Advanced Topics in AI
Filtering

Instructor: Prof. Dr. techn. Wolfgang Nejdl

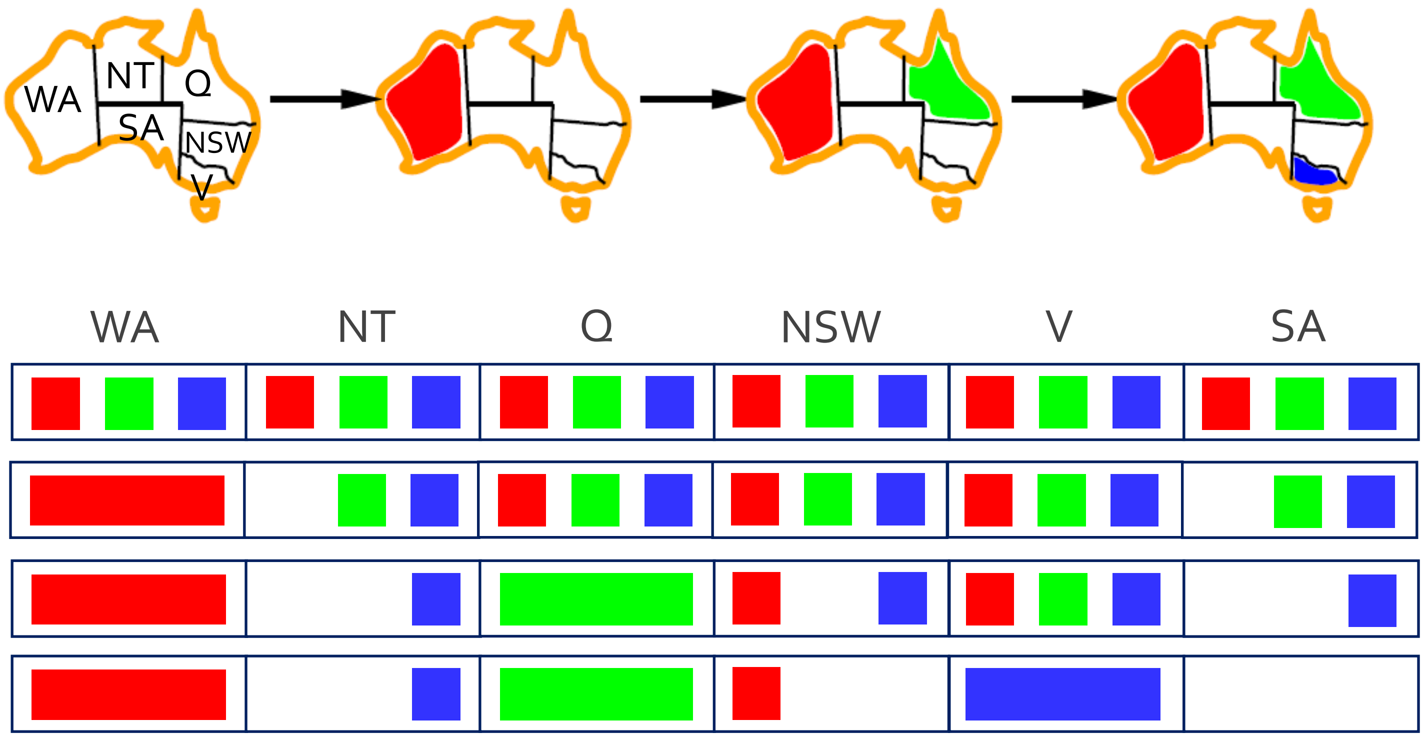
Leibniz University Hannover

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All materials are available at http://ai.berkeley.edu.]

