CS 1571 Introduction to AI
Review

Midterm review

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Search

- **Basic definition of the search problem**
  – Search space, operators, initial state, goal condition
- **Formulation of a problem:**
  – We have some control over the complexity of the **search space size**
- **Two types:**
  – Path vs. configuration search

Search

- **Methods for searching the search space:**
- **Search trace captured by the search tree**

- **Search methods properties :**
  - Completeness, Optimality, Space and time complexity.
- **Complexities**
  - measured in terms of a branching factor (*b*), depth of the optimal solution (*d*), maximum depth of the state space (*m*)

Search

- **Uninformed methods:**
  - Breadth first search, Depth first search, Iterative deepening, Bi-directional search, Uniform cost search (for the weighted path search)
- **Informed methods:**
  - **Heuristic function** (*h*): potential of a state to reach the goal
  - **Evaluation function (*f*) :** desirability of a state to be expanded next
  - **Best first search:**
    - Greedy $f(n) = h(n)$
    - A*: $f(n) = g(n) + h(n)$

      the role of admissible heuristics, optimality

Search

- **Constraint satisfaction problem (CSP)**
  - Variables, constraints on values (reflect the goal)
  - Formulation of a CSP as search
  - Methods and heuristics for CSP search
    - Backtracking, constraint propagation, most constrained variable, least constrained value

- **Complex configuration searches. Use iterative algorithms:**
  - **Methods:** Hill climbing, Simulated annealing, Genetic algorithms
  - **Advantage**: **memory !! Useful for very large optimization problems.**

Search

- **Adversarial search (game playing)**
  - Specifics of a game search, game problem formulation
  - rational opponent
- **Algorithms:**
  - **Minimax algorithm**
    - Complexity bottleneck for large games
  - **Alpha-Beta pruning**: prunes branches not affecting the decision of players
  - **Cutoff** of the search tree and heuristics

KR and logic

- **Knowledge representation:**
  - **Syntax** (how sentences are build), **Semantics** (meaning of sentences), **Computational aspect** (how sentences are manipulated)
- **Logic:**
  - A formal language for expressing knowledge and ways of reasoning
  - **Three components:**
    - A set of sentences
    - A set of interpretations
    - The valuation (meaning) function

Propositional logic

- A language for symbolic reasoning
- **Language:**
  - Syntax, Semantics
- **Satisfiability** of a sentence: at least one interpretation under which the sentence can evaluate to ***True***.
- **Entailment:**

$KB \models \alpha$ is true in all worlds in which KB is true

- **Inference procedure**
  - Soundness If $KB \vdash_i \alpha$ then $KB \models \alpha$
  - Completeness If $KB \models \alpha$ then $KB \vdash_i \alpha$

Propositional logic

- **Logical inference problem:** $KB \models \alpha$?
  - Does KB entail the sentence α ?
- Logical inference problem for the propositional logic is **decidable**.
  - A procedure (program) that stops in finite time exists
- **Approaches:**
  - Truth table approach
  - Inference rule approach
  - Resolution refutation

$KB \models \alpha$ if and only if

$(KB \wedge \neg \alpha)$ is **unsatisfiable**

- **Normal forms**: DNF, CNF, Horn NF (conversions)

