Note that in this exercise we perform GAC as preprocessing in advance of any search, *as well as* during search. Constraint propagation (GAC, FC, etc.) is often performed in advance of initiating search as a preprocessing step to prune variable domains. Once no more propagation can be done, search (with GAC, FC, etc. as illustrated in our algorithms) is commenced. The example that follows illustrates the additional use of GAC as a preprocessing step in advance of commencing search with GAC.

GAC Example

(a) $Dom[X] = \{1, 2, 3, 4\}$
(b) $Dom[Y] = \{1, 2, 3, 4\}$
(c) $Dom[Z] = \{1, 2, 3, 4\}$
(d) $Dom[W] = \{1, 2, 3, 4, 5\}$

And 3 constraints:

(a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$
(b) $C_2(X, W)$ which is satisfied only when $W > X$
(c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

GAC Example

(a) $Dom[X] = \{1, 2, 3, 4\}$
(b) $Dom[Y] = \{1, 2, 3, 4\}$
(c) $Dom[Z] = \{1, 2, 3, 4\}$
(d) $Dom[W] = \{1, 2, 3, 4, 5\}$

And 3 constraints:

(a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$
(b) $C_2(X, W)$ which is satisfied only when $W > X$
(c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

- All constraints put on GAC queue.
- Process C_3 first.

  X = 1 (X=1, Y=1, Z=1, W=3)
  X = 2 (X=2, Y=1, Z=1, W=4)
  X = 3 (X=3, Y=1, Z=1, W=5)
  X = 4 – Inconsistent.
  Dom(X) = {1, 2, 3}
  similarly
  Dom(Y) = {1, 2, 3}
  Dom(Z) = {1, 2, 3}

W = 1 – inconsistent
W = 2 – inconsistent
W = 3 – same support as X=1
W = 4 – same support as X = 2
W= 5 – same support as X = 3

Dom(W) = {3, 4, 5}

All domains pruned, but all other constraints already on GAC queue

GAC Example

(a) $Dom[X] = \{1, 2, 3, 4\}$
(b) $Dom[Y] = \{1, 2, 3, 4\}$
(c) $Dom[Z] = \{1, 2, 3, 4\}$
(d) $Dom[W] = \{1, 2, 3, 4, 5\}$

And 3 constraints:

(a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$
(b) $C_2(X, W)$ which is satisfied only when $W > X$
(c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

- Process C_2 next
  Currently
  Dom(X) = {1, 2, 3}
  Dom(W) = {3, 4, 5}

  X = 1 (X=1, W=3)
  X = 2 (X=2, W=3)
  X = 3 (X=3, W=4)

  W=3, W=4 found supports already

  W=5 (X=1, W=5)

  No domains pruned. Nothing added to GAC Queue

GAC Example

(a) $Dom[X] = \{1, 2, 3, 4\}$
(b) $Dom[Y] = \{1, 2, 3, 4\}$
(c) $Dom[Z] = \{1, 2, 3, 4\}$
(d) $Dom[W] = \{1, 2, 3, 4, 5\}$

And 3 constraints:

(a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$
(b) $C_2(X, W)$ which is satisfied only when $W > X$
(c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

- Process C_1 next

At this stage
Dom(X) = Dom(Y) = Dom(Z)
= {1, 2, 3}

X = 1 – inconsistent
X = 2 – (X=2, Y=1, Z=1)
X = 3 – (X=3, Y=1, Z=2)

Y = 1 – same support as X=2
Y = 2 – (X=3, Y=2, Z=1)
Y = 3 – inconsistent

Z = 1 – same support as X=2
Z = 2 – same support as X=3
Z = 3 – inconsistent

Updated domains
X = {2,3}
Y = {1,2}
Z = {1,2}

Put C_2 and C_3 back onto GAC queue

GAC Example

(a) $Dom[X] = \{1, 2, 3, 4\}$
(b) $Dom[Y] = \{1, 2, 3, 4\}$
(c) $Dom[Z] = \{1, 2, 3, 4\}$
(d) $Dom[W] = \{1, 2, 3, 4, 5\}$

