**The 8000th Busy Beaver number:**

מאמרם של סקוט ארונסון ואדם ידידיה עוסק ב"התחמקותו" של המספר ה-8000 בבעיית הבונה העסוק מתאוריית ZF של תורת הקבוצות.  
ננסה להבין קודם כל את התאוריה ומשם נבין איך (ולמה דווקא) המספר ה8000 בבעיה שלנו מתחמק מקיומה.  
בתחילת המאמר, סקוט מביע את הערכתו הרבה לפרויקט מוצלח של אדם ידידיה, סטודנט לדוקטורט בMIT שהצליח לבנות מכונת טיורינג בעלת בעלת סרט אחד ושני תווים בעלת לא פחות מ-7918 מצבים, שהתנהגותה בהרצה על סרט ריק, לא ניתנת להוכחה מהאקסיומות הקלאסיות של תורת הקבוצות.

הצהרה מדויקת לתוצאה העיקרית היא,  
יש לנו מכונת טיורינג בעלת 7918 מצבים בשם Z, כך ש:  
1. Z פועלת תמידית, בהנחה שישנה עקביות של התאוריה הקרדינלית SRP.

2. לא ניתן להוכיח שZ רצה לעולם בZFC, בהנחה שתאוריה זו עקבית.

**רקע:**  
כתוצאה מיידית [מתאוריית אי השלמות של גֶדֶל](#תאוריית_אי_השלמות_של_גֶדֶל), נובע שישנה תכנית מחשב כלשהיא באורך מסוים, ה"חומקת" מכוחה של המתמטיקה הרגילה להוכיח מה היא עושה, כאשר היא רצה עם כמות זיכרון בלתי מוגבלת.  
לדוגמא, תכנית כזו יכלה למנות את כל ההשלכות האפשריות של אקסיומות ZFC זו אחר זו ולעצור אם תמצא סתירה כלשהיא.  
בהנחה שZFC עקבית, התכנית אמורה לרוץ לנצח, אך מצד שני ZFC **לא יכולה להוכיח** שהתכנית רצה לנצח, כי אם היא הייתה עושה זאת, היא הייתה מוכיחה את העקביות **שלה**, ומפרה בכך את משפט אי השלמות השני.  
  
לצערנו, דיון זה משאיר אותנו בחשיכה לגבי המקום בחלל תוכניות המחשב בו   
"הגרמלין הגֶדֶלי" מרים את ראשו הבלתי ניתן להכרעה.  
תכנית שתחפש אי עקביות כלשהיא בZFC לבטח תהיה יצור די מסובך.. נסביר מדוע.  
יהיה עליה לקודד לא רק את סכמת האקסיומה של ZFC, אלא בנוסף גם את השפה וכללי הסקה מהמעלה הראשונה..  
בשפת תכנות קלאסית (c, java ..), תכנית בסגנון זה עשויה להיות בעלת אלפי שורות/הוראות אם מדובר בשפת מכונה אלמנטרית.  
כנראה מעולם לא נתקלנו ולא ניתקל בתכנית כזו, גם אם היה לנו מחשב מוטרף בגודלו של היקום והיינו מריצים תכנית אחר תכנית ברנדומליות שיטתית של עכבר מעבדה.

מכאן, השאלה החשובה במקומה עומדת, שאלה שבודדים שאלו עד כה בהיסטוריה:  
האם האקסיומות של תורת הקבוצות מספיקות בכדי לנתח לנתח את ההתנהגות של כל תוכנית מחשב שאורכה לכל היותר, נניח, 50 הוראות מכונה? או שיש תוכניות סופר קצרות *שכבר מציגות "התנהגות גֶדֶלית"*?

מדעני המחשב התיאורטיים עשויים להתנגד לכך שזו "רק שאלה של קבועים".   
ובכן בסדר, כן, אבל גם מקור החיים ביקום שלנו - פאזל לא לגמרי לא קשור לנושא אך בהחלט מהותי - הוא גם "רק שאלה של קבועים"!  
ביתר פירוט, אנו יודעים שאפשר בחוקי הפיזיקה הנוכחיים שלנו לבנות מכונה המשכפלת את עצמה: לדוגמא, DNA או RNA והאביזרים הנלווים להם.  
אנו גם יודעים שמולקולות זעירות כמו H2O ו- CO2 אינן משכפלות את עצמן,  
אבל אנחנו לא יודעים **כמה קטנה** יכולה להיות המולקולה הקטנה ביותר המשכפלת את עצמה - וזה נושא שמשפיע אם עלינו לצפות *למציאת חיים כלשהם ביקום אי פעם* או שהתכבדנו להיות רק אנחנו לבד.

