Graphical reasoning for decentralized systems

ADITYA MAHAJAN
YALE UNIVERSITY

Joint work with Sekhar Tatikonda

Acknowlededements: Demos Teneketzis,
Ashutosh Nayyar, and Achilleas Anastasopoulos

Organization

- Model—what are sequential teams?
- Solution concept—team forms and their simplification
- First main idea—ignoring irrelevant data
- Implementing the main idea—directed graphs and graph reductions
- Second main idea—Coordinator for a collection of agents
- Examples along the way—real-time communication, decentralized control

Multi-agent decentralized systems

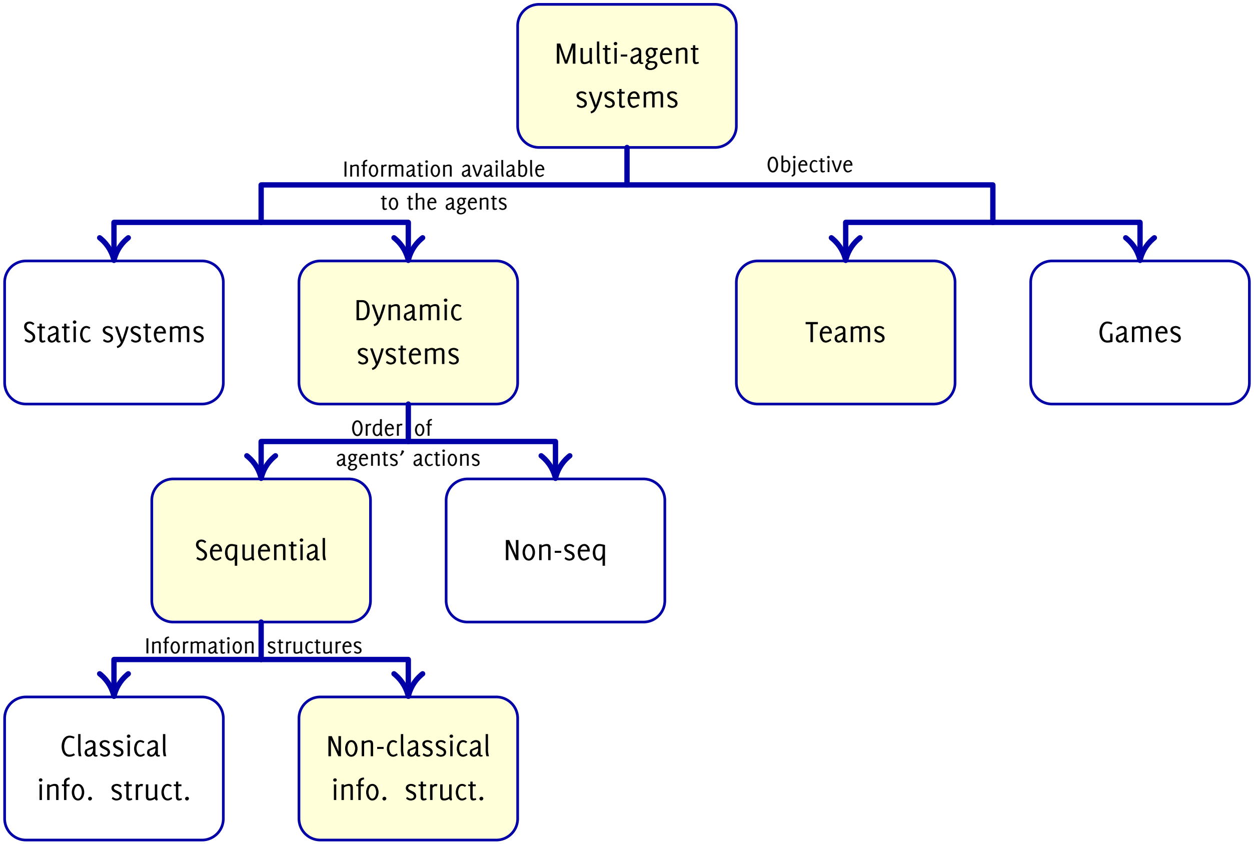
Applications

- telecommunication networks
- sensor networks
- surveillance networks
- transportation networks
- control systems
- monitoring and diagnostic systems
- multi-robot systems
- multi-core CPUs
- . . .

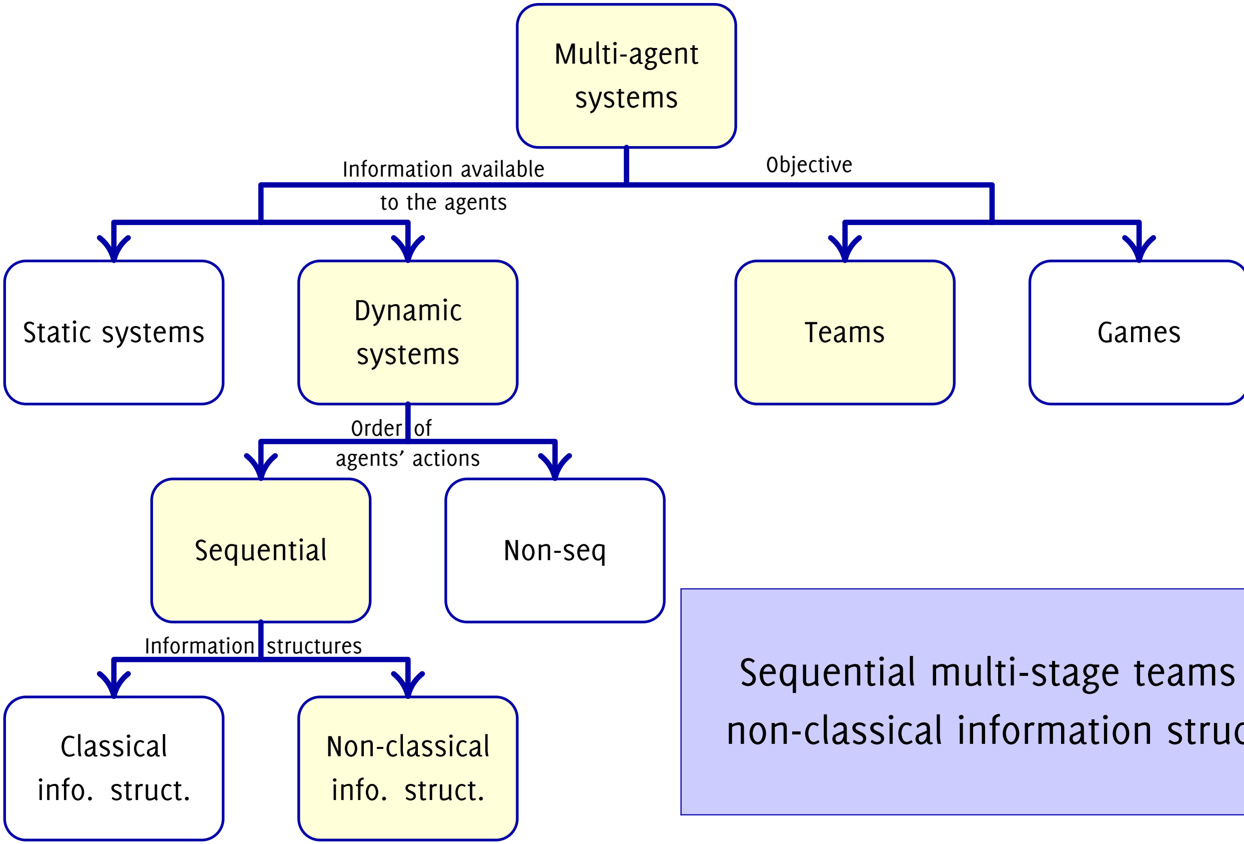
Salient features

- System has different components
- These components know different information
- The components need to cooperate and coordinate

Classification



Classification



Sequential multi-stage teams with non-classical information structures

Model

Notation

For a set M

- Random Variables: $X_M = (X_m : m \in M)$.
- Spaces: $\mathcal{X}_M = \prod_{m \in M} \mathcal{X}_m$
- σ-algebras: $\mathfrak{F}_M = \bigotimes_{m \in M} \mathfrak{F}_m$

Model for a sequential team

- A collection of n system variables, $(X_k, k \in N)$ where $N = \{1, \dots, n\}$
- A collection $\{(\mathcal{X}_k, \mathfrak{F}_k)\}_{k \in N}$ of measurable spaces.
- A set $A \subset N$ of controllers/agents.
  Controller $\alpha \in A$ chooses X_α. Nature chooses X_k, $k \in N \setminus A$.
- A collection $\{I_k\}_{k \in N}$ of information sets such that $I_k \subset \{1, \dots, k-1\}$.
- The variables $X_{N \setminus A}$ are chosen by nature according to stochastic kernels $\{p_k\}_{k \in N \setminus A}$ where p_k is a stochastic kernel from $(\mathcal{X}_{I_k}, \mathfrak{F}_{I_k})$ to $(\mathcal{X}_k, \mathfrak{F}_k)$.
- A set $R \subset N$ of rewards.

Objective

- Choose a strategy $\{g_k\}_{k \in A}$ such that the control law g_k is a measurable function from $(\mathcal{X}_{I_k}, \mathfrak{F}_{I_k})$ to $(\mathcal{X}_k, \mathfrak{F}_k)$.

- Joint measure induced by strategy $\{g_k\}_{k \in N}$

$$P(dX_N) = \bigotimes_{k \in N \setminus A} p_k(dX_k | X_{I_k}) \bigotimes_{k \in A} \delta_{g_k(X_{I_k})}(dX_k)$$

- Choose a strategy to maximize

$$E^{g_A} \left\{ \sum_{i \in R} X_i \right\}$$

This maximum reward is called the value of the team

Information Sets and Information Structures

Information sets are related to information structures.

As a first order approximation, if

for agents k, l such that $k < l$, we have $I_k \subseteq I_l$

system has classical information structures; otherwise it has non-classical information structure.

Generality of the model

This model is a generalization of the model presented in

Hans S. Witsenhausen, Equivalent stochastic control problems, Math. Cont. Sig. Sys.-88

which in turn in equivalent to the intrinsic model (specialized to sequential teams) presented in

Hans S. Witsenhausen, On information structures, feedback and causality, SICON-71

which is as general as it gets.

Solution concept

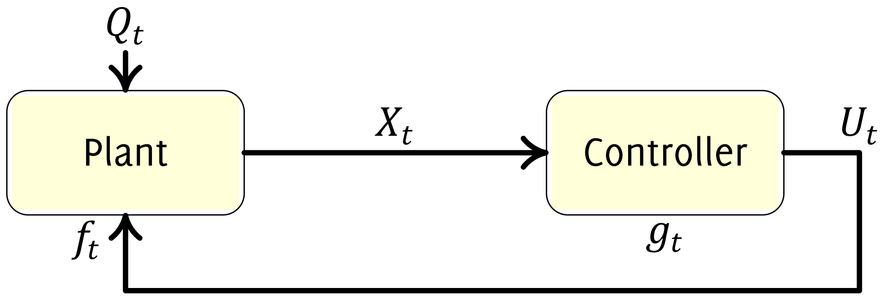
- Structural results

  Can we restrict attention to a subset of control laws without loosing in optimality?
  Examples: Markov policies in MDPs, linear policies in LQG systems, threshold policies in detection, etc.

- Sequential decomposition

  Can we pick the control laws one by one, instead of choosing them all at once.
  Example: Dynamic programming

The foundation of centralized systems: MDP (Markov decision process)

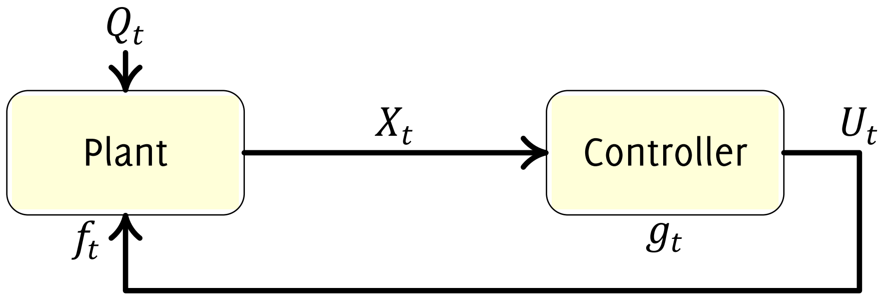


Plant: $X_{t+1} = f_t(X_t, U_t, Q_t)$

Controller: $U_t = g_t(X^t, U^{t-1})$

Minimize: $\mathbb{E}^g \Big\{ \sum_t c(X_t, U_t) \Big\}$

The foundation of centralized systems: MDP (Markov decision process)



Plant: $X_{t+1} = f_t(X_t, U_t, Q_t)$

Controller: $U_t = g_t(X^t, U^{t-1})$

Minimize: $\mathbb{E}^g \left\{ \sum_t c(X_t, U_t) \right\}$

Without loss of optimality, we can restrict attention to

Markov policy: $U_t = g_t(X_t)$

Can we obtain similar results for decentralized systems?

Team form

A (sequential) team form is the team problem where the measurable spaces $\{(\mathcal{X}_k, \mathfrak{F}_k)\}_{k \in N}$ and the stochastic kernels $\{p_k\}_{k \in N \setminus A}$ are not pre-specified.

$\mathcal{T} = (N, A, R, \{I_k\}_{k \in N})$: system variables, control variables, reward variables, and the information sets are specified.

Equivalence of team forms

Two team forms $\mathcal{T} = (N, A, R, \{I_k\}_{k \in N})$ and $\mathcal{T}' = (N', A', R', \{I'_k\}_{k \in N'})$ are equivalent if the following conditions hold:

1. $N = N'$, $A = A'$, and $R = R'$;
2. for all $k \in N \setminus A$, we have $I_k = I'_k$;
3. for any choice of measurable spaces $\{(\mathcal{X}_k, \mathfrak{F}_k)\}_{k \in N}$ and stochastic kernels $\{p_k\}_{k \in N \setminus A}$, the values of the teams corresponding to $\mathcal{T}$ and $\mathcal{T}'$ are the same.

The first two conditions can be verified trivially. There is no easy way to check the last condition.

Simplification of team forms

A team form $\mathcal{T}' = (N', A', R', \{I'_k\}_{k \in N'})$ is a simplification of a team form $\mathcal{T} = (N, A, R, \{I_k\}_{k \in N})$ if

$$\mathcal{T}' \text{ is equivalent to } \mathcal{T}$$

and
$$\sum_{k \in A} |I'_k| < \sum_{k \in A} |I_k|.$$

$\mathcal{T}'$ is a strict simplification of $\mathcal{T}$ if $\mathcal{T}'$ is equivalent to $\mathcal{T}$, $|I'_k| \leq |I_k|$ for $k \in N$, and at least one of these inequalities is strict.

Given a team form,
can we simplify it?

Can we extend the reasoning of MDPs to decentralized systems

For MDPs, if an agent knows the current state it can ignore other data. But, what is the right notion of state in a decentralized system?

- State from whose perspective? In a centralized system, all agents view the world consistently. In a decentralized system, different agents see the world differently.

- State for what? (input-output mapping, choosing control actions, optimization). In an MDP, all these notions of the state coincide. In a general decentralized system, they are different.

Maybe the proof of MDP gives some intuition

The textbook proof

Define: $V_t(x_1,\dots,x_t) = \min_{\text{all policies}} \mathbb{E}^g \Big\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x^{t} \Big\}$

Define: $W_t(x_t) = \min_{\text{Markov policies}} \mathbb{E}^g \Big\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x_{t} \Big\}$

By definition: $W_t(x_t) \ge V_t(x_1,\dots,x_t)$ for any $x_1,\dots,x_t$.

Recursively prove: $W_t(x_t) \le V_t(x_t,\dots,x_t)$ for any $x_1,\dots,x_t$.

$W_t(x_t) = V_t(x_1,\dots,x_t)$ for all $x_1,\dots,x_t$

The textbook proof with no intuition

Define: $V_t(x_1, \dots, x_t) = \min_{\text{all policies}} \mathbb{E}^g \left\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x^t \right\}$

Define: $W_t(x_t) = \min_{\text{Markov policies}} \mathbb{E}^g \left\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x_t \right\}$

By definition: $W_t(x_t) \ge V_t(x_1, \dots, x_t)$ for any $x_1, \dots, x_t$.

Recursively prove: $W_t(x_t) \le V_t(x_t, \dots, x_t)$ for any $x_1, \dots, x_t$.

$W_t(x_t) = V_t(x_1, \dots, x_t)$ for all $x_1, \dots, x_t$

Is there a cleaner proof which gives some intuition?