Co-financed by the Connecting Europe Facility of the European Union

Filtering: Forward Checking

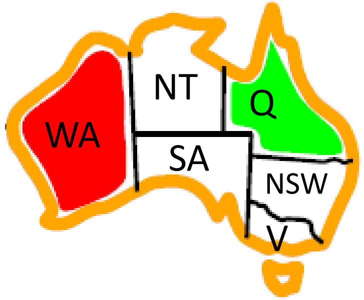
- Filtering: Keep track of domains for unassigned variables and cross off bad options
- Forward checking: Cross off values that violate a constraint when added to the existing assignment

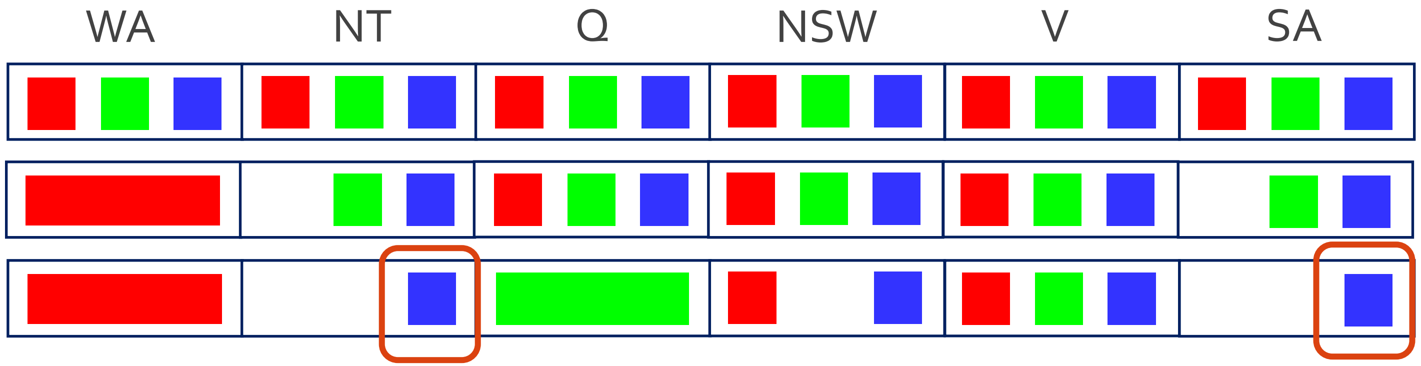


Co-financed by the Connecting Europe Facility of the European Union

Filtering: Constraint Propagation

- Forward checking propagates information from assigned to unassigned variables, but doesn't provide early detection for all failures:



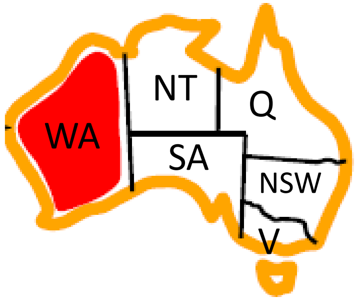


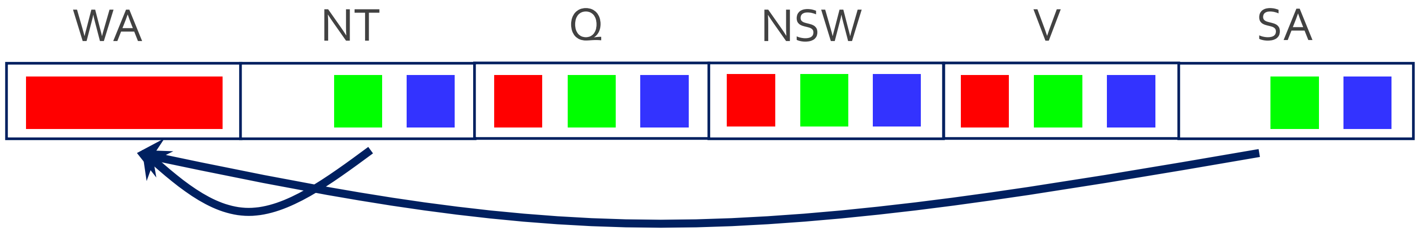
- NT and SA cannot both be blue!
- Why didn't we detect this yet?
- *Constraint propagation:* reason from constraint to constraint