מה אומרים בעצם חוקי הפיזיקה או בכלל, חוקי הטבע?

מדעי הטבע הם מעשה ידי אדם, נועדו בשביל לנסות להבין את דפוסי ההתנהגות בעולמנו.

לכן, יש פיתוח מודלים בהם אנחנו לא מגלים חוקים, אלא מנסחים אותם.

נקודת ההתחלה של כל חוק מתחילה בהנחת יסוד שלא תמיד (לפעמים בהכרח) לא תואמת את המציאות, לדוגמא:

העיקרון הקוסמולוגי - קובע כי היקום כולו איזוטרופי (אין בו כיוון ייחודי ולכן יש בו סימטריה) והומוגני, אחיד בכל מקום ברמת הגלקסיות.

אך כיום ידוע שהיקום לא הומוגני כלל ועיקר, אלא כל גלקסיה עומדת בפני עצמה במרחקים עצומים זו מזו ללא אחידות כלשהיא.

אז איך בוחרים מודלים?

מצד אחד אם נהיה נאמנים מדיי למציאות - המודל יהיה מסובך ולא נצליח לבנות עליו כלום, כי לעולם לא נבין את היקום לאשורו (על המוח האנושי אנחנו יודעים מעט מאוד ומדענים תמימי דעים שחלקים נרחבים ממנו היו ויישארו בגדר חידה).

מצד שני, אם המודל יהיה פשטני , הוא לא ישקף את מה שקורה מסביבנו בצורה ראויה וטווח הטעות יהיה גדול מאוד.

**עקומת מאמץ מַעֲוָות** מתייחסת לחומר שלא קיים בטבע, ובאה להראות לנו איך לכל חומר יש גרף שונה שמייצג את הקשר בין המאמץ, הכוח שהופעל עליו, לבין השינוי שיתבצע בו.

יש חומרים שאחרי שהופעל עליהם כוח יחזרו למצב גיאומטרי זהה, כמו קפיץ

לעומת זאת, חרסינה שנפעיל עליה כוח דומה, תוכל להגיע לנקודת כניעה ומשם לנקודת שבר, אַל חָזור.

בחזרה לשאלה שלנו,  
ישנם אנשים מסוימים העשויים להתנגד לכך שמה שאנחנו שואלים עליו כבר נחקר,   
במהלך חצי המאה של המאה לעצב את מכונת טיורינג האוניברסלית הקטנה ביותר   
(היה זה נושא הפרס של סטיבן וולפרם בסך 25,000 דולר בשנת 2007).   
אבל אני רואה את זה כשונה במהותו, מהסיבה הבאה.   
למכונת טיורינג אוניברסלית - כלומר מכונה המדמה כל מכונה אחרת שמתוארת לה על קלטת הקלט שלה (לדוגמא, מכונת-טיורינג אוניברסלית U, מסוגלת "לסמלץ" חישוב של כל מ"ט M על קלט x) - יש את הפריבילגיה לפרוק כמעט את כל המורכבות שלה לפורמט התיאור של מכונת הקלט. אז כן, זה בדיוק מה שכל [המכונות האוניברסליות הזעירות](#המכונות_האוניברסליות_הזעירות) הידועות עושות! אבל לתוכנית שבודקת (נניח) את ההשערה של גולדבאך, או את השערת רימן, או את העקביות של תורת הקבוצות, על סרט ריק בתחילה, אין חירות כזו.   
עבור מכונות כאלה, מספר מצבים באמת עושה רושם של מדד מהותי לסיבוכיות,   
משום שהסיבוכיות הזו לא יכולה "להינעץ" בשום מקום אחר.  
אפשר לנסח את השאלה שלנו על בסיס פונקציית "הבונה העסוק" הידועה לשמצה.  
נניח ש BB(n) כלומר המספר הn-י של הבונה העסוק, מוגדר להיות מספר הצעדים המקסימלי שכל מכונת טיורינג בעלת n מצבים צועדת בריצה התחלתית על סרט ריק ובהנחה שהיא תיעצר לבסוף.  
מובן לנו ש BB(n) גדל מהר יותר מכל רצף של מספרים שלמים הניתנים לחישוב.  
איך אנחנו כל כך בטוחים בזה?  
כי אחרת, אם קצב הגדילה היה קטן/שווה, אפשר היה להשתמש בעובדה זו כדי לפתור את [בעיית העצירה](#בעיית_העצירה), בסתירה לתֵאוֹרֵמָה (הנחה הטעונה הוכחה) של טיורינג.  
  
