ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τεχνητή Νοημοσύνη

5ο φροντιστήριο (2023-24)

Επιμέλεια: Σοφία Ελευθερίου,
Φοίβος Χαραλαμπάκος

Άσκηση 11.1

α) Μετατρέψτε σε κανονική συζευκτική μορφή (CNF) τους τύπους

$\exists x \left(Course(x) \wedge Passed(John, x)\right)$

$\forall x \left(\left(Student(x) \wedge \exists y \left(Course(y) \wedge Passed(x, y)\right)\right) \Rightarrow Clever(x)\right)$

$\exists x \left(Course(x) \wedge Passed(John, x)\right)$

$Course(\mathrm{C}_1) \wedge Passed(John, C_1)$

$\forall x \left(\left(Student(x) \wedge \exists y \left(Course(y) \wedge Passed(x, y)\right)\right) \Rightarrow Clever(x)\right)$ $\quad \alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$

$\forall x \left(\neg\left(Student(x) \wedge \exists y \left(Course(y) \wedge Passed(x, y)\right)\right) \vee Clever(x)\right)$ $\quad \neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$

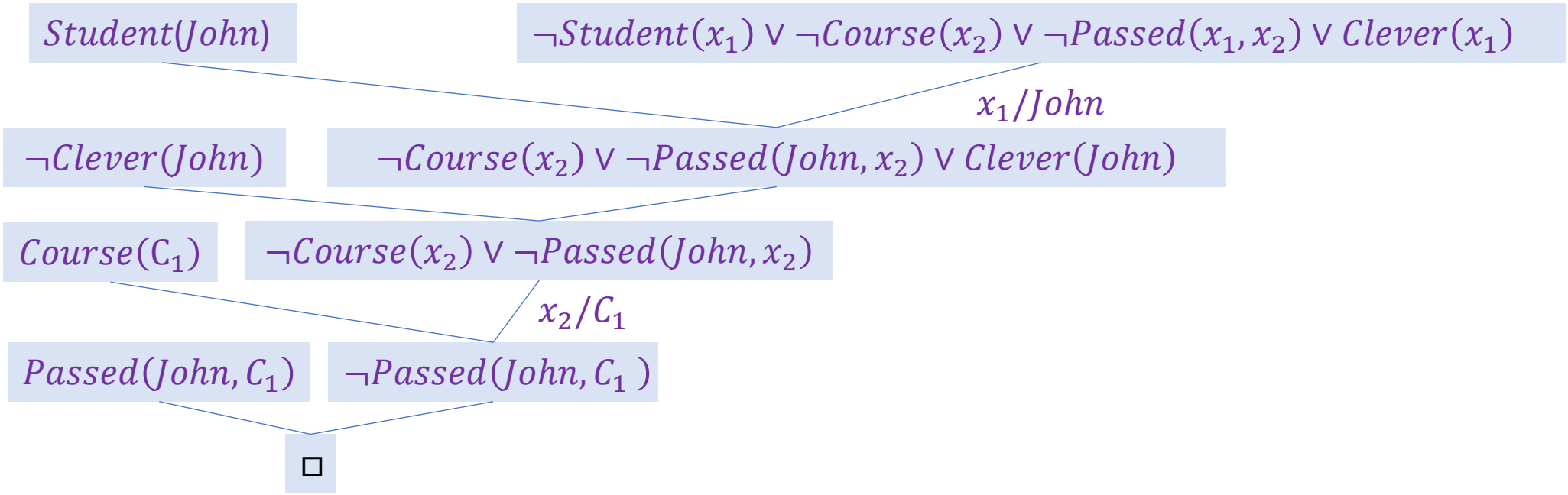
$\forall x \left(\left(\neg Student(x) \vee \neg\exists y \left(Course(y) \wedge Passed(x, y)\right)\right) \vee Clever(x)\right)$ $\quad \neg\exists \mathrm{x} \mathrm{P} \equiv \forall \mathrm{x} \neg \mathrm{P}$

$\forall x \left(\left(\neg Student(x) \vee \forall y \neg\left(Course(y) \wedge Passed(x, y)\right)\right) \vee Clever(x)\right)$ $\quad \neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$

$\forall x \left(\left(\neg Student(x) \vee \forall y \left(\neg Course(y) \vee \neg Passed(x, y)\right)\right) \vee Clever(x)\right)$

$\neg Student(x_1) \vee \neg Course(y) \vee \neg Passed(x, y) \vee Clever(x)$

β) Σχεδιάστε δέντρο απόδειξης που να δείχνει με απαγωγή σε άτοπο χρησιμοποιώντας μόνο τον κανόνα της ανάλυσης (resolution) ότι από τις προτάσεις
$Student(John)$, $Course(\mathrm{C}_1)$, $Passed(John, C_1)$ και
$\neg Student(x_1) \vee \neg Course(y) \vee \neg Passed(x, y) \vee Clever(x)$
ότι μπορούμε να συμπεράνουμε πως ο Γιάννης είναι έξυπνος.
Εισάγουμε στη βάση γνώσης και την άρνηση του αποδεικτέου, δηλαδή $\neg Clever(John)$ Το ζητούμενο δέντρο είναι:



