

# Τεχνητή Νοημοσύνη

*6ο φροντιστήριο (2023-24)*

Επιμέλεια: Σοφία Ελευθερίου,  
Φοίβος Χαραλαμπάκος

**13.1.** Παραστήστε σε OWL με συντακτικό συναρτησιακού ύφους (functional-style syntax) τις σημασίες των προτάσεων (i) – (v). Ακολουθούν πρόσθετες πληροφορίες και υποδείξεις:

Η τάξη των αντικειμένων που έχουν το `:psita` ως τιμή της ιδιότητας (property) `:likes` παριστάνεται ως `ObjectHasValue( :likes :psita )`

Η ένωση δύο τάξεων `C1` και `C2` παριστάνεται ως `ObjectUnionOf( C1 C2 )`.

Η τομή δύο τάξεων `C1` και `C2` παριστάνεται ως `ObjectIntersectionOf( C1 C2 )`.

Η τάξη των αντικειμένων που έχουν ως τιμή της ιδιότητας `:likes` τουλάχιστον ένα αντικείμενο της τάξης `:Cat` παριστάνεται ως `ObjectMinCardinality( 1 :likes :Cat )`.

(i) Ο Μίλος είναι σκύλος και η Ψίτα είναι γάτα.

`ClassAssertion( :Dog :milos )`

`ClassAssertion( :Cat :psita )`

(ii) Κάθε σκύλος γαβγίζει ή κουνάει την ουρά του (ενδεχομένως και τα δύο μαζί).

`SubClassOf( :Dog ObjectUnionOf( :Barks :MovesTail ))`

(iii) Κάθε σκύλος που συμπαθεί την Ψίτα γαβγίζει.

`SubClassOf( ObjectIntersectionOf( :Dog ObjectHasValue( :likes :psita )) :Barks )`

(iv) Ο Μίλος συμπαθεί την Ψίτα.

`ObjectPropertyAssertion( :likes :milos :psita )`

(v) Κάθε σκύλος που συμπαθεί μια (τουλάχιστον) γάτα γαβγίζει.

`SubClassOf( ObjectIntersectionOf( :Dog ObjectMinCardinality( 1 :likes Cat )) :Barks )`

Cheat sheet:

- Κάθε και αν/τότε υποδηλώνουν `SubClassOf`
- Η τάξη έχει ένα αντικείμενο ενώ η ιδιότητα έχει 2 τιμές (αυτόν που ενεργεί και αυτόν που παθαίνει)

**13.2.** Παραστήστε σε OWL (με functional-style syntax) τις προτάσεις (i)–(vii). Επιτρέπεται να χρησιμοποιήσετε μόνο τα ακόλουθα ονόματα τάξεων:

Human, Woman.

Επίσης, επιτρέπεται να χρησιμοποιήσετε μόνο τα ακόλουθα ονόματα ιδιοτήτων (properties):

HasGrandSon, HasChild, HasBrother, HasMother.

(i) Η Μαρία είναι γυναίκα.

ClassAssertion( :Woman :Maria )

(ii) Η Μαρία έχει το Νίκο εγγόνι.

ObjectPropertyAssertion( :hasGrandChild :Maria :Nikos )

(iii) Η Μαρία έχει τουλάχιστον ένα παιδί.

ClassAssertion( ObjectMinCardinality( 1 :hasChild ) :Maria )

(iv) Ο Γιάννης έχει τουλάχιστον έναν αδελφό.

ClassAssertion( ObjectMinCardinality( 1 :hasBrother ) :John )

(v) Ο Γιάννης έχει ακριβώς έναν αδελφό.

