

תרגיל בית מס' 5 - תרגום לקוד ביניים ו-backpatchשאלה 1

בתרגול ראינו סכמת תרגום בשיטת backpatch לתרגום פקודת switch לקוד ביניים כאשר התחביר אילץ break בכל case אחרי ה- S_i המתאים. בפועל, לעומת זאת, לא חייב להיות break בסוף כל S_i .

ללא break, המשמעות של switch היא שבתום ביצוע S_i , הביצוע ממשיך ל- S_{i+1} (או לאחר כל ה-switch עבור ה- S_i האחרון).

break היא פקודה מיוחדת. היא יכולה להופיע רק בתוך מבנה בקרה מורכב (if הינו מבנה בקרה פשוט). משמעותה של פקודת ה-break היא סיום ביצוע מבנה הבקרה המורכב (הפנימי ביותר) בו היא נמצאת. לדוגמה:

```
while ...
:
switch ...
:
case ...:
if ...
break; ← סיום ה-switch
:
:
:
break; ← סיום ה-while
:
```

בפרט, אם בפקודת switch כל ה- S_i -ים גוזרים סדרת פקודות שהאחרונה בהן היא break (ולכן מתאימה לתחביר שניתן בתרגול) אז ה-switch יסתיים בתום ביצוע S_i (כמו שהגדרנו בתרגול).

כתיבת תחביר המאפשר פקודות break רק בתוך מבני בקרה מורכבים אפשרית, אבל לא יעילה. ניתן, במקום זה, לוודא סמנטית שאין break מחוץ למבני הבקרה המורכבים.

א. איך אפשר לדאוג בתחביר שפקודת break תהיה רק בתוך מבני בקרה מורכבים? מדוע זה לא יעיל?

ב. איך אפשר לוודא ע"י תכונות נורשות (וללא שימוש במשתנים גלובלים) ש-break יהיה רק בתוך מבנה בקרה מורכב?

ג. איך אפשר לוודא ע"י תכונות נוצרות (וללא שימוש במשתנים גלובלים) ש-break יהיה רק בתוך מבנה בקרה מורכב? רמז: השגיאה, אם קיימת, לא תתגלה בעת ניתוח ה-break אלא בשלב מאוחר יותר.

ד. כתבו סכמת תרגום בשיטת backpatch עבור switch ו-break (ללא שימוש בגלובליים).

- על הסכמה להיות מבוססת על הסכמה שניתנה בתרגול.

- מותר להוסיף תכונות נוצרות ל-S, אך יש להסביר (בקצרה) מה ניתוח שאר הפקודות עושה עם התכונות החדשות.

שאלה 2

נוסיף את הפקודה הבאה (הרחבה של לולאת for בפסקל):

$S \rightarrow \text{for } id \text{ assign } E_1 \text{ TO } E_2 \text{ STEP } do \ S_1$
 $TO \rightarrow \underline{to} \mid \underline{downto}$
 $STEP \rightarrow \underline{step} \ E \mid \varepsilon$

המשמעות של הפקודה היא:

ביצוע S_1 בלולאה כאשר id מקבל ערכים מ- E_1 ועד (כולל) E_2 בקפיצות של STEP (או 1).

הערות:

1. חישוב E_1, E_2 ו- E של STEP (אם קיים) יבוצע בתחילת הפקודה בלבד (ולפי סדר זה).
2. אם התחום הפוך לא תבוצע אף איטרציה. במקרה כזה אין לשנות את ערכו של id .
- התחום הפוך כאשר יש שימוש ב- to ו- $E_1 > E_2$, או כאשר יש שימוש ב- $downto$ ו- $E_1 < E_2$.
3. בתום כל איטרציה id מקודם (עבור שימוש ב- to) או מופחת (עבור שימוש ב- $downto$) בערך שחושב עבור STEP (או ב-1).
4. קיימים עוד כללי גזירה עבור S ועבור E .

א. הציעו פריסת קוד מתאימה לשיטת backpatch עבור מבנה הבקרה הנ"ל.

ב. כתבו סכמת תרגום בשיטת backpatch המייצרת את פריסת הקוד שנתתם בסעיף הקודם. אין לכתוב פעולות סמנטיות באמצעו של חוק (השתמשו במרקרים) ואין להשתמש במשתנים גלובליים בזמן קומפילציה.

ג. נניח כי ערכם של E_1, E_2 ו-STEP ידועים בזמן קומפילציה. מתי אפשר ומתי אי אפשר לדעת בזמן קומפילציה את מספר האיטרציות שיבוצעו בזמן ריצה? רישמו תנאי בניסוח לא פורמלי.

הגשה:

בזוגות או ביחידים (מומלץ בזוגות), עד לתאריך **9.07.04 בשעה 12:00 בצהריים**.

- הקפידו למלא אחר הוראות הגשת תרגילי הבית, כפי שמפורסמות באתר.

בהצלחה