

Alice and the Locked Box

Alex Holkner

2107062K

1 The Locked Box

IN WHICH ALICE RECEIVES A CURIOUS PARCEL, PROMPTING HER TO CONTEMPLATE THE NATURE OF COMPUTATION.

“You know, I received a most curious parcel in the mail today, Pascal,” said Alice to her good friend as they looked out over the countryside.

“Oh yes? Is that it? It looks rather ordinary to me. Who’s it from?”

“It doesn’t seem to have a return address on it. On the front it says, ‘To Alice,’ and there’s a padlock holding it shut.”

Pascal examined the locked parcel and said, “That is curious. Have you looked inside?”

“Don’t be silly, I don’t have the key!”

“What do you mean? Why would someone send a locked parcel in the mail to you without saying who it’s from, or sending a key along with it?”

Alice shrugged and began to fiddle with the lock, “Perhaps they just forgot to send the key. Do you think you could pick the lock for me?”

“Alice, I’m a mathematician, not a locksmith. What do you expect me to do? Why don’t you try Charles the barber, down the road?”

“Is he a locksmith?” Alice asked.

“No, he’s a barber. Not a very good one though, I’ve heard; spends all his time working with little gears and levers instead of shears and clippers.”

Alice rolled her eyes, bid good day to her friend and set off down the path towards the barbershop.

2 A Barbershop Duet

DESCRIBING AN ENCOUNTER BETWEEN ALICE, THE BARBER AND HIS PARROT, AND HER INTRODUCTION TO FINITE STATE AUTOMATA AND REGULAR EXPRESSIONS.

The barbershop was cluttered and Alice stepped on many little bits and bobs before discovering the barber at the back of the workshop. For contrary to the sign out the front, that was precisely what it looked like.

“Excuse me, are you the barber?” asked Alice.

“Good day to you, little lady. What pretty hair you have. Would you like a trim?”

“No, thank you, sir. My name is Alice and I have a locked box I was hoping you could help me with.” Alice offered the box to the barber.

“I’m afraid I have no experience with locks of boxes, only locks of hair. Would you like me to trim yours?” said the barber, handing the box back and spinning Alice around to inspect her hair.

“That’s very kind of you, but I’m really more interested in this box. My friend Pascal said you were good with machines and might be able to help me.”

“It’s very kind of your friend to recommend me. Would you like me to trim you hair?”

Alice could see she was getting nowhere with this barber and agreed to a quick trim. As she sat watching curls of her hair fall to the floor, she noticed a colourful bird in a cage in the back of the room.

“Is that your bird?” she asked the barber.

“It certainly is. It is a rare breed of parrot. For instead of repeating learned words and phrases, this parrot will squawk whenever it hears the word, *key*.” At that moment, the parrot let out a tremendous SQUAWK!!

“Really? That’s fascinating! May I try?”

“Of course,” replied the barber, wheeling Alice on her chair closer to the bird.

“Hmm, let me see,” said Alice. “You say it will recognise the word ‘key’ (SQUAWK!!) but not ‘quay’. What about ‘donkey’ (SQUAWK!!), ‘monkey’ (SQUAWK!!) or ‘okey dokey’ (SQUAWK!! SQUAWK!!)”.

Alice was by now giggling uncontrollably, and the barber had a hard time cutting her hair in a straight line. Nevertheless he managed to finish up and Alice complemented him on an excellent cut.

“Well, thank you very much for the haircut, Mr. Barber, but I really must be off now. I must find someone who can unlock my box.”

“I’m afraid I have no experience with locks of boxes, only locks of hair. Would you like me to trim yours?” offered the barber.

“But you trimmed it just now!”

“Oh, my. Heavens! I am sorry, I can be terribly forgetful at times. Please forgive me. Would you like me to trim your hair?”