Άσκηση 11.2

α) Μετατρέψτε τους τύπους

i) $\exists x\ \exists y\ \big(Dog(x) \wedge Cat(y) \wedge Bite(x,y)\big)$

ii) $\forall y\ \big(Cat(y) \Rightarrow \neg Likes(y, Suzos)\big)$

iii) $\forall x\ \Big(\big(Dog(x) \wedge \exists y\ \big(Cat(y) \wedge Bite(x,y)\big)\big) \Rightarrow \forall z\ \big(Cat(z) \Rightarrow \neg Likes(z,x)\big)\Big)$

σε κανονική συζευκτική μορφή (CNF).

i) $\exists x\ \exists y\ \big(Dog(x) \wedge Cat(y) \wedge Bite(x,y)\big)$

$Dog(C_1) \wedge Cat(C_2) \wedge Bite(C_1, C_2)$

ii) $\forall y\ \big(Cat(y) \Rightarrow \neg Likes(y, Suzos)\big)$

$\forall y\ \big(\neg Cat(y) \vee \neg Likes(y, Suzos)\big)$

$\neg Cat(x_1) \vee \neg Likes(x_1, Suzos)$

iii) $\forall x\ \Big(\big(Dog(x) \wedge \exists y\ \big(Cat(y) \wedge Bite(x,y)\big)\big) \Rightarrow \forall z\big(Cat(z) \Rightarrow \neg Likes(z,x)\big)\Big)$

$\forall x\ \Big(\neg\big(Dog(x) \wedge \exists y\ \big(Cat(y) \wedge Bite(x,y)\big)\big) \vee \forall z\big(\neg Cat(z) \vee \neg Likes(z,x)\big)\Big)$

$\forall x\ \Big(\big(\neg Dog(x) \vee \neg\exists y\ \big(Cat(y) \wedge Bite(x,y)\big)\big) \vee \forall z\big(\neg Cat(z) \vee \neg Likes(z,x)\big)\Big)$

$\forall x\ \Big(\big(\neg Dog(x) \vee \forall y\neg\big(Cat(y) \wedge Bite(x,y)\big)\big) \vee \forall z\big(\neg Cat(z) \vee \neg Likes(z,x)\big)\Big)$

$\forall x\ \Big(\big(\neg Dog(x) \vee \forall y\big(\neg Cat(y) \vee \neg Bite(x,y)\big)\big) \vee \forall z\big(\neg Cat(z) \vee \neg Likes(z,x)\big)\Big)$

$\neg Dog(x_2) \vee \neg Cat(x_3) \vee \neg Bite(x_2, x_3) \vee \neg Cat(x_4) \vee \neg Likes(x_4, x_2)$

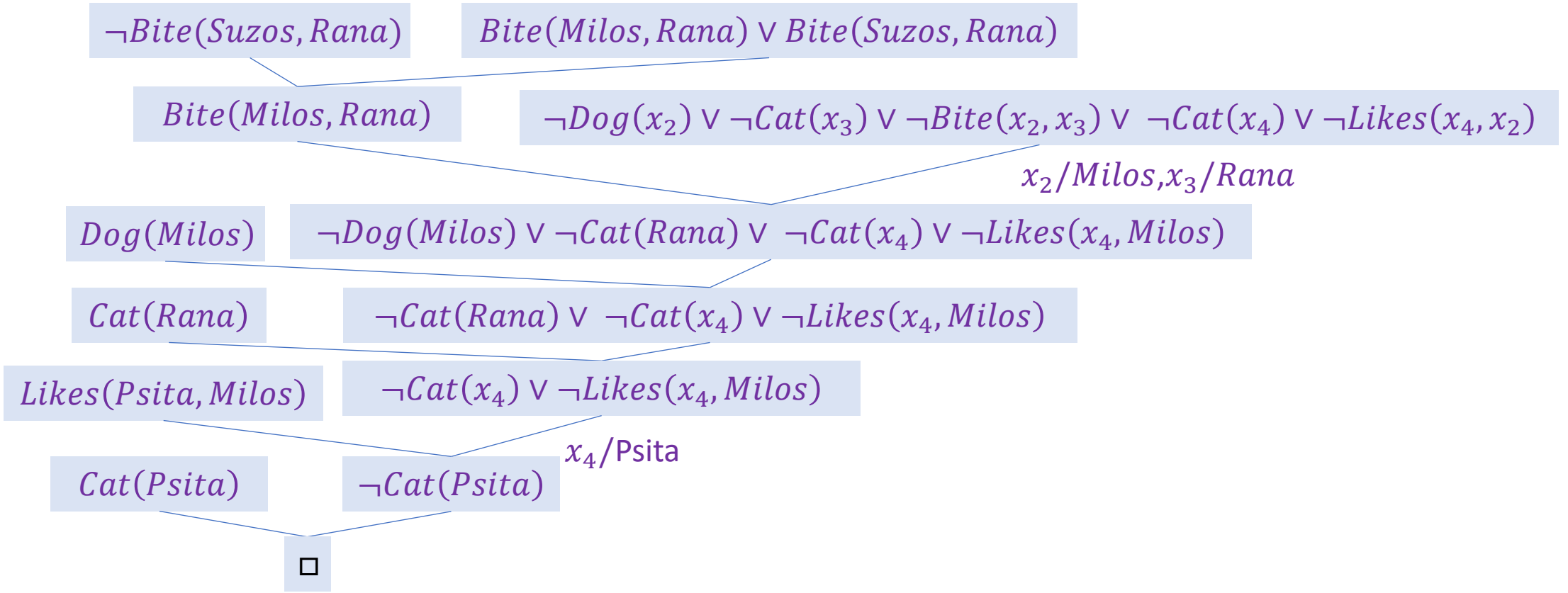
β) Χρησιμοποιώντας μόνο τον κανόνα της ανάλυσης (resolution), κατασκευάστε δέντρο απόδειξης που να αποδεικνύει με απαγωγή σε άτοπο πως από τους τύπους $\boldsymbol{Dog(Milos) \wedge Dog(Suzos),\ Cat(Psita) \wedge Cat(Rana)}$

