Revision

The exam

- Two-hour written exam.
- Full marks will be given to correct answers to THREE questions. Only the best three questions will contribute toward the assessment.

Enumerability & diagonalization

Enumerability: characterizations

You can use the following fact from the lecture: let A be a set. The following are equivalent:

1. A is the range of a function $f : N \longrightarrow A$ from the natural numbers to A (informally, A can be written as a list with holes).
2. A has an **encoding**, i.e., there is a total injective function $c : A \longrightarrow N$ into the natural numbers. (For $a \in A$, the number $c(a)$ is called the **code** of a.)

Pairs of integers

$$N \times N \underset{enumeration}{\overset{encoding}{\rightleftarrows}} N$$

For example:

- Cantor's Zig-Zag;
- The encoding $c(x, y) = 2^x \cdot 3^y$.

Useful facts

To show that a set is enumerable, you can use the following useful facts (this used to be an exercise):

1. If A is enumerable and there is a surjective function $A \to B$, then B is enumerable.
2. If B is enumerable and there is a total injective function $A \to B$, then A is enumerable.

Next follow a couple of exercises, with solutions, that show the usefulness of these two facts.

Exercise

- Show that the set Q^+ of positive rational numbers is enumerable.

Solution: every positive rational number has the form x/y, where x and y are natural numbers and $y \neq 0$. So the function $f : N \times N \rightarrow Q^+$ given by

$$f(x, y) = \begin{cases} x/y & \text{if } y \neq 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

is surjective. So, to see that Q^+ is enumerable, it suffices to show that $N \times N$ is enumerable, which we know to be true.

Exercise

- Let A and B be enumerable sets such that $A \cap B = \emptyset$. Show that $A \cup B$ is enumerable.

Solution: If A and B are enumerable, we have encodings (= total injective functions) $f : A \longrightarrow N$ and $g : B \longrightarrow N$. Consider the following function $h : A \cup B \rightarrow N \times N$:

$$h(x) = \begin{cases} (1, x) & \text{if } x \in A \\ (2, x) & \text{if } x \in B \end{cases}$$

Obviously, h is injective. So, because $N \times N$ is enumerable, $A \cup B$ too is enumerable.