ClassAssertion( ObjectExactCardinality( 1 :hasBrother ) :John )

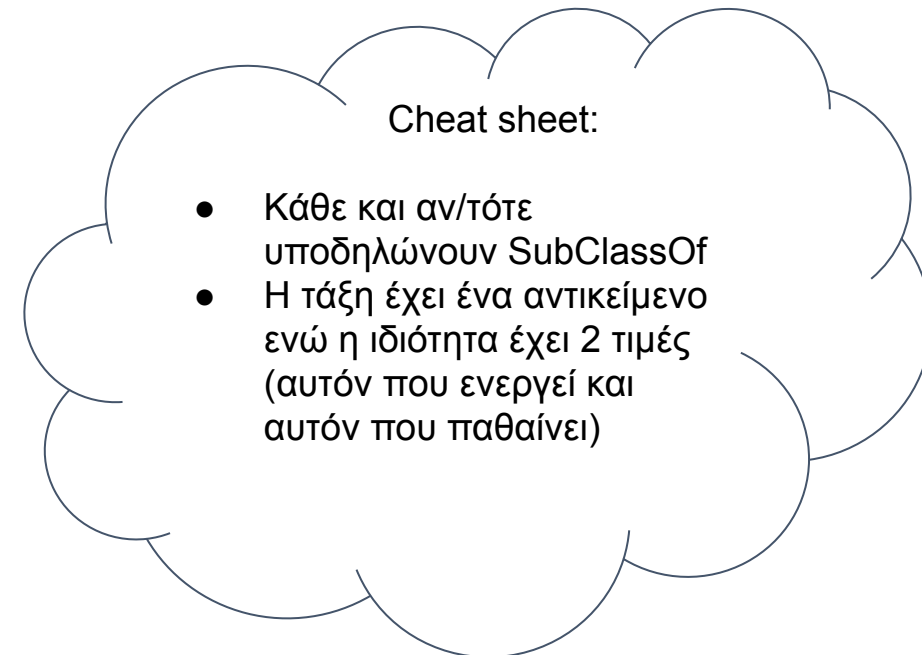
(vi) Κάθε άνθρωπος έχει ακριβώς μία μητέρα.

SubClassOf( :Human ObjectExactCardinality( 1 :hasMother ) )

(vii) Αν μία γυναίκα έχει τουλάχιστον ένα εγγόνι, τότε έχει και τουλάχιστον ένα παιδί.

SubClassOf( ObjectIntersectionOf( :Woman ObjectMinCardinality( 1 :hasGrandChild ) )

ObjectMinCardinality( 1 :hasChild ) )



**13.3.** Ορίστε σε OWL μια οντολογία που να περιλαμβάνει τις ακόλουθες τάξεις, με τις σημασίες που υπονοούν τα ονόματά τους:

Human, Man, Woman.

και τις ακόλουθες σχέσεις, με τις σημασίες που υπονοούν τα ονόματά τους:

isHusbandOf, isWifeOf,

isFatherOf, isMotherOf, isSonOf, isDaughterOf, isParentOf, isChildOf,

isGrandFatherOf, isGrandMotherOf, isGrandSonOf, isGrandDaughterOf.

Η οντολογία σας πρέπει να περιλαμβάνει τις λιγότερες δυνατές δηλώσεις, αλλά να καλύπτει όλους τους εύλογους περιορισμούς, οι οποίοι να αποκλείουν π.χ. να δηλωθεί ως πατέρας κάποιου μία γυναίκα ή να έχει κάποιος περισσότερες από μία (βιολογικές) μητέρες κλπ.

Η οντολογία σας πρέπει, επίσης, να υποστηρίζει την αυτόματη εξαγωγή εύλογων συμπερασμάτων, όπως π.χ. ότι αν ο/η X είναι γονέας του/της Y, τότε ο/η Y είναι παιδί του/της X.

## **/\* Class Definitions \*/**

```
SubClassOf(:Man :Human) //Man ISA Human
SubClassOf(:Woman :Human) //Woman ISA Human
DisjointClasses(:Man :Woman) //Man != Woman
EquivalentClasses(:Human ObjectUnionOf(:Man :Woman))
//Man + Woman = Human
```

## **/\* Property Definitions: Husband – Wife \*/**

```
ObjectPropertyDomain(:isHusbandOf :Man) //isHusbandOf's domain = Man
ObjectPropertyRange(:isHusbandOf :Woman) //isHusbandOf's range = Woman
FunctionalObjectProperty(:isHusbandOf) //a Man can be the husband of only one Woman
FunctionalObjectProperty(:isWifeOf) //a Woman can be the wife of only one Man
InverseObjectProperties(:isHusbandOf :isWifeOf) //if x is the husband of y, y is the wife of x
```

Τάξεις: Human, Man, Woman  
Σχέσεις: isHusbandOf, isWifeOf,  
isFatherOf, isMotherOf, isSonOf,  
isDaughterOf, isParentOf, isChildOf,  
isGrandFatherOf, isGrandMotherOf,  
isGrandSonOf, isGrandDaughterOf.

