Multiagent Systems (BE4M36MAS)

Distributed Constraints Satisfaction

Karel Horák, José Hilario

Artificial Intelligence Center,
Department of Computer Science,
Faculty of Electrical Engineering,
Czech Technical University in Prague

horakka5@fel.cvut.cz
hilarjos@fel.cvut.cz

November 19, 2019

Constraint Satisfaction Problem (CSP)

Find an assignment for variables that satisfy given constraints.

- $\mathcal{X} = \{x_1, \ldots, x_n\}$ — set of *variables* to assign.
- $\mathcal{D} = \{D_1, \ldots, D_n\}$ — set of *domains* ($x_i \in D_i$).
- $\mathcal{C} = \{C_1, \ldots, C_m\}$ — set of *constraints*.

$C_i \subseteq D_{i_1} \times \cdots \times D_{i_r}$ denotes a r-ary constraint over variables $x_{i_1}, \ldots, x_{i_r}$.

Constraint Satisfaction Problem (CSP)

Solution: n-tuple $(d_1, \cdots, d_n)$, such that:

- $d_i \in D_i$, for $1 \leq i \leq n$.
- $(d_{i_1}, \ldots, d_{i_r}) \in C_k$ for every constraint $C_k \subseteq D_{i_1} \times \cdots \times D_{i_r}$.

Centralized algorithm

Synchronized backtracking

$v_i \leftarrow$ value from D_i consistent with $(v_1, \ldots, v_{i-1})$;
if *No such* v_i *exists* **then**
  backtrack ;
else if $i = n$ **then**
  stop ;
else
  `ChooseValue`$(x_{i+1}, (v_1, \ldots, v_i))$;
end

Algorithm 1: `ChooseValue`$(x_i, (v_1, \ldots, v_{i-1}))$

Distributed Constraint Satisfaction Problem (DCSP)

- $\mathcal{X} = \{x_1, \ldots, x_n\}$ — set of *variables* to assign.
- $\mathcal{D} = \{D_1, \ldots, D_n\}$ — set of *domains* ($x_i \in D_i$).
- $\mathcal{C} = \{C_1, \ldots, C_m\}$ — set of *constraints*.
- $\mathcal{A} = \{A_1, \ldots, A_k\}$ — set of *agents*.

Every variable **must be** assigned to one of the agents.

$\rightarrow$ otherwise the DCSP problem is **not fully defined**.

Asynchronous Backtracking (ABT)

Assumptions:

- Every agent controls a single variable.
- Agents communicate via messages.
- Constraints are binary.
- Messages are delivered in a finite time (but this time may vary randomly).
- Whenever an agent A sends messages to agent B, agent B receives them in the same order as A sent them.

Asynchronous Backtracking (ABT)

Initial knowledge of the agent:

- Total ordering (priorities) of agents.
- Constraints he is involved in.
- Domain of the variable that he controls.

Asynchronous Backtracking (ABT)