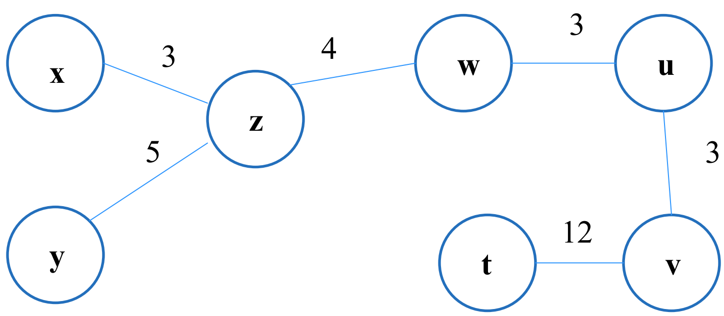
First order logic

- Deficiencies of propositional logic
- **First order logic (FOL):**

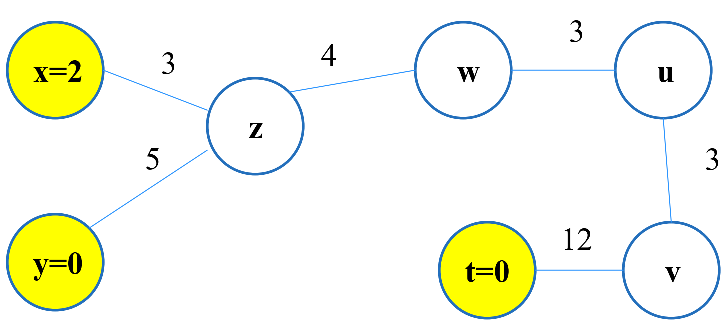
  allows us to represent objects, their properties, relations and statements about them
  - Variables, predicates, functions, quantifiers
  - Syntax and semantics of the sentences in FOL
- Translation of English sentences to FOL

