**CPS 170: Introduction to AI** **Instructor:** Ron Parr

**Homework 2** **Due:** Thursday February 18, 2010

These problems are reproduced or adapted from problems from the textbook. Credit goes to the authors for developing these problems.

## 1 Constraint satisfaction (10 points)

Consider the problem of constructing (not solving) crossword puzzles: fitting words into a rectangular grid. The grid, which is given as part of the problem, specifies which squares are blank and which are shaded. Assume that a list of words (i.e., a dictionary) is provided and that the task is to fill in the blank squares by using any subset of the list. Formulate this problem precisely in two ways:

(a) As a general search problem. Choose an appropriate search algorithm and specify a heuristic function. Is it better to fill in the blanks one letter at a time or one word at a time?

(b) As a constraint satisfaction problem. Should the variables be words or letters?

## 2 Constraint Satisfaction Implementation (30 points)

Implement a backtracking solver for the crossword CSP described in the previous section. Your code should work by reading in a text file that describes crossword grid and then using a standard dictionary to search for valid combinations of words. We suggest you use the dictionary that comes in most Linux distributions, found in `/usr/share/dict/american-english`. For your convenience, we have placed a copy of this file as `dictionary.txt` at the root level of the class web page.

We have also provided some helper functions in Java in `hw2code.zip` (again, at the root level of the class web page). You are free to ignore these and use another language if you prefer. The zip file also contains some sample grids. We guarantee that at least grid0.txt is satisfiable. (Can you think of an easy solution to this?)

**What to turn in:**

- Your code
- Examples of your code running on sample grids

Other issues:

- It is not required, but you may want to implement the heuristic of picking the most constrained variable first and see how much this speeds things up.

- If your code is too slow, you can prune down the dictionary file to something smaller. (A good debugging "sanity check" would be to some up with a solution to the $3 \times 3$ grid by hand, then create a separate dictionary file that contains only the words used in your solution. If your solver can't get this right, then you know you have a bug.)

## 3 Constraint satisfaction (10 points)

Show how a single ternary constraint such as $A+B = C$" can be turned into three binary constraints by using an auxiliary variable. You may assume finite domains. (*Hint:* Consider a new variable that takes on values that are pairs of other values, and consider constraints such as "$X$ is the first element of the pair $Y$.") Next, show how constraints with more than three variables can be treated similarly. Finally, show how unary constraints can be eliminated by altering the domains of variables. This completes the demonstration that any CSP can be transformed into a CSP with only binary constraints.

## 4 Logic (10 points)

Prove, or find a counterexample to, each of the following assertions:

(a) If $\alpha \models \gamma$ or $\beta \models \gamma$ (or both) then $(\alpha \wedge \beta) \models \gamma$.

(b) If $\alpha \models (\beta \wedge \gamma)$ then $\alpha \models \beta$ and $\alpha \models \gamma$.

(c) If $\alpha \models (\beta \vee \gamma)$ then $\alpha \models \beta$ or $\alpha \models \gamma$ (or both).

## 5 Logic (10 points)

Decide whether each of the following sentences is valid, unsatisfiable, or neither. Verify you decisions using truth tables or the equivalence of Figure 7.11 (same in both editions).

(a) $Smoke \Rightarrow Smoke$

(b) $Smoke \Rightarrow Fire$

(c) $(Smoke \Rightarrow Fire) \Rightarrow (\neg Smoke \Rightarrow \neg Fire)$

(d) $Smoke \vee Fire \vee \neg Fire$

(e) $((Smoke \wedge Heat) \Rightarrow Fire) \Leftrightarrow ((Smoke \Rightarrow Fire) \vee (Heat \Rightarrow Fire))$

(f) $(Smoke \Rightarrow Fire) \Rightarrow ((Smoke \wedge Heat) \Rightarrow Fire)$

(g) $Big \vee Dumb \vee (Big \Rightarrow Dumb)$

## 6 Logic (10 points)

This exercise uses the function $MapColor$ and predicates $In(x, y)$, $Borders(x, y)$, and $Country(x)$, whose arguments are geographical regions, along with constant symbols for various regions. In each of the following we give an English sentence and a number of candidate logical expressions. For each of the logical expressions, state whether it (1) correctly expresses the English sentence; (2) is syntactically invalid and therefore meaningless; or (3) is syntactically valid but does not express the meaning of the English sentence.

