Τεχνητή Νοημοσύνη

Project 3 - Σχόλια για τον κώδικα του **Kakuro & Ασκήσεις**

Κωνσταντίνος Πυθαρούλιος
Α.Μ.: 111520 1600142

1. Μοντελοποίηση Kakuro ως πρόβλημα ικανοποίησης περιορισμών (CSP).

- **Σύνολο Μεταβλητών:**
  - Ο αριθμός n είναι το πλήθος των λευκών κελιών.
  - Xi είναι η τιμή που έχει κάθε λευκό κελί.
  - Sj είναι κάθε ακολουθία συνεχόμενων λευκών κελιών.
  - Hj είναι η τιμή-στόχος της ακολουθίας Sj, και αναγράφεται στην διαγώνιο.
- **Πεδίο:**
  - $Xi \in [1,9]$, $\forall i \leq n$
  - n, Hj ανήκουν στους φυσικούς

- **Σύνολο Περιορισμών:**
  - $Xi \in [1,9]$, $\forall i \leq n$
  - **Alldifferent**({Xi|$i \in Sj$}), $\forall Sj$
  - $\sum_{i \in Sj} Xi$ **=** Hj, $\forall Sj, Hj$

2. Επίλυση του προβλήματος ικανοποίησης περιορισμών που ορίσατε στο ερώτημα 1 με δύο τουλάχιστον αποδοτικούς αλγόριθμους που παρουσιάσαμε στο μάθημα.

Αρχικά να δηλώσουμε πως χρησιμοποιήθηκαν τα αρχεία:
- **Search.py**
- **Utils.py**
- **Csp.py**

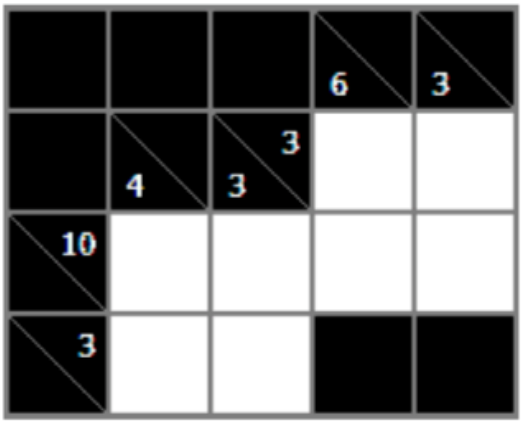
Όπως αυτά δίνονται στο github (https://github.com/aimacode).

***** Παρακαλώ τρέξτε με τα δικά μου αρχεία (Search.py, Utils.py, Csp.py). Σε κάποιο από τα τρία είχα αλλάξει μονάχα μία εντολή για να τρέξει σωστά ο κώδικας αλλά αυτή την στιγμή δεν θυμάμαι και δυστυχώς ούτε μπορώ να εντοπίσω που είχα κάνει την μοναδική αυτή αλλαγή ώστε να την επισημάνω.**

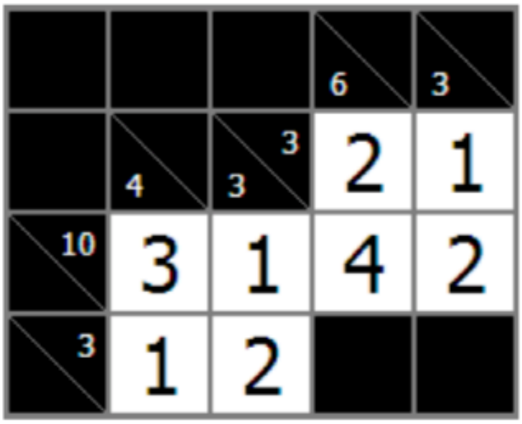
Επιπλέον δημιούργησα το αρχείο kakuro.py, το οποίο περιέχει τον δικό μου κώδικα.

Η σκέψη μου εδώ ήταν, να δοκιμάσω έναν αλγόριθμο πρώιμου ελέγχου (Forward Checking) και έναν αλγόριθμο συνέπειας ακμής (Arc Consistency), ώστε να δω και έμπρακτα την διαφορά των δύο μεθόδων. Έτσι, στην πρώτη περίπτωση κατέληξα στον FC, ενώ στη δεύτερη στον MAC. Αξίζει να σημειωθεί πως ο λόγος που δεν επέλεξα τον BJ, είναι ότι στις σημειώσεις διάβασα πως ο ίδιος είναι περιττός σε μια αναζήτηση που χρησιμοποιεί τον FC ή έναν ισχυρότερο αλγόριθμο διάδοσης περιορισμών όπως ο MAC, γιατί κάθε κλαδί του δέντρου αναζήτησης που κλαδεύεται από τον BJ κλαδεύεται και από τον FC. Αξίζει επίσης να σημειωθεί πως όπως αναφέρεται, πειραματικά αποτελέσματα έχουν δείξει ότι στις περισσότερες περιπτώσεις ένας καλός αλγόριθμος διάδοσης περιορισμών (όπως ο MAC ή ο FC) σε συνδυασμό με ένα καλό σύνολο ευρετικών μηχανισμών (όπως ο MRV και ο LCV) μπορεί να λύσει δύσκολα CSP με επιτυχία.
Και στις δύο περιπτώσεις ο ευρετικός μηχανισμός που χρησιμοποιήθηκε είναι ο συνδυασμός MRV και LCV.
Εξάλλου, Ο MRV είναι συνήθως πιο ισχυρός από τον degree heuristic. Κρίνοντας και από τις αξιολογήσεις στις διαφάνειες, ο συνδυασμός FC+MRV, μπορεί να δώσει πάρα πολύ καλά αποτελέσματα.

