בס"ד

שפות תכנות תרגיל 1

חלק ב'

שאלה (1):

א) צ"ל: לכל ביטוי ב-E_1, מספר ה-"(" שווה למספר ה-")".

הוכחה:

נוכיח באינדוקציה מבנית.

בסיס- עבור איבר הבסיס ε (מחרוזת ריקה) כמובן שמספר ה-"(" שווה למספר ה-")", (מתקיים באופן ריק שניהם שווים ל-0). (*) למטה מקרה נוסף

הנחה- נניח כי עבור מילה כלשהי בשפה, $a \in E_1$, מתקיימת הטענה.

צעד- לפי ההנחה נתון כי מספר ה-"(" ב-a זהה למספר ה-")" ב-a. צריך להוכיח כי עבור (a) מס' ה-"(" זהה למספר ה-")".

קל לראות כי הוספנו 1 למספר ה-"(",
ובנוסף הוספנו 1 למספר ה-")".
ולכן סה"כ מספר ה-"(" ב-(a) נותר זהה למספר ה-")" ב-(a).
(כי פשוט הוספנו 1 לשניהם, ולפני ההוספה הם היו זהים) טריוויאלי.

ולכן סה"כ ע"י אינדוקציה מבנית הוכחנו את הטענה עבור השפה.

מ.ש.ל.

(*) מקרה בסיס נוסף, עבור הבסיס id, הוא לא מכיל כלל סוגריים, ולכן כמו ב-ε גם עבורו מספר ה-"(" שווה למספר ה-")" שווה ל-0.

בס"ד

שפות תכנות תרגיל 1

חלק ב

שאלה (1) - המשך:

ק) טענה: ב-E_2, מספר ה-"(" שווה למספר ה-")" בכל מילה.

הוכחה:

נוכיח באינדוקציה מבנית.

בסיס - עבור ε ו-id, ברור כי מתקיימת הטענה, כיוון שבשניהם אין כלל סוגריים. (ולכן שווה ל-0).

הנחה - נניח כי $x \in E_2$. כלומר מספר הסוגריים בו מאוזן.

צעד - נוכיח כי: $x \underset{E_2}{\Rightarrow} (R \underset{E_2}{\Rightarrow} (x) \in E_2$

קל לראות שכיוון ש-x בעל מס' סוגריים מאוזן, אז כמובן שגם (x) יהיה מאוזן. ולכן עבור המקרה של הגזירה הזאת זה מתקיים.

נוכיח כי: $x \underset{E_2}{\Rightarrow} (R \underset{E_2}{\Rightarrow} () \in E_2$

זה טריוויאלי, כי יש 1 "(" ועוד 1 ")" ולכן מאוזן.

ולכן סה"כ לכל $x \in E_2$, מספר ה-"(" זהה למספר ה-")".

והוכחנו זאת באינדוקציה מבנית.

כנדרש.

Scanned with CamScanner

שפות תכנות תרגיל 1

חלק ב

שאלה 1 - המשך:

ג) צ"ל: E_1 ו- E_2 מגדירים את אותה השפה.
כלומר $E_1 = E_2$.

הוכחה: נוכיח ע"י הכלה דו כיוונית.

$(\subseteq)$ $E_1 \subseteq E_2$.

כלומר, נראה כי $\forall x \in E_1 \Rightarrow x \in E_2$.

בסיס - עבור איברי הבסיס של E_1, נקבל: $\varepsilon, id \in E_2$
כיוון ששניהם מה הסוגרים מאותו. (ס)

הנחה - נניח כי $x \in E_2 \to x \in E_1$.

צעד - בהינתן $x \in E_1$, צ"ל $(x) \in E_2$.

לפי ההנחה $x \in E_2$, ולכן לפי כללי הגזירה מתקיים:

$x \underset{E_2}{\Rightarrow} (R \underset{E_2}{\Rightarrow} (E_2) ::= (x)$

וקיבלנו כי $(x) \in E_2$.

$(\supseteq)$ $E_1 \supseteq E_2$.

בסיס - עבור איברי הבסיס של E_2, נקבל: $\varepsilon, id \in E_1$
טריוויאלי.

הנחה - נניח כי $x \in E_2 \to x \in E_1$.

צעד: בהינתן $x \in E_2$, צ"ל: $(x) \in E_1$.

לפי ההנחה $x \in E_1$, ולכן לפי כללי הגזירה מתקיים:
$(x) \in E_1$

כנדרש.

סה"כ הוכחנו הכלה דו-כיוונית ולכן קיבלנו כי $E_1 = E_2$.
כנדרש.