And 3 constraints:

(a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$
(b) $C_2(X, W)$ which is satisfied only when $W > X$
(c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

- Process C_3 next current domains:
  Dom(X) = {2, 3}
  Dom(Y) = {1, 2}
  Dom(Z) = {1, 2}
  Dom(W) = {3,4,5}

X = 2 – {X=2, W=4, Y=1, Z=1}
X = 3 – {X=3, W=5, Y=1, Z=1}

Y = 1 – found support
Y = 2 – {X=2, W=5, Y=2, Z=1}

Z = 1 – found support
Z = 2 – {X=2, W=5, Y=1, Z=2}

W = 3 inconsistent
W = 4 – found support
W = 5 – found support

Pruned domains
W = {4, 5}

C_2 already on GAC queue

GAC Example

(a) $Dom[X] = \{1, 2, 3, 4\}$
(b) $Dom[Y] = \{1, 2, 3, 4\}$
(c) $Dom[Z] = \{1, 2, 3, 4\}$
(d) $Dom[W] = \{1, 2, 3, 4, 5\}$

And 3 constraints:

(a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$
(b) $C_2(X, W)$ which is satisfied only when $W > X$
(c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

- Process C_2 next current domains:
  Dom(X) = {2, 3}
  Dom(W) = {4,5}

X = 2 – {X=2, W=4}
X = 3 – {X=3, W=4}

W = 4 – found support
W = 5 – {X=3, W=5}

No Domains pruned.
Nothing added to queue

Queue Empty

GAC finished.

GAC domains:

X = {2,3}

Z = {1, 2}

Y = {1, 2}

W = {4,5}

GAC Example

(a) $Dom[X] = \{1, 2, 3, 4\}$
(b) $Dom[Y] = \{1, 2, 3, 4\}$
(c) $Dom[Z] = \{1, 2, 3, 4\}$
(d) $Dom[W] = \{1, 2, 3, 4, 5\}$

And 3 constraints:

(a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$
(b) $C_2(X, W)$ which is satisfied only when $W > X$
(c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

- Note GAC enforce does not find a solution
  To find a solution we must use do search while enforcing GAC.

- Branch on X.
  X = 2
  GAC(C_1) ➔ Y = 1, Z=1
  GAC(C_2) ➔ no changes
  GAC(C_3) ➔ W = 4
  This is a solution.

- Branch on X = 3
  GAC(C_1) ➔ no changes
  GAC(C_2) ➔ no changes
  GAC(C_3) ➔ Prune W=4
  Prune Y = 2
  Prune Z = 2
  Current Domains
  X={3}, Y={1}, Z={1}, W={5}
  GAC(C_1) ➔ Prune Y={1} DWO

  NOTE No solution with X=3 but X=3 not pruned by GAC enforce.

Example

- C1(V1,V2,V3)

V1	V2	V3
~~A~~	~~B~~	~~C~~
~~B~~	~~A~~	~~C~~
~~A~~	~~A~~	~~B~~

✓

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
~~A~~	~~A~~	~~A~~	~~A~~
~~A~~	~~B~~	~~C~~	~~B~~
B	C	B	B
~~C~~	~~A~~	~~B~~	~~C~~
~~C~~	~~B~~	~~A~~	~~B~~

✓

- C3(V2,V3,V5)

V2	V3	V5
~~A~~	~~A~~	~~A~~
~~A~~	~~B~~	~~C~~
B	C	B
~~C~~	~~A~~	~~B~~
~~C~~	~~B~~	~~A~~

✓

V1: ~~a~~ ~~b~~ ~~c~~ DWO

V2: ~~a~~ b ~~c~~

V3: ~~a~~ ~~b~~ c

V4: ~~a~~ b c

V5: ~~a~~ b ~~c~~

~~c1~~

~~c2~~

~~c3~~

c1

→ ~~c2~~

c3 C1

- Dom[V1]…Dom[V5] = {a, b, c}

Example

- C1(V1,V2,V3)

