03rd Oct 2023

CSCE 5210: FUNDAMENTALS OF AI

HW-3: Reasoning with constraints

NAME: Avinash Parasa

UNT ID: 11720006

Question-1:

→ Given figure:

1	2	3
4		
5		

→ Given words: add, age, aid, aim, air, are, arm, art, bad, bat, bee, boa, dim, ear, eel, eft, lee, oaf

→ Here, we need to find six words to fit in the box that satisfies the condition 1-across, 4-across, 5-across, 1-down, 2-down, 3 down

→ First let us solve by using trail & error method:

* If we take add as 1-across or 1-down.

* If 1-across, then:

1 a	2 d	3 d
4	-	-
5	-	-

, Now we need 2 words that start with d. but we have only one.

→ So, we can eliminate it.

→ Now, let us think and take a word that fits in 1-across and having a word that starts with other 2 letters.

→ If we take age, we don't have any woord with 'g'. So, we can't take word with 'g' in this case.

→ when it comes to aid, aim, air, we can skip them.

→ If we take bee as we have two words with 'e'.

→ 1-down should be word with the word starts with 'b'.

→ we can take bad, bat, bee, boa:

1 b	2 e	3 e
4		
5		

→ If we take bad, last, No, first let's take words with ear, eel, eft.

→ In these three we have 2 words to pick.

→ wee can't take ear as there are no words that starts with like with middle word as 'r'.

→ So, the two confirmed words are:

b	e	e
	a	f
	r	t

→ Now we can take word which starts with b and had 'af' or 'rt' as their ending words:

→ The word is 'boa':

b	e	e
o	a	f
a	r	t

→ we can also write it as:

b	o	a
e	a	r
e	f	t

⇒ So, finally we have finalized 6-three letter words, they are boa, ear and eft as 1-across, 4-across and 5-across and bee, oaf and art as 1-down, 2-down, 3-down or else we can write it as vice-versa.

(a) Give an example of pruning due to domain consistency using first representation.

→ First let us take what is domain consistancy:

→ in the simplest case, when a constraint has just one variable in its scope, the arc is domain consistancy if every value of the variable satisfies the constraint.

→ In the given problem, we have to find all the three letters words to fill in the boxes.

→ when we have any constraints and if we make sure all constraints are satisfied then it is in domain consistancy, else not.

→ So, in the above given words are all the three letter words and no constraints found if we have any we have to remove and make domain consistency.

→ Here as there are no constraints that need to be pruned so the above one is in domain domain consistancy.

(b)-ARC CONSISTENCY EXAMPLE:

→ The above problem is not in arc consistency.

→ First let us talk about what the arc consistency is exactly.

→ Suppose constraint c has scope $\{x, y_1, \ldots y_k\}$. Arc $\langle x, c \rangle$ is arc consistent if, for each value $x \in D_x$, there are values $y_1 \ldots y_k$ where $y_i \in D_{y_i}$, such that $c(x = x, y_1 = y_1 \ldots y_k = y_k)$ is satisfied. A network is arc consistent if all its arcs are consistent.

→ Here add can be pruned from the domain of words of 1-across as there is no two elements.

→ This is one example for arc consistency.

→ Let us take other one and try to prune to make arc consistency.

→ Let us take age, if we take age as 1-across. For 2-domain down, we don't have any word that starts with 'g' so we can't prune age.

→ other example is arm, let us take arm and observe, if we keep arm as 1-across, then for 3-down there is no word that starts with m or if we take arm for 5-across there is no word that ends with 'a' so we can prune the arm.