Το ενδεικτικό Kakuro Puzzle που δοκίμασα να λύσω για την επίδειξη των δύο αλγορίθμων είναι το εικονιζόμενο:



Αρχική μορφή πάζλ



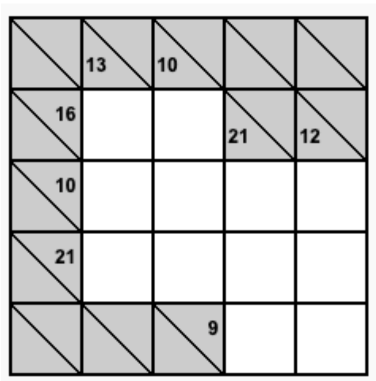
Τελική μορφή πάζλ

Στο ενλόγω puzzle ονοματίζουμε τα λευκά κουτάκια γραμμή γραμμή με αριθμούς από το 1 ως και το 8. Επομένως το κουτάκι νο1 έχει τιμή 2, το κουτάκι νο2 έχει τιμή 1, το νο3 έχει τιμή 3, το νο4 έχει τιμή 1 κοκ.

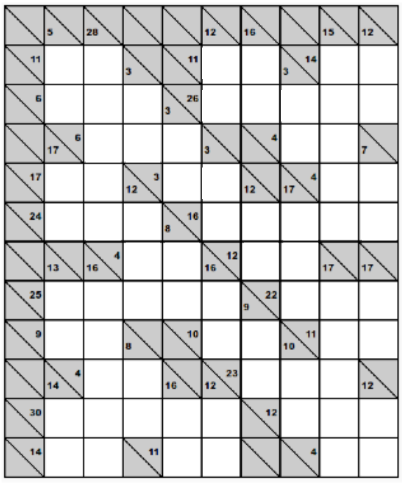
3.Συγκρίσεις

Αρχικά θα πρέπει να σημειωθεί πως το puzzle 1 των δοκιμών είναι το πιο πάνω puzzle που δόθηκε και στην εκφώνηση. Όλα τα υπόλοιπα στην αρχή επιχείρησα να τα πάρω από τηνιστοσελίδα https://www.kakuroconquest.com. Επειδή όμως χρειαζότανε extra χρόνος για να μετατρέψω κάθε puzzle σε λίστα λιστών ώστε να είναι επεξεργάσιμα τα δεδομένα από τον κώδικα, τελικά έψαξα και βρήκα στο github έτοιμες λίστες λιστών που συμβολίζουν τα ακόλουθα kakuro puzzles:

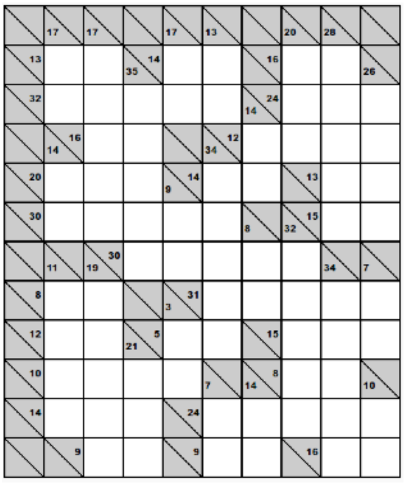
Puzzle 2 - Hard 4x4



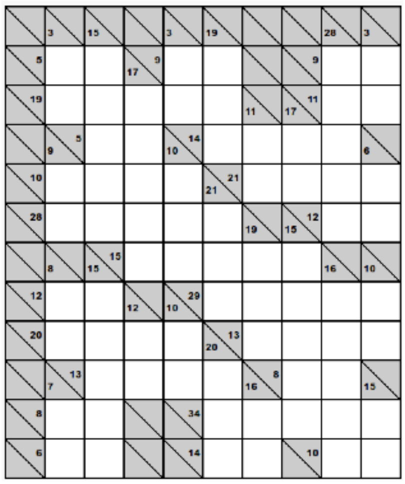
Puzzle 3: Easy 9x11



Puzzle 4: Intermediate 9x12



Puzzle 5: Expert 9x11



Η αξιολόγηση των αλγορίθμων υπαναχώρησης εξαρτάται από τον χρόνο εκτέλεσης και το πλήθος των κόμβων που επισκεπτόμαστε στο δέντρο αναζήτησης. Επομένως, με βάση αυτό το σκεπτικό, σχεδίασα έναν μικρό αλγόριθμο που επιλέγει με τη σειρά κάθε ένα από τα παραπάνω puzzles και το λύνει από δέκα φορές με κάθε μια από τις δυο μεθόδους (MAC+MRV+LCV, FC+MRV+LCV). Στο τέλος αυτών των 10 φορών κρατάει έναν μέσο όρο τόσο του χρόνου που χρειάστηκε όσο και του πλήθους των αναθέσεων(όπως αυτές υπολογίζονται από το CSP.py που πήρα έτοιμο από το github που προτείνεται στην εκφώνηση). Έτσι, τελικά, προέκυψαν οι ακόλουθες μετρήσεις:

Απαιτούμενος Χρόνος:

Αλγόριθμος	Εκφώνησης 3x4	Expert 4x4	Easy 9x11	Intermediate 9x11	Expert 9x11
MAC + MRV + LCV	24.5182 ms	41.6645 ms	889.2107 ms	3097.9492 ms	1751.8843 ms
FC + MRV + LCV	2.6245 ms	2.8315 ms	22.0604 ms	17.2995 ms	17.5268 ms

Αριθμός Αναθέσεων:

Αλγόριθμος	Εκφώνησης 3x4	Expert 4x4	Easy 9x11	Intermediate 9x11	Expert 9x11
MAC + MRV + LCV	15	22	129	165	132
FC + MRV + LCV	30	42	253	286	253

Παρατηρούμε ότι ο MAC είναι σε όλες τις δοκιμές φανερά πιο αργός από τον FC, την ίδια στιγμή ωστόσο είναι και σημαντικά καλύτερος ως προς το πλήθος των αναθέσεων. Κάτι τέτοιο, βέβαια, είναι πολύ λογικό αφού από την φύση του ο MAC, τείνει να απορρίπτει νωρίτερα περιττές αναζητήσεις.

Πρόβλημα 2

Ως δεδομένα από την εκφώνηση μας δίνονται οι ώρες που ο κάθε ύποπτος ξεκινάει και ολοκληρώνει την ομιλία του, οπότε έχουμε:

- **Γιάννης: 9:00 - 9:30**
- **Μαρία: 9:30 - 10:00**
- **Όλγα: 10:00 - 10:30**

ΣΗΜΕΙΩΣΗ: Έχουμε υπόψη πως σε όποιες προσθέσεις γίνονται μεταξύ ωρών και λεπτών προκύπτει πάλι κάποια ώρα. Π.χ. 11:30 (ώρα) + 70 (λεπτά) = 12:40 (ώρα).

1. Μοντελοποίηση

Μεταβλητές:

Για κάθε έναν από τους ομιλητές-υπόπτους O_i ένα σύνολο μεταβλητών που αποτελείται από:

- Τη χρονική στιγμή που ο ομιλητής αποχωρεί από τον χώρο της εκδήλωσης st_i .
- Το χρονικό διάστημα du_i που πέρασε από την αποχώρηση του ομιλητή από τον χώρο μέχρι και την επιστροφή του σε αυτόν.

Πεδίο:

Εφόσον ξέρουμε πότε ακριβώς τελειώνει η ομιλία κάθε ομιλιτή και ότι όλοι έχουν δηλώσει πως αποχώρησαν μετά την ομιλία τους, μπορούμε να ορίσουμε με ακρίβεια το πεδίο των χρονικών στιγμών που ο καθένας τους ενδέχεται να έφυγε:

- $st_1 \quad \in [9:30 ... 10:00]$
- $st_2 \quad \in [10:00 ... 11:00]$
- $st_3 \quad \in [10:30 ... 11:00]$

Η όλη διαδικασία των ομιλιών κρατάει μέχρι της 11:00. Συνεπώς αν *κάποιος έφυγε το συντομότερο δυνατό μετά την ομιλία του, δηλαδή στην περίπτωσή μας στις 9:30 και γυρίσει το αργότερο δυνατό, δηλαδή στις 11, συμπεραίνουμε πως το μέγιστο χρονικό διάστημα* du_i *που μπορεί να λείψει κάποιος είναι 90 λεπτά.*
Επιπλέον, έχοντας υπόψη πως η συντομότερη διαδρομή από τον χώρο της εκδήλωσης είναι αυτή μέχρι τα δωμάτια που διαρκεί 10 λεπτά κατ'ελάχιστο (και συνεπώς άλλα 10 λεπτά για την επιστροφή), καταλαβαίνουμε ότι το μικρότερο χρονικό διάστημα du_i *είναι τα 20 λεπτά. Οπότε:*

- $du_i \quad \in [20 ... 90]$ λεπτά.