An appendix in an obscure paper with the intuition

Hans S. Witsenhausen, On the structure of real-time source coders, BSTJ-79

Suppose we have to minimize cost from the p.o.v. of one agent

$$\mathbb{E}\{\text{cost} \mid \underbrace{\text{relevant data}}_{Y}, \underbrace{\text{irrelevant data}}_{Z}, \underbrace{\text{control action}}_{U=g(Y,Z)}\}$$

$$= \mathbb{E}\{\text{cost} \mid \underbrace{\text{relevant data}}_{Y}, \underbrace{\text{control action}}_{U=g(Y,Z)}\}$$

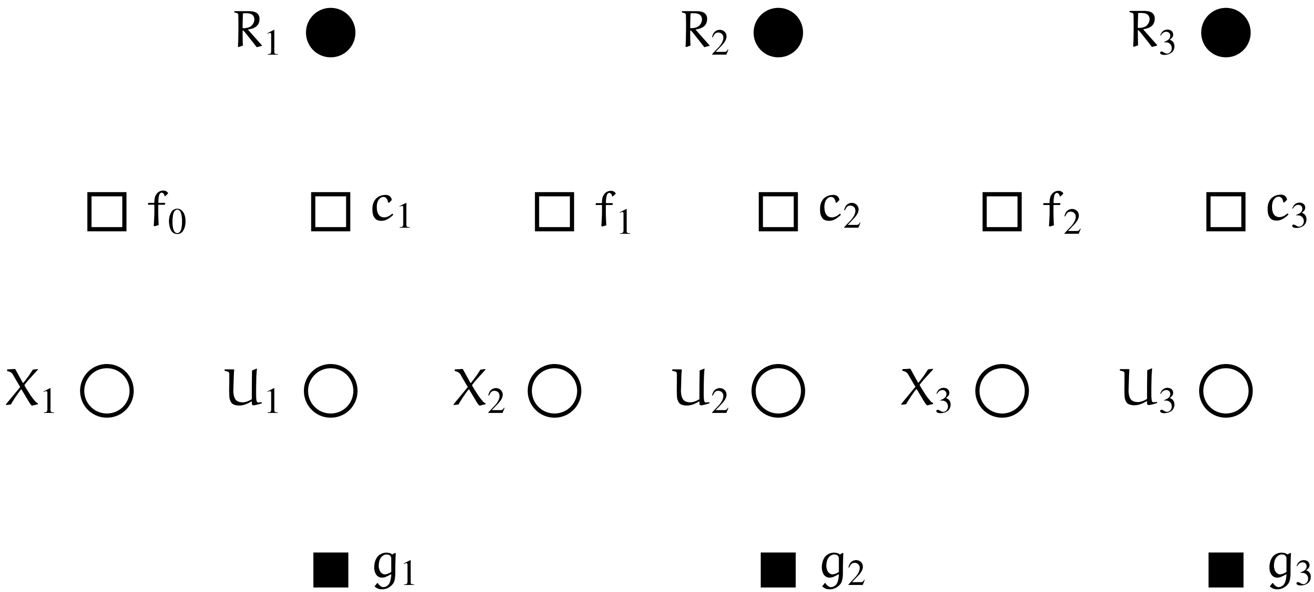
$$= F(Y, g(Y,Z))$$

For any g, there exists a $\hat{g}$ such that for all y and z, $F(y, \hat{g}(y)) \leq F(y, g(y,z))$

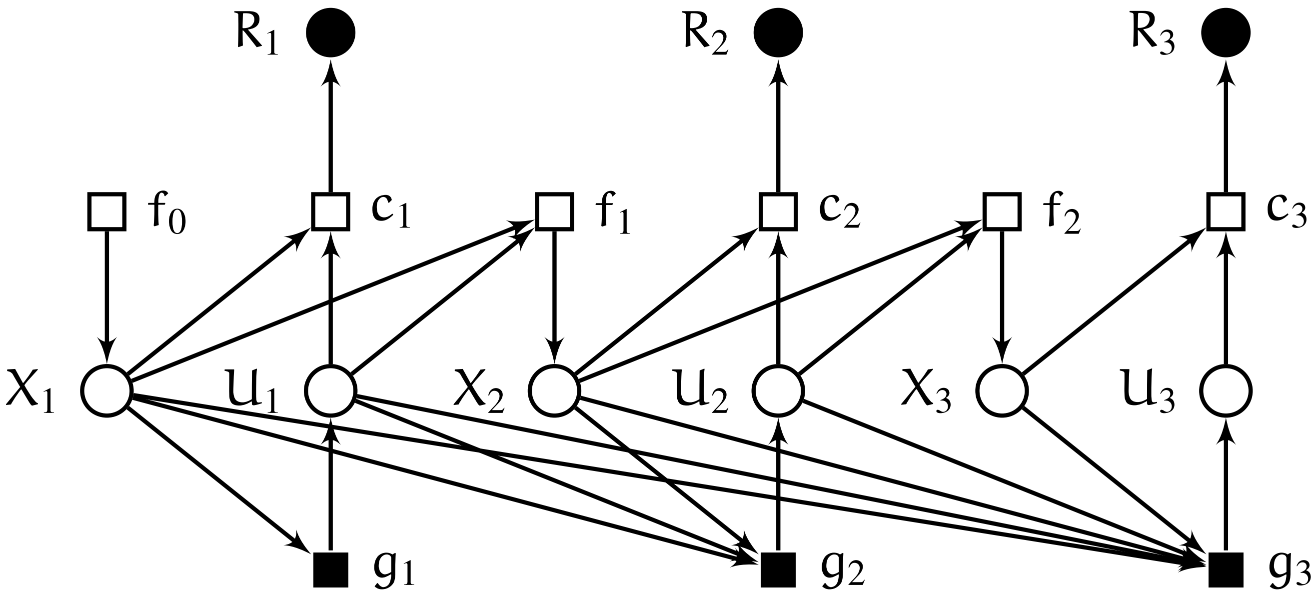
Without loss of optimality, choose $U = g(Y)$.

Rest is just a matter of detail. Find irrelevant data, repeat for all time steps.

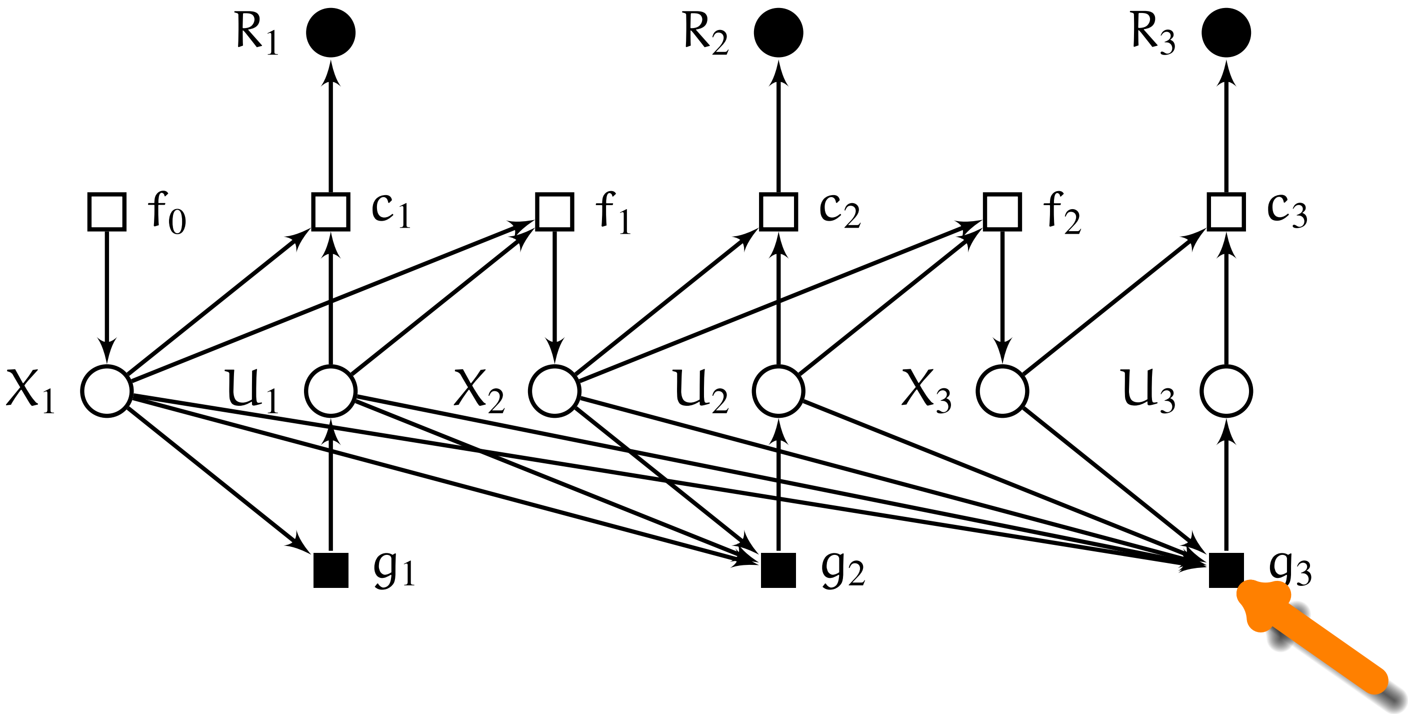
The proof with the intuition



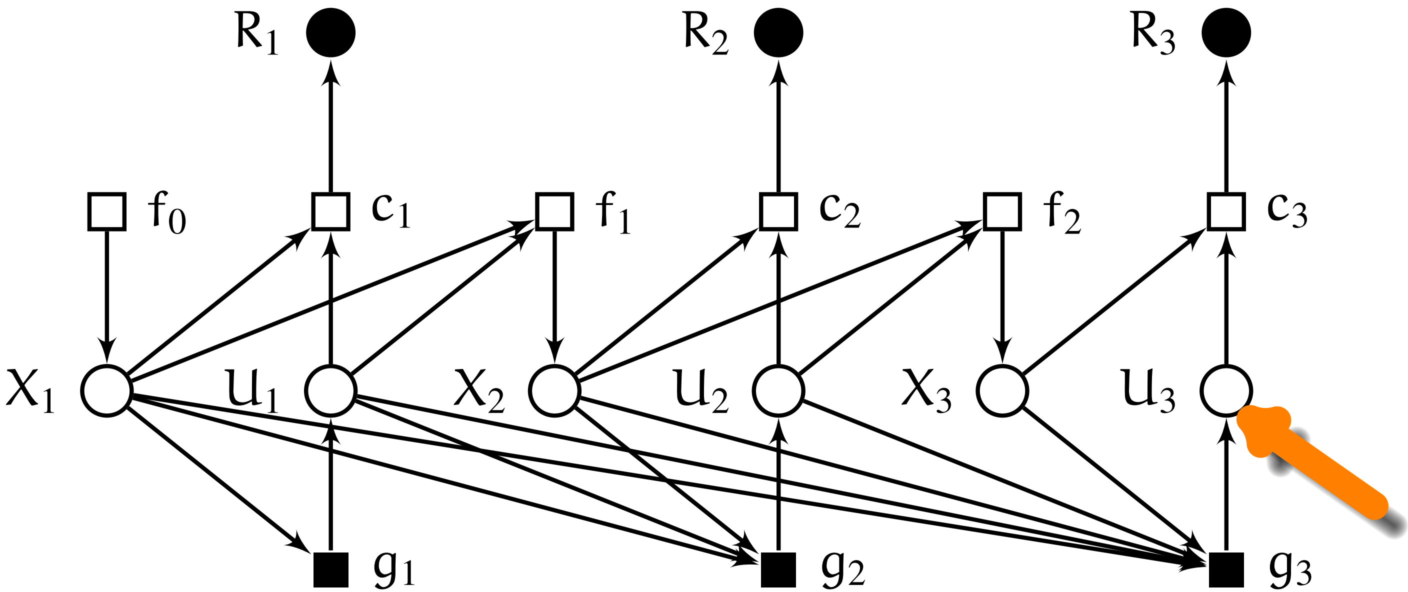
The proof with the intuition



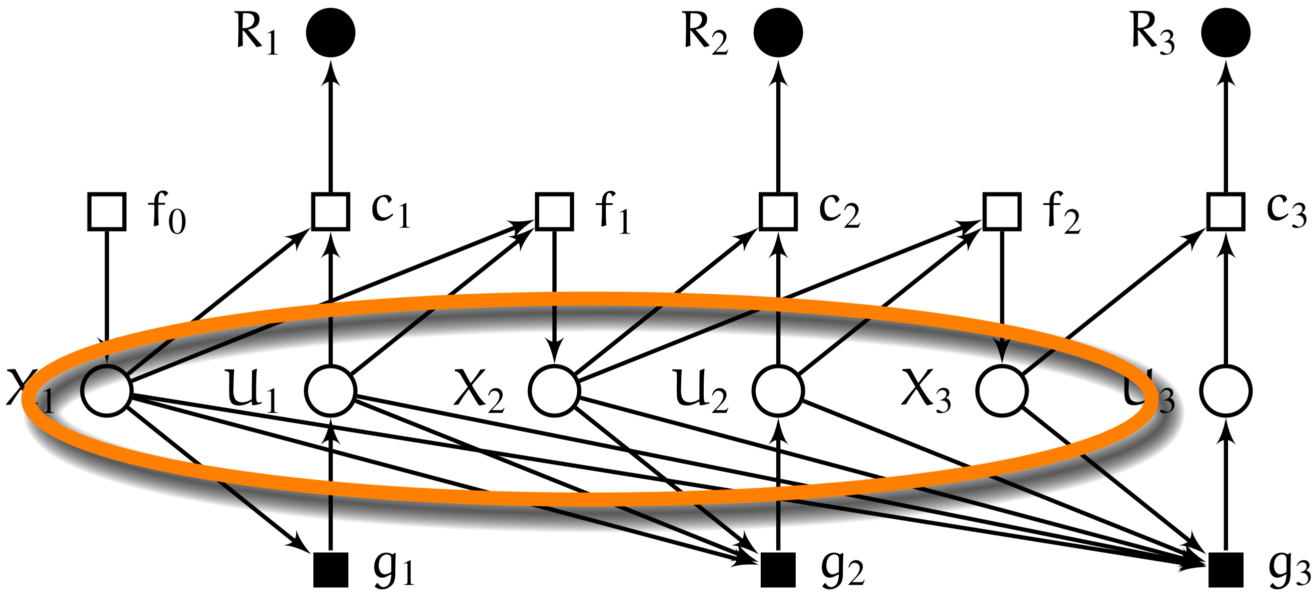
The proof with the intuition: agent at time 3



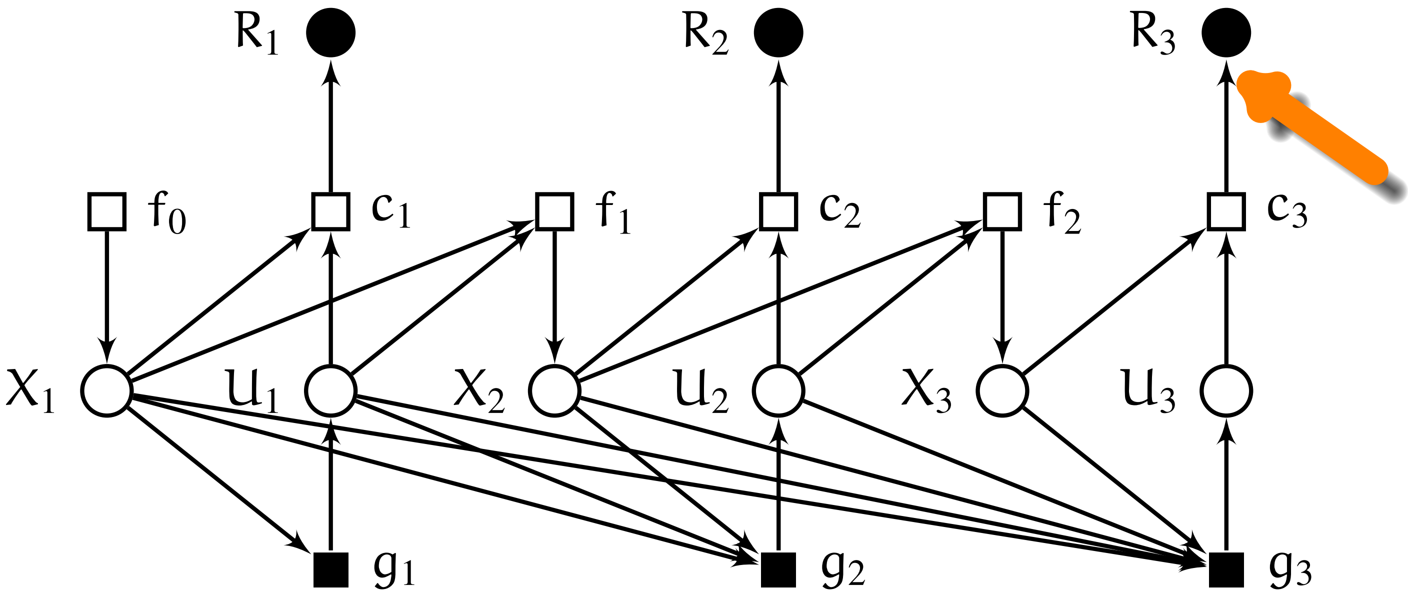
The proof with the intuition: control action



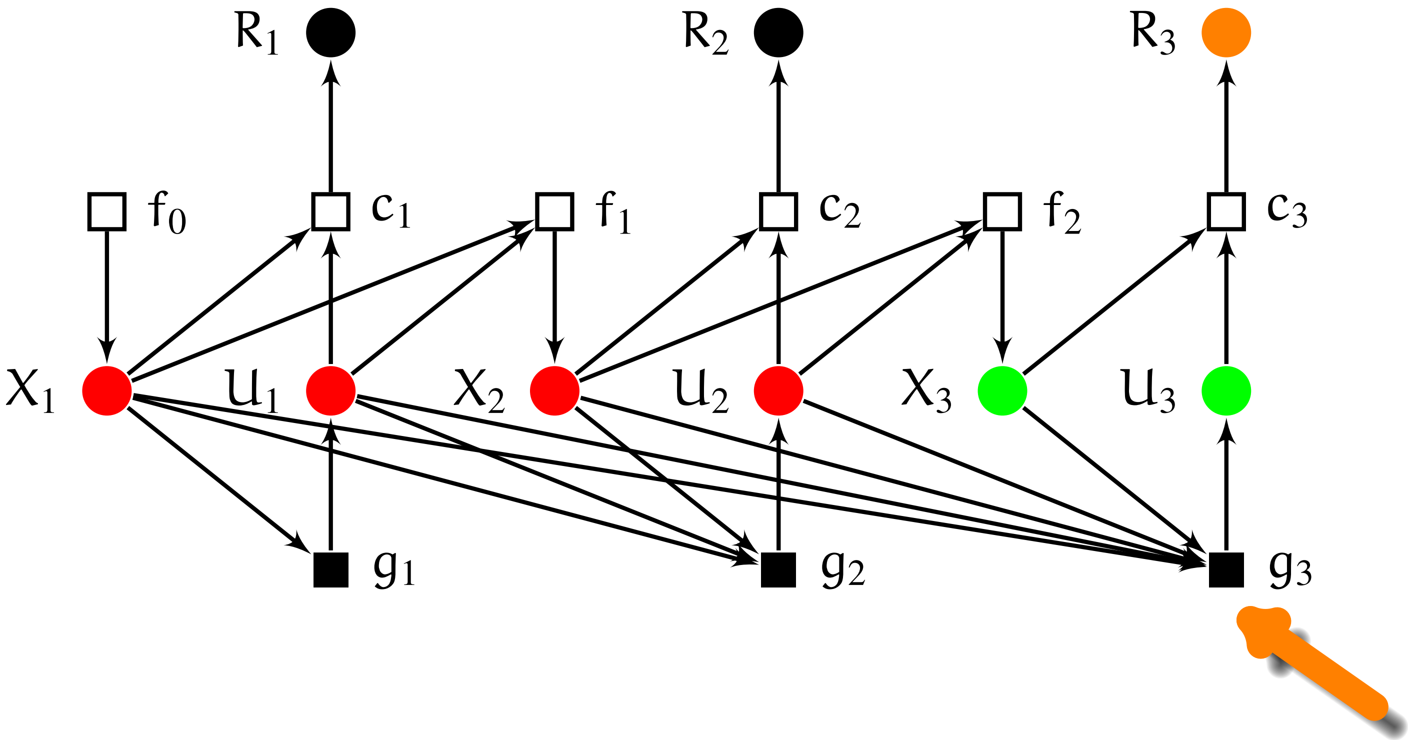
The proof with the intuition: observations