3 Noah and the Otter

THE LATTER SHOWING ALICE THE IMPORTANCE OF CONTEXT-FREE GRAMMAR, AND THE FORMER MAKING USE OF PUSH DOWN AUTOMATA

“I was lucky to get out of that barber shop with any hair at all,” Alice thought. As she walked down the path she came across an otter sitting in the middle of the road. It was referring to a small book and muttering to itself. Alice decided to introduce herself.

“Hello there, Mr. Otter,” she said.

The otter looked up and exclaimed profoundly, “Cat the mat the on sat.” The otter seemed to have expended considerable effort in delivering this nonsense.

“That doesn’t make sense!” Alice exclaimed.

“The sat the cat on mat?” The otter tried.

“No, that doesn’t make sense either,” Alice said, “But I think I know what you are trying to say. Why can’t you speak normal English?”

“Stack my Noah away took.” Alice was beginning to get the idea of the otter’s mixed up speech.

“Noah? Who’s he?”

The otter opened its mouth to explain, then thought better of it, and pointed down the path, in the direction Alice supposed Noah must be.

★

Alice found Noah sitting at a table in a small clearing. He was shuffling a small deck of cards, and many more were scattered all around him.

“Excuse me, Mr. Noah,” Alice began.

“Oh, an audience! Fantastic! I have been waiting ever so long to show my invention to someone.” Noah motioned Alice to sit down. “Watch closely.”

Noah placed a single card on the table, face up. Alice tried to see what was written on the card.

“A,” Noah read out for Alice. He then placed another card on top of the A, and read out its value, “B.”

He looked at Alice expectantly. Alice shook her head, not understanding.

“Where has the A gone?” Noah questioned Alice like a teacher.

“It’s under the B.”

“How do you know? You cannot see under the B.”

“I can show you it’s there, by taking the B away,” explained Alice, taking the top card away and revealing the A.

“Very good,” said Noah, impressed. “But now where is the B?”

Alice looked around, “I seem to have tossed it away, but it must be around here somewhere.”

“Don’t worry about it, I have plenty more here. Now, you write down the letters as I read them out,” Noah said, picking up some more cards off the ground and clearing the table. Alice took out her notebook and pencil from her pocket.

“The first letter is A, then B.” Noah laid an A card and a B card on the table, as before. “Next we have B,” taking the B card off the stack, “and finally A.” There were no more cards on the table.

“Now what have you written down?”

“Abba,” Alice read. “Oh I see, that’s very clever. You’ve made up a new word.”

“No,” said Noah, frustrated, “It’s not made up, it’s the name of a singing group. Anyway, the point is, it’s the same word whether you read it forwards or backwards.”

“Why would I want to read something backwards?”

“I haven’t the faintest idea. But isn’t it fun?”

Alice was about to reply but thought better of it. “I prefer words that you can make by saying the same thing twice, like ‘yoyo’, ‘couscous’ and ‘dodo’. Can we make some of those words with the cards?”

Noah scratched his head and started dealing out cards, then picking them up, then putting them down, and so on. Alice watched as this continued for some time, but none of Alice’s words ever came up.

As she watched Noah deal the cards, she noticed many of them had words on them that had been scribbled out and replaced with Noah’s letters. This reminded her of the poor otter, so she collected some of the cards lying on the ground and left Noah to his palindromes.

★

The otter was still sitting by the path. She offered it the cards, which the otter took gratefully.

“Do the cards help you with your English?” Alice asked.

The otter nodded, and took out its pocket book. Alice watched with amazement as the otter began dealing out cards, referring to the book constantly.

“Thank you for returning my stack. That dreadful Noah took it so he could make up silly names for singing groups.”

“How is it that you can use the same cards to make sentences?”

The otter beamed and explained (amidst a great deal of card shuffling), “It’s very simple really. Each card has a single word on it. I just follow the rules in this book, which tell me when to put a word on my stack, and when to pick one up. When there are no more words on the ground the sentence is finished.”

