ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τεχνητή Νοημοσύνη

11η 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., κ.ά., 3η έκδοση, Β. Γκιούρδας Εκδοτική, 2006 και *Artificial Intelligence – A Modern Approach* των S. Russel και P. Norvig, 2η και 4η έκδοση, Prentice Hall, 2003 και 2020. Τα περισσότερα σχήματα των διαφανειών προέρχονται από αντίστοιχες διαφάνειες του δεύτερου βιβλίου.

Τι θα ακούσετε σήμερα

- Συλλογιστική με πρωτοβάθμια κατηγορηματική λογική:
  - Απαλοιφή καθολικών και υπαρξιακών ποσοδεικτών.
  - Μετατροπή σε κανονική συζευκτική μορφή.
  - Ενοποίηση και εξαγωγή συμπερασμάτων με τον κανόνα της ανάλυσης (resolution).
  - Ημι-αποκρισιμότητα της ΠΚΛ.

Κανόνας απαλοιφής $\forall$

- Αν η ΒΓ περιέχει $\forall x\ ((King(x) \wedge Greedy(x)) \Rightarrow Evil(x))$,

τότε μπορούμε να προσθέσουμε στη ΒΓ:

$((King(John) \wedge Greedy(John)) \Rightarrow Evil(John))$

$((King(Richard) \wedge Greedy(Richard)) \Rightarrow Evil(Richard))$

$((King(FatherOf(John)) \wedge Greedy(FatherOf(John))) \Rightarrow Evil(FatherOf(John)))$

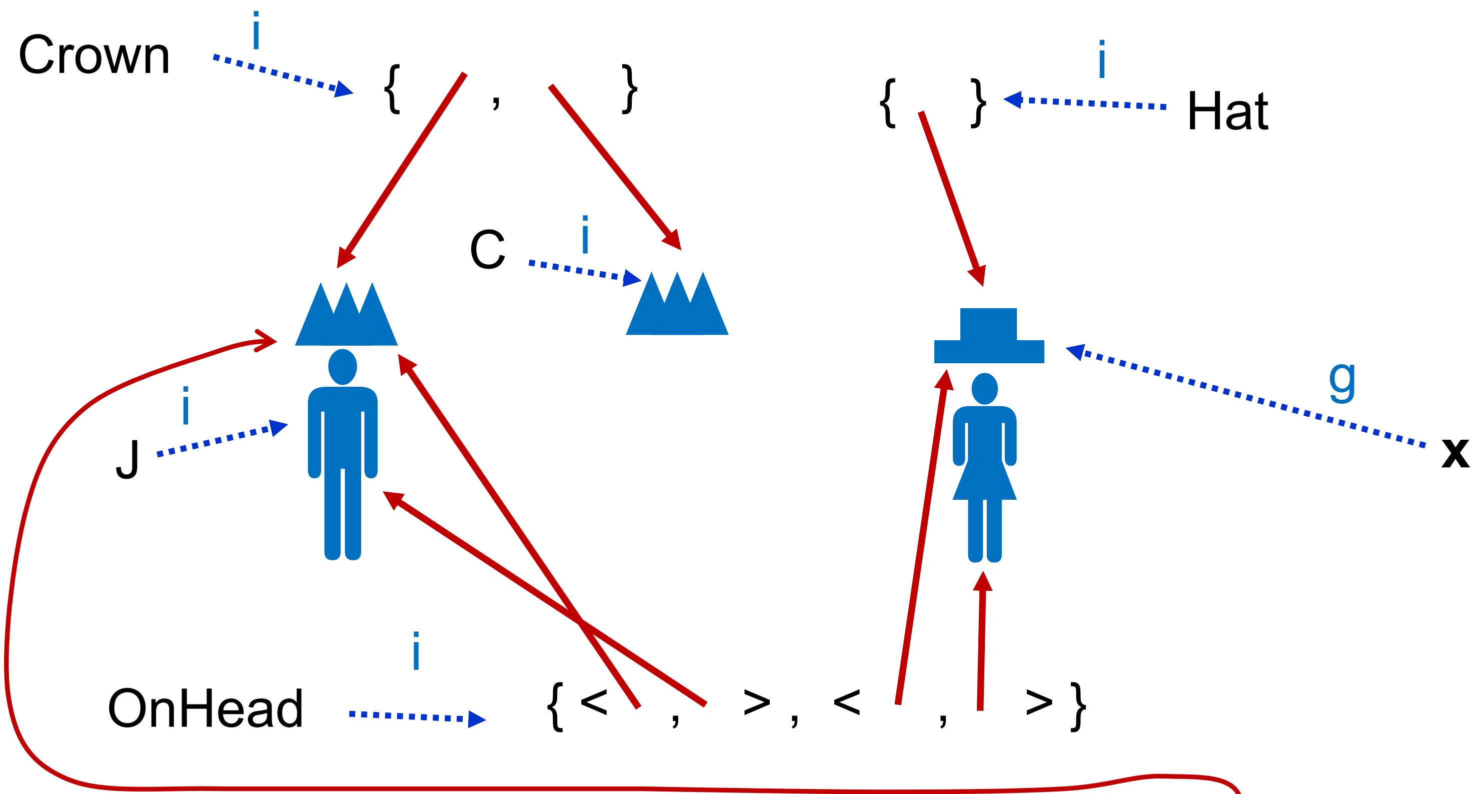
...