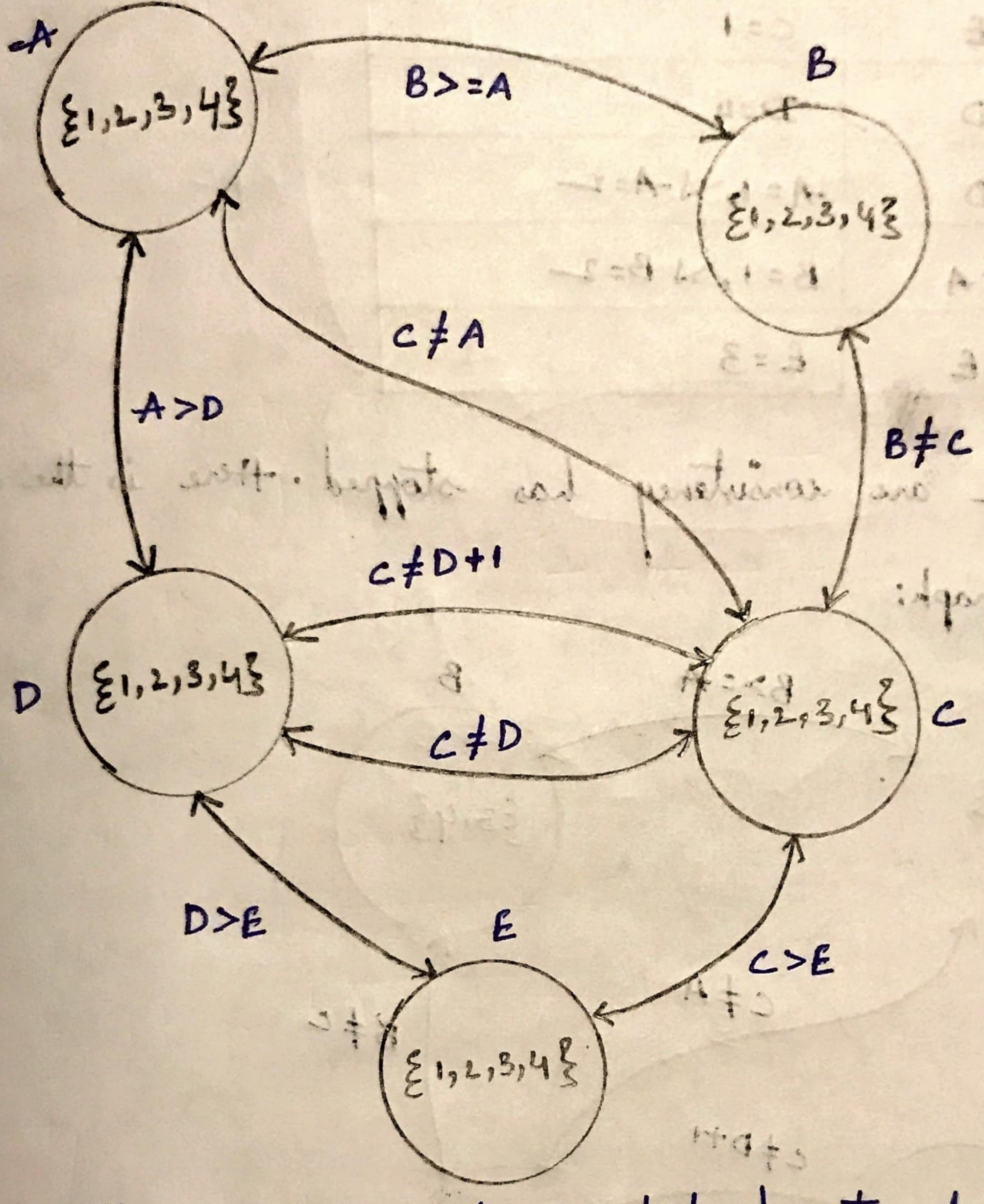
→ These are three examples I have taken for purning arc consistency using first representation.

(c) For the above problem, we can solve using domain consistency and arc consistency as there are few words and can easily be pruned to get the final answer.

→ But to get correct and efficient or best solution we need to involve some search strategies for this problem or any other complex algorithms to get perfect answers.

