The first proposition is true; the second one is false. This can be figured out by most people with a first grade education (where we learn the meaning of "1", "+", "Tuesday", and so on).

Telling whether a proposition is true or false is not always easy, but the *meaning* of a proposition must always be clear. For example, consider the following statements:

> Taxi drivers [are] quoting prices four times the metered fare.
> Humans can [...] outwit the machines, but maybe not for much longer.
> A friendly neighbourhood is one in which people communicate effectively and take care not to cause disputes.

These are all taken from the August 31, 2014 edition of the *South China Morning Post.* What do they mean exactly? Is every taxi driver is quoting a higher fare? Is the quoted fare exactly four times higher than the regular one? For the average newspaper reader, this kind of ambiguity is tolerable and even desirable, but it is not acceptable in mathematics and much of computer science (in particular, when writing computer programs). We will not call such statements propositions.

Propositional logic and truth tables

We can modify and combine propositions using *operators* such as AND, OR, NOT. For example, the proposition

$$\text{NOT } (1 + 1 = 3)$$

is true, while the proposition

$$(1 + 1 = 2) \text{ AND (Tuesday is the day after Tuesday)}$$

is false.

In general, given an arbitrary proposition P, we can build the proposition NOT P. The proposition NOT P is false when P is true, and true when P is false. We can describe the effect of NOT compactly in a *truth table*:

P	NOT P
T	**F**
F	**T**

Given two propositions P and Q, we can form the *compound propositions P AND Q, P OR Q*. Here are their truth tables:

P	Q	P AND Q
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

P	Q	P OR Q
T	**T**	**T**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

This is a different from the way the connective OR is sometimes used in common English. When you see a dinner set in a restaurant that comes with "coffee or tea", it is usually understood that you cannot have both. In contrast, in mathematics and computer programming, P OR Q is true when P is true and Q is true.

The English meaning of "You can have coffee or tea with your dinner" is captured by the logical connective XOR, which stands for "exclusive or":

P	Q	P XOR Q
T	**T**	**F**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

We say P and Q are *logically equivalent* if they take the same truth value. The operator IFF (short for "if and only if") describes logical equivalence:

P	Q	P IFF Q
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**T**

We can go on and on, but in fact every compound proposition is logically equivalent to a propositional formula that uses only the connectives AND, OR, and NOT. For example

P XOR Q is logically equivalent to   ((NOT P) AND Q) OR (P AND (NOT Q)).

One way to verify this is to compute the truth table of the second formula and compare it to the truth table for XOR:

P	Q	NOT P	(NOT P) AND Q	NOT Q	P AND (NOT Q)	((NOT P) AND Q) OR (P AND (NOT Q))
T	**T**	**F**	**F**	**F**	**F**	**F**
T	**F**	**F**	**F**	**T**	**T**	**T**
F	**T**	**T**	**T**	**F**	**F**	**T**
F	**F**	**T**	**F**	**T**	**F**	**F**

Hold on a second; I just made a statement:

> Every compound proposition is logically equivalent to a propositional formula that uses only the operators AND, OR, and NOT.

Is this statement a proposition? We did not learn what a "propositional formula" is in first grade. Yet this term has a precise and unambiguous meaning. We cannot explain it today but we will be able to do so in a few lectures. Indeed, this statement is a proposition (about propositions) and it is true.

2 Quantifiers

A *predicate* is a proposition whose truth may depend on one or more *free variables*. For example, "n is even" is a predicate (about integers) with free variable n. It is true when $n = 2$ and false when $n = 3$. The predicate "$n = 2 \times m$" (also about integers) is true when $n = 4, m = 2$ and false when $n = 2, m = 4$.

A predicate can be turned into a proposition by *quantifying* over the free variables. For example, the statement

> For all integers n, n is even

is a proposition. This proposition is false because when $n = 3$, the proposition "n is even" becomes false. On the other hand, the proposition

> There exists an integer n such that n is even

is true because when $n = 2$, the predicate "n is even" becomes true.

A proposition like "10 is even" can itself be written using quantifiers: An integer is even if it equals the double of *some* other integer, namely

> There exists an integer m such that $10 = 2 \times m$.

This proposition is true because $10 = 2 \times 5$. The predicate "For all integers n, n is even" can be written as

For all integers n there exists an integer m such that $n = 2 \times m$

As we saw, this one is false.