- Γενικότερα: $\boldsymbol{\forall \beta\ \alpha \vdash SUBST(\{\beta/\gamma\}, \alpha)}$
  - $SUBST(\{\beta/\gamma\}, \alpha)$ σημαίνει αντικατάσταση της μεταβλητής β με τον όρο γ στον τύπο α.
  - Ο γ πρέπει **να μην περιέχει μεταβλητές** (ground term).
  - Είναι κανόνας εξαγωγής συμπερασμάτων. Μπορούμε να τον χρησιμοποιήσουμε για να **προσθέσουμε τύπους** στη ΒΓ.

Απαλοιφή ∃

- Αν έχουμε π.χ. $\exists x$ (Crown(x) $\wedge$ OnHead(x, John)), τότε μπορούμε να **αντικαταστήσουμε** τον παραπάνω τύπο με (Crown(**C**) $\wedge$ OnHead(**C**, John)).
  - Το C πρέπει να είναι σταθερά («σταθερά Skolem») που **δε χρησιμοποιείται** πουθενά αλλού στο σύνολο τύπων μας.
  - Διαισθητικά, δίνουμε «όνομα» (C) σε κάτι που υπάρχει.
  - Ακριβέστερα χρησιμοποιούμε συναρτήσεις Skolem (βλ. παρακάτω).
- Ο νέος τύπος **<u>δεν</u> είναι ταυτ/κά ισοδύναμος** με τον αρχικό.
  - Σε κάποια μοντέλα/ερμηνείες μπορεί να μην αληθεύει ότι (Crown(C) $\wedge$ OnHead(C, John)), π.χ. αν το C παριστάνει καπέλο, αλλά να αληθεύει $\exists x$ (Crown(x) $\wedge$ onHead(x, John)).
- Αλλά ο αρχικός είναι **<u>ικανοποιήσιμος</u>** ανν είναι ο νέος.
  - Ικανοποιήσιμος τύπος ΠΚΛ σημαίνει ότι αληθεύει για τουλάχιστον ένα μοντέλο, ερμηνεία, ανάθεση τιμών.

Παράδειγμα



$$\| \, (\, \mathrm{Crown}(C) \wedge \mathrm{OnHead}(C, J) \,) \, \|^{i} = \mathbf{F}$$

$$\| \, (\, \mathrm{Crown}(x) \wedge \mathrm{OnHead}(x, J) \,) \, \|^{i,g[x \to o]} = \mathbf{T}$$

$$\| \, \exists x \, (\mathrm{Crown}(x) \wedge \mathrm{OnHead}(x, J) \,) \, \|^{i} = \mathbf{T}$$