Question-2:

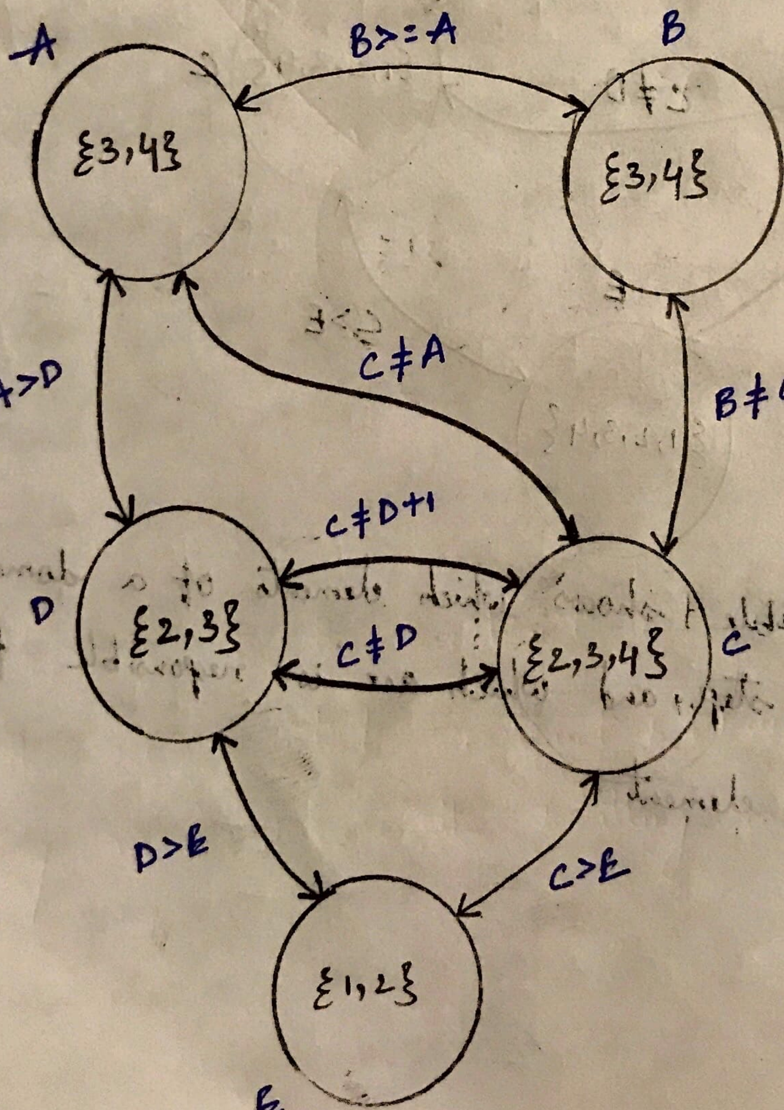
(b) USING ARC CONSISTENCY:



→ The following table shows which elements of a domain are deleted at each step, and which arc is responsible for removing the element:

ARC	RELATION	VALUE(S) REMOVED
<D, E>	D > E	D = 1
<E, D>	D > E	E = 4
<C, E>	C > E	C = 1
<D, A>	A > D	D = 4
<A, D>	A > D	A = 1, & A = 2
<B, A>	B >= A	B = 1, & B = 2
<E, D>	D > E	E = 3

→ At this stage arc consistency has stopped. Here is the arc constraint graph:



→ Note that, if you had one arc between C & D labelled with C != D+1 & C != D, then C=3 can be removed by considering the arc <C,D>. If you have two arcs, C=3 cannot be removed by arc consistency.

* Show how splitting a domain can be used to solve this problem.

→ Lets split the domain of D.