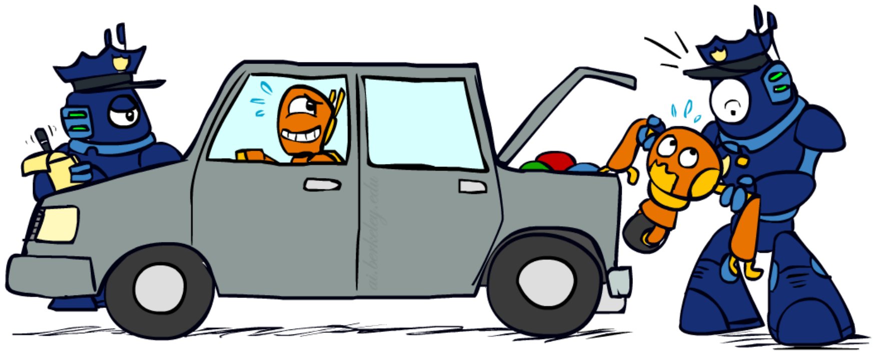
Co-financed by the Connecting Europe Facility of the European Union

Consistency of A Single Arc

- An arc X $\rightarrow$ Y is consistent iff for *every* x in the tail there is *some* y in the head which could be assigned without violating a constraint







Delete from the tail!

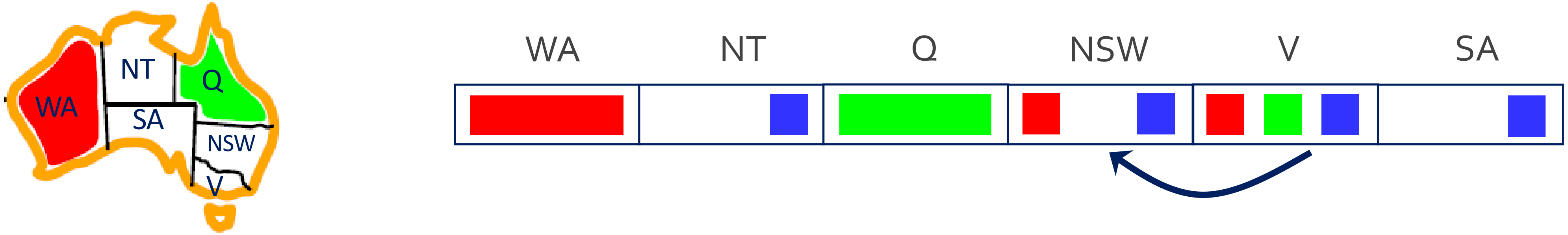
Forward checking?

Enforcing consistency of arcs pointing to each new assignment

Co-financed by the Connecting Europe Facility of the European Union

Arc Consistency of an Entire CSP

- A simple form of propagation makes sure all arcs are consistent:

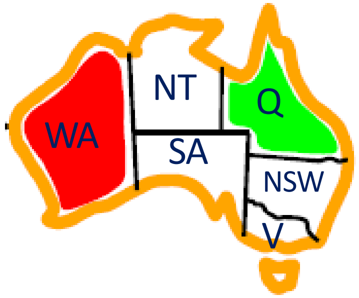


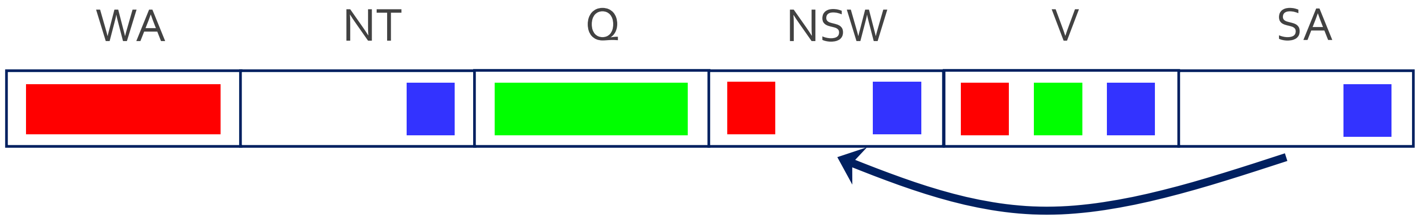
- Arc V to NSW is consistent: for *every* x in the tail there is *some* y in the head which could be assigned without violating a constraint

Co-financed by the Connecting Europe Facility of the European Union

Arc Consistency of an Entire CSP

- A simple form of propagation makes sure all arcs are consistent:



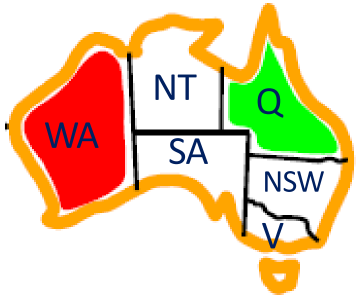


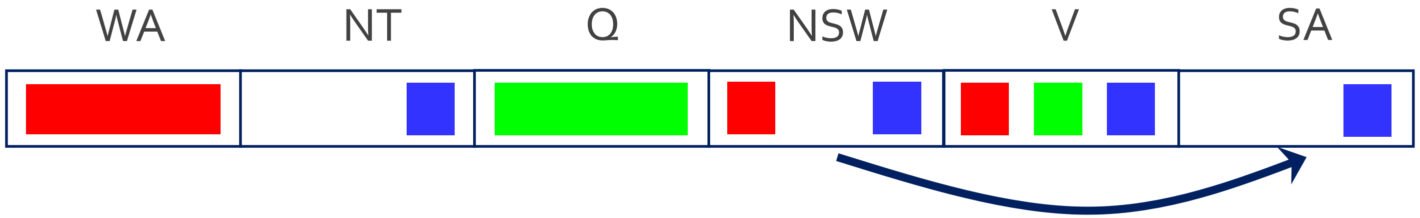
- Arc SA to NSW is consistent: for *every* x in the tail there is *some* y in the head which could be assigned without violating a constraint

Co-financed by the Connecting Europe Facility of the European Union

Arc Consistency of an Entire CSP

- A simple form of propagation makes sure all arcs are consistent:



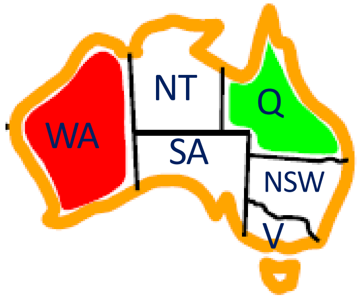


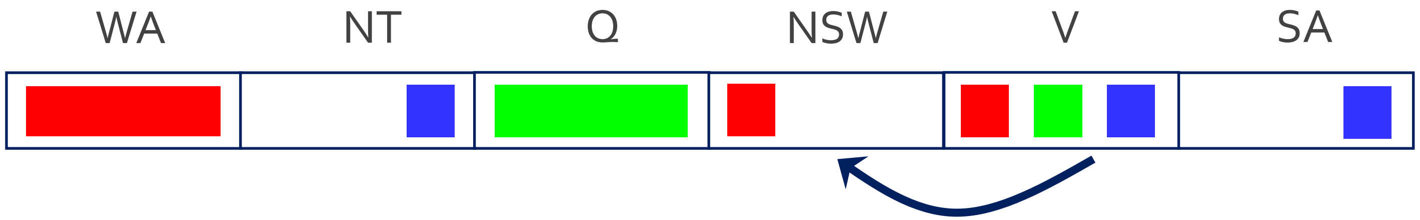
- Arc NSW to SA is not consistent: if we assign NSW = blue, there is no valid assignment left for SA
- To make this arc consistent, we delete NSW = blue from the tail

Co-financed by the Connecting Europe Facility of the European Union

Arc Consistency of an Entire CSP

- A simple form of propagation makes sure all arcs are consistent:





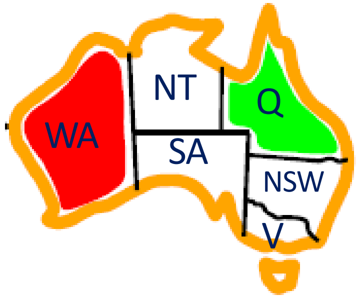
- Remember that arc V to NSW was consistent, when NSW had red and blue in its domain
- After removing blue from NSW, this arc might not be consistent anymore! We need to recheck this arc.