Alice caught sight of the otter’s rule book and saw many complicated rules like:

OrnateNoun → ArticledNoun | AdjectivedNoun

ArticledNoun → Article AdjectivedNoun

AdjectivedNoun → Adjective AdjectivedNoun | Noun

“Is there anything I can do to repay you?” The otter asked.

“I don’t know if you can help me, but I’m trying to find someone who can open this locked box. You see, it’s addressed to me, but I don’t have a key for it.” Alice showed the otter her parcel.

“Say, there’s a turtle near here who just loves problems, I’m sure he’d be able to help you. Just keep following the path.”

4 A Game of Hopscotch

DESCRIBING THE WORLD’S FIRST TURING MACHINE MADE OF CHALK

“Why, it’s a giant game of hopscotch,” Alice exclaimed. For that is exactly what it looked like, except that instead of six squares, there appeared to be hundreds. In fact, from where she was standing, she could not even *see* the last square.

“I wonder if anyone is playing?” she thought, “I’d love to try it out.”

A movement caught her eye. Off in the distance there was a turtle scuttling backwards and forwards over the hopscotch field. It saw her and marked its place with a small piece of chalk, then hurried over to greet Alice.

“Good afternoon to you, madam. I am Alan, at your service.” The turtle was wearing spectacles and looked quite intelligent.

Alice did a polite curtsy, “Alice, at your service. Could I join in your game?”

“That’s very kind of you Alice, but there really is no need. I am progressing just fine by myself.”

“No need?” cried Alice, “But I *want* to play.”

“Well, do you know the rules?” enquired Alan, skeptically. “This is no ordinary game of hopscotch, you know.”

“Perhaps you could show me how to play, then. Please, I’d very much like to learn, it does look like fun.”

“OK, pay attention. You’ll notice that within each square is either a cross or a circle.”

“*That* square doesn’t have a cross or a circle,” Alice pointed out.

“Quite right. Let me clarify — a square can have a cross, a circle, or nothing. You start in the first square, like so.” The turtle stepped into the first square. “And now follow the programme to proceed.”

“Programme? What programme?”

“Yes, I forgot to mention that. Each time you play, you write out a programme to follow, like this one.” Alan showed Alice a scrap of paper with some scribbles on it. “The programme tells you which way to hop, depending if you’re on a square with a cross or a circle.”

“Or a blank square,” Alice interjected.

“Yes, or a blank square. Furthermore, you’ll notice I always play with a piece of chalk, so I can change what’s written in the square I’m in, according to the programme.”

“Well it all sounds very complicated. How do you win the game?”

“The programme will tell you when you’ve won. You see, the programme consists of many instructions. For example, an instruction might be:

Instruction No. 1:

- If you are on a cross, write a circle, hop to the right, and read instruction No. 2.
- If you are on a circle, write a cross, hop to the left, and read instruction No. 1 again.

“And thus you continue reading instructions until you get to a winning one.”

“I see,” Alice said, “And what happens if you are on a blank square in that instruction?”

“If the programme doesn’t tell you what to do, then you have lost the game.”

“Well that seems all too likely, especially with the programme you’re holding there,” Alice said. “It would be terribly disappointing to lose a game after going to so much effort.”

“It depends on what combination of crosses and circles — and blanks — are on the field to begin with. Sometimes I like to play lots of times with the same programme, to see which crosses and circles will win a game, and which will lose.”

Alice sat down to watch Alan return to the hopscotch game. With the programme in his hand, he continued scuttling back and forth over the squares. Sometimes he would change a circle to a cross, sometimes he would change a cross to a circle, and sometimes he would rub out a marking altogether.

5 The Turtle's Deception

AN EXAMINATION OF GÖDEL NUMBERING AND THE UNSOLVABILITY OF THE HALTING PROBLEM.

"How long is this going to take?" Alice was growing weary — and dizzy — watching the turtle scuttle back and forth.