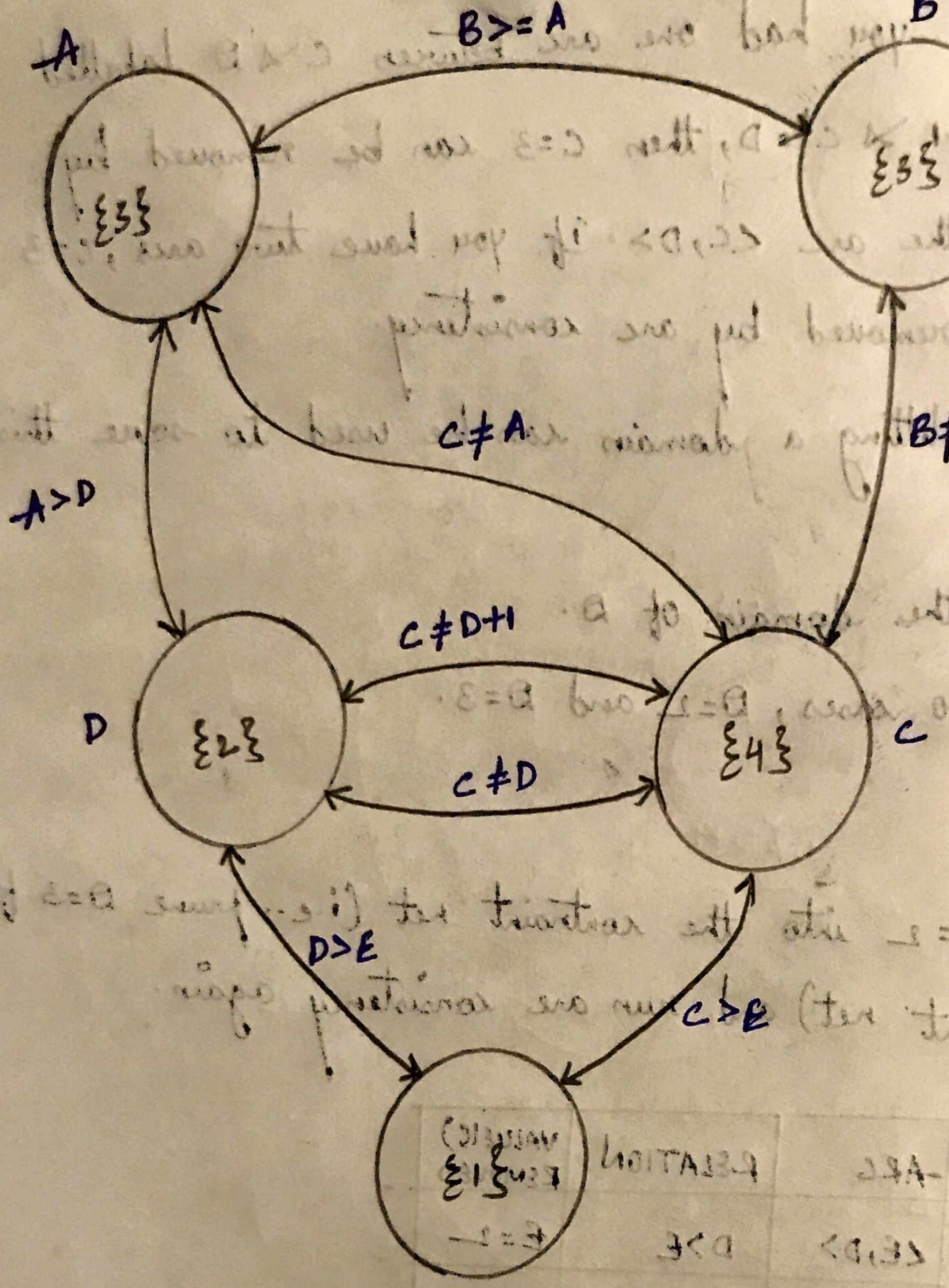
→ There are two cases, D=2 and D=3.

CASE 1:

→ Lets put D=2 into the contraint net (i.e., prune D=3 from the constraint net) and run arc consistency again.