CSP

- **Constraint propagation**
- Congruency network from the homework

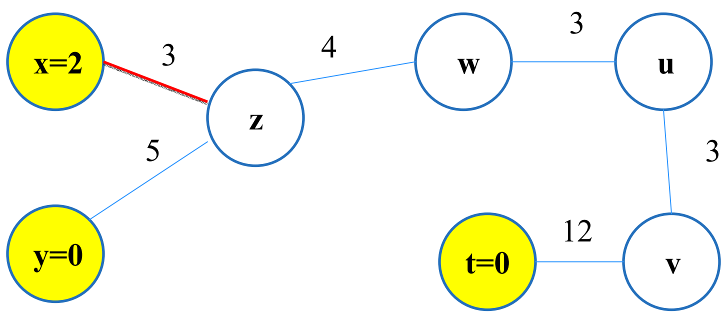


CSP



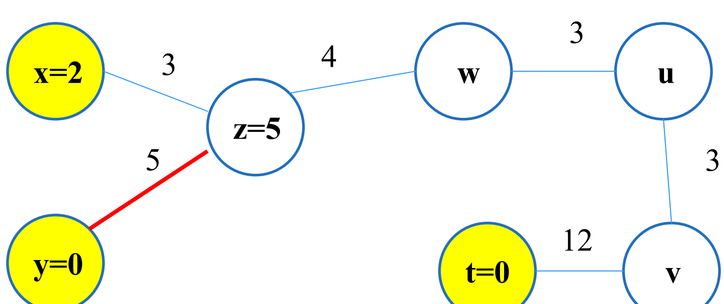
Initial assignments

CSP



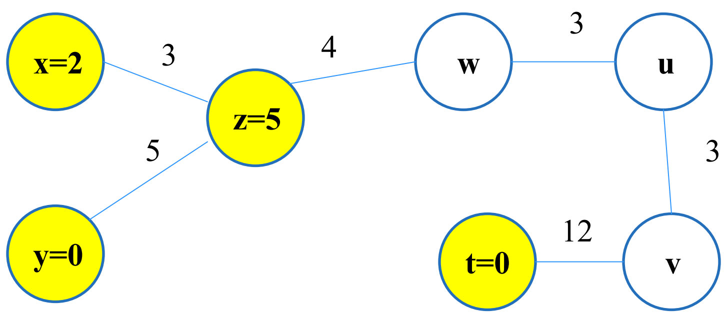
z≠ 0, 1, 3, 4, 6, 7, 9

CSP



z≠ 0, 1, 3, 4, 6, 7, 9, **2, 8**

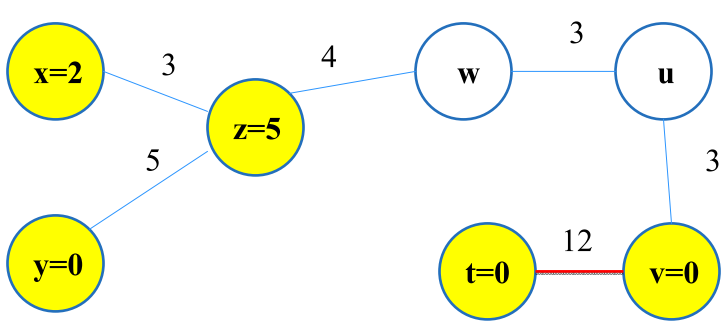
CSP



z≠ 0, 1, 3, 4, 6, 7, 8, 2, 9

z= 5 via exhautions of alternatives

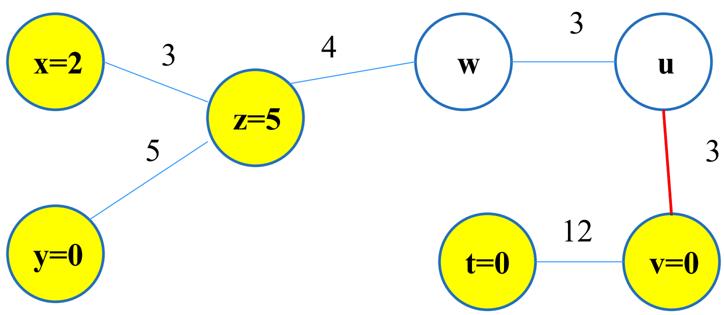
CSP



v≠ 1, 2, 3, 4, 5, 6, 7, 8, 9

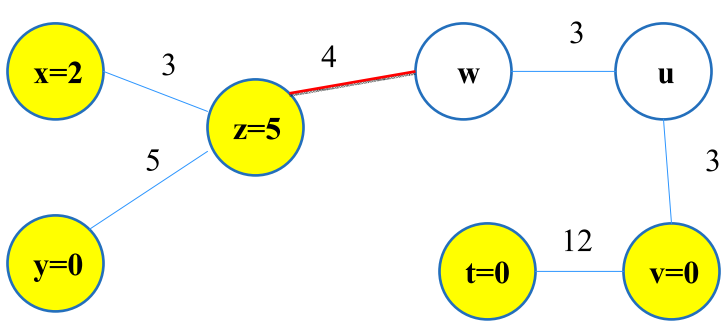
v= 0 via exhaustion of alternatives

CSP



u ≠ 1, 2, 4, 5, 7, 8 → u={0,3,6,9}

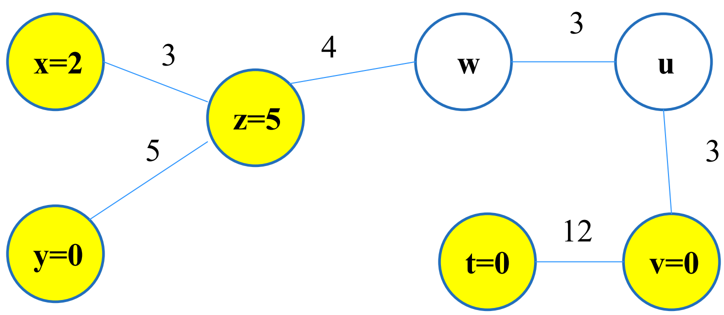
CSP



u ≠ 1, 2, 4, 5, 7, 8 → u={0,3,6,0}

w ≠ 0, 2, 3, 4, 6, 7, 8 → w= {1,5,9}

CSP



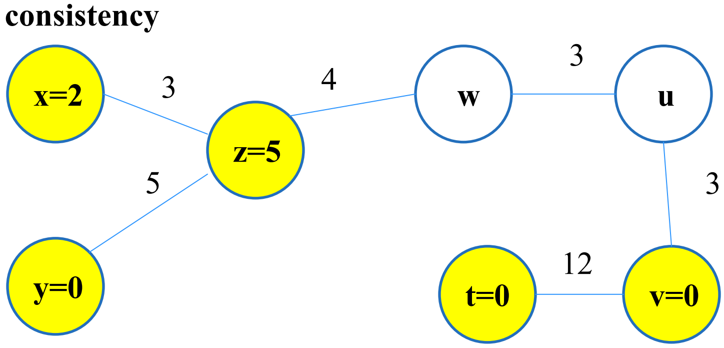
u ≠ 1, 2, 4, 5, 7, 8 → u={0,3,6,0}

w ≠ 0, 2, 3, 4, 6, 7, 8 → w= {1,5,9}

