Decision Making

in Robots and Autonomous Agents

Graphical Models for Strategic Games

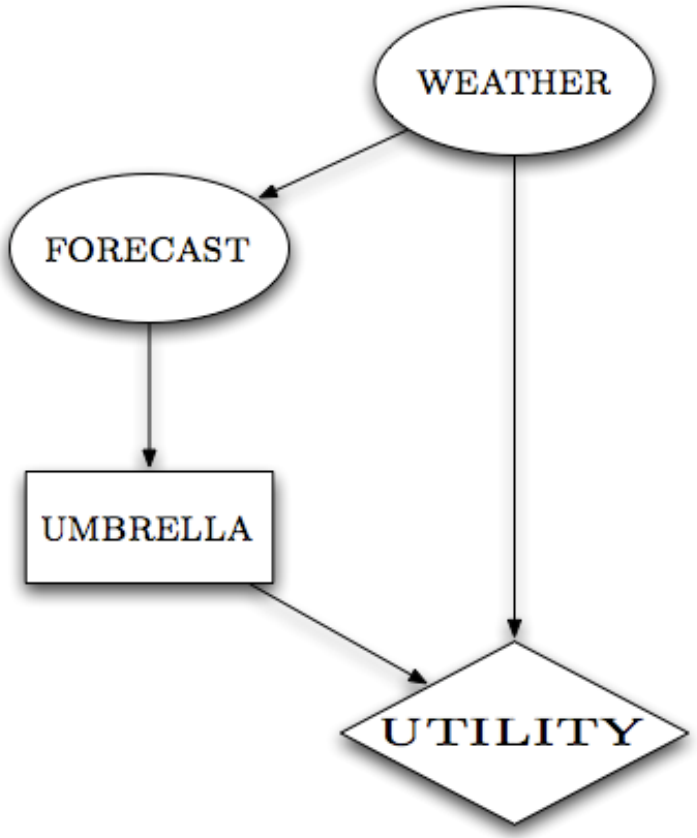
(based on AAAI 2008 Tutorial by Gal & Pfeffer)

Subramanian Ramamoorthy
School of Informatics

6 March, 2015

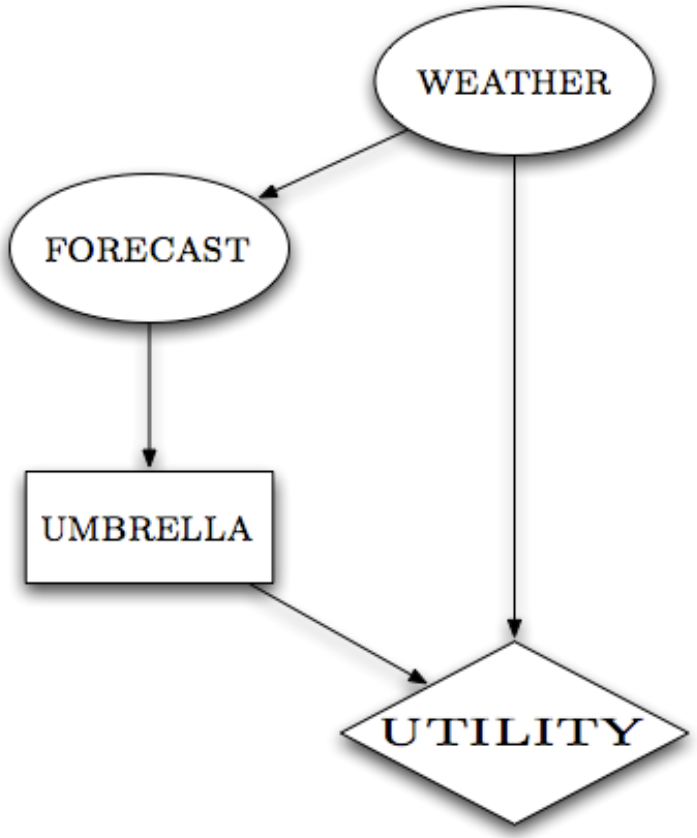
Recap: Influence Diagrams [Howard & Matheson '84]

- Influence Diagrams extend Bayesian Networks for decision making.
- *Rectangles* are decisions; *ovals* are chance variables; *diamonds* are utility functions.
- Graph topology describes decision problem.
- Each node specifies a probability distribution (CPD) given each value of parents.

