Foundations of Constraint Satisfaction
CSCE 990-005, Fall 2000

- Welcome
- Sign-up sheet
- Check `www.cse.unl.edu/~choueiry/F00-CSE990/`

Berthe Y. Choueiry (Shu-we-ri)
Ferguson Hall, Room 104
`choueiry@cse.unl.edu`
`Tel: (402)472-5444`

Admin (I)

Time: M-W-F: 10.30–11.20 a.m.

Office hours: M-W: 11:30 a.m.–12:30 p.m.

Course:

- Credit: 3
- Theory for CS, Software Systems for CSE
- Undergrad welcome: in the future, CSCE426/826
- Lectures by instructor
- Occasionally, presentations by students

Admin (II)

Rules:

- Absence: 6 sessions max.
- Collaboration: encouraged, except for..

Grading:

- Quizzes: 20%, cannot not be made up
- Assignments: 30% (programming, pen+paper)
- One paper presentation: 15%
- Project: 35% (30% project, 5% presentation)

Improve your grades:

- Bonus for: 100% attendance, class interaction, ‘riddles,’ *etc.*
- Critical summary of a studied technical paper (10% per summary)

How to write a critical summary

- We will discuss an example in class
- Generic template provided as an aid but is not mandatory
  - PART I: the paper
  - PART II: your opinion about the paper

PART I: the paper

- Context of the paper
  - problem the authors claim to address (*i.e.*, motivation)
  - assumptions they make
  - solution they claim to provide
- Short Description of proposed technique
  - basic algorithmic steps
  - optimizations, if any
  - evaluation: empirical/theoretical
- Comparison to previous techniques
  - if provided, how?
  - can you identify/propose some other?
- Directions for future research

PART II: Your opinion

- is the paper a ‘real’ advancement of the state of the art?
- is it useful for the theory? for practice?
- can you identify other uses of the technique proposed?
- what are the shortcomings?
- can you identify more? can you propose a fix?
- are there issues swept-under-the-carpet?
- can you identify other directions for future research?

Content of the course

- Introduction: definition and practical examples
- Foundations and basic mechanisms
- Advanced solving techniques
- Extensions to the problem definition
- Alternative approaches to solving the problem

Course support

- Book by Tsang
- Manuscript by Dechter
- Papers from: WWW, course web-page, library,
  electronic reserves, me etc.

Pointers to resources

Check

Textbooks: Tsang'93 (out of print), Dechter (unpublished)

Web: `www.cse.unl.edu/~choueiry/F00-CSE990/`,
Archives at CCC @UNH, `comp.constraints`,
`csp@carlit.toulouse.inra.fr`

Conferences: CP (+WSs), IJCAI, ECAI, NCAI (AAAI), FLAIRS, etc.

Journals: Constraints, AIJ, JACM, AI+Math, etc.

Constraint course challenge: *the best student (ties broken by lottery) will be awarded a trip to a major international conference in the field.*

Your future: jobs

- Commercial companies: i2 Technologies, Trilogy, Ilog, PeopleSoft/Red Pepper, Carmen Systems (Sweden), etc.
- Prestigious research centers: NASA Ames, JPL, BT Labs (UK), IBM Watson+Alamaden, etc.
- Start your own: Blue Pumpkin, Parc Technologies Ltd etc...
- Academic: constraint languages, modeling, representation, reasoning mechanisms

Announcement: *two prominent researchers from two important companies will be visiting the course during the semester*

Constraint Satisfaction 101

- Definition, representation & characterization
- Application examples
- Solution techniques
- Issues & research directions

CSP: Definition

Given $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$:

- $\mathcal{V}$ a set of variables
  $$\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$$
- $\mathcal{D}$ a set of variable domains (domain values)
  $$\mathcal{D} = \{D_{V_1}, D_{V_2}, \ldots, D_{V_n}\}$$
- $\mathcal{C}$ a set of constraints
  $$C_{V_a, V_b, \ldots, V_i} = \{\, (x, y, \ldots, z)\} \subseteq D_{V_a} \times D_{V_b} \times \ldots \times D_{V_i}$$

Query: can we find one value for each variable
such that all constraints are satisfied?