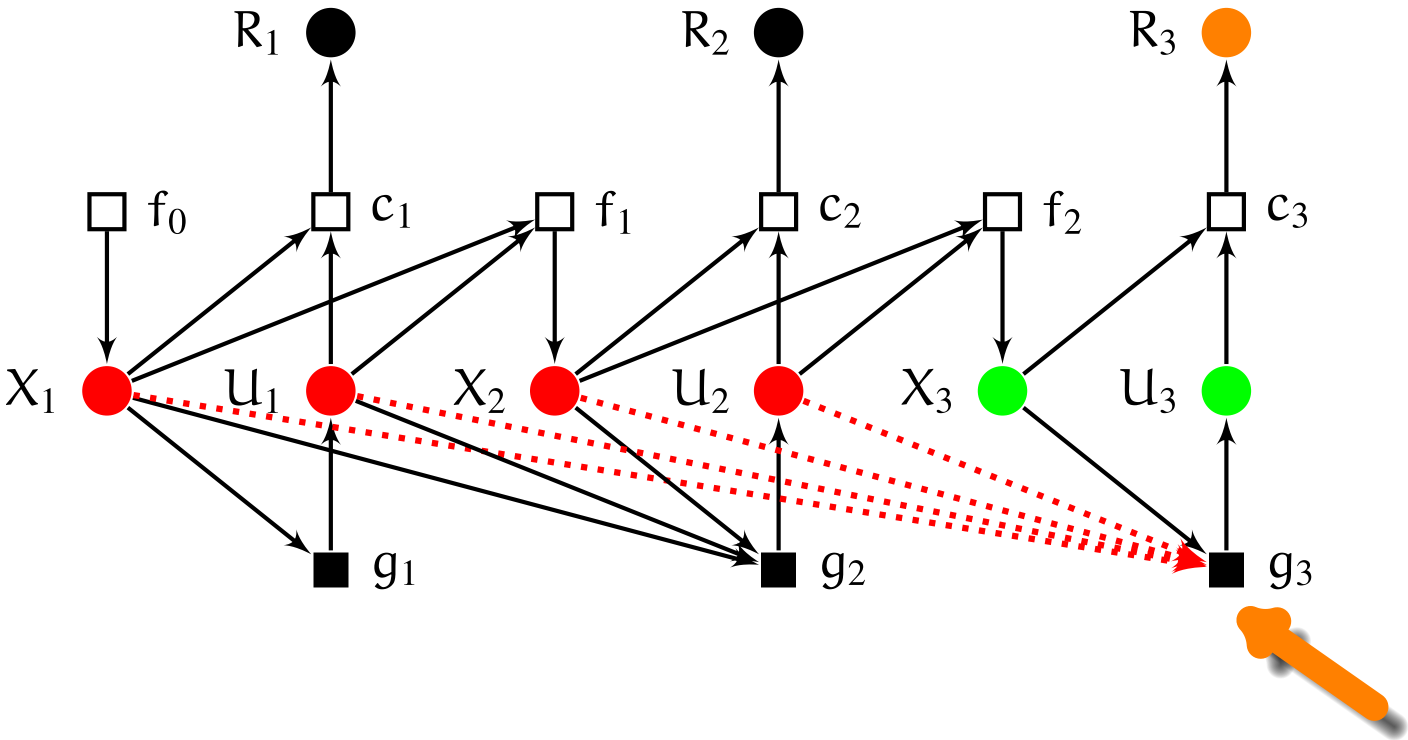
The proof with the intuition: dependent reward



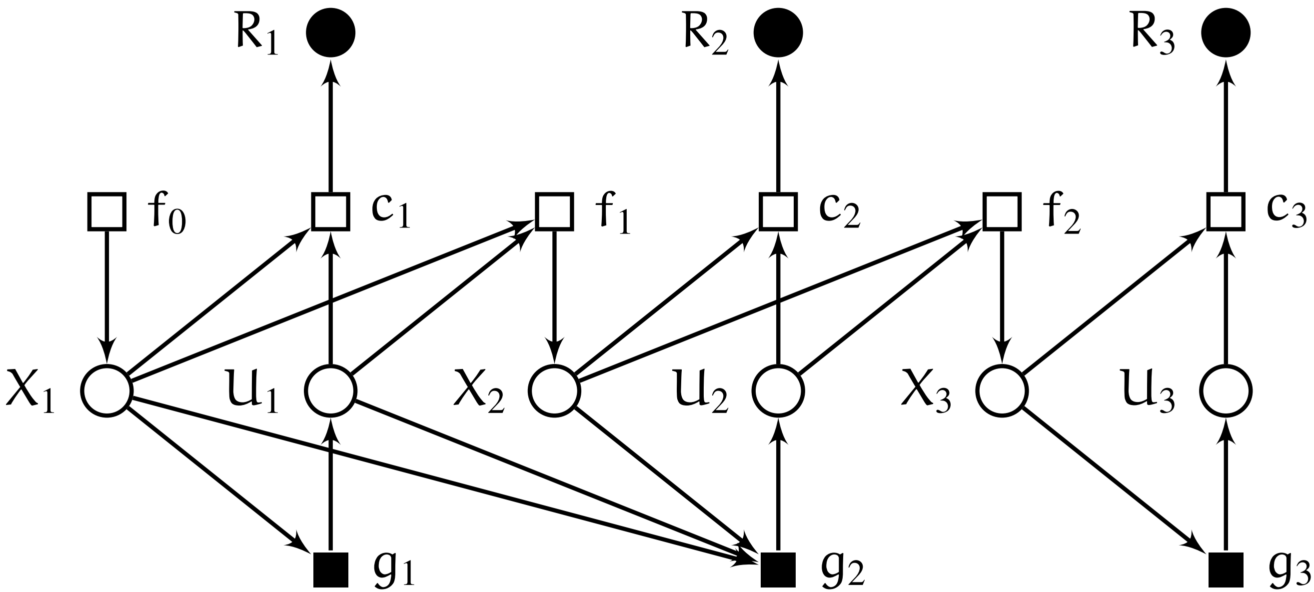
The proof with the intuition: irrelevant observations



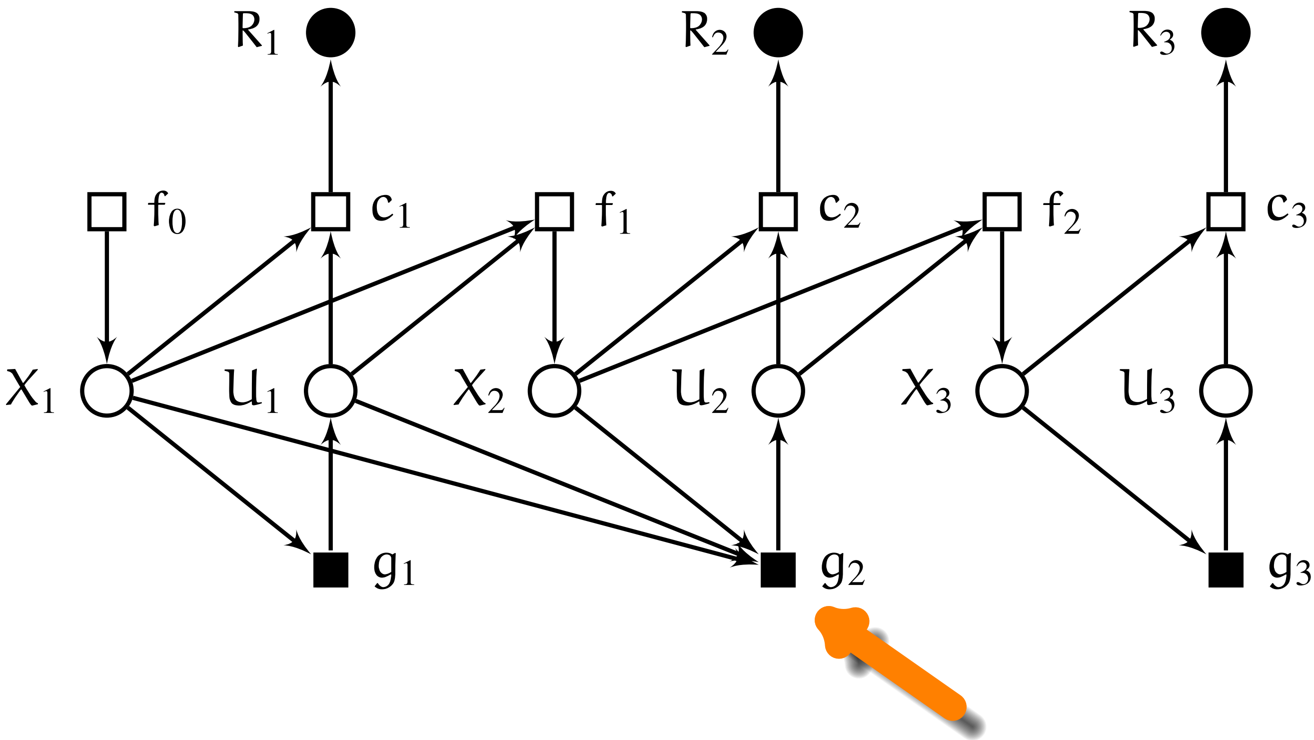
The proof with the intuition: remove edges



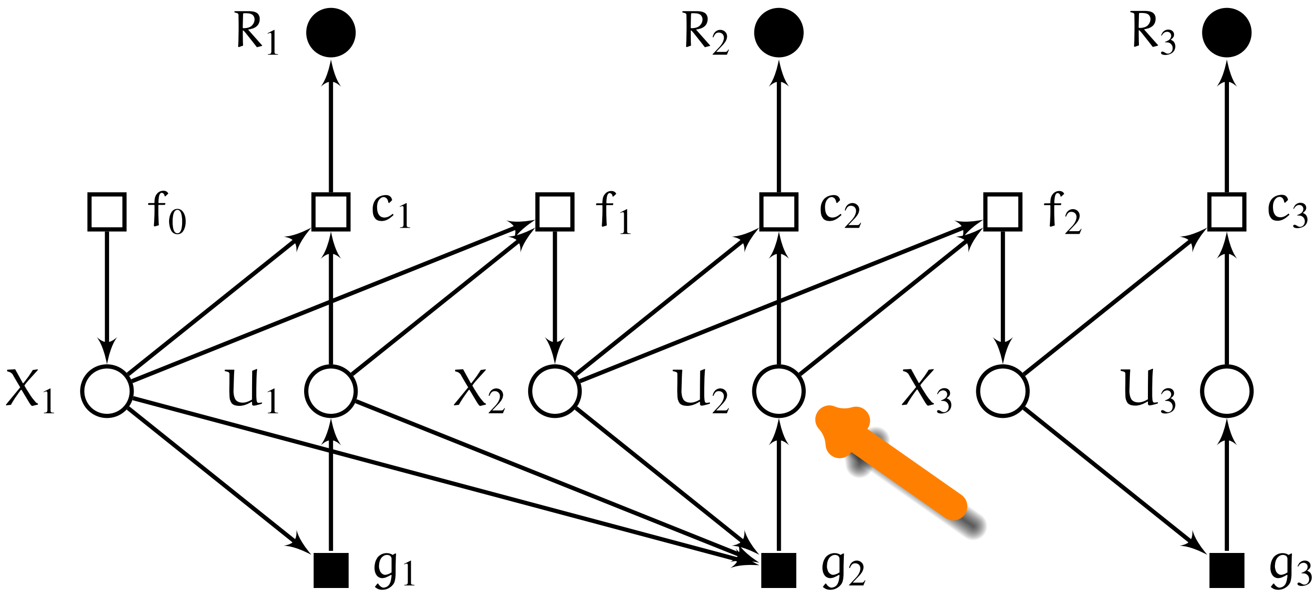
The proof with the intuition: ***repeat***



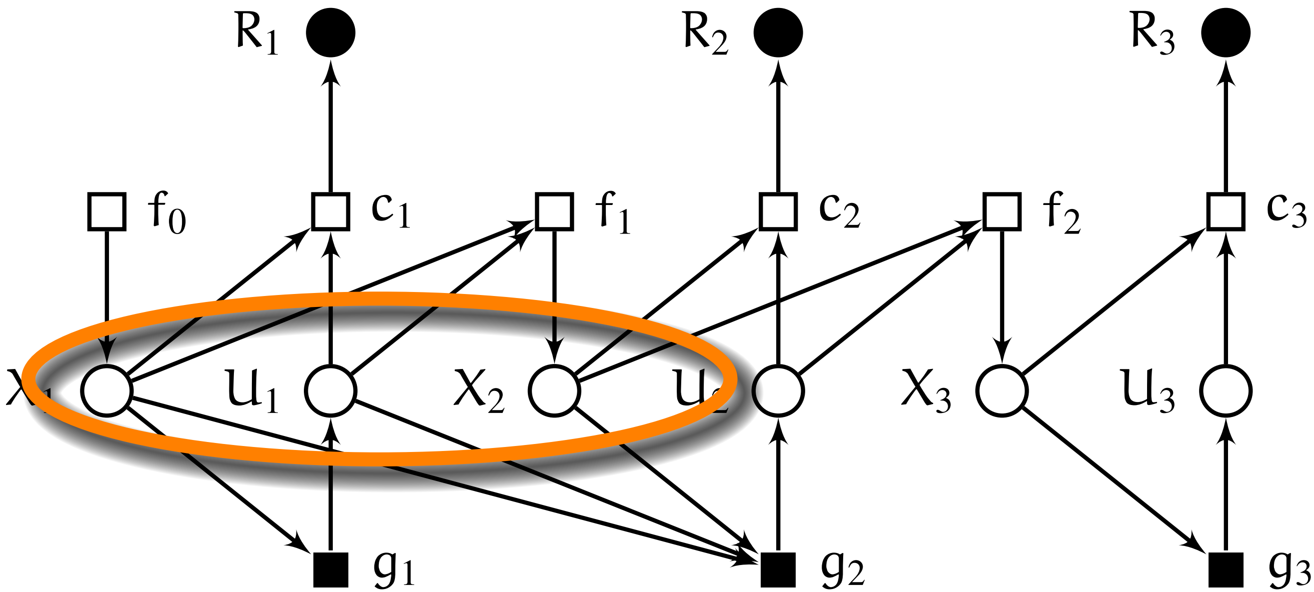
The proof with the intuition: agent at time 2



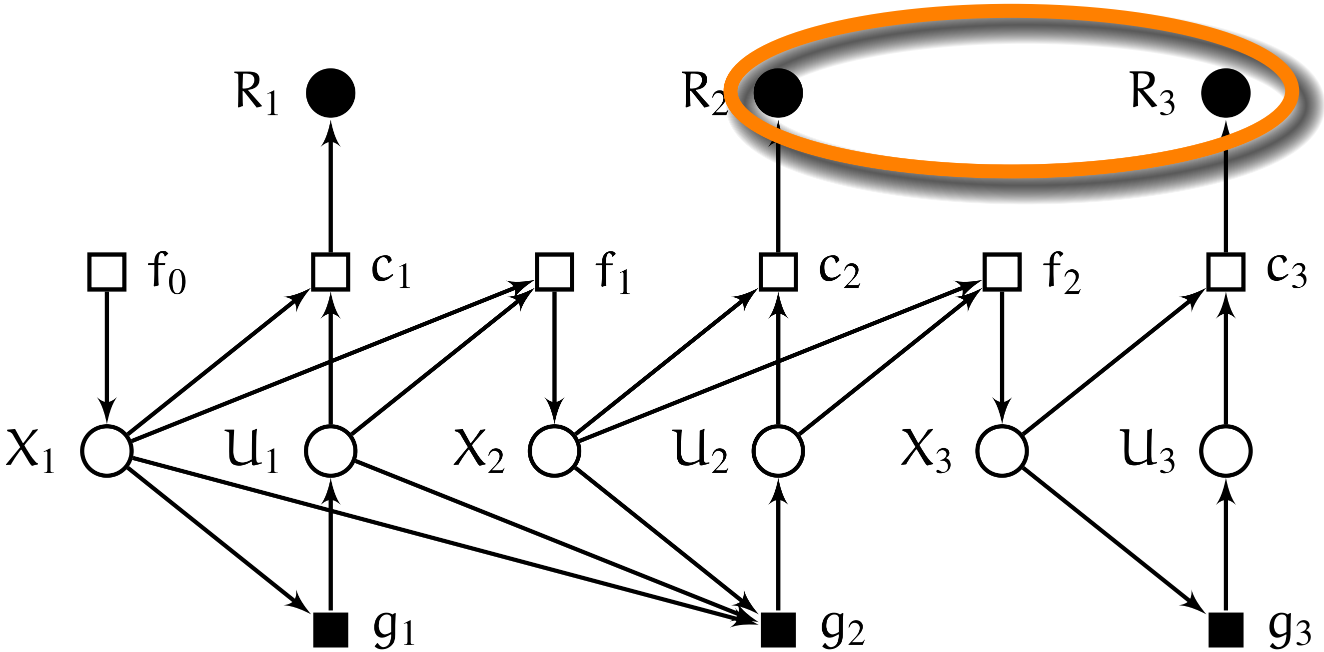
The proof with the intuition: control action