(a) Paris and Marseilles are both in France.
(i) $In(Paris \wedge Marseilles, France)$.
(ii) $In(Paris, France) \wedge In(Marseilles, France)$.
(iii) $In(Paris, France) \vee In(Marseilles, France)$.

(b) There is a country that borders both Iraq and Pakistan.
(i) $\exists c\ Country(c) \wedge Border(x, Iraq) \wedge Border(c, Pakistan)$.
(ii) $\exists c\ Country(c) \Rightarrow [Border(c, Iraq) \wedge Border(c, Pakistan)]$.
(iii) $[\exists c\ Country(c)] \Rightarrow [Border(c, Iraq) \wedge Border(c, Pakistan)]$.
(iv) $\exists c\ Border(Country(c), Iraq \wedge Pakistan)$.

(c) All countries that border Ecuador are in South America.
(i) $\forall c\ Country(c) \wedge Border(c, Ecuador) \Rightarrow In(c, SouthAmerica)$.
(ii) $\forall c\ Country(c) \Rightarrow [Border(c, Ecuador) \Rightarrow In(c, SouthAmerica)]$.
(iii) $\forall c\ [Country(c) \Rightarrow Border(c, Ecuador)] \Rightarrow In(c, SouthAmerica)$.
(iv) $\forall c\ Country(c) \wedge Border(c, Ecuador) \wedge In(c, SouthAmerica)$.

(d) No region in South America borders any region in Europe.
(i) $\neg[\exists c, d\ In(c, SouthAmerica) \wedge In(d, Europe) \wedge Borders(c, d)]$.
(ii) $\forall c, d\ [In(c, SouthAmerica) \wedge In(d, Europe)] \Rightarrow \neg Borders(c, d)]$.
(iii) $\neg\forall c\ In(c, SouthAmerica) \Rightarrow \exists d\ In(d, Europe) \wedge \neg Borders(c, d)$.
(iv) $\forall c\ In(c, SouthAmerica) \Rightarrow \forall d\ In(d, Europe) \Rightarrow \neg Borders(c, d)$.

(e) No two adjacent countries have the same map color.
(i) $\forall x, y\ \neg Country(x) \vee \neg Country(y) \vee \neg Borders(x, y) \vee \neg(MapColor(x) = MapColor(y))$.
(ii) $\forall x, y\ \ (Country(x) \wedge Country(y) \wedge Borders(x, y) \wedge \neg(x = y)) \Rightarrow \neg(MapColor(x) = MapColor(y))$.
(iii) $\forall x, y\ Country(x) \wedge Country(y) \wedge Borders(x, y) \wedge \neg(MapColor(x) = MapColor(y))$.
(iv) $\forall x, y\ (Country(x) \wedge Country(y) \wedge Borders(x, y)) \Rightarrow MapColor(x \neq y)$.

# 7 Logic (10 points)

Consider a first-order logical knowledge base that describes worlds containing people, songs, albums (e.g., "Meet the Beatles") and disks (i.e., particular physical instances of CDs). The vocabulary contains the following symbols:

$CopyOf(d, a)$: Predicate. Disk $d$ is a copy of album $a$.

$Owns(p, d)$: Predicate. Person $p$ owns disk $d$.

$Sings(p, s, a)$: Album $a$ includes a recording of song $s$ sung by person $p$.

$Wrote(p, s)$: Person $p$ wrote song $s$.

$McMartney, Gershwin, BHoliday, Joe, EleanorRigby, TheManILove, Revolver$: Constants with the obvious meanings

Express the following statements in first-order logic:

(a) Gershwin wrote "The Man I Love."

(b) Gershwin did not write "Eleanor Rigby."

(c) Either Gershwin or McCartney wrote "The Man I Love."

(d) Joe has written at least one song.

(e) Joe owns a copy of *Revolver*.

(f) Every song that McCartney sings on *Revolver* was written by McCartney.

(g) Gershwin did not write any of the songs on *Revolver*.

(h) Every song that Gershwin wrote has been recorded on some album. (Possibly different songs are recorded on different albums.)

(i) There is a single album that contains every song that Joe has written.

(j) Joe owns a copy of an album that has Billie Holiday singing "The Man I Love."

(k) Joe owns a copy of every album that has a song sung by McCartney. (Of course, each different album is instantiated in a different physical CD.)

(l) Joe owns a copy of every album on which all the songs are sung by Billie Holiday.