6장. 제약 만족 문제
(CSP: CONSTRAINT SATISFACTION PROBLEMS)

순서

I. CSP 정의

II. 이진 CSP

III. 제약 조건 전파

I. CSP 개념

- 상태(State)를 블랙박스 이상의 어떤 것으로 취급하여 탐색하는 방식
- 문제의 구조를 더 깊이 이해할 수 있는 방식

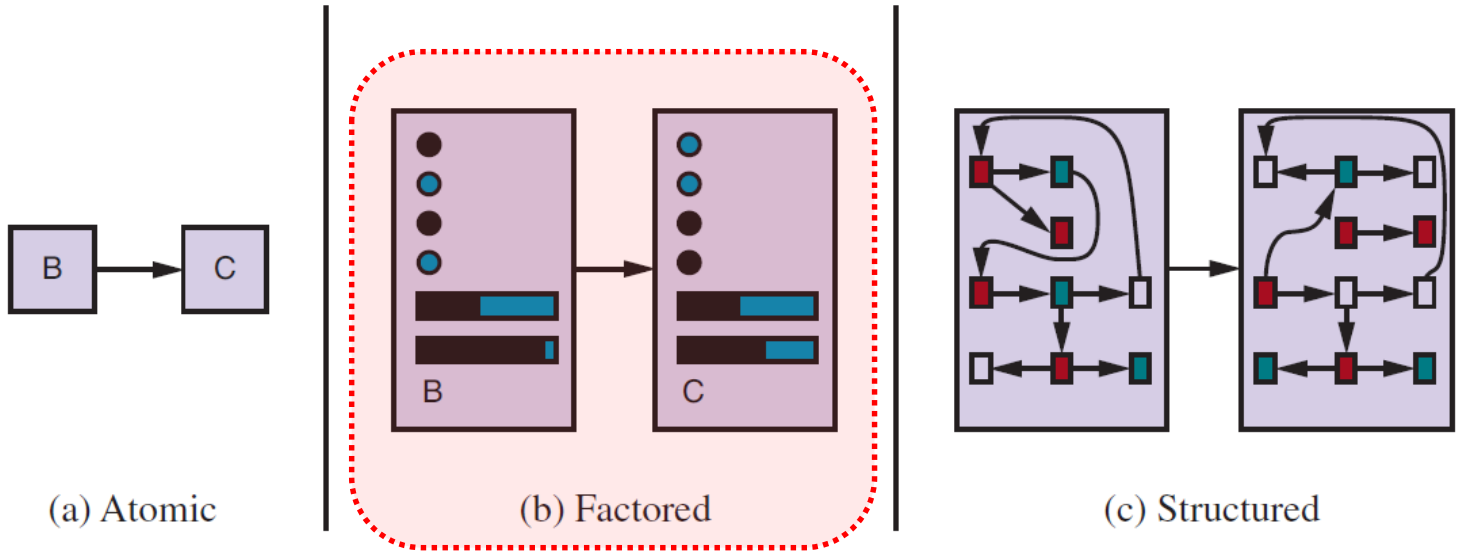


Figure 2.16 Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed set of symbols. (c) Structured representation: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.

CSP 정의

- CSP 정의
  - X is a set of variables, {X1, . . . ,Xn}.
  - D is a set of domains, {D1, . . . ,Dn}, one for each variable.
  - C is a set of constraints that specify allowable combinations of values.
- 조건
  - A domain, Di, consists of a set of allowable values, {v1, . . . ,vk}, for variable Xi.
  - Different variables can have different domains of different sizes.
- 예를 들어,
  - a Boolean variable would have the domain {true, false}.

