

Q1.b:

Proposition: For any functions f and g and a continuation procedure $cont$, $(compose\$ f g cont) = (cont (compose f g))$.

Proof: for every parameter x

$$\begin{aligned}(compose\$ f g cont) x &= (f x (lambda(z) (g z cont))) = cont (g (f(x))) \\(cont (compose f g)) x &= cont (g (f (x))) = cont (g(f(x)))\end{aligned}$$

Proposition: For any list of functions fs and a continuation procedure $cont$, $(pipe\$ fs cont) = (cont (pipe fs))$.

Proof: By induction on the length of fs

Base: For the case of a fs of length 0 [the empty list], the value of $(pipe fs)$ is *void*, and the value of $(pipe\$ fs cont)$ is $(cont void)$, which implies $(pipe\$ fs cont) = (cont (pipe fs))$.

Induction step: We assume the proposition holds for fs of length n , and show the proposition holds for fs of a length $n+1$.

- (a) According to the code, the value of $(pipe fs)$ is $(compose (car fs) (pipe (cdr fs)))$.
- (b) According to the code, the value of $(pipe\$ fs cont)$ is $(compose\$ (car fs) (pipe\$ (cdr fs) cont) cont)$.

Since the first operand of $(compose\$ (car fs) (pipe\$ (cdr fs) cont) cont)$ is a n functions composed from the first to the n th function $(f1(f2(f3(f4(... fn(x)))))$), according to the induction assumption: $(cont (pipe (cdr fs))) = (pipe\$ (cdr fs) cont)$.

- $\Rightarrow (cont (cons (car fs) (pipe (cdr fs)))) = (pipe\$ (cdr fs) cont) ;;;$
- $\Rightarrow (cont (pipe fs)) = (pipe\$ (cdr fs) cont) ;;; (a)$
- $\Rightarrow (cont (pipe fs)) = (pipe\$ fs cont) ;;; (b)$

Q2.a:

כפי שלמדנו בהרצאות, הרעיון הכללי : שתי הפונקציות שקולות, אם עבור אותו קלט (פרמטרים) הן מחזירות אותו ערך, פורמלית:

נתונה פונקציה f עם תחום הגדרה D וטווח R

נתונה הפונקציה g

הפונקציה f שקולה לפונקציה g אם:

1. התחום של g הוא גם d , והטווח של g הוא גם R
2. לכל $x \in D$ מתקיים $f(x) = g(x)$
3. אם $f(x)$ זורקת exception, אז גם $g(x)$ זורקת exception
4. אם $f(x)$ לא מסתיימת, אז גם $g(x)$ לא מסתיימת.

Q2.b:

נוכיח ש הפונקציות $fibs1$, $fibs2$ שקולות:

נשים לב ש:

$fibs1$ Type: $[Void \rightarrow LzL(Number)]$

and

$fibs2$ Type: $[Void \rightarrow LzL(Number)]$

לשתי הפונקציות התחום והטווח שקולות.

שתי הפונקציות לא מסתיימות (תכונות lazyList).

שתי הפונקציות אינן זורקות שגיאות.

נשאר להראות ששתי הפונקציות מייצרות את אותה הרשימה

Proof:

Base: $(\text{head } fibs1) = (\text{head } fibs2) = 0$

$(\text{head } (\text{tail } fibs1)) = (\text{head } (\text{tail } fibs2)) = 1$

Induction step: we assume for every $k < n$ first k indexes in $fibs1$ and $fibs2$ are equal.

Assume the $n-2$ index in $fibs1$ is a , and the $n-1$ index in $fibs1$ is b .

From the assumption we get that $n-2$ index in $fibs2$ is also a , and $n-1$ index in $fibs2$ is also b .

We need to prove that the n th index in $fibs1$ is equal to the n th index in $fibs2$.

The n th index in **fibs1** is $(a+b)$, $(\text{fibgen } b (+ a b))$ is added to the tail of list, b is the $n-1$ index and the n th index is $a+b$.

The n th index in **fibs2** is $(a+b)$, $(\text{lz-lst-add } (\text{tail } lz1) (\text{tail } lz2))$ is called and it returns a lazylist that contains in the head the sum of the heads of $lz1$ and $lz2$, where $lz1$ head is the $n-1$ index and $lz2$ head is the $n-2$ index, therefore we get that the n th index is $a+b$.

Done.

Q3.1.a:

$\text{unify}[p(v(d(M),M,\text{ntuf3}),X)), p(v(d(B),v(B,\text{ntuf3}),KtM))]$

אוסף משוואות : $A = p(v(d(M),M,\text{ntuf3}),X)$, $B = p(v(d(B),v(B,\text{ntuf3}),KtM))$

אוסף המשוואות : $[v(d(M),M,\text{ntuf3}),X = v(d(B),v(B,\text{ntuf3}),KtM)]$

אוסף המשוואות : $v(d(M),M,\text{ntuf3}),X = v(d(B),v(B,\text{ntuf3}),KtM)$

3 פרמטרים != 2 פרמטרים

חלוקת משוואות לא חוקית.

לכן תוצאה כישלון.