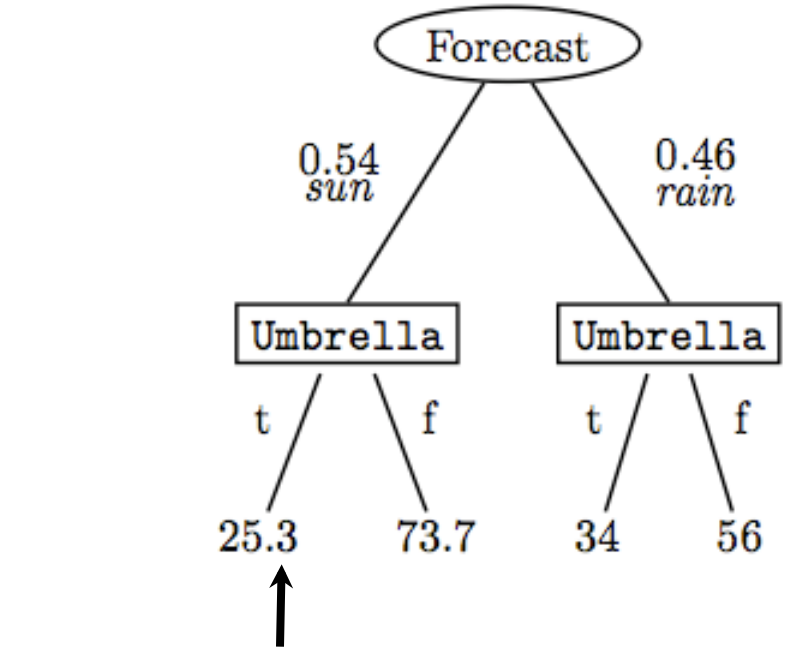


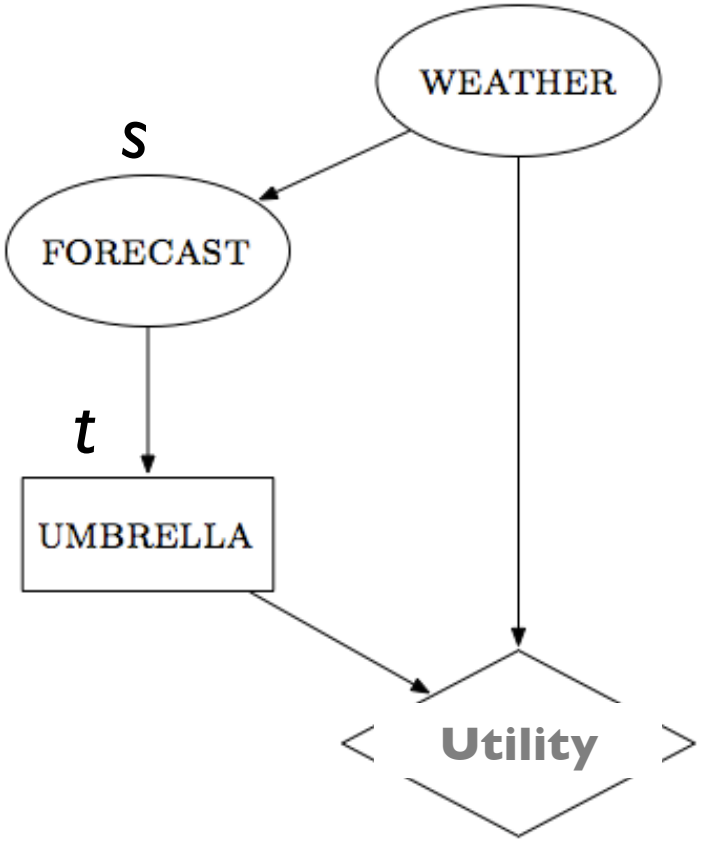
Recap: Influence Diagrams (ID)

- Parents of decisions (informational parents) represent observations.
- Parents of chance nodes represent probabilistic dependence.
- Parents of utility nodes represent the parameters of the utility functions.
- A strategy for a decision is a function from its informational parents to a choice for the decision. For each observation, a *pure* strategy prescribes a single choice of action for an agent.



Recap: Converting ID to Decision Tree (Umbrella Example)





Disadvantage : We lose graph structure

Ultimatum *Game* Example

- Consider this problem from your assignment
- Two players: Proposer and responder.
- Proposer can offer some split of 3 coins to Responder.

Ultimatum Game Example

- Proposer can offer some split of 3 coins to Responder. If Responder accepts, offer is enforced; if Responder rejects, both receive nothing.