조건(Constraint)의 정의

- 조건(constraint) Cj의 구성
  - 형식: <범위, 관계> (a pair <scope, rel>)
  - 예1: <(X1,X2),{(3,1), (3,2), (2,1)}>
  - 예2: <(X1,X2),X1 > X2>
- 할당
  - 변수에 값 할당, {Xi =vi,Xj =vj, . . .}
- 완전 할당
  - 문제를 구성하는 모든 변수에 값 할당
- 솔루션
  - 모든 조건을 만족하는 완전 할당

이진 CSP

단항 제약(unary contraint)은 단일 변수의 값에 제약 조건을 부가

$\langle(SA), SA \neq green\rangle$

이항 제약(binary contraint)은 두개의 변수에 제약 조건을 부가

$SA \neq WA$

고차 제약(higher order constraint) 세개 이상의 변수 제약 조건

삼항 제약의 예: $\langle(X, Y, Z), X < Y < Z \text{ or } X > Y > Z\rangle$

전역 제약(global contraint)은 임의의 개수의 변수를 포함할 수 있지만 (모든 변수를 포함할 필요는 없음).

Alldiff$(v_1, \ldots, v_k)$: 모든 변수 $v_1, \ldots, v_k$ 의 값이 모두 달라야 한다는 제약

예, 스도쿠 (행/열/3x3박스 각 측면에서 모든 변수가 달라야 함)

복면산

(Cryptarithmetic Puzzle)

<숫자가 "복면"을 쓰고 있는 연산>

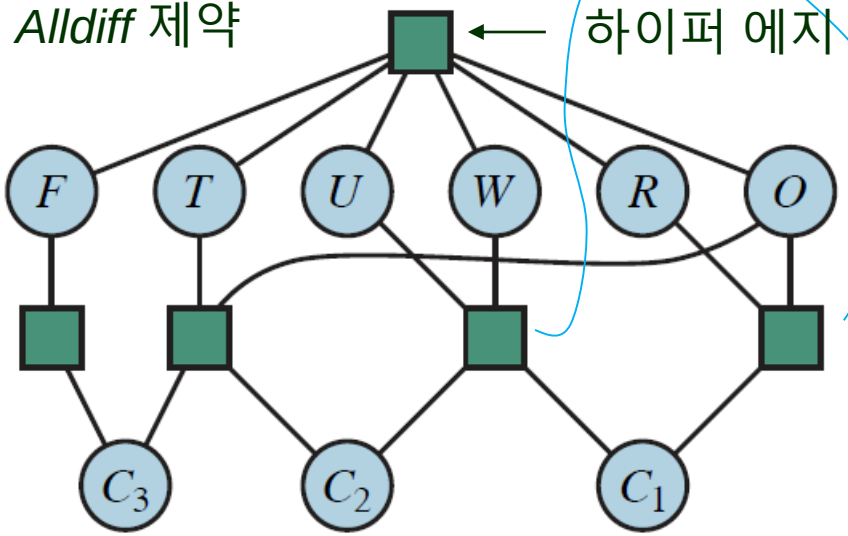
$$\begin{array}{cccc} & T & W & O \\ + & T & W & O \\ \hline F & O & U & R \end{array}$$

4열 짜리 덧셈 제약

$$\begin{aligned} O + O &= R + 10 \cdot C_1 \\ C_1 + W + W &= U + 10 \cdot C_2 \\ C_2 + T + T &= O + 10 \cdot C_3 \\ C_3 &= F \end{aligned}$$