Both the *names* of the quantified variables and the *order* in which they appear matters in such statements – do not be careless with them! For example, if we change the role of m and n, we obtain the proposition

For all integers m there exists an integer n such that $n = 2 \times m$

which is true. Now if we change the order of the quantifiers in the last statement, we obtain

There exists an integer n such that for all integers m, $n = 2 \times m$

which is false again.

The implies operator

The IMPLIES operator, which we also write as $\longrightarrow$, captures the meaning of the English conditional "If P then Q". It has the following truth table:

P	Q	$P \longrightarrow Q$
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

So the proposition

$$1 + 1 = 2 \longrightarrow 1 + 1 = 3$$

is false. On the other hand, the propositions

$$1 + 1 = 3 \longrightarrow 1 + 1 = 2$$
$$1 + 1 = 3 \longrightarrow 1 + 1 = 4$$

are both true. This may sound a bit strange at first, but it makes perfect sense: If $1 + 1 = 3$, which is clearly false, then anything goes. *A false proposition implies any other proposition.*

The *implies* operator comes in handy when reasoning about predicates. For example, let's take the following statement about integers:

(1) Every even number is the sum of two odd numbers.

Let's give the number a name – let's call it n. Statement (1) says that if n happens to be even, then n must be the sum of two odd numbers:

(2) For every n, (n is even) $\longrightarrow$ (n is the sum of two odd numbers)

Let's give the predicate "(n is even) $\longrightarrow$ (n is the sum of two odd numbers)" a name: We'll call it $P(n)$. Then $P(0)$ is true because 0 is even and it is the sum of two odd numbers (1 and -1). $P(1)$ is true because 1 is not even; $P(8)$ is true because 8 is even and it is the sum of 3 and 5. It looks like this proposition may be true.

Using the definitions of "even number" and "odd number", we can further expand statement (2) like this:

(3) For every n, (There exists an m such that $n = 2 \times m$) $\longrightarrow$ (There exist a and b such that $(n = a + b)$ AND (there exists a c such that $a = 2 \times c + 1$) AND (there exists a d such that $b = 2 \times d + 1$)).

The formulations (1), (2), and (3) all describe the same proposition. Which is the best one to use? It all depends on context. For example, if you are not sure whether the proposition is true and want to ask your teacher about it, you should be as succinct as possible and use formulation (1): "Is it true that every even number is the sum of two odd numbers?" If you want to reason about the truth of the statement, say by trying out different cases as we just did, then formulation (2) is more suitable. Formulation (3) is too detailed for most purposes and would be rarely used in practice. Such statements might come up in certain areas of computer science like automated theorem proving.

You will need to become comfortable at translating between different formulations of the same proposition inside your head.

From an English sentence to a proposition

Let's practice translating some English statements into propositions with quantifiers. The propositions will be about people in a group (Alice, Bob, Charlie, ...) and friendships among them. We'll write $F(x, y)$ for the predicate "x and y are friends". When writing propositions formally, it is customary to use the symbol $\exists$ for "there exists" and $\forall$ for "for all".

Example 1. Let's take the statement "Alice has friends". It means that there is *someone* out there who is Alice's friend, which calls for an existential quantifier:

$$\exists x \colon F(\text{Alice}, x).$$

Example 2. By default, Facebook allows friends of friends to view your profile. How do you write "Alice is a friend of a friend of Bob" formally? This statement says there is a friend of Bob out there who is also a friend of Alice:

$$\exists x \colon F(\text{Alice}, x) \text{ AND } F(\text{Bob}, x).$$

Example 3. How about "Alice and Bob have the same friends"? This statement says that *anyone* who is friends with Alice is friends with Bob, and vice versa. The quantifier here is universal, and the operator IFF comes in handy:

$$\forall x \colon F(\text{Alice}, x) \text{ IFF } F(\text{Bob}, x).$$

Example 4. "Everyone has a friend" is expressed as $\forall x \exists y \colon F(x, y)$, while "Someone in the group is everyone's friend" is expressed as $\exists y \forall x \colon F(x, y)$.