In general, **NP-complete**

Representation (I)

Given $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C}) \begin{cases} \mathcal{V} = \{V_1, V_2, \ldots, V_n\} \\ \mathcal{D} = \{D_{V_1}, D_{V_2}, \ldots, D_{V_n}\} \\ \mathcal{C} \text{ set of constraints} \end{cases}$

$$C_{V_k,V_l,V_m} = \{\, (x, y, z)\} \subseteq D_{V_k} \times D_{V_l} \times D_{V_m}$$

Constraint **graph**

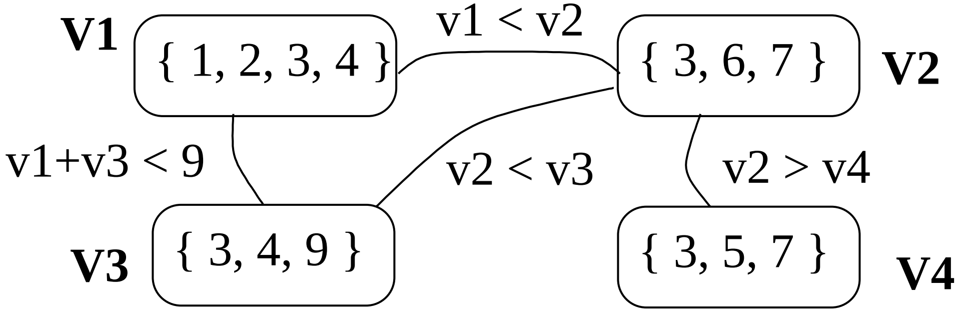
- Variable $\longrightarrow$ node (vertex)
- Domain $\longrightarrow$ node label
- Constraint $\longrightarrow$ arc (edge) between nodes

Representation (II)

Given $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ $\begin{cases} \mathcal{V} = \{V_1, V_2, \ldots, V_n\} \\ \mathcal{D} = \{D_{V_1}, D_{V_2}, \ldots, D_{V_n}\} \\ \mathcal{C} \text{ set of constraints} \end{cases}$

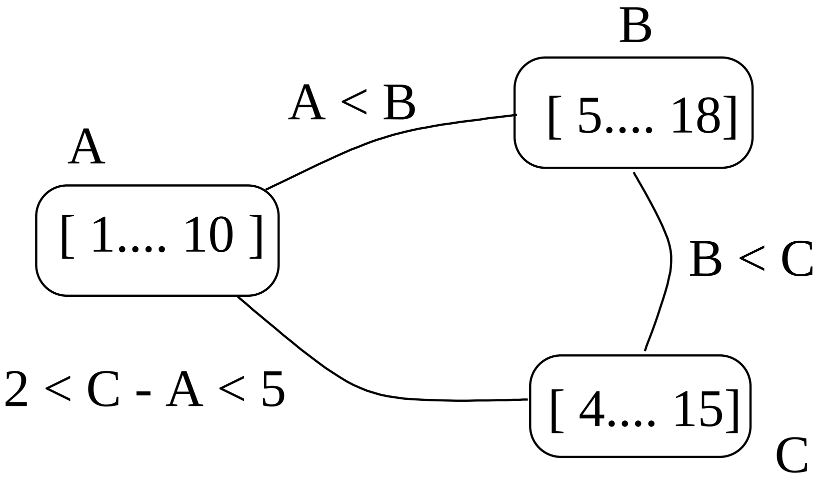
$$C_{V_k, V_l, V_m} = \{\, (x, y, z)\} \subseteq D_{V_k} \times D_{V_l} \times D_{V_m}$$

Example I:



Define $\mathcal{C}$?

Example II: Temporal reasoning



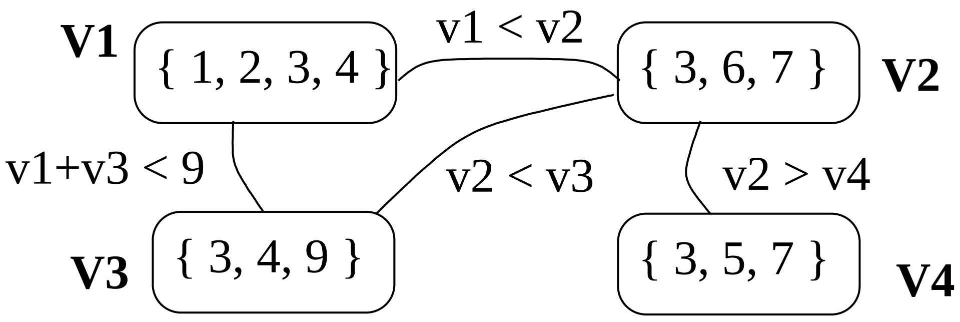
Give one solution: ......