십, 백, 천 자리에서, 자리 올림 발생

Alldiff 제약



제약 하이퍼 그래프

하이퍼그래프는 그래프를 일반화한 것으로, 에지(하이퍼에지라고 함)가 두 개 이상의 정점을 연결

이항 제약으로의 변환

모든 n-ary 제약은 보조 변수를 도입하여 이진 제약 집합으로 축소 변환 가능

⇩

어떤 CSP든 이진 CSP로 변환 가능

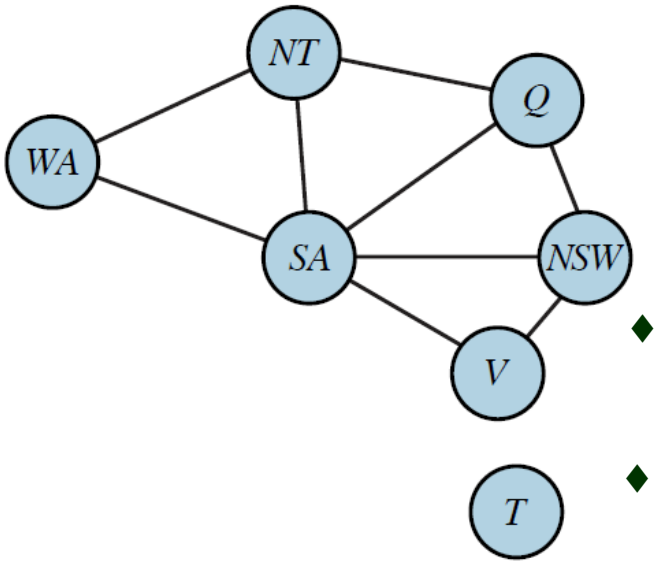
Alldiff와 같은 전역 제약 조건의 장점:

- 작성하기 쉽고 실수가 줄어듬
- 효율적인 특수 목적 추론 알고리즘 적용 가능

II. 제약 조건 전파

- 제약 조건을 사용하여 변수에 대한 합법적인 값의 수 축소
- 다른 변수에도 동일하게 적용

지역 단위로 관철된 일관성은 제약 그래프 전체 범위로 보면 모순될 수 있음
(전체 범위에서 해당 모순 제거)