Μετατροπή τύπου ΠΚΛ σε CNF

- Κάθε τύπος ΠΚΛ μπορεί να μετατραπεί σε **CNF**.
  - Π.χ. ο $\forall x\ \forall y\ \forall z\ ((American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z)) \Rightarrow Criminal(x))$
  - γίνεται $(\neg American(x) \vee \neg Weapon(y) \vee \neg Sells(x, y, z) \vee \neg Hostile(z) \vee Criminal(x))$.
  - Γενικότερα προκύπτει **σύζευξη** πολλών **διαζεύξεων**. Κάθε διάζευξη συνδέει **ατομικούς τύπους ή αρνήσεις** τους.
  - **Δεν υπάρχουν ποσοδείκτες**. Για όλες τις μεταβλητές, **υπονοούνται καθολικοί ποσοδείκτες** στην αρχή του τύπου.
- Ο νέος CNF τύπος **δεν είναι απαραίτητα ταυτολογικά ισοδύναμος** με τον αρχικό, αλλά είναι **ικανοποιήσιμος** αν και μόνο αν είναι ικανοποιήσιμος ο αρχικός τύπος.
  - Που μας αρκεί για την **εξαγωγή συμπερασμάτων** με έλεγχο μη ικανοποιησιμότητας.
  - $ΒΓ \models α$ ανν $(ΒΓ \wedge \neg α)$ μη ικανοποιήσιμος ανν $(ΒΓ \wedge \neg α)^{CNF}$ μη ικανοποιήσιμος, όπου $(...)^{CNF}$ η CNF μορφή.

Μετατροπή τύπου ΠΚΛ σε CNF

- Θεωρούμε ότι κάθε ποσοδείκτης εισάγει τη δική του μεταβλητή.
  - Διαφορετικά αλλάζουμε μεταβλητές, ώστε να μη συμπίπτουν.
- Βήμα 1: **Απαλοιφή ⇒ και ⇔**.

  $\forall x\, (\forall y\, (Animal(y) \Rightarrow Loves(x, y)) \Rightarrow \exists z\, Loves(z, x))$ γίνεται:

  $\forall x\, (\neg\forall y\, (\neg Animal(y) \vee Loves(x, y)) \vee \exists z\, Loves(z, x))$
- Βήμα 2: **Μεταφορά** ¬ στο εσωτερικό.

  Το $\neg\forall\beta\ \varphi$ γίνεται $\exists\beta\ \neg\varphi$ και το $\neg\exists\beta\ \varphi$ γίνεται $\forall\beta\ \neg\varphi$.

  $\forall x\, (\exists y\, \neg(\neg Animal(y) \vee Loves(x, y)) \vee \exists z\, Loves(z, x))$

  $\forall x\, (\exists y\, (Animal(y) \wedge \neg Loves(x, y)) \vee \exists z\, Loves(z, x))$

Συμβουλή: Εφαρμόζετε τα βήματα πάντα **με αυτή τη σειρά** και μην παραλείπετε **παρενθέσεις**!

Μετατροπή τύπου ΠΚΛ σε CNF

- Βήμα 3: **Απαλοιφή των ∃**.
  - Με **σταθερές Skolem** παίρνουμε:
    $\forall x\ ((Animal(A) \wedge \neg Loves(x, A)) \vee Loves(B, x))$
  - Πρόβλημα: ο τύπος του βήματος 2 επέτρεπε να υπάρχει διαφορετικό ζώο y για κάθε x, ενώ εδώ έχουμε το ίδιο A για όλα τα x.
  - Και ο z που αγαπά τον x μπορούσε να είναι διαφορετικός για κάθε x, ενώ εδώ έχουμε το ίδιο B για όλα τα x.
  - Αντιμετώπιση προβλήματος με **συναρτήσεις Skolem**:
    $\forall x\ ((Animal(\mathbf{F(x)}) \wedge \neg Loves(x, \mathbf{F(x)})) \vee Loves(\mathbf{G(x)}, x))$
  - Η κάθε συνάρτηση Skolem (ακριβέστερα συναρτησιακός όρος Skolem) έχει ως ορίσματα τις μεταβλητές των ποσοδεικτών ∀, μέσα στην εμβέλεια των οποίων βρισκόταν ο αντίστοιχος ποσοδείκτης ∃.

Μετατροπή τύπου ΠΚΛ σε CNF

- Βήμα 4: **Απαλοιφή των ∀**.

  Μπορούμε να τους μεταφέρουμε πλέον όλους στην αρχή και να θεωρήσουμε ότι υπονοούνται.

  $((\mathrm{Animal}(F(x)) \wedge \neg \mathrm{Loves}(x, F(x))) \vee \mathrm{Loves}(G(x), x))$

- Βήμα 5: **Επιμερισμός των ∨ και ∧**.