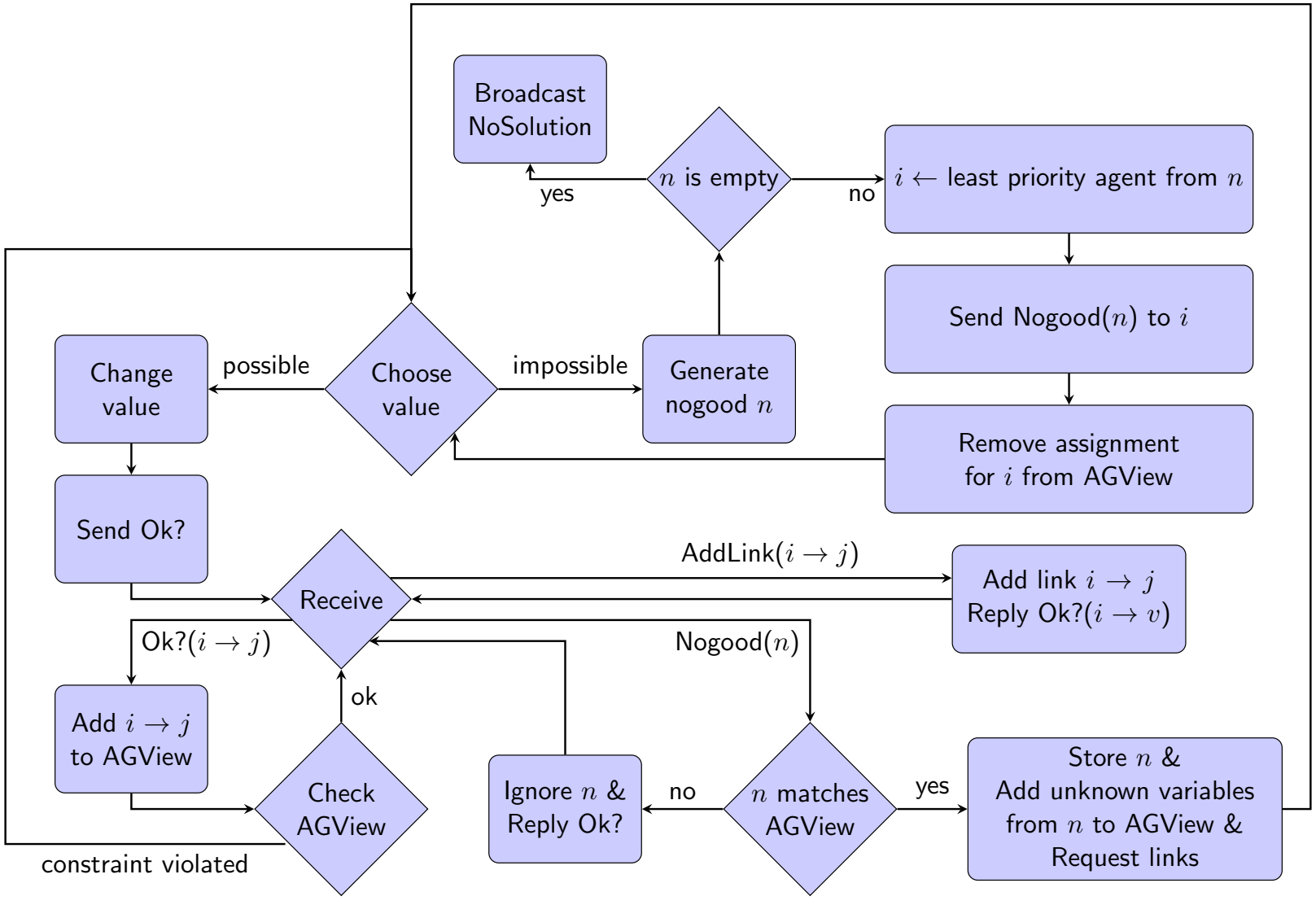
Data structures:

- Agent's current assignment.
- Set of outgoing links (i.e., agents who need to know my assignment).
- Set of incoming links (i.e., agents who will notify me about their assignment).
- **Agent view** — agent's idea about current assignment of other agents.

  $\rightarrow$ May be out of sync!
- **Nogood store** — justification of forbidden values in the domain.

  $\rightarrow$ If Nogood is no longer active (i.e., satisfied in the current context), it is removed, and involved values from the agent's domain become available again.

Asynchronous Backtracking (ABT)



Example: Meeting Scheduling

- John needs to meet during the week with both, Alice and Bob. Their availability is as follows: 1) Alice can meet John on Monday and Thursday; 2) Bob can meet John on Tuesday and Thursday; and, 3) John can meet any of them on Monday, Tuesday and Thursday. Due to time constraints, John can meet them only during the same day.
- Bob doesn't know about Alice's meeting, and vice versa.
- Take the priority ordering, from highest to lowest: 1) Alice; 2) Bob; 3) John.

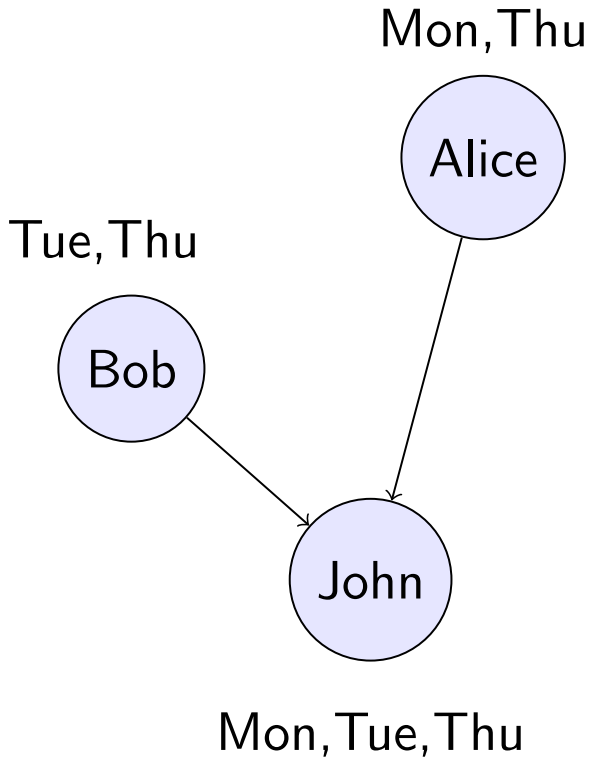
Example: Meeting Scheduling

$\mathcal{A} = \{\text{Alice}, \text{Bob}, \text{John}\}$

$\mathcal{X} = \{x_{\text{Alice}}, x_{\text{Bob}}, x_{\text{John}}\}$
Agent i controls variable x_i.

