**ממן 13 – מבוא לתורת החישוביות והסיבוכיות – ברנדס איתי**

**שאלה 1:**

בתרגיל נניח שהמחרוזת מופיעה בסרט העבודה עם # בראשית, ע"מ שנוכל להבחין מתי מתחיל המחרוזת.

**סעיף א':**

נבנה מכונת טיורינג דטרמיניסטית עם סרט אחד שתכריע את כך:

" על קלט כאשר מילה:

1. כל עוד הראש הקורא-כותב לא נמצא על אות לא מסומנת, סימן # (תחילת הקלט), או סימן רווח (סוף הקלט), הזז את הראש הקורא-כותב ימינה.
2. אם הראש רואה רווח, *קבל*. אחרת, סמן את האות וקרא לה .
3. הזז את הראש הקורא-כותב עד לסוף הקלט (סימן הרווח), הזז את הראש תא אחד שמאלה, סמן את התו אם איננו מסומן, וקרא לו .
4. אם , *דחה*.
5. אחרת, הזז את הראש הקורא-כותב שמאלה, ותמשיך לעשות זאת כל עוד הראש הקורא-כותב נמצא על אות לא מסומנת.
6. חזור לתחילת שלב 1.

"

נכונות:

מכונת הטיורינג מתבוננת בתו הראשון והאחרון במחרוזת שעדיין לא סומנו (ז"א, שעדיין לא עיבדנו אותן). בשלבים 1-2 אנו מגיעים לתו הראשון. בשלב 3 אנו מגיעים לתו האחרון. בשלב 4 אנו משווים את התווים שמצאנו. אם התו הראשון והאחרון אינם זהים, זהו אינו פולינדום, ולכן , ואנו דוחים. אחרת, התווים זהים, ומתאימים להגדרת הפולינדרום. אנו מזיזים בשלב 5 את הראש לתחילת המחרוזת, וחוזרים לשלב 1, ע"מ להשיג תווים חדשים מקצוות המחרוזת שעוד לא סרקנו. בסיום, אם נגיע למצב שכל המחרוזת מסומנת, נקבל בשלב 2. זה יקרה רק אם לא נתקלנו בשום סתירה, ז"א, לא קיימים כך ש , ולכן פולינדרום.

ניתוח סיבוכיות זמן-ריצה:

יהי .

שלבים 1,3,5 רצים במקרה הגרוע ב. שלבים 2,4,6 רצים בסיבוכיות קבועה. לפיכך שלבים 1-6 רצים בסיבוכיות לינארית.

שלבים 1-6 רצים במקרה הגרוע פעמים, ולכן זמן הריצה הכולל של המכונה הוא , ולכן:

**סעיף ב':**

נבנה מכונת טיורינג דטרמיניסטית עם שני סרטים שתכריע את כך:

" על קלט כאשר מילה:

1. הזז את הראש הקורא-כותב הראשון ימינה עד שתראה אות רווח. חזור מקום אחד שמאלה.
2. כל עוד הראש הקורא-כותב הראשון מצביע על תו שאיננו סימן # (תחילת הקלט):
   1. כתוב את התו בראש הקורא-כותב השני, והזז אותו ימינה.
3. הזז את הראש הקורא-כותב השני לראשית הסרט (עד הסימן #).
4. כל עוד הראש הקורא-כותב הראשון מצביע על תו שאיננו סימן רווח (סוף הקלט):
   1. הזז את שני הראשים משבצת אחת ימינה, וקרא את התווים מהם כ ו- בהתאמה.
   2. אם , *דחה*.
5. *קבל*.

"

נכונות:

מכונת הטיורינג נעה לסוף הקלט בסרט הראשון, ואז סורקת אות-אות (משמאל לימין) וכותבת כל אות לסרט השני (בסדר הפוך). (שלבים 1-2)

כעת בסרט הראשון קיימת המחרוזת , ובסרט השני קיימת המחרוזת .

בשלב 3 אנו מחזירים את הראש הקורא-כותב השני לתחילת המחרוזת. בשלב 4 אנו סורקים את 2 המחרוזות איבר-איבר, ודוחים אם אנו נתקלים באי-שוויון. אחרת, לא קיימים כך ש , ולכן פולינדרום.