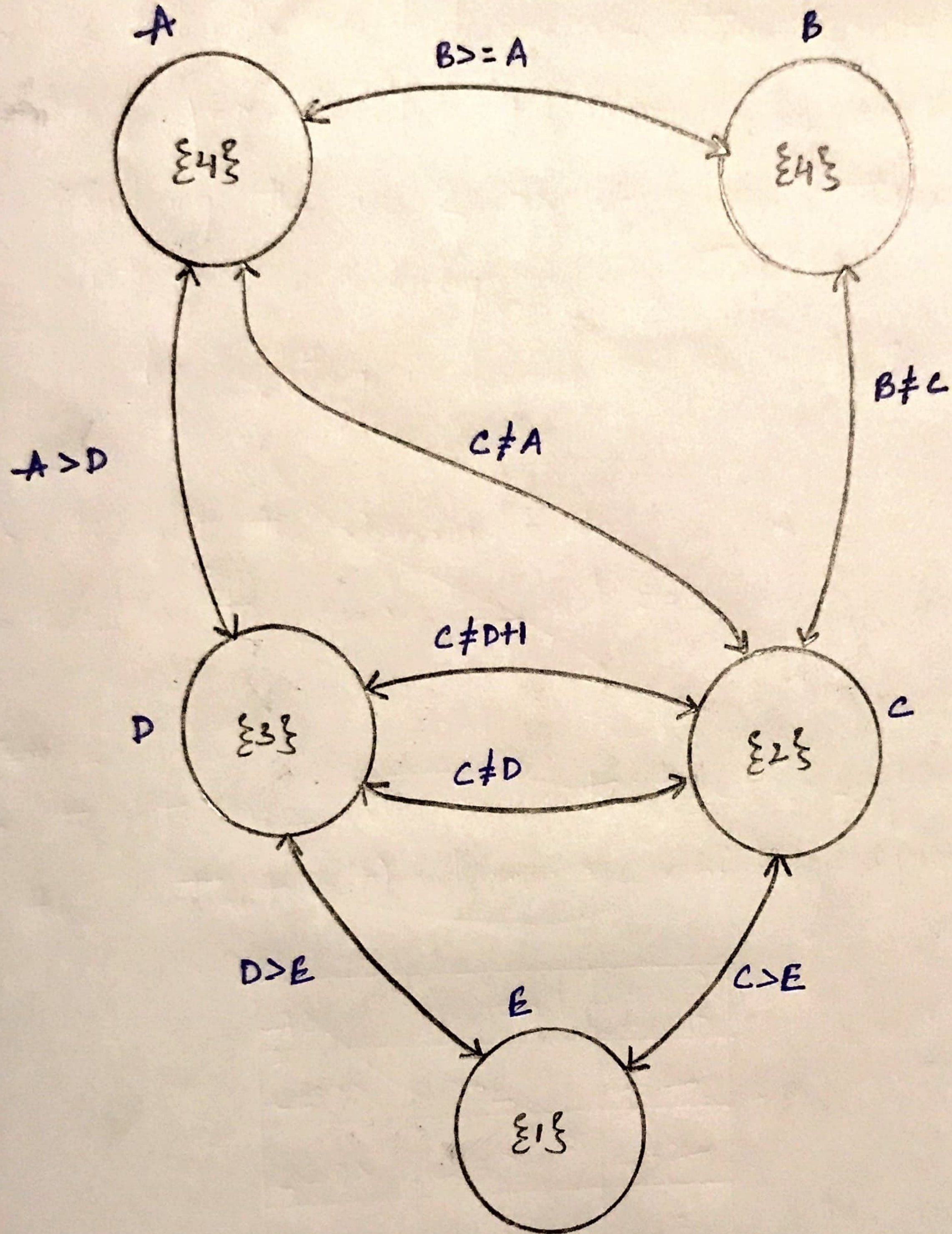
ARC	RELATION	VALUE(S) REMOVED
<E,D>	D>E	E=2
<C,D>	C != D	C=2
<C,D>	C != D+1	C=3
<A,C>	C != A	A=4
<B,C>	B != C	B=4

→ Here is the resulting constraint net.



CASE 2:

→ Lets put D=3 into the constraint net (i.e., prune D=2 from the arc-consistent constraint net) and run arc consistency again. Here is the resulting constraint net.



→ These two constraint nets correspond to the only two solutions.