Περιορισμοί:

Καταρχήν θα πρέπει όλοι να είναι πίσω στις 11:00, άρα:

- $11 - (st_i + du_i) \geq 0$

Τώρα για να βρούμε επιπλέον ποιος από τους υπόπτους είναι τελικά ο ένοχος, πρέπει να ικανοποιείται ο περιορισμός εκείνος που θέλει τον ύποπτο να έλλειψε για περισσότερα από 85 λεπτά (20 λεπτά ελάχιστος χρόνος να προσεγγίσει το χρηματοκιβώτιο, 45 λεπτά ελάχιστος χρόνος να το παραβιάσει, 20 λεπτά ελάχιστος χρόνος να επιστρέψει στον χώρο των ομιλιών). Άρα:

- $du_i \geq 85$

2. Η ώρα της αλήθειας…

Αν ο αστυνόμος Σιεσπής, ακολουθήσει όλα τα παραπάνω, με πολύ εύκολο τρόπο θα καταλήξει στο ότι ο ένοχος που έκλεψε το βραβείο είναι……………**.Ο……………Γιάννης!**

Και αυτό γιατί είναι ο μόνος που το σύνολο των μεταβλητών του μπορούν να ικανοποιήσουν και τους δύο περιορισμούς. Συγκεκριμένα με $du_i = 85$ που είναι το ελάχιστο για να κλέψει κάποιος, μόνο στον Γιάννη μπορεί να ισχύσει ο πρώτος περιορισμός $11-(9:30+85)\geq 0$.

3. Πρόταση μεθόθου διάδοσης περιορισμών.

Ο Σιεσπής θα μπορούσε να χρησιμοποιήσει τον αλγόριθμο FC ή τον ακόμα πιο αποδοτικό MAC, ο οποίοι θεωρούνται καλύτεροι από άλλους αλγορίθμους όπως ο BJ. Και αυτό γιατί όταν ο BJ πραγματοποιεί άλμα προς τα πίσω, όλες οι τιμές ενός πεδίου βρίσκονται σε σύγκρουση με την τρέχουσα ανάθεση, πράγμα που οι δύο πρώτες μέθοδοι θα ανακάλυπταν νωρίτερα.

Πρόβλημα 3

1. Μοντελοποίηση

Με την παραδοχή ότι κάθε ενέργεια έχει συγκεκριμένη διάρκεια da_i έχουμε:

<u>Μεταβλητές:</u>
- n εργασίες
- m μηχανές
- Για κάθε ενέργεια, A_i, ένα σύνολο μεταβλητών που αποτελείται από:
  - Την χρονική στιγμή εκκίνησης st_i
  - Τις απαιτούμενες μηχανές M_{ij}

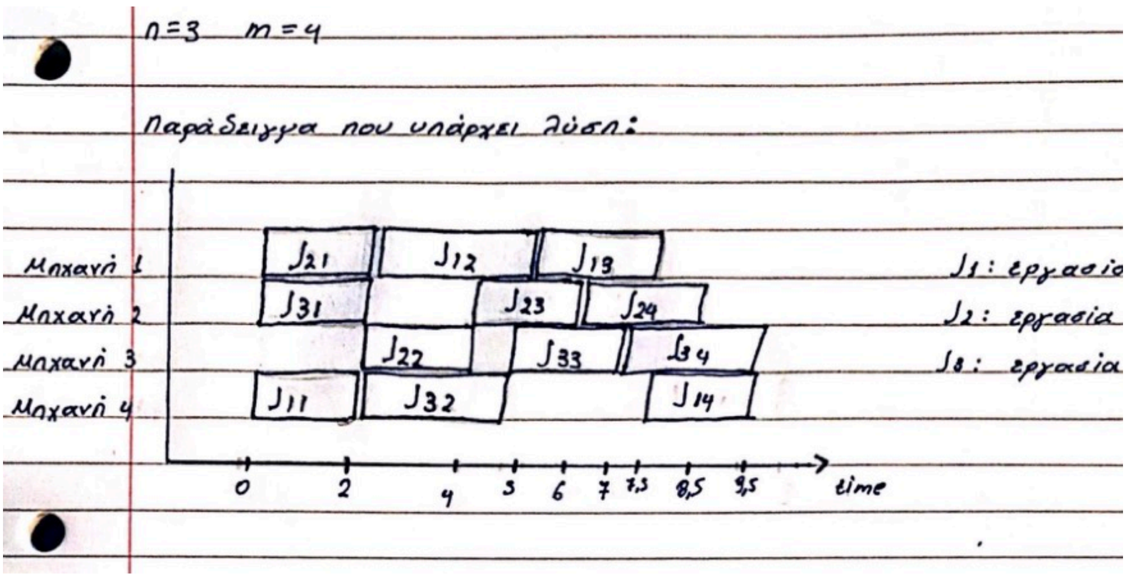
<u>Πεδίο:</u>
Το πεδίο του M_{ij} είναι το [1..m]

