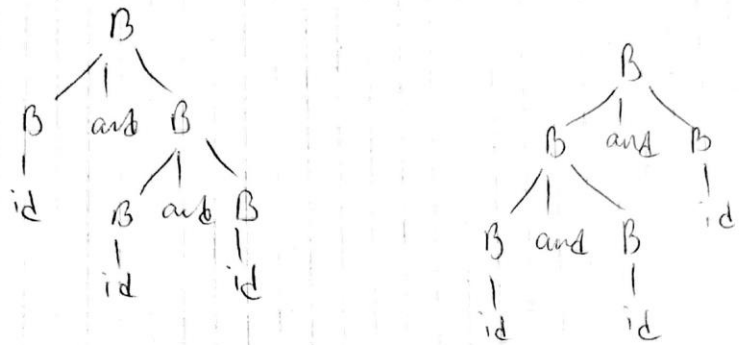
שאלה 1:

א. תאור המילה id and id and id נותן 2 עצי גזירה:



ב.

$B \rightarrow B \text{ and } T \mid T$

$T \rightarrow \text{not } T \mid F$

$F \rightarrow id \mid (B)$

ג. הדקדוק איננו LL(1) משום שקבוצות ה-select תאור שני כללים נחתכות:

$select(B \rightarrow B \text{ and } T) = \{not, (, id\}$

$select(B \rightarrow T) = \{not, (, id\}$

הדקדוק החדש:

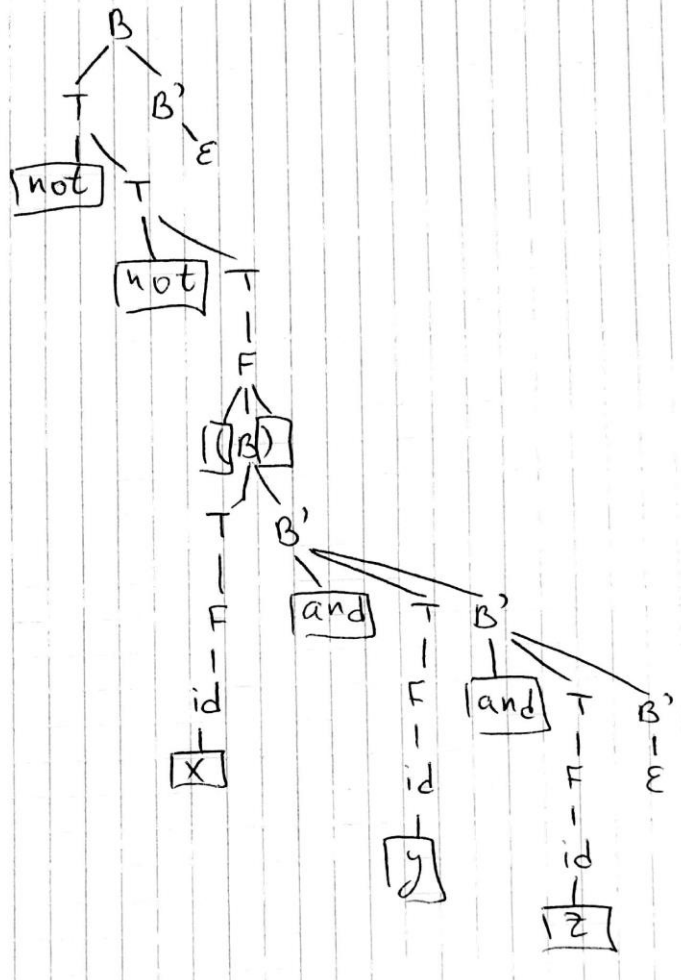
$B \rightarrow T B'$

$B' \rightarrow \text{and } T B' \mid \varepsilon$

$T \rightarrow \text{not } T \mid F$

$F \rightarrow id \mid (B)$

3. עץ גזירה עבור not not (x and y and z):