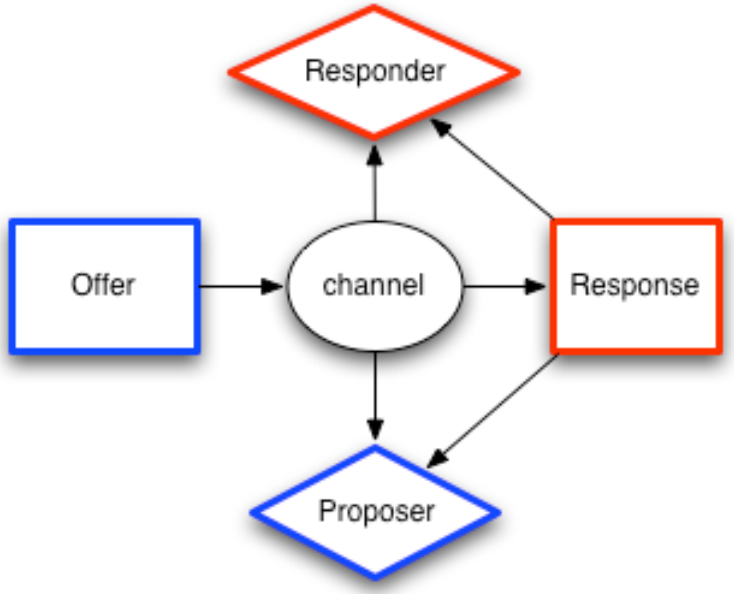
An Ultimatum Game Example

- Proposer can offer some split of 3 coins to Responder. If Responder accepts, offer is enforced; if Responder rejects, both receive nothing. Offer may be corrupted and set to (1,2) split (proposer/responder) by noisy channel with 0.1 probability.

Multi-agent Influence Diagrams

[Milch and Koller '01]

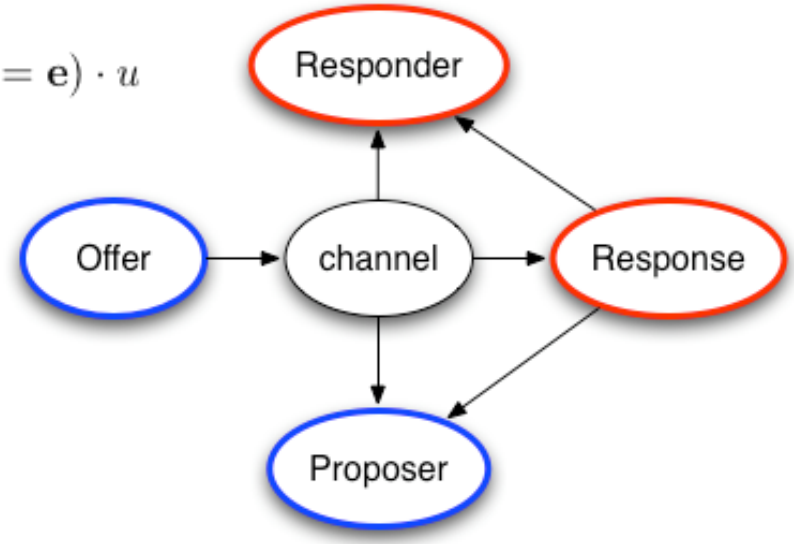
- Extend Influence Diagrams to the multi-agent case.
- Rectangles and diamonds represent decisions and utilities associated with agents; ovals represent chance variables.
- A strategy for a decision is a mapping from the informational parents of the decision to a value in its domain.
- A strategy profile includes strategies for all decisions.



Computing Expected Utility in MAIDs

Let α be an agent, and ***s*** be a strategy profile for all decisions in a MAID. Let $P^{\mathbf{s}}$ be the distribution over the BN that implements ***s*** in a MAID. Let ***U*** be the set of utility nodes for α, and ***E*** be evidence nodes. The utility for α given ***s*** and evidence ***E=e*** is

$$U^{\mathbf{s}}(\alpha \mid \mathbf{E} = \mathbf{e}) = \sum_{U \in \mathbf{U}} \sum_{u \in \mathrm{DOM}(U)} P^{\mathbf{s}}(U = u \mid \mathbf{E} = \mathbf{e}) \cdot u$$



MAID Equilibrium

A strategy profile $\boldsymbol{\Theta}$ in the MAID is a Nash equilibrium if for any decision D_i belonging to agent α, we have

$$\theta_i(. \mid \mathbf{pa}_i) \in \text{argmax}_{\theta_i \in \Delta(S_i)} EU^{(\theta_i, \boldsymbol{\Theta}_{-\mathbf{i}})}(\alpha \mid \mathbf{pa}_i)$$

If the strategy profile $\boldsymbol{\Theta}$ is a Nash equilibrium of a MAID, then $\boldsymbol{\Theta}$ is also the Nash equilbirum of the extensive form game.