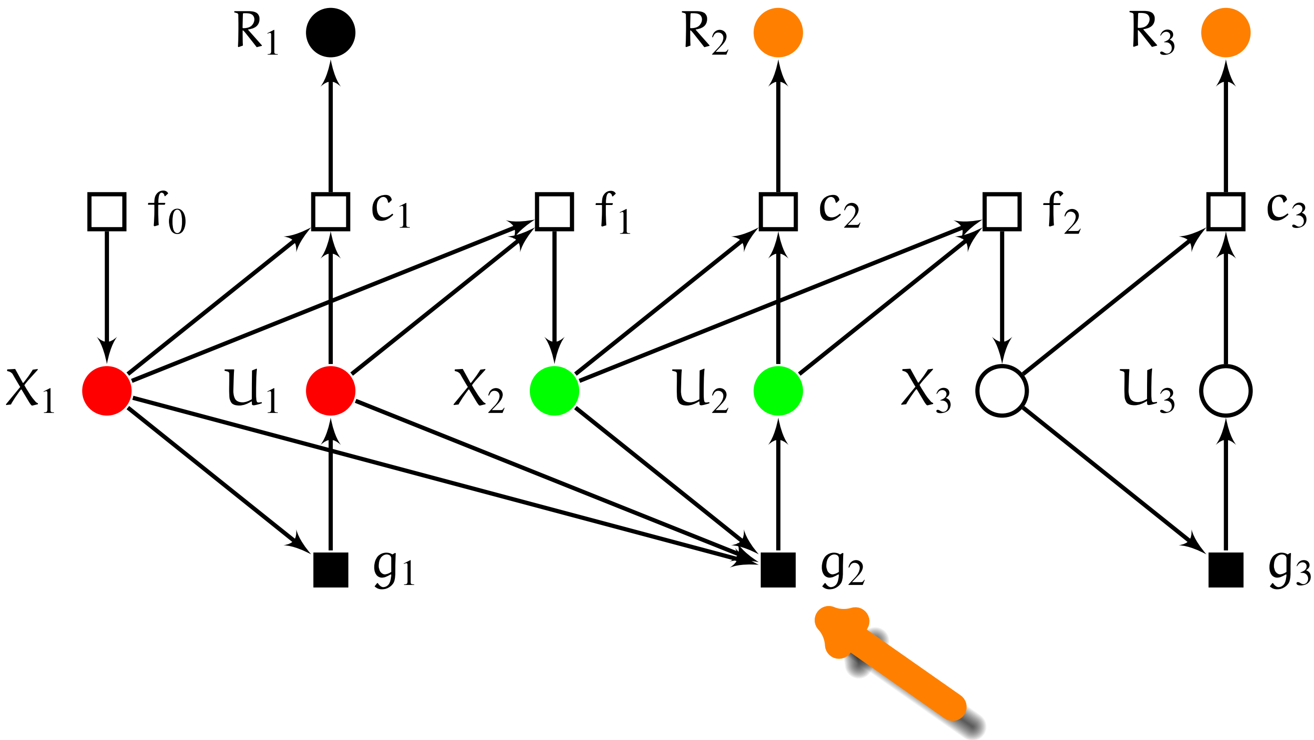
The proof with the intuition: observations



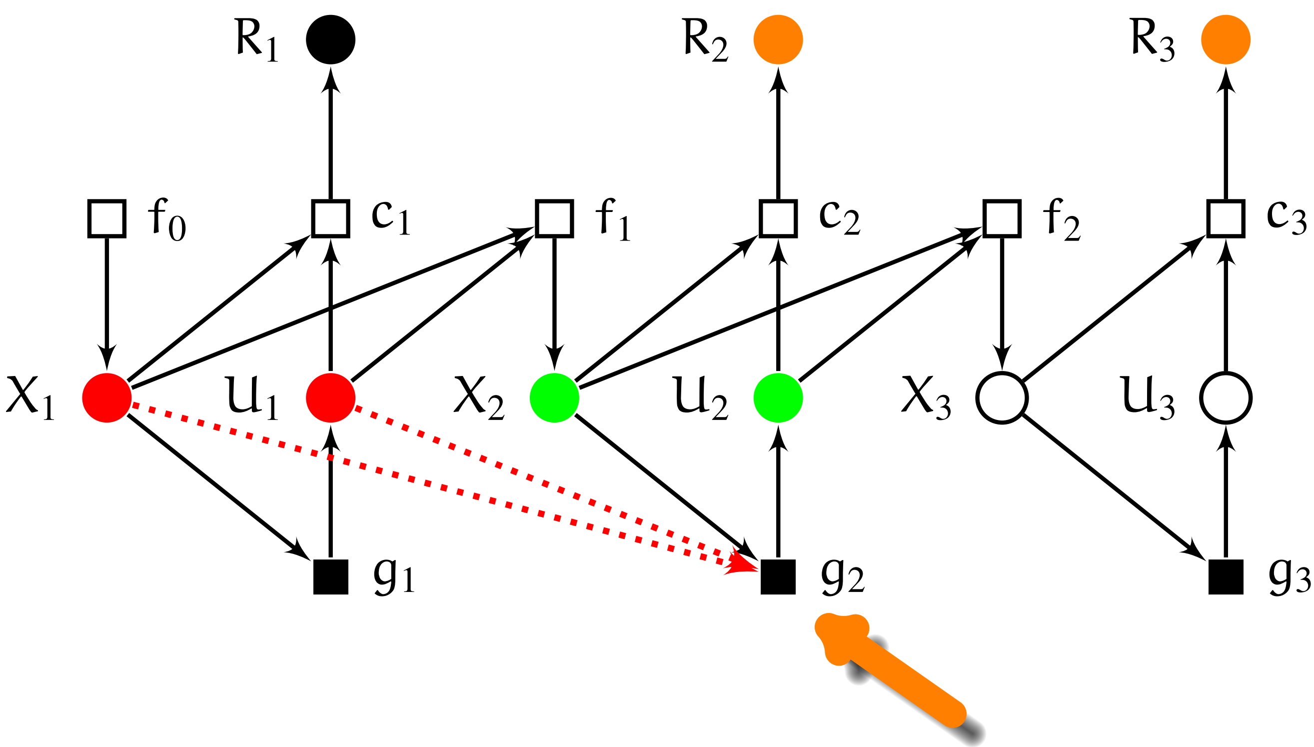
The proof with the intuition: dependent rewards



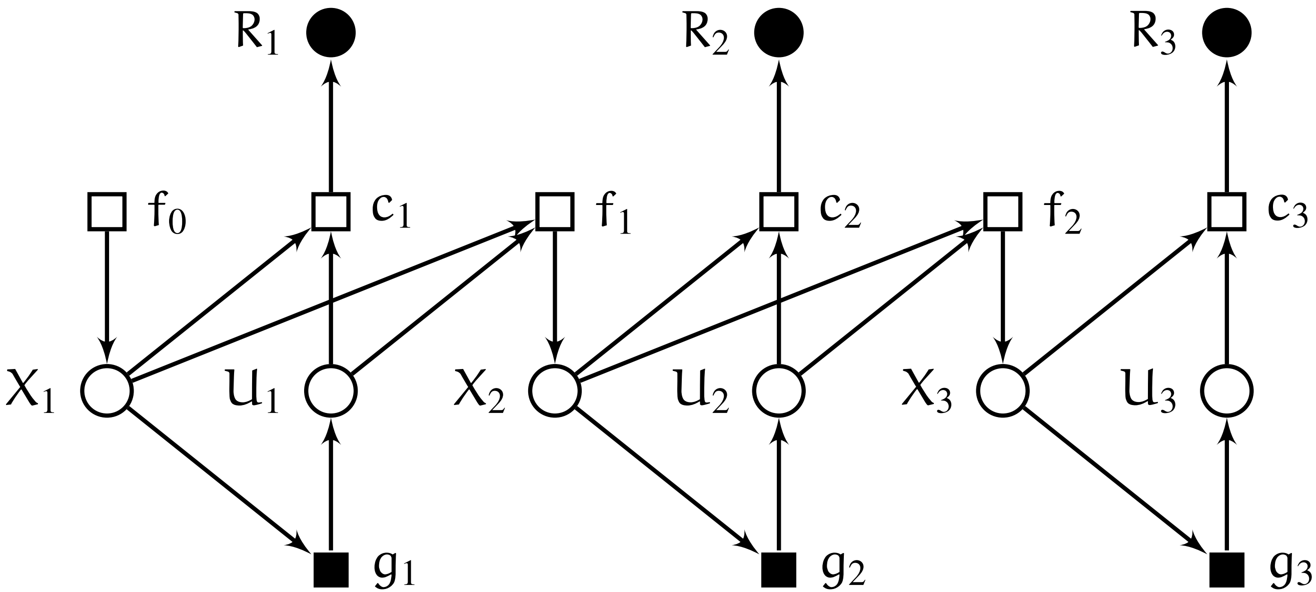
The proof with the intuition: irrelevant observations



The proof with the intuition: remove edges



The proof with the intuition: we are done



The main idea

Step 1: Pick an agent

Step 2: If the agent observes any irrelevant data, ignore those observations

Step 3: Repeat

The main idea

Step 1: Pick an agent

Step 2: If the agent observes any irrelevant data, ignore those observations

Step 3: Repeat

This idea is easy to extend to decentralized systems. We only need to work out the details.

Extending the idea to decentralized systems

To follow the above process in decentralized systems, we have to do two things:

- What is the order in which the agents act?
- What is right notion of irrelevant data? How do find irrelevant observations of an agent

Both questions can be answered using graphical models

Some Preliminaries

Partial Orders

A strict partial order $\prec$ on a set S is a binary relation that is transitive, irreflexive, and asymmetric. i.e., for a, b, c in S, we have

1. if $a \prec b$ and $b \prec c$, then $a \prec c$ (transitive)
2. $a \nprec a$ (irreflexive)
3. if $a \prec b$ then $b \nprec a$ (asymmetric)

The reflexive closure $\preccurlyeq$ of a partial order $\prec$ is given by

$$a \preccurlyeq b \text{ if and only if } a \prec b \text{ or } a = b$$

Partial Order

Let A be a subset of a partially ordered set $(S, \prec)$. Then, the lower set of A, denoted by $\overleftarrow{A}$ is defined as

$$\overleftarrow{A} := \{b \in S : b \preccurlyeq a \text{ for some } a \in A\}.$$

By duality, the upper set of A, denoted by $\overrightarrow{A}$ is defined as

$$\overrightarrow{A} := \{b \in S : a \preccurlyeq b \text{ for some } a \in A\}.$$

Sequential teams and partial orders

Hans S. Witsenhausen, On information structures, feedback and causality, SICON-71

Hans S. Witsenhausen, The intrinsic model for discrete stochastic control: Some open problems, LNEMS-75

A team problem is sequential if and only
if there is a partial order between the agents

Partial orders can
be represented by
directed graphs

So, sequential teams can be
represented as directed graphs

Representing teams using directed graphs

Hans S. Witsenhausen, Separation of estimation and control for discrete time systems, Proc. IEEE-71.

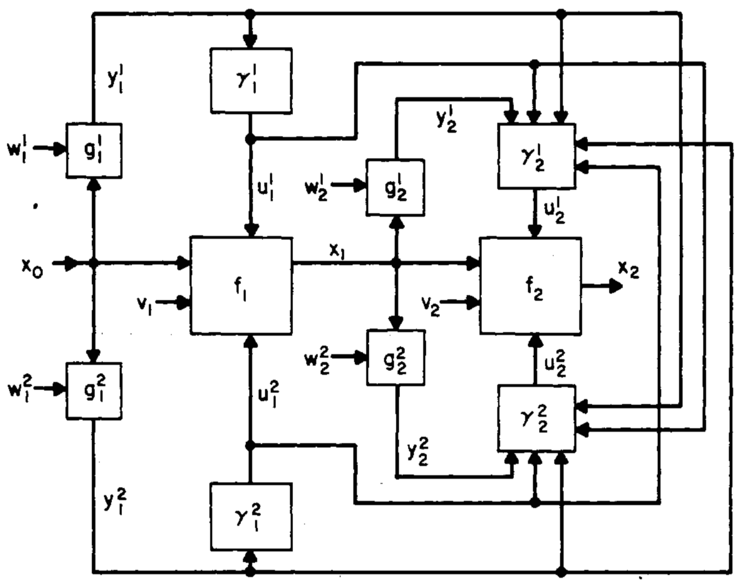


Fig. 1.

Representing teams using directed graphs

Yu-Chi Ho and K'ai-Ching Chu, Team Decision Theory and Information Structures in Optimal Control Problems—Part I, TAC-72.

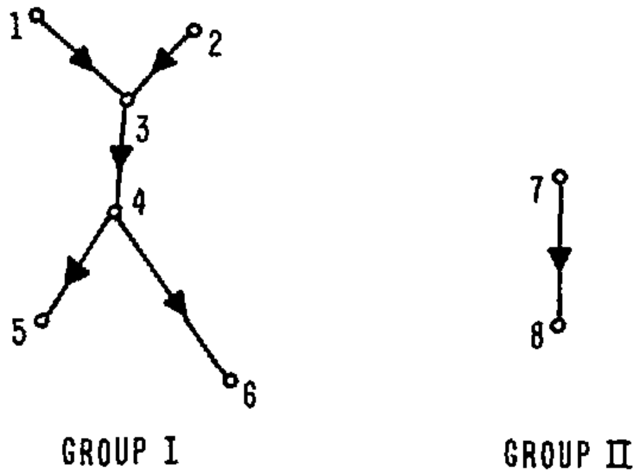


Fig. 3.

Representing teams using directed graphs

Tseneo Yoshikawa, Decomposition of Dynamic Team Decision Problems, TAC-78.

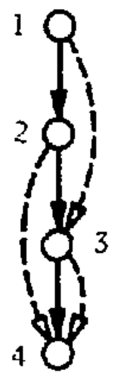
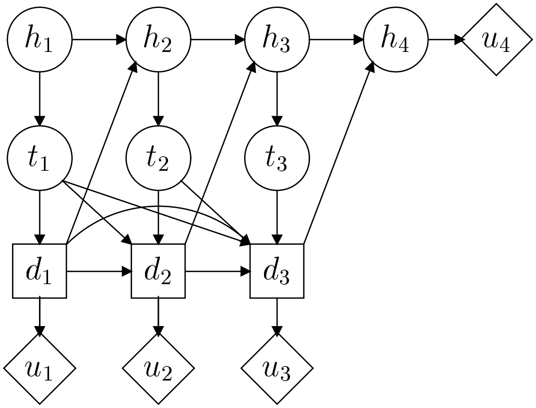


Fig. 1. Precedence diagram.

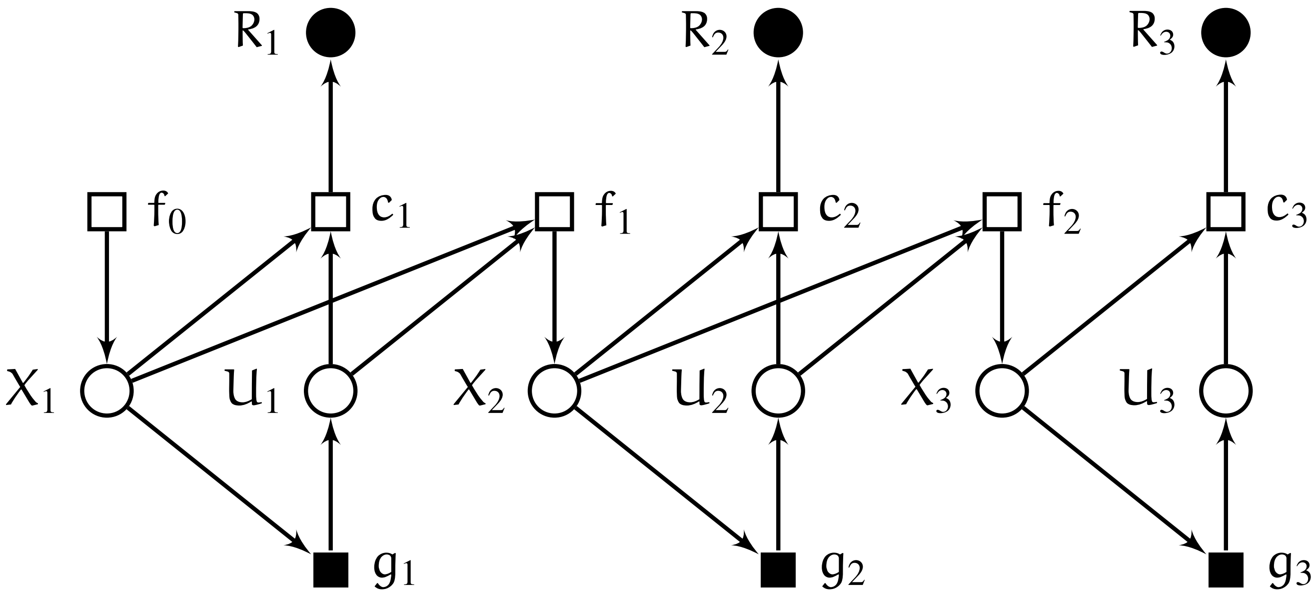
Representing teams using directed graphs

Steffen L. Lauritzen and Dennis Nilsson, Representing and Solving Decision Problems with Limited Information, Management Science-2001.



None of these fit our requirements perfectly. So, we use DAFG (Directed Acyclic Factor Graphs)

A graphical model for sequential team forms



A graphical model for sequential team forms

Directed Acyclic Factor Graph $\mathcal{G} = (V, F, E)$ for $\mathcal{T} = (N, A, R, \{I_k\}_{k \in N})$

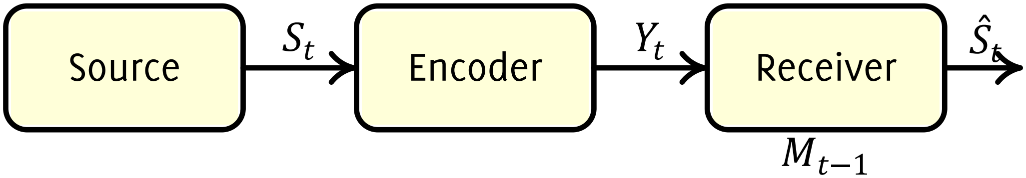
$$V = N \times \{0\}, \quad F = N \times \{1\}$$

$$E = \{(k^1, k^0) : k \in N\} \cup \{(i^0, k^1) : k \in N, i \in I_k\}$$

- Vertices
  - Variable Node $k^0 \equiv$ system variable X_k
  - Factor node $k^1 \equiv$ stochastic kernel p_k or control law g_k.
- Edges
  - (k^1, k^0) for each $k \in N$
  - (i^0, k^1) for each $k \in N$ and $i \in I_k$

An Example: Real-time communication

Hans S. Witsenhausen, On the structure of real-time source coders, BSTJ-79



First order Markov source $\{S_t, t = 1, \ldots, T\}$.