Satisfaction, yes/no: decision problem

Constraint arity

Given $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C}) \begin{cases} \mathcal{V} = \{V_1, V_2, \ldots, V_n\} \\ \mathcal{D} = \{D_{V_1}, D_{V_2}, \ldots, D_{V_n}\} \\ \mathcal{C} \text{ set of constraints} \end{cases}$

$$C_{V_k,V_l,V_m} = \{\, (x,y,z)\} \subseteq D_{V_k} \times D_{V_l} \times D_{V_m}$$



How to represent the constraint $V_1 + V_2 + V_4 < 10$

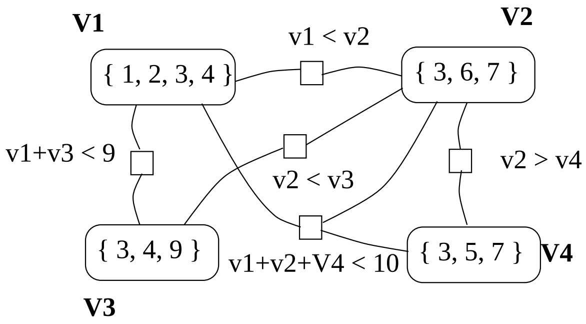
Constraint arity (II)

Given $\mathcal{P} = (\mathcal{V}, \mathcal{D}, \mathcal{C}) \begin{cases} \mathcal{V} = \{V_1, V_2, \ldots, V_n\} \\ \mathcal{D} = \{D_{V_1}, D_{V_2}, \ldots, D_{V_n}\} \\ \mathcal{C} \text{ set of constraints} \end{cases}$

$$C_{V_k,V_l,V_m} = \{\, (x, y, z)\} \subseteq D_{V_k} \times D_{V_l} \times D_{V_m}$$

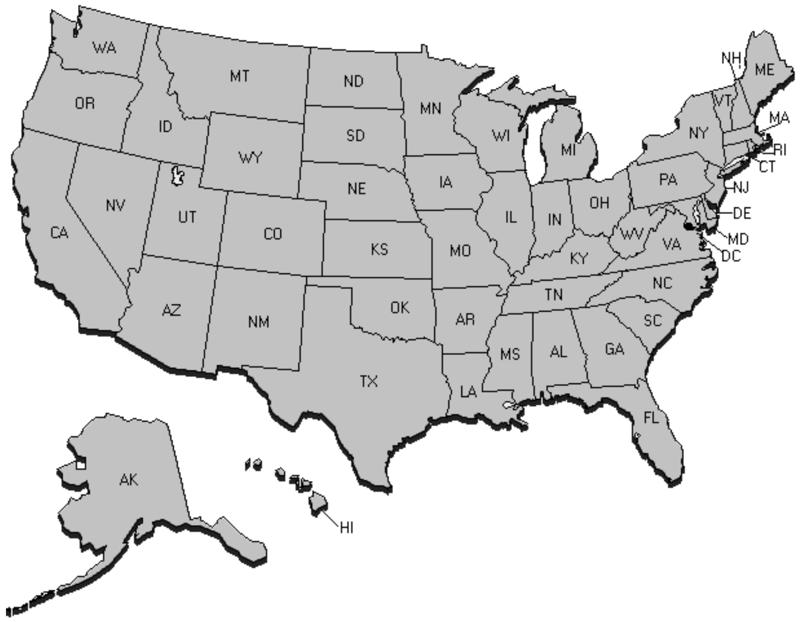
Constraints: universal, unary, binary, ternary, ... global