ניתוח סיבוכיות זמן-ריצה:

יהי .

שלבים 1,2,4 רצים במקרה הגרוע ב. שלבים 3,5 רצים בסיבוכיות קבועה. לכן, זמן הריצה הכולל של המכונה הוא ומתקיים:

נבחין שמשום שבמהלך האלגוריתם, עלינו לקרוא את הקלט כולו, לכן לא קיים אלגוריתם (= לא קיימת מכונת טיורינג דטרמיניסטית) בעל קטן מ.

**סעיף ג':**

נבנה מכונת טיורינג דטרמיניסטית עם סרט אחד בעל שני ראשים קוראים-כותבים שתכריע את כך:

" על קלט כאשר מילה:

1. הזז את הראש הקורא-כותב השני אל סוף הקלט (עד שנתקל בסימן רווח).
2. כל עוד הראש הקורא-כותב הראשון מצביע על תו שאיננו סימן רווח (סוף הקלט):
   1. הזז את הראש הראשון משבצת אחת ימינה, וקרא את התו ממנו כ.
   2. הזז את הראש השני משבצת אחת שמאלה, וקרא את התו ממנו כ.
   3. אם , *דחה*.
3. *קבל*.

"

נכונות:

מכונת הטיורינג נעה מזיזה את הראש הקורא-כותב השני לסוף הקלט (שלב 1) ואז סורקת אות-אות (הראש הראשון נע מימין לשמאל, והשני משמאל לימין). בכך אנו סורקים בו-זמנית את המחרוזת מההתחלה ומהסוף, ודוחים אם אנו נתקלים באי-שוויון. אחרת, לא קיימים כך ש , ולכן פולינדרום.

ניתוח סיבוכיות זמן-ריצה:

יהי .

שלבים 1 ו-2 רצים במקרה הגרוע ב. לכן, זמן הריצה הכולל של המכונה הוא ומתקיים:

נבחין שמשום שבמהלך האלגוריתם, עלינו לקרוא את הקלט כולו, לכן לא קיים אלגוריתם (= לא קיימת מכונת טיורינג דטרמיניסטית) בעל קטן מ.

**שאלה 2:**

**סעיף א':**

יהי .

נבנה אוטומט סופי דטרמיניסטי שיכריע את כך:

" על קלט כאשר אוטומט סופי דטרמיניסטי ו מילה:

1. נסמן את המצב ההתחלתי של .
2. נסרוק את התווים שבמילה תו תו. על כל תו :
   1. לכל מצב מסומן , נמחק את הסימון, ונחשב את כל המצבים שיש מעבר אליהם בפונקציית המצבים בקריאת . נסמן את כל המצבים ה הללו.
3. אם קיים מצב מקבל בקבוצת המצבים המסומנים, *קבל*. אחרת, *דחה*.

"

נכונות:

*נבחין שבכל שלב בלולאה שבשלב 2, קבוצת המצבים המסומנים מציינים את קבוצת המצבים שניתן להגיע אליהם מהמצב ההתחלתי בקריאת כל התווים , כאשר הוא מספר התו שאנו סורקים כרגע.*

*לכן, ברור שכשנסיים לקרוא את , אם נמצא בקבוצת המצבים המסומנים מצב מקבל, סימן שבקריאת נסיים במצב מקבל, ולכן . אחרת, בקריאת אנו מסיימים במצב דוחה, ולכן . לפיכך האוטומט מכריע את השפה כהלכה.*

ניתוח סיבוכיות זמן-ריצה:

יהי .

הלולאה בשלב 2 רצה פעמים, ובכל פעם מבצעת עבודה קבועה (מסמנת במקרה הגרוע מצבים, כאשר זה מספר המצבים באוטומט, שהוא מספר קבוע לצורך חישוב סיבוכיות זמן הריצה), ולכן:

כנדרש.

**סעיף ב':**

יהי

ניתן לבצע שינויים באלגוריתם מהוכחת משפט 7.16 (ע"מ 291 בספר הלימוד) ע"מ שיכריע את השפה . השינויים לא ישפיעו על זמן הריצה, ולכן הוא ישאר כפי שהוכח בספר, ולכן הוא פולינומיאלי, ונמצא ב.

