Τεχνητή Νοημοσύνη

1Η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ

Κουστένης Χρίστος | Α.Μ : el20227
Ακαδημαϊκό Έτος 2022-2023

Άσκηση 1

1.

$(p \Rightarrow \neg(p \lor q)) \Longleftrightarrow ((r \land \neg q) \lor r)$

Βήμα 1 (Αντικατάσταση Συνεπαγωγών)

$(\neg(p \Rightarrow \neg(p \lor q)) \lor ((r \land \neg q) \lor r)) \land (\neg((r \land \neg q) \lor r)) \lor (p \Rightarrow \neg(p \lor q)))$

$(\neg(\neg p \lor \neg(p \lor q)) \lor ((r \land \neg q) \lor r)) \land (\neg((r \land \neg q) \lor r) \lor (\neg p \lor \neg(p \lor q)))$

Βήμα 2 (Μετακίνηση Άρνησης Μπροστά από τις ατομικές προτάσεις)

$((p \land (p \lor q)) \lor ((r \land \neg q) \lor r)) \land (((\neg r \lor q) \land \neg r) \lor (\neg p \lor (\neg p \land \neg q))$

Βήμα 3 (Επιμερισμός διαζεύξεων)

$((r \land \neg q) \lor r)) = ((r \lor r) \land (\neg q \lor r))$

$((\neg p \lor (\neg p \land \neg q))) = (\neg p \lor \neg p) \land ((\neg p \lor \neg q))$

Οπότε έχουμε:

$((p \land (p \lor q)) \lor ((r \lor r) \land (\neg q \lor r))) \land (((\neg r \lor q) \land \neg r) \lor ((\neg p \lor \neg p) \land (\neg p \lor \neg q)))$

Βήμα 4 (Απλοποίηση Σχέσεων)

$((p \land (p \lor q)) \lor (r \land (\neg q \lor r))) \land (((\neg r \lor q) \land \neg r) \lor (\neg p \land (\neg p \lor \neg q)))$

Για την πρώτη μεγάλη παρένθεση δηλαδη $((p \land (p \lor q)) \lor (r \land (\neg q \lor r)))$ έχουμε:

$((p \land (p \lor q)) \lor (r \land (\neg q \lor r))) = ((p \land (p \lor q)) \lor r) \land ((p \land (p \lor q)) \lor (\neg q \lor r))$

$((p \lor r) \land (p \lor q \lor r)) \land (p \lor \neg q \lor r) \land (p \lor q \lor \neg q \lor r)$

$((p \lor r) \land (p \lor q \lor r)) \land (p \lor \neg q \lor r)$

Οπότε η πρώτη πρόταση γράφεται: [p, r], [p, q, r], [p, ¬q, r]

Για τη δεύτερη δηλαδη $(((\neg r \lor q) \land \neg r) \lor (\neg p \land (\neg p \lor \neg q)))$ έχουμε:

$((\neg r \lor q) \land \neg r) \lor (\neg p \land (\neg p \lor \neg q))$

$(((\neg r \lor q) \land \neg r) \lor \neg p) \land (((\neg r \lor q) \land \neg r) \lor (\neg p \lor \neg q))$

$((\neg r \lor q \lor \neg p) \land (\neg r \lor \neg p)) \land ((\neg r \lor q \lor \neg p \lor \neg q) \land (\neg r \lor \neg p \lor \neg q))$

$((\neg r \lor q \lor \neg p) \land (\neg r \lor \neg p)) \land (\land(\neg r \lor \neg p \lor \neg q))$

Οπότε η δεύτερη πρόταση γράφεται: [¬r, q, ¬p], [¬r, ¬p], [¬r, ¬p, ¬q]

Τελικά η πρόταση γράφεται στη μορφή:

[p, r], [p, q, r], [p, ¬q, r], [¬r, q, ¬p], [¬r, ¬p], [¬r, ¬p, ¬q]

και απλοποιώντας περαιτέρω τα ζεύγη των όρων [p, r], [p, q, r] και [¬r, q, ¬p], [¬r, ¬p] έχουμε

{ [p, q, r], [p, ¬q, r], [¬r, q, ¬p], [¬r, ¬p, ¬q] }

1.

