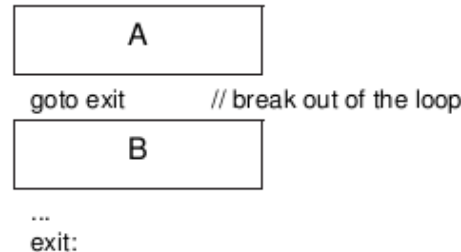
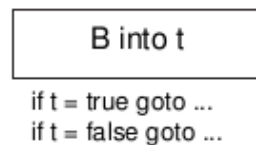


- פריסת קוד אמורה להציג את הקוד שסכימת התרגום תייצר. אין הכוונה לתרשים זרימה או דיאגרמה שמסבירה את הקוד לוגית, אלא ממש לקוד הביניים עצמו, כאשר נתונים שאינם ידועים (קוד של משתנים באגף ימין, כתובות וכו') מוחלפים בייצוג סכימטי ("קופסאות" עבור הקוד של המשתנים הנגזרים, תוויות במקום כתובות וכו'). את הקוד שהסכימה שלכם מייצרת יש להציג במלואו, ולא להחליף אותו בתכנון ברמה גבוהה; למשל, במקום "אתחול flag" יש לרשום "flag := 0".
- אין לכתוב בסוף פריסת הקוד תווית "end" או "exit" וכיובל שמייצגת את נקודת היציאה:



- לא מובטח כלל שהפקודה הבאה אחרי מבנה הבקרה אכן תהיה הפקודה הבאה לביצוע אחרי שהביצוע שלו מסתיים (למשל?).
- בשיטת backpatching, ביטויים בוליאניים (שנגזרים ע"י B) אינם נשמרים בזכרון. אין להניח של-B יש תכונה place, ואין להשתמש בפריסות בסגנון:



- הפונקציה backpatch משלימה כתובות בלבד – אך ורק חורים מהצורה goto \_\_ (קפיצה או קפיצה מותנית). אין להשתמש בה כדי להשלים אופרטורים או שמות של משתנים זמניים!
- כאשר מבקשים "קוד יעיל", הכוונה (אלא אם כן צוין אחרת) לקוד יעיל מבחינת זמן ריצה. אל תחסכו בפקודות בשפת ביניים אם פירוש הדבר הגדלת זמן הריצה. בתרגיל לא הורדו על כך נקודות, למרות שרבות מהפריסות שהוצעו היו לא יעילות מבחינה זו.
- מצד שני, כשהדבר אינו מגדיל את זמן הריצה יש לחסוך גם בגודל הקוד. אין לשכפל פקודות בלי צורך, וכו'.
- בשפת ביניים אין משפטי if, יש רק קפיצות מותנות בהן תנאי הקפיצה הוא מהצורה: [operand1] [rel-op] [operand2], ואין תנאים מורכבים יותר.
- אל תייחסו לשפת ביניים תכונות של השפה העילית החביבה עליכם. למשל, פקודות כגון:

```

if t goto ...
if !t goto ...
  
```

אולי עובדות ב-C, אבל בשפת ביניים יש לכתוב:

```

if t != 0 goto ...
if t = 0 goto ...
  
```

דוגמאות נוספות: בשפת ביניים אין scope-ים ואין הכרזה על משתנים, ולכן אין בה קוד כגון:

```

int tmp1, tmp2;
if(tmp1 = 0)
{
    goto ...
}
  
```

בוודאי ובוודאי שאין להוסיף ל-if הנ"ל:

```

else ...
  
```

שפת ביניים הוגדרה היטב בהרצאה – אל תחרגו מסוגי הפקודות שהוגדרו. בתרגיל לא הורדו על כך נקודות, כל עוד ההבדלים היו קלים ולא הרחיבו את הכוח של הפקודות.

- כאשר מופיעים בכלל משתנים מהצורה M1, M2, M3 אין הכוונה לשלושה משתנים נפרדים M1, M2, M3 אלא לשלושה מופעים שונים של אותו המשתנה M. האינדקס נועד אך ורק כדי להבדיל בין המופעים השונים שלהם באגף ימין של אותו כלל. אין לכתוב כללים נפרדים עבור כל מופע. בתרגיל לא הורדו על כך נקודות.
- על פריסת הקוד שלכם להיות מובנת מבלי שהבודקת תצטרך לשלוף זכוכית מגדלת ו/או לסמן על הפתרון סימונים קרטוגרפיים כדי להבין איזה חץ הולך לאן. אם יש יותר מדי חיצים, החליפו אותם במטציה הבאה:

```

labelA: [A]
        goto labelA
labelB: [B] // truelist: labelB
           // falselist: labelA
        ...
        goto labelB

```