## **/\* Property Definitions: Parent – Child \*/**

ObjectPropertyDomain(:isParentOf :Human) //isParentOf's domain = Human

ObjectPropertyDomain(:isChildOf :Human) //isChildOf's domain = Human

IrreflexiveObjectProperty(:isParentOf) // x can't be the parent of x

AsymmetricObjectProperty(:isParentOf)  
// if x is a parent of y, y can't be a parent of x

InverseObjectProperties(:isParentOf :isChildOf)  
//if x is a parent of y, then y is a child of x

SubObjectPropertyOf(ObjectPropertyChain(:isChildOf :isHusbandOf) :isChildOf)  
//if x is a child of y and y is the husband of z, then x is a child of z

SubObjectPropertyOf(ObjectPropertyChain(:isChildOf :isWifeOf) :isChildOf)  
//if x is a child of y and y is the wife of z, then x is a child of z

SubClassOf(:Human :ObjectExactCardinality(1 :isChildOf :Man))  
//a Human is the child of exactly one Man

SubClassOf(:Human :ObjectExactCardinality(1 :isChildOf :Woman))  
//a Human is the child of exactly one Woman

Τάξεις: Human, Man, Woman  
Σχέσεις: isHusbandOf, isWifeOf,  
isFatherOf, isMotherOf, isSonOf,  
isDaughterOf, isParentOf, isChildOf,  
isGrandFatherOf, isGrandMotherOf,  
isGrandSonOf, isGrandDaughterOf.

### **/\* Property Definitions: Father – Mother \*/**

```
SubObjectPropertyOf(:isFatherOf :isParentOf) //if x is the father of y, then x is a
parent of y
SubObjectPropertyOf(:isMotherOf :isParentOf) //if x is the mother of y, x is a parent
of y
ObjectPropertyDomain(:isFatherOf :Man) //isFatherOf's domain = Man
ObjectPropertyDomain(:isMotherOf :Woman)
//isMotherOf's domain = Woman
```

### **/\* Property Definitions: Son – Daughter \*/**

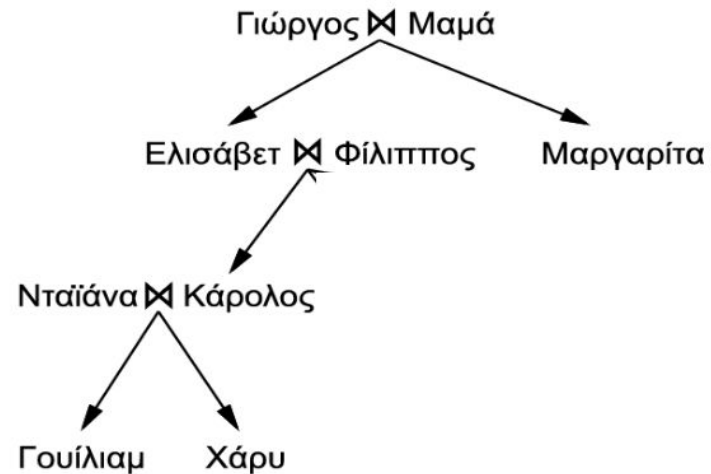
```
SubObjectPropertyOf(:isSonOf :isChildOf) //if x is a son of y, then x is a child of y
SubObjectPropertyOf(:isDaughterOf :isChildOf) //if x is a daughter of y, then x is a child of y
ObjectPropertyDomain(:isSonOf :Man) //isSonOf's domain = Man
ObjectPropertyDomain(:isDaughterOf :Woman) //isDaughterOf's domain = Woman
```