"Hard to say," Alan panted, "Sometimes these things take a while."

"A while? How long is that? Can't you take a break?" Alice asked.

"Certainly I can," the turtle finished off his marking and came back to sit with Alice.

"Just what is the point of it all anyway?" Alice asked.

"What do you mean?"

"Well, all those crosses and circles — they don't really *say* anything do they? You can't make words or sentences out of them, like Noah and the otter with their cards."

"Pfft! Words and sentences, such things are trivial. Why, my crosses and circles can spell any such things, and more."

At that Alan picked up his chalk and drew something on the ground:

$$\begin{aligned} \text{XO} &= \text{A} \\ \text{XOO} &= \text{B} \\ \text{XOOO} &= \text{C} \end{aligned}$$

"A, B, C," Alice read out. "I suppose D would be XOOOO?"

"Yes, now you're getting the hang of it. The crosses and circles do not mean anything by themselves, but combined according to a scheme, they can represent anything, like words or letters."

"That's very clever, did you think of that yourself?"

"No, a friendly goat showed me that trick."

Alice got up to investigate the crosses and circles currently in play on the game squares. She started at the beginning of the hopscotch field and counted the crosses and circles until they stopped, several hundred metres down the track. As she went she tried to mentally convert each sequence into its corresponding letter.

"Couscous!" she shouted, and ran back to Alan.

"That's right."

"Well that's very impressive. Noah couldn't do that with his cards."

"Noah's a silly old man. But word games are just for fun, I can write far more interesting programmes."

"I suppose you could write out anything, couldn't you? You could make a programme that answers questions. Like, 'What should I have for breakfast?', or, 'Why is the sky blue?', or, 'What's the square root of 4?'"

"Yes, I have a programme around here somewhere that answers questions," Alan started rifling through his files. "Here it is," he said, pulling out what looked like a telephone directory.

"Is that it? Goodness, how on earth did you manage to write it?"

"Turtles live for a long time, Alice."

“I see. Can you get it to answer a question for me?”

“Certainly. What’s the question?”

Alice thought for a while, then wrote down — in the code of circles and crosses:

Does this question give a false answer, when answered?

Alan looked at the question and scrunched up his face. “Are you sure you want to ask *that* question?”

“Yes, why not? It’s a simple enough question.”

“No it’s not, it can’t be answered. If the answer is ‘yes’, then the answer is wrong, because the answer should have been false; but if the answer is ‘no’, the answer is still wrong, because it’s true.”

“I thought your programme could answer any question.”

“It can, it can! Just ask another question.”

Alice started getting suspicious. “Give me a look at that,” she said, grabbing the book out of Alan’s hands. “Why, the pages are all blank! It’s not a programme at all!”

Alan’s face flushed. “I was going to write it! I just need to work out the details.”

6 Coffee Dissertation

IN WHICH ALICE IS INTRODUCED TO COFFEE, THE HAMILTON CIRCUIT PROBLEM, AND THE NOTION OF COMPLEXITY.

“I didn’t mean to embarrass Alan,” Alice thought. “He brought it upon himself.” She had left Alan and was now continuing down the same path. “And I still haven’t found anyone who can open my parcel.”

As she came to a fork in the road, she noticed a handsome traveller puzzling over a map. She could tell he was a traveller by his luggage.

“Hello, my name’s Alice. Are you lost?” Alice introduced herself.

“Hi! No, I’m not lost, just perplexed.”

“Is that a map you’re holding?”

“Yes, as a matter of fact. It shows all the coffee shops in the neighbourhood. I’ll be reviewing them for the new issue of *Caffeine Fix*.”

“I’m not much of a coffee drinker myself,” Alice admitted.

“Well neither am I, but my editor gives me these jobs. Anyway, I can’t seem to find a route between these cafés. It’s easy enough to navigate between them, but I’d hate to have to backtrack — my editor would think I’m slacking off if I visit the same place twice.”

