

# Τεχνητή Νοημοσύνη

*12η διάλεξη (2024-25)*

Ίων Ανδρουτσόπουλος

<http://www.aueb.gr/users/ion/>

Οι διαφάνειες αυτής της διάλεξης βασίζονται στα βιβλία *Τεχνητή Νοημοσύνη των Βλαχάβα κ.ά.*, 3η έκδοση, Β. Γκιούρδας Εκδοτική, 2006 και *Artificial Intelligence – A Modern Approach* των S. Russel και P. Norvig, 2η και 4<sup>η</sup> έκδοση, Prentice Hall, 2003 και 2020. Τα σχήματα των διαφανειών προέρχονται από αντίστοιχες διαφάνειες των δύο βιβλίων.

# Τι θα ακούσετε σήμερα

- **Εξαγωγή συμπερασμάτων** προς τα εμπρός και προς τα πίσω με προτάσεις **Horn** πρωτοβάθμιας κατηγορηματικής λογικής.
- **Λογικός προγραμματισμός** και βασικές αρχές λειτουργίας **Prolog**.

# Προτάσεις Horn στην ΠΚΛ

- Όπως και στην ΠΛ, διαζεύξεις σαν της μορφής CNF αλλά που περιέχουν το πολύ ένα «θετικό» ατομικό τύπο.
- Εδώ ασχολούμαστε μόνο με «οριστικές» προτάσεις, δηλ. προτάσεις Horn με ακριβώς ένα «θετικό» ατομικό τύπο («definite clauses»):

$$((\text{King}(x) \wedge \text{Greedy}(x)) \Rightarrow \text{Evil}(x)) \equiv \quad (\text{παράδειγμα «κανόνα»})$$
$$(\neg \text{King}(x) \vee \neg \text{Greedy}(x) \vee \text{Evil}(x))$$

$\text{Greedy}(y), \text{King}(\text{John})$  (παραδείγματα «γεγονότων», «facts»)

- Θεωρούμε ότι για όλες τις μεταβλητές υπονοούνται **καθολικοί ποσοδείκτες** στην αρχή των τύπων.
  - Αντί για υπαρξιακούς ποσοδείκτες, **συναρτήσεις Skolem.**
- Ευκολότερη εξαγωγή συμπερασμάτων από ό,τι αν χρησιμοποιούμε ολόκληρη την ΠΚΛ.
  - Εξαγωγή συμπερασμάτων προς τα **εμπρός ή πίσω.**

# ΒΓ σε μορφή προτάσεων Horn

- Είναι έγκλημα να πουλήσει Αμερικανός όπλα σε εχθρική χώρα:  
 $((\text{American}(x) \wedge \text{Weapon}(y) \wedge \text{Sells}(x, y, z) \wedge \text{Hostile}(z)) \Rightarrow \text{Criminal}(x))$
- Η Νόνο έχει τουλάχιστον έναν πύραυλο.  
Ας χρησιμοποιήσουμε τη σταθερά M1 για να αναφερθούμε σε αυτόν.  
 $\text{Missile}(M1)$  και  $\text{Owns}(\text{Nono}, M1)$
- Όλους τους πυραύλους της η Νόνο τους αγόρασε από τον West.  
 $((\text{Missile}(x) \wedge \text{Owns}(\text{Nono}, x)) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono}))$
- Οι πύραυλοι είναι όπλα.  
 $(\text{Missile}(x) \Rightarrow \text{Weapon}(x))$
- Οι χώρες που είναι εχθροί της Αμερικής είναι εχθρικές.  
Το δικαστήριο είναι αμερικανικό...  
 $(\text{Enemy}(x, \text{America}) \Rightarrow \text{Hostile}(x))$
- Ο West είναι Αμερικανός.  
 $\text{American}(\text{West})$
- Η Nono είναι εχθρός της Αμερικής.  
 $\text{Enemy}(\text{Nono}, \text{America})$

# Εξαγωγή συμπεράσματος προς τα εμπρός

*Criminal(West)*

Αυτό που θέλουμε να αποδείξουμε.



προς τα εμπρός

*American(West)*

*Missile(M1)*

*Owns(Nono,M1)*

*Enemy(Nono,America)*

Τα γεγονότα που έχουμε στη ΒΓ.

# Εξαγωγή συμπεράσματος προς τα εμπρός

*Criminal(West)*



προς τα  
εμπρός

*American(West)*

*Missile(M1)*

*Sells(West, M1, Nono)*

*Owns(Nono, M1)*

*Enemy(Nono, America)*

$((\text{Missile}(x1) \wedge \text{Owns}(\text{Nono}, x1)) \Rightarrow \text{Sells}(\text{West}, x1, \text{Nono}))$  με  $\{x1/M1\}$

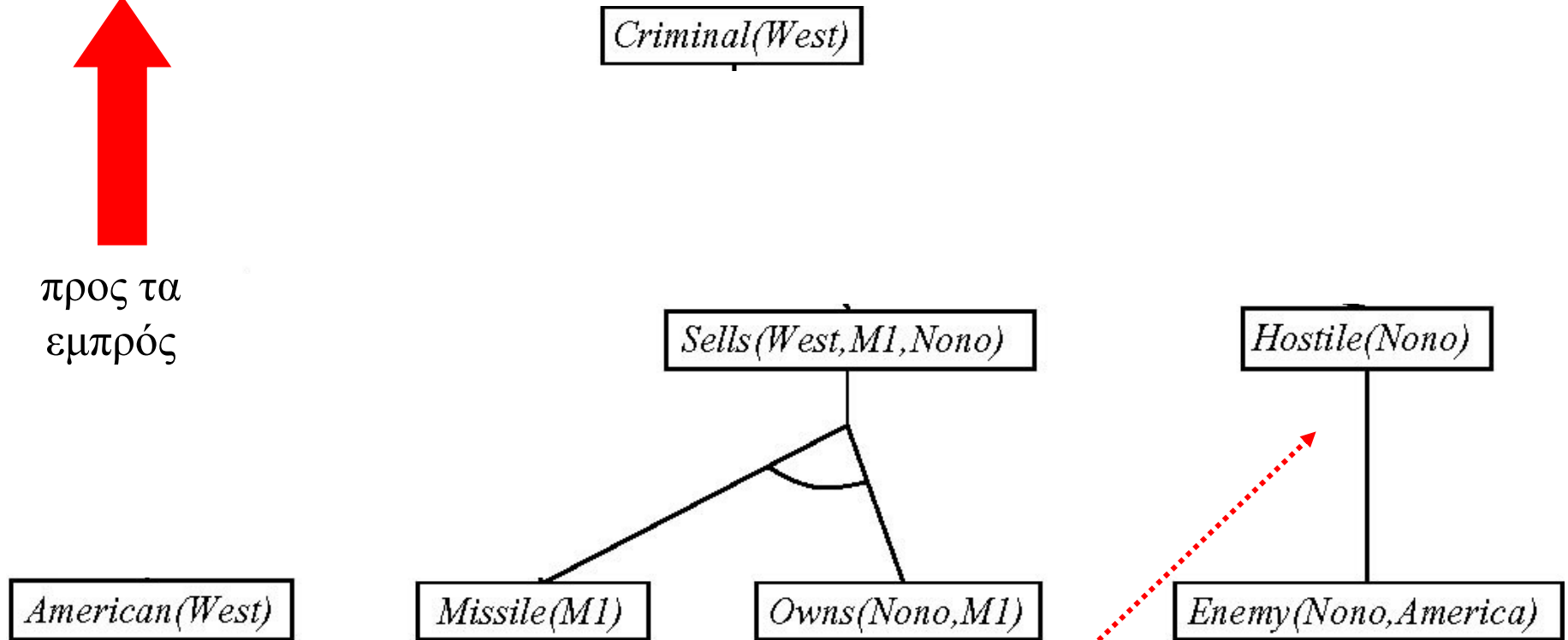
Κανόνας της ΒΓ, του οποίου οι υποθέσεις ταιριάζουν με γεγονότα που γνωρίζουμε.