<u>Περιορισμοί:</u>
- Πρέπει να διασφαλίζουμε ότι κάθε ενέργεια (μιας διαδοχής ενεργειών μιας εργασίας) γίνεται αφότου έχουν τελειώσει οι προηγούμενές της. Άρα πρέπει: $st_i + du_i \leq st_j$, $1 \leq i \leq n, 1 \leq j \leq m$

- Είναι σημαντικό επίσης να μεριμνούμε ώστε αν δύο ενέργειες χρησιμοποιούν τον ίδιο πόρο να μην επικαλύπτονται χρονικά. Επομένως πρέπει: $st_i + du_i \leq st_j \vee st_j + du_j \leq st_i$, $\forall p \forall q M_{ip} \neq M_{jq}$.

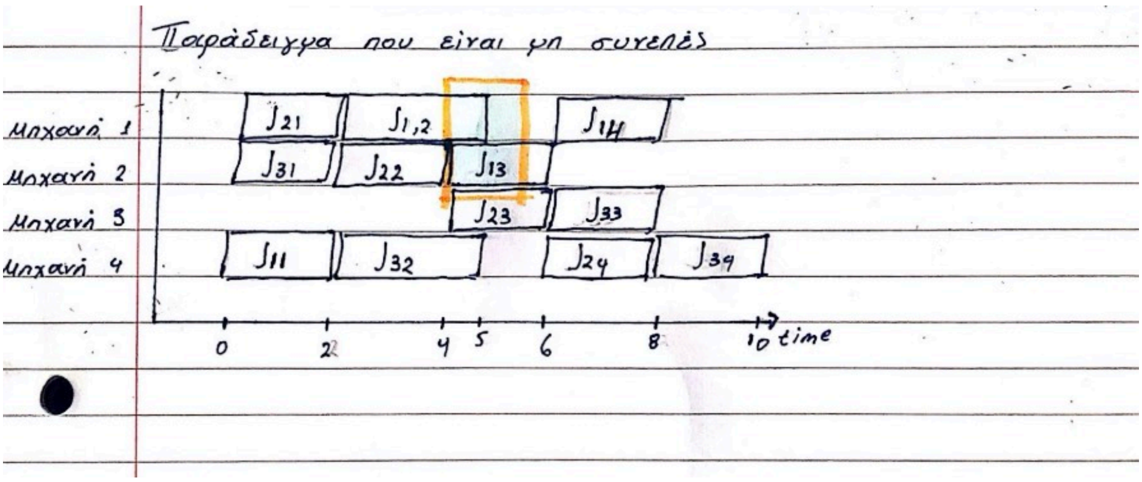
2. Παραδείγματα για n=3 και m=4

Έστω J_i, οι n εργασίες και:
- $du_{1,1}, du_{3,1}, du_{2,1}, du_{2,2}, du_{2,3}, du_{3,3}, du_{2,4}, du_{3,4}, du_{1,4}, du_{1,3} = 2s$
- $du_{1,2}, du_{3,2} = 3s$



Εδώ έχουμε μια λύση η οποία ικανοποιεί όλους μας τους περιορισμούς και τελικά όλες οι εργασίες J διεκπεραιώνονται την χρονική στιγμή t = 9,5. Να σημειωθεί φυσικά πως δεν πρόκειται για την βέλτιστη λύση, αλλά για μια εφικτή λύση:

- **Μηχανή 1**: $J_{2,1}, J_{1,2}, J_{1,3}$
- **Μηχανή 2**: $J_{3,1}, J_{2,3}, J_{2,4}$
- **Μηχανή 3**: $J_{2,2}, J_{3,3}, J_{3,4}$
- **Μηχανή 4**: $J_{1,1}, J_{3,2}, J_{1,4}$



Για τους ίδιους χρόνους των ενεργειών κάθε εργασίας που δώσαμε παραπάνω, εδώ έχουμε μια μη συνεπή λύση αφού (όπως φαίνεται και στο πορτοκαλί πλαίσιο της εικόνας) δεν ικανοποιούνται όλοι οι περιορισμοί. Συγκεκριμένα παρατηρούμε πως η ενέργεια $J_{1,3}$ ξεκινάει πριν τελειώσει η ενέργεια $J_{1,2}$, γεγονός που παραβιάζει τον πρώτο μας περιορισμό.