- **노드 일치성**

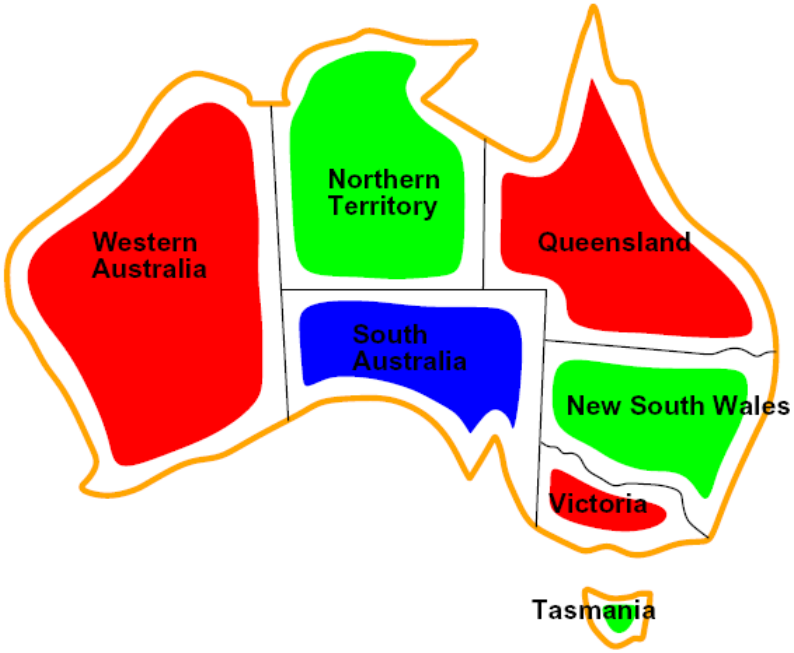
South Australians: 녹색을 거부 한다고 가정

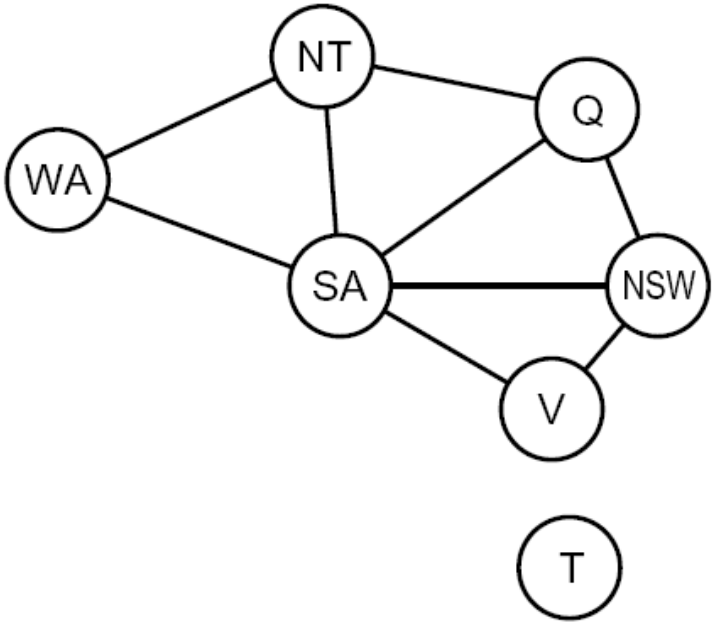
⇩

SA 의 도메인: {red, blue}

- 시작할 때 관련된 변수들의 도메인을 축소하는 방식으로 모든 단항 제약 제거
- 도메인의 모든 값이 해당 변수의 단항 제약조건을 만족하면 노드 일치성 성취

제약 조건 그래프

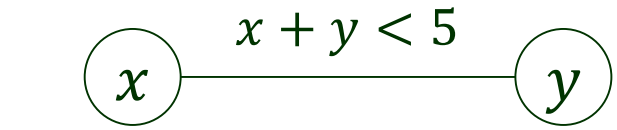

Western Australia
Northern Territory
Queensland
South Australia
New South Wales
Victoria
Tasmania


NT
Q
WA
SA
NSW
V
T

호 일치성

변수 Xi가 그와 연계된 다른 변수 Xj에 대해 다음을 만족하면, 변수 Xi는 변수 Xj에 대해 "**호 일치성**"을 갖는다.
즉, Xi의 각 값 Di와 그에 연계된 Xj의 각 값 Dj에 대해 해당 호(즉 에지 (Xi, Xj))에 부여된 이진 제약을 만족한다면.

교환 법칙이 성립하지 않음



$D_x = \{1, 2, 3\}$ $D_y = \{1,3, 5\}$

x 는 y에 대해 호 일치성을 갖는다.
역은 성립하지 않는다.

만약 해당 변수가 이진 제약조건을 공유하는 모든 다른 변수에 대해 이 조건을 만족한다면, 해당 변수는 **호 일치 변수**임.

제약 그래프를 구성하는 모든 변수가 호 일치 변수이면 해당 그래프는 **호 일치 그래프**임.

호 일치성 적용

예 1

$$Y = X^2 \quad \text{where } X, Y \in \{0,1, \ldots, 9\}$$

X arc-consistent w.r.t. Y ⟹ Reduced domain of X: $\{0, 1, 2, 3\}$

Y arc-consistent w.r.t. X ⟹ Reduced domain of Y: $\{0, 1, 4, 9\}$

예 2 호주 지도 색칠 문제

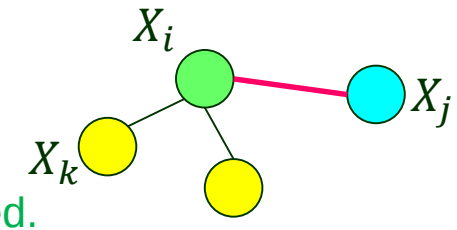
SA≠WA

두 변수의 도메인 {*red*, *green*, *blue*}에 축소 변화 없음

호 일치성 알고리즘: AC-3

function AC-3(*csp*) **returns** false if an inconsistency is found and true otherwise
  queue ← a queue of arcs, initially all the arcs in *csp*

  while *queue* is not empty **do**
    $(X_i, X_j) \leftarrow$ POP(*queue*)
    if REVISE(*csp*, X_i, X_j) **then** // domain of X_i has been reduced.
      if size of $D_i = 0$ **then return** *false*
      for each X_k **in** X_i.NEIGHBORS - $\{X_j\}$ **do** // propagate to other variables
        add (X_k, X_i) to *queue* // sharing a constraint with X_i
  return *true*