### **/\* Property Definitions: GrandFather – GrandMother – GrandSon – GrandDaughter \*/**

```
SubObjectPropertyOf(ObjectPropertyChain(:isFatherOf :isParentOf) :isGrandFatherOf)
//if x is the father of y and y is a parent of z then x is a grand-father of z
SubObjectPropertyOf (ObjectPropertyChain(:isMotherOf :isParentOf) :isGrandMotherOf)
//if x is the mother of y and y is a parent of z then x is a grand-mother of z
SubObjectPropertyOf (ObjectPropertyChain(:isSonOf :isChildOf) :isGrandSonOf)
//if x is a son of y and y is a child of z then x is grand-son of z
SubObjectPropertyOf (ObjectPropertyChain(:isDaughterOf :isChildOf) :isGrandDaughterOf)
//if x is a daughter of y and y is a child of z then x is grand-daughter of z
```

Τάξεις: Human, Man, Woman  
Σχέσεις: isHusbandOf, isWifeOf,  
isFatherOf, isMotherOf, isSonOf,  
isDaughterOf, isParentOf, isChildOf,  
isGrandFatherOf, isGrandMotherOf,  
isGrandSonOf, isGrandDaughterOf.

**13.4.** Χρησιμοποιήστε την οντολογία της προηγούμενης άσκησης για να περιγράψετε πλήρως σε OWL τις οικογενειακές πληροφορίες του παρακάτω σχήματος με όσο το δυνατόν λιγότερες δηλώσεις. Δεν χρειάζεται να γράψετε δηλώσεις `differentIndividuals`.



`ObjectPropertyAssertion(:isHusbandOf :Γιώργος :Μαμά)`  
`ObjectPropertyAssertion(:isDaughterOf :Ελισάβετ :Γιώργος)`  
`ObjectPropertyAssertion(:isDaughterOf :Μαργαρίτα :Γιώργος)`  
`ObjectPropertyAssertion(:isHusbandOf :Φίλιππος :Ελισάβετ)`  
`ObjectPropertyAssertion(:isSonOf :Κάρολος :Φίλιππος)`  
`ObjectPropertyAssertion(:isHusbandOf :Κάρολος :Νταϊάνα)`  
`ObjectPropertyAssertion(:isSonOf :Γουίλιαμ :Κάρολος)`  
`ObjectPropertyAssertion(:isSonOf :Χάρυ :Κάρολος)`