אלגוריתם המקורי שומר בכניסה את רשימת המשתנים שניתן לגזור מהם את תת המחרוזת . באלגוריתם החדש נשמור לא רק את המשתנים הללו, אלא גם את מספר האפשרויות השונות לגזור מהן את תת המחרוזת . בניגוד לאלגוריתם המקורי, שבו משתנים נמצאים בכניסה כלשהי רק אם הם גוזרים את המחרוזת הרלוונטית, הפעם בכל כניסה ימצאו כל המשתנים, אך אם משתנה אינו גוזר את התת-מחרוזת, הוא יופיע עם 0 אפשרויות גזירה.

לצורך נוחות, נגדיר את הסימון ע"מ לגשת למספר אפשרויות הגזירה של המשתנה A ב.

האלגוריתם החדש יהיה, אם כן:

(השורות המודגשות בצהוב מהוות את השינויים מהאלגוריתם המקורי, שאר השורות מועתקות מן האלגוריתם המקורי)

" על קלט כאשר הוא דקדוק חסר-הקשר בצורה הנורמלית של חומסקי ו מילה:

1. אם , *דחה*.
2. לכל משתנה A ולכל כניסה בטבלה, נגדיר .
3. לכל :
   1. לכל משתנה A, אם קיים חוק , הגדר .
4. לכל :
   1. לכל :
      1. הגדר
      2. לכל :
         1. לכל חוק בצע:
5. אם , *קבל*. אחרת, *דחה*.

"

נכונות:

*בספר הלימוד הוכח שהאלגוריתם המקורי נכון. נבחן את השינויים שביצענו באלגוריתם ונראה שהם גורמים לאלגוריתם להכריע את השפה .*

*בשלב 1 אנו דוחים תמיד את הקלט כאשר , שכן אפילו אם האלגוריתם המקורי מקבל עבור קלט ריק, אנו לא יכולים לקבל, שכן היא גזירה אחת בדיוק, ולכן לא קיימת בשפה .*

*בשלב 2 אנו מאתחלים את הטבלה והמשתנים עם ערכי 0.*

*בשלב 3, באלגוריתם המקורי אנו מכניסים את המשתנה למקום הרלוונטי בטבלה. כאן אנו פונים למקום זה, שכבר מאותחל ל-0, ומגדילים את מספר הגזירות ל-1.*

*בשלב 4, עבור כלל , אנו מבצעים את אותה פעולה כמו באלגוריתם המקורי, רק שכעת איננו רק מכניסים משתנה למקום הרלוונטי, אלא מגדירים את מספר הגזירות שלו כמכפלת מספר הגזירות של כל אחד מהמשתנים בנפרד. נבחין שאם B או C אינם גוזרים, אז מספר הגזירות שלהן הוא 0, ואז כמובן שגם עבור A נציב מספר גזירות 0.*

*בסיום, בשלב 5, אנו מקבלים רק אם מספר הגזירות של המשתנה ההתחלתי את המילה גדול מ1, ודוחים אחרת, וזה כמובן בודק כהלכה האם המילה נמצאת בשפה ביחס לדקדוק G.*

ניתוח סיבוכיות זמן-ריצה:

יהי , ו מספר המשתנים בדקדוק .

נבחן את זמן הריצה של השינויים באלגוריתם מן האלגוריתם המקורי:

שלב 1 רץ באותו זמן ריצה.

שלב 2 הוא שלב חדש שלא קיים באלגוריתם המקורי. זמן הריצה שלו הוא .

בשלב 3 אנו מחליפים עבודה קבועה אחת באחרת, וזמן הריצה לא משתנה.

בשלב 4 אנו מבצעים כפל בין 2 מספרים, במקום לבצע 2 בדיקות. זמן הריצה נשאר קבוע.

שלב 5 נשאר בזמן ריצה קבוע.

לפיכך, זמן הריצה החדש גדול ב מזמן הריצה המקורי של האלגוריתם .

בספר הלימוד הראו שהאלגוריתם המקורי רץ בזמן של , ולכן זמן הריצה איננו גדל, ומתקיים:

כנדרש.

**שאלה 3:**

**סעיף א':**

יהי .

לכל מילה השייכת לשפה יש מילה שלא נמצאת ב (שכן ).