Example 5. "Alice has no friends" says there does not exist a person who is Alice's friend, or NOT $\exists x \colon F(\text{Alice}, x)$. How about Alice has "exactly one friend"? This statement is a bit tricky – it actually says two things: (1) Alice has at least one friend ($\exists x \colon F(\text{Alice}, x)$) and (2) Alice has no more than one friend. One way to interpret the second statement is to say "Any two friends of Alice must in fact be the same person", or "If x and y are both friends with Alice, then $x = y$". We can express statement (2) as:

$$\forall x, y \colon (F(\text{Alice}, x) \text{ AND } F(\text{Alice}, y)) \longrightarrow x = y$$

and so the statement "Alice has exactly one friend" becomes

$$(\exists x \colon F(\text{Alice}, x)) \text{ AND } (\forall x, y \colon (F(\text{Alice}, x) \text{ AND } F(\text{Alice}, y)) \longrightarrow x = y).$$

Mathematical books and manuscripts, for example our textbook, rarely use such formal notation because it is very difficult to read. In your own mathematical writing – including your homework solutions – I also encourage you to express your propositions in plain English as much as possible. However, it is important that the meaning of your proposition always be *clear* and *unambiguous*; if the need arose, you ought to be able to express your proposition using logical symbols.

3 Quantifier logic

Here are two rules that come in handy for quantified predicates. Suppose you have a predicate $P(x)$.

Negating quantifiers The proposition "NOT (For every $x, P(x)$)" is equivalent to "There exists an x such that NOT $P(x)$".

For example, the proposition "Not every integer is greater than 2" is equivalent to "There exists an integer no greater than 2."

The second rule concerns predicates $P(x, y)$ in two variables.

Universal instantiation If the proposition "There exists an x such that for all y, $P(x, y)$" is true, then the proposition "For all y there exists a x such that $P(x, y)$" is also true.

For example, take the (true) proposition "There is a day of the week when there are no classes" (about the CUHK class schedule). This is the same as saying "There is a d such that for every c, class c does not meet on day d of the week". By the universal instantiation rule, we can conclude

that "For every c, there is a d such that class c does not meet on day d of the week", i.e., "For every class there is a day of the week when the class does not meet." Indeed, that day is Sunday.

The *converse* of this rule is invalid: For example, the predicate "Every class meets on some day of the week" is true, but "There exists a day of the week on which every class meets" is false.

4   Reasoning about quantifiers

It is often tricky to figure out whether a proposition that involves quantifiers is true or false.

Let's start with the proposition

(1) Every even integer is the sum of two odd integers.

How do we go about figuring out if this is true? First, we identify that the leading quantifier refers to "every even integer." We want to know if something is true for every even integer, so we start by trying out some examples. Is the statement true for 2? Yes, because $2 = 1 + 1$. Is it true for 4? Yes, $4 = 3 + 1$. Is it true for 0? Yes, $0 = 1 + (-1)$. These "experiments" seem to indicate that the proposition is true. Should we leave it at that?

Unfortunately we can't. A predicate $P(n)$ may well turn out to be true for all the cases n we can think of checking, but the proposition "For all n, $P(n)$" could still be false. Here is a nice example:

(2) For every nonnegative integer n, the number $n^2 + n + 41$ is prime.

To determine if this proposition is true, let's try out some examples. $0^2 + 0 + 41 = 41$, which is prime. $1^2 + 1 + 41 = 43$, which is prime. $2^2 + 2 + 41 = 47$, which is prime. Let's try something bigger. $10^2 + 10 + 41 = 151$, which also happens to be prime. Can we conclude that the proposition is true? If we did, we would be wrong: The number $41^2 + 41 + 41$ is not prime; it is the product of 41 and 43.

Does this mean that the effort we spent in checking cases was wasted? No! If you look carefully, in our analysis of proposition (1) all the examples we checked show a pattern: We write our even number as another number plus one. We are on to something: Every even number n is the sum of $n - 1$ and 1. But when n is even, $n - 1$ must be odd. So of course n is the sum of two odd numbers!

In contrast, for proposition (2), there is no discernible pattern in the examples we worked out. This may mean two things: Maybe the pattern is there but we cannot see it, or maybe the proposition is false. Here, the proposition turned out to be false.

Unfortunately, there is no foolproof method for finding the truth of propositions with quantifiers. Mathematicians can spend their whole lives trying to figure out one or a handful of propositions. Here is one, called Goldbach's Conjecture, that still eludes them:

Every even integer n greater than 2 is the sum of two primes.

People (and computers) have checked many values of n (try it out!) but still do not know if it is true.

The best way to learn to reason about propositions is through practice. But first we have to agree on a standard by which we can agree that a proposition has been established as true: a mathematical proof.

References

This lecture is based on Chapter 1 of the text *Mathematics for Computer Science* by E. Lehman, T. Leighton, and A. Meyer. Material from slides by Prof. Lap Chi Lau were also used in the preparation.