Constraint network



Example III: Map coloring

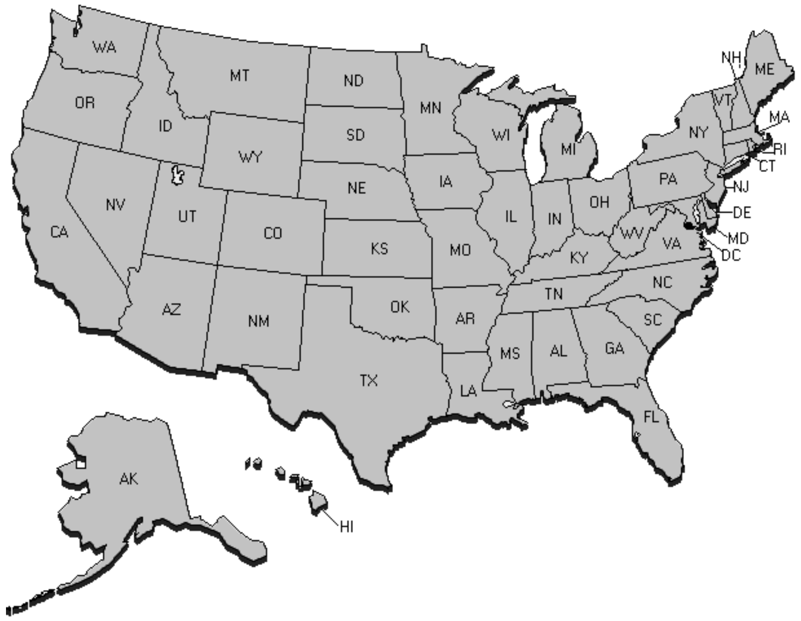
Using 3 colors (R, G, & B), color the US map such that no two adjacent states do have the same color

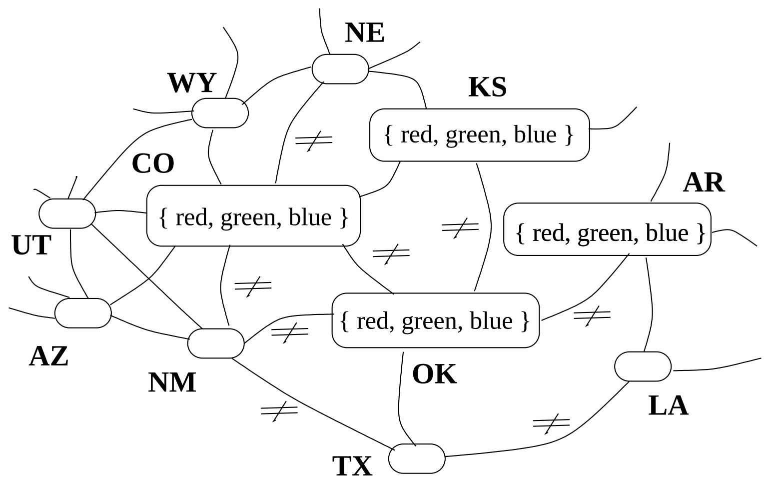


Variables? Domains? Constraints?

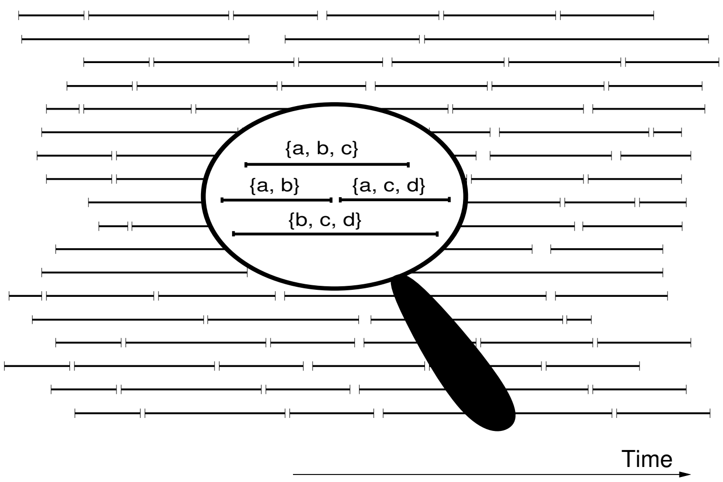
Example III: Map coloring

Using 3 colors (R, G, & B), color the US map such that no two adjacent states do have the same color



