Foundations of Constraint Processing
CSCE421/821, Fall 2002

- Welcome
- GTA: Mr Andrew Breiner (abreiner@cse.unl.edu)
- Sign-up sheet
- Check `www.cse.unl.edu/~choueiry/F02-421-821/`

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

Admin (I)

Time: Tuesday, Thursday: 9.30–10.45 a.m.

Office hours:

- Instructor: Tue, Thur: 10:45 a.m.–12:00 p.m.
- GTA: Mon, Wed 10:00 am to 11:00 a.m.
- GRAs: Tue, Thu, Fri. **Course:**
- Theory track, 3 credit/hours
- Lectures by instructor
- Occasionally, presentations by students
- Communications: web page, email `cse421@cse.unl.edu`,

Rules:

- Absence: 4 sessions max.
- Collaboration: encouraged, except for homework, code, etc.

Admin (II)

Grading:

- Pretest: 3%
- Quizzes: 32%, cannot not be made up
- Assignments: 30% (programming, pen+paper)
- Search/Project: 35%
- $(\neg$mid-term$) \wedge (\neg$final$)$

Improve your grades:

- Present a research paper (10% per presentation)
- Critical summary of a research paper (5% per summary)
- Write a chapter of a 'textbook' (20% total)
- Glossary, weekly and final (5% total)
- Bonus for: 100% attendance, class interaction, 'riddles,' *etc.*

Important dates:

- **Regularly** check class schedule on WWW
- Thursday, Aug 28th: pretest (20 min. max)
- Thursday, Sep 5th: project topic must be chosen.
- Tue, Oct 16: progress reports/pre-presentation due
- Tue, Dec 10: final reports due
- Tue, Dec 10, Thu, Dec 13 (dead weak), & Tue Dec 17:
  project presentations

Projects:

- Search competition (Lisp)

- Implement and evaluate an algorithm
- Model and solve a (simple) practical problem
- Investigate an advanced theoretical concept
- Conduct a critical literature survey (at least 3 papers)

$\longrightarrow$ Propose your own or discuss with instructor
$\longrightarrow$ Work with a GRAP (Buettner, Guddeti, Xu, Zou)

How to write a critical summary

Generic template provided as an aid but is not mandatory

- PART I: your understanding of 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 (on reserve at LL, out of print)
- Manuscript by Dechter (to appear in January, 2003)
- Papers from: WWW, course web-page, library,
  electronic reserves, instructor, etc.

Pointers to resources

Web:

- Some links: `www.cse.unl.edu/~choueiry/F02-421-821/`
- Archives at CCC @UNH
- Benchmark problems: `www.csplib.org`
- eLists: `csp@carlit.toulouse.inra.fr`, `comp.constraints`

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

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

Your future: jobs

- Commercial companies: i2 Technologies, In Time Systems Inc, Trilogy, Ilog, PeopleSoft/Red Pepper, Carmen Systems (Sweden), etc.
- Prestigious research centers: NASA Ames, JPL, BT Labs (UK), IBM Watson+Almaden, etc.
- Start your own: Blue Pumpkin, Parc Technologies Ltd etc.
- Academic:
  - constraint languages
  - modeling and constraint representation
  - reasoning and propagation mechanisms
  - dedicated reasoning: diagnosis, CBR, planning & scheduling, design, etc.
  - etc.

Constraint Satisfaction 101

- Motivating example, application areas
- Definition, representation & characterization
- Application examples
- Solution techniques
- Issues & research directions

What is this about?

Context: You are a senior in college

Problem: You need to register in 4 courses for Spring'2000

Possibilities: Many courses offered in Math, CSE, EE, *etc.*

Constraints: restrict the choices you can make

- *Unary:* Courses have prerequisites you have/don't have
  Courses/instructors you like/dislike
- *Binary:* Courses are scheduled at the same time
- *n-ary:* In CompEng, 4 courses from 5 tracks such as at least 3 tracks are covered

You have choices, but are restricted by constraints

$\longrightarrow$ make the right decisions

Practical Applications

Adapted from E.C. Freuder

- Radio resource management (RRM)
- Databases (computing joins, view updates)
- Temporal and spatial reasoning
- Planning, scheduling, resource allocation
- Design and configuration
- Graphics, visualization, interfaces
- Hardware verification and software eng.
- HCInteraction and decision support
- Molecular biology
- Robotics, machine vision and computational linguistics
- Transportation
- Qualitative and diagnostic reasoning