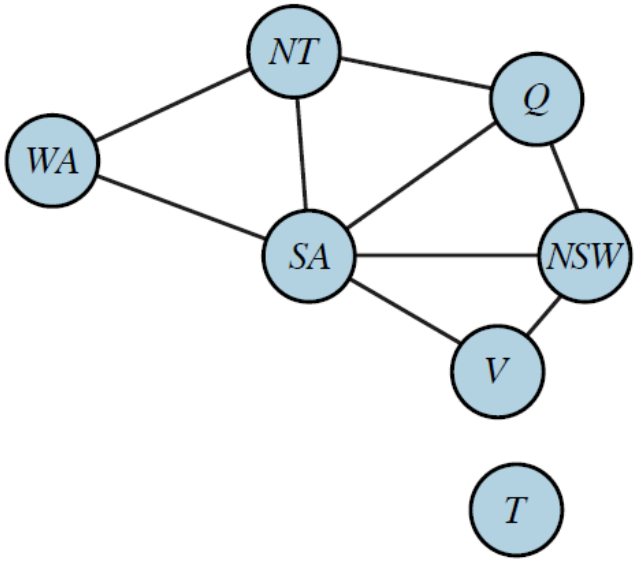
function REVISE(*csp*, X_i, X_j) **returns** true iff we revise the domain of X_i
  revised ← *false*
  for each x **in** D_i **do**
    if no value y in D_j allows (x,y) to satisfy the constraint between X_i and X_j **then**
      delete x from D_i
      revised ← *true*
  return *revised*

도메인 크기 $\leq d$, c 개의 이항 제약 조건

- 각 호가 큐로 들어가는 횟수 $\leq d$ 회
  (X_i 의 값 중 삭제할 수 있는 값 $\leq d$ 개
- 제약 관철 시간: $O(d^2)$

=> 총 복잡도: $O(cd^3)$

경로 일치성

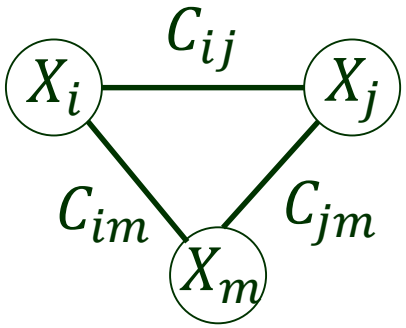


2색만 사용할 경우, 개별 변수들은 제약 조건을 만족하지만 솔루션은 부재 (예. 빨간색-파란색의 경우 임의의 Xi-Yi에 대해 제약 조건 만족. 그러나 2색으로 지도 칠하는 것은 불가능)

