הסקה אוטומטית ושימושיה - 2024

תרגיל בית 1

נתונים טכניים

1. תאריך פרסום התרגיל: 7 בינואר 2024.
2. תאריך הגשת התרגיל: 21 בינואר 2024.
3. מומלץ להגיש בזוגות, אך אין חובה לעשות זאת.
4. יש להגיש את התרגיל דרך מערכת הסאבמיט. ההגשה צריכה לכלול קובץ זיפ עם הדברים הבאים:
   (א) קובץ פייתון בשם $install\_bool.py$ שמכיל את הפתרון לשאלה 1.
   (ב) קובץ cnf שמכיל את הפתרון לשאלה 2ב.
   (ג) קובץ pdf שיכלול שמות, תעודות זהות, ותשובות לשאלות.
5. אשמח אם תשאלו שאלות בפורום הקורס במודל (וגם אם תענו, אך מבלי לגלות את התשובות לשאלות שבתרגיל).
6. תוכלו להשתמש בפורום גם למציאת בן/בת זוג להגשה.

הכנה לתרגיל

1. מומלץ מאוד לעבוד בסביבת לינוקס. למשתמשים בווינדוז, אפשר להתקין WSL. למשתמשים במק -- כנראה שתסתדרו משום שהיא מבוססת על מערכת דומה ללינוקס. אפשרות נוספת היא להשתמש ב-$docker$ או $virtual-box$.
2. יש להתקין $sat-solver$ כלשהו, למשל $minisat$. הוראות להתקנת $minisat$ מצויות כאן: `http://minisat.se/MiniSat.html`.
3. יש להתקין ספריית פייתון בשם $pysmt$: $pip\,install\,pysmt$.
4. כדי לוודא שהספרייה מותקנת, יש לכתוב קובץ פייתון עם שורה אחת, להריץ אותו, ולוודא שאין שגיאות. השורה היא:

```
import pysmt
```

5. בהינתן ש-$pysmt$ מותקן, יש להריץ את הפקודה הבאה:

```
pysmt-install --z3
```

לאחר מכן יש להריץ

```
pysmt-install --env
```

תקבלו הדפסה של פקודה אותה יש להעתיק, להדביק ולהריץ. הפקודה משנה את משתנה המערכת $PYTHONPATH$ כדי ש-$z3$ יהיה זמין. יש לעשות זאת מחדש בכל פעם שפותחים חלון טרמינל.

6. כדי לוודא שהספרייה מותקנת ועובדת יחד עם $z3$, יש לכתוב קובץ פייתון ולהריץ אותו, ולוודא שאין שגיאות. תוכן הקובץ הוא:

```
from pysmt import Solver
a = Solver("z3")
```

7. לקריאה נוספת:

(א) להריץ

```
pysmt-install --help
```

(ב) מידע על התקנת ספריות פייתון:
https://packaging.python.org/en/latest/tutorials/installing-packages/

(ג) הקוד של $pysmt$: https://github.com/pysmt/pysmt

(ד) הדוקיומנטציה של $pysmt$: https://pysmt.readthedocs.io/en/latest/

תרגיל

1. כתבו תכנית פייתון שמקבלת כקלט בעיית התקנה וקובעת האם יש תכנית התקנה שמתאימה לה. במידה ויש, על התכנית להציג אחת כזו. פרטים מלאים ודוגמאות זמינים כאן:

https://github.com/yoni206/ar-class-2023-hw1.

2. כתבו תכנית נוספת שמקבלת כקלט בעיית התקנה ומספר חיובי k, וקובעת האם יש לפחות k תכניות התקנה ששונות זו מזו שמתאימות לה. במידה ויש, על התכנית להציג את כולן בזו אחר זו.

רמז: בשאלה הקודמת זה היה מספיק לבקש מהסולבר לפתור בעיה אחת. כאן יש צורך לפתור k בעיות בזו אחר זו.

3. הביטו בנוסחאות הבאות:

$$\varphi_1 = (a \land \neg c) \lor (f \to (h \lor \neg a))$$

$$\varphi_2 = ((a \land f) \lor (\neg a \to (b \land g)))$$

$$\varphi = \varphi_1 \leftrightarrow \varphi_2$$

(א) רשמו נוסחת cnf שמסתפקת ביחד עם φ לפי האלגוריתם של צייטין. נקרא לנוסחה שהתקבלה A.