Όποτε χρησιμοποιούμε έναν κανόνα, του αλλάζουμε τις μεταβλητές με νέες.

# Εξαγωγή συμπεράσματος προς τα εμπρός



προς τα  
εμπρός



Κανόνας της ΒΓ, του οποίου  
οι υποθέσεις ταιριάζουν με  
γεγονότα που γνωρίζουμε.

$(\text{Enemy}(x2, \text{America}) \Rightarrow \text{Hostile}(x2))$   
με  $\{x2/\text{Nono}\}$

Όποτε χρησιμοποιούμε έναν κανόνα, του αλλάζουμε τις μεταβλητές με νέες.

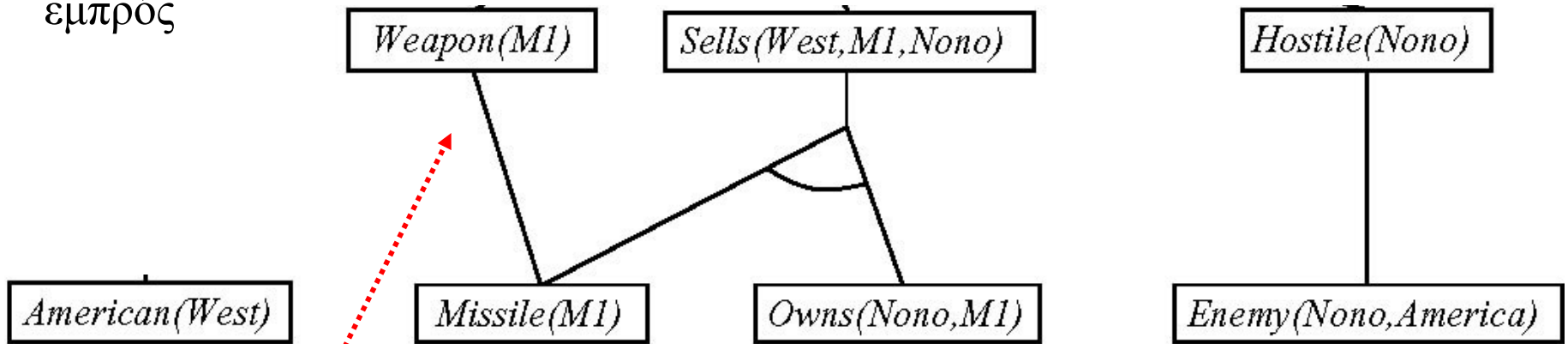


# Εξαγωγή συμπεράσματος προς τα εμπρός



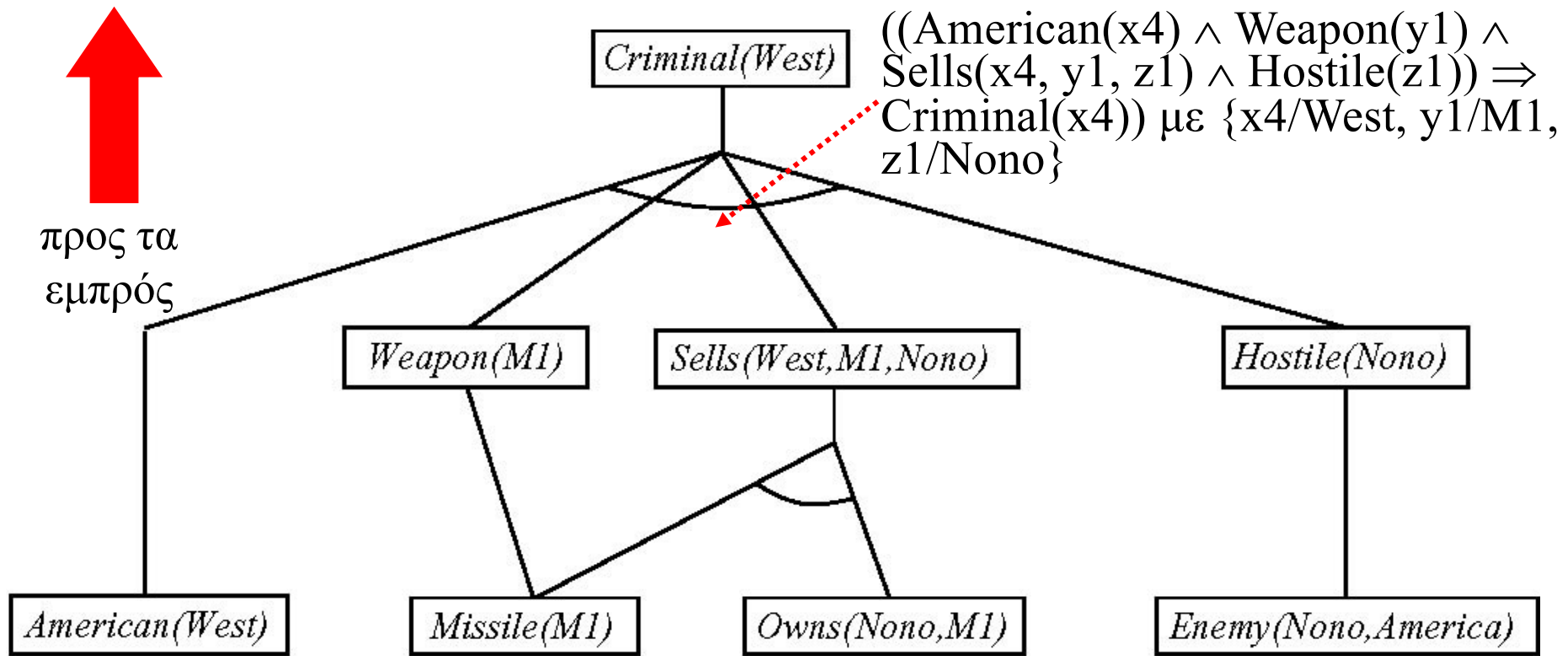
*Criminal(West)*

προς τα  
εμπρός



$(\text{Missile}(x3) \Rightarrow \text{Weapon}(x3)), \text{ με } \{x3/M1\}$

# Εξαγωγή συμπεράσματος προς τα εμπρός



# Εξαγωγή συμπεράσματος προς τα εμπρός

συνάρτηση  $\text{fol-fc-ask}(B\Gamma, \alpha)$  επιστρέφει ενοποιητή ή αποτυχία

είσοδοι:  $B\Gamma$ : η βάση γνώσης, σύνολο από οριστικές προτάσεις ΠΚΛ

$\alpha$ : η ερώτηση, ατομικός τύπος ΠΚΛ

βρόχος

$\text{νέο} \leftarrow \{\}$

για κάθε τύπο  $\tau \in B\Gamma$  με μορφή

$((\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n) \Rightarrow \beta) \leftarrow \text{new-vars}(\tau)$

