Midterm Grand Tutorial
General CS I (320101)

October 25, 2014

1 Mathematical Foundations

Problem 1.1 (Nicely Greater Sets)
Consider two arbitrary finite sets A and B of real numbers. We say that A is *nicely greater* 0pt than B, denoted $A \gg B$, iff the following conditions are fulfilled:

1. $\#(A) > \#(B)$
2. $\forall a \in A \backslash B. \forall b \in B. a > b$

Prove or refute that the relation $\gg$ is transitive.

Example: For $A = \{1,2,3\}$, $B = \{1,2\}$ and $C = \{2,4\}$, we only have $A \gg B$.

Note: $\#(A)$ is the size of the set A.

Solution: Consider the arbitrary sets A, B, C such that $A \gg B$ and $B \gg C$ and let's prove that $A \gg C$.

First of all, it's easy to prove that $\#(A) > \#(C)$ by the transitivity of the " $>$ " relation, i.e.

$$(\#(A) > \#(B)) \wedge (\#(B) > \#(C)) \Rightarrow \#(A) > \#(C)$$

For the second property, we need to get a bit more involved. Consider an arbitrary element x of $A \backslash C$. We want to prove that x is greater than any element of C.

We distinguish 2 cases:

1. $x \in B$: We have $x \in (A \backslash C)$, so $x \notin C$. As we assumed $x \in B$, we have that $x \in (B \backslash C)$.

   By $B \gg C$, we have $\forall b \in B \backslash C. \forall c \in C. b > c$, hence $\forall c \in C. x > c$

2. $x \notin B$: We have $x \in (A \backslash C)$, so $x \in A$. As we assumed $x \notin B$, we have that $x \in (A \backslash B)$.

   By $A \gg B$, we get that $\forall b \in B. x > b$.

   As $\#(B) > \#(C)$, there is a $b_1 \in B$ such that $b_1 \in (B \backslash C)$. As $B \gg C$, we have $\forall c \in C. b_1 > c$

   But now we are done, as $b_1 \in B$, so $x > b_1$, thus $x > b_1 > c$ for all $c \in C$.

Finally, we have deduced that $x > c$ for all $c \in C$, hence the second property also holds and $A \gg C$, as desired.

Problem 1.2 (UNN practice)

Let's redefine the successor function for UNN as follows: 0pt

$$s(0) = 1$$

$$s(n) = n^2 - 3n + 2, n > 0$$

Using this definition, write down the set of unary natural numbers.

Do the first four Peano axioms still hold? If not, which one fails and why?

Solution: The set is: $\{0, 1\}$.

P1: 0 is a unary natural number - YES

P2: YES ($s(0) = 1,\ s(1) = 0$)

P3: NO ($s(1) = 0$)

P4: YES ($s(0) \neq 1 s(1)$)

2 Sets, Relations and Functions

Problem 2.1 (Injective functions)

Let f be a function $f\colon U \to V$, where U and V are not-empty sets. Prove or refute that f injective iff there is a function $g\colon V \to U$ with $g \circ f = \mathrm{Id}_U$. 0pt 5min

Solution:

- $\Rightarrow$: Let $f\colon U \to V$ be an injective function. Let's make a function $g\colon V \to U$ in the following way. If $v \in codom(f)$ and $v = f(u)$ for some $u \in U$, then u is defined in a single way due to injectivity. Then let $g(v) = u$. For all $v \in (V \backslash codom(f))$ let $g(v) = u_0 \in U$, where u_0 is some fixed element in U. Then $g \circ f(u) = u = \mathrm{Id}_U$ for all $u \in U$.
- $\Leftarrow$: If there is a $g\colon V \to U$ with $g \circ f = \mathrm{Id}_U$ and $f(u_1) = f(u_2)$ for all $u_1, u_2 \in U$, then $u_1 = \mathrm{Id}_U(u_1) = g \circ f(u_1) = g(f(u_1)) = g(f(u_2)) = g \circ f(u_2) = \mathrm{Id}_U(u_2) = u_2$.

Problem 2.2: Prove or refute the following statements. If you want to refute it, a counterexample is enough. 0pt

1. Define a character code from $\mathbb{D} = \{0, 1 \ldots 9\}$ to $\mathbb{B} = \{0, 1\}$
   $c : \mathbb{D} \to \mathbb{B}^+; (n)_{10} \mapsto (n)_2$. ie. $c(1) = 1, \ldots c(7) = 111, c(8) = 1000, c(9) = 1001$
   The string code generated by this chracter code c is injective.
2. $f : \mathbb{R} \to \mathbb{R}; n \mapsto 2^n$ is surjective.

Problem 2.3 (Functions practice)

0pt

1. Prove or refute

   The function $f : \mathbb{R} \to \mathbb{R}$, $f(x) = \dfrac{e^x}{\sqrt{x^3 + 4}}$ is a `total function`.

2. Prove or refute bis

   The function $g : \mathbb{R} \to \mathbb{R}$, $g(z) = \sqrt{z}$ is a `partial function`.