בס"ד

שפות תכנות תרגיל 1

חלק ב:

שאלה (2)

א) a.

obj
↓
{member}
↓
{member, member}
↓
{member, member, member}
↓
{keyvalue, keyvalue, keyvalue}
↓
{string: value, string: value, string: value}
↓
{string: string, string: int, string: int}
↓
{"course": "consept in PL", "ex": 1, "grade": 100}

וקיבלנו אכן את המילה a כנדרש, ולכן a נמצאת בשפת הדקדוק הנ"ל והראנו עץ גזירה.

b. נניח כי אין עץ גזירה עבור המילה b.
נב"ש קיים עץ גזירה עבור b.
לפי כללי הגזירה, הדרך היחידה לקבל סוגריים מסולסלים הוא מהכלל: obj→{member}. לכן לפי ההנחה בשאלה, מתקיים:
$member \Rightarrow^{*}_{G} 100$.
בהכרח בשלב הבא בגזירה נעבור ב- keyvalue, כלומר:
$member \Rightarrow_{G} keyvalue \Rightarrow^{*}_{G} 100$
כעת הגזירה היחידה האפשרית של keyvalue לפי כללי הגזירה היא לגזירה string : value, כלומר בהכרח הוא תכיל את ":", אך 100 לא מכיל ":", בסתירה להנחה.
ולכן לא קיימת מילה בשפה שמכילה את {100}, ולכן בפרט b אינה שייכת לשפה.

מ.ש.ל.

שפות תכנות תרגיל 1

חלק ב:

שאלה ② - המשך

ב) נראה שהדקדוק הוא רב משמעי.

ראינו בסעיף הקודם עץ גזירה עבור המילה a הנתונה.

כעת נראה עץ גזירה נוסף שונה עבורה.

obj

{member}

{member , member}

בעץ הקודם גזרנו את ה-member השמאלי ל-2, וכאן את הימני.

{member , member, member}

{keyValue, keyValue, keyValue}

כמו בסעיף א'

⋮ ⋮ ⋮

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

וקיבלנו כי ל-a יש 2 עצי גזירה שונים, ולכן הוכחנו כי הדקדוק הוא רב משמעי.

כנדרש.

שפות תכנות תרגיל 1

חלק א'

שאלה 2 :

ב) לא היינו משנים בקוד שכתבנו בסעיף א' שום דבר, כיוון שהקוד עובד גם עבור עץ חיפוש בינארי שאיבריו הם float.

שאלה 3

א. נפריך / לא נוכל להוכיח שלכל מופע exp מטיפוס bool_expr מתקיים:

$$num\_of\_vars(exp) = num\_of\_connectives(exp) + 1$$

נפריך עבור הביטוי Not (And (x,y))

$num\_of\_vars\ Not(And(x,y)) =$

$num\_of\_vars\ And(x,y) =$

$num\_of\_vars\ x + num\_of\_vars\ y = 1+1=2$

$num\_of\_connectives\ Not(And(x,y)) =$

$= num\_of\_connectives\ And(x,y) + 1 =$

$num\_of\_connectives\ x + num\_of\_connectives\ y + 1 + 1$

$= 0+0+2$

$2 \neq 2+1$

לכן נראה להוכיח שלא מתקיים הנדרש והפרכנו.

<u>המשך שאלה 3</u>

ב. לכל שפה הגו exp מטיפוס bool_expr בו לא מופיע Not

מתקיים: $num\_of\_vars(exp) = num\_of\_connectives(exp)+1$

<u>נוכיח באינדוקציה מבנית</u>

מקרה בסיס:

עבור איבר הבסיס ___ מתקיים ש

$num\_of\_vars(\_) = 1 = 0+1 \quad num\_of\_connectives(exp)+1$

הנחה- נניח שעבור הביטויים e_1, e_2 כלשהם מטיפוס bool_expr שאין בהם Not

(ייתכן ש $e_1 = e_2$)

מתקיים ש

$i = \{1,2\} \quad num\_of\_vars(e_i) = num\_of\_connectives(e_i)+1$

:

נוכיח עבור הביטויים מהצורה $And(e_1,e_2)$! $Or(e_1,e_2)$

יהי e מהצורה $e = And(e_1,e_2)$

אז מתקיים:

$$num\_of\_vars\ e = num\_of\_vars\ And(e_1,e_2) =$$

$$num\_of\_vars\ e_1 + num\_of\_vars\ e_2$$

$$\underset{\text{הנחת האינדוקציה}}{=} num\_of\_connectives\ e_1 + 1 + num\_of\_connectives\ e_2 + 1$$

$$num\_of\_connectives\ And(e_1,e_2) =$$

$$num\_of\_connectives\ e_1 + num\_of\_connectives\ e_2 + 1$$

נשים לב שהם שווים

$$num\_of\_vars\ And(e_1,e_2) = num\_of\_connectives\ And(e_1,e_2)+1$$

ט e מהצורה $e = Or(e_1,e_2)$ באופן דומה כמו ב And כיון שמקיימת
את הדבר.