$\mathcal{D} = \{D_{\text{Alice}}, D_{\text{Bob}}, D_{\text{John}}\}$
$D_{\text{Alice}} = \{\text{Mon}, \text{Thu}\}$
$D_{\text{Bob}} = \{\text{Tue}, \text{Thu}\}$
$D_{\text{John}} = \{\text{Mon}, \text{Tue}, \text{Thu}\}$

$\mathcal{C} = \{x_{\text{Bob}} = x_{\text{John}}, x_{\text{Alice}} = x_{\text{John}}\}$



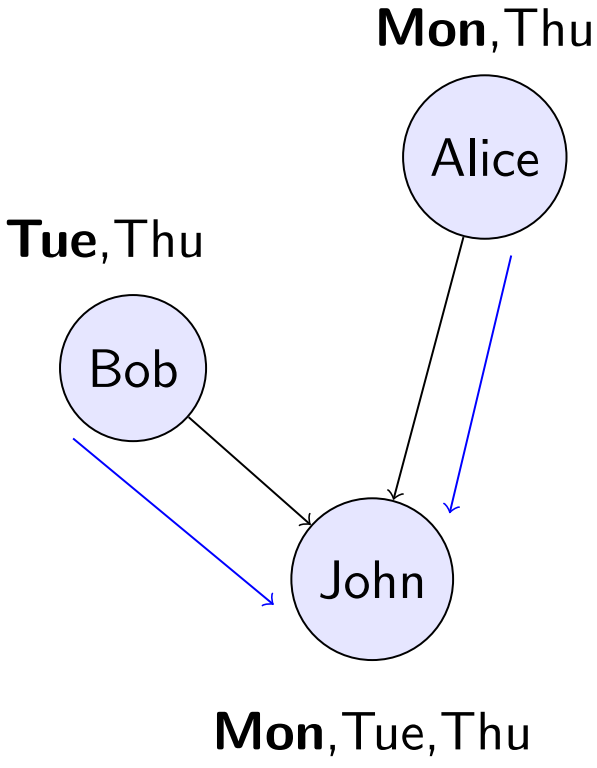
Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\varnothing$ John: $\varnothing$

Let's all propose a date and see what happens!

Bob $\rightarrow$ John:
Ok?(Bob $\rightarrow$ Tue)

Alice $\rightarrow$ John:
Ok?(Alice $\rightarrow$ Mon)



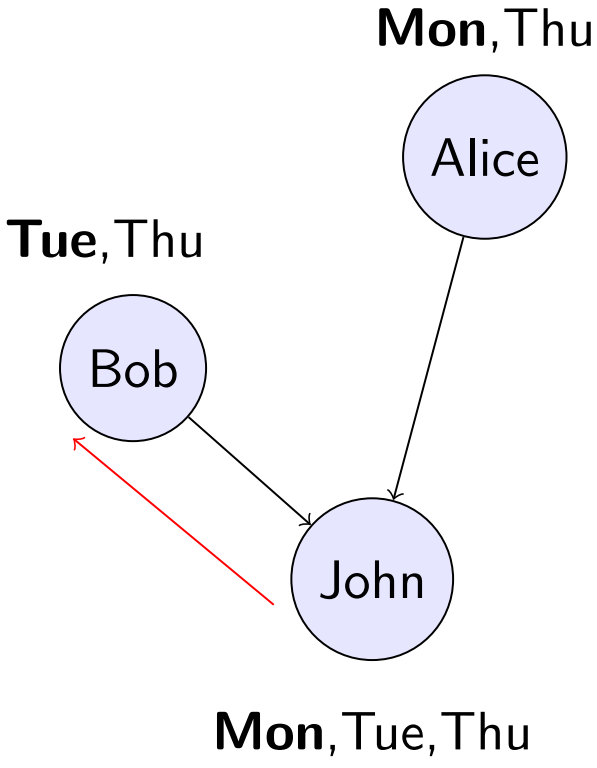
Alice: $\varnothing$ Bob: $\varnothing$ John: $\{\text{Alice} \rightarrow \text{Mon}, \text{Bob} \rightarrow \text{Tue}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\varnothing$ John: $\{\text{Alice} \to \text{Mon}, \text{Bob} \to \text{Tue}\}$

John: Argh, I wanted to have both meetings in one day :-(Let's make them change their minds...

