מערכות מרובות סוכנים

סיכום הרצאה 3 – 037482502

מרצה: פרופ' משה טננהולץ

מתרגל: אלון אלטמן

מגישה: אלישבע שמש

<u>מודל רב יישויות – השגת יעדים משותפת</u>

בהינתן מודל רב יישויות ויעד לכל שחקן (מודל יכול להיות כללי מאד, עם אי-וודאות), נרצה למצוא תוכנית משותפת להשגת כל היעדים. ישנן שתי שאלות אשר עולות ומנחות.

שאלה 1: האם ניתן לחסום את אורך התוכנית להשגת היעד? כלומר, האם היא כריעה decidable? היא סופית? קיים אלגוריתם?

שאלה 2: במודלים ספציפיים יותר – האם הבעיה החישובית ניתנת לפתרון במרחב יעיל? כלומר, האם כמות הזכרון הנחוץ פולינומיאלי?

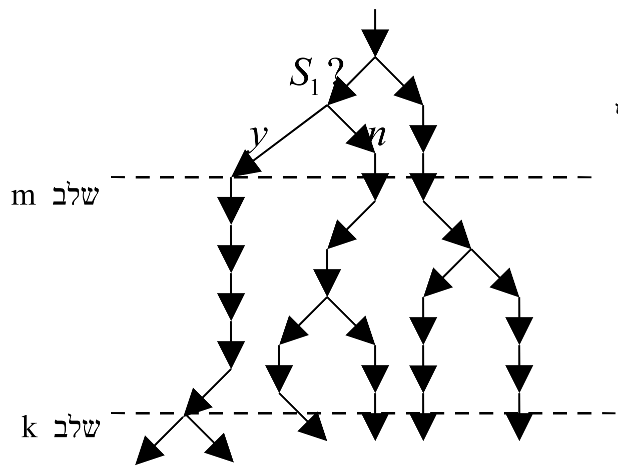
מטרתנו לבדוק את קיומה של תוכנית סופית להשגת היעדים (Comparative Goal Achievement). קיימת אפשרות שלא ניתן להשיג את כל הייעדים של כל היישויות, אולם אם קיימת דרך, אזי קיים גם אלגוריתם מתאים.

<u>סקיצת הוכחת קיום תוכנית סופית (בהנחה שקיימת תוכנית סופית) :</u>

ראשית: יש לשים לב שמכיוון שתוכניות הסוכנים ידועות לכל (common knowledge), ניתן לתרגם כל אובסרבציה (עובדה) שנלמדת, לעובדה על הקונפיגורציה ההתחלתית. כלומר, מכיוון שאני יודעת הכל על המצב הנתון, ועל כל התוכניות של כל הפרטים, הנעלם היחיד הינו הקונפיגורציה ההתחלתית.

בלי הגבלת הכלליות – ניתן להסתכל על התוכנית של סוכן כעל עץ החלטה, כאשר התפצלות בעץ תתקיים רק כאשר עובדה נלמדת. אם קיימת סיטואציה כך שלאורך כל העצים (תוכניות) אין פיצול באף ענף לאורך מספר גבוה או שווה למספר

הקונפיגורציות $|C|$



(כאשר $|C|$ מתארת את מספר הקונפיגורציות במודל הבסיסי – ללא מצבי ידע), אזי ניתן למצוא תוכנית קצרה יותר. ניתן לראות בציור, אם $k-m>|C|$ אזי ביקרנו בקונפיגורציה כלשהי יותר מפעם אחת, וניתן לצמצם את העץ.

לכן בשל $|C|$ ו $|C_0|$ ($|C_0|$ מספר הקונפיגורציות ההתחליות) סופיות, נקבל את מה שרצינו להוכיח.

מודל עשיר המוכל במודל האמור הינו המודל של אינפורמציה מלאה $C_0=\{C_0\}$. למקרה של אינפורמציה מלאה קוראים C-CGA (Complete information Cooperative Goal Achievement). במקרה זה אם קיימת תוכנית להשגת היעדים, אזי הבעיה ניתנת לחישוב במרחב פולינומיאלי. במקרים כאלו ניתן אלגוריתם שהינו לא דטרמניסטי, כלומר יכול לבצע ניחושים הניתנים לכתיבה במרחב פולינומיאלי, כאשר תחת ניחושים אלו – לא ננצל יותר ממרחב פולינומיאלי.