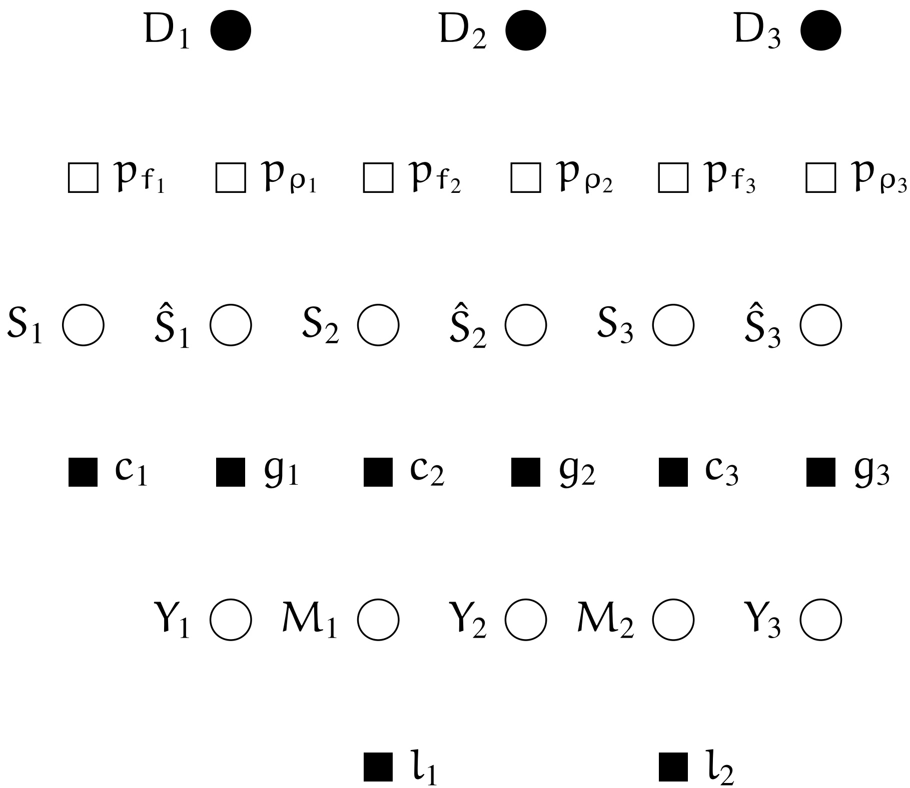
Real-Time Encoder: $Y_t = c_t(S^t, Y^{t-1})$

Real-Time Finite Memory Decoder: $\left|\begin{array}{l} \hat{S}_t = g_t(Y_t, M_{t-1}) \\ M_t = l_t(Y_t, M_{t-1}) \end{array}\right.$

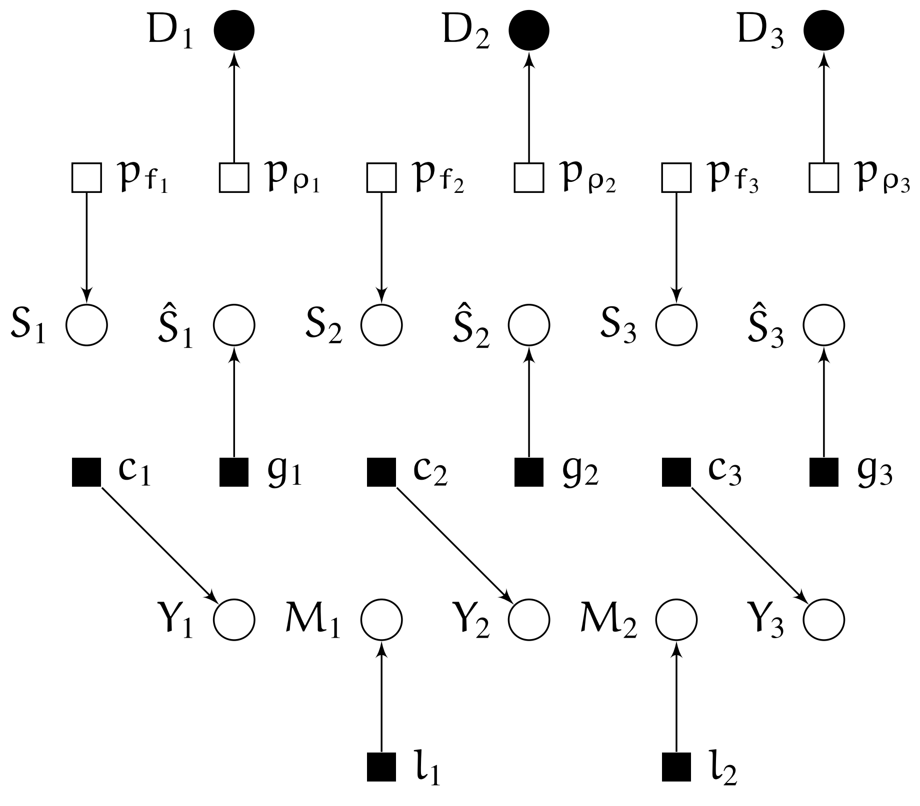
Instantaneous distortion $\rho(S_t, \hat{S}_t)$

Objective: minimize $E\left\{ \sum_{t=1}^{T} \rho(S_t, \hat{S}_t) \right\}$

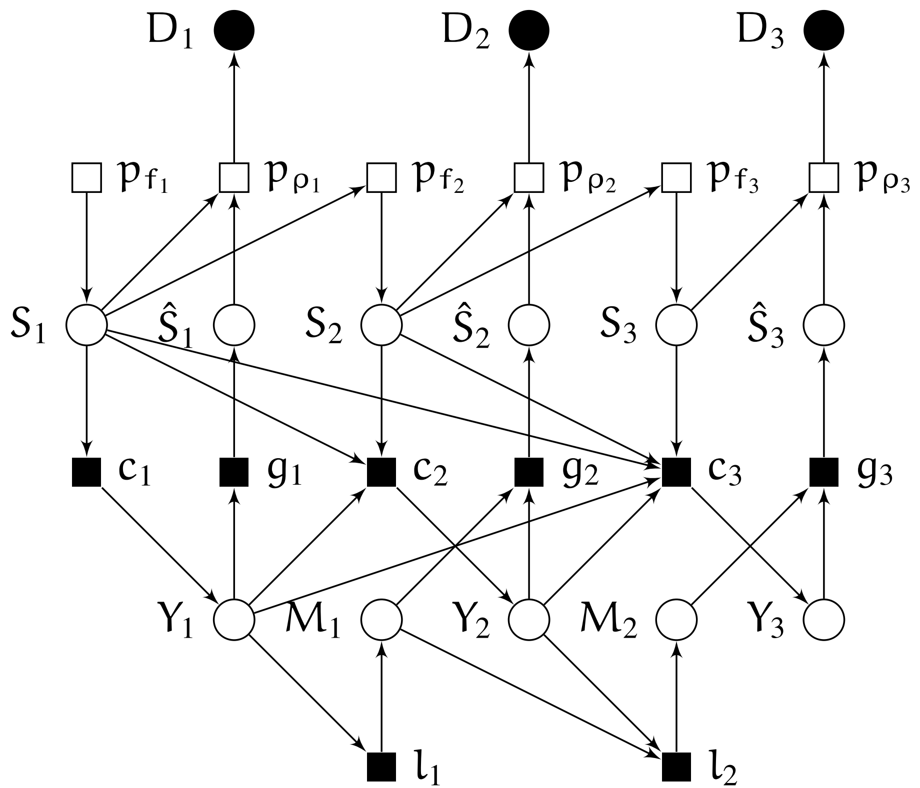
An Example: Real-time communication



An Example: Real-time communication



An Example: Real-time communication



Checking conditional independence

Dan Geiger, Thomas Verma, and Judea Pearl, Identifying independence in Bayesian networks, Networks-90.

Conditional independence can be efficiently checked on a directed graph.

Given a DAFG $\mathcal{G} = (V, F, E, D)$ and sets $A, B, C \subset V$, X_A is irrelevant to X_B given X_C if X_A is independent to X_B given X_C for all joint measures $P(dX_V)$ that recursively factorize according to $\mathcal{G}$.

Data irrelevant to X_A given X_C is

$$R^{-}_{\mathcal{G}}(X_A|X_C) = \{k \in C : X_k \text{ is irrelevant to } X_A \text{ given } X_C \setminus \{X_k\}\}$$

Back to simplification of team forms

Completion of a team

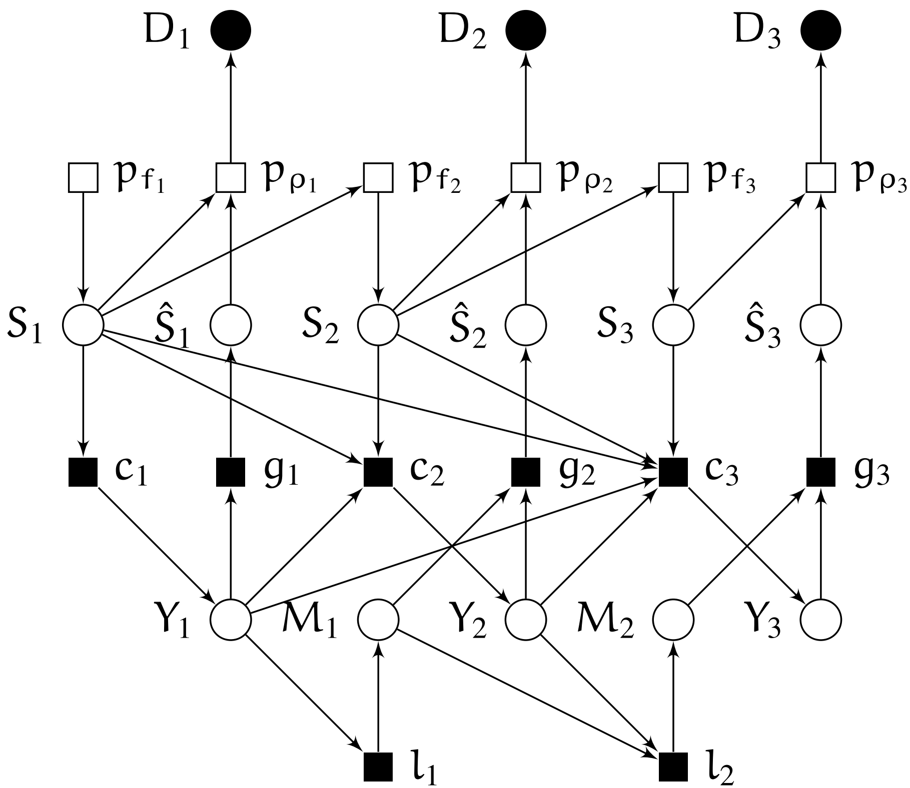
A team form $\mathcal{T} = (N, A, R, \{I_k\}_{k \in N})$ is complete if for $k, l \in A$, $k \neq l$, such that $I_k \subset I_l$ we have $X_k \in I_l$. (If l knows the data available to k, then l also knows the action taken by k).

If a team is not complete, it can be completed by sequentially adding “missing links”

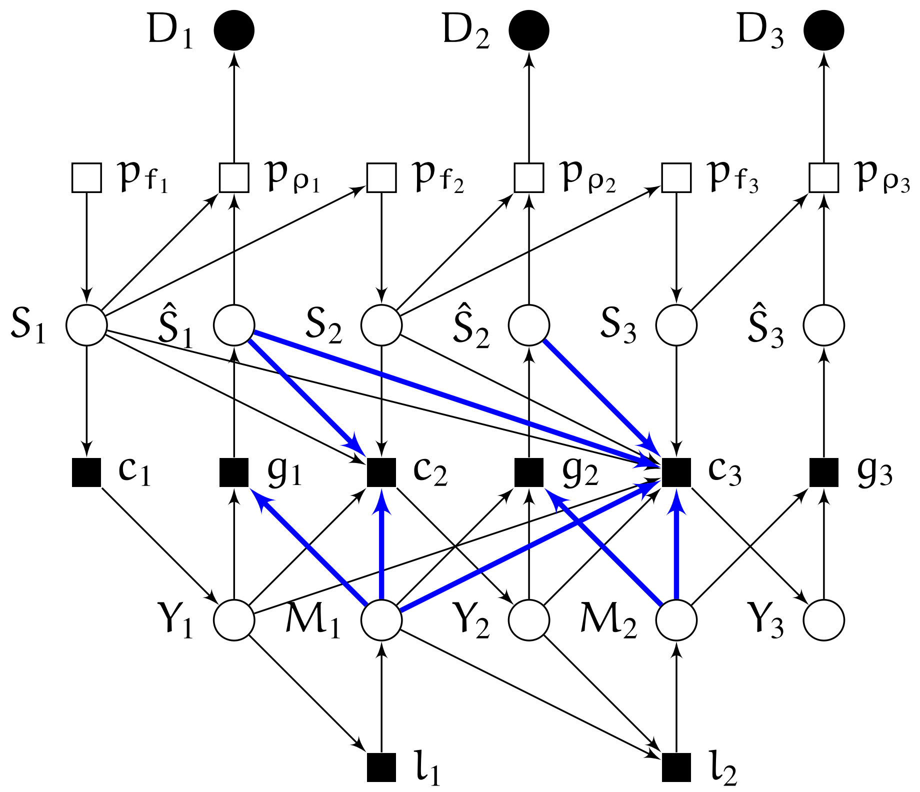
Depending on the order in which we proceed, we can end up with different completions. However,

> all completions of a team form are equivalent.

Completion of a team form



Completion of a team



Simplification of team forms

Step 1: Complete the team form.

(Note: All completions of a team form are equivalent to the original)

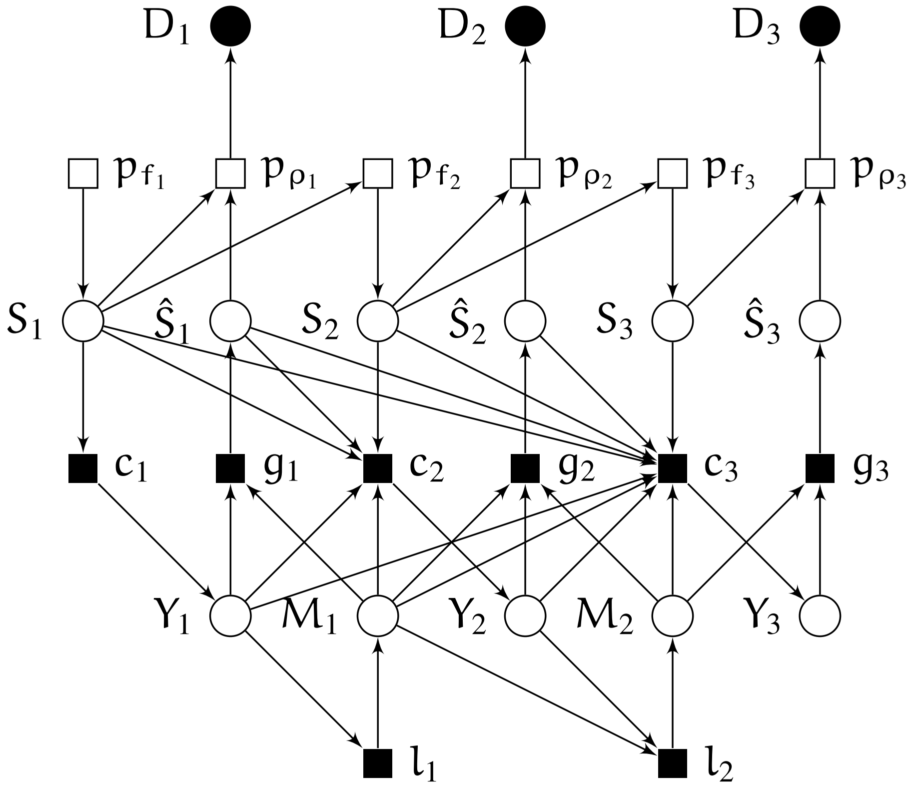
Removing irrelevant nodes

Recall Given a DAFG $\mathcal{G} = (V, F, E, D)$ and sets $A, B, C \subset V$, X_A is irrelevant to X_B given X_C if X_A is independent to X_B given X_C for all joint measures $P(dX_V)$ that recursively factorize according to $\mathcal{G}$ and

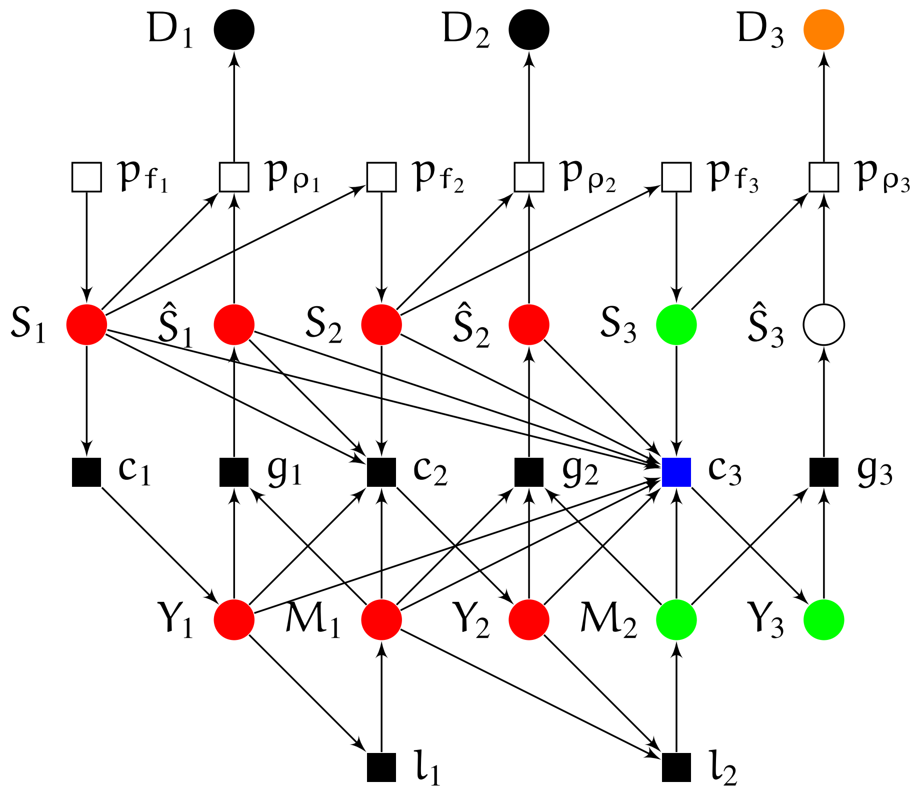
$$R^-_{\mathcal{G}}(X_A | X_C) = \{k \in C : X_k \text{ is irrelevant to } X_A \text{ given } X_C \setminus \{X_k\}\}$$

For any $k \in A$ in a team form $\mathcal{T} = (N, A, R, \{I_k\}_{k \in N})$, replacing X_{I_k} by $X_{I_k} \setminus \left(R^-_{\mathcal{G}}(X_R \cap \overrightarrow{X_k} \mid X_{I_K}, X_k) \setminus X_k \right)$ does not change the value of the team.