για κάθε  $\theta$  με  $\text{unify}(\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n, \alpha_1' \wedge \alpha_2' \wedge \dots \wedge \alpha_n') = \theta \neq \text{αποτυχία}$

όπου  $\alpha_1', \alpha_2', \dots, \alpha_n' \in B\Gamma$

$\beta' \leftarrow \text{subst}(\theta, \beta)$

αν το  $\beta'$  δεν είναι «αντίγραφο» τύπου της  $B\Gamma$  ή του  $\text{νέο}$  τότε

$\text{νέο} \leftarrow \text{νέο} \cup \{\beta'\}$

$\theta' \leftarrow \text{unify}(\beta', \alpha)$

αν το  $\theta'$  δεν είναι αποτυχία τότε επιστρέψε  $\theta'$

$B\Gamma \leftarrow B\Gamma \cup \text{νέο}$

μέχρι το  $\text{νέο}$  να είναι κενό

επίστρεψε αποτυχία

Κάθε φορά νέες μεταβλητές. Π.χ.  $(\text{Missile}(x_1) \Rightarrow \text{Weapon}(x_1)), (\text{Missile}(x_2) \Rightarrow \text{Weapon}(x_2)), \dots$

Αλλάζουμε και στα  $\alpha_i'$  μεταβλητές.

Π.χ. το  $\text{Likes}(x, \text{Mary})$  είναι «αντίγραφο» του  $\text{Likes}(y, \text{Mary})$ : σε όλους αρέσει η Μαρία.

Αν ενοποιείται με το στόχο, τελειώσαμε.

# Ερωτήσεις προς τη ΒΓ

- Μπορούμε να ρωτήσουμε **αν έπεται ταυτολογικά** από τη ΒΓ ένας ατομικός τύπος χωρίς μεταβλητές.
  - Π.χ. ερώτημα:  $\text{Criminal}(\text{West})$ . Επιστρέφει ενοποιητή (όχι αποτυχία), επομένως απαντά «ναι».
- Μπορούμε να ζητήσουμε ένα συνδυασμό **τιμών των μεταβλητών** ενός τύπου-ερωτήματος, για τους οποίους το ερώτημα έπεται ταυτολογικά από τη ΒΓ.
  - Π.χ. ερώτημα:  $\text{Criminal}(x)$ , απάντηση:  $\{x/\text{West}\}$ .
  - Στο παράδειγμα θα παραχθεί  $\text{Criminal}(\text{West})$  και θα ενοποιηθεί με το στόχο  $\text{Criminal}(x)$  επιστρέφοντας  $\{x/\text{West}\}$ .
- Μπορούμε να τροποποιήσουμε τον αλγόριθμο, ώστε να επιστρέφει **όλους** τους συνδυασμούς τιμών μεταβλητών για τους οποίους το ερώτημα έπεται ταυτολογικά από τη ΒΓ.
  - Π.χ. σε μια άλλη ΒΓ, ερώτημα:  $\text{Likes}(x, y)$ , απάντηση:  $\{x/\text{John}, y/\text{Mary}\}, \{x/\text{George}, y/\text{Anna}\}, \dots$

# Χαρακτηριστικά του fol-fc-ask

- **Ορθός:** Εφαρμόζουμε ουσιαστικά μόνο **Modus Ponens**.
  - Για την ακρίβεια, εφαρμόζουμε «γενικευμένο Modus Ponens», που περιλαμβάνει **ενοποίηση** των συνθηκών κανόνα με γεγονότα.
- **Πλήρης:** αν  $B\Gamma \models \alpha$ , τότε απαντά  $B\Gamma \vdash_i \alpha$  (για ατομικό  $\alpha$ ).
  - Η απόδειξη παραλείπεται.
  - Για όσους ανησυχούν: Αν έχουμε κανόνες όπως  $(Human(x) \Rightarrow Human(Father(x)))$ , δεν κινδυνεύουμε να παγιδευτούμε σε άπειρο κλαδί (π.χ.  $Human(John)$ ,  $Human(Father(John))$ , ...), κάτι που ενδεχομένως θα μας εμπόδιζε να φτάσουμε στο  $\alpha$ , γιατί πρώτα παράγουμε όλα τα δυνατά συμπεράσματα από την υπάρχουσα  $B\Gamma$  (τα βάζουμε στο «νέο») και μετά τα προσθέτουμε στη  $B\Gamma$  και βλέπουμε τι νέα συμπεράσματα προκύπτουν.

# Χαρακτηριστικά του fol-fc-ask

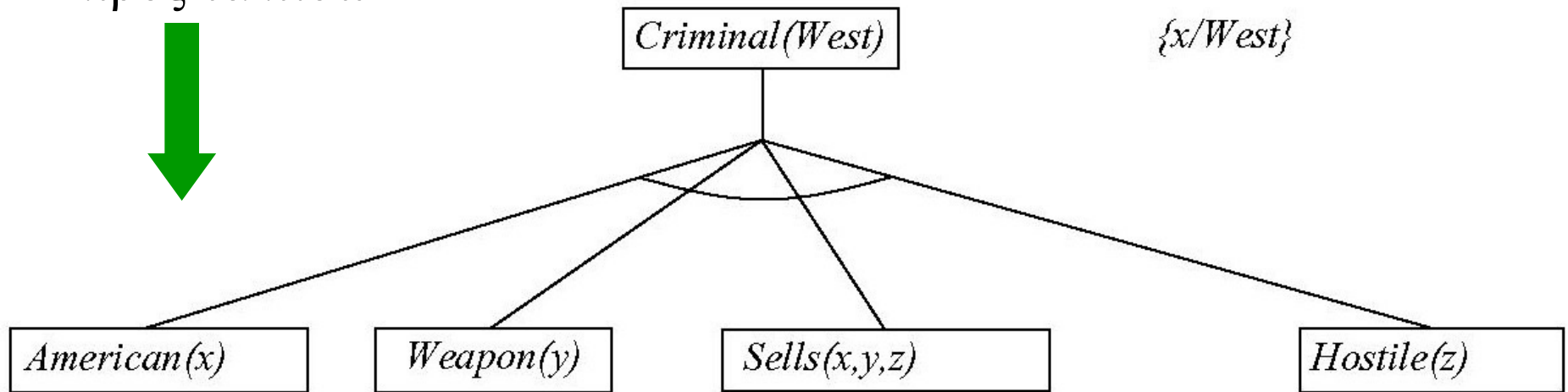
- **Δεν τερματίζει** πάντα, αν ΒΓ  $\not\equiv$  α.
  - Αν υπάρχουν **σύμβολα συναρτήσεων**, ίσως μπορούν να παραχθούν άπειρα συμπεράσματα, οπότε θα παράγει επ' άπειρον συμπεράσματα που δεν θα είναι, όμως, ποτέ το α.
  - Π.χ. έστω ότι το α είναι Human(Mary). Αν έχουμε στη ΒΓ μόνο (Human(x)  $\Rightarrow$  Human(Father(x))) και Human(John), μπορούμε να συμπεράνουμε: Human(**Father**(John)), Human(**Father**(**Father**(John))), ... χωρίς να παραγάγουμε ποτέ το α.
  - Όπως και με ολόκληρη την ΠΚΛ, η εξαγωγή συμπερασμάτων με προτάσεις Horn είναι **ημι-αποκρίσιμο** πρόβλημα.
  - Ενώ στην **Datalog**, όπου όλοι οι τύποι είναι οριστικές προτάσεις, αλλά (μεταξύ και άλλων περιορισμών) **δεν επιτρέπονται σύμβολα συναρτήσεων**, το πρόβλημα είναι **αποκρίσιμο**.