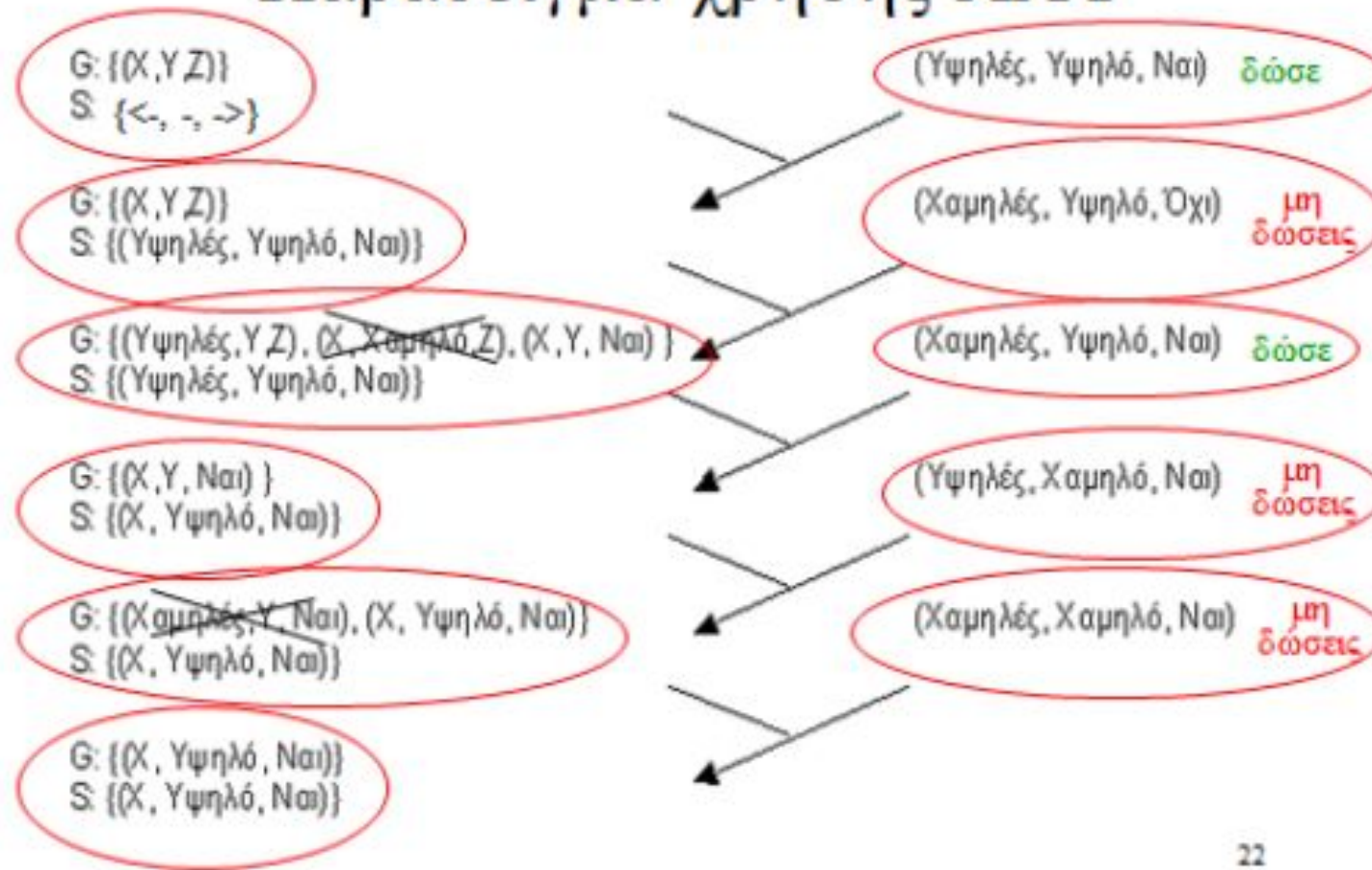


# Αλγόριθμος απαλοιφής υποψηφίων

- Για κάθε νέο **θετικό** παράδειγμα  $d$ :
  - **Αφαίρεσε από το  $G$**  τις υποθέσεις που είναι **ασυνεπείς** με το  $d$ .
  - Για κάθε υπόθεση  $s$  του  $S$  που είναι **ασυνεπής** με το  $d$ :
    - **Αφαίρεσε την  $s$  από το  $S$ .**
    - **Πρόσθεσε στο  $S$  κάθε ελάχιστη γενίκευση  $h$  της  $s$ , για την οποία ισχύει ότι:**
      - $H h$  είναι **συνεπής** με το  $d$ .
      - $H h$  είναι **ειδικότερη ή ίδια με μια υπόθεση του  $G$ .**  
(Ξέρουμε ότι η υπόθεση που ψάχνουμε είτε υπάρχει ήδη στο  $G$  είτε είναι ειδικότερη μιας υπόθεσης του  $G$ .)
  - **Αφαίρεσε από το  $S$  κάθε υπόθεση που είναι γενικότερη από μια άλλη υπόθεση του  $S$ .**

- Για κάθε νέο **αρνητικό** παράδειγμα  $d$ :
  - **Αφαίρεσε από το  $S$**  τις υποθέσεις που είναι **ασυνεπείς** με το  $d$ .
  - Για κάθε υπόθεση  $g$  του  $G$  που είναι **ασυνεπής** με το  $d$ :
    - **Αφαίρεσε την  $g$  από το  $G$ .**
    - **Πρόσθεσε στο  $G$  κάθε ελάχιστη ειδικέυση  $h$  της  $g$ , για την οποία ισχύει ότι:**
      - $H h$  είναι **συνεπής** με το  $d$ .
      - $H h$  είναι **γενικότερη ή ίδια με μια υπόθεση του  $S$ .**  
(Ξέρουμε ότι η υπόθεση που ψάχνουμε είτε υπάρχει ήδη στο  $S$  είτε είναι γενικότερη μιας υπόθεσης του  $S$ .)
  - **Αφαίρεσε από το  $G$  κάθε υπόθεση που είναι ειδικότερη από μια άλλη υπόθεση του  $G$ .**

## Παράδειγμα χρήσης ΑΑΥ



## Άσκηση 14.1.

Στο παράδειγμα χρήσης του Αλγορίθμου Απαλοιφής Υποψηφίων της 14ης διάλεξης, θεωρήστε ότι υπάρχει ένα ακόμα, έκτο παράδειγμα εκπαίδευσης:

(Υψηλές, Υψηλό, Ναι): **μη δώσεις**.