  $((\mathrm{Animal}(F(x)) \vee \mathrm{Loves}(G(x), x)) \wedge$
  $(\neg \mathrm{Loves}(x, F(x)) \vee \mathrm{Loves}(G(x), x)))$

- Με αυτόν τον τρόπο κάθε τύπος ΠΚΛ μπορεί να μετατραπεί σε CNF.
  - Ο νέο τύπος δεν είναι ταυτολογικά ισοδύναμος με τον αρχικό (γιατί χρησιμοποιήσαμε απαλοιφή ∃), αλλά είναι <u>ικανοποιήσιμος</u> ανν είναι ικανοποιήσιμος ο αρχικός, που μας αρκεί για αποδείξεις με έλεγχο μη ικανοποιησιμότητας.

Ενοποίηση

- **UNIFY(π_1, π_2) = θ** σημαίνει ότι υπάρχει ένα σύνολο αντικαταστάσεων θ με SUBST(θ, π_1) = SUBST(θ, π_2).
  - Δηλαδή υπάρχει ένα σύνολο αντικαταστάσεων μεταβλητών («**ενοποιητής**») που κάνει τους π_1, π_2 **ίδιους**.
  - UNIFY(Knows(John, x), Knows(John, Jane)) = {x/Jane}
  - UNIFY(Knows(John, x), Knows(y, Bill)) = {x/Bill, y/John}
  - UNIFY(Knows(John, x), Knows(y, MotherOf(y))) = {y/John, x/MotherOf(John)}
  - Θεωρούμε ότι οι τύποι δεν έχουν κοινές μεταβλητές.
- **Γενικότερος ενοποιητής**:
  - UNIFY(Knows(John, x), Knows(y, z)) = ?
  - θ_1 = {y/John, x/z}, θ_2 = {y/John, x/John, z/John}
  - Ο **ενοποιητής** θ_1 είναι **γενικότερος** του θ_2. Θέτει λιγότερους περιορισμούς στις τιμές των μεταβλητών.
  - Θα θεωρούμε ότι το αποτέλεσμα της ενοποίησης είναι ο **γενικότερος ενοποιητής** (που είναι πάντα **μοναδικός**).

Αλγόριθμος ενοποίησης 1/2

συνάρτηση unify-var(β, ξ, θ) **επιστρέφει** ενοποιητή ή αποτυχία

είσοδοι: β: μια <u>μεταβλητή</u>

ξ: οποιαδήποτε έκφραση

θ: ο ενοποιητής ως τώρα

αν $\{\beta/\textit{τιμή}\} \in \theta$ **τότε επίστρεψε** unify(*τιμή*, ξ, θ)

διαφορετικά αν $\{\xi/\textit{τιμή}\} \in \theta$ **τότε επίστρεψε** unify(β, *τιμή*, θ)

διαφορετικά αν occur-check?(β, ξ) **τότε επίστρεψε** αποτυχία

διαφορετικά επίστρεψε $\{\beta/\xi\} \cup \theta$

Ελέγχει αν η β εμφανίζεται μέσα στην ξ. Π.χ. είναι αδύνατον να ενοποιήσουμε τα x και MotherOf(x).

Αλγόριθμος ενοποίησης 2/2

συνάρτηση unify(ζ, ξ, θ) **επιστρέφει** το γενικότερο ενοποιητή των ζ και ξ ή αποτυχία

είσοδοι: ζ, ξ: ατομικοί τύποι, όροι, σύμβολα σχέσεων, σύμβολα συναρτήσεων ή λίστες αυτών

θ: ο ενοποιητής ως τώρα (αρχικά {}) ή αποτυχία

αν θ = αποτυχία **τότε επίστρεψε** αποτυχία

διαφορετικά αν $\zeta = \xi$ **τότε επίστρεψε** θ

διαφορετικά αν variable?(ζ) **τότε επίστρεψε** unify-var(ζ, ξ, θ)

διαφορετικά αν variable?(ξ) **τότε επίστρεψε** unify-var(ξ, ζ, θ)

διαφορετικά αν compound?(ζ) **και** compound?(ξ) **τότε**
επίστρεψε unify(args(ζ), args(ξ), unify(op(ζ), op(ξ), θ))

διαφορετικά αν list?(ζ) **και** list?(ξ) **τότε**
επίστρεψε unify(rest(ζ), rest(ξ), unify(first(ζ), first(ξ), θ))

διαφορετικά επίστρεψε αποτυχία

Ενοποιούμε πρώτα τα σύμβολα σχέσεων ή συναρτήσεων (op(ζ) και op(ξ)). Αν δεν ενοποιούνται, ο ενοποιητής θα γίνει *αποτυχία.*

Ενοποιούμε αρχικά τα πρώτα στοιχεία των λιστών και μετά τα υπόλοιπα των λιστών.