$Likes(Psita, Milos)$ $Bite(Milos, Rana) \vee Bite(Suzos, Rana)$

$\neg Dog(x_2) \vee \neg Cat(x_3) \vee \neg Bite(x_2, x_3) \vee \neg Cat(x_4) \vee \neg Likes(x_4, x_2)$

προκύπτει ως συμπέρασμα ότι τη Ράνα τη δάγκωσε ο Σούζος.

Η άρνηση του αποδεικτέου είναι: $\neg Bite(Suzos, Rana)$ Το δέντρο απόδειξης είναι:



γ) Εξηγήστε πώς θα μπορούσαμε να κατασκευάσουμε αυτόματα το δέντρο απόδειξης με αναζήτηση σε χώρο καταστάσεων.

γ1) Τι θα παρίστανε κάθε κατάσταση;

Κάθε κατάσταση θα περιείχε

- ένα (πιθανώς ημιτελές) δέντρο απόδειξης
- τους αρχικούς τύπους της ΒΓ και την άρνηση του αποδεικτέου (σε μορφή CNF, για την ακρίβεια κάθε διάζευξη τύπου CNF θα παριστανόταν ως ξεχωριστός τύπος),
- τους τύπους που έχουν προκύψει ως συμπεράσματα από τις εφαρμογές του κανόνα της ανάλυσης του δέντρου.

γ2) Ποια θα ήταν η αρχική κατάσταση και ποιες θα ήταν οι τελικές καταστάσεις;

Η αρχική κατάσταση θα περιείχε ένα κενό δέντρο απόδειξης, τους αρχικούς τύπους της ΒΓ και την άρνηση του αποδεικτέου (σε μορφή CNF). Τελική θα ήταν μια κατάσταση της οποίας το δέντρο απόδειξης καταλήγει σε κενή διάζευξη.

γ3) Ποιοι θα ήταν οι τελεστές μετάβασης;

Θα υπήρχε μόνο ένας τελεστής μετάβασης, ο οποίος θα επέκτεινε το δέντρο της τρέχουσας κατάστασης εφαρμόζοντας τον κανόνα της ανάλυσης σε ένα από τα φύλλα του δέντρου και έναν από τους διαθέσιμους τύπους της κατάστασης. Ο τελεστής θα αποθήκευε επίσης το συμπέρασμα που θα προέκυπτε από την εφαρμογή του κανόνα, ως πρόσθετο τύπο της νέας κατάστασης.

γ4) Σχεδιάστε το δέντρο αναζήτησης που θα κατασκεύαζε ο αλγόριθμος αναζήτησης πρώτα σε πλάτος (BFS) στην περίπτωση του σκέλους (γ). Αρκεί να σχεδιάσετε τη ρίζα και δύο παιδιά της.

Άσκηση 11.3

α) Παραστήστε σε πρωτοβάθμια κατηγορηματική λογική τις σημασίες των παρακάτω ελληνικών προτάσεων. Συμβολίστε με $z = x$ (ή $z \neq x$) ότι οι μεταβλητές z και x παριστάνουν (ή όχι) την ίδια οντότητα.

i) Υπάρχει ένας σκύλος που γαβγίζει και φοβάται όλες τις γάτες:

$$\exists x \left(IsDog(x) \wedge Barks(x) \wedge \forall y \left(IsCat(y) \Rightarrow IsAfraidOf(x, y) \right) \right)$$

ii) Ο Μίλος φοβάται τουλάχιστον μία γάτα που φοβάται τουλάχιστον ένα σκύλο:

$$\exists x \exists y \left(IsCat(x) \wedge IsDog(y) \wedge IsAfraidOf(Milos, x) \wedge IsAfraidOf(x, y) \right)$$

iii) Κάθε σκύλος φοβάται κάθε γάτα που τον φοβάται:

$$\forall x \forall y \left(\left(IsDog(x) \wedge IsCat(y) \wedge IsAfraidOf(y, x) \right) \Rightarrow IsAfraidOf(x, y) \right)$$

iv) Κάθε γάτα φοβάται τουλάχιστον ένα σκύλο (πιθανώς διαφορετικό για κάθε γάτα):

$$\forall y \left(IsCat(y) \Rightarrow \exists x \left(IsDog(x) \wedge IsAfraidOf(y, x) \right) \right)$$

v) Κάθε γάτα φοβάται τουλάχιστον ένα σκύλο (πιθανώς διαφορετικό για κάθε γάτα) που τη φοβάται:

$$\forall y \left(IsCat(y) \Rightarrow \exists x \left(IsDog(x) \wedge IsAfraidOf(y, x) \wedge IsAfraidOf(x, y) \right) \right)$$

vi) Κάθε σκύλος φοβάται <u>ακριβώς</u> δύο γάτες (πιθανώς διαφορετικές για κάθε σκύλο):

$$\forall x \left(IsDog(x) \right.$$
$$\Rightarrow \exists y_1 \exists y_2 \left(IsCat(y_1) \wedge IsCat(y_2) \wedge y_1 \right.$$
$$\left. \left. \neq y_2 \wedge IsAfraidOf(x, y_1) \wedge IsAfraidOf(x, y_2) \wedge \forall z \left(\left(IsCat(z) \wedge IsAfraidOf(x, z) \right) \Rightarrow (z = y_1 \vee z = y_2) \right) \right) \right)$$

