(1.b

שקולה_$pipe טענה: הפרוצדורה CPS לפרוצדורה pipe

לכל מספר טבעי $|fs| = n > 0$, $(pipe\$\ fs\ c) = (c\ (pipe\ fs))$,

טענת עזר: $(compose\$\ f\$\ g\$\ c) = (c\ (compose\ f\ g))$,

הוכחת טענת עזר:

$$a\_e[\ (compose\$\ f\$\ g\$\ c)\ ] => a\_e[(f\$\ \ x\ (lambda\ (res\_f)(g\$\ res\_f\ c))] =>$$

f$ and g$ are unary CPS functions

$$a\_e[(\ (lambda\ (res_f)(g\$\ res_f\ c))\ (f\ x))\ ] => a\_e[(g\$\ (f\ x)c)]\ =>$$

$$=> a\_e[(c\ (g\ (f\ x)))] => a\_e[(c\ (compose\ f\ g))]$$

הוכחת הטענה: נוכיח באינדוקציה

בסיס האינדוקציה: $|\{a\}| = |fs| = n = 1$

$$a\_e[\ (pipe\$\ fs\ c)\ ] =>$$

$$a\_e[\ (pipe\$\ fs\ c)\ ] => a\_e[\ (pipe\$\ \{a\}\ c)\ ] => a\_e[(c\ (car\ \{a\}))] => a\_e[(c\ a)]$$

$$=> a\_e[(c\ (pipe\ \{a\}))] => a\_e[(c\ (pipe\ fs))]$$

הטענה מתקיימת לכל $n \in N$ הנחת האינדוקציה: עבור $|fs| = n$ כלומר

$$(pipe\$\ fs\ c) = (c\ (pipe\ fs))$$

צעד האינדוקציה : עבור $|fs| = n > 1$

$$a\_e[(pipe\$\ fs\ c)] => a\_e[(compose\$\ (car\ fs)\ (pipe\$\ (cdr\ fs)\ c)\ id))]\ =>$$

לפי טענת עזר נקבל

$$=> a\_e[(id\ (compose(car\ fs)\ (pipe\$\ (cdr\ fs)\ c)\ )]$$

$$=> a\_e[(compose(car\ fs)(pipe\$\ (cdr\ fs)c)]$$

לפי הנחת האנדוקציה

$$=> a\_e[(compose(car\ fs)(c\ (pipe\ (cdr\ fs)))]$$

לפי הגדרת pipe

$$=>a\_e[(c\ (pipe\ fs))]$$

(2.a)

הרשימות העצלות lz2 וlz1 שוות אם הן סופיות אז יש להן אותו אורך או שתי הרשמות

אינסופיות. האיברים של שתי הרשמות שווים במקביל.

(2.b)

שתי הרשמות אינסופיות כי קוד ההמשך תמיד מייצר איברים בשתי הרשמות. נוסף לכך, האיברים של שתי הרשמות שווים במקביל

b ו a רצופים האיבר אחרי הb בשתי הרשמות הוא (a + b)

Fibs1אז :

$$a\_e[(fibgen\ a\ b)] => a\_e[(lambda\ (a\ b)\ (cons\ -lzl\ a\ (lambda\ ()\ (fibgen\ b\ (\ +a\ b)))]$$

האיבר הראשון ברשימה הזו הוא a ו אחריו הואb ו אחריו (a + b)

Fibs2אז :

האיבר הראשון ברשימה הזו הוא a ו האיבר הראשון ברשימה הזנב הואb

אז בחיבור שתי הרשמות

האלו נקבל רשמה שהאיבר הראשון בה הוא (a + b)

Question 3 - Logic programming

3.1 Unification

What is the result of the operations? Provide all the algorithm steps. Explain in case of failure.