$\forall x.\forall y.\exists z.(\forall w.(p(x, y) \Rightarrow q(w)) \lor (p(y, z) \Rightarrow \neg q(w)))$

Βήμα 1 (Αντικάτασταση Συνεπαγωγών)

$\forall x.\forall y.\exists z.(\forall w.(\neg p(x, y) \lor q(w)) \lor (\neg p(y, z) \lor \neg q(w)))$

Βήμα 2 (Μετακίνηση Άρνησης Μπροστά από τις ατομικές προτάσεις) -

Βήμα 3 (Απόδοση Μοναδικών Ονομάτων σε όλες τις μεταβλητές)

$\forall x.\forall y.(\forall w.(\neg p(x, y) \lor q(w)) \lor \forall w_1.(\neg p(y, z) \lor \neg q(w_1)))$

Βήμα 4 (Αφαίρεση Υπαρξιακών Ποσοδεικτών - Χρήση συναρτήσεων Skolem ή μοναδικών σταθερών)

Η μεταβλητή z εξαρτάται από τις μεταβλητές x και y, γι'αυτό θεωρώ $z = f(x,y)$

$\forall x.\forall y.(\forall w.(\neg p(x, y) \lor q(w)) \lor \forall w_1.(\neg p(y, f(x, y)) \lor \neg q(w_1)))$

Βήμα 5 (Μετακίνηση Καθολικών Ποσοδεικτών εκτός του πεδίου των ∨ και ∧)

$\forall x.\forall y.\forall w.\forall m.(\neg p(x, y) \lor q(w) \lor \neg p(y, f(x, y)) \lor \neg q(w_1))$

Η πρόταση γράφεται στη μορφή: $[\neg p(x, y), q(w), \neg p(y, f(x, y)), \neg q(w_1)]$

Άσκηση 2

Θεωρούμε την ερμηνεία ΔΙ : {a, b, m},PI : {(a, b),(b, a)},QI : {(a),(b),(m)} της γνώσης K.

Μπορούμε να γράψουμε την τελευταία πρόταση στη μορφή:

[¬p(x, f(x, y)), q(w), ¬p(y, f(x, y)), ¬q(m)].

Οπότε για w = a,b η q(a), q(b) είναι θετικά. Επίσης για w = m, έχουμε [q(m), ¬q(m)] που αληθεύει.

Επομένως, η ερμηνεία είναι πράγματι μοντέλο της γνώσης K.

Άσκηση 3

Για να είναι η γνώση K συνεπής θα πρέπει κάθε έκφρασή της να είναι λογικό συμπέρασμα από τις υπόλοιπες εκφράσεις της γνώσης. Για τις εκφράσεις {[p(a, b)], [p(b, a)], [q(a)], [¬q(b)], }, αυτό αποδεικνύεται εύκολα.

Στη συνέχεια εφαρμόζουμε αλγόριθμο ανάλυσης για να δείξουμε το ζητούμενο.

Αφαιρούμε την τελευταία πρόταση, έστω u από τη γνώση K και προσθέτουμε την αντίθετή της, οπότε προκύπτει η {K' = K ∪ {¬u}} με K' = {[p(a, b)], [p(b, a)], [q(a)], [¬q(b)], [p(f(x, y), y)], [¬q(w)], [p(y, f(x, y))], [q(m)]}. Αυτή η γνώση οδηγεί σε αντίφαση, γιατί για w=m έχω {[¬q(m)], [q(m)]} . Επομένως, ο αλγόριθμος τερματίζει με επιτυχία, δηλαδή η γνώση K συνεπάγεται σημασιολογικά την πρόταση p και η γνώση K είναι συνεπής.