3. Πρόταση αλγορίθμου οπισθοδρόμησης

Ο αλγόριθμος οπισθοδρόμησης που θα πρότεινα είναι ο Forward Checking (FC). Και αυτό γιατί, κάθε φορά που ένα μηχάνημα δεσμεύεται από μια ενέργεια, ο αλγόριθμος FC ανιχνεύει τις ενέργειες που θα ήθελαν να δεσμεύσουν το συγκεκριμένο μηχάνημα μέσα στην χρονική περίοδο που αυτό έχει ήδη δεσμευτεί και τις αφαιρεί.

Πρόβλημα 4

Πρόβλημα 4

1. Ποιες από τις προτάσεις είναι έγκυρες;

Γνωρίζουμε ότι για πρόταση Φ της προτασιακής λογικής είναι έγκυρη αν για κάθε ερμηνεία I, ισχύει $I(\Phi) = true$. Οπότε για κάθε μία από τις προτάσεις θα σχηματίσουμε τους πίνακες αλήθειας.

α) $(A \land B \land C \Rightarrow D) \Leftrightarrow (A \Rightarrow (B \Rightarrow (C \Rightarrow D)))$

A	B	C	D	$A \land B \land C$	$A \land B \land C \Rightarrow D$	$C \Rightarrow D$	$B \Rightarrow (C \Rightarrow D)$	$A \Rightarrow (B \Rightarrow (C \Rightarrow D))$	α)
A	A	A	A	A	A	A	A	A	A
A	A	A	Ψ	A	Ψ	Ψ	Ψ	Ψ	A
A	A	Ψ	A	Ψ	A	A	A	A	A
A	Ψ	A	A	Ψ	A	A	A	A	A
Ψ	A	A	A	Ψ	A	A	A	A	A
A	A	Ψ	Ψ	Ψ	A	A	A	A	A
A	Ψ	A	Ψ	Ψ	A	Ψ	A	A	A
Ψ	A	A	Ψ	Ψ	A	Ψ	Ψ	A	A
A	Ψ	Ψ	A	Ψ	A	A	A	A	A
Ψ	A	Ψ	A	Ψ	A	A	A	A	A
Ψ	Ψ	A	A	Ψ	A	A	A	A	A
A	Ψ	Ψ	Ψ	Ψ	A	A	A	A	A
Ψ	Ψ	A	Ψ	Ψ	A	Ψ	A	A	A
Ψ	A	Ψ	Ψ	Ψ	A	A	A	A	A
Ψ	Ψ	Ψ	A	Ψ	A	A	A	A	A
Ψ	Ψ	Ψ	Ψ	Ψ	A	A	A	A	A

Άρα η α) είναι έγκυρη.

β) $A \wedge (A \Rightarrow B) \wedge (A \Rightarrow \neg B)$

A	B	$\neg B$	$A \Rightarrow B$	$A \wedge (A \Rightarrow B)$	$A \Rightarrow \neg B$	β)
Α	Α	Ψ	Α	Α	Ψ	Ψ
Α	Ψ	Α	Ψ	Ψ	Α	Ψ
Ψ	Α	Ψ	Α	Ψ	Α	Ψ
Ψ	Ψ	Α	Α	Ψ	Α	Ψ

Άρα η β) δεν είναι έγκυρη.

γ) $(A \vee B) \wedge (\neg A \vee C) \wedge \neg B \wedge \neg C$

A	B	C	$\neg A$	$\neg B$	$\neg C$	$A \vee B$	$\neg A \vee C$	$(A \vee B) \wedge (\neg A \vee C)$	γ)	$\neg B \wedge \neg C$
Α	Α	Α	Ψ	Ψ	Ψ	Α	Α	Α	Ψ	Ψ
Α	Α	Ψ	Ψ	Ψ	Α	Α	Ψ	Ψ	Ψ	Ψ
Α	Ψ	Α	Ψ	Α	Ψ	Α	Α	Α	Ψ	Ψ
Ψ	Α	Α	Α	Ψ	Ψ	Α	Α	Α	Ψ	Ψ
Α	Ψ	Ψ	Ψ	Α	Α	Α	Ψ	Ψ	Ψ	Α
Ψ	Α	Ψ	Α	Ψ	Α	Α	Α	Α	Ψ	Ψ
Ψ	Ψ	Α	Α	Α	Ψ	Ψ	Α	Ψ	Ψ	Ψ
Ψ	Ψ	Ψ	Α	Α	Α	Ψ	Α	Ψ	Ψ	Α

Άρα η γ) δεν είναι έγκυρη

δ) $(A \vee B) \wedge (\neg A \vee C) \wedge (B \vee C)$

Εδώ συμπεραίνουμε εύκολα πως η δ, δεν είναι έγκυρη και αυτό γιατί $(B \vee C) = \neg(\neg B \wedge \neg C)$, άρα έχει τον ίδιο πίνακα αλήθειας με την γ) αν στην τελευταία στήλη αντικαταστήσουμε με τα συμπληρωματικά. Και πάλι θα παρατηρήσουμε πως η τελική πρόταση δεν είναι πάντα αληθής.