(ב) רשמו נוסחת cnf ששקולה ל-φ לפי האלגוריתם הנאיבי שתיארנו באחת ההוכחות בכיתה. נקרא לנוסחה שהתקבלה B.

הערה: האלגוריתם הנאיבי למציאת cnf שקולה מתבסס על הפעלת השקילויות הבאות:

$$\begin{array}{ccc}
A \to B & \equiv & \neg A \lor B \\
A \leftrightarrow B & \equiv & (\neg A \lor B) \land (\neg B \lor A) \\
\neg\neg A & \equiv & A \\
\neg(A \lor B) & \equiv & \neg A \land \neg B \\
\neg(A \land B) & \equiv & \neg A \lor \neg B \\
A \lor (B \land C) & \equiv & (A \lor B) \land (A \lor C)
\end{array}$$

(ג) צרו קבצי cnf שמייצגים את הנוסחאות שרשמתם.

(ד) הריצו את $minisat$ על הקבצעים שקיבלתם.

i. מה התוצאות שהתקבלו?

ii. מה הן אומרות לגבי φ_1 ו-φ_2?

iii. האם A ו-B שקולות? נמקו.

iv. האם A ו-B מסתפקות יחד? נמקו.

4. נביט באלגוריתם של צייטין (מובא להלן). הוכיחו כי A ספיקה אם ורק אם B ספיקה. עשינו זאת באופן חלקי בכיתה, אך השארנו הרבה מקרים ללא הוכחה, בטענה שהם דומים. כתבו הוכחה שכוללת את כל המקרים.

לנוחותכם, להלן ההגדרה המלאה של שיטת צייטין. תהי נוסחה A. ניצור נוסחה B כדלהלן:
עבור כל תת נוסחה C של A שאינה משתנה נגדיר משתנה חדש p_C. נגדיר את B להיות הנוסחה $p_A \wedge \bigwedge_{\{C\in sub(A)\}} E(C)$,
כאשר $E(C)$ מוגדרת כך:

$$E(C) = \begin{cases} CNF(p_C \leftrightarrow C) & C\ is\ variable \\ CNF(p_C \leftrightarrow true) & C\ is\ true \\ CNF(p_C \leftrightarrow false) & C\ is\ false \\ CNF(p_C \leftrightarrow \neg p_D) & C = \neg D \\ CNF(p_C \leftrightarrow (p_{C_1} \wedge p_{C_2})) & C = C_1 \wedge C_2 \\ CNF(p_C \leftrightarrow (p_{C_1} \vee p_{C_2})) & C = C_1 \vee C_2 \\ CNF(p_C \leftrightarrow (p_{C_1} \rightarrow p_{C_2})) & C = C_1 \rightarrow C_2 \\ CNF(p_C \leftrightarrow (p_{C_1} \leftrightarrow p_{C_2})) & C = C_1 \leftrightarrow C_2 \end{cases}$$

$$CNF(p \leftrightarrow C) = \begin{cases} (\neg p_C \vee C) \wedge (\neg C \vee p_C) & C\ is\ variable \\ (\neg p_C \vee true) \wedge (false \vee p_C) & C\ is\ true \\ (\neg p_C \vee false) \wedge (true \vee p_C) & C\ is\ false \\ (\neg p_C \vee \neg P_D) \wedge (p_D \vee p_C) & C\ is\ \neg D \\ (\neg p_C \vee p_{C_1}) \wedge (\neg p_C \vee p_{C_2}) \wedge (\neg p_{C_1} \vee \neg p_{C_2} \vee p_C) & C\ is\ C_1 \wedge C_2 \\ (\neg p_C \vee p_{C_1} \vee p_{C_2}) \wedge (\neg p_{C_1} \vee p_C) \wedge (\neg p_{C_2} \vee p_C) & C\ is\ C_1 \vee C_2 \\ (\neg p_{\{C\}} \vee \neg p_{C_1} \vee p_{C_2}) \wedge (p_{C_1} \vee p_C) \wedge (\neg p_{C_2} \vee p_C) & C\ is\ C_1 \rightarrow C_2 \\ (\neg p_C \vee \neg p_{C_1} \vee p_{C_2}) \wedge (\neg p_C \vee p_{C_1} \vee \neg p_{C_2}) \wedge & C\ is\ C_1 \leftrightarrow C_2 \\ (p_C \vee \neg p_{C_1} \vee \neg p_{C_2}) \wedge (p_C \vee p_{C_1} \vee p_{C_2}) & \end{cases}$$