Alice looked at the map. There were about ten coffee shops on the map, and it didn’t take long for her to find an appropriate route for the traveller.

“Thanks. Say, you’re good at this. Can I buy you a coffee?”

“Will your editor mind?”

“No, I just won’t tell him.”

★

Alice and the traveller were enjoying their mochas in a small café. With the exception of three shadowy figures in the corner of the room, they were the only customers present.

“The key to a successful brew is all in the choice of bean,” the traveller was saying.

“I didn’t know there was a choice,” said Alice, for really she wasn’t much of a coffee drinker.

“Oh yes. There are about twenty-five species of the *Rubiaceae* plant, each yielding a different coffee bean. Though typically only three are harvested commercially; Arabica, Canephora and Liberica.”

“And they all taste different?”

“Oh yes, quite. There are also several varieties of each species, such as the *bourbon* variety of Arabica. And many growers will cross-pollinate different varieties to produce mutants, such as the Caturra.”

“Goodness! How can you tell the difference between all the different beans?” Alice asked.

“I make it my business to be familiar with all of them. However, the difficulty only begins there. A truly excellent coffee will often be a blend of several bean varieties.”

“How many blends are there?”

“Well, I couldn’t say. You can mix any two beans together to make up a blend.”

Alice thought for a while, remembering her mathematics lessons, “I suppose there are three blends of the three main beans; Arabica and Canephora, Arabica and Liberica, Canephora and Liberica. And if you include Caturra, there are an additional three; Arabica and Caturra, Canephora and Caturra, and Liberica and Caturra.”

“But there are far more blends than that. Even an occasional drinker could familiarise him or herself with those blends. What about the other varieties?”

“I suppose every coffee variety must be mated with every other variety, but only once for each variety.” Alice was recalling what she had learned about permutations and combinations. “So if there are twenty-five varieties, there will be about three hundred different blends.”

“Quite achievable for a man whose business it is to know coffee,” the traveller said proudly. “But you’re forgetting about the crossed varieties of plants; there are several more of those to consider as well.”

Suddenly there was a crash from the kitchen. The doors flung open and the cook advanced towards them.

“No, no no!” he boomed. “The real secret to perfecting a coffee is to blend different *blends* together. That’s what the experts do.”

“My,” said Alice, “I hadn’t even thought of that. How many blends of blends can be made, I wonder?”

“Millions of millions!” the cook shouted and returned to his kitchen in a huff.

“Surely not that many,” Alice said quietly.

7 The Locked Box

IN WHICH ALICE MEETS A MAN WITH A SECRET, AND THE CONTENT OF HER OWN BOX IS FINALLY REVEALED.

“Actually, there are many more,” said an unfamiliar voice.

Alice looked behind her and was stunned to see the three men who had been at the other end of the café were now standing directly behind her.

“Apologies, madam. Allow me to introduce myself. My name is Ron. My two companions and I were sitting at that table, and we could not help but overhear your fascinating conversation.”

“This is the last time I try to hold a private conversation in a café,” Alice thought. What she said was, “Pleased to meet you.”

To the traveller, Ron said, “We have heard of your legendary coffee expertise, and are big fans of your regular column in *Caffeine Fix*.”

“I am honoured. Are you aficionados of coffee blends?”

“You could say so — we are actually avid coffee blenders. I wonder if you would care to try our latest brew?” Ron produced a small jar from the folds of his cloak.

“I would be delighted,” the traveller said, taking the jar. Alice watched in horror as he deftly spun the top off the jar and scooped an entire spoonful of the coffee into his mouth.

The traveller considered the taste for quite some time before declaring, “Why, this is simply a tremendous achievement. What a subtle mix of flavours you have created. You simply must tell me your blend.”

“I’m afraid that’s quite impossible,” Ron said, “The blend is a very closely guarded secret which only myself and my associates can know.”