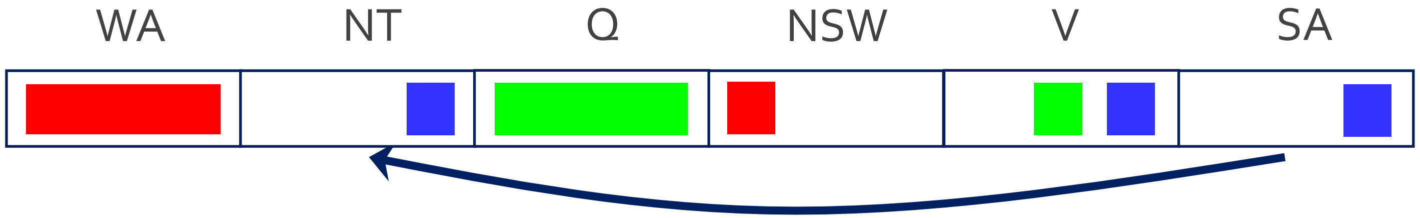
- Important: If X loses a value, neighbors of X need to be rechecked!

Co-financed by the Connecting Europe Facility of the European Union

Arc Consistency of an Entire CSP

- A simple form of propagation makes sure all arcs are consistent:



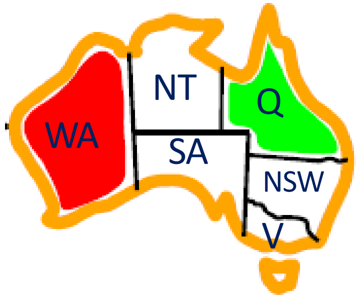


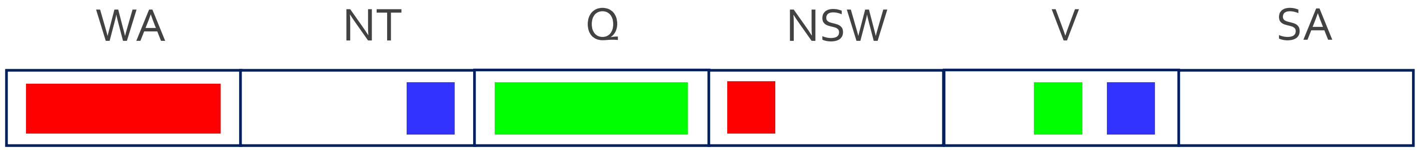
- Arc SA to NT is inconsistent. We make it consistent by deleting from the tail (SA = blue).

Co-financed by the Connecting Europe Facility of the European Union

Arc Consistency of an Entire CSP

- A simple form of propagation makes sure all arcs are consistent:





- SA has an empty domain, so we detect failure. There is no way to solve this CSP with WA = red and Q = green, so we backtrack.
- Arc consistency detects failure earlier than forward checking
- Can be run as a preprocessor or after each assignment

Co-financed by the Connecting Europe Facility of the European Union

Enforcing Arc Consistency in a CSP

function AC-3(*csp*) **returns** the CSP, possibly with reduced domains
 inputs: *csp*, a binary CSP with variables $\{X_1, X_2, \ldots, X_n\}$
 local variables: *queue*, a queue of arcs, initially all the arcs in *csp*

 while *queue* is not empty **do**
  $(X_i, X_j) \leftarrow$ REMOVE-FIRST(*queue*)
  if REMOVE-INCONSISTENT-VALUES(X_i, X_j) **then**
   for each X_k **in** NEIGHBORS[X_i] **do**
    add (X_k, X_i) to *queue*

---

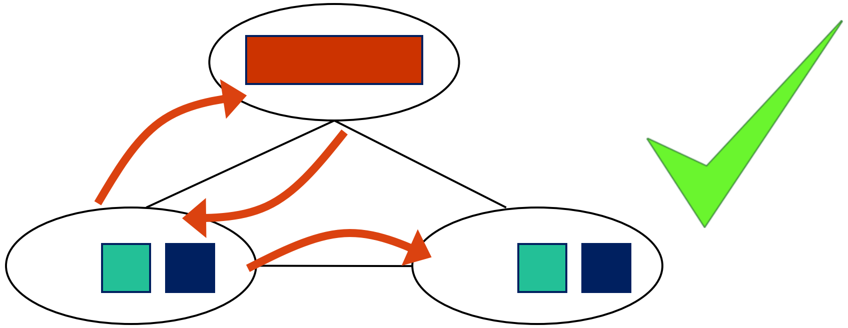
function REMOVE-INCONSISTENT-VALUES(X_i, X_j) **returns** true iff succeeds
 removed ← *false*
 for each *x* **in** DOMAIN[X_i] **do**
  if no value *y* in DOMAIN[X_j] allows (*x*, *y*) to satisfy the constraint $X_i \leftrightarrow X_j$
   then delete *x* from DOMAIN[X_i]; *removed* ← *true*
 return *removed*