נגדיר מאמת שיקבל כקלט ובודק בעזרת האלגוריתם להכרעת (ע"מ 198 בספר הלימוד), האם המילה נגזרת **לא** מהדקדוק .

= " על קלט כאשר דקדוק חסר-הקשר ו מילה:

1. אם איננו בצורה הנורמלית של חומסקי, נמיר אותו לכזה.
2. נסמלץ את האלגוריתם עם הפרמטרים . אם האלגוריתם מקבל, *נדחה*. אחרת, *נקבל*.

"

**סעיף ב':**

גודל המילה שמוכיחה את השייכות של לשפה איננו חסום. לכן, הוא יכול גם להיות לא פולינומיאלי לגודל הקלט.

**סעיף ג':**

לפי משפט 5.13 (ע"מ 225 בספר הלימוד), השפה איננה כריעה. לכן גם השפה איננה כריעה.

נניח בשלילה ש שייכת ל. לפי משפט 7.20 (ע"מ 294 בספר הלימוד), כריעה (בזמן פולינומיאלי ע"י מכונת טיורינג לא-דטרמיניסטית), בסתירה לכך ש לא כריעה.

לפיכך, לא ב.

**שאלה 4:**

נבנה מאמת V ש*יקבל כקלט כאשר n,m המספרים המוגדרים בהגדרת השפה, וc היא מחרוזת האימות – הפירוק לגורמים ראשוניים, כך:*

= " על קלט כאשר מספרים ו הוא פירוק לגורמים:

1. נוודא שכל הגורמים בפירוק הם אכן ראשוניים. אחרת, *נדחה*.
2. נוודא שמכפלת כל הגורמים בפירוק היא n. אחרת, *נדחה*.
3. נוודא שמספר הגורמים השונים בפירוק הוא איננו גדול מ-m. אחרת, *נדחה*.
4. אם הגענו לשלב זה, *נקבל*.

"

*נבחין שn וm שניהם מיוצגים ע"י מחרוזת בגודל ו- בהתאם.*

*מספר הגורמים הראשוניים של n הוא אינו גדול מ*, ולכן אורך הייצוג שלהם במחרוזת הוא כמובן לא גדול מ. לפיכך, גודל מחרוזת האישור היא פולינומיאלית בגודל הקלט.

שלב 1 יכול להיות פולינומיאלי בגודל הקלט אם נשתמש באלגוריתם AKS, שפורסם בשנת 2002 ובודק בזמן לינארי האם מספר הוא ראשוני.

שלבים 2-3 גם הם פולינומיאלים בגודל הקלט.

*ולכן, זמן הריצה של כל אחד מהשלבים בV הוא פולינומיאלי בגודל הקלט.*

*ובזאת הצגנו מאמת שמאמת בזמן ריצה פולינומיאלי את השפה . לכן, שייכת ל-.*

**שאלה 5:**

בהוכחת Cook-Levin (ע"מ 304 בספר הלימוד) בונים את הנוסחה הבוליאנית

הדרישה הכרחית, שכן שאר הדרישות מניחות, בצורה נאיבית, שקיים תו אחד בדיוק בכל מקום בטבלה, ולא טורחות לבדוק זאת. בפרט הדרישה בודקת האם כל חלון של הוא חלון חוקי, ע"י כך שהיא בודקת האם המעבר אפשרי ע"י פונקציית המעברים .

(\*) נניח שפונקציית המעברים תאפשר מעבר ל2 תווים שונים (ע"י אי-דטרמיניזם):

(\*\*) ושבטבלה שנבנה יופיע חלון עם 2 תווים אלה באותו מקום (ז"א, מאחורי הקלעים, הביטים המתאימים להימצאות 2 התווים באותו תא יהיו דלוקים):

|  |  |  |
| --- | --- | --- |
|  |  | x |
|  |  | x |

הכלל יקבל ערך אמת, שכן הוא רק בודק שקיים מעבר תקין, ולא שהמעבר הוא יחיד ובודד. עם זאת, ברור שהטבלה, המבטאת מסלול חישוב, לא יכולה בפעולה אחת לעבוד עם 2 שורות מטבלת המעברים, ולכן מכונת הטיורינג שננסה ליצור מן הטבלה תהיה תקולה, או שאף לא יהיה ניתן לבנות אותה.