John $\to$ Bob:
Nogood($\{\text{Bob} \to \text{Tue}, \text{Alice} \to \text{Mon}\}$)



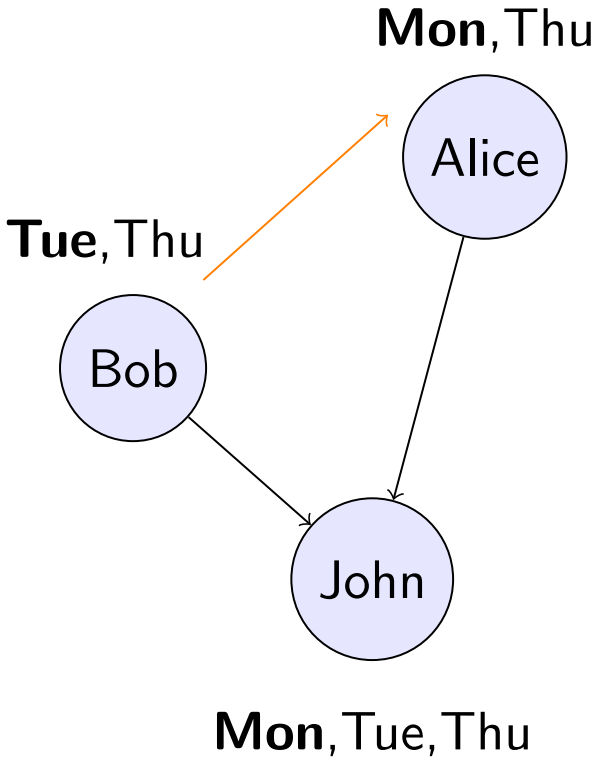
Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}\}$

Bob: Who is that Alice? I've never heard of her.

Bob $\to$ Alice:
AddLink(Alice $\to$ Bob)



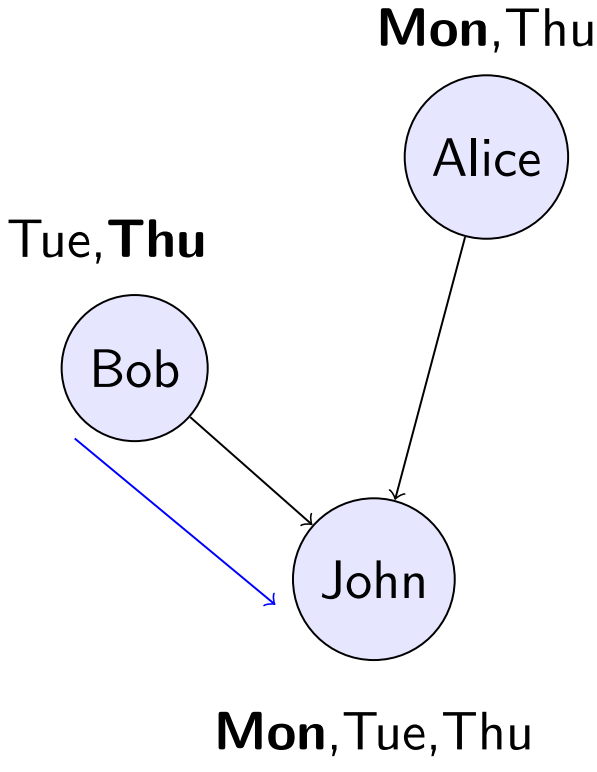
Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}\}$

Bob: John told me that the meeting cannot happen on Tuesday if Alice opts for Monday. Let's try Thursday then...