Q3.1.b:

$\text{unify}[n(d(D),D,d,k,n(N),K),n(d(d),D,d,k,n(N),d)]$

אוסף משוואות : $A = n(d(D),D,d,k,n(N),K)$, $B = n(d(d),D,d,k,n(N),d)$

$[n(d(D),D,d,k,n(N),K) = n(d(d),D,d,k,n(N),d)]$

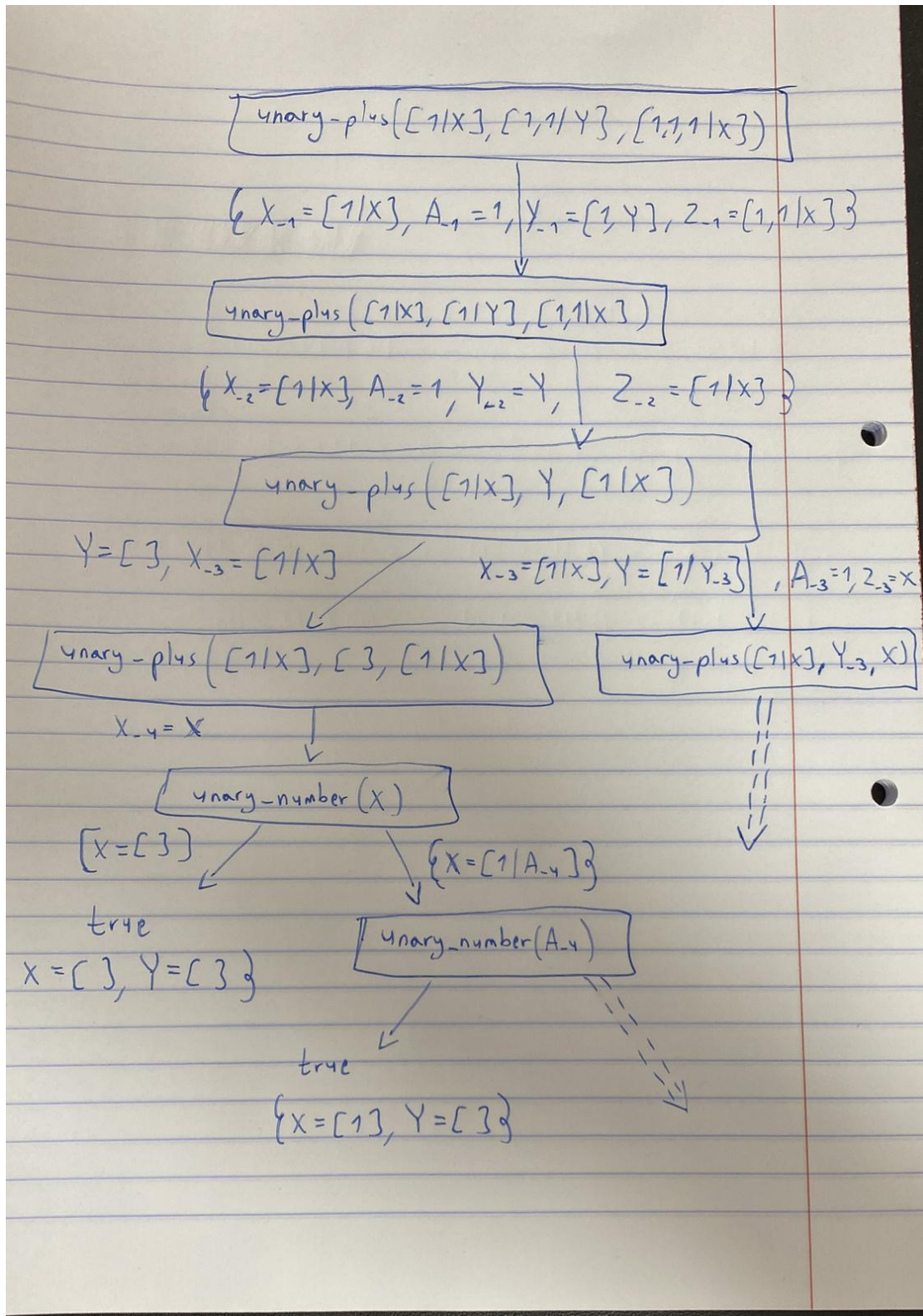
$[d(D),D,d,k,n(N),K = d(d),D,d,k,n(N),d]$

$[d(D) = d(d), D = D, d = d, k = k, n(N) = n(N), K = d]$

$[D = d, K = d]$

המאחד הכללי ביותר הוא ההצבה $\{D=d, K=d\}$

Q3.3.a:



Q3.3.b:

עץ הצלחה כי קיים לפחות מסלול הצלחה אחד סופי.

Q3.3.c:

עץ הצלחה אינסופי, העץ אינסופי וקיים בו מסלול הצלחה סופי.

Q3.3.d:

כן והיא מוצאת מספר אינסופי של פתרונות.

Q3.3.e:

לא מכיוון ש cons יכול להיות מבנה רקורסיבי, אז יתקיימו שאילתות שלא יסתיימו.