במילים אחרות, אם אנו מייתרים את הדרישה ומגדירים את הנוסחה הבוליאנית הבאה:

*תהי שפה לא-ריקה כלשהי, שבמכונת הטיורינג הלא-דטרמיניסטית שלו קיימים התנאים (\*).*

*נניח בשלילה שהרדוקציה תקפה. אז מתקיים שלכל , אם ורק אם ספיקה.*

*נבחר שבמסלול החישוב שלו משתמשים במעברים (\*), ונבנה טבלה מתאימה שתציג מסלול חישוב תקין, שיכיל גם את החלון (\*\*).*

*הנוסחה תהיה ספיקה, אך מכונת הטיורינג הלא-דטרמיניסטית שננסה לבנות מטבלה זו יהיה תקול, ולכן יתכן שלא יקבל את .*

*ולכן, הדרישה הינה הכרחית לרדוקציה מבעית ל.*

**שאלה 6:**

נבנה רדוקציה חשיבה פולינומיאלית מ ל. נשתמש בהדרכה שבתרגיל:

יהי גרף כלשהו. כל צמתי הגרף יצבעו באחד משלושה צבעים – צבע 1,2,3.

נגדיר משתנה בוליאני שנקצה לו תמיד את הערך 0 (שקר). נשתמש בו כדי "לעבות" פסוקיות של 2 משתנים ל3 משתנים, ע"מ שיתאימו לצורת .

לכל צומת נגדיר שלושה משתנים בוליאניים: .

המשתנה יהיה שווה ל-1 אם הצומת צבוע בצבע . כל שאר המשתנים יהיו שווים ל-0. על מנת לכפות דרישה זו, נגדיר לכל צומת את הפסוקיות הבאות:

1. (הצומת צבוע באחד הצבעים)
2. (הצומת לא צבוע בצבעים 1 ו-2 גם יחד)
3. (הצומת לא צבוע בצבעים 2 ו-3 גם יחד)
4. (הצומת לא צבוע בצבעים 1 ו-3 גם יחד)

כמו כן, לכל קשת נדרוש ש ו- יצבעו בצבעים שונים. על מנת לכפות דרישה זו, נגדיר לכל קשת את הפסוקיות הבאות:

1. (הצמתים לא צבועים שניהם בצבע 1)
2. (הצמתים לא צבועים שניהם בצבע 2)
3. (הצמתים לא צבועים שניהם בצבע 3)

לסיום, ניקח את כל הפסוקיות שיצרנו ונציב אותם בנוסחה עם סימני (AND) ביניהם.

**נראה מדוע הרדוקציה תקפה:**

נראה שלגרף יש צביעה חוקית בשלושה צבעים אמ"מ ההשמה המתאימה מספקת את הנוסחא .

כיוון =>:

נניח שלגרף יש צביעה חוקית בשלושה צבעים. לכל צומת , אם בצביעה , הצבע של הוא , אז ערכו של הוא 1, וערכו של שאר המשתנים הוא 0. הנ"ל מספק את פסוקיות 1-4 כהלכה.

כמו כן, משום שהצביעה חוקית, לכל קשת יתקיים בהכרח שהצבע של שונה מהצבע של , ולכן גם מסתפקים פסוקיות 5-7.

ובכך ההשמה מספקת את הנוסחה המתקבלת.

כיוון <=:

נניח שהנוסחה ספיקה ע"י ההשמה . אז לפי פסוקיות 1-4, לכל קיים ערך בודד עבורו , וכל שאר המשתנים מקיימים . לפיכך, בגרף שיתקבל מהרדוקציה, כל צומת יהיה צבוע בצבע אחד בדיוק, כדרוש.

בנוסף, פסוקיות 5-7 ספיקות גם הן, ולכן אנו יודעים שלא קיימים צמתים שונים בעלי אותם צבעים המחוברים בקשת. לפיכך, הצביעה של הינה צביעה חוקית.

**נראה שהרדוקציה ניתנת לחישוב בזמן פולינומיאלי:**

לכל צומת אנו בונים ארבע פסוקיות, בעלי 3 ליטרלים כל אחד, ולכל קשת אנו בונים שלוש פסוקיות בעלי 3 ליטרלים. ברור שהסיבוכיות לכך היא , ועל כן ודאי פולינומיאלית.

