

תרגול 6

4 במאי 2017

הגדרה 0.1 שלמות ונאותות

נאותות: $\vdash \subseteq \models$

שלמות: $\models \subseteq \vdash$

משפט 0.2 מערכת הוכחה HPC נאותה ושלמה (ביחס לסמנטיקה קלאסית)

מסקנה 0.3 $\models$ קומפקטי

תרגיל: הוכח כי שתי ההגדרות לקומפקטיות $\models$ שקולות

1. לכל תורה T ופסוק $A: T \models A \iff \text{יש תת קבוצה סופית } T' \subseteq T \text{ כך שמתקיים } T' \models A$
2. לכל תורה $T: T \models \text{ספיקה} \iff \text{כל תת קבוצה סופית } T' \subseteq T \text{ ספיקה.}$

פתרון: $1 \Rightarrow 2$

ננסח את 1 מחדש "לכל תורה T ופסוק $A: T \models A \iff T \models A$ לכל $T' \subseteq T$ סופית"
 תהי תורה T , נשים לב כי עבור $A = p_1 \wedge \neg p_1$ מתקיים $\Delta \models A \iff \Delta \models \text{ספיקה.}$
 נציב את T או T בנוסח החדש של 1 ונקבל את 2 עבור T או T שרירותית ולכן סיימנו.
 $2 \Rightarrow 1$

....
 ■

תרגיל: נאמר כי תורה היא דו־ספיקה אם היא מסתפקת על ידי השמה ρ עבורה $\forall k. \rho(p_{2k}) = f$
 הראו שקומפקטיות (2) מתקיימת עבור דו־ספיקות.

רעיון: נשתמש במשפט הקומפקטיות על ידי כך שנתרגם דו־ספיקות לספיקות על ידי הקבוצה $\neg p_{2k} = \{\neg p_2, \neg p_4, \dots\}$
 "לכל תורה T , $T \models \text{ספיקה} \iff T \cup \neg p_{2k} \models \text{ספיקה.}$ "

פתרון: כיוון אחד ברור, נראה את השני.

תהי תורה T עבורה כל $T' \subseteq T$ סופית היא דו־ספיקה.
 תהי $T' \subseteq T \cup \neg p_{2k}$ סופית, אז $T' \setminus \neg p_{2k} \subseteq T$ סופית.
 לכן היא דו־ספיקה ולכן $T' \subseteq T' \setminus \neg p_{2k} \cup \neg p_{2k} \models \text{ספיקה.}$
 אז כל תת־קבוצה סופית של $T \cup \neg p_{2k}$ ספיקה ולכן מקומפקטיות היא ספיקה, לכן T דו־ספיקה.

תרגיל: כל סדר חלקי ניתן למלא.

$(s, <)$ סדורה חלקית אם

$$\forall a \in S \ a < a$$

$$\forall a, b \in S \ a < b \wedge b < a \rightarrow a = b$$

$$\forall a, b, c \ a < b \wedge b < c \rightarrow a < c$$

(סדורה מלא: אומר גם $\forall a, b \in S \ a < b \vee b < a$)

פתרון: נתאם לכל יחס בינארי $(\mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N})$ השמה:

$$\rho(p_{a,b}) = \begin{cases} t & a * b \\ f & \text{else} \end{cases}$$

קידוד: נבנה קבוצה כך שתהיה ספיקה אמ"מ ניתן למלא את הסדר החלקי $(S, <)$ איחוד של הבאים:

$$\tilde{C} = \{P_{a,b} | a < b\}$$

כל השמה שתספק את $\tilde{C}$ מתאימה ליחס המכיל את $<$

$$\tilde{R} = \{P_{a,b} | a = b\}$$

$$\tilde{A} = \{\neg P_{a,b} \vee \neg P_{b,a} | a \neq b\}$$

$$\tilde{T} = \{P_{a,b} \wedge P_{b,c} \rightarrow P_{a,c} | a, b, c\}$$

$$\tilde{F} = \{P_{a,b} \vee P_{b,a} | a, b\}$$

כולמר עלינו להראות

$$\tilde{S} = \tilde{C} \cup \tilde{R} \cup \tilde{A} \cup \tilde{T} \cup \tilde{F}$$

ספיקה.

תהי תת קבוצה סופית $\tilde{S}' \subseteq \tilde{S}$ ב- $\tilde{S}'$ מופיעים רק כמות סופית של אינדקסים, נסמן $S' \subseteq S$. אז $(s', <')$ סדר חלקי. סדר חלקי סופי קל למלא (תרגיל לקורא לחשוב על האלגוריתם), ולכן $\tilde{S}'$ ספיקה. מקומפקטיות $\tilde{S}$ ספיקה.

השפה הפורמלית: תחילה בוחרים מילון:

1 סימני קבועים

2 סימני פונקציה

3 סימני יחס

כאשר סימני פונקציה ויחס מגיעים עם מספר המתאר את כמות הכניסות השפה:

1 (אינסוף) משתנים

2 קשרים לוגיים

3 כמתים $\forall \exists$

4 סימני עזר $()$,

5 הסימנים מהמילון

הצגות: (1) "יש כלב חכם"
 ניקח מילון עם יחסים חד־מקומיים $smart, dog$

$$\exists x (dog(x) \wedge smart(x))$$

(2) "כל חתול חכם"

$$\forall x (cat(x) \rightarrow smart(x))$$

(4) (כן 4 זה לא טעות) "יש למלכה בדיוק שני יורשים"
 ההצגה היא $\varphi \wedge \psi$

$$\varphi = \exists x \exists y (heir(q, x) \wedge heir(q, y) \wedge x \neq y)$$

$$\psi = \forall x \forall y \forall z (heir(q, x) \wedge heir(q, y) \wedge heir(q, z) \rightarrow x = y \vee y = z \vee z = x)$$

מתקיים

$$F : D^n \rightarrow D$$

$$R : D^n \rightarrow \{t, f\}$$

נוסחאות חוקיות:
 שמות עצם מוגדרים אינדוקטיבית
 בסיס: כל משתנה וכל קבוע
 צעד: $f(t_1, \dots, t_n)$ כאשר f פונקציה N מקומית, $t_1, \dots, t_n$ שמות עצם.
 כל שם עצם מתקבל כך.
 נוסחאות חוקיות מוגדרות אינדוקטיבית
 בסיס:
 $R(t_1, \dots, t_n)$ כאשר R סימן יחס N מקומי $t_1, \dots, t_n$ ש"ע.
 צעד:
 אם A, B נוסחאות אז $(QxA), (A \circ B), (\neg A)$

$$\circ \in \{\vee, \wedge, \rightarrow, \leftrightarrow\} \quad Q \in \{\exists, \forall\}$$

x משתנה. כל נוסחה מתקבלת כך.
 גם כאן יש לנו משפט קריאה יחידה (הפעם גם לש"ע) ולכן ניתן להגדיר פונ' באופן אינדוקטיבי.