Alice interrupted, “But if you keep giving out free samples like that, aren’t you worried someone will discover your secret blend?”

Ron smiled, “It is highly unlikely. As the cook said, even considering only the twenty-five pure species of plant, there are more than a million, million, million, million different blends possible. Someone like your friend here would have to try all of them to discover our secret combination. And not even the most dedicated of coffee tasters could have time for that.”

“He’s right you know,” the traveller nodded, “He is the only one with the key to unlock the secret of this blend.”

“Key!” cried Alice, “I’d almost forgotten. I’ve got to get this box open.” She took out the parcel and showed it to Ron.

He glanced at it only briefly. “A lock is like a blend of coffee. Only the person who possesses the key may reveal its secrets.”

“But I don’t have the key,” Alice complained, “How am I supposed to open it?”

“You must trust me, only the person who locked that box can open it.” He looked at Alice with such intensity that she knew he was implying something significant by this. “Think, why is the box locked?”

“Because they didn’t want anyone else to see what’s inside, apart from me,” Alice reasoned. “But only they can open it. I’ll have to somehow persuade them to open it for me. How can I do that?”

Ron continued to look at Alice intensely. “You will have to persuade them that no-one else will be able to see inside, even after they have unlocked it.”

Suddenly Alice understood. From her pocket she produced a padlock of her own and snapped it onto the box alongside the original lock. Now there were two locks on the parcel.

“I don’t understand,” the traveller looked aghast, “That doesn’t help you open it at all — you’ve only made it more difficult for yourself by adding another lock.”

“Don’t worry,” Alice said, “I think this will work.” She quickly scribbled ‘Return to Sender’ on the parcel and ran outside to drop it into the mailbox.

★

Alice had just finished telling Ron, his two friends and the traveller about her adventures that day when a postman walked in.

“Package for Miss Alice,” the postman cried.

Alice leapt up and eagerly took the parcel.

“It’s here! And look — it’s still locked up tight, except now the other lock has been taken off and only my lock is holding it shut.” She fished out the key and snapped off the lock.

Alice opened the box and took out the note inside.

Dear Alice,
Glad to see this note got to you safely,
Love,
Bob.

Addendum

The story of *Alice and the Locked Box* is an exploration of computational theory. Alice’s encounters with the various inhabitants of her fantasy world demonstrate some of the fundamental concepts used in this science. In this addendum, these concepts will be made explicit, and some will be dealt with in more detail than was possible in the story.

The simplest type of computer is a Finite State Machine. Such devices consist of a series of *states* and rules to guide the machine between states based on the input. These machines have no memory beyond the implicit memory of their state. The barber likewise has no memory, he can’t even carry out a simple conversation. These devices can be programmed for a particular task — say, cutting Alice’s hair — but cannot reuse information.

A Finite State Machine, or FSM, which has exactly one rule to follow for each given input and state is said to be *deterministic*. It is a straightforward task with these machines to trace an input through to its logical final state.

Another class of FSMs is the *nondeterministic* type, which has less restrictions. Such machines do not have to have a rule for each possible input, or may have many possible options for a given input and state. They may also contain λ (lambda) transitions, in which the state changes but no input is consumed. These nondeterministic machines can potentially have a number of possible outputs for a given input. Typically we would let some of these states be “yes” states and others “no” states; if for a given input a “yes” state can be reached, then the entire process is considered successful.

It can be shown that all nondeterministic finite state machines can be reduced to a deterministic finite state machine. The significance of a machine being deterministic or nondeterministic will be examined later.