Solving MAIDs

- Naïve solution: Convert MAID to extensive form game, and solve it.

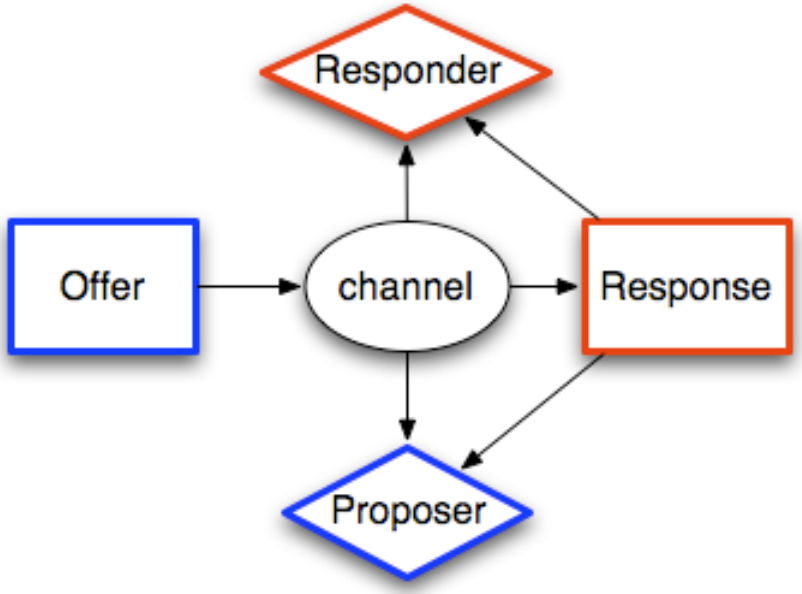
  ...but lose the structure of the MAID.

  (If you have looked at HW 1, you know this one)

- There is an alternative method that works directly on the MAID graph. We will define a new graphical criterion for expressing dependence between decisions.

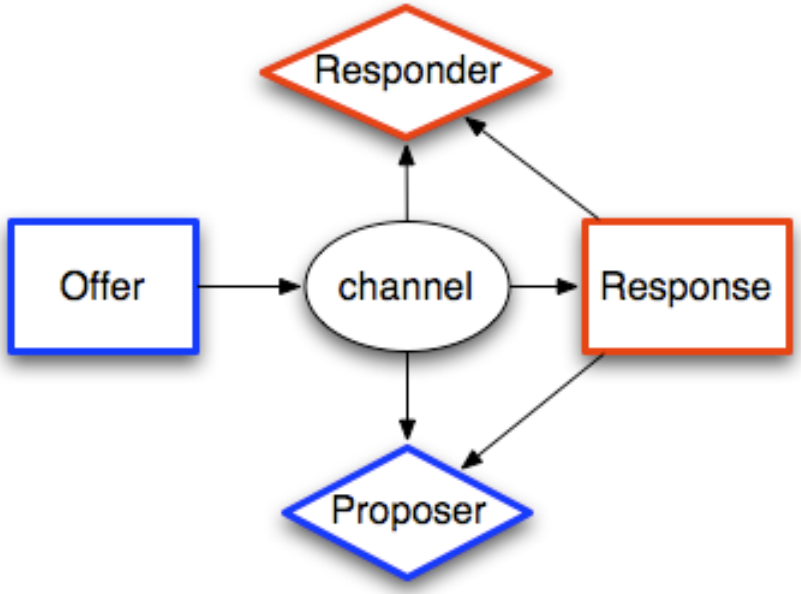
Strategic Relevance

- A decision *D* is strategically relevant to decision D' belonging to some agent if its utility depends on the strategy for D.
- Strategic relevance is a relation that holds between any two decisions in the MAID.