לפיכך היא רדוקציה חשיבה פולינומיאלית מ ל, ולכן מתקיים , כנדרש.

**שאלה 7:**

נמצא את כל השפות L המקיימות :

כמובן ש מקיימת *, שכן הפונקציה היא פונקציה חשיבה פולינומיאלית כך שלכל מתקיים .*

*כעת, נניח בשלילה שקיימת שפה כך ש .*

*בפרט מתקיים , כלומר, קיימת פונקציה חשיבה פולינומיאלית כך שלכל מתקיים (\*).*

*מתקיים , ולכן קיימת מילה המקיימת . לפי (\*) מתקיים , בסתירה לכך ש היא שפת כל המילים. לפיכך, לא קיימת שפה כזאת.*

*לסיכום, כל השפות L המקיימות*  הן בלבד.

נמצא את כל השפות L המקיימות :

כמובן ש מקיימת *, שכן הפונקציה היא פונקציה חשיבה פולינומיאלית כך שלכל מתקיים (במובן הריק).*

*כעת, נניח בשלילה שקיימת שפה כך ש .*

*בפרט מתקיים , כלומר, קיימת פונקציה חשיבה פולינומיאלית כך שלכל מתקיים (\*\*).*

*מתקיים ולכן קיימת מילה המקיימת . לפי (\*\*) מתקיים , בסתירה לכך ש היא השפה הריקה. לפיכך, לא קיימת שפה כזאת.*

*לסיכום, כל השפות L המקיימות*  הן בלבד.

**שאלה 8:**

**סעיף א':**

יהי .

נגדיר מכונת טיורינג שתכריע את השפה .

= " על קלט כאשר אוטומטים סופים דטרמיניסטים:

1. בנה אוטומט מכפלה של השפות ו .
2. בדוק בעזרת מכונת טיורינג המכריעה את השפה (ע"מ 196 בספר הלימוד), האם .
3. אם המכונה מקבלת, *דחה*. אחרת, *קבל*.

"

*בשורה 1 אנו בונים את אוטומט המכפלה , ששפתו היא חיתוך שפות A ו-B. מובן שבעזרת נוכל לדעת האם היא שפה ריקה או לא. נחזיר תשובה הפוכה לתשובת במכונת הטיורינג שלנו.*

*בניית אוטומט המכפלה לוקחת זמן פולינומיאלי לגודל הקלט (אם כל אוטומט ניתן לקודד למחרוזת בגודל n, אז במקרה הגרוע בניית אוטומט המכפלה ייקח זמן של ).*

*האוטומט של השפה גם הוא רץ בזמן פולינומיאלי לגודל הקלט. לכן, מתקיים:*

כנדרש.

**סעיף ב':**

יהי

ע"מ להראות שהיא שפה -קשה, עלינו להראות רדוקציה מבעיית -שלמה ידועה אליה. את הרדוקציה נעשה מ, שידועה כבעיה -שלמה.

נבנה רדוקציה חשיבה פולינומיאלית מ ל. נשתמש בהדרכה שבתרגיל:

(\*) לכל פסוקית בנוסחה נרצה לבנות אוטומט סופי דטרמיניסטי שיכריע את שפת כל המילים באורך (כאשר מספר המשתנים הבוליאניים השונים בנוסחה) המייצגות השמה שמספקת פסוקית זו.

(נניח שאם באוטומט לא קיימת פונקציית מעברים בקריאת ערך כלשהו, המשמעות היא שאנו נכנסים למצב מלכודת דוחה)

לצורך פשטות, נגדיר 2 אוטומטים, ואז נדרוש להגיע למצב מקבל בשניהם: אוטומט וידוי תקינות קלט , ואוטומט וידוי השמה מספקת .

בתוך האוטומט , נגדיר מצבים העוברים בתור מאחד להבא בתור בקריאת 0 או 1. המצב האחרון יהיה מצב מקבל. כך נוודא שהקלט הוא בגודל בדיוק ומכיל קלטים תקינים (0 או 1 בלבד).

בתוך האוטומט נגדיר מצבים, העוברים בתור מאחד להבא בתור בקריאת 0 או 1.