דבר מדהים נוסף בפונקציית הBB , הוא שמדובר בפונקציה שלמה (כל פונקציה בוליאנית בינארית יכולה לבוא לידי ביטוי במונחים שלה) מוגדרת היטב, ועם זאת ברגע ש"נתקן" את האקסיומות של המתמטיקה, ניתן יהיה להוכיח אי פעם רק ערכים סופיים של הפונקציה, אפילו באופן עקרוני.  
למה?  
ניקח שוב מכונת טיורינג M שתעצור אם ורק אם יש סתירה בתורת הקבוצות של ZF.  
ברור שניתן לבנות מכונה כזו עם מספר סופי של k מצבים, אך לא נוכל לחשב את הערך של BB(k) כי ZF לא תוכל לקבוע את ערכם של BB(k) (ומכאן גם BB(k+1)/BB(k+2)..) אלא כן ZF לא עקבית! בשביל זה, ZF צריכה להוכיח שM רצה לעד וכך להוכיח את העקביות שלה וכך להיות לא עקבית במשפט גֶדֶל.  
  
הגענו למצב בו נוכל לשאול שאלה פרקטית, כמותית:  
כמה ערכי פונקציית BB ניתן לדעת?  
מהי הנקודה בה הפונקציה נוטשת את תחום בני האנוש ומגיעה לעולמות עליונים?   
האם זה קרוב יותר לn=10 או בכלל לn=10,000,000?  
בפועל, נקבעו ארבעה ערכים של BB עד כה סה"כ.  
  
  
  
BB(1)=1

BB(2)=6

BB(3)=21 – לין וראדו, 1965.

BB(4)=107 (Brady 1975) – בריידי, 1975.  
  
אנו יודעים גם כמה חסמים תחתונים למספרים גבוהים יותר:  
BB(5) ≥ 47,176,870 (Marxen and Buntrock 1990)

BB(6) ≥ 7.4 × 1036,534 (Kropitz 2010)  
BB(7)>10^10^10^10^10^7.  
  
הפסימיים מבין חובבי בעיית הBB סוברים כי BB(6) לא יהיה ידוע לעולם.  
מצד שני, הטיעון המופשט שהבאנו קודם, אומר שאם נסתפק בתורת הקבוצות של ZF, יכולים להיות עשרות מיליוני k אופציונליים, כך שכל ערכי BB(K+i) לעולם לא ניתן יהיה להוכיח. אז השאלה שלנו חוזרת: האם מספר הערכים הידועים של פונקציית BB ישאף לכמה בודדים, או יתקרב למיליון?  
  
זו השאלה בהם עסקו אדם ידידיה וד"ר סקוט ארונסון.  
אין סיכוי לתכנן מכונת טיורינג בצורה ידנית לכל המשימות הללו, כך שבתור התחלה,  
יצר אדם שפת תכנות חדשה בשם לקוניק, המיועדת לכתיבת תוכניות שמתאגדות לשפת מתווך בשם TMD ומשם לכדי מכונות טיורינג קטנות.  
  
גם לאחר כל זה, אנו מעריכים כי ניסיון ישיר לכתוב תוכנית "לקונית" שתחפש סתירה ב- ZFC יוביל למכונת טיורינג עם מיליוני מצבים.   
לכן נציג שלושה רעיונות הדרושים בכדי להוריד את כמות המצבים למספר סביר.  
  
הראשון היה להיעזר בעבודתו של הארווי פרידמן, שכתב על הבעיות האלה בעבר.  
פרידמן עמל מאז שנות השישים למצוא הצהרות אריתמטיות "טבעיות", שאינן תלויות באופן עצמאי ב- ZFC או בתיאוריות קבועות אחרות.  
הצהרותיו - הכרוכות בדרך כלל באובייקטים המכונים "גרפים בעלי סדר-משתנה" – נראות זרות ומרוחקות מקו המחשבה הסביר.  
אך כך או כך, הצהרותיו של פרידמן עדיין נראות הרבה יותר קלות לקידוד כתוכנות מחשב קצרות מאשר המנגנון המלא של הלוגיקה ותורת הקבוצות מהסדר הראשון!  
  