Bob $\to$ John:
Ok?(Bob $\to$ Thu)



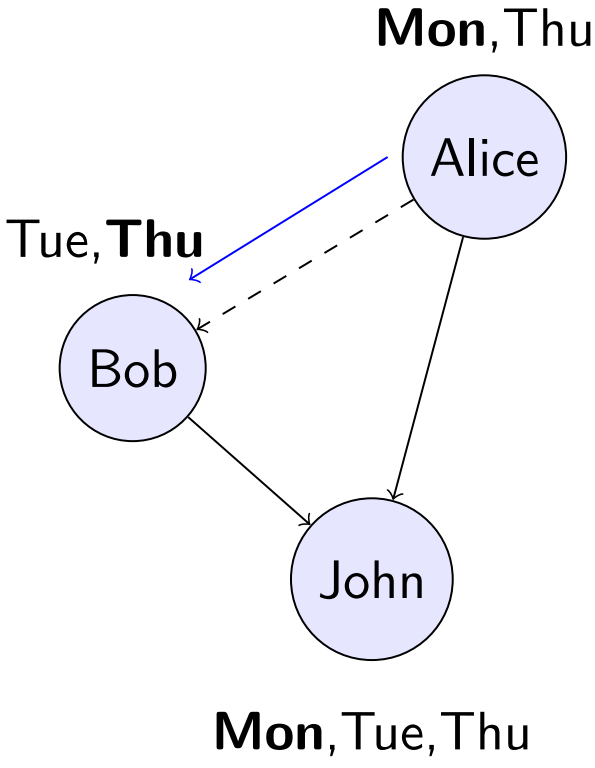
Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}, \text{Bob} \to \text{Thu}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}, \text{Bob} \to \text{Thu}\}$

Alice: Bob, why are you so curious?

Alice $\to$ Bob:
Ok?(Alice $\to$ Mon)



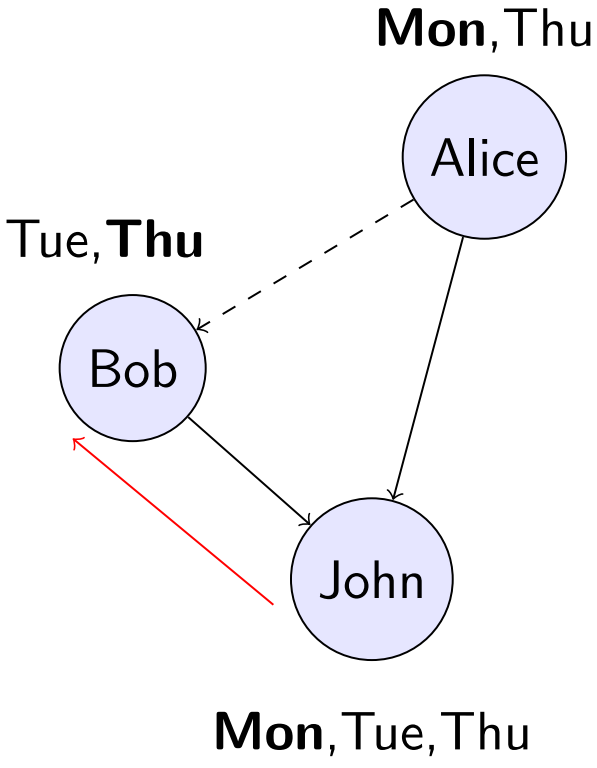
Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}, \text{Bob} \to \text{Thu}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}, \text{Bob} \to \text{Thu}\}$

John: They tried it again. Alright, one more try...

John $\to$ Bob:
Nogood($\{\text{Bob} \to \text{Thu}, \text{Alice} \to \text{Mon}\}$)



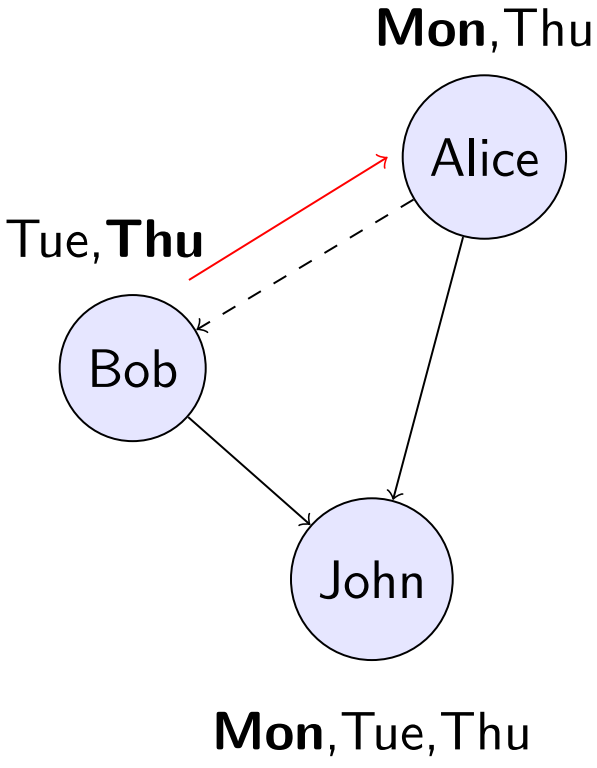
Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Mon}\}$ John: $\{\text{Alice} \to \text{Mon}\}$

Bob: I have run out of options. It's up to Alice now...