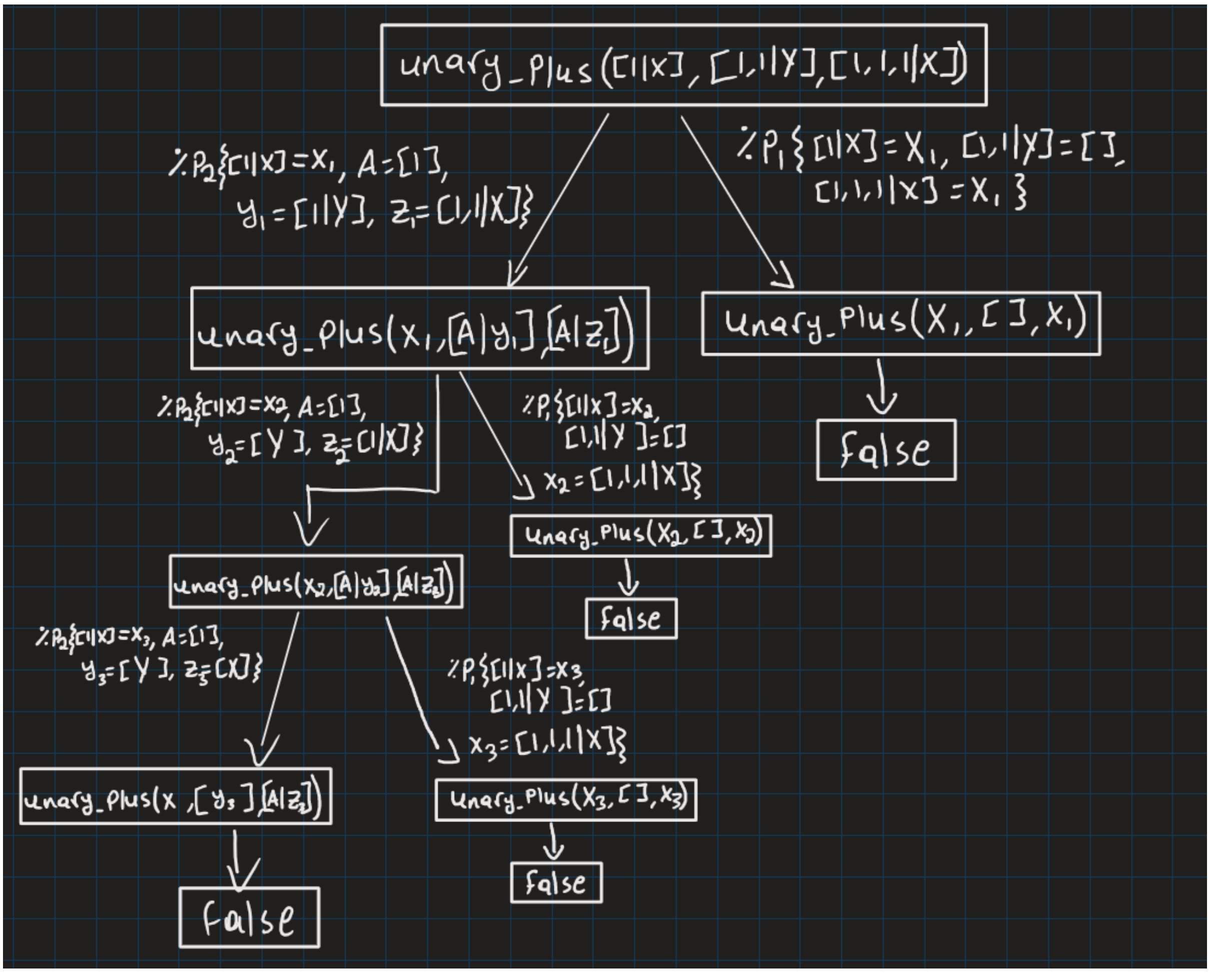
a. unify[p(v(v(d(M),M,ntuf3),X)), p(v(d(B),v(B,ntuf3),KtM))]
   1. v = d
   2. d = B

      we get failure in this case because d(B) cant be equal to B(B).

b. unify[n(d(D),D,d,k,n(N),K),n(d(d),D,d,k,n(N),d)]
   1. D = d
   2. K = D

3.3 Proof tree

a.



b. a failure proof tree

c. this is a finite tree

d. You can’t Prove it from the given program.

e. Yes, its decidable