בקונפוגורציה ההתחלתית ננחש מה הפעולה המשולבת של הסוכנים, ואז נחשב מה הקונפיגורציה הבאה. לאחר מכן נחשב את הפעולה המשולבת הבאה וכן הלאה. ניתן לראות שמספיקים לנו $|C|$ ניחושים כדי להגיע לכל יעד, אם קיימת דרך כזאת, כלומר אם הבעיה כריעה.

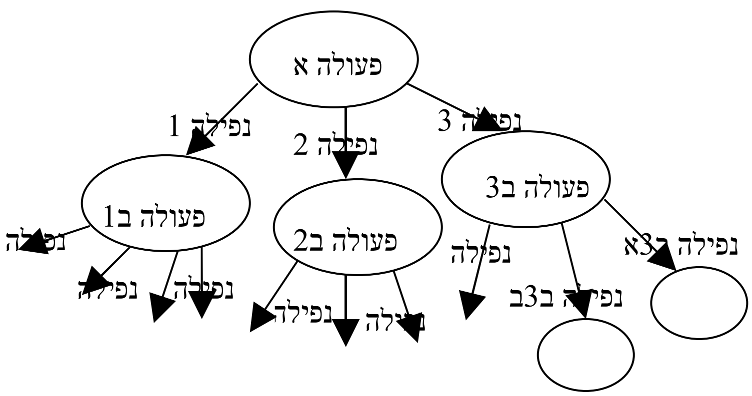
אלגוריתם כזה הינו אלגוריתם הנמצא ב NPSPACE (ראה נספח).

קיים משפט האומר כי $NPSPACE=PSPACE$, לכן קיים אלגוריתם גם $PSPACE$, כלומר אלגוריתם דטרמיניסטי במרחב הפולינומיאלי.

<u>דוגמא נוספת:</u> מקרה נוסף הינו F-CGA, שמדבר על מקרה שבו לכל היותר t משתמשים יכולים ליפול. כלומר: בסיטואציה מסויימת, לא ידועה מראש, משתמש יכול לבצע פעולה כלשהי וליפול. כדי לציין שמשתמש הינו במצב נפילה, נסמן זאת על ידי זה שהוא מבצע את הפעולה ה null : Λ. קיימת כאן הנחה שסוכן שנפל לא יחזור. נניח שסוכנים יכולים לתאם את פעולתם ולראות את כל המצב. דבר זה מייצר לנו בעיה של מציאת איסטרטגיה "מנצחת" נגד הנפילות (כלומר, קיימת פעולה לסוכנים כך שלכל נפילה קיימת פעולה לסוכנים, כך שלכל נפילה קיימת פעולה לסוכנים... שתוביל ל"נצחון". ($\exists behaviour \forall fall \exists behaviour \forall fall ...$).

במקום לייצר את כל העץ, ניתן לתת אלגוריתם ב $NPSPACE$, כך שבמקום התייחסות ל $\forall$, נעבור על הנפילות האפשריות, כאשר האפשרות "אין נפילה" תהיה האפשרות האחרונה. למשל, ניתן לייצור משהו ב $NPSPACE$, אשר נראה כך:

על פי הדיאגרמה הזאת, מוצאים מהן הפעולות האופטימליות עבור נפילה ב 3א, ולאחר מכן חוזרים לפעולה ב 3, ברדוקציה, ובודקים עבור נפילה ב 3ב, וכן הלאה.



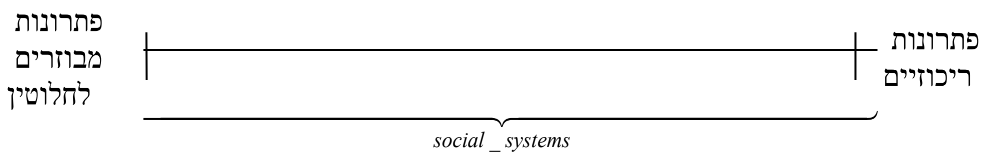
עד כה ראינו:

1) הצגת ידע קנוני באמצעות מודלים רבי יישויות.

2) ראינו שבמערכת שכזאת, כאשר מסתכלים בנקודת המבט של מתכנן מרכזי, מספיק להסתכל על התוכניות הסופיות.

3) באמצעות הטכניקה שהדגמנו, בעיות טיפוסיות בהקשר של בחירת דרכים להשגת יעדים ניתנות לפתרון ב PSPACE (ראה נספח).