- Runtime: $O(n^2d^3)$, can be reduced to $O(n^2d^2)$
- … but detecting all possible future problems is NP-hard – why?

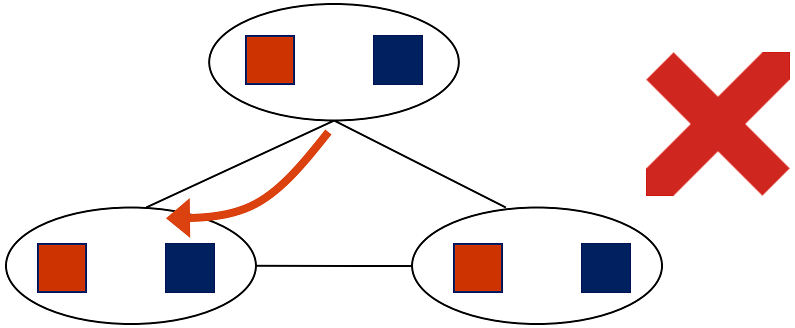
Co-financed by the Connecting Europe Facility of the European Union

Limitations of Arc Consistency

- After enforcing arc consistency:
  - Can have one solution left
  - Can have multiple solutions left
  - Can have no solutions left (and not know it)



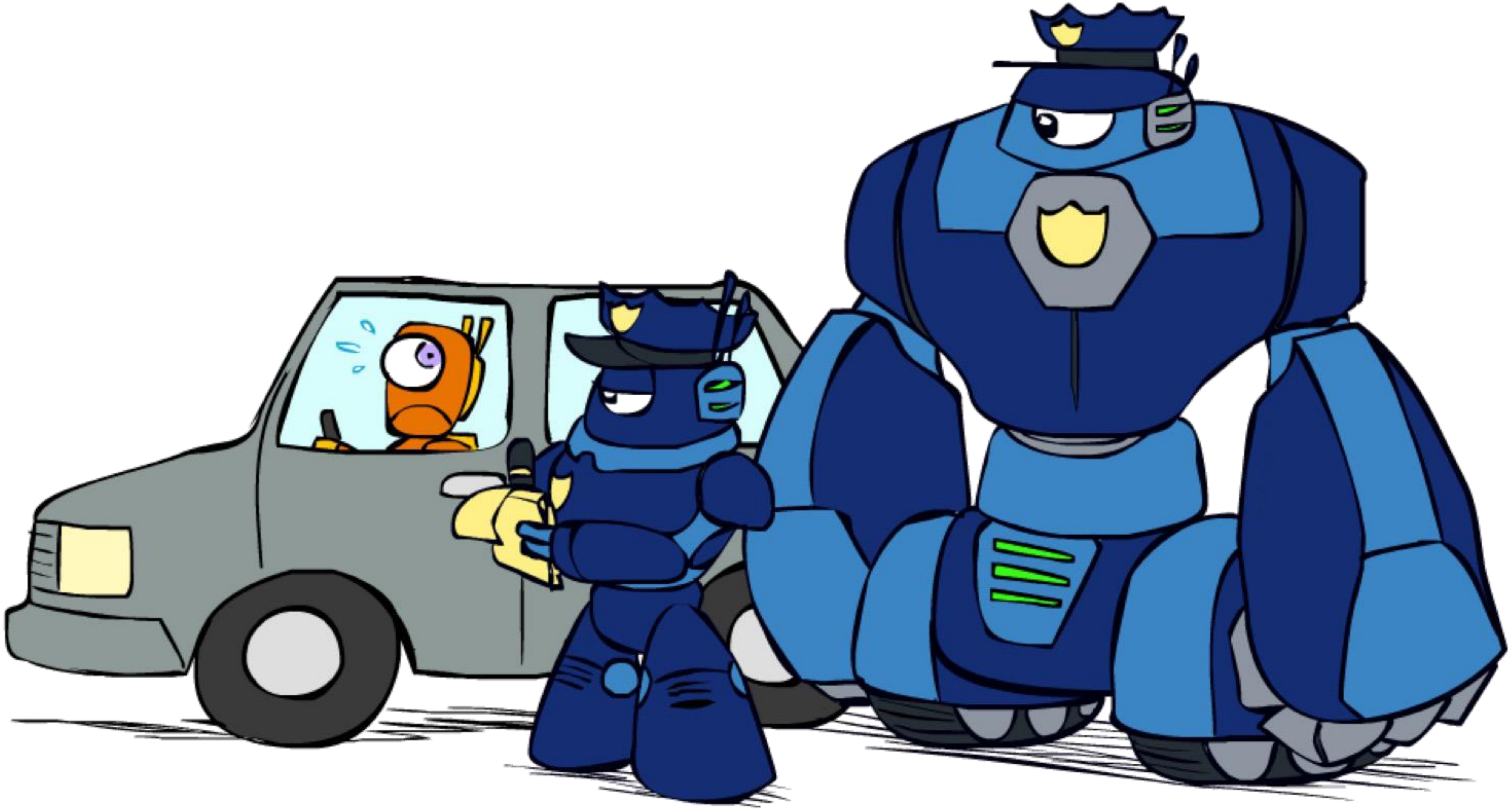
- Arc consistency still runs inside a backtracking search!