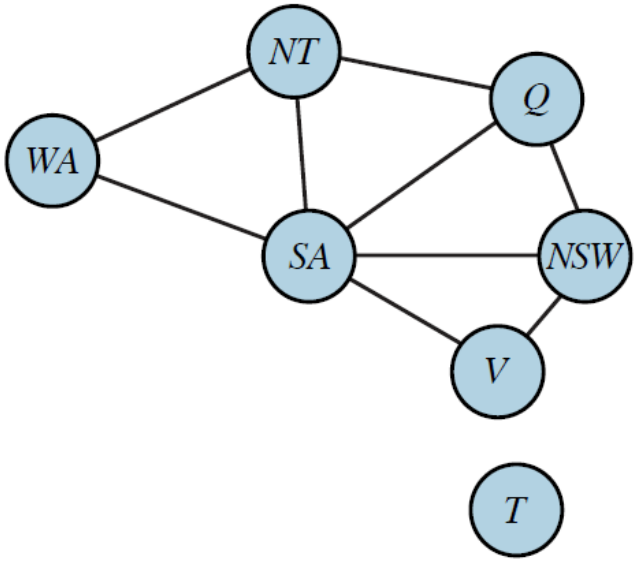
호 일치성은 도메인을 축소시켜 주기만 함

경로 일치성은 세 변수 쌍에서 추리한 암묵적 제약 조건을 조사하여 이용할 수 있게 해줌

$\{X_i, X_j\}$가 세번째 변수 X_m에 대해 **경로 일치성**을 갖는 경우 -->
즉, X_i 의 모든 할당에 대해 X_j 가 그들의 제약 C_{ij} 를 준수하는 상태에서, X_m 과 그들과의 제약조건 C_{im} 과 C_{jm} 을 만족하는 X_m 으로의 할당이 존재



경로 일치성의 적용



{*WA*, *SA*} 를 *NT*에 대해 경로 일치하도록 만들어 보자

두 색만 고려: *red*, *blue*.

아래 두가지 할당만 가능

{*WA* = *red*, *SA* = *blue*}

{*WA* = *blue*, *SA* = *red*}

⇩

두 경우 모두 *NT* 에 적절한 값을 할당할 수 없게 만듦

⇩

WA 와 *SA*로의 할당 제거

⇩

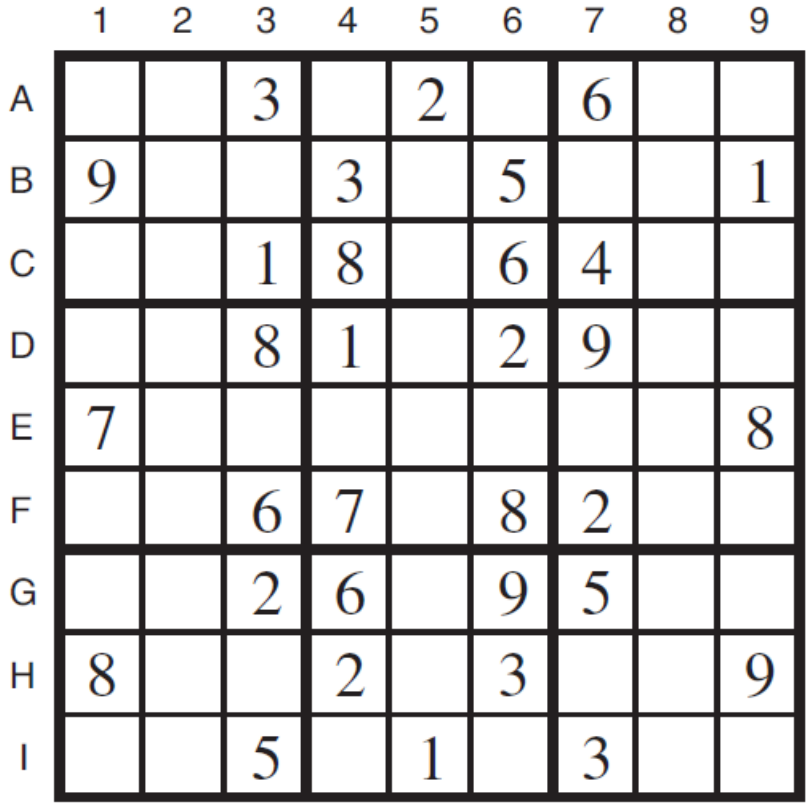
솔루션 못 찾음

스도쿠

9×9 격자에 숫자 1부터 9까지를 채우는데, 각 행, 열 또는 3×3 박스에 같은 숫자가 두 번 나타나지 않도록 하는 게임.