Κανόνας ανάλυσης για ΠΚΛ

- Οι τύποι πρέπει πρώτα να έχουν μετατραπεί σε **CNF**.
  - Κατόπιν θεωρούμε κάθε διάζευξη ως ξεχωριστό τύπο.
- Αν UNIFY(l_i, $\neg m_j$) = θ, τότε:

$$(\boldsymbol{l_1} \vee \ldots \vee \boldsymbol{l_k}), (\boldsymbol{m_1} \vee \ldots \vee \boldsymbol{m_n}) \vdash \mathrm{SUBST}(\theta, (\boldsymbol{l_1} \vee \ldots \vee \boldsymbol{l_{i-1}} \vee \boldsymbol{l_{i+1}} \ldots \vee \boldsymbol{l_k} \vee \boldsymbol{m_1} \vee \ldots \vee \boldsymbol{m_{j-1}} \vee \boldsymbol{m_{j+1}} \vee \ldots \vee \boldsymbol{m_n})$$

  - Ενώνουμε τις δύο διαζεύξεις και αφαιρούμε τα l_i και m_j.
  - Στην ΠΚΛ τα $\boldsymbol{l_i}$ και $\boldsymbol{\neg m_j}$ δεν απαιτείται να είναι τα ίδια. Αρκεί να **ενοποιούνται**. Εφαρμόζουμε τον ενοποιητή τους θ και στο συμπέρασμα. Παράδειγμα:
  - [Animal(F(x)) ∨ **Loves(G(x), x)**], [**¬Loves(u, v)** ∨ ¬Kills(u, v)] ├ [Animal(F(x)) ∨ ¬Kills(G(x), x)] με θ = {u/G(x), v/x}.
  - Επίσης **αν κάποιοι ατομικοί τύποι** (ή αρνήσεις) **του συμπεράσματος μπορούν να ενοποιηθούν**, τους ενοποιούμε και **κρατάμε μόνο έναν**. Εφαρμόζουμε τον ενοποιητή τους στο συμπέρασμα.

Παράδειγμα ΒΓ

$\forall x\ (\forall y\ (Animal(y) \Rightarrow Loves(x, y)) \Rightarrow \exists z\ Loves(z, x))$

Όποιον αγαπάει όλα τα ζώα τον αγαπάει κάποιος.

$\forall x\ (\exists y\ (Animal(y) \wedge Kills(x, y)) \Rightarrow \forall z\ \neg Loves(z, x))$

Όποιον σκοτώσει ένα ζώο δεν τον αγαπάει κανείς.

$\forall x\ (Animal(x) \Rightarrow Loves(Jack, x))$

Ο Jack αγαπάει όλα τα ζώα.

$(Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna))$

$Cat(Tuna)$

$\forall x\ (Cat(x) \Rightarrow Animal(x))$

$\neg Kills(Curiosity, Tuna)$

Άρνηση αυτού που θέλουμε να αποδείξουμε.