# Χαρακτηριστικά του fol-fc-ask

- Χρονική **πολυπλοκότητα**:
  - Ο αλγόριθμος **περιλαμβάνει** έλεγχο ταιριάσματος των συνθηκών ενός κανόνα με τα γεγονότα της ΒΓ.
  - Πρόβλημα **ταιριάσματος προτύπων**, που είναι **NP-hard**.
  - Αλλά στην πράξη ο αριθμός των συνθηκών των κανόνων και ο αριθμός των ορισμάτων των κατηγορημάτων είναι μικροί, οπότε **δεν μας απασχολεί** το πρόβλημα του ταιριάσματος προτύπων.
  - **Πολυωνυμική πολυπλοκότητα** ως προς τον αριθμό γεγονότων χωρίς μεταβλητές (ground facts) της ΒΓ.
- Παράγονται και **πολλά άσχετα συμπεράσματα**.
  - Μπορεί να αντιμετωπιστεί μερικώς ενσωματώνοντας **προβλέψεις** για το τι είναι σχετικό με το στόχο. Ουσιαστικά συνδυασμός με απόδειξη κατά την προς τα πίσω κατεύθυνση.

# Εξαγωγή συμπεράσματος προς τα πίσω

προς τα πίσω

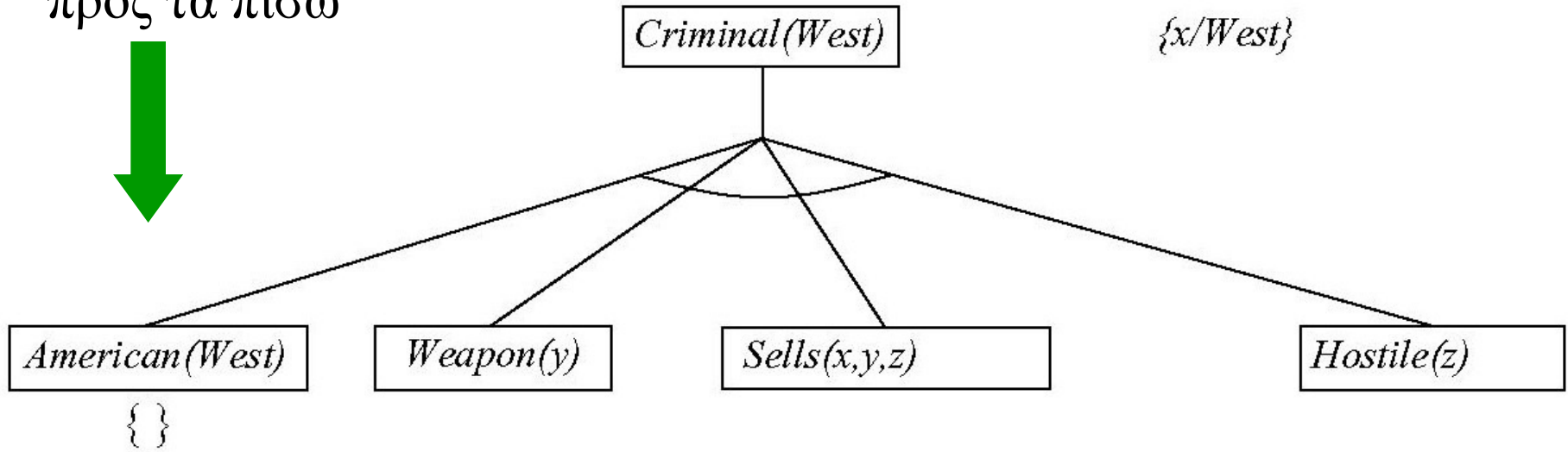


- Θέλουμε να αποδείξουμε ότι  $Criminal(West)$ .
- Με  $\{x/West\}$ , ο στόχος  $Criminal(West)$  ενοποιείται με την κεφαλή του  $((American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z)) \Rightarrow Criminal(x))$  και λαμβάνουμε ως συμπέρασμα το ζητούμενο
- Αρκεί να αποδείξουμε τα  $American(x)$ ,  $Weapon(y)$ ,  $Sells(x, y, z)$  και  $Hostile(z)$  με  $\{x/West\}$  για κάποιες τιμές των  $y, z$ .



