Φροντιστήριο 7 – Λύσεις Ασκήσεων

Άσκηση 1

Οι προτάσεις σε Κατηγορηματικό Λογισμό:

1. Όλα τα παιδιά αγαπούν τον Άγιο Βασίλη
   $\phi1 = \forall x\, (Ch(x) \rightarrow L(x, Santa))$
2. Όποιος αγαπά τον Άγιο Βασίλη αγαπά και τα ελάφια
   $\phi2 = \forall y\, (L(y, Santa) \rightarrow (\forall z\, (RD(z) \rightarrow L(y,z)))$
3. Ο Ρούντολφ είναι ελάφι και έχει κόκκινη μύτη.
   $\phi3 = RD(Rudolph) \wedge RN(Rudolph)$
4. Οτιδήποτε έχει κόκκινη μύτη μπορεί να είναι είτε παλιάτσος είτε παράξενο.
   $\phi4 = \forall w\, (RN(w) \rightarrow (CL(w) \vee W(w)))$
5. Κανένα ελάφι δεν είναι παλιάτσος
   $\phi5 = \forall v\, (RD(v) \rightarrow \neg\, CL(v))$
6. Ο Σκρούτζ δεν αγαπά οτιδήποτε παράξενο
   $\phi6 = \forall u\, (W(u) \rightarrow \neg L(Scrooge\ u))$
7. Ο Σκρούτζ δεν είναι παιδί.
   $\psi = \neg\, Ch(Scrooge)$

Μας ενδιαφέρει να δείξουμε ότι η πρόταση $(\phi1\wedge..\wedge\phi6)\rightarrow\psi$ είναι έγκυρη, το οποίο ισχύει αν και μόνο αν η άρνηση της πρότασης δεν είναι ικανοποιήσιμη. Επομένως, μεταφράζουμε την πρόταση $\neg[(\phi1\wedge\ldots\wedge\phi6)\rightarrow\psi]$ σε προτασιακή μορφή και εφαρμόζουμε τη Μέθοδο της Επίλυσης για να τη διαψεύσουμε, έτσι δείχνοντας ότι ο αρχικός συλλογισμός $(\phi1\wedge\ldots\wedge\phi6)\rightarrow\psi$ είναι έγκυρος.

Μετατροπή σε Κανονική Μορφή Prenex:

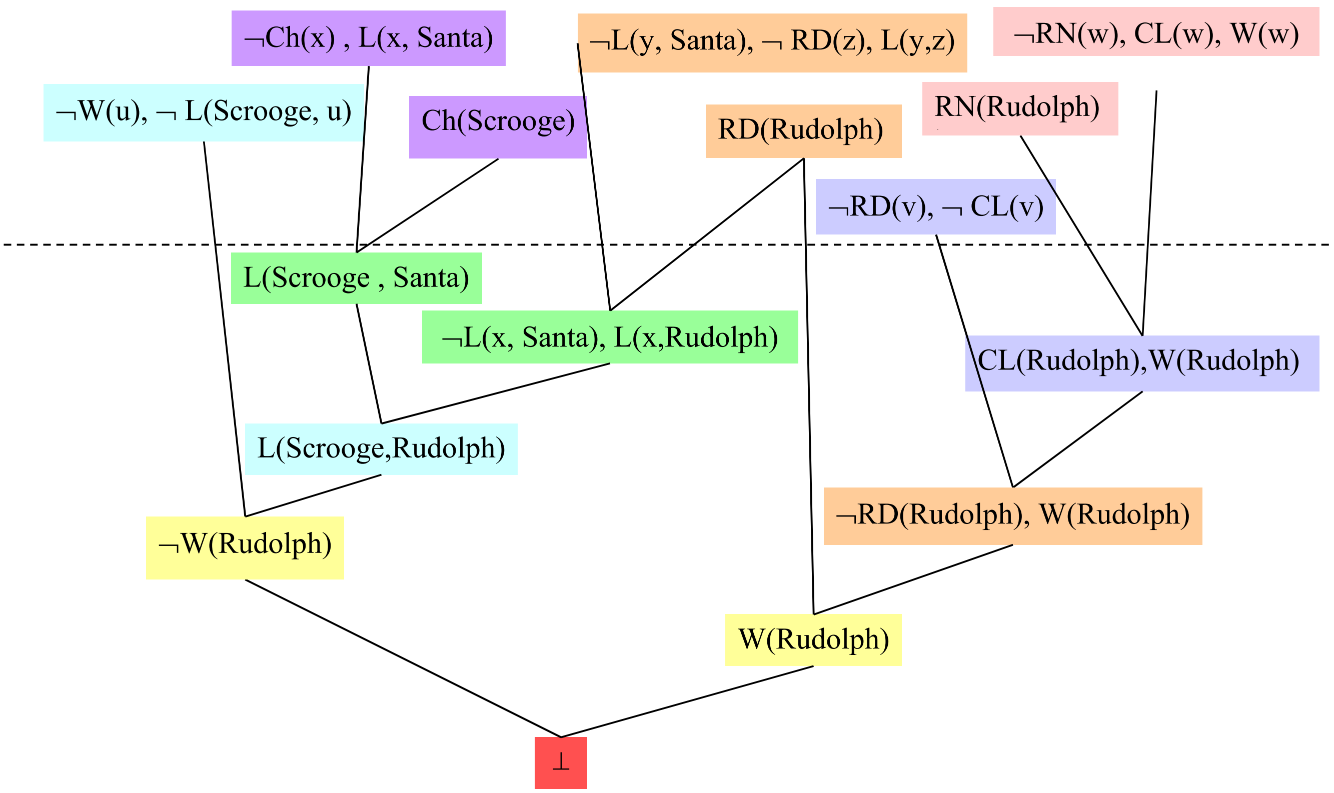
1. $\forall x\, (\neg Ch(x) \vee L(x, Santa))$
2. $\forall y\, (L(y, Santa) \rightarrow (\forall z\, (RD(z) \rightarrow L(y,z)))$
   $= \forall y\, (L(y, Santa) \rightarrow (\forall z\, (\neg RD(z) \vee L(y,z)))$
   $= \forall y\, (\neg L(y, Santa) \vee (\forall z\, (\neg RD(z) \vee L(y,z)))$
   $= \forall y \forall z\, (\neg L(y, Santa) \vee \neg RD(z) \vee L(y,z))$
3. $RD(Rudolph) \wedge RN(Rudolph)$
4. $\forall w\, (RN(w) \rightarrow (CL(w) \vee W(w)))$
   $= \forall w\, (\neg RN(w) \vee (CL(w) \vee W(w)))$
5. $\forall v\, (RD(v) \rightarrow \neg\, CL(v))$
   $= \forall v\, (\neg RD(v) \vee \neg\, CL(v))$
6. $\forall u\, (W(u) \rightarrow \neg L(Scrooge, u))$
   $= \forall u\, (\neg W(u) \vee \neg\, L(Scrooge, u)$
7. $Ch(Scrooge)$ (άρνηση της πρότασης)

Κατασκευή Προτασιακών Μορφών:

1. $\{\{\neg Ch(x)\ , L(x, Santa)\}\}$

2. {{¬L(y, Santa), ¬ RD(z), L(y,z)}}
3. {{RD(Rudolph)}, {RN(Rudolph)}}
4. {{¬RN(w), CL(w), W(w)}}
5. {{¬RD(v), ¬ CL(v)}}
6. {{¬W(u), ¬ L(Scrooge,u)}}
7. {{Ch(Scrooge)}}

Εφαρμογή της Μεθόδου Επίλυσης μέσω Ενοποίησης:



Συμπέρασμα: Ο συλλογισμός είναι έγκυρος.

<u>Άσκηση 2</u>

Θα εφαρμόσουμε τη Μέθοδο της Επίλυσης στο συλλογισμό που ακολουθεί.

$\forall x \exists y$ Father(x,y) $\rightarrow$ $\exists y \forall x$ Father (x,y).

Βήμα 1: Μετατροπή της άρνησης της πρότασης σε ΚΜΡ

$\neg[\ \forall x \exists y$ Father(x,y) $\rightarrow$ $\exists y \forall x$ Father (x,y)]

$\equiv \neg[\neg \forall x \exists y$ Father(x,y) $\vee$ $\exists y \forall x$ Father (x,y)]

$\equiv \forall x \exists y$ Father(x,y) $\wedge$ $\neg \exists y \forall x$ Father (x,y)

$\equiv \forall x \exists y$ Father(x,y) $\wedge$ $\forall y \exists x$ $\neg$Father (x,y)

$\equiv \forall x_1 \exists y_1$ Father(x_1,y_1) $\wedge$ $\forall y_2 \exists x_2$ $\neg$Father (x_2,y_2)

$\equiv \forall x_1 \exists y_1 \forall y_2 \exists x_2$ [Father(x_1,y_1) $\wedge \neg$Father (x_2,y_2)]