A typical use of finite state machines is as language recognisers. These machines will take as input a string (a sequence of characters or tokens) and classify it as either belonging to a particular language or not. The class of languages that an FSM can recognise is very strict and is known as *regular*. A regular language can be specified with a *regular expression*. For example:

$$a^*b^*cc \quad (1)$$

This language accepts all strings with zero or more a's, followed by zero or more b's, followed by exactly two c's. For example, the strings "aabbcc", "bcc", "aaaacc" are all valid in this language. Regular expressions are commonly used for searching text on a computer. For instance, the expression:

$$.*key.* \quad (2)$$

will match any string containing any number of any characters (the full stop can stand for any character) followed by the sequence "key" followed by any number of characters. In other words, any string containing the substring "key" will match. Alice observes this capability when she is introduced to the parrot in the barber shop.

Alice's meeting with Noah begins with a demonstration of some card-play. Noah is demonstrating a vital memory technique used in computer science known as a *stack*. As he demonstrates, only the top most item on the stack can be accessed at any time.

If a stack is added to an FSM, we have a push-down automaton or PDA. As well as governing the transitions between states, the rules in a PDA can also *push* and *pop* data on and off a single stack. This affords the machine a simple memory, which can be used as a simple counter or as a means of recursion.

The set of languages that a PDA can recognise is larger than the regular languages and is known as the set of *context-free* languages. Such languages are usually specified using *set notation*. Note that all regular languages are also context-free. The regular expression shown at (1) can be written in set notation as follows:

$$\{a^ib^jcc \mid i, j \geq 0\} \quad (3)$$

While a regular language cannot place constraints on the number of characters with respect to each other (for instance, there is no regular expression that specifies the language with the same number of a's as b's), a context free language can, as in the following example:

$$\{a^ib^{2i} \mid i \geq 0\} \quad (4)$$

This will accept strings with some number of a's followed by exactly twice as many b's, such as "aabbbb" or "abb".

Any language can also be specified with a *grammar*. Grammars consist of a number of *variables* (denoted in uppercase A, B, \dots), some *terminals* (which are characters from the actual language), a start symbol S which is itself a

variable, and a set of *rules* for transforming variables into strings. Strings in the grammar consist of variables and terminals. The grammar of the language shown at (4) is:

$$\begin{aligned} S &\rightarrow A \\ A &\rightarrow aAbb \mid \lambda \end{aligned} \tag{5}$$

For instance, to derive the string “aabbbb”, start with the start symbol “S”, then substitute according to the first rule to get “A”. The A then follows the first rule to expand as “aAbb”. The same rule is applied again to the new A , yielding “aaAbbbb”. To terminate the recursion the lambda (or empty) rule is applied, giving us the desired string “aabbbb”.

The terminals need not stand for letters. The otter demonstrates a use of push-down automata to construct a simplified grammar for English. The rules in his pocket book follow the context-free grammar described above.

As with FSMs, PDAs can be classified as either deterministic or nondeterministic. Unlike FSMs, however, nondeterministic PDAs are not reduceably deterministic, and in fact are more powerful. The complement of a deterministic context-free language is not necessarily deterministic, though it will certainly be nondeterministic.

Even nondeterministic context-free grammars cannot specify every language. In particular, they cannot specify a language which requires more than one stack to parse. For example, the language:

$$\{a^i b^i c^i \mid i \geq 0\} \tag{6}$$

cannot be specified without using context. Alice exposes this weakness when she asks Noah to deal the cards in such a way as to spell “couscous”, which does not follow the centre-recursive (or palindromic) rules he was using.

The analogy of Noah’s stack and the otter’s language to push-down automata is fairly imprecise in parts, and the barber’s reduction to a finite state automaton is even more opaque. Alice’s introduction to Alan and his hopscotch field is, on the other hand, a precise description of the workings of a *Turing machine*.

The Turing machine expands on the memory of a push-down automata by replacing the stack with an infinite *tape* (hopscotch field) which can be read and written at random. Turing machines are the abstract mathematical description of current computers and, with the exception of the still theoretical Quantum computer, represent the most powerful computing device known to date. Real computers differ only in that they have a finite amount of memory, rather than the infinite hopscotch field of a Turing machine.