# Εξαγωγή συμπεράσματος προς τα πίσω

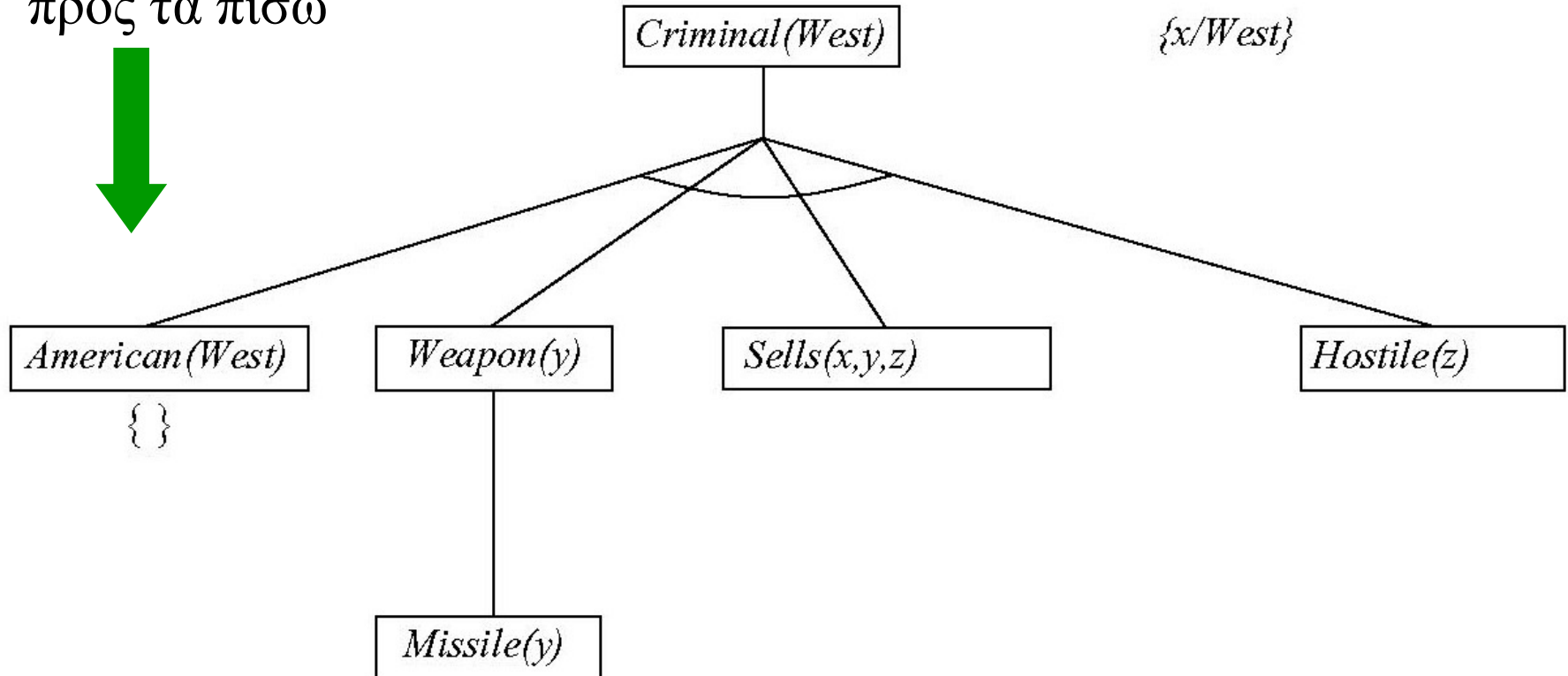
προς τα πίσω



- Με  $\{x/West\}$ , ο νέος στόχος  $American(x)$  είναι γεγονός της ΒΓ.
- Απομένει να αποδείξουμε τα  $Weapon(y)$ ,  $Sells(x, y, z)$  και  $Hostile(z)$  με  $\{x/West\}$  για κάποιες τιμές των  $y, z$ .

# Εξαγωγή συμπεράσματος προς τα πίσω

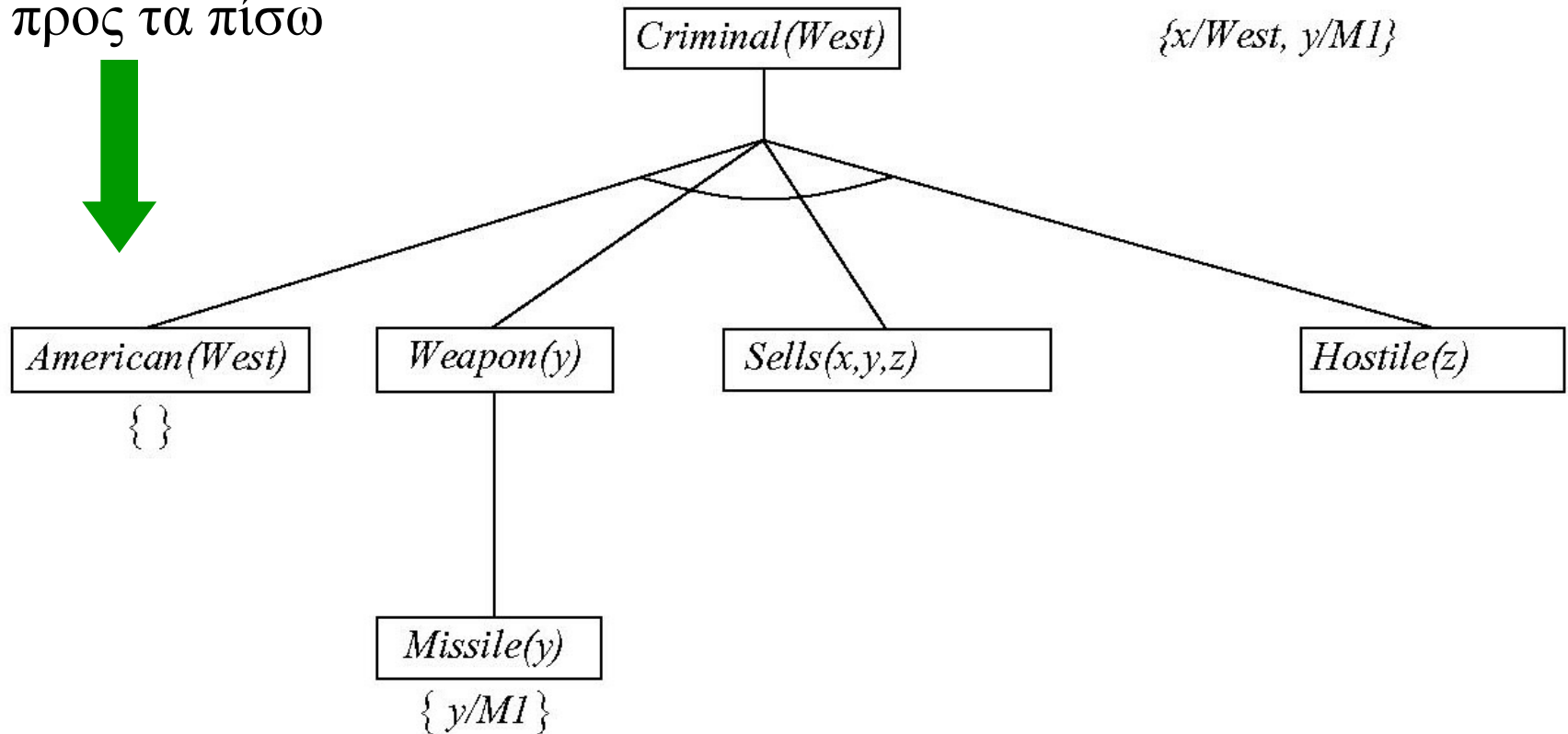
προς τα πίσω



- Το  $Weapon(y)$  ενοποιείται με την κεφαλή του ( $Missile(y) \Rightarrow Weapon(y)$ ).
  - Στην πραγματικότητα αλλάζουμε τις μεταβλητές των κανόνων κάθε φορά που τους χρησιμοποιούμε. Για απλούστευση εδώ το αγνοούμε αυτό.
- Άρα για να αποδείξουμε  $Weapon(y)$ , αρκεί να αποδείξουμε  $Missile(y)$  με  $\{x/West\}$  για κάποια τιμή του  $y$ .