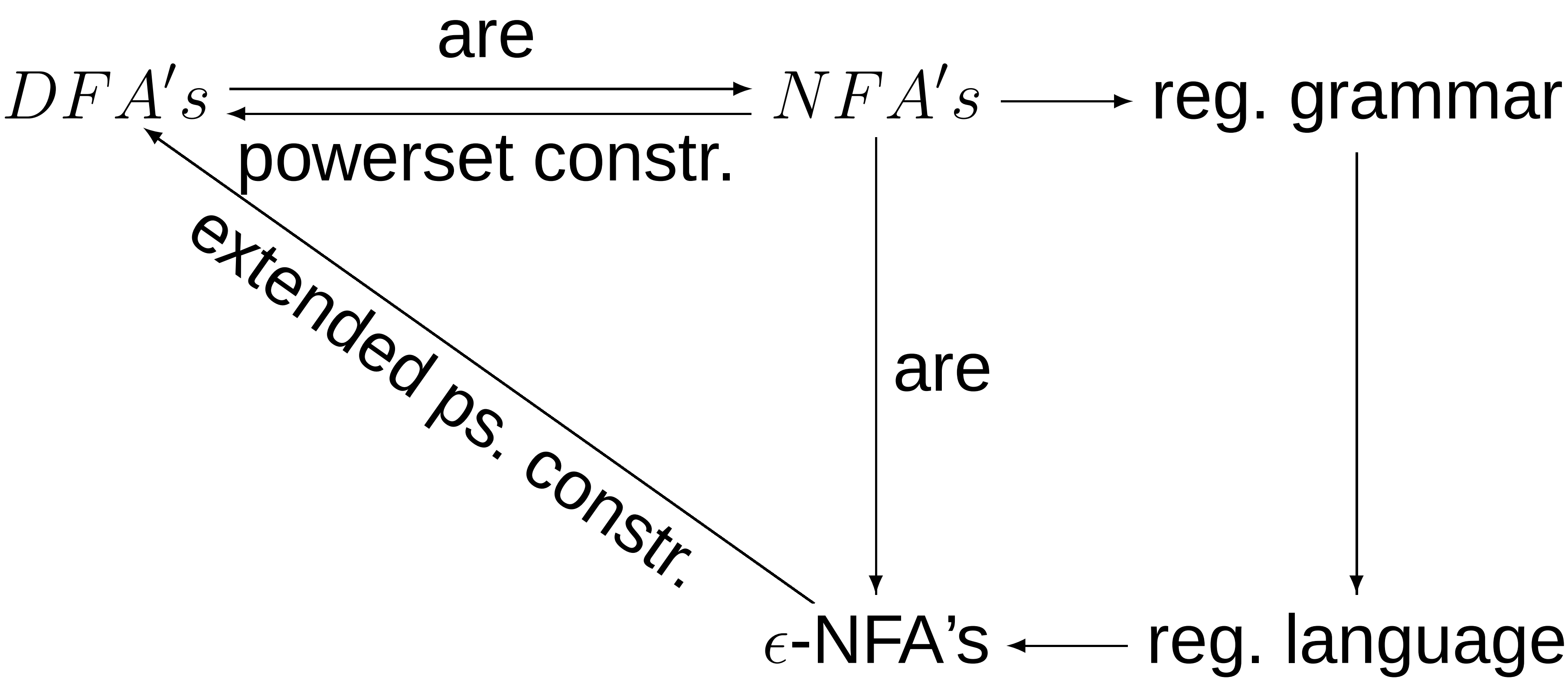
The diagonal argument

The diagonal argument, in its most intuitive form, shows that for every enumeration $f_1, f_2, f_3, \ldots$ of functions, we can construct a new function g which is not in that enumeration, by letting $g(n)$ be any value different from $f_n(n)$, e.g.,

n	1	2	3	4	5	$\ldots$
$f_1(n)$	1^{2}	9	0	8	$\perp$	$\ldots$
$f_2(n)$	0	$\perp^{0}$	1	0	3	
$f_3(n)$	1	4	$9^{\perp}$	2	$\perp$	
$f_4(n)$	4	7	1	7^{8}	8	
$f_5(n)$	2	3	5	7	2^{3}	
$\vdots$	$\vdots$					

Automata & languages

Automata & languages: summary



Exercise

Use the powerset construction to transform the following NFA into a DFA (you can present the DFA as a transition table or as a transition graph).

	0	1
$\rightarrow A$	$\{A, B\}$	$\{A, C\}$
$* B$	$\{B, C\}$	$\{\}$
$* C$	$\{\}$	$\{B, C\}$

Solution

	0	1
$\{\}$	$\{\}$	$\{\}$
$\rightarrow \{A\}$	$\{A, B\}$	$\{A, C\}$
$* \{B\}$	$\{B, C\}$	$\{\}$
$* \{C\}$	$\{\}$	$\{B, C\}$
$* \{A, B\}$	$\{A, B, C\}$	$\{A, C\}$
$* \{A, C\}$	$\{A, B\}$	$\{A, B, C\}$
$* \{B, C\}$	$\{B, C\}$	$\{B, C\}$
$* \{A, B, C\}$	$\{A, B, C\}$	$\{A, B, C\}$

Exercise

Give the regular expression for the NFA below.

	0	1
$\rightarrow X$	$\{X\}$	$\{Y\}$
$*Y$	$\{Y\}$	$\{Z\}$
Z	$\{Z\}$	$\{X\}$

Solution (part 1/2)

The regular grammar corresponding to the NFA is

$$X \to 0X|1Y$$
$$Y \to 0Y|1Z|\epsilon$$
$$Z \to 0Z|1X$$

The corresponding equation system is

$$(1) X = 0X + 1Y$$
$$(2) Y = 0Y + 1Z + \epsilon$$
$$(3) Z = 0Z + 1X$$

where X is the start symbol.