Άσκηση 4

Πρόταση 1: ∀x.(∃y.p(x, y) ⇒ q(a)) ∀x.(¬(∃y.p(x, y)) ∨ q(a)) ∀x.((∀y.¬p(x, y)) ∨ q(a)) ∀x.∀y.¬p(x, y) ∨ q(a) CNF : {¬p(x, f(x)), q(a)}

Πρόταση 2: (∀x.∃y.p(x, y)) ⇒ q(a)) ¬(∀x.∃y.p(x, y))) ∨ q(a) (∃x.∀y.¬p(x, y)) ∨ q(a) ∃x.∀y.¬p(x, y) ∨ q(a) CNF : {¬p(c, y), q(a)}

Άσκηση 5

1.

Το σύμπαν του λογικού προγράμματος UP = {a, b, c, d, e, f}.

Τα στοιχεία στη βάση είναι συνολικά 6(κατηγορηματα)*36(συνδυασμοί μεταξύ UP στοιχείων) = 216

2

Για forward chaining :

A_1.parent(x, y) ← father(x, y).

A_2.parent(x, y) ← mother(x, y).

A_3.sibling(y, z) ← parent(y, x), parent(z, x).

A_4. sibling(x, y) ← sibling(y, x).

A_5.grandparent(x, z) ← parent(x, y), parent(y, z).

A_6.cousin(y, z) ← grandparent(y, x), grandparent(z, x).

A_7.mother(a, b) ← .

A_8.father(b, c) ← .

A_9.mother(d, b) ← .

A_{10}.mother(e, c) ← .

A_{11}.mother(f, b) ← .

α) parent(b, c) ⇒ A2, A8(A1′)

parent(a, b) ⇒ A2, A7(A2′)

parent(d, b) ⇒ A2, A9(A3′)

parent(e, c) ⇒ A2, A10(A4′)

parent(f, b) ⇒ A2, A11(A5′)

sibling(a, d) ⇒ 3, A2, A3 ′ (A6′)

sibling(a, f) ⇒ A3, A2, A5 ′ (A7′)

sibling(d, f) ⇒ A3, A3, A5 ′ (A8′)

sibling(d, a) ⇒ A4, A6 ′ (A9′)

sibling(f, a) ⇒ A4, A7 ′ (A10′)

sibling(f, d) ⇒ A4, A8 ′ (A11′)

grandparent(a, c) ⇒ A5, A1 ′ , A2 ′ (A6′′)

grandparent(d, c) ⇒ A5, A3 ′ , A1 ′ (A7′′)

grandparent(f, c) ⇒ A5, A5 ′ , A1 ′ (A8′′)

cousin(a, d) ⇒ A6 A6′′ A7 ′′

cousin(a, F) ⇒ A6 A6′′ A8 ′′

cousin(d, f) ⇒ A6 A7′′ A8 ′′

Οπότε συμπεραίνουμε ότι : cousin(a,e): αποτυχία, cousin(a,f): επιτυχία, και sibling(a,e): αποτυχία.

3.

Αν μπορούσα να προσθέσω μια πρόταση θα προσέθετα την εξής: cousin(y, z) ← (¬sibling(y, z) ∧ (grandparent(y, x), grandparent(z, x)),προκειμένου να αποφύγω τα αδέρφια να είναι ταυτόχρονα και ξαδέρφια, γεγονός άτοπο. Επίσης θα προσέθετα επιπλέον και την εξής: ¬mother(x, y) ← father(x, y).