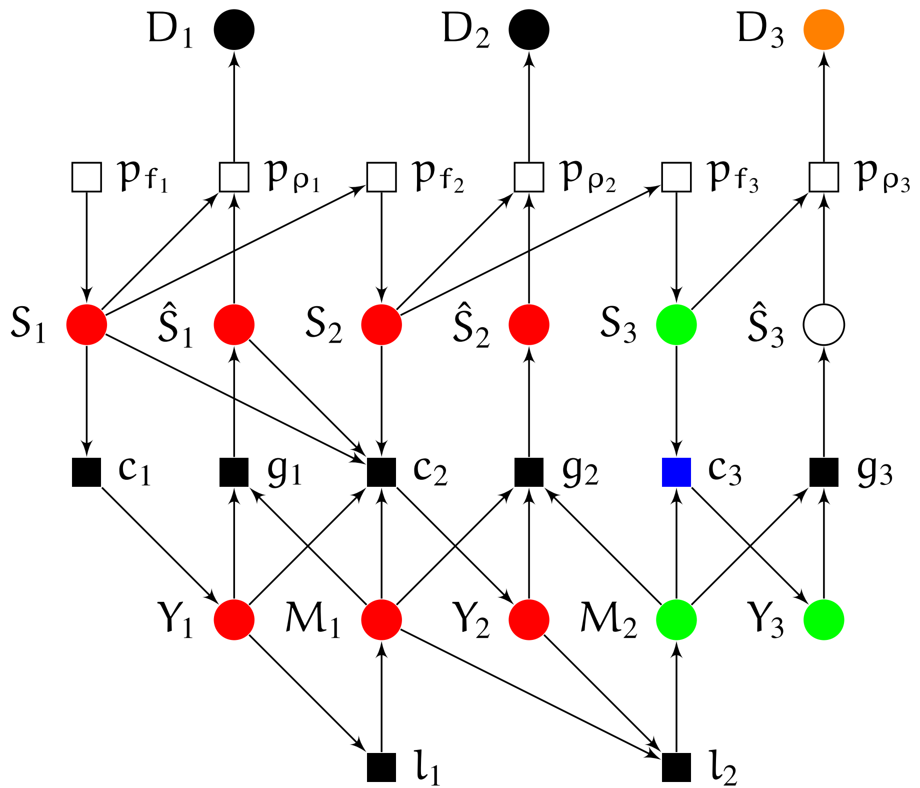
Remove irrelevant nodes



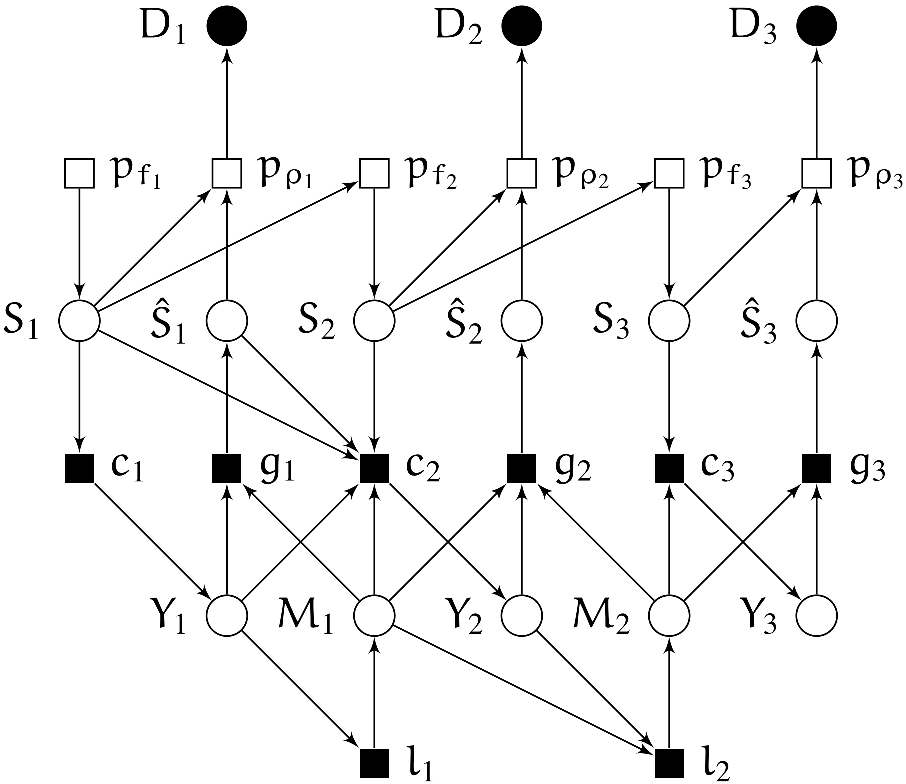
Remove irrelevant nodes



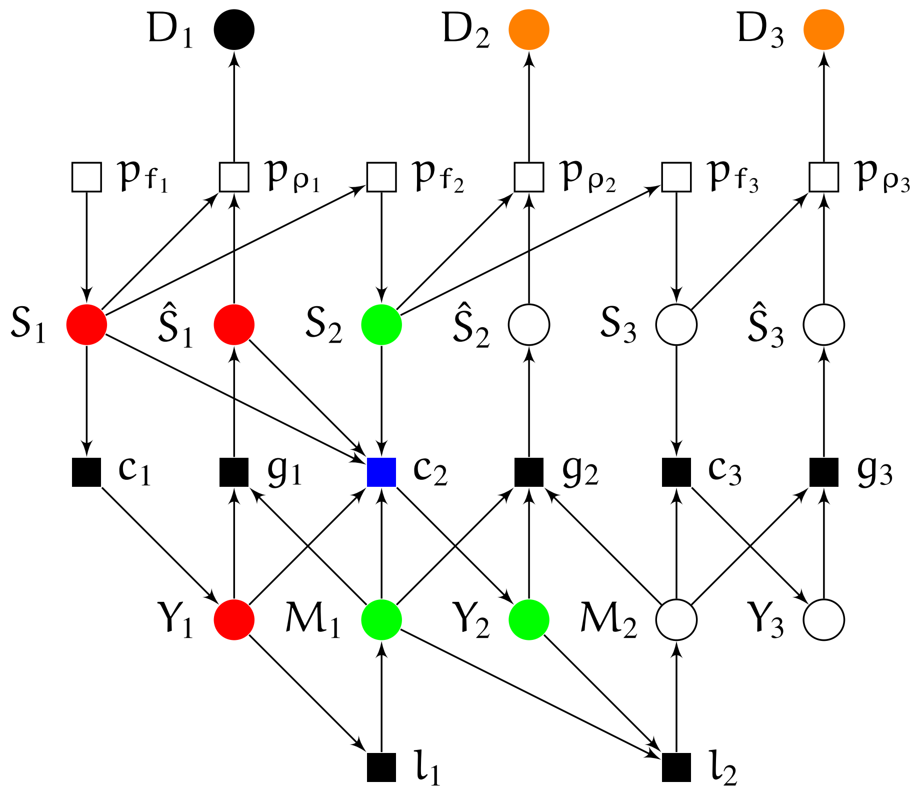
Remove irrelevant nodes



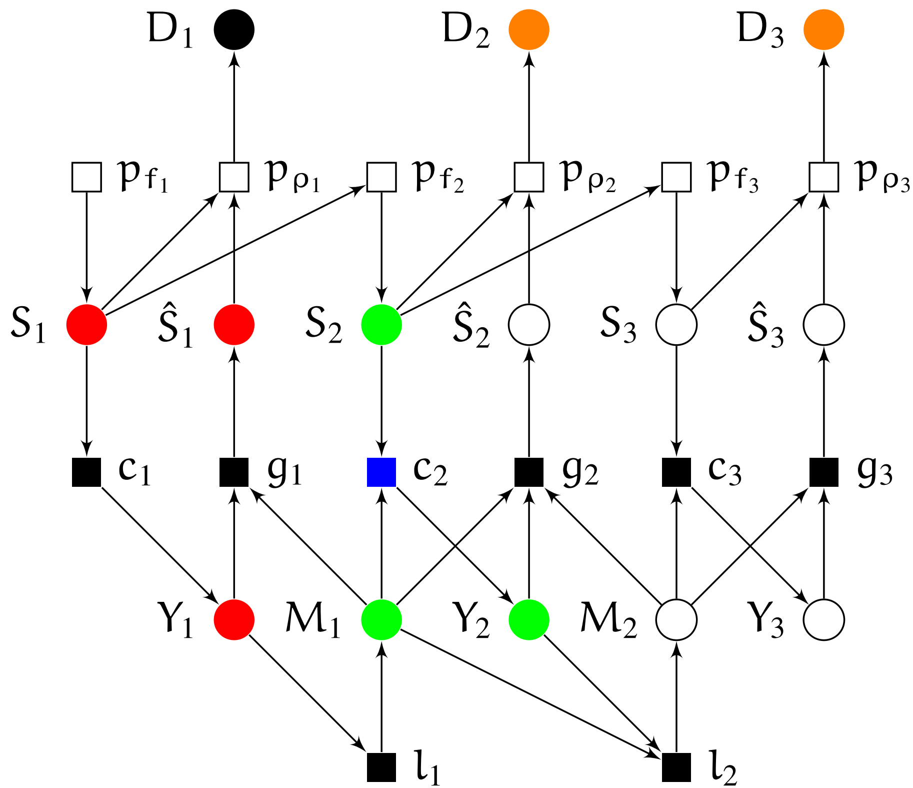
Remove irrelevant nodes



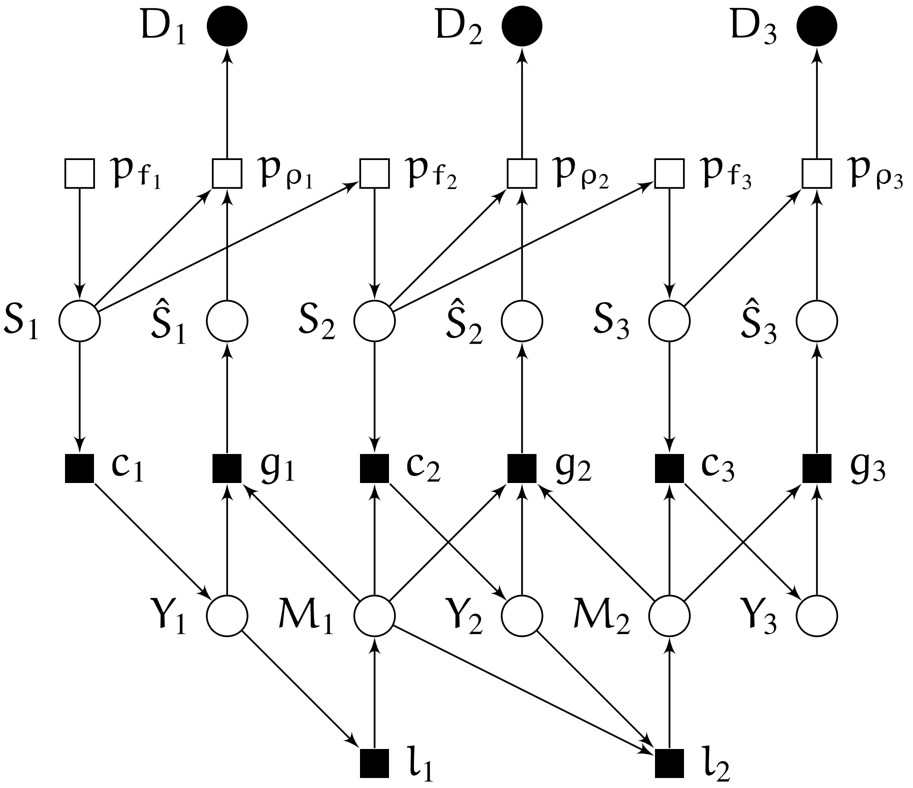
Remove irrelevant nodes



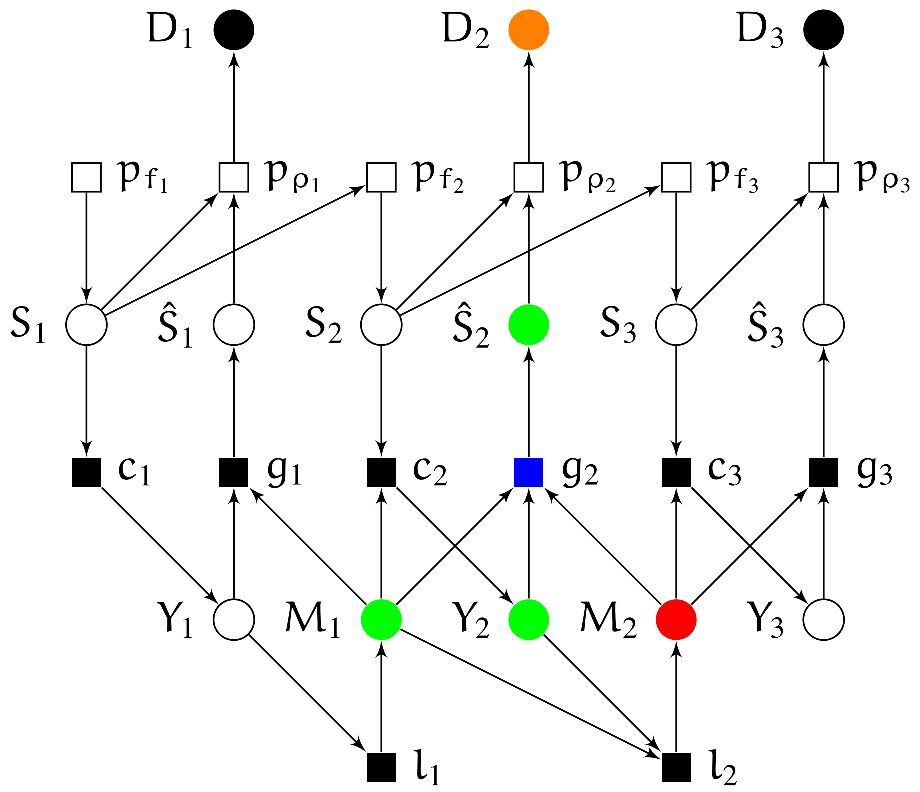
Remove irrelevant nodes



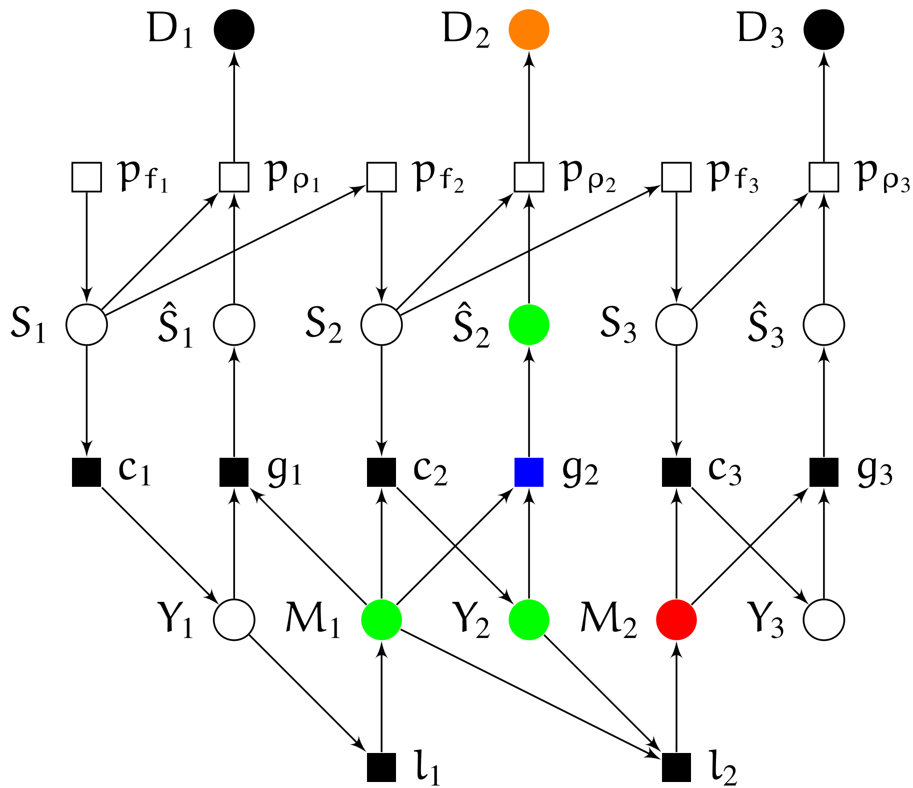
Remove irrelevant nodes



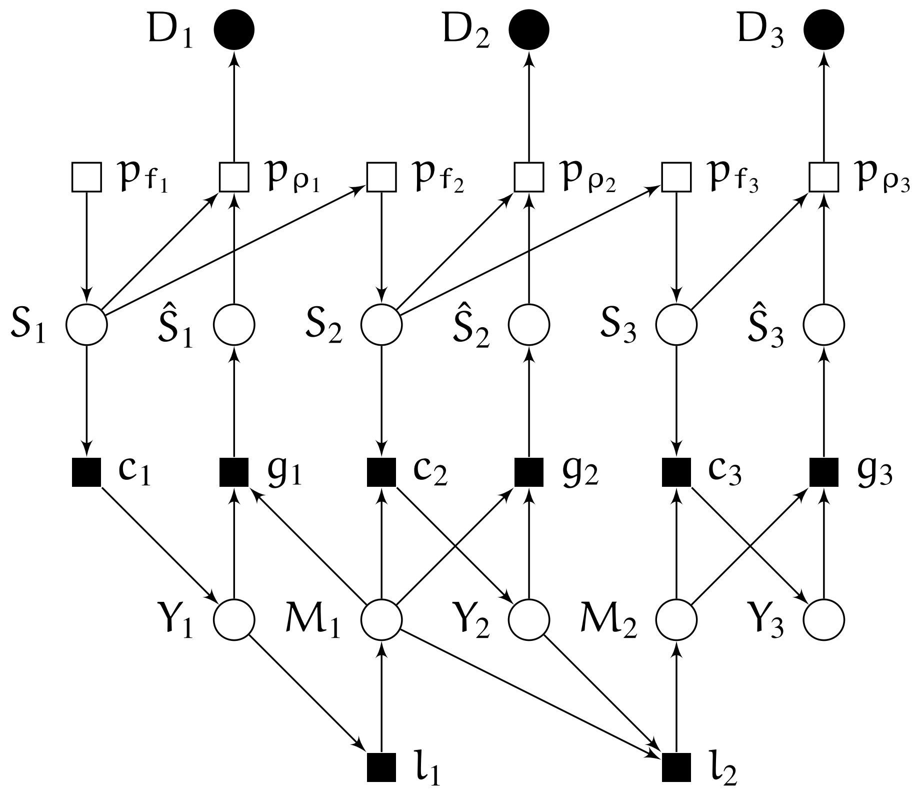
Remove irrelevant nodes



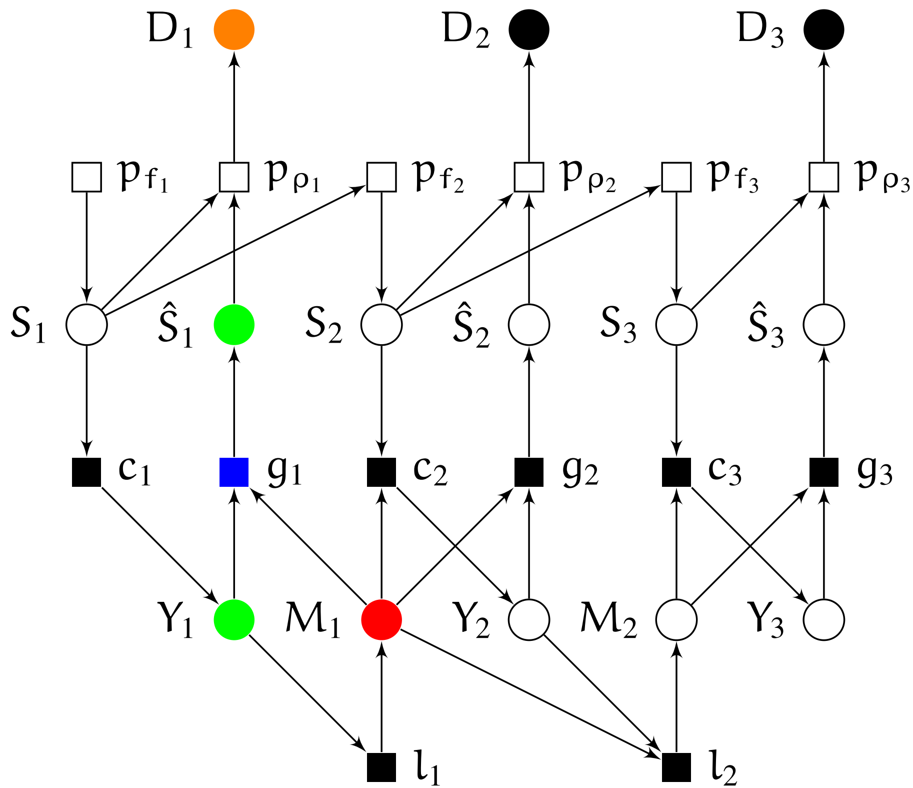
Remove irrelevant nodes



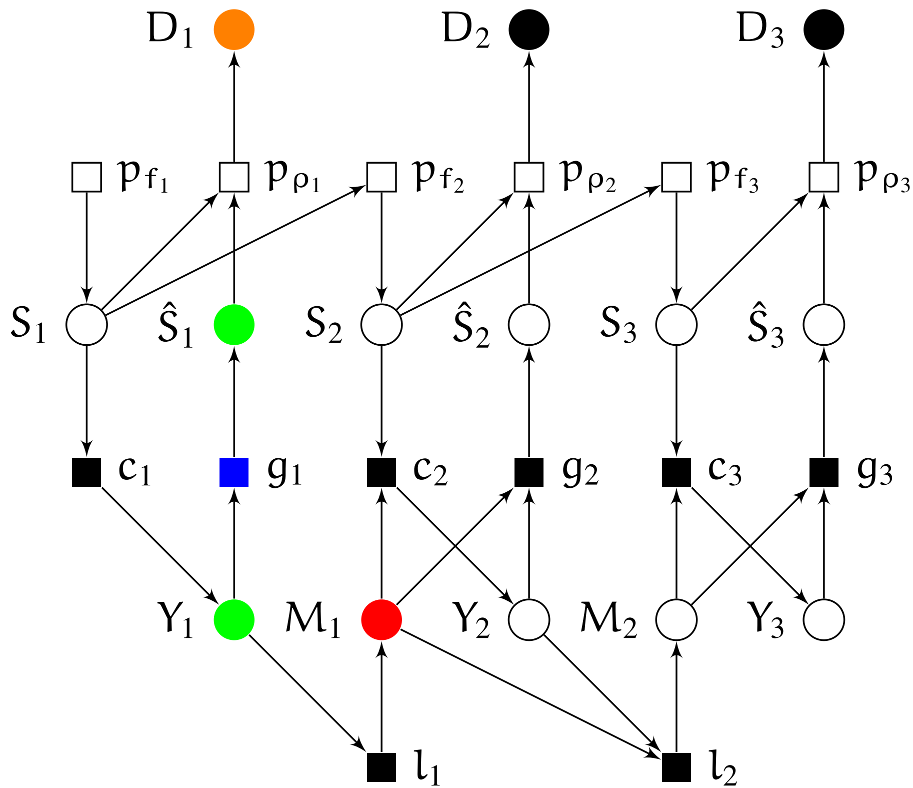
Remove irrelevant nodes



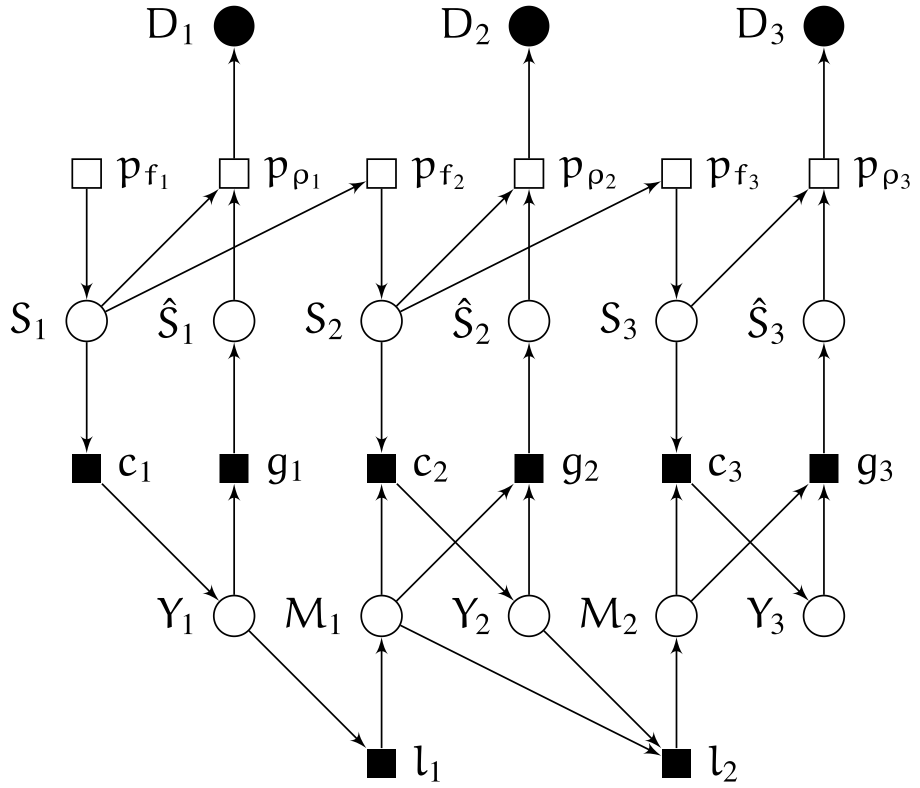
Remove irrelevant nodes



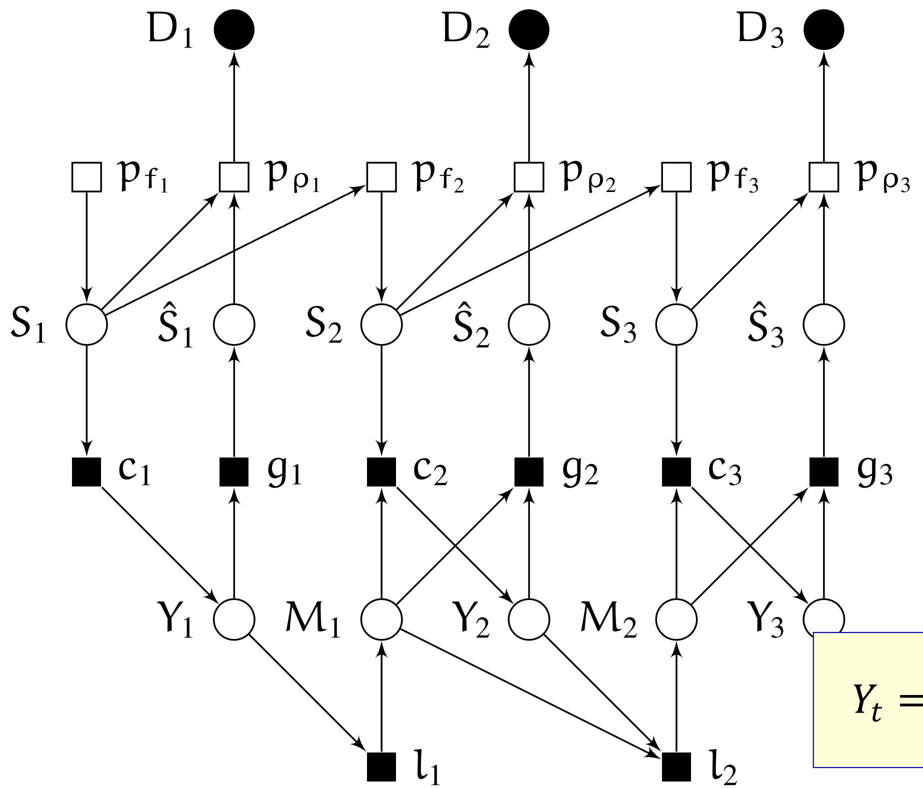
Remove irrelevant nodes



Remove irrelevant nodes



Removing irrelevant nodes



$$Y_t = c_t(S_t, M_{t-1})$$

Simplification of team forms

Step 1: Complete the team form.

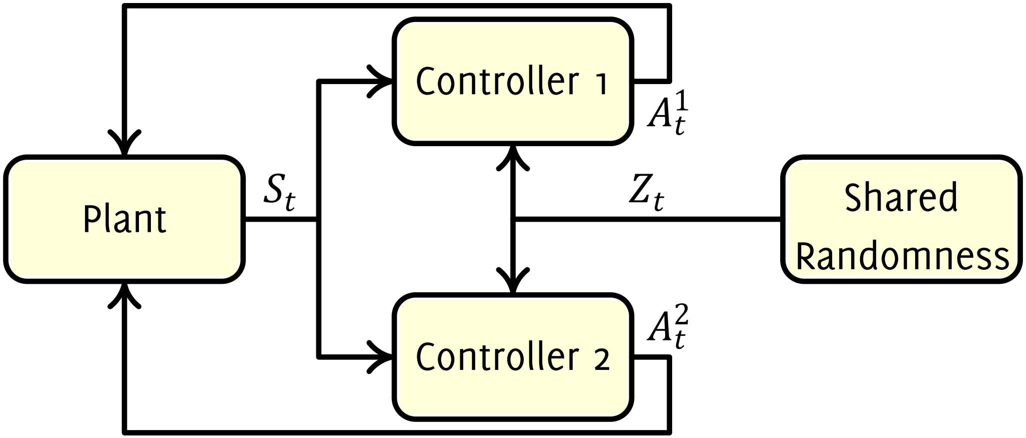
(Note: All completions of a team form are equivalent to the original)

