

# הסדרים השונים ב $\lambda$ -calculus

את כל הסדרים השונים ניתן לתאר כ Operational Semantics על בסיס הכללים הבאים:

$$(\lambda x. t_{12}) t_2 \rightarrow [x \mapsto t_2]t_{12} \quad (\text{E-APPABS})$$

$$\frac{t_1 \rightarrow t'_1}{\lambda x. t_1 \rightarrow \lambda x. t'_1} \quad (\text{E-ABS})$$

$$\frac{t_1 \rightarrow t'_1}{t_1 t_2 \rightarrow t'_1 t_2} \quad (\text{E-APP1})$$

$$\frac{t_2 \rightarrow t'_2}{t_1 t_2 \rightarrow t_1 t'_2} \quad (\text{E-APP2})$$

כל סדר מוגדר ע"י הפעלה של חלק מהכללים, ובמידת הצורך עם סדר קדימויות.

## מנטיקת **Call by value - eager - strict**

מפעילים את הכללים ללא E-Abs לפי סדר הקדימויות הבא:

1. E-App1

2. E-App2

3. E-AppAbs

ניסוח אלטרנטיבי ע"י הגדרה של קטגוריה סינטקטית נוספת של ערכים וללא סדר קדימויות מופיע בשקפי ההרצאה והתרגול.

## מנטיקת **Call by name - lazy**

מפעילים רק את הכללים E-AppAbs ו-E-App1. אין צורך לקבוע קדימויות מאחר שבמקרה זה שני הכללים פועלים על איברים שונים סינטקטית (חישבו מדוע).

## מנטיקת **Normal order**

מפעילים את כל ארבעת הכללים לפי סדר הקדימויות הבא:

1. E-AppAbs

2. E-App1

3. E-App2

4. E-Abs

ניסוח אלטרנטיבי: תמיד בוחרים redex חיצוני ביותר ושמאלי ביותר.