V1	V2	V3
A	B	C
B	A	C
A	A	B

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
A	A	A	A
A	B	C	B
B	C	B	B
C	A	B	C
C	B	A	B

- C3(V2,V3,V5)

V2	V3	V5
A	A	A
A	B	C
B	C	B
C	A	B
C	B	A

- V1=C: no support
- V2=C: no support
- V3=A: no support

- V1={a,b}
- V2={a,b}
- V3={b,c}

Example

- C1(V1,V2,V3)

V1	V2	V3
A	B	C
B	A	C
A	A	B

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
A	A	A	A
A	B	C	B
B	C	B	B
C	A	B	C
C	B	A	B

- C3(V2,V3,V5)

V2	V3	V5
A	A	A
A	B	C
B	C	B
C	A	B
C	B	A

- V1=C: no support
- V2=C: no support
- V3=A: no support

- V1={a,b}
- V2={a,b}
- V3={b,c}

Example

- C1(V1,V2,V3)

V1	V2	V3
A	B	C
B	A	C
A	A	B

- V1=C: no support
- V2=C: no support
- V3=A: no support

- V1={a,b}
- V2={a,b}
- V3={b,c}

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
A	A	A	A
A	B	C	B
B	C	B	B
C	A	B	C
C	B	A	B

- V4=A: no support
- V5=A: no support
- V5=C: no support

- V4={C,B}
- V5={B}

- C3(V2,V3,V5)

V2	V3	V5
A	A	A
A	B	C
B	C	B
C	A	B
C	B	A

Example

- C1(V1,V2,V3)

V1	V2	V3
A	B	C
B	A	C
A	A	B

- V1=C: no support
- V2=C: no support
- V3=A: no support

- V1={a,b}
- V2={a,b}
- V3={b,c}

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
A	A	A	A
A	B	C	B
B	C	B	B
C	A	B	C
C	B	A	B

- V4=A: no support
- V5=A: no support
- V5=C: no support

- V4={C,B}
- V5={B}

- C2(V2,V3,V5)

V2	V3	V5
A	A	A
A	B	C
B	C	B
C	A	B
C	B	A

Example

- C1(V1,V2,V3)

V1	V2	V3
A	B	C
B	A	C
A	A	B

- V1=C: no support
- V2=C: no support
- V3=A: no support

- V1={a,b}
- V2={a,b}
- V3={b,c}

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
A	A	A	A
A	B	C	B
B	C	B	B
C	A	B	C
C	B	A	B

- V4=A: no support
- V5=A: no support
- V5=C: no support

- V4={C,B}
- V5={B}

- C2(V2,V3,V5)

V2	V3	V5
A	A	A
A	B	C
B	C	B
C	A	B
C	B	A

- V2=A: no support
- V3=B: no support

- V2={B}
- V3={C}

Example

- C1(V1,V2,V3)

V1	V2	V3
A	B	C
B	A	C
A	A	B

- V1=B has no support
- V1={A}

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
A	A	A	A
A	B	C	B
~~B~~	~~C~~	~~B~~	~~B~~
C	A	B	C
C	B	A	B

- V4={C,B}
- V5={B}

- C2(V2,V3,V5)

V2	V3	V5
A	A	A
A	B	C
B	C	B
C	A	B
C	B	A

- V2={B}
- V3={C}

V1 = a
V2 = b
V3 = c

V4 = c, ~~b~~
V5 = b

Example

- C1(V1,V2,V3)

V1	V2	V3
A	B	C
B	A	C
A	A	B

- V1=B has no support
- V1={A}

- C2(V1,V3,V4,V5)

V1	V3	V4	V5
A	A	A	A
A	B	C	B
B	C	B	B
C	A	B	C
C	B	A	B

- V4=B has no support
- V4={B}
- V5={B}
- V3=C has no support

- V3={} DWO

- C2(V2,V3,V5)

V2	V3	V5
A	A	A
A	B	C
B	C	B
C	A	B
C	B	A

- V2={B}
- V3={C}