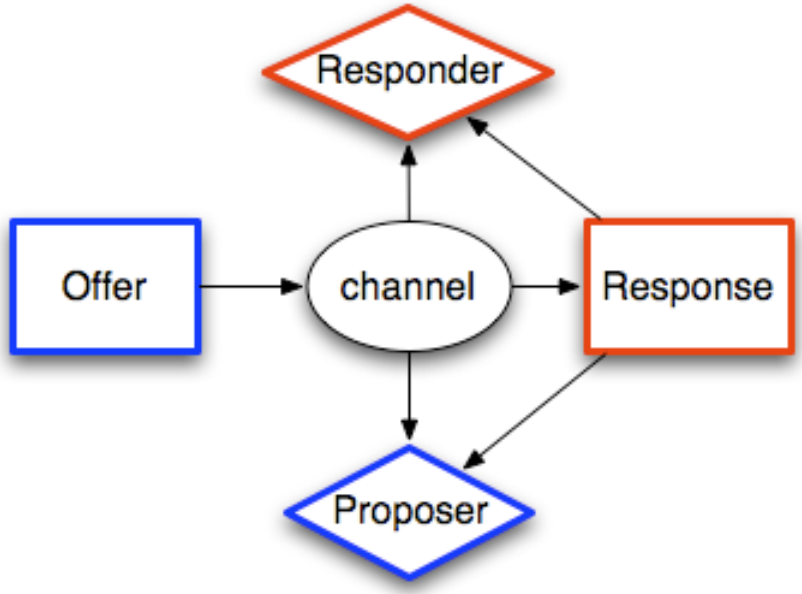
Strategic Relevance: Example

- Given a strategy for the responder
  - accept splits (1,2), (0,3).
- there exists an optimal strategy for the proposer
  - offer (1,2) split.
- Conclusion: Proposer could do well if it knew the responder's strategy.



Strategic Relevance: Example

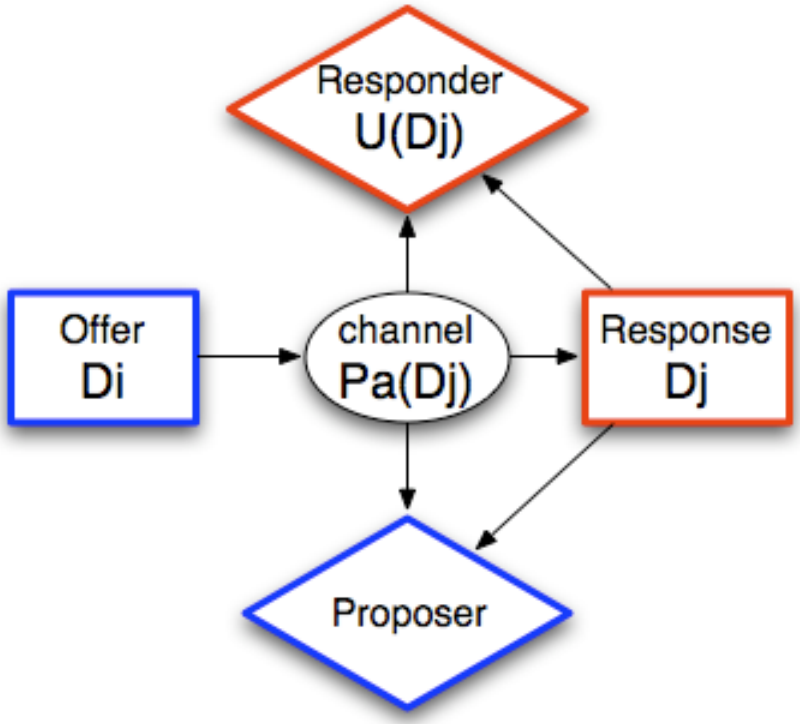
- Given strategy for Proposer
  - propose split (2,1).
- The optimal strategy for responder is
  - agree to beneficial offer reported by channel.
- The proposer's action affects the channel.
- Responder cares about proposer's *action*, but not its *strategy*.



