ΠΡΟΗΓΜΕΝΗ *ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ*

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΊΑ

Νικόλαος Μαρούλης
AM: M1605
5/9/2019

Η εργασία αναπτύχθηκε στην γλώσσα προγραμματισμού Minizinc, καθώς είχα και στο παρελθόν μερική εμπειρία με το συντακτικό της συγκεκριμένης γλώσσας.
Έχουν απαντηθεί όλα τα ερωτήματα, και το καθένα είναι σε διαφορετικό αρχείο με την κατάληξη .mzn.
2-fraction problem -> puzzle_2_fractions.mzn
3-fraction problem -> puzzle_3_fractions.mzn
4-fraction problem -> puzzle_4_fractions.mzn
5-fraction problem -> puzzle_5_fractions.mzn
6-fraction problem -> puzzle_6_fractions.mzn

Εκτελείται με την εντολή > minizinc [file_name], με την προϋπόθεση να είναι εγκατεστημένος ο minizinc compiler (σε ubuntu τον εγκατέστησα με την εντολή sudo snap install minizinc --classic)

Τα αποτελέσματα που έχουν βγάλει οι υλοποιήσεις μου (τα επιβεβαίωσα και με τη χρήση calculator), ικανοποιώντας όλους τους περιορισμούς, είναι:

- *2-fraction*: 9/(13) + 8/(26) = 1 | A=9, B=1, C=3, D=8, E=2, F=6
- *3-fraction*: 9/(12) + 5/(34) + 7/(68) = 1 | A=9, B=1, C=2, D=5, E=3, F=4, G=7, H=6, I=8

για τα υπόλοιπα έβαλα ονόματα μεταβλητών (X1, X2 ...), όπου X1/(X2X3) + ... + Xn-2 / (Xn-1Xn) = 1.
Επίσης ικανοποιούνται όλοι οι περιορισμοί

- *4-fraction*: 4/(15) + 9/(27) + 8/(36) + 8/(45) = 1 | X1=4, X2=1, X3=5, X4=9, X5=2, X6=7, X7=8, X8=3, X9=6, X10=8, X11=4, X1
- *5-fraction*: 4/(15) + 7/(28) + 9/(36) + 2/(36) + 8/(45) = 1 | X1=4, X2=1, X3=5, X4=7, X5=2, X6=8, X7=9, X8=3, X9=6, X10=2, X11=3, X12=6, X13=8, X14=4, X15=5
- *6-fraction*: 4/(16) + 7/(28) + 5/(39) + 1/(24) + 8/(39) + 7/(56) = 1 | X1=4, X2=1, X3=6, X4=7, X5=2, X6=8, X7=5, X8=3, X9=9, X10=1, X11=2, X12=4, X13=8, X14=3, X15=9, X16=7, X17=5, X18=6

Στις 3 τελευταίες περιπτώσεις, εμφανίζονται όλοι οι αριθμοί, και κανένας πάνω από 2 φορές.

Τα **constraints** που επέβαλα στο πρόγραμμα είναι τα εξής:
1) all_different(Vars), όπου ο πίνακας Vars έχει τις μεταβλητές. Το all_different είναι εντολή της γλώσσας Minizinc και απλά λέει όλες οι μεταβλητές να έχουν διαφορετικές τιμές.
Για τα 4/5/6-fractions χρησιμοποίησα και άλλον ένα πίνακα Vars1 με την ίδια λογική.
2) Για κάθε κλάσμα, πχ το A/BC, έθεσα μεταβλητές D1, D2 ... Dn οι οποίες δηλώνουν τους παρανομαστές, και πρέπει D1 = B*10 + C , αντίστοιχα για όλα τα κλάσματα.
3) Κανένα κλάσμα να μην είναι μεγαλύτερο ή ίσο του 1, καθώς όλα τα κλασματα είναι μη μηδενικά και μεγαλύτερα του 1. Οπότε για το πρώτο π.χ. A/D1 < 1.
4) Το πιο βασικό constraint το οποίο βγαίνει (για παράδειγμα του 3-fraction) απ'το:

> A/BC + D/EF + G/HI = 1, με D1 = BC, D2 = EF και D3 = HI
> A/D1 + D/D2 + G/D3 = 1, όλα τα μέλη επί D1*D2*D3
> **A*D2*D3 + D*D1*D3 + G*D1*D2 = D1*D2*D3**

Αυτό είναι και το τελευταίο constraint. Αξίζει να σημειωθεί ότι στις περιπτώσεις των 5 και 6 fractions. Το γινόμενο στην δεξιά πλευρά
(D1*D2*D3*D4*D5 και D1*D2*D3*D4*D5*D6 αντίστοιχα), έβγαινε πολύ μεγάλο και το πρόγραμμα έσκαγε από μνήμη, οπότε στην περίπτωση των 5-fractions, διαίρεσα και τα δύο μέλη με το D5 και στην περίπτωση των 6-fractions διαίρεσα και τα δύο μέλη με το D5*D6, ώστε να μην σκάει το πρόγραμμα.
Τέλος, καθώς η minizinc είναι γλώσσα χωρίς περίπλοκο συντακτικό, ανοίγωντας τα αρχεία, μπορείτε να διακρίνετε ό,τι υλοποιήσεις έχω κάνει, οι οποίες δεν έχουν αναλυθεί εδώ.