עד כאן התייחסנו לשתי קצוות של אותה הבעיה. ניתן להתייחס לספקטרום הפתרונות האפשריים כמקביל לטווח בין פתרונות ריכוזיים – בהן הכל ידוע – לבין פתרונות מבוזרים לחלוטין – בהן כל סוכן פועל לבדו. לטווח זה נקרא social systems.



כאשר יעדי הסוכנים אינם ידועים מראש, או אין מתכנן מרכזי, אזי כל סוכן פועל לבדו, הסוכן מוגבל למערכת חוקים והוא יודע ששאר הסוכנים גם מוגבלים למערכת החוקים. כלומר, ניתן לבצע רדוקציה מבעית ה multi-agent לבעית ה single-agent, כך שכל סוכן פועל לבד – בהתאם לחוקיות חלקית.

קיימים שני שלבים:

1) חוקים חברתיים (אילוצים) הנקבעים (לפני ריצת התוכנית).

2) כל סוכן יודע שהוא חייב להתנהג על פי החוקים. כמו כן, הוא יודע שכל יתר הסוכנים מתנהגים על פי חוקים אלה.

<u>נציג דוגמא:</u>

הנחות:

Grid $n \times n$

m robots

For simplicity $m \le n-1$

Discrete moves

No perception (a robot can not see the other robots)

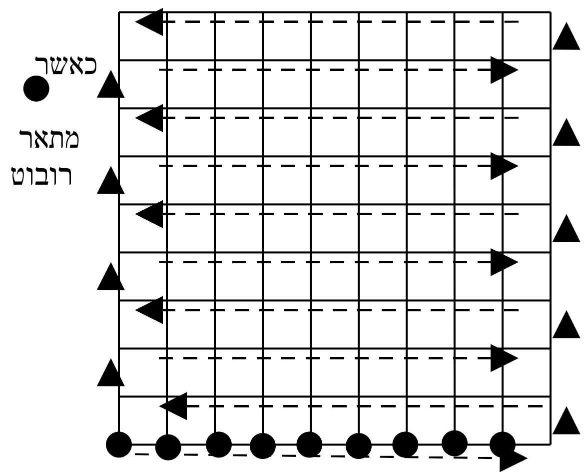
No communication

היעדים:

Avoid collisions (safety)

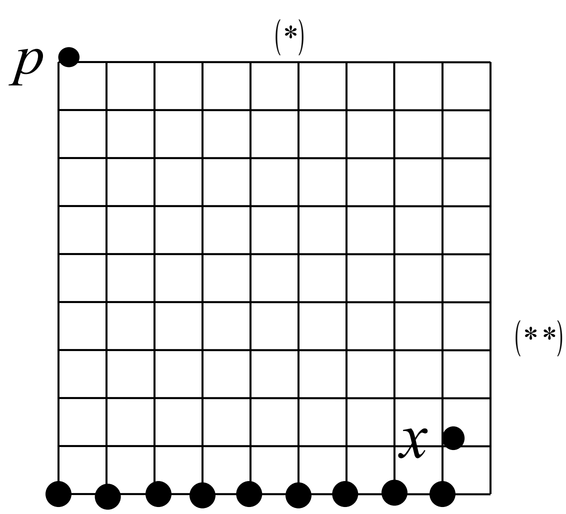
Enable goal achievements (liveness).

ניתן לפתור בעיה זו בעזרת חוקים חברתיים אשר יהוו חוקי תנועה. פתרון בעזרת multi-agent model יהיה קשה פרקטית. בהינתן אוסף ההגבלות על התנועה האפשרית, וציותם של הסוכנים לחוק החברתי, נקבל אי-התנגשויות (safety). אם תוך כדי ההתנהגות החברתית הגיע הסוכן ליעד, אזי יש לו תוכנית חברתית (המצייתת לחוקים) שמבטיחה הגעה ליעד כל עוד גם הסוכנים האחרים מצייתים לחוקים החברתיים.



חוק חברתי נאיבי יוביל את כל הסוכנים דרך כל נקודה אפשרית על ה grid על ידי מעבר דרך החצים המקווקווים (מסלול המילטון). חוק זה אינו מעניין, שכן הוא אינו מאפשר לסוכנים כל בחירה או חופש. כל סוכן יגיע ליעדו תוך $O(n^2)$.