Step 2: At control factor node k, remove incoming edges from nodes irrelevant to $X_R \cap \overrightarrow{X}_k$ given (X_{I_k}, X_k)

(Note: The resultant team form is equivalent to the original)

Coordinator for a subset of agents

Another Example: Shared randomness



Plant: $S_{t+1} = f_t(S_t, A_t^1, A_t^2, W_t)$

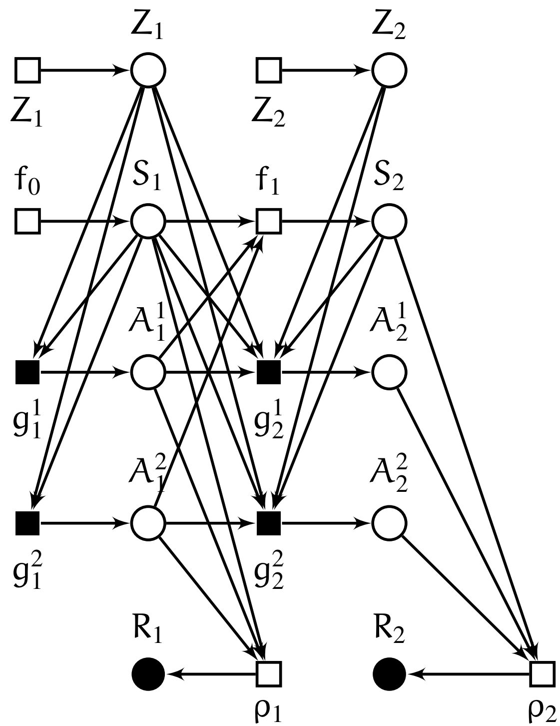
Shared Randomness: $\{Z_t, t = 1, \ldots, T\}$ indep. of rest of system

Control Station 1: $A_t^1 = g_t^1(S^t, A^{1,t-1}, Z^t)$

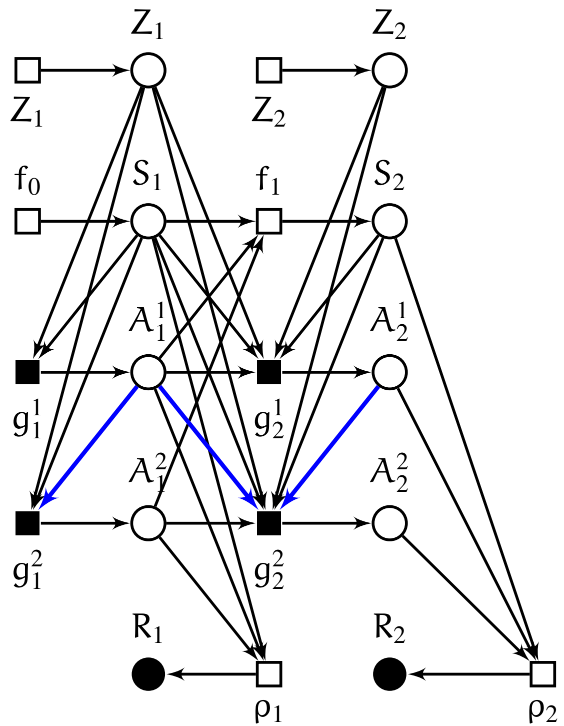
Control Station 2: $A_t^2 = g_t^2(S^t, A^{2,t-1}, Z^t)$

Instantaneous cost: $\rho_t(S_t, A_t^1, A_t^2)$

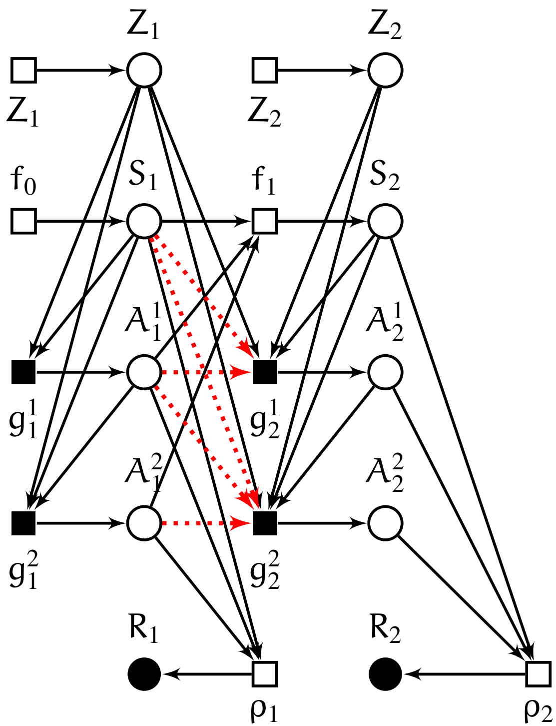
Another Example: Shared randomness



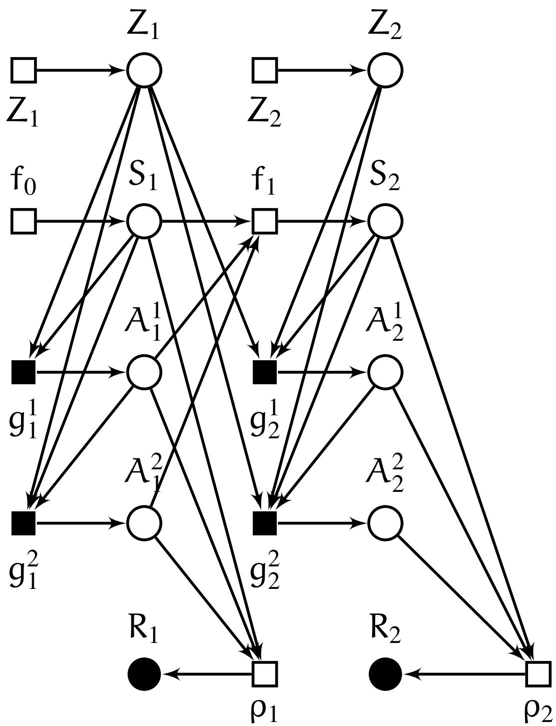
Another Example: Shared randomness (Step 1)



Another Example: Shared randomness (Step 2)



Another Example: Cannot remove useless sharing



Each agent thinks that the other might use it

Coordinator for a subset of agents

For $a, b \in A$, consider a coordinator that observes shared information $X_C := X_{I_a} \cap X_{I_b}$ and chooses partial functions $\hat{g}_a : X_{I_a \setminus C} \to X_a$ and $\hat{g}_b : X_{I_b \setminus C} \to X_b$.