	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

CSP로 풀어보는 스도쿠



변수: $A1, \ldots, A9, B1, \ldots I9$

도메인: $D = \{1,2, \ldots, 9\}$

27 개의 *Alldiff* 제약 조건:

$Alldiff(A1, A2, A3, A4, A5, A6, A7, A8, A9)$

$\vdots$

$Alldiff(I1, I2, I3, I4, I5, I6, I7, I8, I9)$

$Alldiff(A1, B1, C1, D1, E1, F1, G1, H1, I1)$

$\vdots$

$Alldiff(A9, B9, C9, D9, E9, F9, G9, H9, I9)$

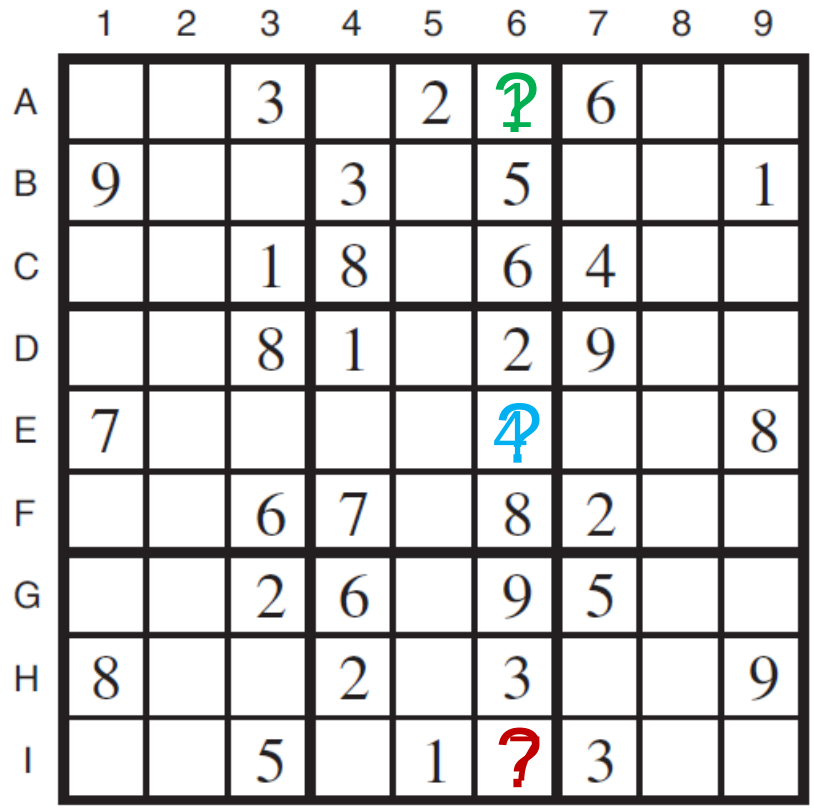
$Alldiff(A1, A2, A3, B1, B2, B3, C1, C2, C3)$

$\vdots$

$Alldiff(G7, G8, G9, H7, H8, H9, I7, I8, I9)$

- CSP 해법은 수천개짜리를 초 단위로 해결!
- 가장 간단한 형태의 스도쿠만 AC-3로 해결 가능

CSP로 스도쿠를 푸는 예



	1	2	3	4	5	6	7	8	9
A			3		2	?	6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7					4			8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1	?	3		

Consider E6:

$D_{E6} \leftarrow \{1,2,\dots,9\}$

⇩ constraints in the box

$D_{E6} \leftarrow D_{E6} \setminus \{1,2,7,8\}$

$= \{3,4,5,6,9\}$

⇩ constraints in the column

$D_{E6} \leftarrow D_{E6} \setminus \{2,3,5,6,8,9\}$

$= \{4\}$

Consider I6:

$D_{I6} \leftarrow \{1,2,\dots,9\}$

⇩ constraints in the column

$D_{I6} \leftarrow D_{I6} \setminus \{2,3,4,5,6,8,9\} = \{1,7\}$

⇩ constraints in the box

$D_{I6} \leftarrow D_{I6} \setminus \{1,2,3,6,9\} = \{7\}$

Consider A6:

$D_{A6} \leftarrow \{1,2,\dots,9\}$

⇩ constraints in the column

$D_{A6} \leftarrow \{1\}$

6주차 과제

CSP 방법으로 아래의 스도쿠 문제를 푸는 과정을 단계 별로 상세히 설명하라.