β) Μετατρέψτε τις προτάσεις (iii), (iv) και την άρνηση της (v) του σκέλους (α) σε προτάσεις Horn πρωτοβάθμιας κατηγορηματικής λογικής.

iii)

$$\forall x \forall y \left(\left(IsDog(x) \wedge IsCat(y) \wedge IsAfraidOf(y, x) \right) \Rightarrow IsAfraidOf(x, y) \right)$$
$$\forall x \forall y \left(\neg \left(IsDog(x) \wedge IsCat(y) \wedge IsAfraidOf(y, x) \right) \vee IsAfraidOf(x, y) \right)$$
$$\forall x \forall y \left(\left(\neg IsDog(x) \vee \neg IsCat(y) \vee \neg IsAfraidOf(y, x) \right) \vee IsAfraidOf(x, y) \right)$$
$$\neg IsDog(x) \vee \neg IsCat(y) \vee \neg IsAfraidOf(y, x) \vee IsAfraidOf(x, y)$$

iv)

$$\forall y \left(IsCat(y) \Rightarrow \exists x \left(IsDog(x) \wedge IsAfraidOf(y,x) \right) \right)$$

$$P \Rightarrow Q \equiv \neg P \vee Q$$

$$\forall y \left(\neg IsCat(y) \vee \exists x \left(IsDog(x) \wedge IsAfraidOf(y,x) \right) \right)$$

Απαλοιφή του υπαρξιακού ποσοδείκτη

$$\forall y \left(\neg IsCat(y) \vee \left(IsDog\left(F(y)\right) \wedge IsAfraidOf\left(y, F(y)\right) \right) \right)$$

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$

$$\forall y \left(\left(\neg IsCat(y) \vee IsDog\left(F(y)\right) \right) \wedge \left(\neg IsCat(y) \vee IsAfraidOf\left(y, F(y)\right) \right) \right)$$

Απαλοιφή του καθολικού ποσοδείκτη

$$\left(\neg IsCat(y) \vee IsDog\left(F(y)\right) \right) \wedge \left(\neg IsCat(y) \vee IsAfraidOf\left(y, F(y)\right) \right)$$

Επομένως, προκύπτουν δύο προτάσεις

$$\neg IsCat(y) \vee IsDog\left(f(y)\right)$$

$$\neg IsCat(y) \vee IsAfraidOf\left(y, F(y)\right)$$

Άρνηση της v)

$$\neg\forall y \left(IsCat(y) \Rightarrow \exists x \left(IsDog(x) \wedge IsAfraidOf(y,x) \wedge IsAfraidOf(x,y)\right)\right)$$

$$\mathrm{P} \Rightarrow Q \equiv \neg P \vee Q$$

$$\neg\forall y \left(\neg IsCat(y) \vee \exists x \left(IsDog(x) \wedge IsAfraidOf(y,x) \wedge IsAfraidOf(x,y)\right)\right)$$

$$\neg\forall \mathrm{y}\mathrm{P} \equiv \exists y \neg P$$

$$\exists y \neg \left(\neg IsCat(y) \vee \exists x \left(IsDog(x) \wedge IsAfraidOf(y,x) \wedge IsAfraidOf(x,y)\right)\right)$$

$$\neg(P \vee Q) \equiv \neg\mathrm{P} \wedge \neg\mathrm{Q}$$

$$\exists y \left(IsCat(y) \wedge \neg\exists x \left(IsDog(x) \wedge IsAfraidOf(y,x) \wedge IsAfraidOf(x,y)\right)\right)$$

$$\neg\exists \mathrm{x}\mathrm{P} \equiv \forall \mathrm{x} \neg \mathrm{P}$$

Άρνηση της v)

$$\exists y \left(IsCat(y) \wedge \forall x \neg \left(IsDog(x) \wedge IsAfraidOf(y,x) \wedge IsAfraidOf(x,y) \right) \right)$$

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

$$\exists y \left(IsCat(y) \wedge \forall x \left(\neg IsDog(x) \vee \neg IsAfraidOf(y,x) \vee \neg IsAfraidOf(x,y) \right) \right)$$

Απαλοιφή του υπαρξιακού ποσοδείκτη

$$IsCat(\mathrm{C}) \wedge \forall x \left(\neg IsDog(x) \vee \neg IsAfraidOf(C,x) \vee \neg IsAfraidOf(x,C) \right)$$

