
אדמיניסטרטיבי

- אתר הקורס - <http://u.cs.biu.ac.il/~sarit/advai15.html>
- בסוף יש מבחן(חובה)
- תרגילים - 20% או 30%:
 - תרגילים בכתב
 - תרגיל תכנות - לכתוב מערכת שמשחקת איזשהו משחק. לא נצטרך להתעסק בתוכנה של המשחק עצמו, אלא רק לתכנת את הסוכן.
- אין ספר - מכסים הרבה נושאים - אבל כל נושא שנלמד יש איזה ספר שמכסה אותו בצורה הרבה יותר מפורטת, או מאמר שאפשר להסתכל עליו.

נושאים בבינה מלאכותית

- Machine Learning
- Decision Making
- פתרון בעיות אופטימיזציה
 - ייצוג ידע
 - לוגיקה
 - Planning
 - היסקים
 - Interferences
 - טיפול באי וודאות
 - מודלים
 - הרבה Data
 - טיפול באי וודאות

תזכורת - Propositional Logic

הבסיס של לוגיקה זה atomic sentences - עובדות פשוטות. בדרך כלל משתמשים באותיות בודדות (p, q), אבל לפעמים, כדי שבני אדם יבינו יותר טוב, משתמשים במילים.

• syntax: p, q - atomic sentences

• כמתים - $\wedge, \vee, \neg, \rightarrow$

• ביטויים (sentences) - $p \wedge q \rightarrow r$

semantics: מה נכון ומה לא נכון בכל מצב עולם:

p	q	$\neg p$	$p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	T	F
F	T	T	T	T
F	F	T	F	T

הכמת הכי מעניין הוא \rightarrow . צריך תמיד לזכור שאם p הוא שקר, $p \rightarrow q$ תמיד אמת. למשל: הביטוי $5 \text{ is even} \rightarrow 6 \text{ is odd}$ הוא אמת, כי "5 הוא זוגי" זה שקר, ולכן זה לא מעניין ש"6 אי זוגי" גם שקר. ביטויים הם satisfiable אם יש מצב עולם בו הם נכונים. יש שני סימנים מאוד חשובים בלוגיקה:

• \models ("models") - אם למשל יש לנו את Knowledge Base $KB = \left\{ \begin{array}{l} p \rightarrow q \\ \neg q \\ r \\ r \vee w \end{array} \right\}$, אז נאמר ש $KB \models P$

אם בכל העולמות בהם KB מתקיים, גם P מתקיים.

• \vdash ("infers") - כמו \models , אבל בהקשר של סמנטיקה. כלומר $KB \vdash P$ אם אפשר להוכיח את P מתוך הטענות ב KB .

במערכת לוגית "טובה" ("sound"), מתקיים:

$$KB \models \alpha \iff KB \vdash \alpha$$

כאשר: • הצד $KB \models \alpha \iff KB \vdash \alpha$ נקרא נאותות - כל מה שמצליחים להוכיח בהכרח נכון בעולם. • הצד $KB \models \alpha \implies KB \vdash \alpha$ נקרא שלמות - כל מה שנכון בעולם אפשר גם להוכיח. את הצד הזה הרבה יותר קשה להוכיח.

Sound inference rules

- כל השקילויות הלוגיות (למשל $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$ או $(\alpha \implies \beta) \equiv (\neg \alpha \implies \neg \beta)$)
- Modus Ponens (בלטינית - מצב מאשר): אם $\alpha \implies \beta$ וגם α אז בהכרח β
- And-elimination: אם $\alpha \wedge \beta$ אז בהכרח α

תזכורת - First Order Logic

proposition logic אפשר להביע דברים פשוטים (למשל "יורד גשם"), אבל מה אם רוצים להביע משפט כמו "כל יום בשנה יורד גשם"? נצטרך 365 משפטים ("ב1 בינואר יורד גשם", "ב2 בינואר יורד גשם"...).

בעוד שב propositional logic הפרימיטיביים שלנו הם רק עובדות, ב first order logic יש גם אובייקטים ויחסים בין האובייקטים.
syntax הוא הרחבה של propositional logic syntax. עדיין יש משפטים, אבל מוסיפים פרימיטיביים נוספים:

- קבועים - באמצעותם ניתן לתת שמות לדברים בעולם
- פונקציות - מיפוי מקבועים לקבועים אחרים
- פרדיקטים - מיפוי מקבועים לערכי אמת/שקר. למשל Teacher (Jerry, you)
- משתנים
- כמתים
- \forall ("לכל") ו \exists ("קיים")

למשל:

- "כל הנאנשים הם יונקים": $\forall x \text{human}(x) \implies \text{mammal}(x)$
- נשים לב שזה שונה מהמשפט $\forall x \text{human}(x) \wedge \text{mammal}(x)$, שאומר שכל דבר בעולם שלנו זה גם אדם וגם יונק.
- "יש ציפורים שעפות" - $\exists x \text{Bird}(x) \wedge \text{Fly}(x)$
- נשים לב ש $\exists x \text{Bird}(x) \implies \text{Fly}(x)$ חלש מדי - הוא יהיה נכון אם קיים משהו שהוא לא ציפור
- "כל הציפורים עפות" - $\forall x \text{Bird}(x) \implies \text{Fly}(x)$
- נשים לב ש $\forall x \text{Bird}(x) \wedge \text{Fly}(x)$ חזק מדי - הוא לא יהיה נכון אם קיים משהו שאינו ציפור
- צריך להיזהר מהחלפת הסדר בין "לכל" ו"קיים". $\forall x \exists y$ זה לא אותו דבר כמו $\exists y \forall x$. לדוגמה:
 - $\forall x \exists y \text{likes}(x, y)$ - כל אחד אוהב מישהו
 - $\exists y \forall x \text{likes}(x, y)$ - קיים מישהו אחד שכולם אוהבים
- אפשר להפוך "לכל" ל"קיים" ולהפך באמצעות שלילות:
 - $\forall x \text{sleep}(x)$ - כולם ישנים
 - $\exists x \neg \text{sleep}(x)$ - קיים מישהו שלא ישן
 - $\neg \exists x \neg \text{sleep}(x)$ - לא קיים מישהו שלא ישן - שקול לזה שכולם ישנים