שם: עמית סידס. תעודת זהות: 206768780. קורס: בינה מלאכותית 20551

השתמשתי ב- Python 2.7.16.

שאלה 1

בדקתי את המקרה של inconsistent effects ע"י בדיקה האם קיים proposition ברשימת ה-Add של אחד מהם וברשימת ה-Delete של השני (ולהפך).

בדקתי את המקרה של interference ע"י בדיקה האם קיים proposition ברשימת ה-Preconditions של אחד מהם וברשימת ה-Delete של השני (ולהפך).

שאלה 2

בדקתי אם לשתי הפעולות קיימים propositions ב-precondition שלהם שנמצאים ברשימת ה-MutexProp.

שאלה 3

עבור שני ה-propositions חיפשתי זוג פעולות שהן לא mutex שיוצרות אותן. אם לא נמצא כזה, סימן שאם ניתן להשיג אותם, זה רק בעזרת פעולות שהן mutex.

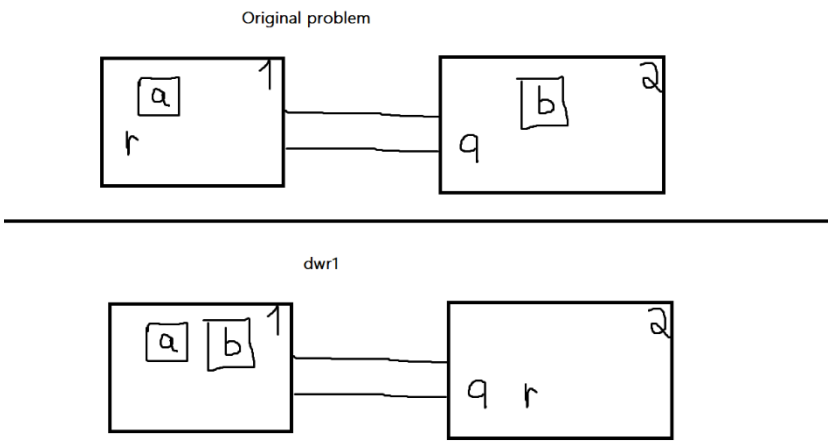
שאלה 4

עבור כל פעולה, בדקתי שכל proposition ב-precondition שלה הוא לא mutex עם proposition אחר ב-precondition, ורק אם זה נכון, סימן שצריך להוסיף את הפעולה.

שאלה 5-8

די סטרייט-פורוורד...

שאלה 9



לקח לי טיפה זמן להבין מה ה-propositions מייצגים, אבל ברגע שציירתי את זה, זה נהיה קל.

בעזרת השרטוט קל לראות שבבעיה המקורית הפתרון הוא להעמיס את a על r ואת b על q (2 פעולות), לנסוע לצד השני (2 פעולות), ולפרוק (2 פעולות).

לכן, יצרתי מצב דומה שבו שני הרובוטים מתחילים ב-2, וצריכים להעביר לשם את a ו-b שנמצאים ב-1. זה גם כן לוקח 6 פעולות, ולכן הוספתי דרישה ששני הרובוטים יסיימו ב-1, ולכן סך כל הפעולות הוא 8.

בשביל dwr2, פשוט יצרתי מצב התחלתי די מנוון שאין בו מספיק propositions בשביל לבצע אף פעולה, ולכן לא נוכל להגיע למצב המטרה.

שאלה 10

רוב המימוש היה פשוט. החלטתי לממש מנגנון שזוכר באילו מצבים כבר ביקרנו, ולכן נחסוך את החישוב עבורם שוב. בשביל שאייצג מצבים בצורה ייחודית, לפני שהוספתי את המצבים (כאשר מצב הוא רשימה של propositions) קודם כל מיינתי את ה-proposition של המצב. גיליתי שהמיון עצמו באופן מקרי לחלוטין חוסך הרבה מצבים ומשום מה מוצא את הפתרון יותר מהר.

שאלה 11

מימשתי את expandWithoutMutex ובניתי graph (רשימה) של השכבות כדי שאוכל להשתמש ב-isFixed. בכל פעם שעשיתי expand הגדלתי את ה-level וכאשר הגעתי למצב מטרה, ה-level מייצג את ערך היוריסטיקה.

שאלה 12

עשיתי משהו דומה לשאלה הקודמת רק שהפעם יצרתי dict שממפה בין שם של proposition לפעם הראשונה שנתקלנו בו. ברגע שמגיעים ל-goal (כפי שביקשו בשאלה), אז בהכרח כל ה-propositions יופיעו שם וכל שנותר הוא לסכום אותם ולהחזיר זאת.

שאלה 13

השאלה הכי מעניינת לדעתי. הבנתי שצריך לייצג את הנתונים בצורה טובה כך שהחישוב יתבצע מהר ללא הרבה פעולות מיותרות. אחרי הרבה זמן מחשבה הבנתי שבשביל להזיז דיסקית מעמוד a לעמוד b (לדוגמא) **חייב** שכל הדיסקיות הקטנות ממנו יהיה בעמוד c (כי אחרת דיסקית קטנה ממנו נמצאת מעליה ולכן אי אפשר להזיז את הדיסקית או שאנו ננסה להניח את הדיסקית על דיסקית קטנה ממנה). לכן, כל מה שצריך לשמור הוא על איזה עמוד נמצאת כל דיסקית, ועושים זאת עם 3 propositions עבור כל דיסקית.

בפעולות עצמן, ה-precondition יהיה שהדיסקית נמצאת בעמוד שממנו מנסים להזיז ושכל הדיסקיות הקטנות ממנו יהיה בעמוד השלישי שלא מעבירים ממנו או אליו.

שאלה 14

סטרייט-פורוורד, בהתאם למימוש של 13.