Constraint Satisfaction

1) Consider the two formulations of the N-Queens problem as a constraint satisfaction problem from the slide set. Compare these two formulations in terms of the size and branching factor of the state space and the depth of the search tree.

2) In the crossword puzzle, we have a grid with blocked and unblocked cells and a dictionary of words. We want to assign a letter to each unblocked cell so that each vertical or horizontal contiguous segment of unblocked cells form a word that appears in the dictionary. An example of a solved crossword puzzle is given below[1].

A	D	I	M		R	I	P	S		F	A	T
C	U	T	E		E	T	A	T		A	L	E
D	E	S	D	E	M	O	N	A		I	A	N
C	L	A	U	D	I	O		T	U	R	N	S
			S	E	T		T	E	R	M		
A	S	W	A	N		P	E	N	N	A	M	E
D	I	I			H	A	L			I	M	S
O	R	L	A	N	D	O		E	D	D	I	E
		D	I	A	S		A	S	U			
S	T	O	M	P		P	R	A	N	C	E	D
E	R	A		P	E	T	R	U	C	H	I	O
L	I	T		E	L	S	A		A	I	N	T
L	O	S		R	I	D	S		N	A	S	H

Formulate this puzzle as a constraint satisfaction problem. Describe the variables, their domains and the constraints. (Bonus question: Try to come up with a second formulation of this puzzle as a constraint satisfaction problem.)

3) Consider the following constraint satisfaction problem with variables x, y and z, each with domain $\{1, 2, 3\}$, and constraints C_1 and C_2, defined as follows:

- C_1 is defined between x and y and allows the pairs $(1, 1), (2, 2), (3, 1), (3, 2), (3, 3)$. (A pair (a, b) means that assigning $x = a$ and $y = b$ does not violate the constraint. Any assignment that does not appear in the list violate the constraint.)
- C_2 is defined between y and z and allows the pairs $(1, 1), (1, 2), (3, 1), (3, 2), (3, 3)$.

[1] http://www.americanshakespearecenter.com/v.php?pg=684

Which values does arc consistency rule out from the domain of each variable? Suppose that we started search after establishing arc consistency, and we assign $x = 1$. Which values does a) forward checking and b) maintaining arc consistency rule out from the domain of each variable?

4) Suppose you have a search problem defined by more or less the usual stuff:

- a set of states S;
- an initial state s_{start};
- a set of actions A including the $NoOp$ action, that has no effect;
- a transition model $Result(s, a)$ (that determines the successor state when action a is executed in state s);
- a set of goal states G.

Unfortunately, you have no search algorithms! All you have is a CSP solver.

(a) Given some time horizon T, explain how to formulate a CSP such that (1) the CSP has a solution exactly when the problem has a solution of length T steps; (2) the solution to the original problem can be "read off" from the variables assigned in CSP solution. Your formulation must give the variables, their domains, and all applicable constraints expressed as precisely as possible. You should have at least one variable per time step, and the constraints should constrain the initial state, the final state, and consecutive states along the way.

(b) Explain how to modify your CSP formulation so that the CSP has a solution when the problem has a solution of length $\leq T$ steps, rather than exactly T steps.

5) 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.