Εξηγήστε αναλυτικά πώς θα μεταβληθούν τα G και S κατά την επεξεργασία του έκτου παραδείγματος εκπαίδευσης. Τι πρόβλημα προκύπτει; Ήταν αναμενόμενο και γιατί;



- Για κάθε νέο **αρνητικό** παράδειγμα d:
  - **Αφαίρεσε από το S** τις υποθέσεις που είναι ασυνεπείς με το d.
  - Για κάθε υπόθεση g του G που είναι ασυνεπής με το d:
    - **Αφαίρεσε** την g από το G.
    - **Πρόσθεσε στο G** κάθε **ελάχιστη ειδίκευση** h της g, για την οποία ισχύει ότι:
      - Η h είναι **συνεπής** με το d.
      - Η h είναι **γενικότερη ή ίδια με μια υπόθεση του S**.(Ξέρουμε ότι η υπόθεση που ψάχνουμε είτε υπάρχει ήδη στο S είτε είναι γενικότερη μιας υπόθεσης του S.)
  - **Αφαίρεσε από το G** κάθε υπόθεση που είναι **ειδικότερη από μια άλλη υπόθεση του G**.

## Άσκηση 14.1.

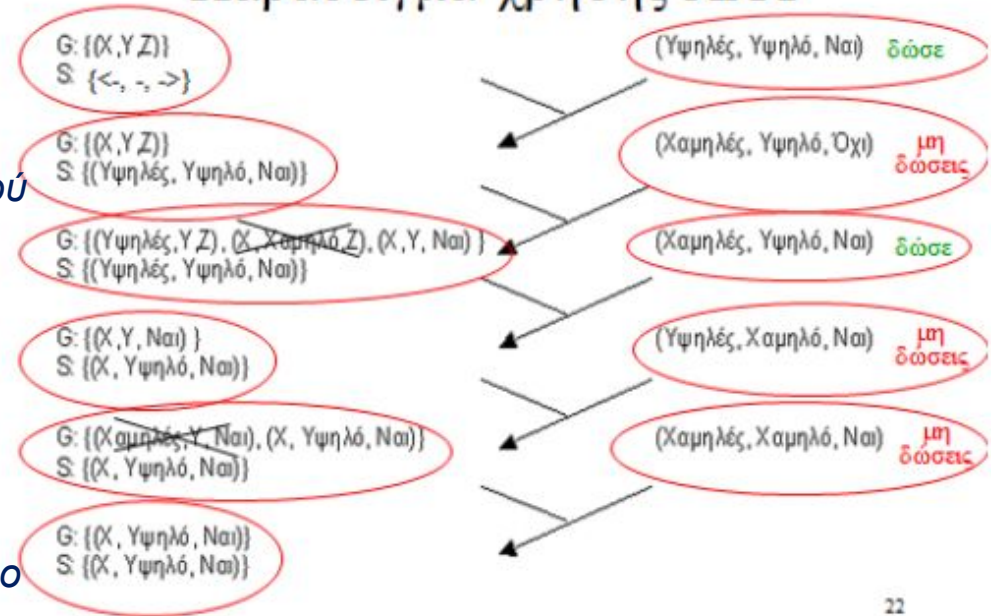
Απάντηση:

Αφαιρούμε πρώτα από το  $S$  τις υποθέσεις που είναι ασυνεπείς με το νέο παράδειγμα. Η υπόθεση  $(X, \text{Υψηλό}, \text{ΝΑΙ})$  είναι ασυνεπής με το νέο παράδειγμα, αφού δίνει δάνειο στον πελάτη του έκτου παραδείγματος, ενώ δεν θα έπρεπε. Άρα την αφαιρούμε και έχουμε πλέον  $S = \{\}$ .