<u>Τελικά</u> έγκυρη είναι μόνο η πρόταση α).

2. Ποιες από τις προτάσεις είναι ικανοποιήσιμες;

Για να είναι ικανοποιήσιμες, θα πρέπει να μην είναι αντίφαση, δηλαδή πάντα ψευδείς για κάθε αποτίμηση. Έτσι βασισμένοι στους πίνακες αλήθειας έχουμε ότι ικανοποιήσιμες είναι οι: $\boxed{\alpha), \gamma), \delta)}$.

3. Ποιες από τις προτάσεις είναι μη ικανοποιήσιμες;

Γνωρίζουμε ότι μια πρόταση φ της προτασιακής λογικής είναι μη ικανοποιήσιμη αν δεν υπάρχει καμία ερμηνεία I τέτοια ώστε $I(\varphi) = true$.
Επομένως, με βάση τους πίνακες αλήθειας του ερωτήματος 1, μη ικανοποιήσιμη είναι η πρόταση $\boxed{\beta)}$.

Με τη χρήση του κανόνα της ανάλυσης φτάνουμε σε αντίφαση:

$A \land (A \Rightarrow B) \land (A \Rightarrow \neg B) \equiv$
$A \land (\neg A \lor B) \land (\neg A \lor \neg B) \equiv$
$((A \land \neg A) \lor (A \land B)) \land (\neg A \lor \neg B) \equiv$
$(A \land B) \land \neg (A \land B) \rightsquigarrow$ αντίφαση, άρα επαληθεύεται η μη ικανοποιησιμότητα.

4. <u>Ποιες από τις προτάσεις έχουν τουλάχιστον ένα μοντέλο;</u>

Από τη θεωρία έχουμε ότι αν φ μια πρόταση της προτασιακής λογικής τότε αν I είναι μια ερμηνεία τέτοια ώστε $I(\varphi) = true$, τότε λέμε ότι η I ικανοποιεί τη φ ή ότι η I είναι ένα μοντέλο της φ.

Εμείς, ερμηνείες όπου να δίνουν αποτίμηση true -όπως φαίνεται στους πίνακες αλήθειας- έχουμε βρει στις προτάσεις α), γ), δ).

Συνεπώς τουλάχιστον ένα μοντέλο έχουν οι: $\boxed{α), γ), δ)}$

5. <u>Ποιες από τις προτάσεις είναι ταυτολογίες;</u>

Οι έγκυρες προτάσεις λέγονται και ταυτολογίες, άρα ταυτολογίες είναι οι: μόνο η $\boxed{α)}$

6.Ποιες από τις προτάσεις είναι σε μορφή Horn;

Σύμφωνα με τη θεωρία, *μια φράση Horn είναι μια διάζευξη λεκτικών από τα οποία το πολύ ένα είναι θετικό.*

Εδώ, παρατηρούμε πως καμία από τις προτάσεις, τουλάχιστον στην συγκεκριμένη τους μορφή δεν ικανοποιούν το πάνω κριτήριο. Φυσικά, κάποιες από αυτές ίσως να μπορούσαν με την κατάλληλη μεταροπή να το ικανοποιήσουν, όμως έπειτα από σχετική ερώτηση στο piazza και την αντίστοιχη διευκρίνηση από τους βοηθούς κατάλαβα ότι δεν πρέπει να τις μετατρέψουμε.

Πρόβλημα 5

Πρόβλημα 5

Θα βασιστούμε στο Θεώρημα που αναφέρει:

[Έστω φ και ψ προτάσεις της προτασιακής λογικής. Τότε $\varphi \models \psi$ ανν $\varphi \land \neg\psi$ είναι μη ικανοποιήσιμη.]

Επομένως, θα προσπαθήσουμε να δείξουμε ότι για $\varphi \equiv A \land (B \Leftrightarrow C)$ και $\psi \equiv (A \land B) \Leftrightarrow (A \land C)$ ισχύει ότι η πρόταση $\overset{(1)}{A \land (B \Leftrightarrow C)} \land \overset{(2)}{\neg((A \land B) \Leftrightarrow (A \land C))}$ είναι μη ικανοποιήσιμη.