Constraint Satisfaction

Given

- A set of variables *4 courses at UNL*
- For each variable, a set of choices (values)
- A set of constraints that restrict the combinations of values the variables can take at the same time

Questions

- Does a solution exist? *classical decision problem*
- How two or more solutions differ? How to change specific choices without perturbing the solution?
- If there is no solution, what are the sources of conflicts? Which constraints should be retracted?
- *etc.*

Constraint Processing is about

- solving a decision problem
- while allowing the user to state arbitrary constraints in an expressive way and
- providing concise and high-level feedback about alternatives and conflicts

Power of Constraints Processing

- flexibility & expressiveness of representations
- interactivity, users can $\left\{\begin{array}{c}\text{relax}\\\text{reinforce}\end{array}\right\}$ constraints

Related areas: AI, OR, Algorithmic, DB, Prog. Languages, *etc.*

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)

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

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

Constraint **graph**

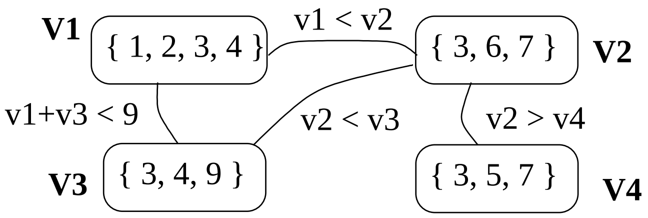
- Variable $\longrightarrow$ node (vertex)
- Domain $\longrightarrow$ node label
- Constraint $\longrightarrow$ arc (edge) between nodes

Representation (II)

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

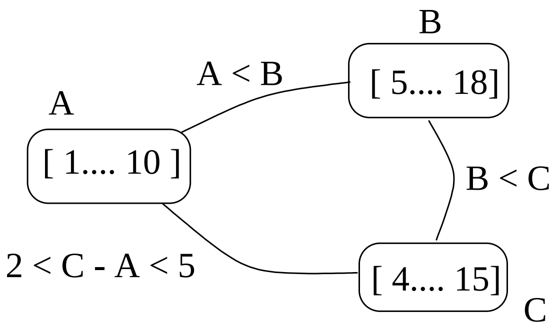
$$C_{V_k,V_l,V_m} = \{\,(x, y, z)\} \subseteq D_{V_k} \times D_{V_l} \times D_{V_m}$$

Example I:



Define $\mathcal{C}$?

Example II: Temporal reasoning



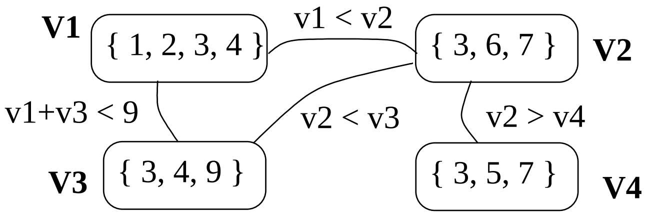
Give one solution: ......

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 $V_1 + V_2 + V_4 < 10$

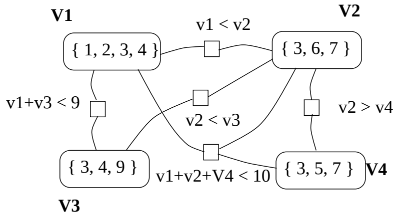
Constraint arity (II)

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

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

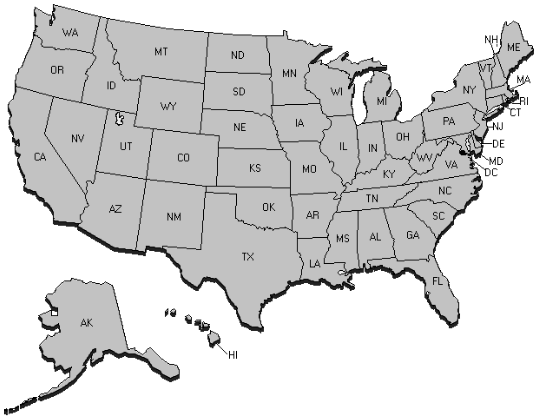
Constraints: universal, unary, binary, ternary, ... global