Bob $\to$ Alice:
Nogood($\{\text{Alice} \to \text{Mon}\}$)



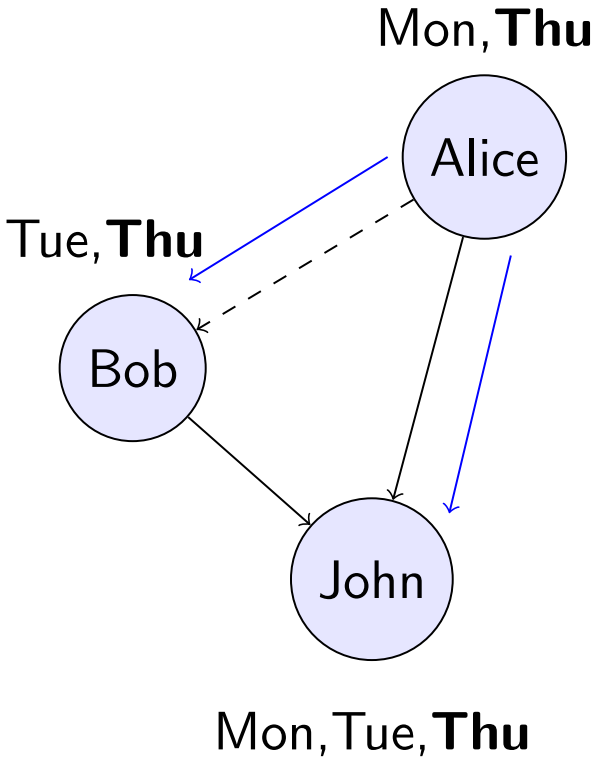
Alice: $\varnothing$ Bob: $\varnothing$ John: $\{\text{Alice} \to \text{Mon}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\varnothing$ John: $\{\text{Alice} \to \text{Mon}\}$

Alice: I have one more option, let's try Thursday.

Alice $\to$ Bob, John:
Ok?($\{\text{Alice} \to \text{Thu}\}$)

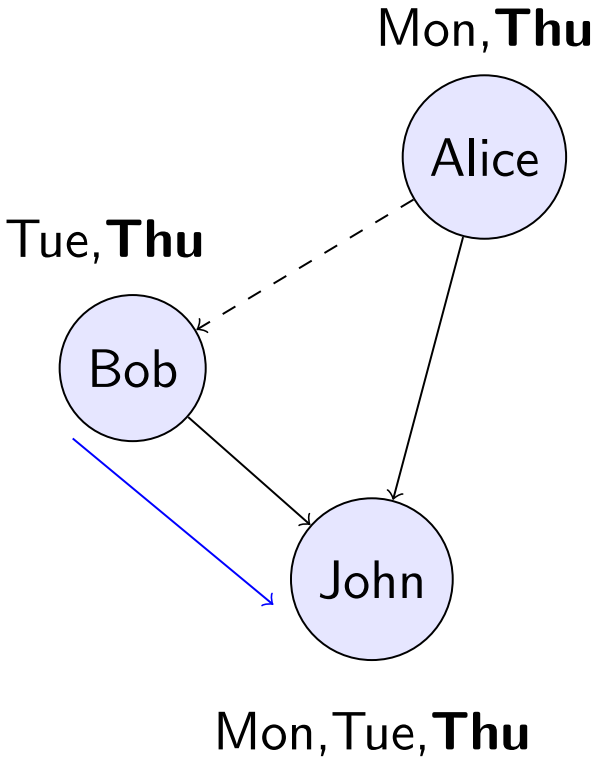


Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Thu}\}$ John: $\{\text{Alice} \to \text{Thu}\}$

Example: Meeting Scheduling

Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Thu}\}$ John: $\{\text{Alice} \to \text{Thu}\}$

John: Finally. Thursday seems like a viable option.



Alice: $\varnothing$ Bob: $\{\text{Alice} \to \text{Thu}\}$ John: $\{\text{Alice} \to \text{Thu}, \text{Bob} \to \text{Thu}\}$

Task: Production Line

From the exercise sheet available in the CourseWare, solve the task:

`Production Line`.