Artificial Intelligence Assignment Week 4 Reference Key

09/24/2018

3. Consider the crossword puzzles shown in Figure 4.15.

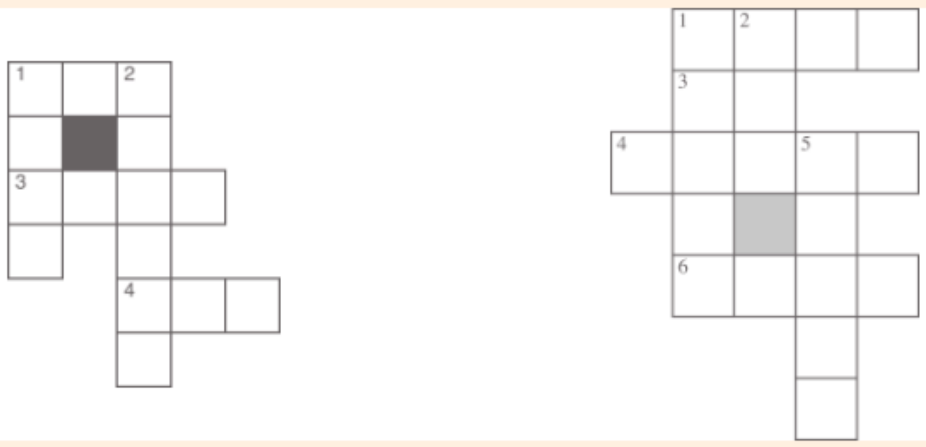


Figure 4.15: Two crossword puzzles

The possible words for (a) are:

ant, big, bus, car, has, book, buys, hold, lane, year, ginger, search, symbol, syntax.

The available words for (b) are

at, eta, be, hat, he, her, it, him, on, one, desk, dance, usage, easy, dove, first, else, loses, fuels, help, haste, given, kind, sense, soon, sound, this, think.

(a) Draw the constraint graph nodes for the positions (*1-across*, *2-down*, etc.) and words for the domains, after it has been made domain consistent

Constraints:

1. length of node;

2. intersection of nodes;

Node domains:

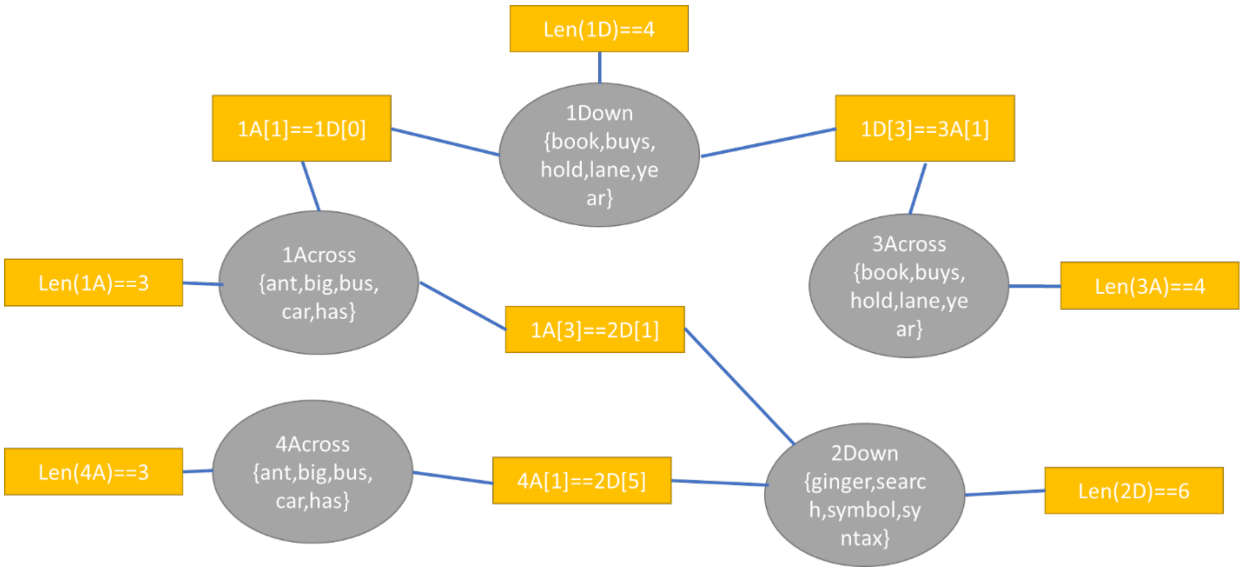
· 1Across: { ant, big, bus, car, has }

· 3Across: { book, buys, hold, lane, year }

· 4Across: { ant, big, bus, car, has }

· 1Down: { book, buys, hold, lane, year }

· 2Down: { ginger, search, symbol, syntax }


Len(1D)==4
1A[1]==1D[0]
1Down {book,buys, hold,lane,year}
1D[3]==3A[1]
Len(1A)==3
1Across {ant,big,bus, car,has}
1A[3]==2D[1]
3Across {book,buys, hold,lane,year}
Len(3A)==4
Len(4A)==3
4Across {ant,big,bus, car,has}
4A[1]==2D[5]
2Down {ginger,search,symbol,syntax}
Len(2D)==6

(b) Give an example of pruning due to arc consistency.

Pruned:

For 1Across and 2Down to be arc consistent, we must prune all words from both domains where the first letter of 2Down **cannot be** the first letter of 1Across.

1Across becomes: { big, bus, has }

2Down becomes: { ginger, search, symbol, syntax }

(c) What are the domains after arc-consistency has halted?

After arc consistency has halted:

· 1Across: { bus, has }

· 3Across: { lane, year }

· 4Across: { ant, car}

· 1Down: { buys, hold }

· 2Down: { search, syntax }

(d) Consider the dual representation, in which the squares on the intersection of words are the variables. The domains of the variable contain the letters that could go in these positions. Give the domains after this network has been made arc consistent. Does the result after arc consistency in this representation correspond to the result in part (c)?

There are five intersection nodes need to be considered:

1across^1down: {b, h}

1down^3across: {l, y}

1across^2down: {s}

2down^3across: {a, n}

2down^4across: {a, c}

Yes, the final results correspond. For example, examine the domains of 1across and 1down

from part (c) and the domain of (1across^1down) from (d). (1across^1down) intersects at the first letter

of 1across and 1down. All words in 1across and 1down must start with the letters in 1across^1down.

(e) Show how variable elimination solves the crossword problem. Start from the arc-consistent network from part (c).

1.Eliminate 1down.

1down	1cross	1down	3cross
Buys	Bus	Buys	year
Hold	Has	Hold	lane

Get:

1across	3across
Bus	Year
Has	Lane

2. Eliminate 2down.

2down	1across	2down	3across	2down	4across
Search	Buys	Search	Year	Search	Car
Syntax	Has	Syntax	Lane	Syntax	ant

Get:

1across	3across	4across
Bus	Year	Car
Has	Lane	Ant

3.Eliminate 1across.

1across	3across	1across	4across
Bus	Year	Bus	Car
Has	Lane	Has	Ant

Get.

3across	4across
Year	Car
Lane	Ant

4.Eliminate 3across

3across	4across

Year	Car
Lane	Ant

Get.

4across
Car
Ant

Now we can see '4across' has only two valid words in its domain: "car" , "ant."

Now backtrack and get solutions(1a,1d,2d,3a,4a):

1across	1down	2down	3across	4across
Has	Hold	Syntax	Lane	Ant
Bus	Buys	Search	Year	Car

(f) Does a different elimination ordering affect the efficiency? Explain.

Yes.

Elimination ordering affects intermediate table size. For example, a join that considers one table might reduce the size of it to 1 while not joining it will increase the size. If this one table is joined last, all the prior joins will have been wasted whereas if the one table was joined first, all the results from future joins will remain at size 1.

4. Consider the complexity for generalized arc consistency beyond the binary case considered in the text. Suppose there are nn variables, cc constraints, where each constraint involves k variables, and the domain of each variable is of size d. How many arcs are there? What is the worst-case cost of checking one arc as a function of c, k and d? How many times must an arc be checked? Based on this, what is the time complexity of GAC as a function of cc, kk and dd? What is the space complexity?

There are O(ck) arcs since there must be an arc between a constraint and the variables it involves.

The worst-case of checking one arc is O(d^k). Say you are checking a constraint with k variables with domain d, and in the worst case you check every possibility.

An arc is checked again if a value is pruned from the domain of any variable which has an arc to it. As a result, an arc has to be checked O(kd) times.

The time complexity is O(c*d^k*k^2).
The total space complexity is O(nd + ck). The set of arcs is O(ck) and the valid numbers in domains to keep track of is O(nd).