Constraint network



Example III: Map coloring

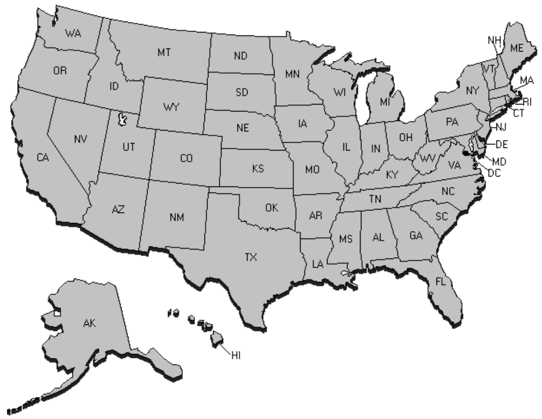
Using 3 colors (R, G, & B), color the US map such that no two adjacent states do have the same color

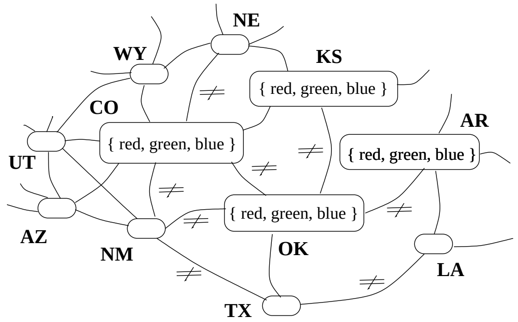


Variables? Domains? Constraints?

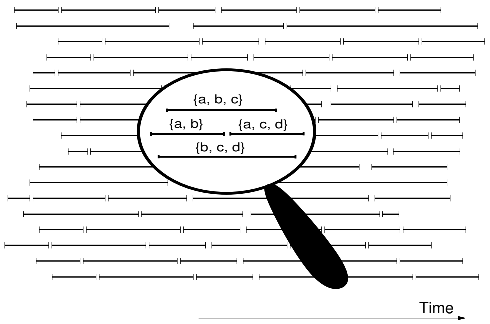
Example III: Map coloring

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



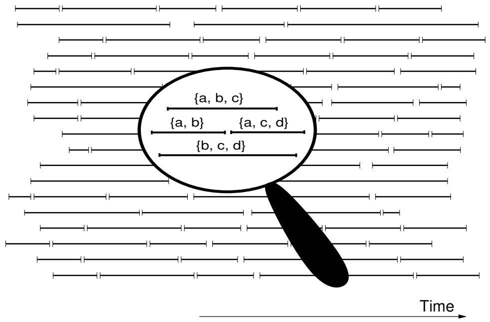


Example IV: Resource Allocation

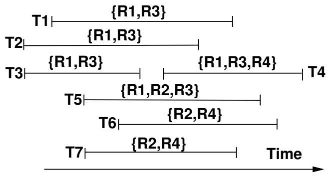


CSP formulation?

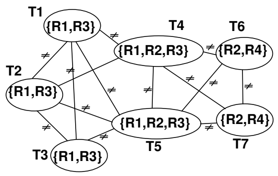
Example IV: Resource Allocation



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

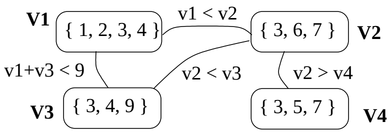
Train, elevator, car, etc.

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

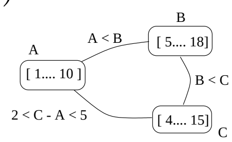
Find: a product configuration
i.e., an acceptable combination of components
that realizes the required functionalities

Constraint types

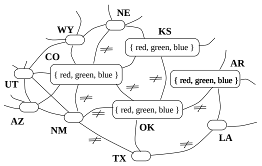
- Example I: algebraic constraints

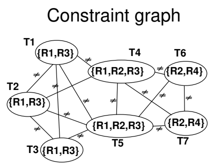


- Example II: (algebraic) constraints of bounded difference



- 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 materialization is a CSP

VIII. Interactive systems

Data-flow 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, Algorithmic, DB, Prog. Languages