CS 5100 – PS3

Yijing Xiao

Problem 1: Give precise formulations for each of the following as constraint satisfaction problems

a) **Rectilinear floor-planning: find non-overlapping places in a large rectangle for a number of smaller rectangles**

Variables: X = each of smaller rectangles

Domain of variables: D = the size and position of each smaller rectangle

Constraints: C = any smaller rectangle cannot overlap with others

b) **Hamiltonian tour: given a network of cities connected by roads, choose an order to visit all cities in a country without repeating any**

Variables: X = each of city in a country

Domain of variables: D = roads for this city connected to others

Constraints: C = each city can only visit once

Problem 2: Explain why it is a good heuristic to choose the variable that is most constrained but the value that is least constraining in a CSP search

The reason we choose the most-constrained variable is it always fails fast because if there no solution we want to it know this fact as fast as possible.

Least-constraining value heuristic choose value that rules out fewest assignments to remaining variables, and maximum flexibility for future assignments.

Problem 3 Extra Credit: how your work by listing which assignments you make to each variable, when you find a conflict (and what that leads you to try assigning), and where you jump back to when you do a backjump (use the conflict sets to explain this).

	A1	H	A4	F1	A2	F2	A3	T
Initial	R,G,B	R,G,B	R,G,B	R,G,B	R,G,B	R,G,B	R,G,B	R,G,B

Choose variables in order shown above, values in RGB order

1. Assign A1 = R

Add arcs H-> A1, A2->A1 to queue to check for consistency

- H->A1: H = R is inconsistent with A1 = R, hence remove R from H, add A4-> H, A2->H,

A3->H, and T->H to queue

- A2->A1: A2 = R is inconsistent with A1 = R, hence remove R from A2, and add H->A2, and A3->A2 to queue
- A4->H: Each value in A4 is consistent with some assignment in H (H now can be G/B), so no values need to be removed. Similar for A2->H, A3->H, T->H, H->A2, and A3->A2

	A1	H	A4	F1	A2	F2	A3	T
1	R	G,B	R,G,B	R,G,B	G,B	R,G,B	R,G,B	R,G,B

2. Assign H = G

Add arcs A4-> H, A2->H, A3->H, and T->H to queue to check for consistency

- A4->H: A4 = G is inconsistent with H = G, hence remove G from A4, add A3->A4 to queue
- A2->H: A2 = G is inconsistent with H = G, hence remove G from A2, add A3->A2 to queue
- A3->H: A3 = G is inconsistent with H = G, hence remove G from A3, add A2->A3, and A4->A3 to queue
- T->H: T = G is inconsistent with H = G, hence remove G from T, add F1->T, and F2->T to queue
- H->A2: Each value in H is consistent with some assignment in A2, so no values need to be removed.
- A3->A2: A3 = B is inconsistent with A2 = B, hence remove B from A3, add A4->A3 to queue
- A3->A4: A4 = R is inconsistent with A3 = R, hence remove R from A4
- F1->T: Each value in F1 is consistent with some assignment in T (T now can be R/B), so no values need to be removed. Similar for F2->T

	A1	H	A4	F1	A2	F2	A3	T
2	R	G	B	R,G,B	B	R,G,B	R	R,B

3. Assign T=R

Add F1->T, and F2->T to queue to check for consistency

- F1->T: F1 = R is inconsistent with T = R, hence remove R from F1
- F2->T: F2 = R is inconsistent with T = R, hence remove R from F2

	A1	H	A4	F1	A2	F2	A3	T
3	R	G	B	G,B	B	G,B	R	R

4. Assign F1 = G

5. Assign F2 = G

	A1	H	A4	F1	A2	F2	A3	T
final	R	G	B	G	B	G	R	R