What went wrong here?

Co-financed by the Connecting Europe Facility of the European Union

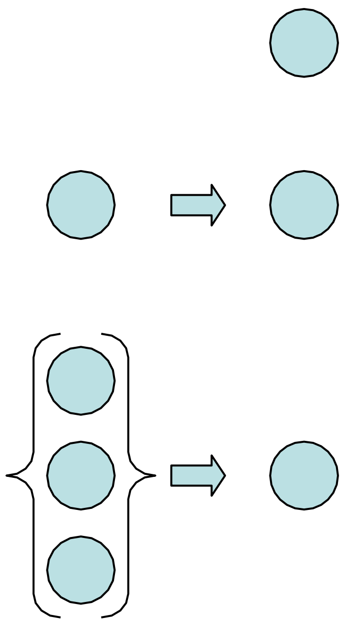
K-Consistency



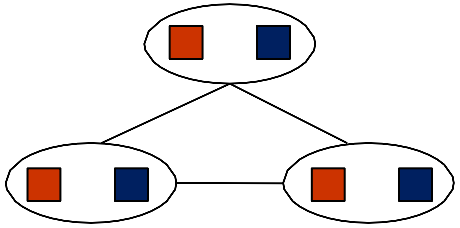
Co-financed by the Connecting Europe Facility of the European Union

K-Consistency

- Increasing degrees of consistency
  - 1-Consistency (Node Consistency): Each single node's domain has a value which meets that node's unary constraints
  - 2-Consistency (Arc Consistency): For each pair of nodes, any consistent assignment to one can be extended to the other
  - K-Consistency: For each k nodes, any consistent assignment to k-1 can be extended to the k^{th} node.



- Higher k more expensive to compute
- (You need to know the k=2 case: arc consistency



Co-financed by the Connecting Europe Facility of the European Union

Strong K-Consistency

- Strong k-consistency: also k-1, k-2, … 1 consistent
- Claim: strong n-consistency means we can solve without backtracking!
- Why?
  - Choose any assignment to any variable
  - Choose a new variable
  - By 2-consistency, there is a choice consistent with the first
  - Choose a new variable
  - By 3-consistency, there is a choice consistent with the first 2
  - …
- Lots of middle ground between arc consistency and n-consistency! (e.g. k=3, called path consistency)

Co-financed by the Connecting Europe Facility of the European Union

Advanced Topics in AI

Next: Ordering

Instructor: Prof. Dr. techn. Wolfgang Nejdl

Leibniz University Hannover

Co-financed by the Connecting Europe Facility of the European Union