לעומת זאת, חוק חברתי יעיל יותר, יוביל את כל הסוכנים ליעדם תוך $O(n)$. נניח שכל הסוכנים נמצאים בשורה התחתונה. הסוכנים ינועו עד x, ומשם ינועו במסלול – על פי בחירתם – ל p (סביר להניח שיבחרו לעבור דרך יעדם), כאשר הם חייבים לבחור באחד המסלולים הקצרים ביותר, בלי לעבור דרך השורה $(*)$, או העמודה $(**)$. אחרי שהם הגיעו ל p, הם יחזרו חזרה ל x, דרך השורה $(*)$ והעמודה $(**)$.



חוק חברתי זה מאפשר לכל סוכן להגיע לכל נקודה אפשרית, פרט לשורה התחתונה.

קל לראות שאם אין התנגשות אזי ניתן להגיע לכל נקודת יעד תוך $O(n)$. ה safety מתקיים מכיוון שהמרחק בין כל שתי נקודות נשאר תמיד קבוע, ולכן לא תהיה התנגשות.

כאשר קובעים מהירות קבועה לכל סוכן, ניתן לבחור

כל גרף דו-קשיר עם משקולות, ולקבל תוצאה עם

Safety ועם liveness.

<u>הבעייתיות עם הגישה (Social Systems):</u>

נניח שקיימים כמה יעדים אפשריים, כלומר היעד כולל כמה מצבי מכונה. חלק מהפעולות נקבעות כאסורות, וכל הסוכנים מקבלים הגבלות אשר משותפות לכל

הסוכנים, או פרטניים. הבעיה הבסיסית נקראת "מסלול הזהב" "The Golden Mean Problem".

ניקח שני סוכנים כאשר לכל סוכן יש כמה יעדים אפשריים. לכל סוכן יש את קבוצת האסטרטגיות S :

$S=\{S_1,\ldots,S_n\}$, וקבוצת היעדים $G=\{g_1,\ldots,g_n\}$

כאשר g מסמל יעד טוב (good), ו b מסמל פתרון רע (bad).

$$\begin{array}{c|cccccc}
I \backslash II & S_1^{II} & S_2^{II} & . & . & . & S_n^{II} \\
\hline
S_1^{I} & g & b & g & . & . & g \\
S_2^{I} & g & g & g & . & . & . \\
. & b & g & b & . & . & . \\
. & . & . & & & & \\
. & . & . & & & & \\
S_m^{I} & . & . & & & &
\end{array}$$

נגביל את הסוכנים (נניח אותן הגבלות). הגבלות אלה ימנעו חלק מהתוצאות הרעות, אולם יכולות למנוע מהסוכנים השגת חלק מהיעדים הטובים.

אנו מחפשים הגבלות, כך שלכל i, ו g קיים s כך שהתועלת של i, בהינתן σ_{-i} יהיה גדול או שווה ל ε.

$$\forall i \forall g \in G \exists s \in \overline{S} : u_i^g\left(s_i, \sigma_{-i}\right) \geq \varepsilon \qquad \forall \sigma_{-i} \in \overline{S}^{n-1}$$

כלומר, ביצוע הפעולה s, בהינתן התנהגות הסוכנים האחרים, תתן לי תועלת טובה (גבוהה מ ε).

המקרה של $n=2$ מדגים זאת בצורה טובה. בהינתן יעד g_1 לסוכן I, ישנם דרכים רבות להשגת יעד זה, אך הן דורשות הגבלתו של סוכן II. אולם, הגבלת סוכן 2 תגרור, באופן סימטרי, גם את הגבלת סוכן 1, כך שסוכן 1 לא יוכל להגיע ליעד g_1 . לכן עדיף לשאוף להגבלות אשר לא יגבילו יתר על המידה. לבעיה הזאת יש קשר הדוק לבעית Satisiability (Sat).

סיכום: הגישה של מערכות חברתיות פותרת בצורה מאד נקייה, אולם מובילה לבעיות חישוביות אחרות.

נספח:

NP : הבעיה ניתנת לפתרון על ידי ניחושים, כלומר פתרון לא דטרמניסטי.

NP קשה : בעיה שיש רדוקציה יעילה אליה מכל בעיה אחרת ב NP .

NP שלמה : גם ב NP , וגם NP קשה.

$PSPACE$: הבעיה הינה ב $PSPACE$ אם המרחב הנדרש לפתרון הבעיה הוא פולינומיאלי.

$NPSPACE$: הבעיה ניתנת לפתרון במרחב פולינומיאלי בעזרת ניחושים. כלומר אני אנחש פתרון ואבדוק אם הוא הפתרון הנכון בזמן פולינומיאלי.

קיים משפט $NPSPACE = PSPACE$