Forward checking stops

CSP

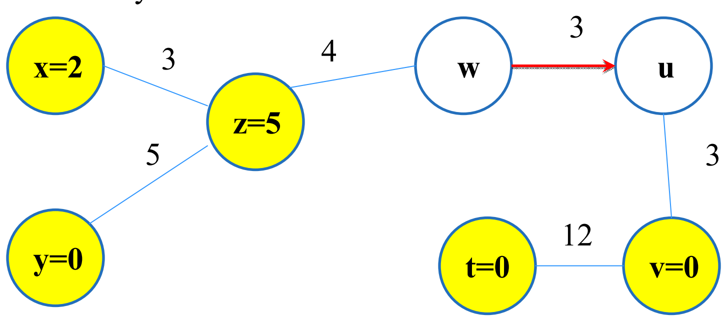
Arc consistency



w= {1,5,9} & u={0,3,6,0}

CSP

Arc consistency



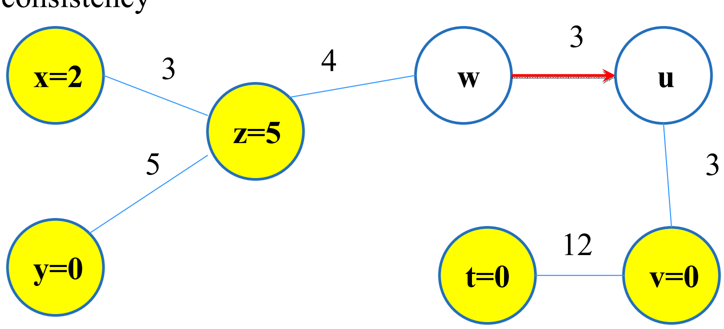
w= {1,5,9}   & u={0,3,6,9}

w= 1 → no valid value for u exists (u should be either 1,4,7}

Hence w ≠ 1 → w= {5,9}

CSP

Arc consistency



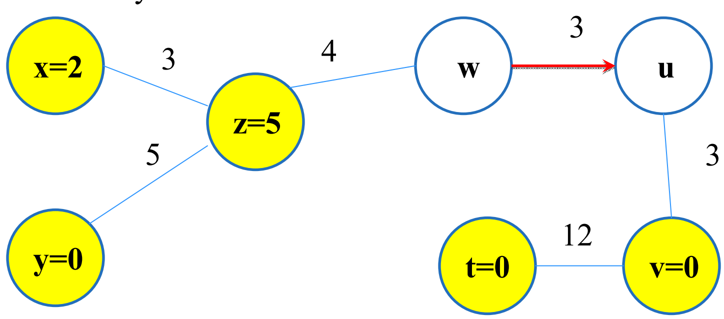
w= {5,9}   & u={0,3,6,9}

w= 5 → no valid value for u exists (u should be either 2,5,8}

Hence w ≠ 5 → w= {9}

CSP

Arc consistency



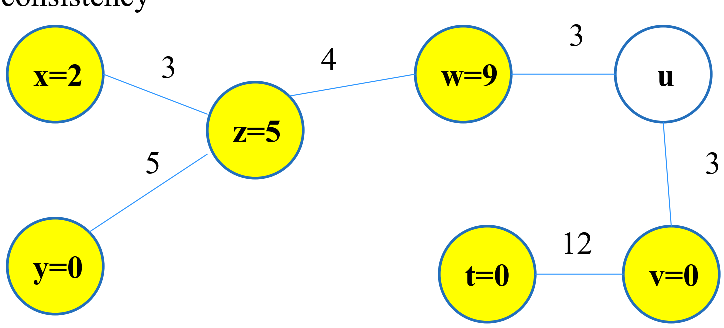
w= {9}   & u={0,3,6,9}

w= 9 → consistent with the remaining u values

Hence w= {9}

CSP

Arc consistency

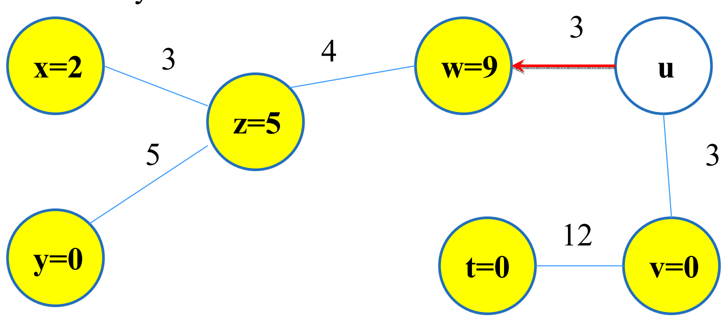


w= {9}   & u={0,3,6,9}

w= 9 → exhaustions of alternatives

CSP

Arc consistency



u={0,3,6,9} & w= {9}

All u consistent are consistent with w=9

Arc consistency stops