Αφαιρούμε κατόπιν από το  $G$  τις υποθέσεις που είναι ασυνεπείς με το νέο παράδειγμα. Η μοναδική υπόθεση του  $G$  είναι ασυνεπής με το νέο παράδειγμα. Επομένως, έχουμε πλέον  $G = \{\}$ . Θα πρέπει επίσης να προσθέσουμε στο  $G$  κάθε ελάχιστη ειδίκευση της υπόθεσης που αφαιρέσαμε η οποία (i) είναι συνεπής με το νέο παράδειγμα και (ii) είναι γενικότερη ή ίδια με μια υπόθεση του  $S$ , αλλά αφού το  $S$  είναι πλέον κενό, δεν υπάρχει καμία υπόθεση που να ικανοποιεί τη συνθήκη (ii). Άρα παραμένει  $G = \{\}$ .

Καταλήγουμε, επομένως, σε  $S = G = \{\}$ , δηλαδή δεν επιλέγουμε κατά τη μάθηση καμία υπόθεση του χώρου αναζήτησης. Αυτό είναι αναμενόμενο, γιατί το έκτο παράδειγμα είναι ασυνεπές με το πρώτο και γνωρίζουμε ότι με ασυνεπή παραδείγματα εκπαίδευσης, ο AAY καταλήγει σε  $S = G = \{\}$ .

## Παράδειγμα χρήσης AAY



$(Υψηλός, Υψηλό, Ναι)$ : μη δώσεις.



## Άσκηση 14.2.

Εξηγήστε αναλυτικότερα πώς ο αλγόριθμος FOIL (βλ. διαφάνειες 31–33) θα μπορούσε να μάθει τις τέσσερις προτάσεις Horn για το GrandParent (GP) της διαφάνειας 32 από το παρακάτω υποσύνολο δεδομένων της διαφάνειας 31.

Απάντηση:

Ξεκινάμε από την αρχική πρόταση Horn:

$$\mathbf{GP(x, y) \Leftarrow \text{True}}$$

η οποία, όμως, είναι ασυνεπής π.χ. με το  $\text{not GP}(\text{Φίλιππος}, \text{Φίλιππος})$

Ειδικεύουμε π.χ. προσθέτοντας τη συνθήκη  $\text{Πατέρας}(x, z)$ , οπότε λαμβάνουμε τη νέα πρόταση Horn:

$$\mathbf{GP(x, y) \Leftarrow \text{Πατέρας}(x, z)}$$

που είναι όμως πάλι ασυνεπής π.χ. με το  $\text{not GP}(\text{Φίλιππος}, \text{Κάρολος})$

Ειδικεύουμε περαιτέρω π.χ. δοκιμάζοντας την πρόταση:

$$\mathbf{GP(x, y) \Leftarrow \text{Πατέρας}(x, z) \text{ and } \text{Πατέρας}(z, r)}$$

που είναι πάλι ασυνεπής π.χ. με το  $\text{not GP}(\text{Φίλιππος}, \text{Κάρολος})$ , γιατί για  $\{x/\text{Φίλιππος}, y/\text{Κάρολος}, z/y, r/\text{Γουίλιαμ}\}$  παράγει το συμπέρασμα  $\text{GP}(\text{Φίλιππος}, \text{Κάρολος})$ .

*Περιγραφές παραδειγμάτων εκπαίδευσης:*

Πατέρας(Φίλιππος, Κάρολος)  
Πατέρας(Φίλιππος, Άννα)  
Μητέρα(Ελισάβετ, Κάρολος)  
Μητέρα(Ελισάβετ, Άννα)  
Πατέρας(Κάρολος, Γουίλιαμ)  
Πατέρας(Κάρολος, Χάρυ)  
Μητέρα(Νταϊάνα, Γουίλιαμ)  
Μητέρα(Νταϊάνα, Χάρυ)  
Πατέρας(Μάρκος, Πέτρος)  
Πατέρας(Μάρκος, Ζάρα)  
Μητέρα(Άννα, Πέτρος)  
Μητέρα(Άννα, Ζάρα)

Θετικά παραδείγματα εκπαίδευσης:

1.  $\text{GP}(\text{Φίλιππος}, \text{Γουίλιαμ})$
2.  $\text{GP}(\text{Φίλιππος}, \text{Χάρυ})$
3.  $\text{GP}(\text{Φίλιππος}, \text{Πέτρος})$
4.  $\text{GP}(\text{Φίλιππος}, \text{Ζάρα})$
5.  $\text{GP}(\text{Ελισάβετ}, \text{Γουίλιαμ})$
6.  $\text{GP}(\text{Ελισάβετ}, \text{Χάρυ})$
7.  $\text{GP}(\text{Ελισάβετ}, \text{Πέτρος})$
8.  $\text{GP}(\text{Ελισάβετ}, \text{Ζάρα})$

*Αρνητικά παραδείγματα εκπαίδευσης:*

Όλοι οι υπόλοιποι συνδυασμοί ζευγών σταθερών, π.χ:  
 $\text{not GP}(\text{Φίλιππος}, \text{Φίλιππος})$   
 $\text{not GP}(\text{Φίλιππος}, \text{Κάρολος})$

...

## Άσκηση 14.2.

Εξηγήστε αναλυτικότερα πώς ο αλγόριθμος FOIL (βλ. διαφάνειες 31–33) θα μπορούσε να μάθει τις τέσσερις προτάσεις Horn για το GrandParent (GP) της διαφάνειας 31 από το παρακάτω υποσύνολο δεδομένων της διαφάνειας 30.

Δοκιμάζουμε την ελαφρά διαφορετική ειδίκευση:

**GP(x, y)  $\Leftarrow$  Πατέρας(x, z) and Πατέρας(z, y)**

η οποία είναι συνεπής με όλα τα αρνητικά παραδείγματα και καλύπτει τα θετικά παραδείγματα 1, 2.

Έχουμε βρει ήδη μια πρόταση που είναι συνεπής με όλα τα αρνητικά παραδείγματα και καλύπτει τα θετικά παραδείγματα 1 και 2.

Με κατάλληλη ευρετική να μας καθοδηγεί βρίσκουμε με τον ίδιο τρόπο την πρόταση

**GP(x, y)  $\Leftarrow$  Πατέρας(x, z) and Μητέρα(z, y)**

που είναι συνεπής με όλα τα αρνητικά παραδείγματα και καλύπτει τα θετικά 3, 4.

Με όμοιο τρόπο βρίσκουμε και τις προτάσεις

**GP(x, y)  $\Leftarrow$  Μητέρα(x, z) and Μητέρα(z, y)**

**GP(x, y)  $\Leftarrow$  Μητέρα(x, z) and Πατέρας(z, y)**

*Περιγραφές παραδειγμάτων εκπαίδευσης:*

Πατέρας(Φίλιππος, Κάρολος)  
Πατέρας(Φίλιππος, Άννα)  
Μητέρα(Ελισάβετ, Κάρολος)  
Μητέρα(Ελισάβετ, Άννα)  
Πατέρας(Κάρολος, Γουίλιαμ)  
Πατέρας(Κάρολος, Χάρυ)  
Μητέρα(Νταϊάνα, Γουίλιαμ)  
Μητέρα(Νταϊάνα, Χάρυ)  
Πατέρας(Μάρκος, Πέτρος)  
Πατέρας(Μάρκος, Ζάρα)  
Μητέρα(Άννα, Πέτρος)  
Μητέρα(Άννα, Ζάρα)

*Θετικά παραδείγματα εκπαίδευσης:*

1. GP(Φίλιππος, Γουίλιαμ)
2. GP(Φίλιππος, Χάρυ)
3. GP(Φίλιππος, Πέτρος)
4. GP(Φίλιππος, Ζάρα)
5. GP(Ελισάβετ, Γουίλιαμ)
6. GP(Ελισάβετ, Χάρυ)
7. GP(Ελισάβετ, Πέτρος)
8. GP(Ελισάβετ, Ζάρα)

*Αρνητικά παραδείγματα εκπαίδευσης:*

Όλοι οι υπόλοιποι συνδυασμοί ζευγών σταθερών, π.χ:  
not GP(Φίλιππος, Φίλιππος)  
not GP(Φίλιππος, Κάρολος)

...