Μετατροπή σε CNF και σπάσιμο σε διαζεύξεις

$(Animal(F(x)) \vee Loves(G(x), x))$

$(\neg Loves(x, F(x)) \vee Loves(G(x), x))$

$(\neg Animal(y) \vee \neg Kills(x, y) \vee \neg Loves(z, x))$

$(\neg Animal(x) \vee Loves(Jack, x))$

$(Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna))$

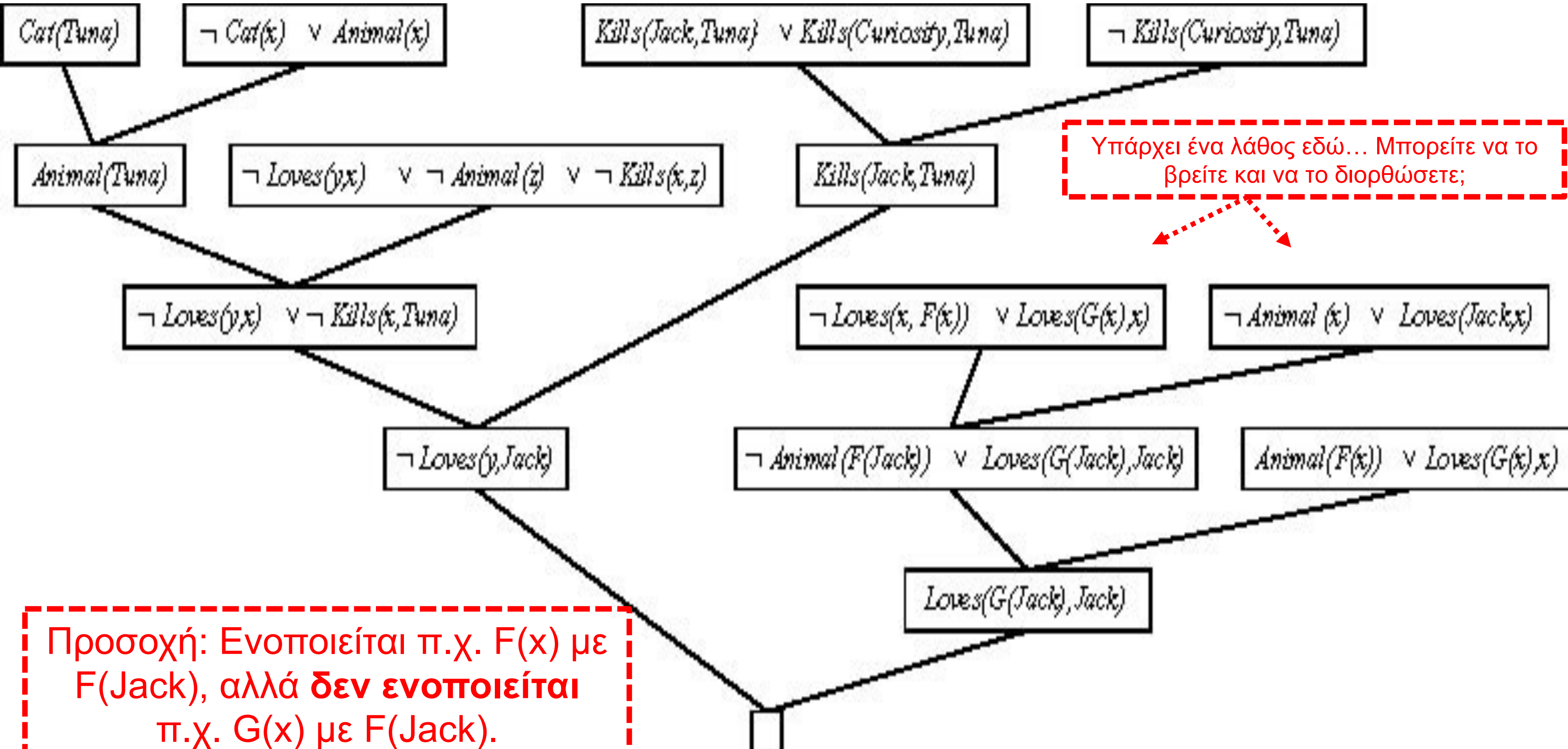
$Cat(Tuna)$

$(\neg Cat(x) \vee Animal(x))$

$\neg Kills(Curiosity, Tuna)$

Προσοχή: Πριν εφαρμόσουμε τον κανόνα της ανάλυσης (επόμενη διαφάνεια), κάθε τύπος πρέπει να είναι **μία διάζευξη** (ατομικών τύπων ή αρνήσεών τους). Αν από την μετατροπή σε CNF έχει προκύψει ένας τύπος που είναι σύζευξη πολλών διαζεύξεων, τον σπάμε σε ξεχωριστούς τύπους, έναν για κάθε διάζευξη.

Δένδρο απόδειξης



Υπάρχει ένα λάθος εδώ… Μπορείτε να το βρείτε και να το διορθώσετε;

Προσοχή: Ενοποιείται π.χ. F(x) με F(Jack), αλλά **δεν ενοποιείται** π.χ. G(x) με F(Jack).