Responder's strategy is relevant to Proposer
Proposer's strategy is not relevant to Responder

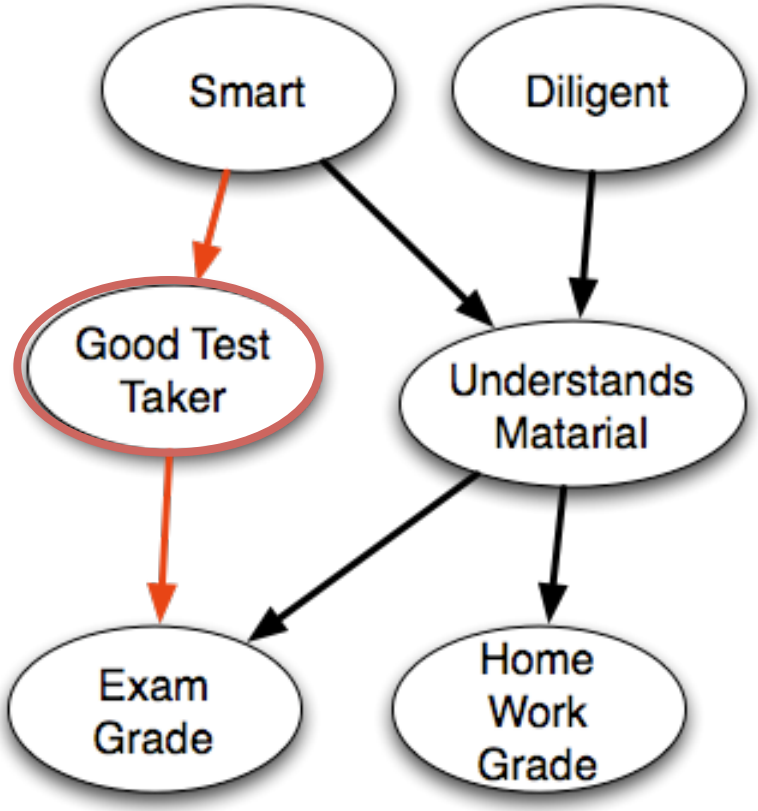
S-Reachability: Graphical Criterion for Relevance

- Decision nodes D_j, D_i
- Informational parents for D_j, denoted ***Pa(Dj)***.
- Utility node for agent that owns *Dj*, denoted *U(Dj)*.



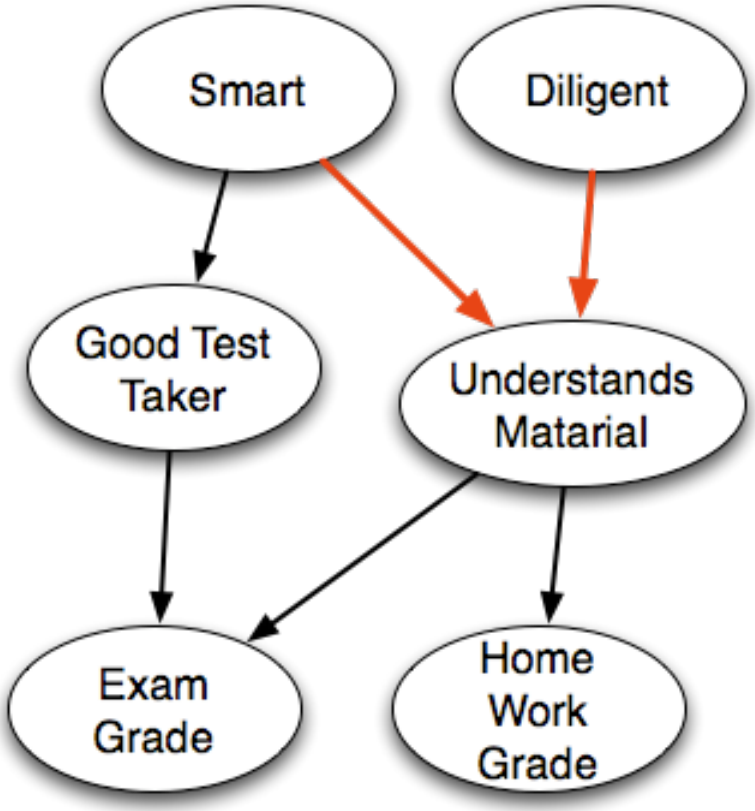
Concept: Blocked Paths

- A path is *blocked* at node *X* if reasoning cannot continue through *X.* Occurs if one of the following holds:
  - The path does not have converging arrows at *X*, and *X* is observed.
  - The path has converging arrows at *X*, and neither *X* nor any of its descendants is observed.