Solution (part 2/2)

$$(1) X = 0X + 1Y \qquad (2) Y = 0Y + 1Z + \epsilon \qquad (3) Z = 0Z + 1X$$

Because X is the start symbol, we are interested in the solution for X. We get

$(4) Z = 0^*1X$	from (3)
$(5) Y = 0Y + 10^*1X + \epsilon$	from $(2,4)$
$(6) Y = 0^*(10^*1X + \epsilon) = 0^*10^*1X + 0^*$	from (5)
$(7) X = 0X + 1(0^*10^*1X + 0^*) = 0X + 10^*10^*1X + 10^*$	from $(1,6)$
$= (0 + 10^*10^*1)X + 10^*$	from $(1,6)$
$(8) X = (0 + 10^*10^*1)^*10^*$	from (7)

Exercise

Consider the grammar

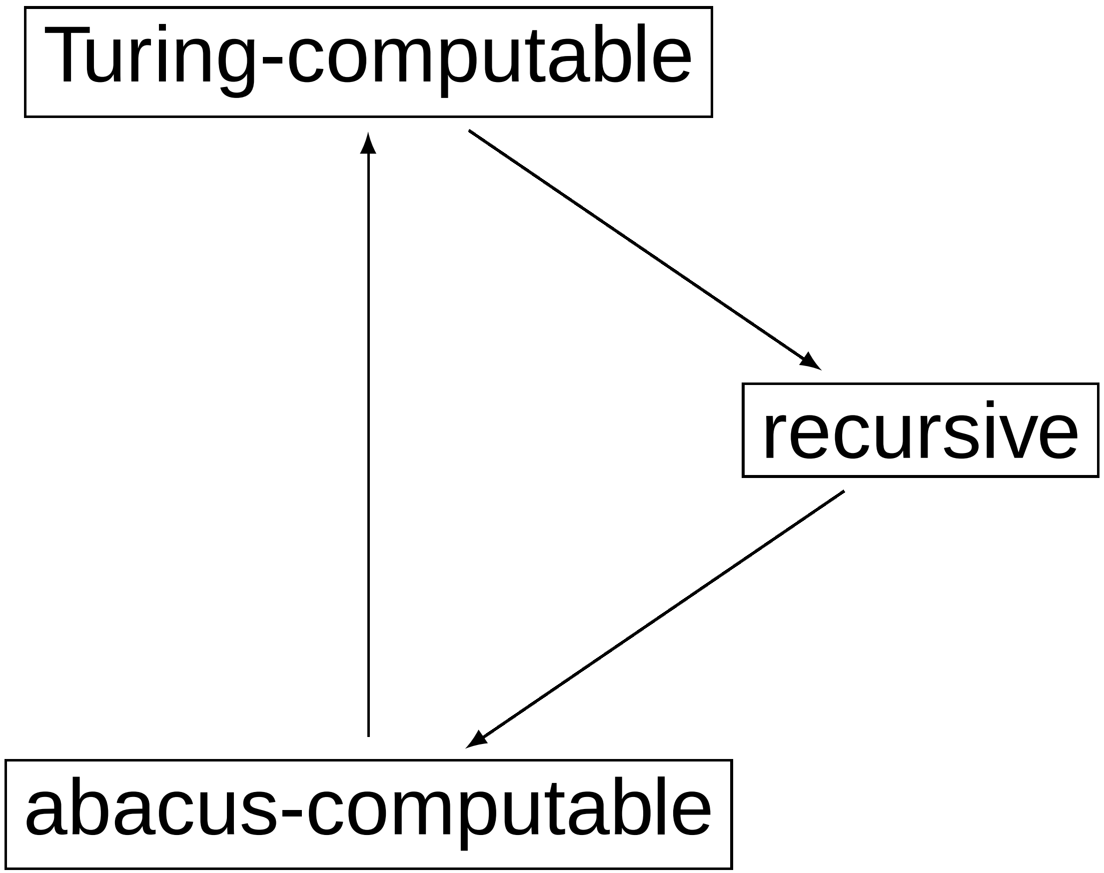
$$S \to aS \mid aSbS \mid \epsilon.$$

Show that this grammar is ambiguous.

Solution: E.g., there are two parse trees for the word aab.

Computability

Overview



Exercise

The **predecessor function** $pred$ takes one argument y and returns $y - 1$ if y is greater than 0, and returns 0 otherwise. Show that $pred$ is primitive recursive.