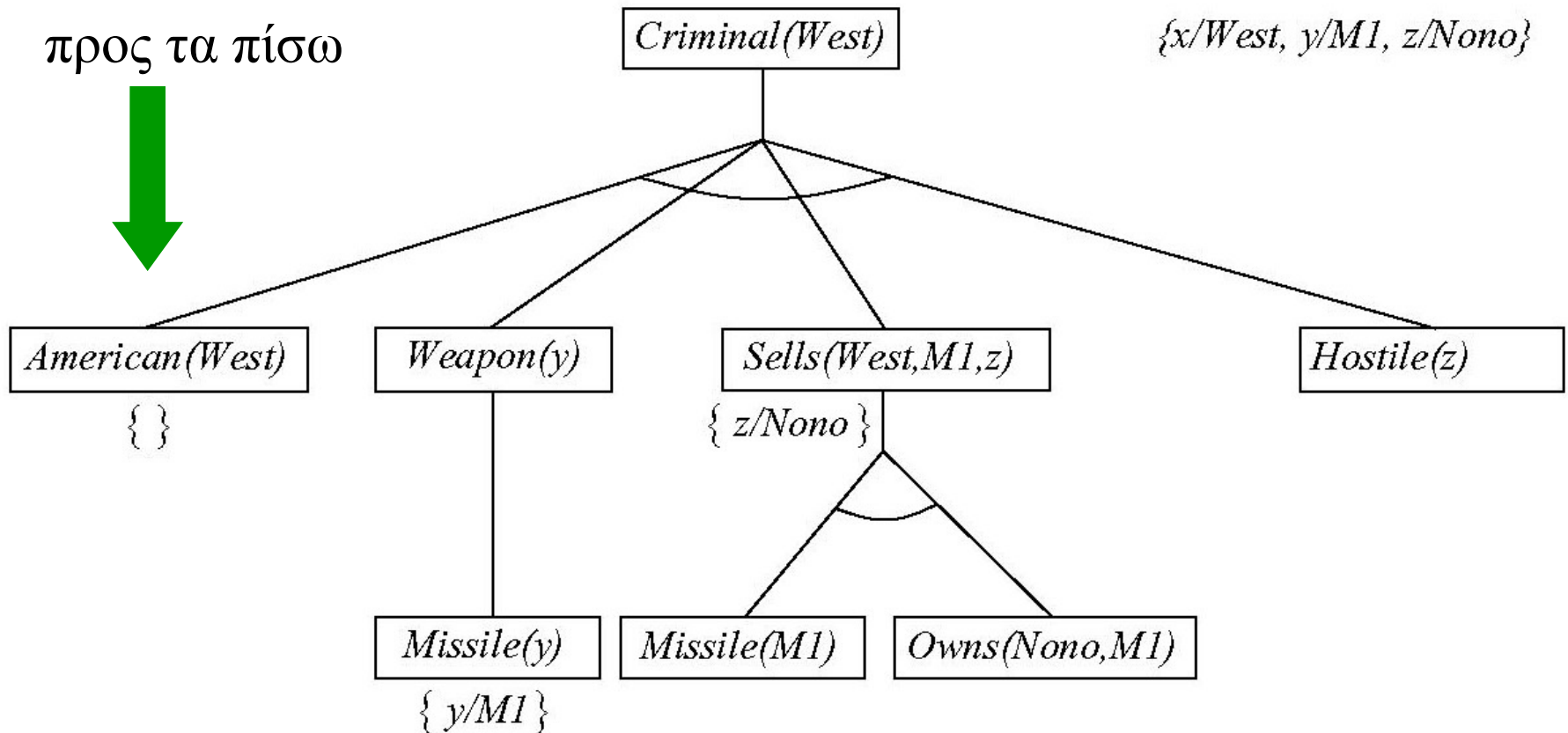
# Εξαγωγή συμπεράσματος προς τα πίσω

προς τα πίσω



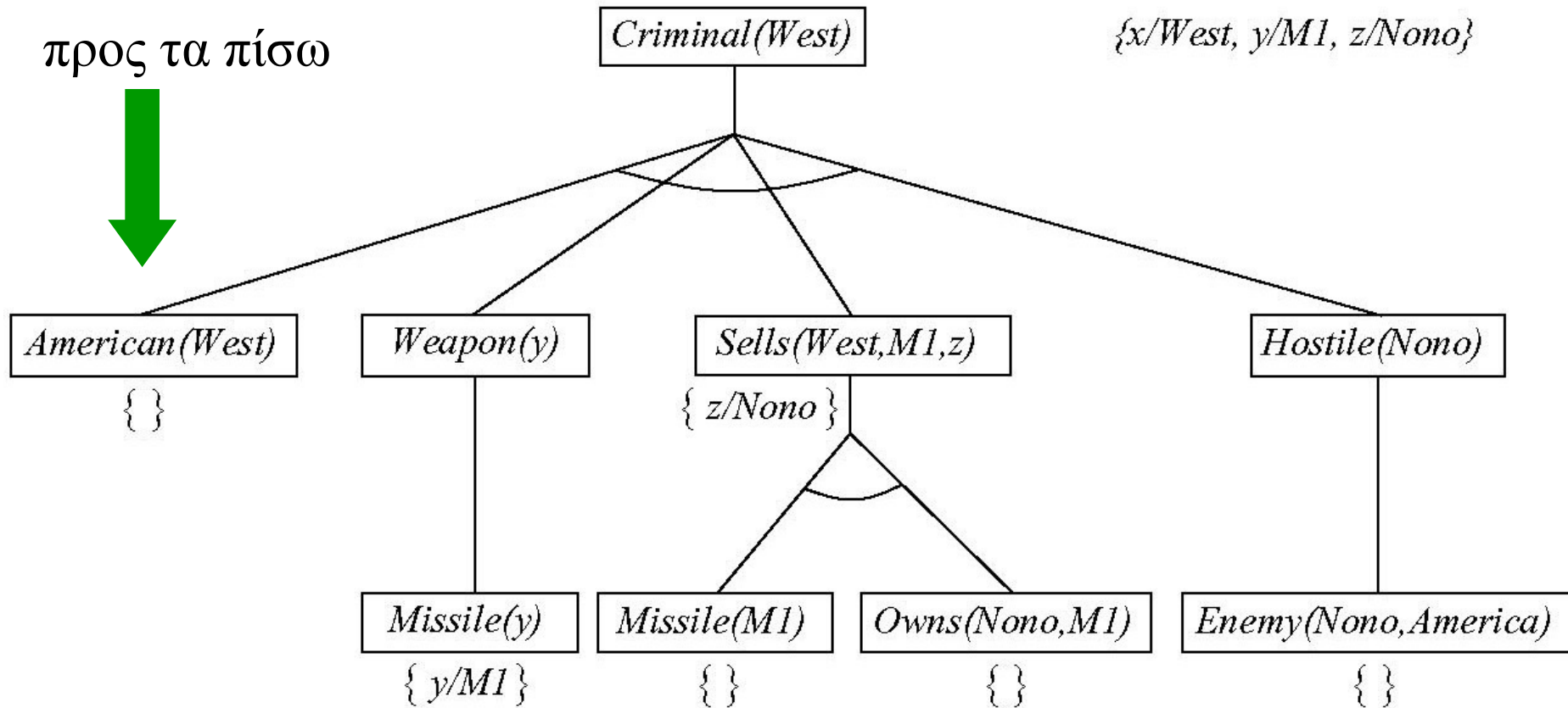
- Με  $\{x/West, y/M1\}$ , το  $Missile(y)$  είναι γεγονός της ΒΓ.
- Απομένει να αποδείξουμε τα  $Sells(x, y, z)$  και  $Hostile(z)$  με  $\{x/West, y/M1\}$  για κάποια τιμή του  $z$ .

# Εξαγωγή συμπεράσματος προς τα πίσω



- Με  $\{x/West, y/M1, z/Nono\}$ , το  $Sells(x, y, z)$  ενοποιείται με την κεφαλή του  $((Missile(y) \wedge Owns(Nono, y)) \Rightarrow Sells(West, y, Nono))$ .
- Αρκεί να αποδείξουμε τα  $Missile(y)$ ,  $Owns(Nono, y)$ ,  $Hostile(z)$  με  $\{x/West, y/M1, z/Nono\}$ .

# Εξαγωγή συμπεράσματος προς τα πίσω



- Με  $\{x/West, y/M1, z/Nono\}$ , τα  $Missile(y)$ ,  $Owns(Nono, y)$  είναι γεγονότα της ΒΓ.
- Ομοίως για το  $Hostile(Nono)$ .