Ορθότητα: Με το δέντρο απόδειξης έχουμε δείξει (αφού εφαρμόζουμε μόνο τον κανόνα της ανάλυσης, που είναι ορθός) ότι $(B\Gamma \wedge \neg \alpha)^{CNF} \models \square$. Και αφού η κενή διάζευξη δεν αληθεύει ποτέ, έχουμε δείξει ότι ο $(B\Gamma \wedge \neg \alpha)^{CNF}$ είναι μη ικανοποιήσιμος, οπότε και ο $(B\Gamma \wedge \neg \alpha)$ είναι μη ικανοποιήσιμος, άρα $B\Gamma \models \alpha$.

Άλλο παράδειγμα

$(\neg Dog(x_1) \vee Animal(x_1))$

$Dog(Fido)$

$(Eats(x_2) \vee \neg Animal(x_2))$

$\neg Eats(Fido)$

άρνηση του
προς απόδειξη
τύπου

φ_1 φ_2'

$(\neg Dog(x_1) \vee Animal(x_1))$

φ_3 $\neg\varphi_2$

$(Eats(x_2) \vee \neg Animal(x_2))$

$\{x_1/x_2\}$

$(\neg Dog(x_2) \vee Eats(x_2))$

$Dog(Fido)$

$\{x_2/Fido\}$

$\neg Eats(Fido)$ $Eats(Fido)$

$\square$ άτοπο

Άλλο δέντρο απόδειξης

$(\neg Dog(x_1) \lor Animal(x_1))$
$Dog(Fido)$
$(Eats(x_2) \lor \neg Animal(x_2))$
$\neg Eats(Fido)$

$(\neg Dog(x_1) \lor Animal(x_1))$ $\quad$ $Dog(Fido)$

$\{x_1/Fido\}$

$Animal(Fido)$

$(Eats(x_2) \lor \neg Animal(x_2))$

$\{x_2/Fido\}$

$\neg Eats(Fido)$ $\quad$ $Eats(Fido)$

$\square$ άτοπο

Προσοχή: Σε κάθε «ζευγάρωμα» τύπων, φεύγει πάντα **ένα μόνο** στοιχείο διάζευξης από τον έναν τύπο **και ένα μόνο** από τον άλλον.

Αναζήτηση δέντρου απόδειξης

- **Αναζήτηση** σε χώρο καταστάσεων:
  - Κάθε **κατάσταση** περιέχει το **δέντρο απόδειξης** που έχουμε κατασκευάσει ως εκείνη τη στιγμή και την **τρέχουσα ΒΓ**.
  - **Αρχική κατάσταση: κενό δέντρο απόδειξης** και **αρχική ΒΓ**, στην οποία έχουμε προσθέσει το **¬α**, όπου α το προς απόδειξη. Η ΒΓ (μαζί με το ¬α) έχουν μετατραπεί σε **CNF**. **Κάθε διάζευξη** της CNF μορφής θεωρείται **ξεχωριστός τύπος**.
  - **Τελεστής μετάβασης**: επεκτείνουμε το δέντρο απόδειξης της κατάστασης χρησιμοποιώντας τύπους της ΒΓ και τον **κανόνα της ανάλυσης**. Το συμπέρασμα προστίθεται και στη ΒΓ.
  - **Τελική κατάσταση**: το **δέντρο** απόδειξης **καταλήγει σε** □.
- Ψάχνουμε τον χώρο καταστάσεων με **<u>πλήρη</u> αλγόριθμο αναζήτησης** (π.χ. BFS, A*, ...).

Εξαγωγή συμπερασμάτων με ανάλυση

- Εισάγουμε την **άρνηση** ¬α του προς απόδειξη α στη ΒΓ, μετατρέπουμε σε CNF και ψάχνουμε το χώρο αναζήτησης της προηγούμενης διαφάνειας με **πλήρη αλγόριθμο αναζήτησης**.
  - **Αν παραχθεί** □, τότε απαντάμε **ΒΓ** $\vdash_i$ **α**.
  - Διαφορετικά, απαντάμε ΒΓ $\nvdash_i$ α
- **Ορθός** τρόπος εξαγωγής συμπερασμάτων: αν **ΒΓ** $\vdash_i$ **α**, τότε ΒΓ $\models$ α (βλ. σημειώσεις διαφάνειας 17).
- **Πλήρης** εξαγωγή συμπερασμάτων, με την έννοια ότι δοθέντος ενός **συγκεκριμένου** α με ΒΓ $\models$ α, απαντά πάντα ΒΓ $\vdash_i$ α.
  - Παράγει την κενή διάζευξη μετά από πεπερασμένο αριθμό βημάτων. Η απόδειξη παραλείπεται.
  - Δεν μπορεί όμως να απαριθμήσει όλα τα συμπεράσματα α που έπονται ταυτολογικά από τη ΒΓ.