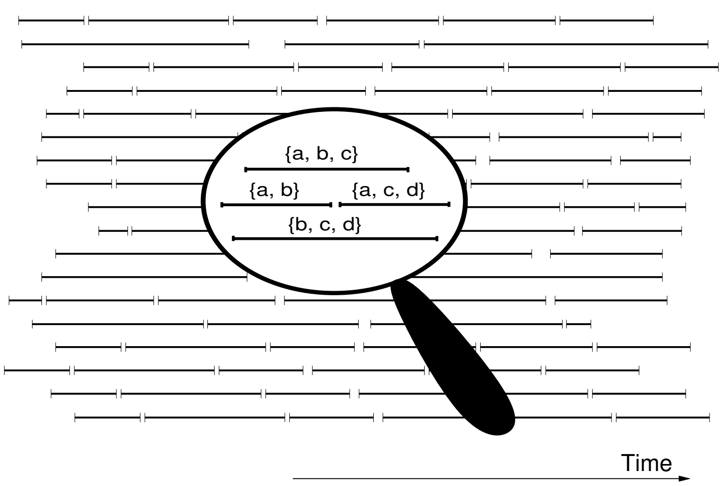


Example IV: Resource Allocation

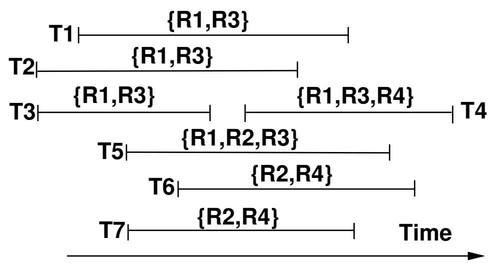


CSP formulation?

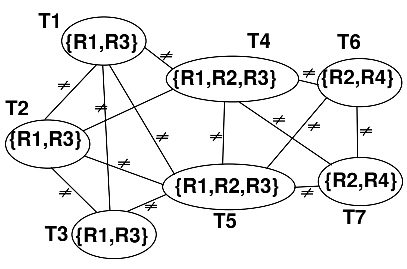
Example IV: Resource Allocation



Interval Order



Constraint graph



Example V: Puzzle

Given: Four musicians, Fred, Ike, Mike, and Sal play bass, drums, guitar and keyboard, not necessarily in that order. They have 4 successful songs, “Blue Sky”, “Happy Song”, “Gentle Rhythm” and “Nice Melody.” Ike and Mike are, in one order or the other, the composer of “Nice Melody” and the keyboardist, etc..

Query: who plays which instrument and who composed which song?

Example V: Puzzle

Formulation 1

- Variables: Bass, Drums, Guitar, Keyboard, Blue Sky, Happy Song, Gentle Rhythm and Nice Melody.
- Domains: { Fred, Ike, Mike, Sal }
- Constraints: ...

Formulation 2

- Variables:
  Fred's-instrument, Ike's-instrument, Mike's-instrument..,
  Fred's-song, Ikes's-song, ..
- Domains:
  { bass, drums, guitar, keyboard }
  { Blue Sky, Happy Song, Gentle Rhythm, Nice Melody}
- Constraints: ...

Example VI: Product configuration

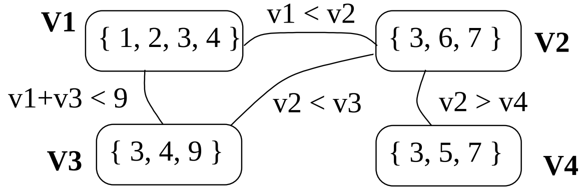
Train, elevator, car, etc.

Given: components and their attributes (variables)
domain covered by each characteristic (values)
relations among the components (constraints)
a set of required functionalities (more constraints)

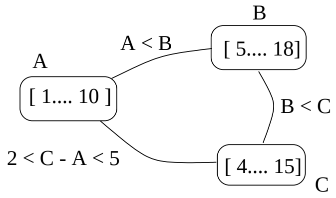
Find: a product configuration
i.e., an acceptable combination of components
that realizes the required functionalities

Constraint types

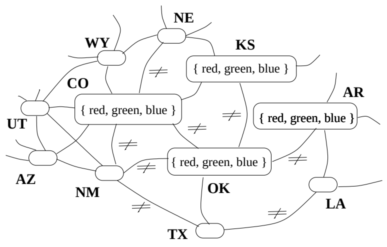
- Example I: algebraic constraints

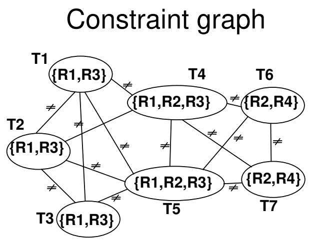


- Example II: (algebraic) constraints of bounded difference



- Example III & VI: coloring, mutual exclusion, difference constraints





- Example V & VI: elements of $\mathcal{C}$ must be made explicit

VII. Databases

- Join operation in relational DB is a CSP
- View update is a CSP

VIII. Interactive systems

Dataflow constraints

- Spreadsheets
- Graphical layout systems and animation
- Graphical user interfaces

IX. Molecular biology (bioinformatics)

Threading

etc.

Advantages:

Flexible, expressive, compact representation

Related areas:

AI, OR, Algorithmics, DB, Prog. Languages