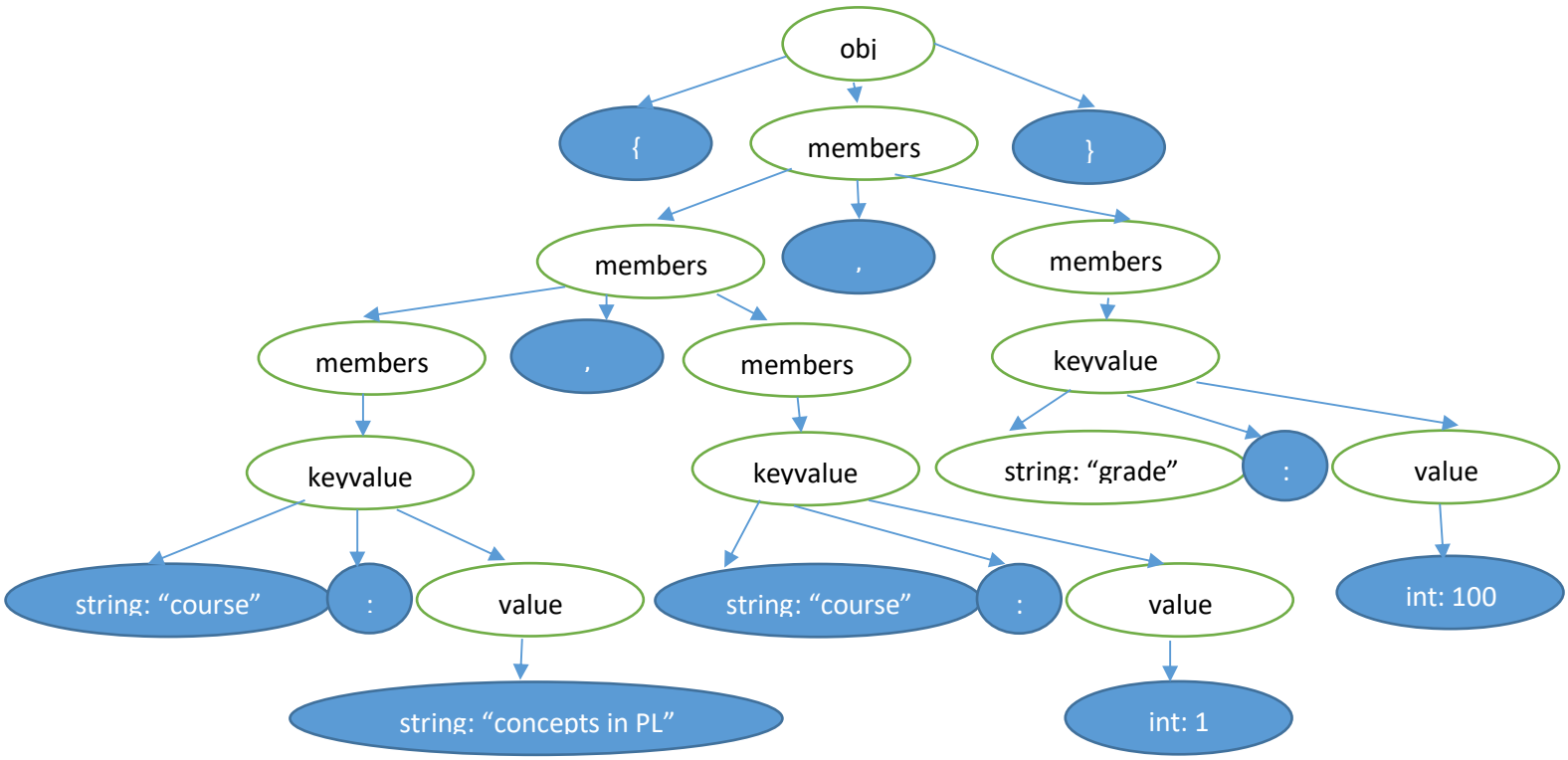


שאלה 3

א. המילה הבאה **נמצאת** בשפה:

```
{"course":"concepts in PL", "ex":1, "grade":100}
```

להלן עץ גזירה (הטוקנים צבועים בכחול):



ב. המילה הבאה **לא** נמצאת בשפה:

```
{"course":"concepts in PL", "ex":1, "grade":{100}}
```

<u>טענה</u>: בכל המילים בשפה, הטרמינלים היחידים שיכולים להופיע אחרי הטרמינל } הם: string }.

על פי הטענה, אותה נסביר מיד, המילה הנתונה לא בשפה מכיוון שמופיע בה הטרמינל int מיד לאחר הטרמינל }.

<u>הסבר לטענה:</u>

א. מכלל הגזירה הראשון אפשר לראות שהטרמינל { יכול לבוא מיד אחרי הטרמינל }.