Άσκηση 6

Καταρχάς θα δείξουμε ότι ανεξαρτήτως learning rate, ο αλγόριθμος batch percepton χρειάζεται πάντα τον ίδιο αριθμό επαναλήψεων. Πράγματι, έστω στη συγκεκριμένη περίπτωση, πως τερματίζει ο batch percepton με learning rate ίσο με 1 (ο άλλος έχει learning rate ίσο με η) πρώτος μετά από k επαναλήψεις. Τότε, μιας και τα αρχικά βάρη είναι ίσα με 0 για κάθε i, θα ισχύει ότι $w_1 = kxiyi$. Θα ισχύει επίσης ότι $w_2(k) = k\eta x_i y_i$. Ο αλγόριθμος batch percepton τερματίζει όταν ισχύει για όλα τα i η σχέση $y_i\langle w_1, x_i\rangle > 0$. Όμως έτσι θα ισχύει και ότι $y_i\langle \eta w_1, x_i\rangle > 0$, άρα ότι $y_i\langle w_2(k), x_i\rangle > 0$. Άρα στην k-οστή επανάληψη τερματίζει και ο batch percepton με learning rate ίσο με η. Ομοίως χειριζόμαστε την περίπτωση του να τερματίζει πρώτος ο batch percepton με learning rate ίσο με η, όπου δείχνουμε ότι θα τερματίσει ταυτόχρονα και η εκτέλεσή του με rate ίσο με 1. Άρα, για να απαντήσουμε και στα δύο ερωτήματα, η σύγκλιση των $w1$ και $w2$ θα πραγματοποιηθεί μετά από ίδιο αριθμό επαναλήψεων χωρίς αυτός να εξαρτάται από τη $\boldsymbol{\eta}$ και αν πούμε ότι αυτός ο αριθμός ισούται με k, θα είναι:

$w_2/w_1 = k\eta x_i y_i \,/\, k x_i y_i = \boldsymbol{\eta}$

Άσκηση 7

1.

Θεωρούμε καταρχάς πως το πρόβλημα ισοδυναμεί με το να μας δώσει κάποιος $2d$ αριθμούς γραμμένους στο δυαδικό σύστημα αρίθμησης, οι οποίοι θα είναι από το 0 έως το $2d - 1$ και θα έχουν d bits, και εμείς θα πρέπει να αντιστοιχίσουμε κάθε τέτοιο αριθμό του πεδίου ορισμού σε μια τιμή 0, 1 (YES/NO δηλαδή). Ένας απλοϊκός τρόπος σκέψης για να σχεδιάσουμε τα δέντρα αποφάσεων χωρίς να μας ενδιαφέρει να τα κάνουμε όσο το δυνατόν πιο μικρά είναι ο εξής: Αρχίζουμε με το MSB από τα d bits, θεωρώντας σε κάθε επόμενο επίπεδο να καθορίζει την απόφαση το επόμενο λιγότερο πιο σημαντικό bit. Αρχίζοντας, λοιπόν με το MSB στη ρίζα, έχουμε 2 περιπτώσεις, να είναι αυτό 1 ή 0. Υπάρχουν $2d{-}1$ αριθμοί στο πεδίο ορισμού που έχουν MSB 1 και $2d{-}1$ που έχουν 0. Όπως θα χωρίσουμε λοιπόν τους αριθμούς σε δύο κατηγορίες, αν όλοι μιας κατηγορίας έχουν ίδια απόφαση (YES/NO), τότε αυτό το «παρακλάδι» δεν χρειάζεται να εξεταστεί με βάση άλλα κριτήρια απόφασης (τα υπόλοιπα bits) και οπότε βάζουμε σε αυτό το παρακλάδι κατευθείαν σαν φύλο την απόφαση (1 ή 0). Διαφορετικά, για να αποφασίσουμε για αυτό το παρακλάδι πρέπει να φτιάξουμε ένα υποδέντρο απόφασης, το οποίο τώρα αγνοεί το πρώτο bit του αριθμού και θα εξετάζει $2d{-}1$ αριθμούς. Βλέπουμε λοιπόν πως ανάγουμε το αρχικό πρόβλημα στο ίδιο με το d να έχει μειωθεί, οπότε προχωράμε και επαναλαμβάνουμε τη διαδικασία αναδρομικά. Βέβαια ανάλογα με το mapping που θα μας δίνεται θα βολεύει να πάρουμε τα bit με μια πιο συγκεκριμένη σειρά από ότι το να πάμε με τη σειρά από MSB σε LSB. Εδώ μπορούμε να χρησιμοποιήσουμε κριτήριο εντροπίας για παράδειγμα, για να βρούμε από πιο ψηφίο είναι πιο βολικό να ξεκινήσουμε, ώστε να κάνουμε το δέντρο να έχει

λιγότερα παρακλάδια. Κατά τα άλλα, όπως και να επιλέξουμε τα bits στη σειρά, δεν θα αλλάξει η παραπάνω διαδικασία.