Απαλοιφή του καθολικού ποσοδείκτη

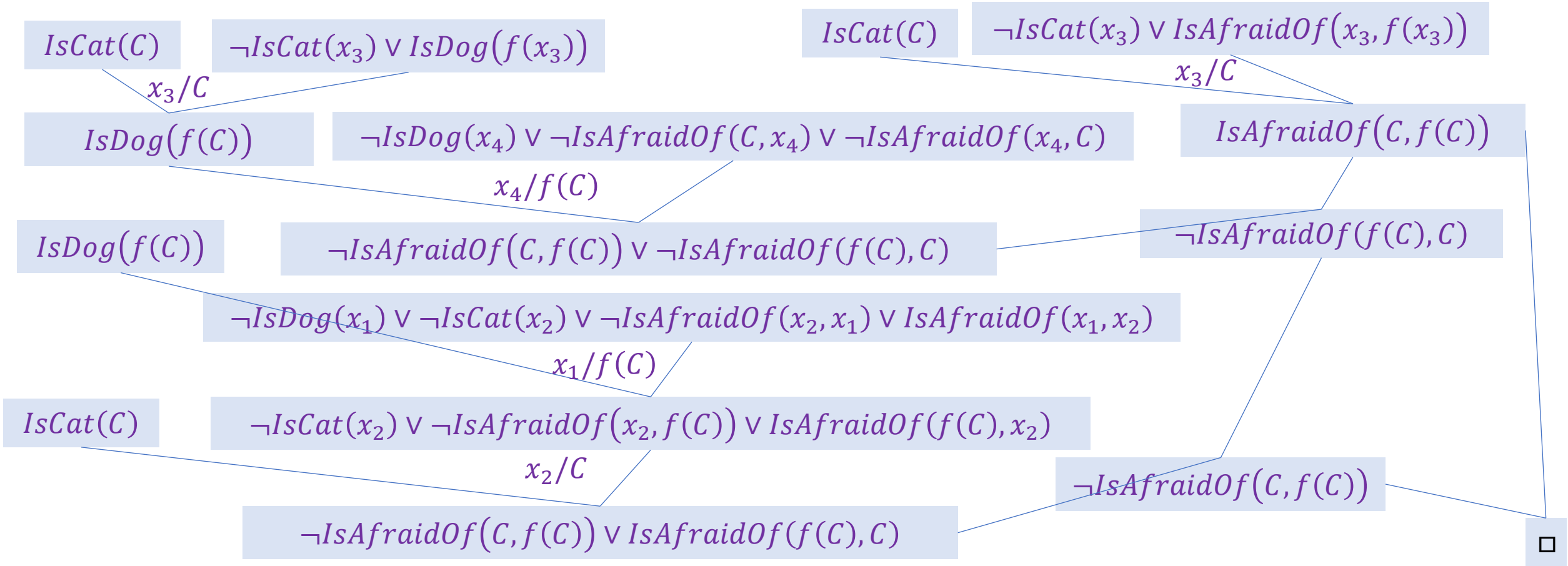
$$IsCat(C) \wedge \left(\neg IsDog(x) \vee \neg IsAfraidOf(C,x) \vee \neg IsAfraidOf(x,C) \right)$$

Επομένως προκύπτουν οι προτάσεις

$$IsCat(C)$$
$$\neg IsDog(x) \vee \neg IsAfraidOf(C,x) \vee \neg IsAfraidOf(x,C)$$

γ) Σχεδιάστε δέντρο απόδειξης που να αποδεικνύει με απαγωγή σε άτοπο χρησιμοποιώντας μόνο τον κανόνα της ανάλυσης (resolution) ότι η πρόταση (v) του σκέλους (α) έπεται λογικά από τις προτάσεις (iii) και (iv).

Οι προτάσεις Horn είναι και προτάσεις σε μορφή *CNF, οπότε μπορούμε να χρησιμοποιήσουμε απόδειξη με τον κανόνα της ανάλυσης. Ξεκινάμε με την άρνηση της v) και τις iii) και iv).*



Άσκηση 11.4

Επιλέξτε τις σωστές απαντήσεις, μόνο μία σε κάθε ερώτηση

α) Η προτασιακή λογική είναι:

i) Αποκρίσιμη
ii) ημι-αποκρίσιμη
iii) μη αποκρίσιμη.

Απάντηση

Η προτασιακή λογική είναι αποκρίσιμη αφού κάθε τύπος περιέχει πεπερασμένο πλήθος μεταβλητών και επομένως το πλήθος των μοντέλων που πρέπει να εξεταστούν είναι πεπερασμένο.

β) Η πρωτοβάθμια κατηγορηματική λογική είναι:

i) Αποκρίσιμη
ii) ημι-αποκρίσιμη
iii) μη αποκρίσιμη.

Απάντηση

Η πρωτοβάθμια κατηγορηματική λογική είναι ημι-αποκρίσιμη αφού

- Αν ΒΓ $\vDash \alpha$ τότε παράγει □ μετά από πεπερασμένο πλήθος βημάτων.
- Διαφορετικά επειδή ο χώρος αναζήτησης μπορεί να είναι άπειρος ενδέχεται να μην τερματίζει.

γ) Κάθε τύπος προτασιακής λογικής μπορεί να μετατραπεί σε κανονική συζευκτική μορφή:

Απάντηση:
Συμφωνώ και μάλιστα ο νέος τύπος είναι ταυτολογικά ισοδύναμος με τον αρχικό

δ) Κάθε τύπος πρωτοβάθμιας κατηγορηματικής λογικής μπορεί να μετατραπεί σε κανονική συζευκτική μορφή:

Απάντηση:
Συμφωνώ, αλλά ο νέος τύπος δεν είναι σίγουρα ταυτολογικά ισοδύναμος με τον αρχικό

Άσκηση 12.1

Δείξτε χρησιμοποιώντας τον αλγόριθμο εξαγωγής συμπερασμάτων προς τα εμπρός *fol-fc-ask* ότι ο τύπος $Path\left(A, NextOf\left(NextOf(A)\right)\right)$ προκύπτει ως συμπέρασμα από την ακόλουθη βάση γνώσεων προτάσεων Horn. Τα *x,y,z,u,v,w* είναι μεταβλητές, ενώ το *A* είναι σταθερά.