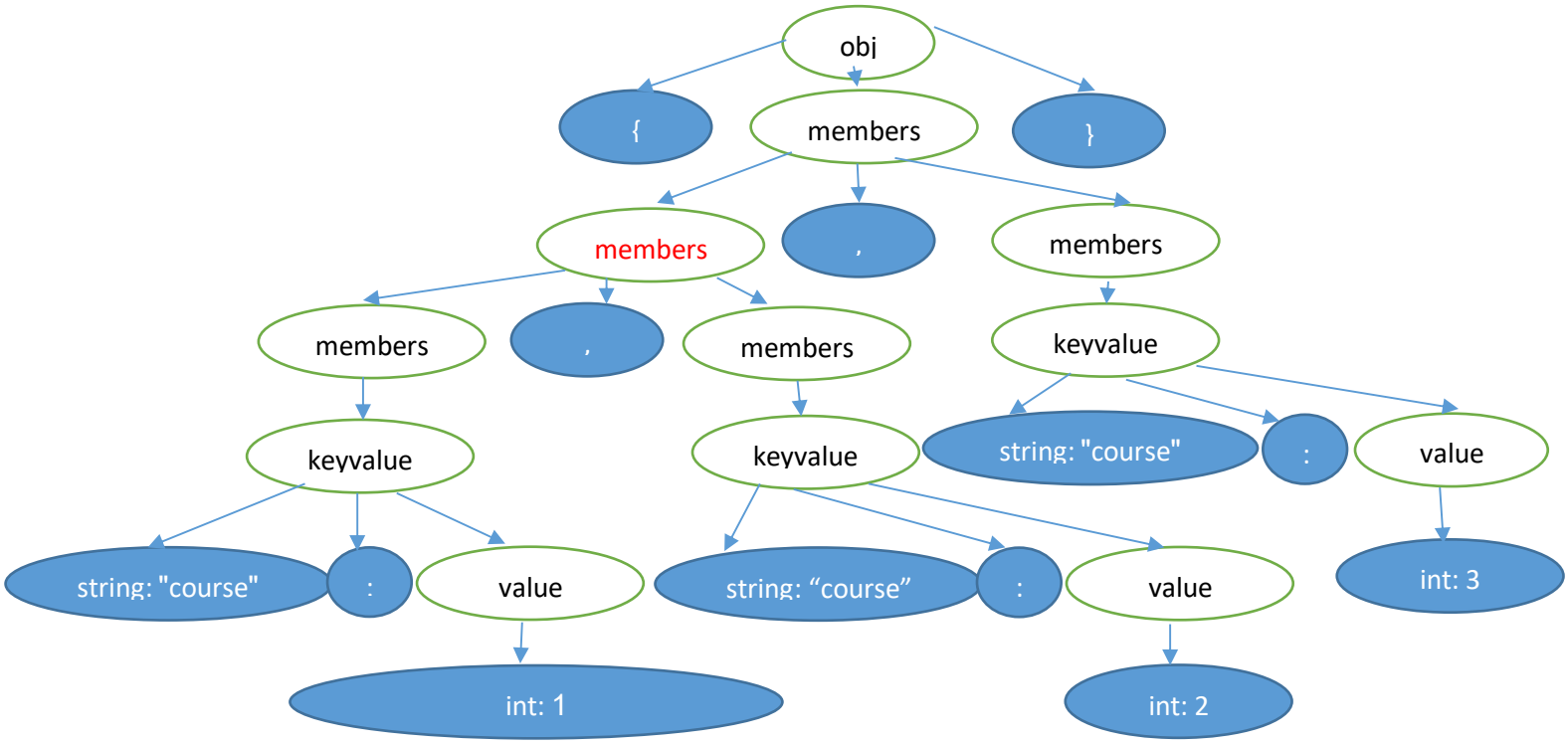
ב. הכלל הנוסף היחיד שגוזר את הטרמינל } הוא הכלל השני. על פיו, לאחר } יבוא המשתנה members.
מכיוון שאף כלל לא גוזר אפסילון, נקבל שאחרי } יכולים לבוא רק הטרמינלים ששייכים לסט first(members) (בנוסף על { שראינו מקודם).
נחשב את first(members): ל-members יש שני כללי גזירה – באחד רקורסיה שמאלית, ובשני הוא גוזר את המשתנה keyvalue. לכן first(members) = first(keyvalue). על פי הכלל היחיד של keyvalue נקבל ש-string הוא הטרמינל היחיד שנמצא ב-first(keyvalue). לכן הוא גם היחיד שנמצא ב-first(members).

