Homework 6: Course Scheduling

Brandon McKinzie

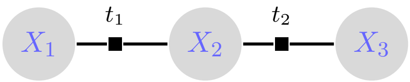
Problem 0: CSP Basics

(a) The CSP for this problem can be defined as follows:

- Variables: one for each of the m buttons. The domain of each variable has two possible values: *pressed* and *unpressed*.
- Constraints: one for each of the n lights. The scope of the ith constraint, f_i, is all buttons j for which $i \in T_j$.

Since each light bulb is initially off, the value of f_i, given some assignment of the buttons in its scope, will equal 1 if the number of pressed buttons (in its scope) is odd (constraint satisfied). Otherwise, it will equal zero (constraint not satisfied).

(b) We are given the following CSP:



i. There are 2 consistent assignments: $(1, 0, 1)$ and $(0, 1, 0)$.

ii. `backtrack()` will be called 9 times. The first call is backtrack($\varnothing$, 1, $\{X_1 : [0, 1], X_2 : [0, 1], X_3 : [0, 1]\}$), which results in the following 8 subsequent calls:

- backtrack($\{X_1 : 0\}$, 1, $\{X_2 : [0, 1], X_3 : [0, 1]\}$)
    - backtrack($\{X_1 : 0, X_3 : 0\}$, 1, $\{X_2 : [0, 1]\}$)
        - backtrack($\{X_1 : 0, X_2 : 1, X_3 : 0\}$, 1, $\{\}$)
    - backtrack($\{X_1 : 0, X_3 : 1\}$, 1, $\{X_2 : [0, 1]\}$)
- backtrack($\{X_1 : 1\}$, 1, $\{X_2 : [0, 1], X_3 : [0, 1]\}$)
    - backtrack($\{X_1 : 1, X_3 : 0\}$, 1, $\{X_2 : [0, 1]\}$)
    - backtrack($\{X_1 : 1, X_3 : 1\}$, 1, $\{X_2 : [0, 1]\}$)
        - backtrack($\{X_1 : 1, X_2 : 0, X_3 : 1\}$, 1, $\{\}$)

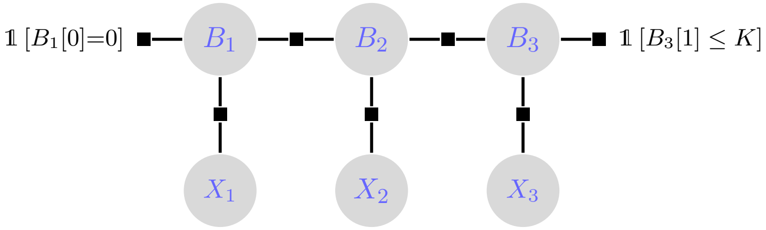
iii. `backtrack()` (with AC-3) will be called 7 times. Again, the first call is backtrack($\varnothing$, 1, $\{X_1 : [0, 1], X_2 : [0, 1], X_3 : [0, 1]\}$), which results in the following 6 subsequent calls:

- backtrack($\{X_1 : 0\}$, 1, $\{X_2 : [1], X_3 : [0]\}$)
    - backtrack($\{X_1 : 0, X_3 : 0\}$, 1, $\{X_2 : [1]\}$)
        - backtrack($\{X_1 : 0, X_2 : 1, X_3 : 0\}$, 1, $\{\}$)
- backtrack($\{X_1 : 1\}$, 1, $\{X_2 : [0], X_3 : [1]\}$)
    - backtrack($\{X_1 : 1, X_3 : 1\}$, 1, $\{X_2 : [0]\}$)
        - backtrack($\{X_1 : 1, X_2 : 0, X_3 : 1\}$, 1, $\{\}$)

Problem 2: Handling n-ary factors

(a) *Suppose we have a CSP with three variables X_1, X_2, X_3 with the same domain $\{0, 1, 2\}$ and a ternary constraint $[X_1 + X_2 + X_3 \leq K]$. How can we reduce this CSP to one with only unary and/or binary constraints? Explain what auxiliary variables we need to introduce, what their domains are, what unary/binary factors you'll add, and why your scheme works. Add a graph if you think that'll better explain your scheme.*

Similar to the event scheduling done in our section lecture, we introduce auxiliary variables B_i whose values take on two-tuples. The graph structure takes the form:



which shows the initial and final unary factors for B_1 and B_3 respectively. Additionally, we have the transition factors from B_{i-1} to B_i and state factors from X_i to B_i, respectively, which are defined by:

$$B_i[0] = B_{i-1}[1] \tag{1}$$

$$B_i[1] = \min(B_i[0] + X_i, K{+}1) \tag{2}$$

We see that the domain of each auxiliary variables B_i is $\{(a, b) : a \in [0..K+1], b \in [0..K+1]\}$.

Problem 3: Course Scheduling

(c) My profile.txt is shown below. I'm currently doing the Artificial Intelligence Graduate Certificate (SCPD) and this profile aligns with my goals for the certificate:

```
# Unit limit per quarter.
minUnits 0
maxUnits 7

# These are the quarters that I need to fill. It is assumed that
# the quarters are sorted in chronological order.
register Aut2019
register Win2019
register Spr2020

# Courses I've already taken
taken CS103
taken CS106B
taken CS107
taken CS109
taken CS140
taken CS161
taken CS221
taken MATH51
taken CS145
taken CS124
taken PHIL150
taken CS157
taken LINGUIST180
taken LINGUIST130A

# Courses that I'm requesting
request CS224N weight 5
request CS224U after CS224N weight 5
request CS228
request CS229
```

The schedule this produced was: This is indeed a reasonable schedule. Note that I had to enter a weight greater than 1 (arbitrarily chose 5) to get CS224N and CS224U in the recommended

Quarter	Units	Course
Aut2019	4	CS224N
Aut2019	3	CS229
Win2019	4	CS228
Spr2020	4	CS224U

schedule. I'm assuming the reason for this is that, when all are given equal weight, my fate is essentially determined by the ordering of the CSP search. By assigning non-equal weights, I can actually get my preferred set of courses.