Βήμα 2: Απαλοιφή ποσοδεικτών

Father(x_1,f(x_1)) $\wedge \neg$Father (g(x_1, y_2),y_2)

Βήμα 3: Κατασκευή προτασιακού συνόλου

{{Father(x_1,f(x_1))},{$\neg$Father (g(x_1, y_2),y_2)}}

Αν αγνοήσουμε τον κανόνα *ελεγχου-εμφάνισης* (occurs-check) ο οποίος αναφέρει ότι μία μεταβλητή είναι ενοποιήσιμη με μία συνάρτηση μόνο εάν η μεταβλητή δεν εμφανίζεται στη συνάρτηση, τότε τα προτασιακά σύνολα {Father(x_1,f(x_1))} και {Father (g(x_1, y_2),y_2)} είναι ενοποιήσιμα γεγονός που μας οδηγεί σε διάψευση. Εντούτοις, ο αρχικός συλλογισμός δεν είναι έγκυρος: για παράδειγμα, αν θεωρήσουμε ως σύμπαν όλους τους ανθρώπους και ως Father(x,y) τη σχέση παιδιού/ πατέρα, προφανώς, ο συλλογισμός είναι ψευδής.

Επιπρόσθετα, αν επιτρέψουμε την ενοποίηση του x με μια συνάρτηση του τύπου f(x) (κάτι που επιτρέπεται π.χ. από τον compiler της Prolog για λόγους απόδοσης), κατά την αντικατάσταση του x από το f(x) θα εισέλθουμε σε ένα ατέρμονο βρόχο κατά τον οποίο σε κάθε αντικατάσταση του x θα δημιουργείται μια νέα εμφάνιση του x και κατ' επέκταση ανάγκη για καινούρια αντικατάσταση. Επομένως, δεν θα είναι πάντα εφικτό να προσδιορίσουμε τιμή του x τέτοια ώστε x = f(f(….. (x)…))

<u>Άσκηση 3</u>

(α) Αριθμούμε τις γραμμές του προγράμματος.

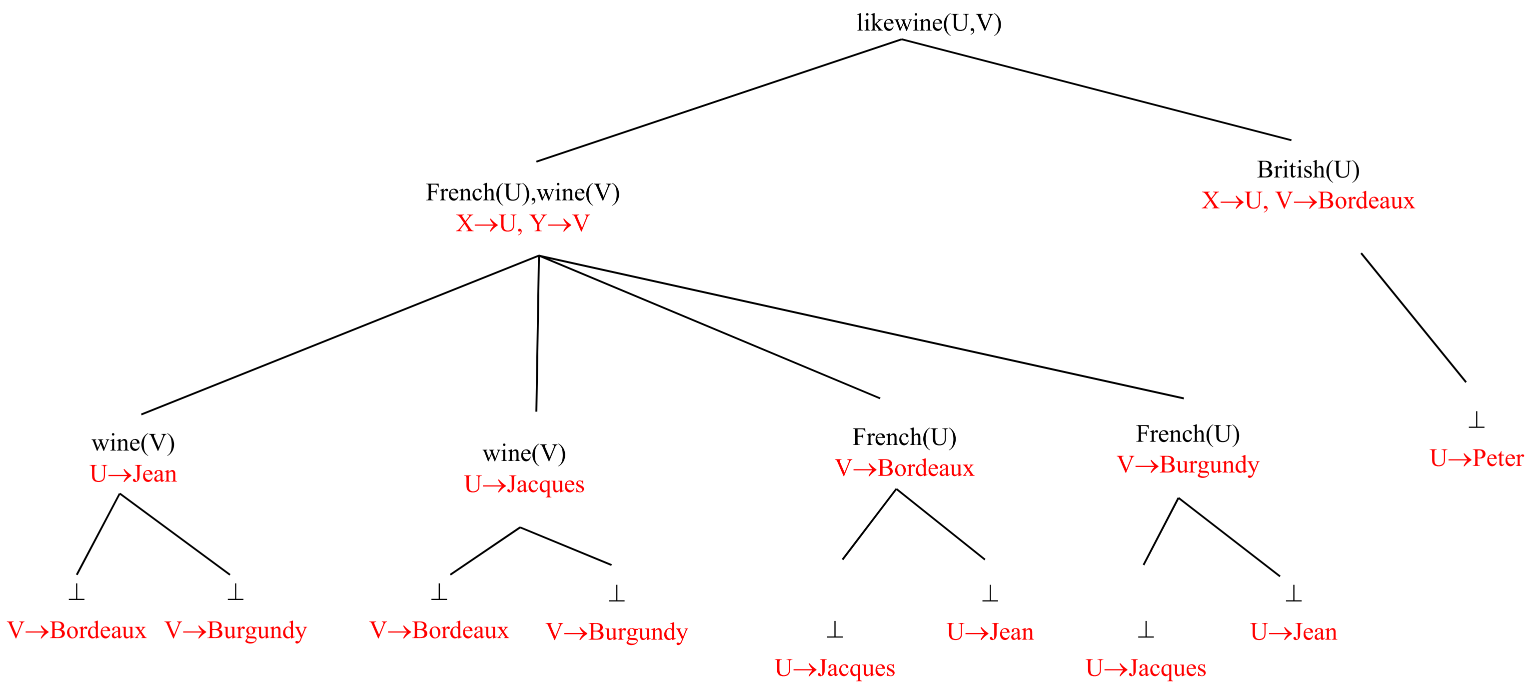
1. French(Jean)
2. French(Jacques)
3. British(Peter)
4. likewine(X, Y) $\leftarrow$ French(X), wine(Y)
5. likewine(X, Bordeaux) $\leftarrow$ British(X)
6. wine(Burgundy)
7. wine(Bordeaux)
8. $\leftarrow$ likewine(U, V)