2.

Μέγιστο βάθος θα είναι d+1 (θεωρούμε και τα φύλλα στο βάθος του δέντρου), αφού κάθε φορά που εμβαθύνουμε, διαιρούμε με το δύο το πλήθος των αριθμών που εξετάζουμε στο κάθε παρακλάδι, οπότε στη χειρότερη στο βάθος d θα έχουμε σε αυτό το παρακλάδι ένα στοιχείο που θα έχουμε από τον πίνακα αντιστοίχισης κατευθείαν την απόφαση μετά, οπότε θα πάει στο επόμενο βάθος η απόφαση ως φύλλο.

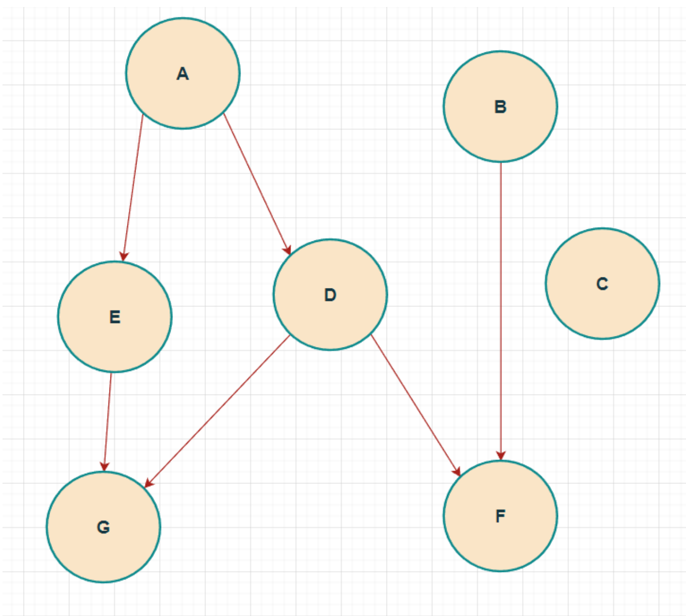
3.

Λόγω του ότι κάθε αριθμός του πεδίου ορισμού μπορεί να ταξινομηθεί με το δέντρο αυτό σε μια απόφαση, έχουμε ότι το VC είναι $2d$. Προφανώς, το VC τετριμμένα δεν μπορεί να είναι μεγαλύτερο γιατί δεν έχουμε άλλο διαφορετικό σημείο στο πεδίο ορισμού, ώστε να πούμε ότι μπορούμε να κατακερματίσουμε επιπλέον σημείο.

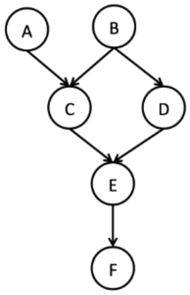
Άσκηση 8

1.

Το δίκτυο πίστης της κατανομής P (A,B,C,D,E,F) = P(A) P(B) P(C) P(D|A) P(E|A) P(F|B, D) P(G|D,E)



2.



Για το παραπάνω δίκτυο πίστης ισχύει :

$J(<A, B, C, D, E, F>) = P(A)P(B)P(C|A, B)P(D|B)P(E|C, D)P(F|E)$

3.

minimum_parameters = $2^0 + 2^0 + 2^2 + 2^1 + 2^2 + 2^1 = 14$

Ο εκθέτης προκύπτει από το πλήθος γονεών κάθε κόμβου και η βάση από το γεγονός ότι είναι boolean ο τύπος δεδομένων του κάθε κόμβου.