Solution (part 1/3)

Solution: Naively, we want to define $pred$ by primitive recursion, so we need a 0**-place** function f and a 2-place function g such that

$$pred(0) = f()$$
$$pred(s(y)) = g(y, pred(y))$$

At a first glance, this seems to be solved by $f() = 0$ and $g = \pi_1^2$. But we don't have any 0-place functions!

Solution (part 2/3)

We address this issue by defining, by primitive recursion, an auxiliary function

$$aux(x, 0) = f(x)$$
$$aux(x, s(y)) = g(x, y, aux(x, y))$$

with a **dummy variable** x, and let $f(x) = z(x)$ and $g = \pi^3_2$. That is, $aux = \mathrm{Pr}[z, \pi^3_2]$. Then we let

$$pred(y) = aux(y, y),$$

i.e., $pred = \mathrm{Cn}[aux, \pi^1_1, \pi^1_1]$.

Solution (part 3/3)

However, saying that $pred$ is primitive recursive because it can be defined by primitive recursion as follows:

$$pred(0) = 0$$

$$pred(s(y)) = \pi^2_1(y, pred(y))$$

is **morally** the right answer, so I would accept it.

Exercise

Show that the factorial function is primitive recursive. (You can assume that multiplication is primitive recursive.)

Solution:

$$fac(0) = 1 \qquad fac(s(y)) = (s(y)) * fac(y)$$

That is,

$$fac(0) = 1 \qquad fac(s(y)) = g(y, fac(y))$$

where $g = \mathrm{Cn}[*, \mathrm{Cn}[s, \pi_1^2], \pi_2^2]$. Like for $pred$, we have the issue with the missing 0-place function (we don't have a function $f() = 1$), but it is acceptable to gloss over that.

Exercise

Suppose that the function $f(x, y)$ looks like this:

	0	1	2	3	4	$\ldots y$
0	1	2	3	4	5	$\ldots$
1	7	0	7	0	7	$\ldots$
2	$\perp$	0	$\perp$	0	$\perp$	$\ldots$
3	0	$\perp$	0	$\perp$	0	$\ldots$
$\vdots$						
x						

What are $\mathrm{Mn}[f](0)$, $\mathrm{Mn}[f](1)$, $\mathrm{Mn}[f](2)$, $\mathrm{Mn}[f](3)$?

Solution

$\mathrm{Mn}[f](0) = \bot$, $\mathrm{Mn}[f](1) = 1$, $\mathrm{Mn}[f](2) = \bot$, $\mathrm{Mn}[f](3) = 0$.

More exercises

- To get the exams of the last two years (Prof. Pym): enter

  `http://www.bath.ac.uk/library/exampapers/search.html`

  and search for “comp0020”.
- 2002 exam: Exercise 1(f), 2(a-f) (“countable” = “enumerable”), 3(a), 4(a-c), 5(a-d) (except 5c).
- 2003 exam: 2(a-b), 3(a-d) (“partial recursive” = “recursive”), 4(a-d).

Addendum: complete proof of the last theorem of the last lecture

Theorem

Theorem. Let R be 1-place relation on the natural numbers. The following are equivalent:

1. R is semi-recursive;
2. R is the empty set, or recursively enumerable by a **total** recursive function;
3. R is recursively enumerable.

Proof (part 1/2)

That (2) implies (3) is trivial. To see that (1) implies (2), suppose that R is semi-recursive. If R is empty, we are done, so suppose R non-empty. Let $z \in R$, and suppose that R is the domain of some recursive function f computed by the TM with code m. Define another function

$$g(x,t) = \begin{cases} x & \text{iff } stdh(m,x,t) = 0 \\ z & \text{otherwise} \end{cases}$$

We have $R = domain(f) = range(g)$. Letting

$$h(y) = g(first(y), second(y)),$$

R is the range of h.

Proof (part 2/2)

To see that (3) implies (1), assume that R is the range of the k-place recursive function g. Then

$$R(y) \quad \text{iff} \quad \exists x_1. \cdots .\exists x_k.g(x_1, \ldots, x_k) = y.$$

The $(k+1)$-place relation $g(x_1, \ldots, x_k) = y$ is easily seen to be semi-recursive. Because

semi-recursive relations are closed under $\exists$ (earlier proposition), R is semi-recursive.