και εφαρμόζουμε SLD-επίλυση ως εξής:

9.	$\leftarrow$French(U), wine(V)	από (8) και (4) με ενοποιήτρια συνάρτηση σ = {U/X, V/Y}
10.	$\leftarrow$ wine(V)	από (9) και (1) με ενοποίηση {Jean/U}
11.	$\perp$	από (6) και (10) με ενοποίηση {Burgundy/V}

Αντικατάσταση ορθής απάντησης: σ = {Jean/U, Burgundy/V}

(β)



Οι αντικαταστάσεις ορθής απάντησης που προκύπτουν είναι:

{Jean/U, Bordeaux/V}
{Jean/U, Burgundy/V}
{Jacques/U, Bordeaux/V}
{Jacques/U, Burgundy/V}
{Peter/U, Burgundy/V}

Άσκηση 2

(α) Αριθμούμε τις γραμμές του προγράμματος:

1. add(X, 0, X)
2. add(X, succ(Y), succ(Z)) ← add(X, Y, Z)
3. ← add (succ (succ (0)), succ (succ (0)), U)

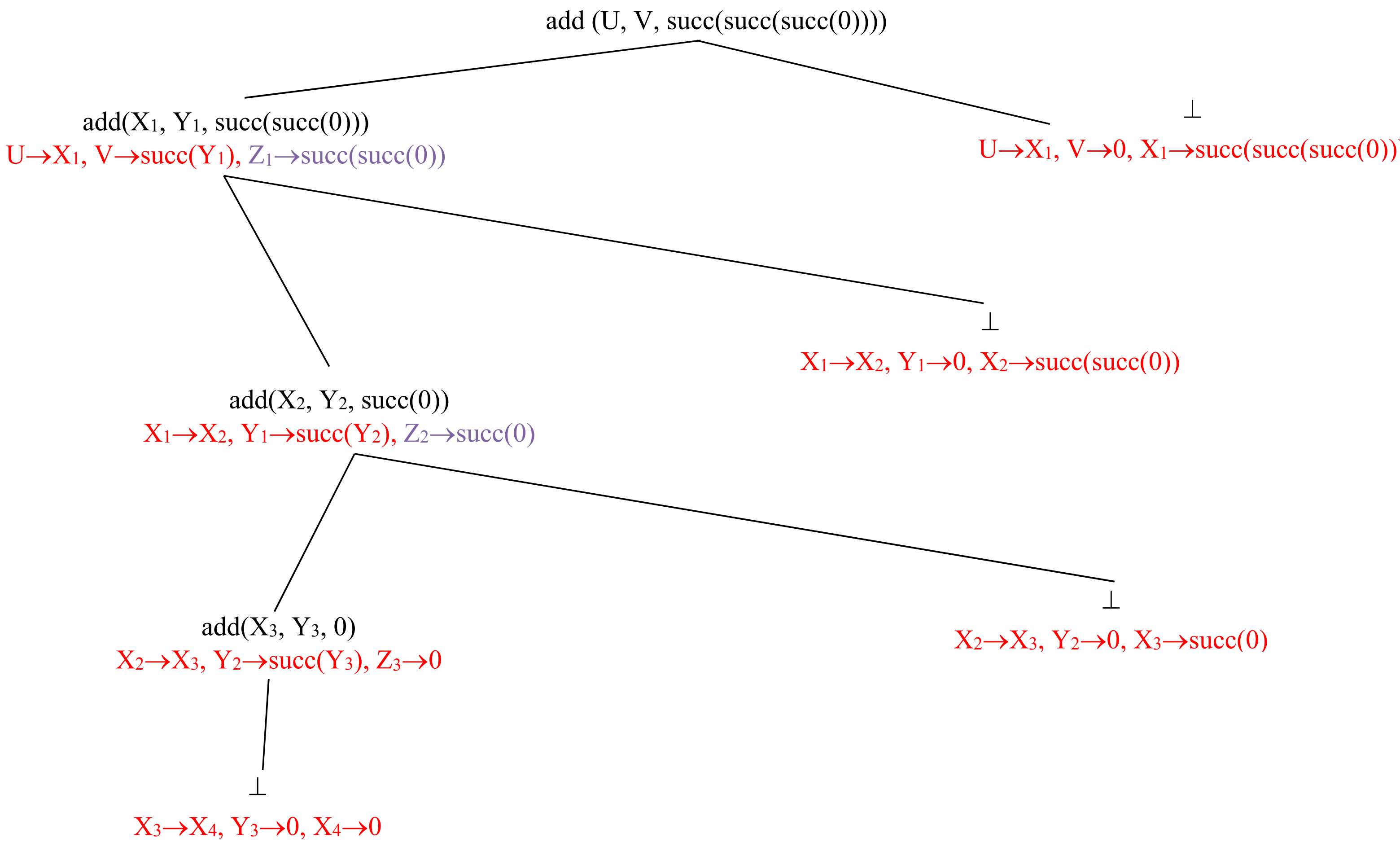
Στο (2) χρησιμοποιούμε μετονομασία μεταβλητών:
add(X', succ(Y'), succ(Z'))
← add(X', Y', Z')