Blocked Paths

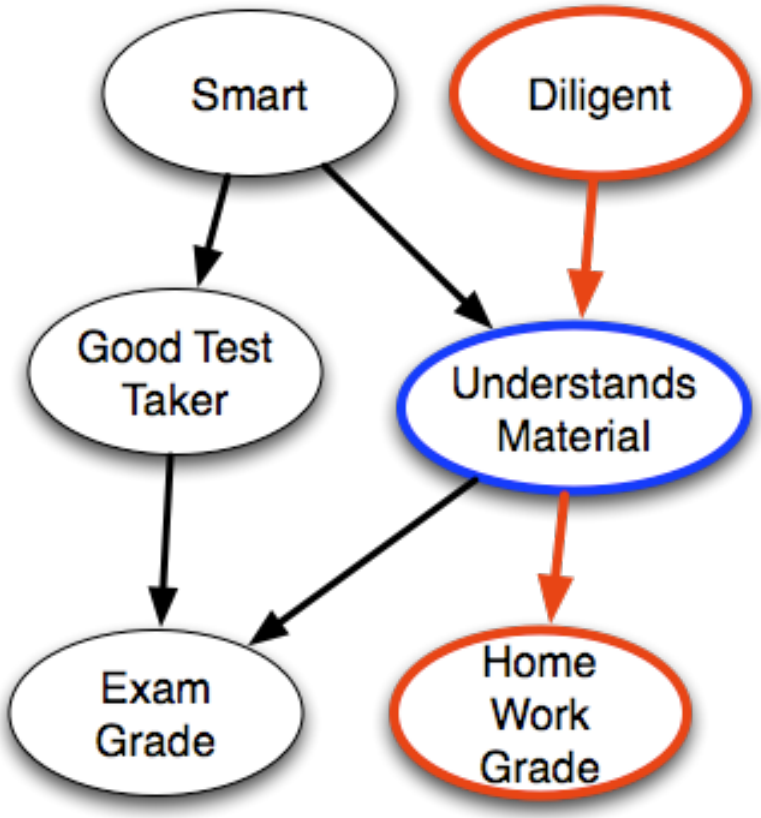
- A path is *blocked* at node *X* if reasoning cannot continue through *X.* Occurs if one of the following holds:
  - The path does not have converging arrows at *X*, and *X* is observed.
  - The path has converging arrows at *X*, and neither *X* nor any of its descendants is observed.



Blocked Paths

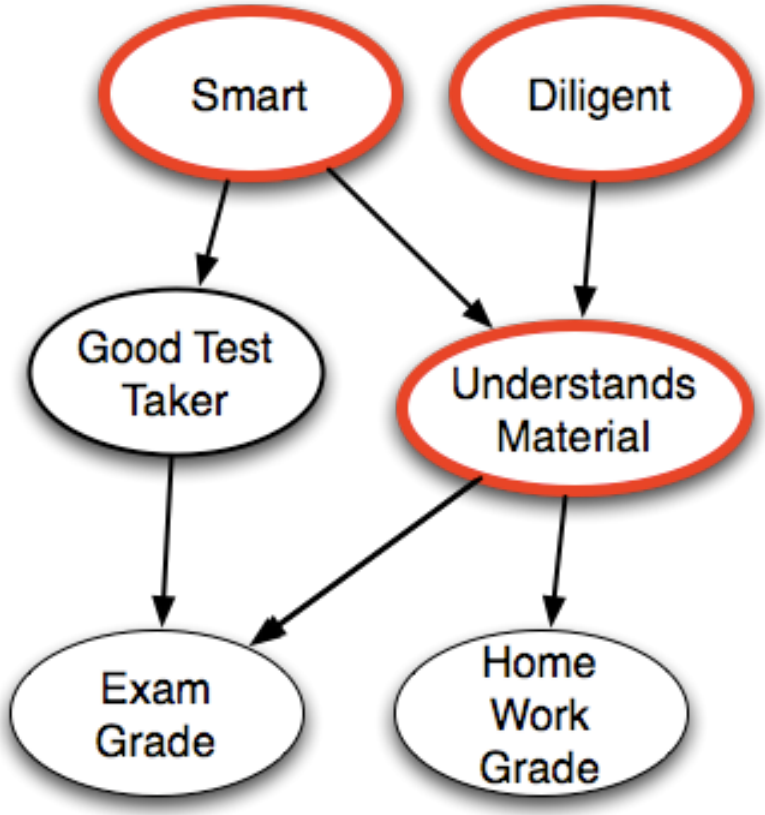
Example:

- Path *D,U,H* Given *U* is blocked at *U* (non converging arrows)