▸ Αρχικά θα μετατρέψουμε κάθε μία από τις (1), (2) σε CNF

• $A \land (B \Leftrightarrow C) \equiv$

$A \land ((B \Rightarrow C) \land (C \Rightarrow B)) \equiv$

$A \land ((\neg B \lor C) \land (\neg C \lor B))$

• $\neg((A \land B) \Leftrightarrow (A \land C)) \equiv$

$\neg(((A \land B) \Rightarrow (A \land C)) \land ((A \land C) \Rightarrow (A \land B))) \equiv$

$\neg((\neg(A \land B) \lor (A \land C)) \land (\neg(A \land C) \lor (A \land B))) \equiv$

$(\neg(\neg(A \land B) \lor (A \land C)) \lor \neg(\neg(A \land C) \lor (A \land B))) \equiv$

$(((A \land B) \land \neg(A \land C)) \lor ((A \land C) \land \neg(A \land B))) \equiv$

$(((A \land B) \land (\neg A \lor \neg C)) \lor ((A \land C) \land (\neg A \lor \neg B))) \equiv$

$((((A \land B) \lor (A \land C)) \land ((A \land B) \lor (\neg A \lor \neg B))) \lor ((\neg A \lor \neg C) \lor (A \land C)) \land ((\neg A \lor \neg C) \lor (\neg A \lor \neg B))) \equiv$

$(((A\vee A)\wedge(A\vee C)\wedge(B\vee A)\wedge(B\vee C))\wedge((A\vee\neg A\vee\neg B)\wedge(B\vee\neg A\vee\neg B))\wedge$

$\wedge((\neg A\vee\neg C\vee A)\wedge(\neg A\vee\neg C\vee C))\wedge(\neg A\vee\neg C\vee\neg A\vee\neg B)\equiv$

$((A\wedge(A\vee C)\wedge(B\vee A)\wedge(B\vee C))\wedge((A\vee\neg A\vee\neg B)\wedge(B\vee\neg A\vee\neg B))\wedge$

$\wedge((\neg A\vee\neg C\vee A)\wedge(\neg A\vee\neg C\vee C))\wedge(\neg A\vee\neg B\vee\neg C)$

Και τελικά έχουμε τις ακόλουθες φράσεις:

(3) A	(7) $A\vee B$	(11) $\neg A\vee\neg C\vee A$
(4) $\neg B\vee C$	(8) $B\vee C$	(12) $\neg A\vee\neg C\vee C$
(5) $\neg C\vee B$	(9) $A\vee\neg A\vee\neg B$	(13) $\neg A\vee\neg B\vee\neg C$
(6) $A\vee C$	(10) $B\vee\neg A\vee\neg B$	

Από τη φράση (10) $B\vee\neg A\vee\neg B$ και την (12) $\neg A\vee\neg C\vee C$ με τον κανόνα της ανάλυσης μπορούμε να συμπεράνουμε την:

$$\neg A\vee B\vee C\vee\neg B\vee\neg C$$

Ομοίως, από την $\neg A\vee B\vee C\vee\neg B\vee\neg C$ που προέκυψε πάνω και την (4) $\neg B\vee C$ συμπεραίνουμε την:

$$\neg A\vee\neg B\vee C$$

Και από την $\neg A\vee\neg B\vee C$ και την (5) $\neg C\vee B$ συμπεραίνουμε:

$$\neg A$$

Όμως από τα λεκτικά $\neg A$ και (3) A έχουμε την κενή φράση δηλαδή αντίφαση.

Επομένως η αρχική μας πρόταση: $A\wedge(B\Leftrightarrow C)\wedge\neg((A\wedge B)\Leftrightarrow(A\wedge C))$ είναι μη ικανοποιήσιμη και συνεπώς η πρόταση $A\wedge(B\Leftrightarrow C)$ καλύπτει λογικά την $(A\wedge B)\Leftrightarrow(A\wedge C)$.

Πρόβλημα 6

Το συντακτικό μιας γλώσσας αναπαράστασης γνώσης είναι αυτό που καθορίζει με ακρίβεια τους καλά ορισμένους τύπους και προτάσεις της γλώσσας. Επομένως, το συντακτίκο είναι αυτό που θα βασιστούμε για να επιλέξουμε ποιες από τις εκφράσεις είναι καλά ορισμένες. Σύμφωνα, λοιπόν, με αυτό, οι λογικοί σύνδεσμοι που μπορούν να χρησιμοποιύνται είναι οι:

$$\neg, \wedge, \vee, \Rightarrow \text{ και } \Leftrightarrow.$$

Οπότε έχουμε:

- (A) ✓
- $(A \to B)$ ✗
- $A \equiv B$ ✗
- $A \models B$ ✗
- $(A \wedge 1)$ ✓

Σημείωση: *Εδώ ο λόγος που δεν θεωρήσαμε την έκφραση καλά ορισμένη είναι ότι χρησιμοποιείται βελάκι αντί*

Σημείωση: Με συγχωρείτε πολύ για αυτό το υβρίδιο που παραδίδω, που τα μισά είναι χειρόγραφα και τα μισά στον υπολογιστή αλλά δυστυχώς λόγω του αυξημένου φόρτου δεν είχα χρόνο για να τα ψηφιοποιήσω όλα. Πιστεύω τουλάχιστον πως σαν λύσεις είναι αξιοπρεπείς.