$$Link\left(x, NextOf(x)\right)$$
$$Link(y, z) \Rightarrow Path(y, z)$$
$$\left(Path(u, v) \wedge Link(v, w)\right) \Rightarrow Path(u, w)$$

Βρίσκουμε τύπο - κανόνα $\tau \in$ ΒΓ με μορφή $\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \Rightarrow \beta$. Αλλάζω τις μεταβλητές του κανόνα.

$$Link(y_1, z_1) \Rightarrow Path(y_1, z_1)$$

Βρίσκουμε γεγονότα της ΒΓ που ενοποιούνται με υπόθεση του κανόνα. Αλλάζω τις μεταβλητές των γεγονότων.

$$Link\left(x_2, NextOf(x_2)\right)$$

Βρίσκουμε τον ενοποιητή θ.

$$\theta = \{y_1/x_2, z_1/NextOf(x_2)\}$$

Εισάγουμε τα νέα συμπεράσματα στη βάση.

$$\text{ΒΓ} = \text{ΒΓ} \cup \left\{Path\left(x_2, NextOf(x_2)\right)\right\}$$

$$Link\big(x, NextOf(x)\big)$$
$$Link(y,z) \Rightarrow Path(y,z)$$
$$\big(Path(u,v) \wedge Link(v,w)\big) \Rightarrow Path(u,w)$$
$$Path\big(x_2, NextOf(x_2)\big)$$

Επαναλαμβάνουμε:
Από τον κανόνα $\big(Path(u_3,v_3) \wedge Link(v_3,w_3)\big) \Rightarrow Path(u_3,w_3)$ και

$$Path\big(x_5, NextOf(x_5)\big)$$
$$Link\big(x_4, NextOf(x_4)\big)$$

έχουμε

$$\theta = \{u_3/x_5, v_3/NextOf(x_5), x_4/v_3, w_3/NextOf(v_3)\}$$

Επομένως προσθέτουμε το μοναδικό νέο συμπέρασμα του κανόνα στη ΒΓ.

$$\text{ΒΓ} = \text{ΒΓ} \cup \Big\{Path\Big(x_5, NextOf\big(NextOf(x_5)\big)\Big)\Big\}$$

Το συμπέρασμα ενοποιείται με τον στόχο με $\theta = \{x_5/A\}$ οπότε ο αλγόριθμος τερματίζει επιστρέφοντας το θ που σημαίνει ότι ο στόχος προκύπτει ως συμπέρασμα από την αρχική ΒΓ.

Άσκηση 12.2

Γράψτε ένα πρόγραμμα Prolog το οποίο να αντιστρέφει λίστες, όπως στο παρακάτω παράδειγμα (με πλάγια γράμματα φαίνονται οι αποκρίσεις της Prolog, με έντονα ό,τι γράφει ο χρήστης). Μπορείτε να χρησιμοποιήσετε το κατηγόρημα append των διαφανειών (προσοχή: τα πρώτα δύο ορίσματα του append είναι λίστες). Χρησιμοποιήστε π.χ. την SWI-Prolog (βλ. ιστοσελίδα «σύνδεσμοι» του μαθήματος), για να βεβαιωθείτε ότι το πρόγραμμά σας δουλεύει σωστά.

$?- \boldsymbol{myReverse}([\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}],\boldsymbol{R}).$
$\boldsymbol{R} = [\boldsymbol{d},\boldsymbol{c},\boldsymbol{b},\boldsymbol{a}]$;
no
$?- \boldsymbol{myReverse}([\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}],[\boldsymbol{d},\boldsymbol{c},\boldsymbol{b},\boldsymbol{a}]).$
Yes

$myReverse([\ ],[\ ]).$
$myReverse([FirstIn\ |RestIn],ListOut) :- myReverse(RestIn,RestOut),append(RestOut,[FirstIn],ListOut).$

Ο πρώτος κανόνας δηλώνει ότι το αντίστροφο της κενής λίστας είναι ο εαυτός της. Χρησιμεύει για την περίπτωση που δώσουμε ως είσοδο την κενή λίστα αλλά και ως τερματική συνθήκη για την αναδρομή του δεύτερου κανόνα.

Στον δεύτερο κανόνα, [FirstIn | RestIn] είναι η λίστα προς αντιστροφή. Αν RestOut είναι η αντεστραμμένη RestIn, τότε προσθέτοντας το FirstIn στο τέλος της RestOut, δηλαδή συνενώνοντας τις λίστες RestOut και [FirstIn], προκύπτει η ζητούμενη λίστα ListOut.

Άσκηση 12.3

α) Παραστήστε τις προτάσεις (i), (ii), (iii) και (iv) σε κατάλληλη μορφή πρωτοβάθμιας κατηγορηματικής λογικής, ώστε να είναι δυνατόν να αποδειχθεί με τους αλγορίθμους fol-fc-ask και fol-bc-ask ότι τα (i), (ii) και (iii) συνεπάγονται το (iv).
Υπόδειξη: Χρησιμοποιήστε στην παράσταση του (ii) ένα κατηγόρημα της μορφής OrCut(x, y, z) και στην παράσταση του (iii) ένα κατηγόρημα της μορφής NotCut(x,y). Προσθέστε κανόνες που να συνδέουν τα OrCut(x,y,z), Cut(x,y) και NotCut(x,y).