עם זאת, כאשר אנו מגיעים למצב של משתנה בוליאני בפסוקית שאנו קוראים עבורו ערך 1, במקום לעבור למצב הבא בתור, נעבור ישירות למצב מקבל חדש שנגדיר, יחד עם מלכודת שתמשיך לחזור לאותו מצב בקריאת כל תו.

כעת נגדיר את האוטומט להיות אוטומט המכפלה של האוטומטים ו-.

האוטומט יהיה, אם כן:

= " על קלט כאשר היא נוסחת :

1. לכל פסוקית בנוסחה , נבנה אוטומט סופי דטרמיניסטי , כפי שתואר ב(\*).
2. החזר את .

"

**נראה מדוע הרדוקציה תקפה:**

נראה ש*ספיקה ע"י ההשמה s אמ"מ שפת חיתוך שפות האוטומטים המתקבלים אינה ריקה.*

כיוון =>:

נניח שהנוסחה ספיקה ע"י ההשמה s. אז בהשמה s של משתני הנוסחה הבוליאניים, יש בכל פסוקית לפחות משתנה בוליאני אחד שערכו 1. את ההשמה מתרגמים למספר בינארי באורך בשם . לפיכך, במספר הבינארי קיים ביט דלוק (המכיל 1). לכן, בכל אוטומט מכפלה מתאים, האוטומט בקריאת מגיע למצב מקבל, שכן הוא באורך ומכיל רק 0 ו-1, והאוטומט בקריאת מגיע גם למצב מקבל, שכן כפי שראינו, קיים ביט דלוק. לפיכך, כל האוטומטים שבנינו בדרך זו מקבלים את מילת המספר הבינארי , ולכן גם שפת חיתוך כל שפות האוטומטים הללו מכילה את , ולכן שפת החיתוך איננה ריקה, כנדרש.

כיוון <=:

נניח ש*שפת חיתוך שפות האוטומטים המתקבלים אינה ריקה. אז היא מכילה מספר בינארי כלשהו. אם נבנה נוסחה באופן הפוך לרדוקציה, נקבל נוסחה של משתנים בוליאניים ו פסוקיות. נסתכל במספר כהשמה כלשהי. נבחין ששפת כל אוטומט מקבלת את , ולכן סימן שיש בחלק ה של האוטומט מעבר למצב מקבל. כל אחד מ הפסוקיות נבנה מאוטומט מתאים, ולכן מתקבל שבפסוקית בהכרח יהיה ליטרל אחד שיקבל את הערך 1. לפיכך, כל אחד מהפסוקיות ספיק, ולכן הנוסחה כולה ספיקה.*

**נראה שהרדוקציה ניתנת לחישוב בזמן פולינומיאלי:**

הקלט מכיל משתנים ו פסוקיות, ובסה"כ תווים.

בניית כל אוטומט אורכת כזמן בניית 2 אוטומטיים עם משתנים ו מעברים (), ואז ביצוע אוטומט מכפלה ביניהם (). עלינו לבנות אוטומטים כאלה, ולכן זמן הריצה הכולל הוא . ברור שזמן הריצה זה הוא פולינומיאלי לגודל הקלט.

לפיכך היא רדוקציה חשיבה פולינומיאלית מ , ולכן מתקיים , ואז נקבל שמשום ש, בעיה -שלמה, היא בעיה -קשה.

**סעיף ג':**

ההבדל הוא שמספר האוטומטים בשפה של סעיף א' הוא 2 (מספר קבוע) ומספר האוטומטים בשפה של סעיף ב' הוא משתנה שניתן כקלט ואיננו ידוע מראש.

זמן הריצה הדרוש לבניית אוטומט מכפלה של שני האוטומטים הוא ריבועי לגודל האוטומטים, *ז"א, .*

*זמן הריצה הדרוש לבניית אוטומט מכפלה של אוטומטים הוא בחזקת לגודלם, ז"א, . זמן זה איננו פולינומיאלי בגודל הקלט, שכן המעריך איננו קבוע, ומשתנה עם הקלט.*

*לכן, הגיוני שהשפה של סעיף א' שייכת ל-P כאשר השפה של סעיף ב' היא NP-קשה.*