Solution:

1. f is not total. Counter-example:

$$f(-2) = \frac{e^{-2}}{\sqrt{-4}} = \pm i \cdot \frac{1}{2e^2} \notin \mathbb{R}$$

2. g is not partial. Counter-example:

   $g(4) = \sqrt{4}$ could either be 2 or -2. However, for a partial function g there is supposed to be at most one value for $g(4)$.

---

3 Abstract Data Types

Problem 3.1 (Base 10 Natural Numbers)

0pt

A natural number in base 10 is any combination of digits between 0 and 9. Construct an ADT that represents natural numbers in base 10. Please note that leading zeros are **not** allowed. Represent the number 472 with your ADT.

Hint: Treat 0 as a separate sort.

Solution: $\langle\{\mathbb{Z}, \mathbb{N}, \mathbb{NZ}\}, \{[0 : \mathbb{Z}], [1 : \mathbb{NZ}], ...[9 : \mathbb{NZ}], [zero : \mathbb{Z} \to \mathbb{N}], [dig : \mathbb{NZ} \to \mathbb{N}], [add : \mathbb{NZ} \times \mathbb{N} \to N]\}\rangle$

472 is represented as $add(4, add(7, dig(2)))$.

---

4 Programming in Standard ML

Problem 4.1 (Crates at sea)

A certain ship is at sea on a voyage from Bremen to New York. There is a crate on her deck. However the crate has not been attached/fixed properly and when in a slight storm the ship goes over a wave, the crate can move one meter to the right or to the left. When it reaches the edge of the deck it is stopped by the rails. Assume that the crate is originally in the middle of the deck and that it is 1 meter wide. Write an SML function that given the half the decks width (so the distance from the original position of the crate to the edge of the deck) and a string describing the crates movements, determines how many times the crate will reach the edges of the deck. The string consists of the words "right", "left" for the respective movements. (You can ignore any other characters in between the words). 0pt

The function will have the following signature:

```
val count = fn : string * int -> int
```

Example:

```
- count("right left left left left_left_a_ right right right right -a-
right_ right right right right left right move left",4);
val it = 3 : int
```

Solution:

```
fun movements([],width, possition, times) = ([],width, possition, times)
  | movements((#"l")::(#"e")::(#"f")::(#"t")::l, width, possition, times) =
    if (possition - 1) = 0-width then movements(l,width, possition-1,times+1)
    else if possition = 0-width then movements(l,width, possition,times)
    else movements(l,width, possition-1,times)
  | movements((#"r")::(#"i")::(#"g")::(#"h")::(#"t")::l, width, possition,times) =
if (possition + 1) = width then movements(l,width, possition+1,times+1)
else
if possition = width then movements(l,width, possition,times)
else movements(l,width, possition+1,times)
  | movements(a::l, width, possition, times) = movements(l, width, possition,times);
 fun count(l,width) = let val (m,w,p,t) = movements(explode(l),width,0,0) in t
end;
```

Problem 4.2 (Angry Birds)

Your new favorite game is Angry Birds, and, after a lazy afternoon when you have played the game, you observed the following rules for deducing the score: 0pt

- the red bird will always add 5000 points to your score (no matter what it hits)
- the blue bird always is split into 3 smaller birds, and every bird adds 1000 points if it hits an object
- the yellow bird will add 2500 birds only if it hits a green pig
- the white bird is considered peaceful and will add no points to your score

Your version of Angry Birds permits you to choose K from a list of birds which you will fire. Therefore, you design the following SML datatype:

```
datatype angrybird = white | red | blue of int | yellow of bool;
```

The int parameter of the blue bird tells you how many birds hit an object, and the bool parameter of the yellow bird tells you if the bird hits a pig or not.

You are required to write an SML function getScore which takes a list of Angry Birds and the number K of birds you can fire and returns the maximum score you can get by firing your choice of K birds:

```
val getScore = fn : angrybird list * int -> int;
- getScore( [white, white, red, blue(3)], 3 );
val it = 8000 : int; (* red, blue and one of the whites are chosen *)
- getScore( [red, yellow(false), yellow(true), blue(2)], 2);
val it = 7500 : int; (* red and yellow(true) are chosen *)
```

Hint: For sorting a list of ints in descending order, you can use the following SML snippet:

```
- ListMergeSort.sort (op<) [1,5,1,2];
val it = [5,2,1,1] : int list;
```

Solution:

```
datatype angrybird = white | red | blue of int | yellow of bool;

fun convert(white) = 0
  | convert(red) = 5000
  | convert(blue(x)) = x*1000
  | convert(yellow(x)) = if x then 2500 else 0;

fun getScore(l,k)=
let
val l1 = map convert l
val l2 = ListMergeSort.sort (op<) l1
in
foldl (op+) 0 (List.take(l2, k))
end;
```

---

5 Formal Languages

Problem 5.1: Given the alphabet $A = \{0, 1\}$ and a $L = \cup_{i=1}^{\infty} L_i$ where 0pt

- $L_1 = \{\epsilon\}$
- $L_n = \{0x1, 1x0, 0x0, 1x1 | x \in L_{n-1}\}$

1. Write out explicitly what is in L_3. Show L contains all the words of even length.
2. Give another inductive definition of the language which contains all the words of even length.(Other than what is shown in the question, of course.)