With their random access memory, Turing machines can be built to accept any language, as demonstrated when Alice discovers that Alan’s current project has produced “couscous”, a sequence not possible using PDA.

Various additions can be made to Turing machines in an attempt to make them still more powerful. For example, a machine with two or more tapes could be conceived. Alice, when she asks Alan if she can play, is effectively offering to add another tape-head to the system. Alan turns her down, saying she is not needed – as any such improvements are reducible to the basic Turing machine as described.

Although Alan's hopscotch field is made up of crosses and circles, he shows how these can be made to represent letters in the alphabet with a simple coding scheme. The scheme he shows Alice is reminiscent of Gödel numbering in which any mathematical statement can be represented as a number (in this case binary) with some simple rules. As such, the choice of symbols that a Turing machine operates on is entirely arbitrary.

It follows that since anything can be encoded onto a tape for a Turing machine, the instructions (rules) for the machine itself can be encoded as input. This kind of self-reference can lead to problems, as Alice exposes when she poses the question, "Does this question give a false answer, when answered?" In essence she is exposing Gödel's *Incompleteness Theorem*, which is closely related to the results of the *Halting Problem*.

The Halting Problem is, "Does a Turing machine M halt with input w ?" By "halt", we mean, "Will the machine ever terminate its program, or will it run forever?" The proof that no Turing machine can be created to answer this question for all Turing machines is shown with self-reference and contradiction. The significance of this result is that certain problems — by reducing them to the Halting Problem — can be shown to be uncomputable.

Note that Turing machines can conceivably be devised which will answer any of Alice's other questions and machines can even be made to answer the Halting Problem for certain machines. The Halting argument shows that no one machine can answer the question for all machines without itself halting.

The idea of complexity — how long something will take to compute — was touched on briefly by Alan, but is explored in more depth by Alice and the traveller.

The problem of the traveller's route between coffee shops is known as the *Hamilton Circuit Problem* and is famous in the field of computer science. As Alice demonstrates, solutions to the problem for small datasets are easily achievable. The problem of finding a particular blend of coffee is reducible to the Hamilton circuit problem. When a slightly larger dataset is introduced (twenty-five bean varieties) the problem quickly becomes insurmountable and is said to be *intractable*.

The complexity of an algorithm is measured by the function that models its growth with larger datasets. A naïve solution to the Hamilton circuit problem would operate in exponential time, meaning each addition to the data set increases processing time by an order of magnitude.

There are many problems which can be reduced to the Hamilton circuit problem, and we classify them as being *NP-complete*. Problems are known to be in P-space if there is a Turing machine with polynomial complexity which can solve it. Problems in NP-space have a nondeterministic polynomial-time solution, but typically have exponential or worse time on deterministic machines. A subset of NP-space are the NP-complete problems. These are problems which can be reduced in polynomial time to each other — in a way, any solution to one of these problems is equally applicable to all the others.

A special property of NP problems is that while they are thought to be intractable, no mathematical proof has yet been found which shows that there is *not* a way to solve them in a reasonable amount of time. This is currently an open problem in the field of computer science.

As Ron demonstrates to Alice, we use intractable problems as a way to keep

secrets; that is, to encrypt data. *Public key cryptography* is a way of encrypting data for a person without having to establish a prior secret channel for the exchange of keys. The most common algorithm for public key cryptography currently is RSA (named after the initials of its creators). To break something encrypted with RSA requires the eavesdropper to factorise a very large number into its prime constituents. This problem is in NP-space, and as such is believed — but has not been proven — to be intractable.

The mathematical reasoning behind RSA is complex, but a suitable analogy is that of a box with two locks, each able to be opened by only one of the parties. Alice's final exchange with Bob is a traditional demonstration of this technique.

Alice's quest to open a locked box led her to investigate various types of machine and the language classes they accept, the notions of computability and complexity, and the application of complexity to cryptography. In essence, this is what is known about computation to date.