Agent a and b simply carry out the computations prescribed by $\hat{g}_a$ and $\hat{g}_b$

Remove irrelevant incoming edges at the coordinator!

Equivalently, at agents a and b, remove edges from nodes that are irrelevant to $X_R \cap \overrightarrow{X}_{\{a,b\}}$ given $(X_C, \hat{g}_a, \hat{g}_b)$.

Coordinator for a subset of agents

For any $B \subset A$ in a team form $\mathcal{T} = (N, A, R, \{I_k\}_{k \in N})$
and any $b \in B$, let $X_C = \bigcap_{b \in B} X_{I_b}$.

Then, replacing X_{I_b} by $X_{I_b} \setminus \left(R^{-}_{\mathcal{G}}(X_R \cap \overrightarrow{X}_B \mid X_C, \hat{g}_B) \setminus \hat{g}_B \right)$
does not change the value of the team

Simplification of team forms

Step 1: Complete the team form.

(Note: All completions of a team form are equivalent to the original)

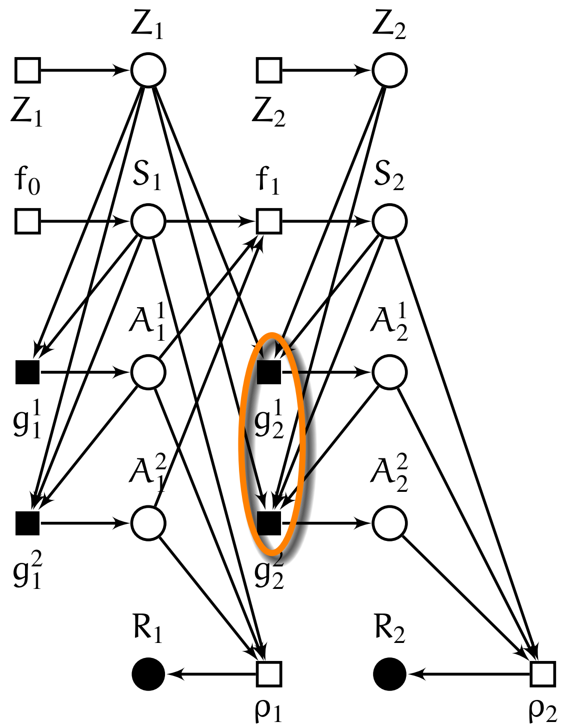
Step 2: At control factor node k, remove incoming edges from nodes irrelevant to $X_R \cap \overrightarrow{X}_k$ given (X_{I_k}, X_k)

(Note: The resultant team form is equivalent to the original)

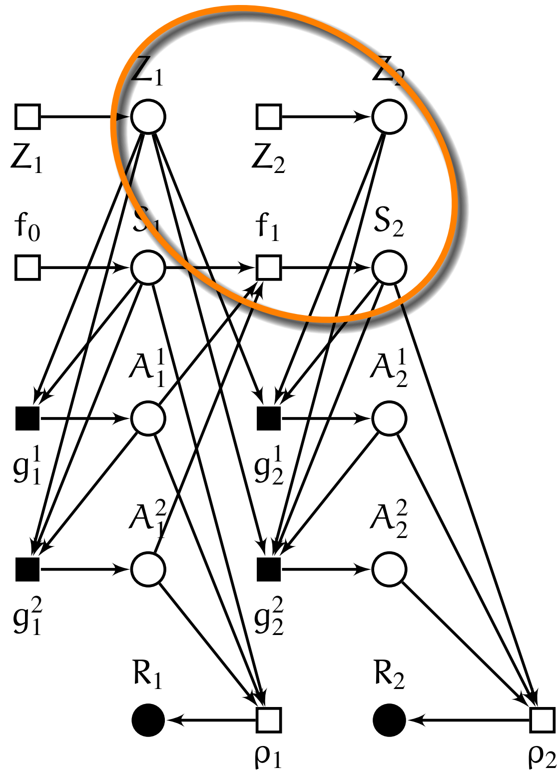
Step 3: At all nodes of any subset B of A, remove incoming edges from nodes irrelevant to $X_R \cap \overrightarrow{X}_B$ given $(\bigcap_{b \in B} X_{I_b}, \bigcup_{b \in B} \hat{g}_b)$.

(Note: The resultant team form is equivalent to the original.)

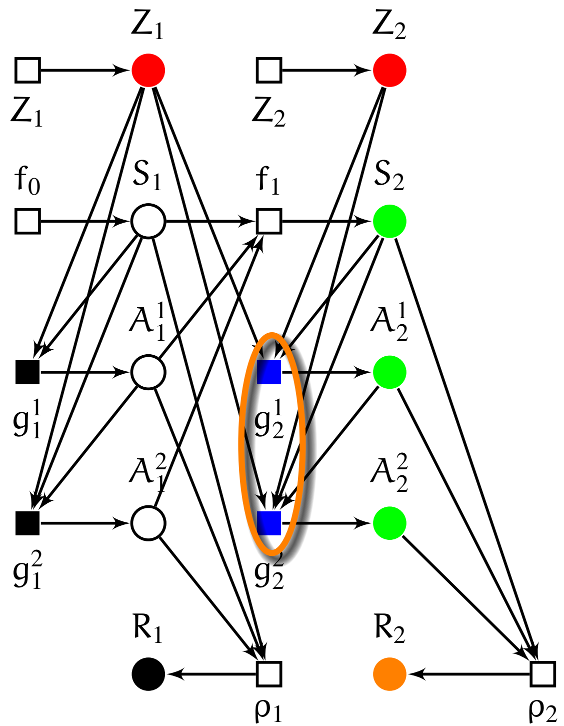
Removing shared randomness: Coordinator



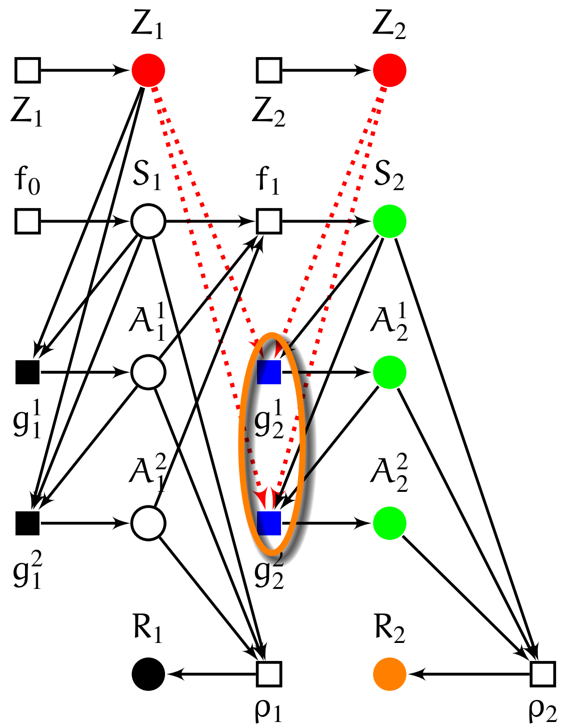
Removing shared randomness: Coordinator's observation



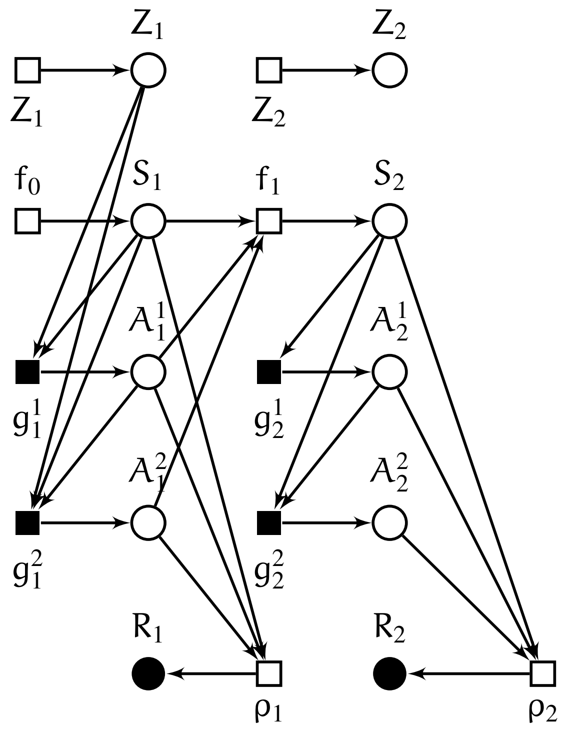
Removing shared randomness: Coordinator



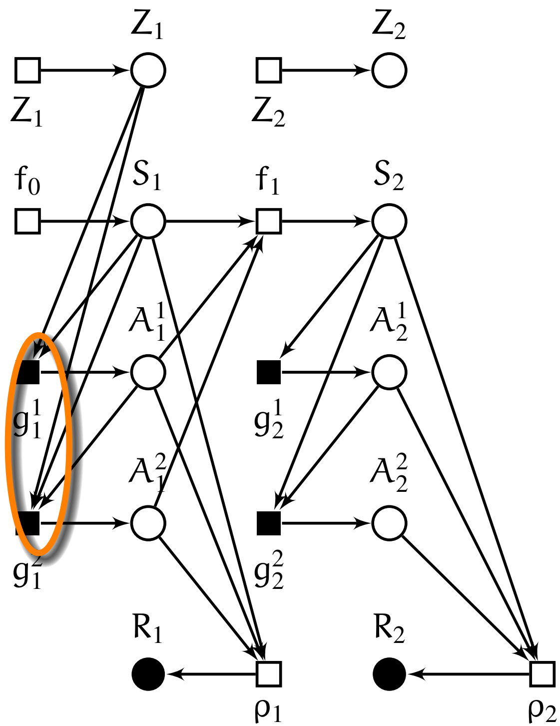
Removing shared randomness: Coordinator



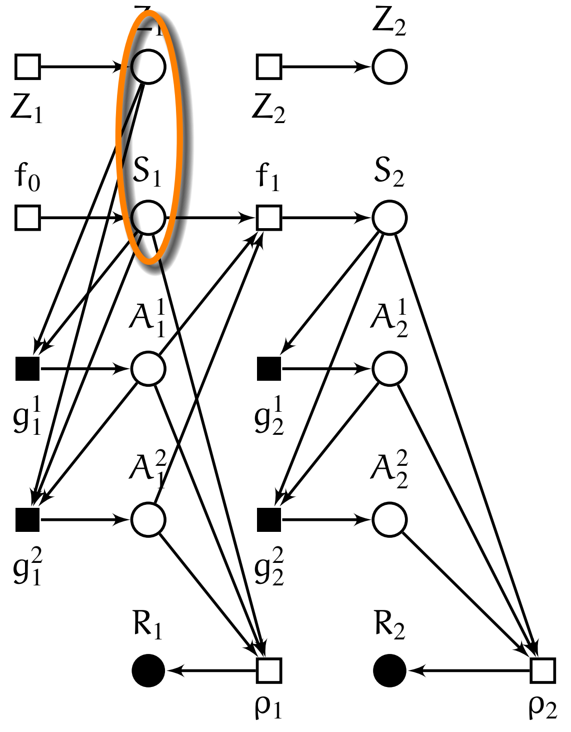
Removing shared randomness: Edges removed



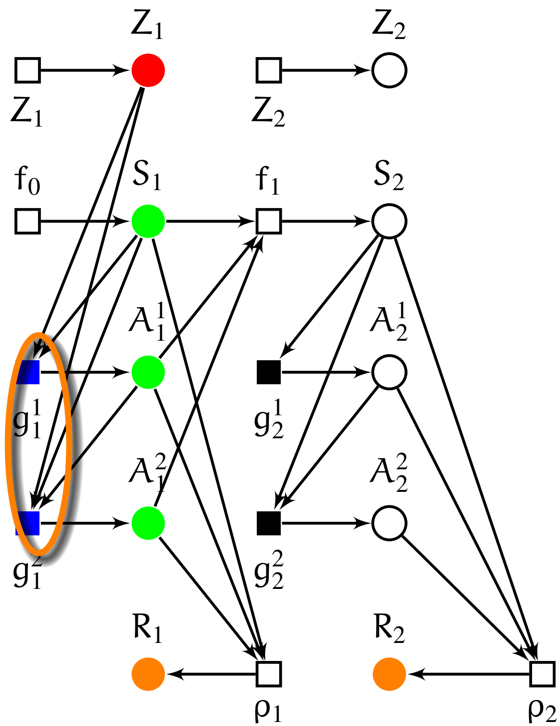
Removing shared randomness: New coordinator



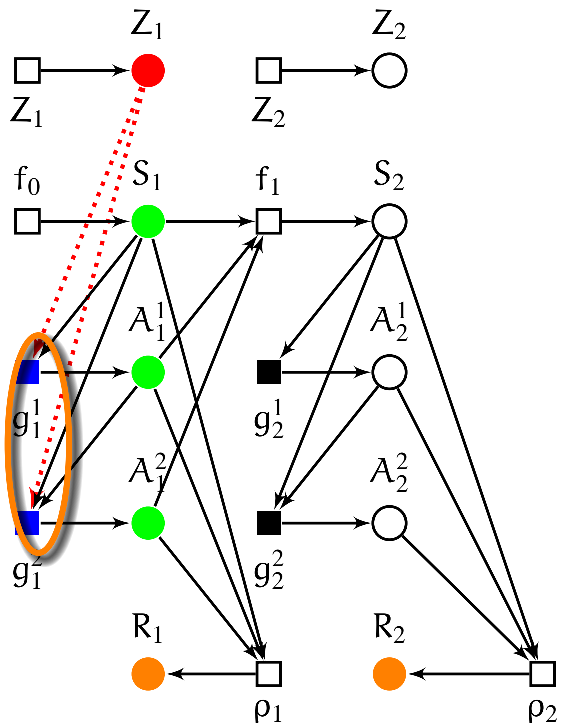
Removing shared randomness: Shared Observation



Removing shared randomness: Coordinator



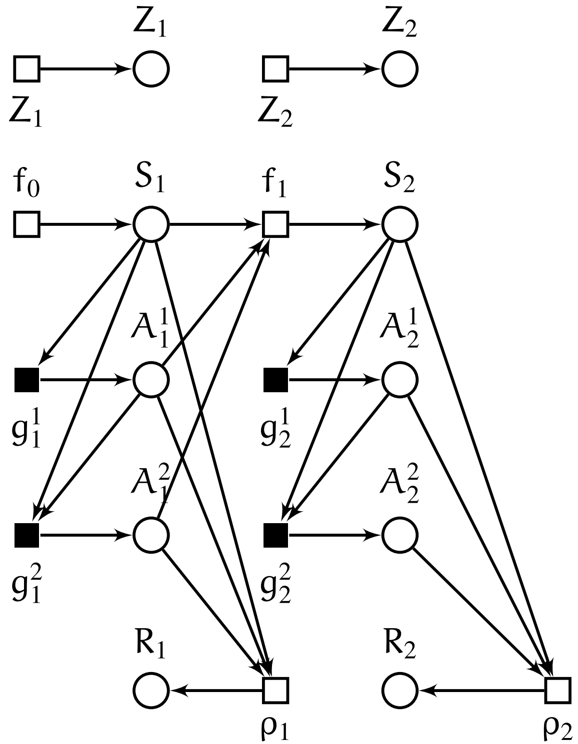
Removing shared randomness: Coordinator



Shared randomness: final result

$$A_t^1 = g_t^1(S_t)$$

$$A_t^2 = g_t^2(S_t, A_t^1)$$



Summary

Simplification of team forms

Step 1: Complete the team form.

(Note: All completions of a team form are equivalent to the original)

Step 2: At control factor node k, remove incoming edges from nodes irrelevant to $X_R \cap \overrightarrow{X}_k$ given (X_{I_k}, X_k)

(Note: The resultant team form is equivalent to the original)

Step 3: At all nodes of any subset B of A, remove incoming edges from nodes irrelevant to $X_R \cap \overrightarrow{X}_B$ given $(\bigcap_{b \in B} X_{I_b}, \bigcup_{b \in B} \hat{g}_b)$.

(Note: The resultant team form is equivalent to the original.)

Main ideas

- Observed data that is irrelevant for dependent rewards can be ignored
  - Irrelevant data can be identified using standard graphical models algorithms
- A coordinator for a collection of agents
  - Shared information between collection of agents can be efficiently represented as a lattice

More examples

Works for all examples of (MDP-like) structural results in the literature.

- Real-time communication (point-to-point with and without feedback, multi-terminal communication with feedback)

- Networked control systems

- specific forms of information structures (delayed state sharing, stochastically nested, etc.)

Conclusion

- Presented team forms for decentralized systems, and the notions of equivalence and simplification of team forms.

- A team form can be naturally represented as a DAFG

- The DAFG of a team form can be simplified axiomatically.
  - The process in intuitive
  - The algorithm is efficient and can be automated easily. (see http://pantheon.yale.edu/~am894/code/teams/ for software implementation)

Future Directions

- What about other types of structural results? Adding belief variables in POMDPs? Adding beliefs on beliefs in decentralized teams.

  Is equivalent to adding nodes representing conditional independence on a graphical model. Need to develop conditional independence properties of such a graphical model.

  Is related to notions of state in systems of interacting probabilistic automata and interacting particle systems.

- What about other models? Graphical model is not the only way to check condition independence

  Conditional independence can also be checked on a relationship lattice. Lattices naturally capture important notions of decentralized systems like shared information, partial orders, and state with respect to a cut, etc.

Future Directions

- What about sequential decomposition? Can we write optimality equations of a general decentralized system axiomatically?

  Has already been done—Witsenhausen's standard form. However, it is not the most efficient solution. The model presented in this talk can be used to identify optimality equations what have a smaller state space.

  Many engineering systems have more structure. Can we exploit that structure to say something about infinite horizon systems?

- What about non-sequential systems? Everything here is based on partial orders. Non-sequential systems do not have a partial order between agents.

  Non-sequential systems form a pre-order. Not sure about the right notion for irrelevant variables. There are some relations between pre-orders and finite topological spaces.

Thank you