Ανάλυση και ημι-αποκρισιμότητα

- **Αν ΒΓ $\models$ α**, τότε παράγει πάντα □ μετά από πεπερασμένο αριθμό βημάτων και **αποκρίνεται ΒΓ $\vdash_i$ α** (πληρότητα).
- **Αν ΒΓ $\not\models$ α** και υπάρχουν σύμβολα συναρτήσεων, **δεν τερματίζει πάντα**, γιατί ο χώρος αναζήτησης μπορεί να είναι άπειρος.

(¬Human(FirstSon(x1)) ∨ Human(x1))

¬Human(John)

άρνηση του προς απόδειξη

x1/John

(¬Human(FirstSon(x2)) ∨ Human(x2))

¬Human(FirstSon(John))

x2/FirstSon(John)

¬Human(FirstSon(FirstSon(John)))

…

Όταν ξαναχρησιμοποιούμε έναν τύπο της ΒΓ, του αλλάζουμε μεταβλητές. Εδώ χρησιμοποιούμε διαρκώς το μοναδικό τύπο της ΒΓ (Human(FirstSon(x)) $\Rightarrow$ Human(x)), που σε CNF γράφεται: (¬Human(FirstSon(x)) ∨ Human(x))). Δεν υπάρχει τελική κατάσταση (δέντρο απόδειξης που να καταλήγει σε □), αλλά η αναζήτηση συνεχίζεται επ' άπειρον, γιατί ο χώρος αναζήτησης είναι άπειρος.

Ημι-αποκρισιμότητα της ΠΚΛ

- Γενικότερα, η **εξαγωγή συμπερασμάτων στην ΠΚΛ** είναι **ημι-αποκρίσιμο** πρόβλημα.
  - **Υπάρχουν** (πλήρεις) αλγόριθμοι που για κάθε ΒΓ και α με **ΒΓ $\models$ α** αποκρίνονται πάντα ***αληθές*** (ΒΓ $\vdash_i$ α).
  - **Δεν υπάρχει** αλγόριθμος που επιπλέον όταν **ΒΓ $\not\models$ α** να αποκρίνεται πάντα ***ψευδές*** (ΒΓ $\nvdash_i$ α).
- Αντίθετα, στην **προτασιακή λογική** μπορούμε πάντα να απαντήσουμε (το πρόβλημα είναι **αποκρίσιμο**).
  - Π.χ. ελέγχοντας εξαντλητικά όλα τα μοντέλα (βλ. TT-Entails?), γιατί υπάρχει πεπερασμένος αριθμός μοντέλων.
  - Ενώ τα μοντέλα/ερμηνείες στην ΠΚΛ είναι πάρα πολλά (άπειρα αν π.χ. περιλάβουμε τους φυσικούς αριθμούς στο D) και δεν μπορούμε να τα ελέγξουμε εξαντλητικά.

Βιβλιογραφία

- Russel & Norvig (4η έκδοση): εισαγωγή ενότητας 9.1, εισαγωγή ενότητας 9.2 και υπο-ενότητα 9.2.1, ενότητα 9.5 ως και υπο-ενότητα 9.5.3.
  - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά και τα υπόλοιπα τμήματα των ενοτήτων 9.1, 9.2, 9.5.
  - Με τις υπο-ενότητες 9.3 και 9.4 θα ασχοληθούμε στην επόμενη διάλεξη.
- Βλαχάβας κ.ά: υπόλοιπο ενότητας 9.2.1 (περί ενοποίησης), ενότητα 9.2.2 (εκτός των υπο-ενοτήτων «Προτασιακή μορφή της ΚΛ», «Μορφή Kowalski»), ενότητα 9.2.3 (εκτός της υπο-ενότητας «Αρχή της ανάλυσης και μορφή Kowalski»), ενότητα 9.2.4.
  - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά (εκτός εξεταστέας ύλης) και τα τμήματα του κεφαλαίου 9 που εξαιρέθηκαν.