(i) Η Μαρία συμπαθεί τον Γιάννη.

$$Likes(Mary, John)$$

(ii) Τον Νίκο τον έκοψε ο Γιάννης ή ο Γιώργος.

$$OrCut(John, George, Nick)$$

(iii) Αν κάποιος συμπαθεί τον x, τότε ο x δεν έκοψε κανέναν.

$$Likes(y, x) \Rightarrow NotCut(x, z)$$

(iv) Τον Νίκο τον έκοψε ο Γιώργος.

$$Cut(George, Nick)$$

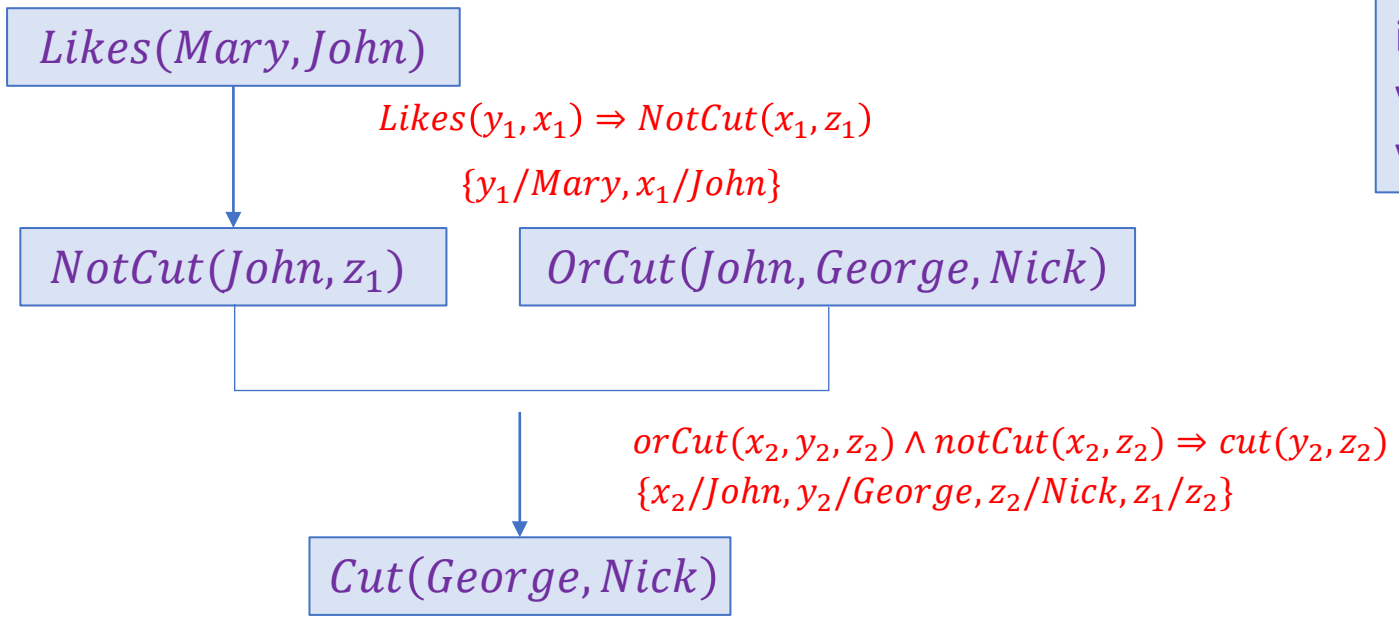
Προσθέτουμε τους κανόνες:

(v) $orCut(x, y, z) \wedge notCut(y, z) \Rightarrow cut(x, z)$

(vi) $orCut(x, y, z) \wedge notCut(x, z) \Rightarrow cut(y, z)$

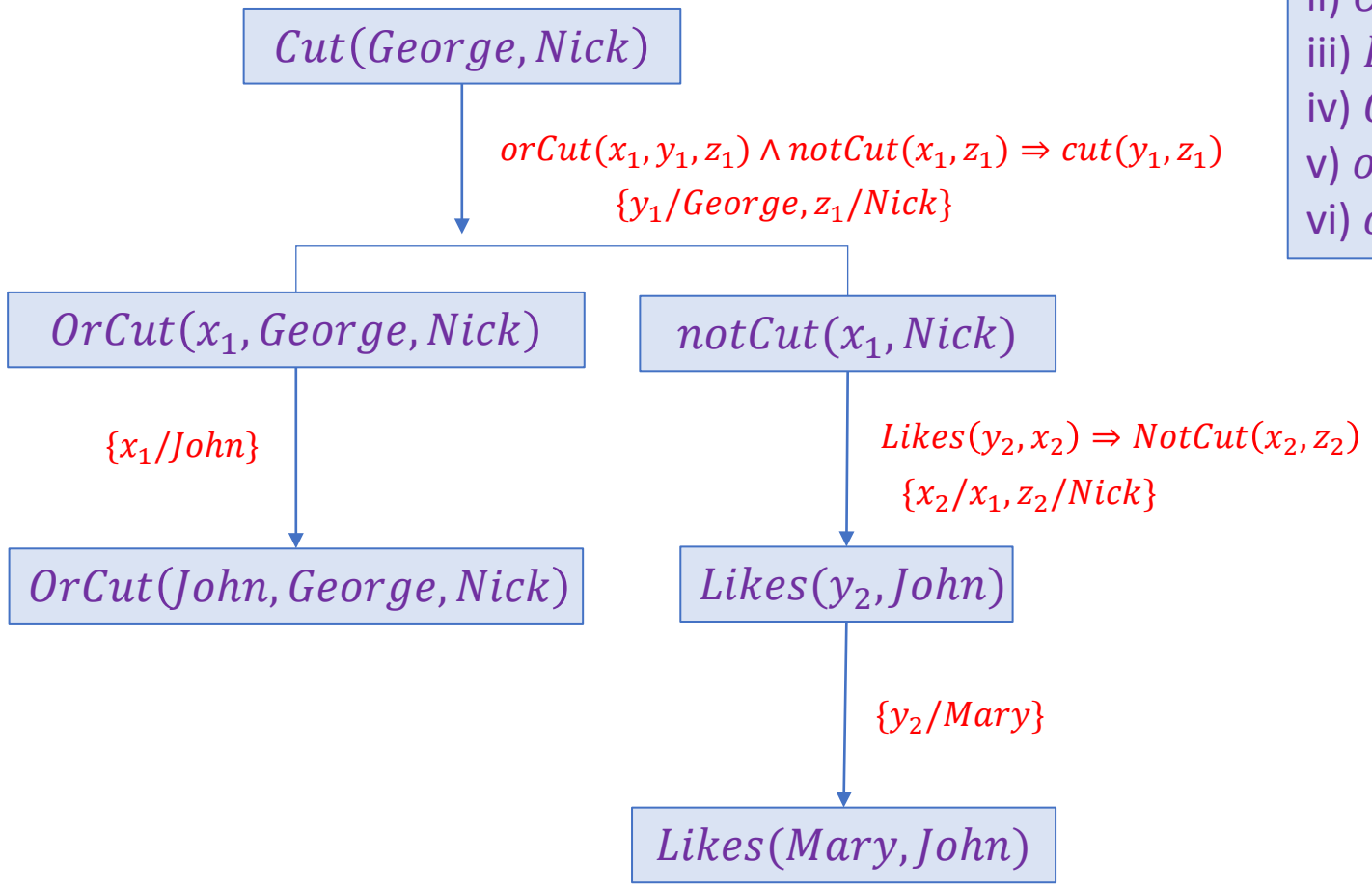
β) Δείξτε το δέντρο απόδειξης του αλγόριθμου FOL-FC-ASK για την πρόταση iv).

i) $Likes(Mary, John)$
ii) $OrCut(John, George, Nick)$
iii) $Likes(y, x) \Rightarrow NotCut(x, z)$
iv) $Cut(George, Nick)$
v) $orCut(x, y, z) \wedge notCut(y, z) \Rightarrow cut(x, z)$
vi) $orCut(x, y, z) \wedge notCut(x, z) \Rightarrow cut(y, z)$



γ) Δείξτε το δέντρο απόδειξης του αλγόριθμου FOL-BC-ASK: για την πρόταση iv).

i) $Likes(Mary, John)$
ii) $OrCut(John, George, Nick)$
iii) $Likes(y, x) \Rightarrow NotCut(x, z)$
iv) $Cut(George, Nick)$
v) $orCut(x, y, z) \wedge notCut(y, z) \Rightarrow cut(x, z)$
vi) $orCut(x, y, z) \wedge notCut(x, z) \Rightarrow cut(y, z)$



δ) Γράψτε ως πρόγραμμα Prolog τους τύπους του σκέλους (α) για τα (i), (ii), (iii), (iv) και τους επιπλέον κανόνες για τις σχέσεις μεταξύ των OrCut(x,y,z), Cut(x,y) και NotCut(x,y). Χρησιμοποιήστε το πρόγραμμά σας και την SWI-Prolog για να αποδείξετε ότι τον Νίκο τον έκοψε ο Γιώργος, όπως φαίνεται παρακάτω:

?- cut(george, nick).
yes
?- cut(X, nick).

X = george ;
No

Απάντηση:

```
likes(mary, john).
orCut(john, George, nick).
notCut(X, _) :- likes(_, X).
cut(X, Z) :- orCut(X, Y, Z), notCut(Y, Z).
cut(Y, Z) :- orCut(X, Y, Z), notCut(X, Z).
```

i) $Likes(Mary, John)$
ii) $OrCut(John, George, Nick)$
iii) $Likes(y, x) \Rightarrow NotCut(x, z)$
iv) $Cut(George, Nick)$
v) $orCut(x, y, z) \wedge notCut(y, z) \Rightarrow cut(x, z)$
vi) $orCut(x, y, z) \wedge notCut(x, z) \Rightarrow cut(y, z)$

Άσκηση 12.4

Συμπληρώστε τα υπογραμμισμένα κενά στο παρακάτω πρόγραμμα Prolog, ώστε να συνδυάζει δύο λίστες ως εξής:

?- combine([a, b, c], [d, e, f], L).
L = [a, d, b, e, c, f]

?- combine([a, b, c], [d, e], L).
L = [a, d, b, e, c]

?- combine([a, b], [d, e, f, g], L).
L = [a, d, b, e, f, g]

?- combine([a, b], [], L).
L = [a, b]

?- combine([], [d, e, f], L).
L = [d, e, f].

?- combine([], [], []).
Yes

?- combine([a, b, c], [d, e], [a, d, b, e, c]).
Yes

```
combine([], B, B).
combine(A, [], A).
combine([F1|R1], [F2|R2], [F1, F2|R]):- combine(R1, R2, R).
```