# Εξαγωγή συμπεράσματος προς τα πίσω

**συνάρτηση**  $\text{fol-bc-ask}(B\Gamma, \text{στόχοι}, \theta)$  επιστρέφει σύνολο ενοποιητών

**είσοδοι:**  $B\Gamma$ : η βάση γνώσης σε μορφή οριστικών προτάσεων ΠΚΛ

$\text{στόχοι}$ : λίστα ατομικών τύπων (χωρίς ισότητες), παριστάνει σύζευξη

$\theta$ : ο ενοποιητής ως τώρα, αρχικά  $\{\}$

**τοπικές μεταβλητές:**  $\text{απαντήσεις}$ : αρχικά κενό σύνολο ενοποιητών (αποτυχία)

**αν**  $\text{στόχοι} = []$  **τότε επιστρέψε**  $\{\theta\}$

$\beta' \leftarrow \text{subst}(\theta, \text{first}(\text{στόχοι}))$

**για** **κάθε**  $\tau \in B\Gamma$  με  $((\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n) \Rightarrow \beta) \leftarrow \text{new-vars}(\tau)$  και  $\theta' \leftarrow \text{unify}(\beta, \beta')$

$\text{νέοι-στόχοι} \leftarrow [\alpha_1, \alpha_2, \dots, \alpha_n \mid \text{rest}(\text{στόχοι})]$

$\text{απαντήσεις} \leftarrow \text{απαντήσεις} \cup \text{fol-bc-ask}(B\Gamma, \text{νέοι-στόχοι}, \text{compose}(\theta', \theta))$

**επίστρεψε**  $\text{απαντήσεις}$

Αν το  $\beta'$  ενοποιείται με τις κεφαλές  $\beta$  πολλών κανόνων, θα έχουμε πολλές απαντήσεις-ενοποιητές, μία από κάθε κανόνα. Επιστρέφουμε τελικά ένα σύνολο που περιέχει όλες τις απαντήσεις (ένα σύνολο ενοποιητών).

# Λογικός προγραμματισμός

- Παριστάνουμε σε λογική τη γνώση του κόσμου ενός προβλήματος. Εύρεση λύσης με αλγορίθμους **εξαγωγής συμπερασμάτων**.
- **Prolog**: χρησιμοποιεί οριστικές προτάσεις (definite clauses) ΠΚΛ.  
criminal(X) :- american(X), weapon(Y), sells(X,Y,Z), hostile(Z).  
Σημαίνει:  $Criminal(x) \Leftarrow (American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z))$
- **Πρόγραμμα Prolog**: συλλογή (σύζευξη) οριστικών προτάσεων.  
append([], Y, Y). *% «πρόγραμμα» συνένωσης λιστών.*  
append([A|X], Y, [A|Z]):- append(X, Y, Z).  
Το [First|Rest] παριστάνει μια λίστα με πρώτο στοιχείο First και υπόλοιπο Rest. Το Rest είναι λίστα.

# Ερωτήσεις προς την Prolog

- ?- append([a], [b, c], [a, b, c]).

yes

- ?- append([a], [b, c], Result).

Result = [a, b, c]

- ?- append(A, B, [a, b, c]).

A = [], B = [a, b, c] ;

A = [a], B = [b, c] ;

A = [a, b], B = [c] ;

A = [a, b, c], B = [] ;

no

Με «;» ζητάμε άλλη λύση.

- Αναζητεί αποδείξεις κατά την **ανάστροφη φορά**, σε γενικές γραμμές όπως ο fol-bc-ask.

– **Άρνηση ως αποτυχία**: αν δεν μπορεί να αποδείξει κάτι ( $B\Gamma \not\vdash_i \alpha$ ), το θεωρεί ψευδές ( $B\Gamma \vdash_i \neg\alpha$ ).

– Το « $\Rightarrow$ » παριστάνει ενοποίηση.

– Η ενοποίηση γίνεται χωρίς έλεγχο occur-check (βλ. unify-var).

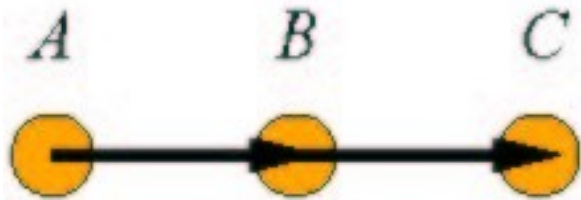


# Άλλο παράδειγμα Prolog

- ?- member(b, [a, b, c]).  
Yes
- ?- member(X, [a, b, c]).  
X = a ;  
X = b ;  
X = c ;  
no
- Ορισμός του κατηγορήματος member:  
member(X, [X|Rest]).  
member(X, [Head|Rest]):- member(X, Rest).
- Ή πιο σύντομα (αποφεύγοντας και προειδοποιήσεις ότι π.χ. η μεταβλητή Rest δε χρησιμοποιείται πουθενά αλλού στον 1<sup>ο</sup> κανόνα):  
member(X, [X|\_]).  
member(X, [\_|Rest]):- member(X, Rest).

Ανώνυμες μεταβλητές.