και εφαρμόζουμε SLD-επίλυση ως εξής:

4.	← add (succ (succ (0)), succ (0), Z)	από (2), (3) και μέσω της αντικατάστασης: {succ(succ(0))/X, succ(0)/Y, succ(Z)/U}
5.	← add (succ (succ (0)), 0,Z')	από (2), (4) και μέσω της αντικατάστασης: {succ(succ(0))/X', 0/Y', succ(Z')/Z}
6.	⊥	από (1), (5) και μέσω της αντικατάστασης: {succ(succ(0))/X, succ(succ(0))/Z'}

Η αντικατάσταση ορθής απάντησης είναι η {succ(succ(succ(succ(0))))/U}.

(β) Το ζητούμενο SLD-δένδρο φαίνεται την επόμενη σελίδα


add (U, V, succ(succ(succ(0))))
add(X1, Y1, succ(succ(0)))
U→X1, V→succ(Y1), Z1→succ(succ(0))
⊥
U→X1, V→0, X1→succ(succ(succ(0)))
add(X2, Y2, succ(0))
X1→X2, Y1→succ(Y2), Z2→succ(0)
⊥
X1→X2, Y1→0, X2→succ(succ(0))
add(X3, Y3, 0)
X2→X3, Y2→succ(Y3), Z3→0
⊥
X2→X3, Y2→0, X3→succ(0)
⊥
X3→X4, Y3→0, X4→0

Άσκηση 3

Αριθμούμε τις γραμμές του προγράμματος:

1. append(X:L1, L2, X:L3) ← append(L1, L2, L3)
2. append(nil, L1, L1)
3. ← append([a,b], [b,c], Z)

και εφαρμόζουμε SLD-επίλυση ως εξής:

4. ← append([b], [b,c],L3) από (1), (3) και μέσω της αντικατάστασης {a/X, [b]/L1, [b,c]/L2, a:L3/Z)}
5. ← append(nil, [b,c], L3') από (1), (4) και μέσω της αντικατάστασης { b/X', nil/L1', [b,c]/L2', b:L3'/L3)}
6. ⊥ από (2), (5) και μέσω της αντικατάστασης {[b,c]/L1'', [b,c]/L3'}

(1) με μετονομασία μεταβλητών:
append(X':L1', L2', X':L3') ← append(L1', L2', L3')

(2) με μετονομασία μεταβλητών:
append(nil, L1'', L1'')

Αντικατάσταση ορθής απάντησης: σ = {[a,b,b,c]/Z}