שאלה 4

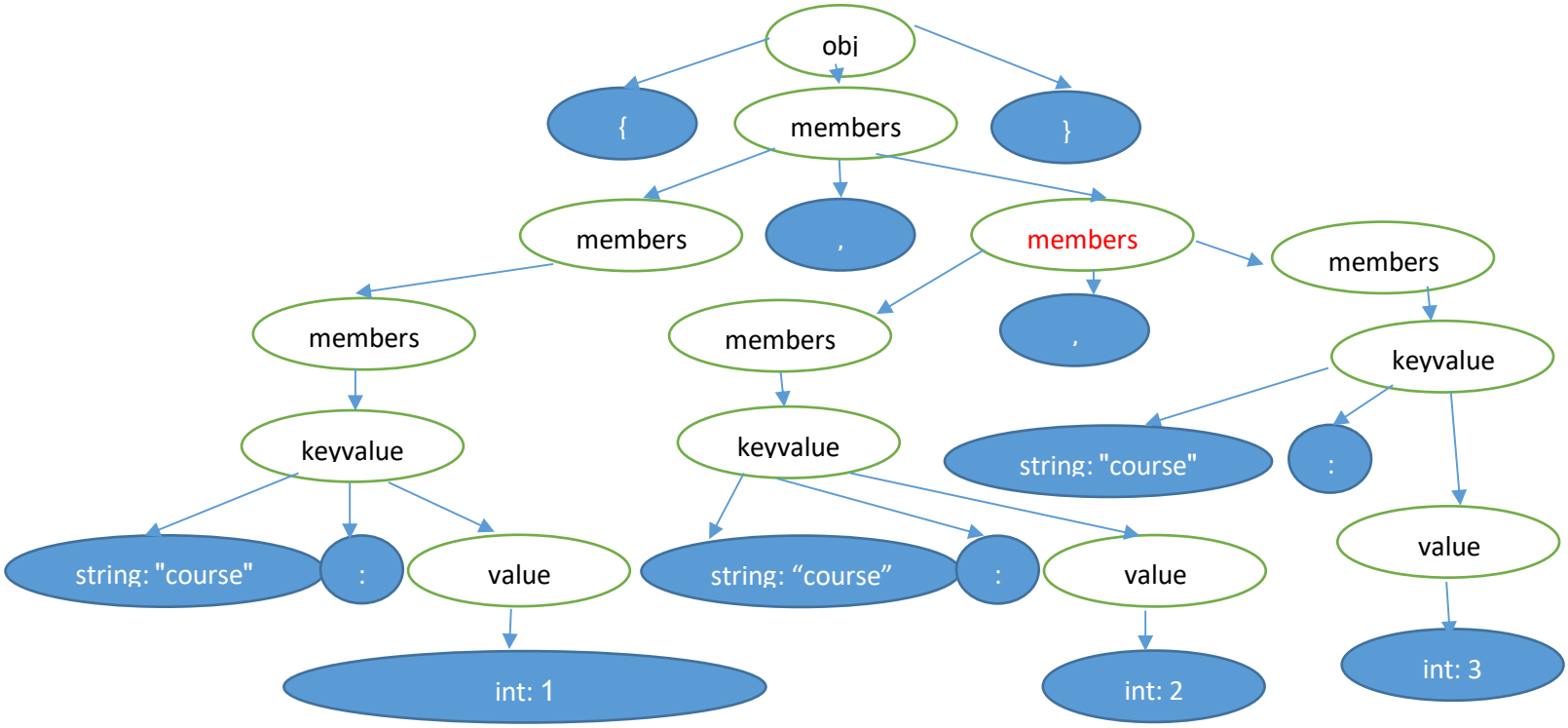
נראה שני עצי גזירה שונים עבור המילה: {"course" : 1 , "course" : 2 , "course" : 3}

(כמובן שזה נכון עבור כל מילה מהצורה {string : int , string : int , string : int})

עץ 1

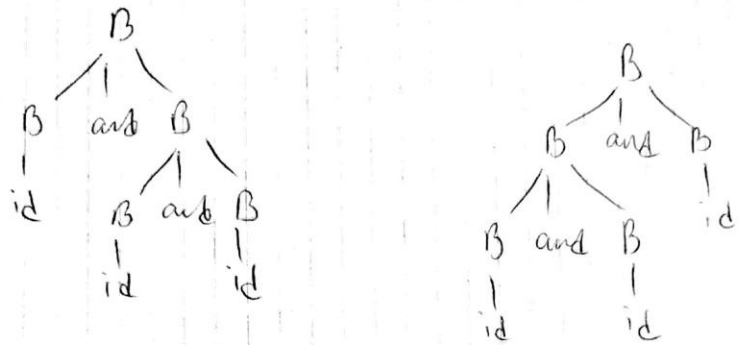


עץ 2



שאלה 1:

א. תאור המילה id and id and id נותן 2 עצי גזירה:



ב.