# Παράδειγμα προγράμματος Prolog

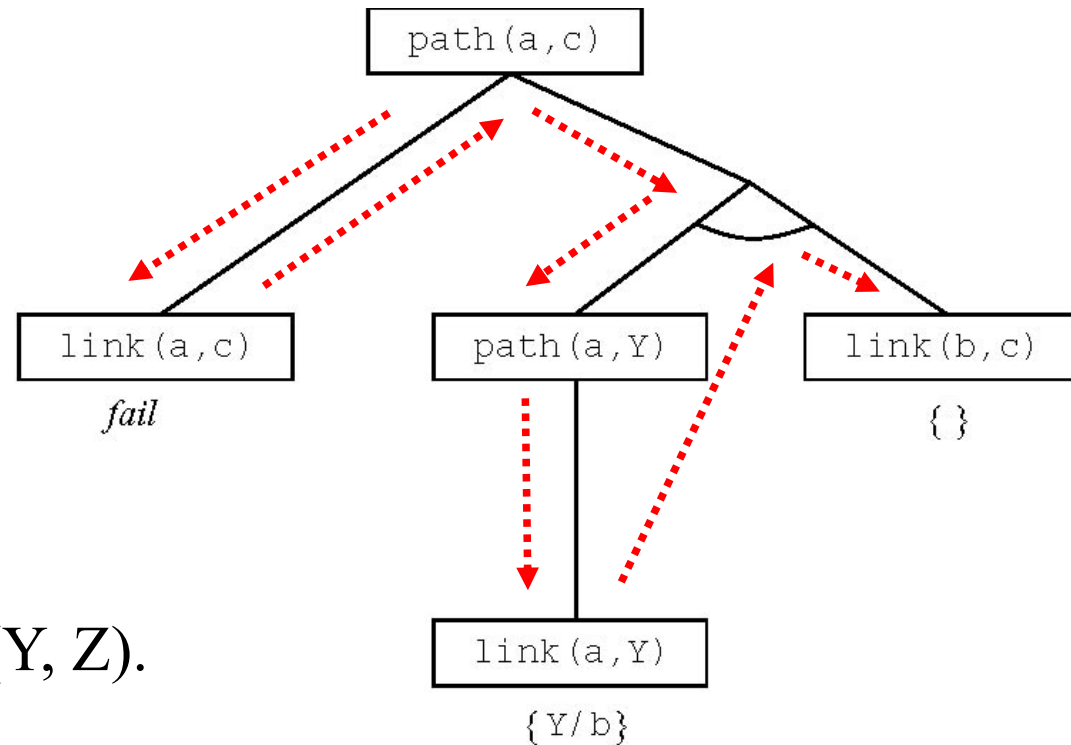


`link(a, b).`

`link(b, c).`

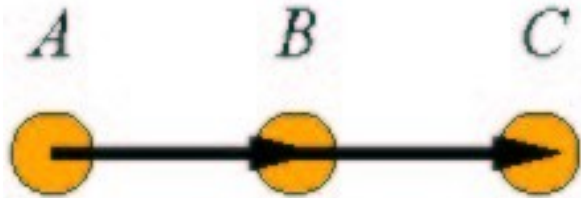
`path(X, Z):- link(X, Z).`

`path(X, Z):- path(X, Y), link(Y, Z).`



- Η Prolog προχωρά όπως ο `fol-bc-ask`, δοκιμάζοντας πρώτα τους κανόνες που βρίσκονται ψηλότερα στο αρχείο του προγράμματος, μέχρι να βρει μια απόδειξη.
- Ερώτηση: `path(a, c)`. Απάντηση: ναι.

# Άπειρα μονοπάτια στην Prolog



`link(a, b).`

`link(b, c).`

`path(X, Z):- path(X, Y), link(Y, Z).`

`path(X, Z):- link(X, Z).`

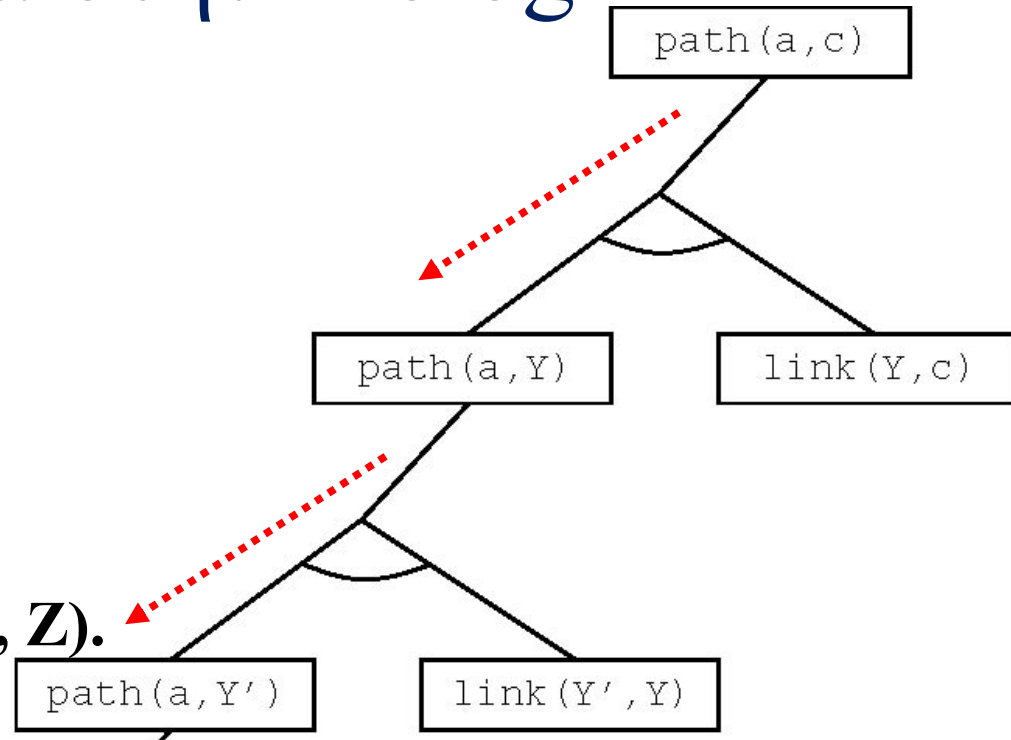
- Ερώτηση: `path(a, c)`. **Δεν απαντά ποτέ.**

- Παγιδεύεται σε **άπειρο μονοπάτι.**

- $B\Gamma \not\models_i \text{path}(a, c)$ , παρ' όλο που  $B\Gamma \models \text{path}(a, c)$ .

- **Μη πλήρης**, ακόμα και χωρίς σύμβολα συναρτήσεων.

- Ενώ ο `fol-fc-ask` είναι πλήρης (για συμπεράσματα ατομικούς τύπους) και χωρίς σύμβολα συναρτήσεων τερματίζει πάντα.



# Δοκιμάστε την Prolog...

- Θα χρειαστείτε μεταγλωττιστή/διερμηνέα **Prolog**.
  - Π.χ. **SWI-Prolog** (βλ. <http://www.swi-prolog.org/>).
- **Φόρτωμα** αρχείων με κώδικα σε Prolog:
  - Συνήθως έχουν κατάληξη «.pl».
  - Γράψτε π.χ. σε ένα αρχείο το «πρόγραμμα» `append`.
  - `consult(...)` στη γραμμή εντολών της Prolog.
  - Σε Windows: διπλό κλικ στο αρχείο .pl της γραμματικής.
- **Ερωτήσεις** προς την Prolog.
  - Μετά το φόρτωμα.
  - Στη γραμμή εντολών της Prolog, όπως στα προηγούμενα παραδείγματα.

Περισσότερα για την Prolog στα **φροντιστήρια**. Εντός ύλης αλλά μπορεί να ζητηθούν μόνο πολύ απλά προγράμματα, σαν των ασκήσεων μελέτης.

# Βιβλιογραφία

- Russel & Norvig (4η έκδοση): ενότητες 9.3, 9.4.
  - Χωρίς τις υπο-ενότητες 9.3.3 («αποδοτική προς τα εμπρός αλυσίδα εκτέλεσης»), 9.4.4 («σημασιολογία βάσεων δεδομένων της Prolog»), 9.4.5 («λογικός προγραμματισμός με περιορισμούς»).
  - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά (εκτός εξεταστέας ύλης) και τις ενότητες του κεφ. 9 που εξαιρέθηκαν.
- Βλαχάβας κ.ά: Απλή ανάγνωση των κεφ. 11, κεφ. 21 (εκτός της ενότητας 21.2.2).
  - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά (εκτός εξεταστέας ύλης) και τις ενότητες των κεφ. 11, 21 που εξαιρέθηκαν, καθώς και το κεφάλαιο 23.

# Βιβλιογραφία

- Για μια εκτενέστερη εισαγωγή στην Prolog, προτείνεται το βιβλίο «The Art of Prolog» των L. Sterling και E. Shapiro, 2<sup>η</sup> έκδοση, MIT Press, 1993.
  - Υπάρχει στη βιβλιοθήκη του ΟΠΑ, μαζί με πολλά άλλα βιβλία για προγραμματισμό σε Prolog.
  - Για τις εξετάσεις χρειάζεται να ξέρετε μόνο τα στοιχεία της Prolog που αναφέρουν οι διαφάνειες και οι ασκήσεις μελέτης.

