

Artificial Intelligence

Programming Assignment 4

Propositional logic: SAT Solver

due : 11/4 월요일 자정 (24:00)

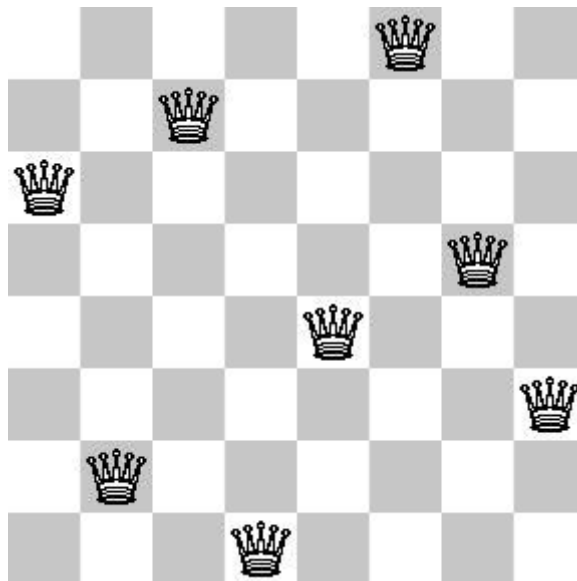
1. 개요

수업시간에 배운 propositional logic문제를 다루는 SAT solver z3를 활용하여 'N-Queens'문제를 해결한다.

2. 과제 설명

a. 'N-Queens'문제란?

- i. 'N x N'의 체스판이 있다.
- ii. N개의 Queen이 서로 공격하지 않아야한다.
(Queen은 가로, 세로, 대각선에 있는 말들을 모두 공격할 수 있다.)
- iii. N=8일 때의 답 예시



b. 'N-Queens' 문제 해결

SAT solver z3를 활용하여 N-Queens를 효율적으로 풀어본다. 본인 컴퓨터에서 **N=20** 기준으로, **nQueens native 보다 x배 빠른 만큼 구현 점수에 x점을 부여한다**. 예를 들어, **30배 빨라지면 구현 점수 30점을** 받을 수 있고, **100배 이상 빨라지면 구현 점수를 모두 받을 수 있다** (즉, 구현 점수 100점).

c. 입력 형식

입력은 다음과 같이 받을 수 있도록 한다:

```
C:\Users\progi\Desktop\2019_인공지능\z3-master\z3-master\build\python>py nQueensOpt.py
N:
15
```

py nQueensOpt.py를 실행시키면 N을 받을 수 있게 한다. 위의 예시에서는 사용자가 15를 입력으로 준 경우이다.

d. 출력 형식

```
N:
15
[11, 3, 5, 7, 9, 12, 14, 1, 4, 6, 8, 10, 2, 13, 15]
elapsed time: 0.14763665199279785 sec
```

위 예제와 같이, 1) 1차원 배열로 queen의 위치를 표기하고, 2) queen의 위치는 **1부터 N으로 표시**하고, 3) 마지막에 elapsed time을 적어준다.

e. 보고서 필수 기재 내용

- i. Symbol, formula 정의.
- ii. 실행 화면 출력
- iii. 결과에 대한 분석

ex> Optimization 방법, N에 따라서 성능이 어느정도 증가하는지 설명.

3. 사용 언어

- a. Python 3 (그 외 다른 언어는 안됨).

4. 코드 및 보고서 제출 방법

- a. “assignment4” 디렉토리 생성 후 해당 디렉토리 안에 소스코드 위치시킨 후 commit & push 할 것
- b. 보고서는 GitLab Wiki에 “assignment4” 페이지 생성 후 작성
- c. GitLab clone 또는 push시에 인증 오류가 난다면, GitLab web page 에서 파일 첨부를 통해 제출 할 것.
- d. GitLab을 통한 자세한 제출 방법은 포털 내 인공지능 강의실 홈페이지 공지 참조

5. 주의 사항

- a. 코드가 완성되면 반드시 GitLab에 push 해야 함
- b. 실행 파일과 소스코드를 같이 제출 할 것!**
- c. 코드에 주석 작성 필수 (주석 없으면 0점 처리)
- d. 컴파일 오류시 0점 처리
- e. 제출 기한 초과시

- i. 기본 20% 감점 후 매 24시간마다 20%씩 추가 감점
(ex. 50시간 초과시 60% 감점)
- ii. 72시간 이후 : 0점 처리