$B \rightarrow B \text{ and } T \mid T$

$T \rightarrow \text{not } T \mid F$

$F \rightarrow id \mid (B)$

ג. הדקדוק איננו LL(1) משום שקבוצות ה-select תאור שני כללים נחתכות:

$select(B \rightarrow B \text{ and } T) = \{not, (, id\}$

$select(B \rightarrow T) = \{not, (, id\}$

הדקדוק החדש:

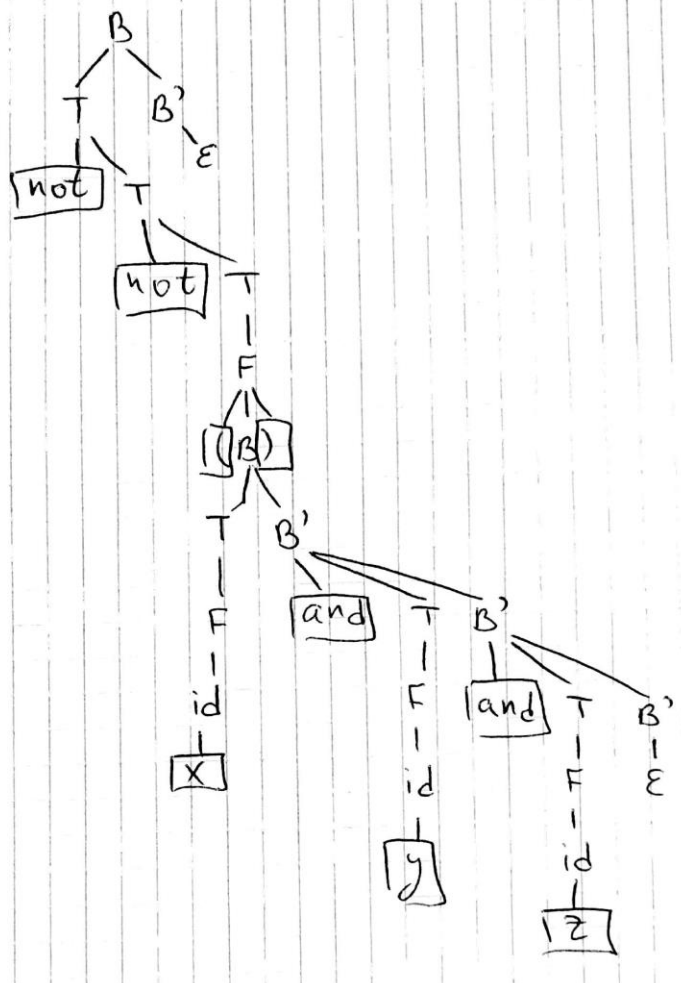
$B \rightarrow T B'$

$B' \rightarrow \text{and } T B' \mid \varepsilon$

$T \rightarrow \text{not } T \mid F$

$F \rightarrow id \mid (B)$

3. עץ גזירה עבור not not (x and y and z):



שאלה 5

A. נשים לב שקבוצות ה-select נחתכות ולכן הדקדוק הוא לא LL1.
הקבוצות הן:

```
select(members -> keyvalue) = set(['STRING'])
select(members -> members COMMA members) = set(['STRING'])
```

- ניתן לראות שSTRING משותף לשתי הקבוצות ובפרט החיתוך אינו ריק.

```
select(obj -> LB RB) = set(['LB'])
select(obj -> LB members RB) = set(['LB'])
```

- ניתן לראות שLB משותף לשתי הקבוצות.

B. הדקדוק החדש מגדיר הוספת פסיק משמאל בלבד ובכך הופך את הדקדוק שלנו לחד משמעי. הוא אינו LL1 עדיין, משום שקבוצות של הselect עדיין נחתכות:

```
select(obj -> LB RB) = set(['LB'])
select(obj -> LB members RB) = set(['LB'])
```

```
select(members -> keyvalue) = set(['STRING'])
select(members -> members COMMA keyvalue) = set(['STRING'])
```

C. יש רקורסיה שמאלית: הורדתי אותה ע"י החלפת כללי הגזירה הבאים:

```
# members -> keyvalue
# members -> members, keyvalue
```

בכללי הגזירה האלו:

```
# members -> keyvalueMembersTag
# MembersTag -> , keyvalueMembersTag
# MembersTag -> epsilon
```

יש left factoring משום ששני הכללים הבאים מתחילים ב: "{":

```
# obj -> {}
# obj -> {members}
```

תיקנתי את זה ע"י החלפת הכללים הנ"ל בכללים הבאים:

```
# obj -> {E}
# E -> members
# E -> epsilon
```