  
  
  
  
  
  
הרעיון השני היה משהו שקראנו לו "עיבוד בסרט".  
בעיקרון, במקום לקמפל ישירות מ- Laconic מטה עד למכונת טיורינג,  
אדם כתב מתורגמן ([interpreter](#interpreter)) במכונת טיורינג (שלקחה כ -4000 מצבים - עלות קבועה יחידה), ואז מכונת טיורינג הסופית כתבה תחילה תוכנית ברמה גבוהה יותר לסרט שלה ופירשה אותו לאחר מכן. במקום שתהליך ההידור מייצר תְּקוּרָה עצומה במספר מצבי מכונות טיורינג (ומכונה חוזרת), גישה זו נותנת לנו תקורה נוספת בלבד.   
נמצא שרעיון זה הפחית את מספר המצבים לאין שיעור.  
  
הרעיון השלישי הוצע לראשונה בשנת 2002 ואנו קוראים לזה "קידוד אינטרוספקטיבי (מביט פנימה)". כאשר אנו כותבים את התוכנית שתתפרש על קלטת מכונת טיורינג, הגישה הנאיבית תשתמש במצב מכונת טיורינג אחד לכל ביט. אבל זה לחלוטין בזבזני, מכיוון שבמכונת טיורינג בעלת n מצבים, כל מצב מכיל log (n) ביטים של מידע (בגלל המצבים האחרים שהוא צריך להצביע עליהם).   
גישה טובה יותר מנסה לנצל כמה שיותר מאותם ביטים, פעולה זו מקנה לנו חיסכון נוסף של גורם של 5 במספר המדינות.  
  
When we write the program to be interpreted onto the Turing machine tape, the naïve approach would use one Turing machine state per bit. But that’s clearly wasteful, since in an n-state Turing machine, every state contains ~log(n) bits of information (because of the other states it needs to point to). A better approach tries to exploit as many of those bits as it can; doing that gave us up to a factor-of-5 additional savings in the number of states.  
הרעיון השלישי הוצע לראשונה בשנת 2002 על ידי בן עמרם ופיטרסן (ושוכלל עבורנו על ידי לוק שפר); אנו קוראים לזה "קידוד אינטרוספקטיבי". כאשר אנו כותבים את התוכנית שתתפרש על קלטת מכונת טיורינג, הגישה הנאיבית תשתמש במצב מכונת טיורינג אחד לכל ביט. אבל זה בבירור בזבזני, מכיוון שבמכונת טיורינג במצב n, כל מדינה מכילה ~ log (n) פיסות מידע (בגלל המצבים האחרים שהיא צריכה להצביע עליהם). גישה טובה יותר מנסה לנצל כמה שיותר מאותם סיביות; פעולה זו הקנתה לנו חיסכון נוסף של גורם של 5 במספר המדינות.

מושגים:

מכונות אוניברסליות זעירות:  
כשאלן טיורינג העלה את הרעיון של מכונה אוניברסלית, הוא חשב על מודל המחשוב הפשוט ביותר שיהיה חזק דיו בכדי לחשב את כל הפונקציות האפשריות שניתנות לחישוב.  
היה זה קלוד שאנון שבשנת 1956 העלה לראשונה באופן מפורש את שאלת מציאת מכונת טיורינג האוניברסלית הקטנה ביותר האפשרית.  
הוא הראה ששני תווים (בד"כ 0 ו-1) מספיקים כל עוד משתמשים במצבים מספיקים   
(או להפך), ותמיד ניתן להחליף מצבים בתווים הללו.  
 הוא גם הראה שאף לא מכונת טיורינג אוניברסלית אחת של מצב בודד יכולה להתקיים.

בעיית העצירה:

מהמרכזיות בבעיות בתחום החישוביות.  
מנוסחת כבעיית ההכרעה (כן/לא) כדלקמן:  
בהינתן תוכנית מחשב וקלט, האם התוכנית תסיים את פעולתה בשלב כלשהו עבור קלט זה?  
  
תאוריית אי השלמות של גֶדֶל  
  
  
interpreter