

"בינה מלכותית" – הצגת פרויקט**קישור לפרויקט בGithub**

יוני 2022

מטרת על: מטרתנו המרכזית היא ליצור תוכנית המקבלת חידה קריפטו-אריתמטית כלשהי, ופותרת אותה כבעיית CSP.

מטרת קובץ README: תיאור העבודה וכן את האופן שבו יש להריץ את הפרויקט.

דרך ביצוע: תחילה נגשנו להרצאה בה למדנו על אלגוריתם CSP כדי להבין כיצד האלגוריתם עובד. לאחר מעבר ראשוני על האלגוריתם – הבנו שהכי נכון עבורנו לממש באמצעות הרעיון של תכנות "מונחה עצמים".

הגדרנו מספר אובייקטים אשר ייצגו לנו בצורה פשוטה את הבעיה:

- משתנה – EncryptVar
- דומיין – Domain
- אילוץ – Constraint (מחלקה אבסטרקטית)
- אילוץ מסוג AllDifferent (יורש Constraint)
- ➡ האילוץ מוודא שאין שני משתנים אשר קיבלו את אותו הערך
- אילוץ מסוג SumEquals (יורש Constraint)
- ➡ האילוץ מוודא ש: [סכום משתני החיבור = (משתנה התוצאה + $10 * carry$)]
- אובייקט מרכזי לפתרון הבעיה - CryptArithmeticSolver

פרוט תוכן הקוד:

שם מחלקה: EncryptVar			
~ שדות ~			
סוג	שם	שימוש	
str	var	ייצוג אות מתוך החידה	
Domain	domain	ייצוג דומיין רלוונטי בהתאם למשתנה	
~ פונקציות ~			
שם הפונקציה	מקבלת	מחזירה/מעדכנת	מטרה
getVar	----	var	החזרת האות של המשתנה
getDomain	----	domain	החזרת הדומיין של המשתנה

שם מחלקה: Domain			
~ שדות ~			
סוג	שם	שימוש	
List(int)	domain	החזקת רשימת הדומיין של משתנה	
List(int)	originalDomain	שמירת רשימת הערכים החוקים למשתנה	
~ פונקציות ~			
שם הפונקציה	מקבלת	מחזירה/מעדכנת	מטרה
addToDomain	value	domain	החזרת ערך לרשימת הדומיין לאחר ביטול השמה
removeFromDomain	value	domain	מחיקת ערך מרשימת הדומיין לאחר הוספת השמה
getNextFreeDomain	--	Int	החזרת הערך הראשון ברשימת הדומיין
getDomainList	--	List(int)	החזרת רשימת הדומיין

שם מחלקה: ConstraIt			
~ שדות ~			
סוג	שם	שימוש	

~ פונקציות ~			
שם הפונקציה	מקבלת	מחזירה/מעדכנת	מטרה
isConsist	Dict(str->int)	-----	פונקציה אבסטרקטית למימוש במחלקות היורשות

שם מחלקה: SumEquals			
~ שדות ~			
סוג	שם	שימוש	
List(str)	vars	החזקת רשימת משתני החיבור הרלוונטיים לאילוץ	
Str	res	החזקת משתנה התוצאה	
Str	carry	החזקת משתנה carry	
List(str)	varList	החזקת כלל המשתנים יחד	
~ פונקציות ~			
שם הפונקציה	מקבלת	מחזירה/מעדכנת	מטרה
isConsist	Dict(str->int)	bool	בדיקה האם האילוץ סופק

שם מחלקה: AllDifferent			
~ שדות ~			
סוג	שם	שימוש	
List(str)	vars	החזקת רשימת המשתנים של הבעיה (ללא משתני carry)	
~ פונקציות ~			
שם הפונקציה	מקבלת	מחזירה/מעדכנת	מטרה
isConsist	Dict(str->int)	bool	בדיקה האם האילוץ סופק

שם מחלקה: CryptarithmicSolver			
~ שדות ~			
סוג	שם	שימוש	
Str	first	החזקת המילה הראשונה בקלט	
Str	second	החזקת המילה השניה בקלט	
str	result	החזקת המילה השלישית בקלט	
List(str)	vars	החזקת רשימת המשתנים של הבעיה (ללא משתני carry)	
Dict(str->int)	assignments	החזקת ההשמות במהלך ריצת האלגוריתם	
List(constraint)	constraints	החזקת רשימת האילוצים של החידה	
~ פונקציות ~			

שם הפונקציה	מקבלת	מחזירה/מעדכנת	מטרה
domainReduction	----	Domain וגם את assignments	צימצום דומיין של משתנים בעזרת חוקים מתמטיים. השמת משתניים במידה ואפשרי.
getVars	---	Dict(str->EncryptVar)	החזרת משתני הקלט
getDomain	var	Domain	החזרת דומיין של משתנה var
getConstraints	----	List(constraint)	החזרת רשימת האילוצים של הקלט
isComplete	Dict(str->int)	bool	בדיקה האם יש השמה לכלל המשתנים
backtracking	Dict(str->int)	Int או Dict(str->int)	ביצוע אלגוריתם backtracking על הקלט
checkConsistency	Dict(str->int), str, int	bool	בדיקה האם ההשמה החדשה מספקת את כלל האילוצים
getUnassignedVar	Dict(str->int)	List(str)	החזרת רשימת המשתנים שטרם קיבלו השמה
sortByMRV	List(str)	List(str)	מיון רשימת המשתנים שטרם קיבלו השמה ע"פ יוריסטיקת MRV
updateDomains	Str, int	EncryptVar .domain	צמצום דומיין של המשתנים לאחר השמה חדשה
cancelDomains	Str, int	EncryptVar .domain	ביטול צמצום דומיין של המשתנים לאחר ביטול השמה

תוצר מוגמר:

לאחר סיום כתיבת הקוד החלטנו להוסיף GUI כדי לשפר את נראות הפרויקט, ולהנגיש את הכנסת הקלט והצגת התוצאה בצורה ברורה למשתמש.

כמו כן ב-GUI הוספנו את ההגדרות הבסיסיות של השאלה עם הגבלות תואמות:

- לא ניתן להכניס מספרים בקלט
- לא ניתן להכניס מעל 10 תווים שונים
- לא ניתן להכניס קלט שגודל התוצאה קטן מגודל משתני החיבור

סיכום אישי:

נהנינו לעבוד על הפרויקט, מימנו למדנו טוב יותר לעבוד עם שפת פייתון ובנוסף ירדנו לרזולוציה של הבנה עמוקה יותר בנוגע לאלגוריתם CSP וכך התחדדנו כיצד לממש בפרויקט גמר הכללי של התואר.

בנוסף חשוב לנו לציין:

לאחר סיום אלגוריתם backtracking הבסיסי היו קלטים שמשך השערוך שלהם היה כ-40 שניות. לאחר מימוש יוריסטיקת MRV צומצם זמן השערוך לכ-3 שניות במקסימום!! מכך למדנו עד כמה יורסטיקה משפיעה בפתירת בעיות מסוג זה.

בברכה,

פינחס זיו, אלכס חן