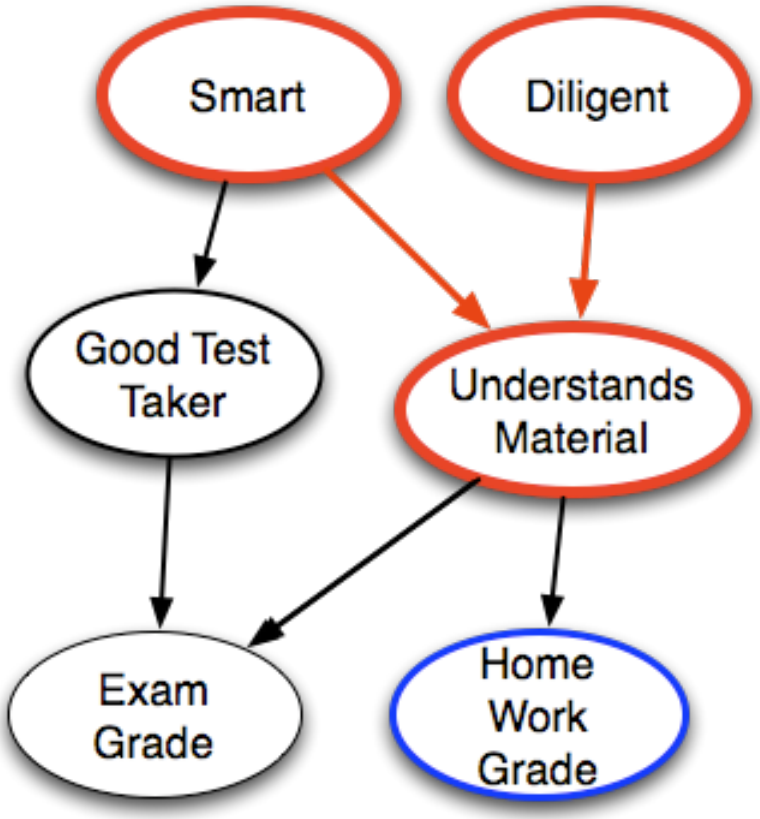
Blocked Paths

- Example:
- Path *D,U,H* Given *U* is blocked at *U* (non converging arrows)
- Path *S,U,D* given no nodes is blocked at *U*. (converging arrows)



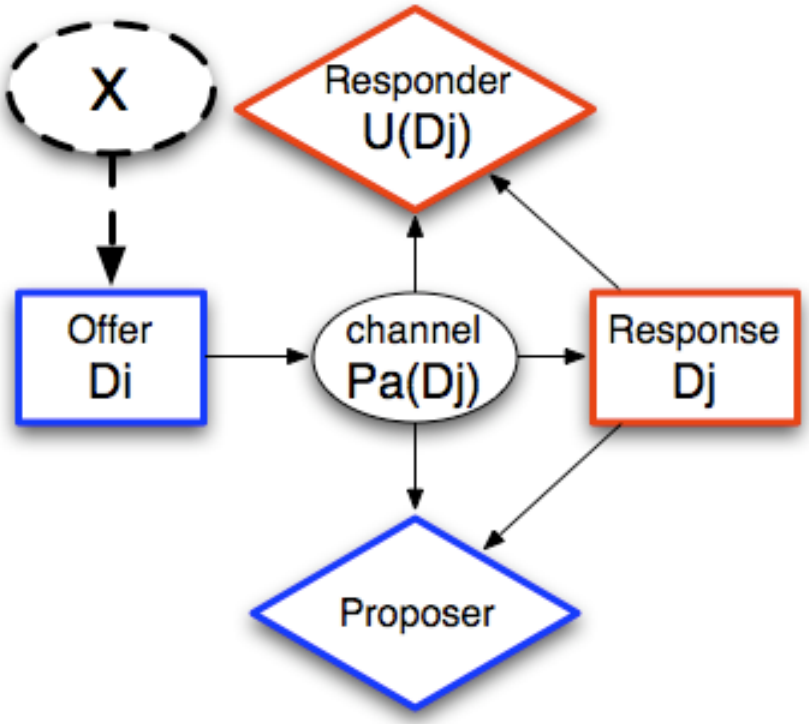
Blocked Paths

- Example:
- Path *D,U,H* is blocked at *U* (non converging arrows)
- Path *S, U, D* given no nodes is blocked at *U*. (converging arrows)
- Path *S, U, D* Given *H* is *not* blocked at *U*.



S-Reachability: Graphical Criterion for Relevance

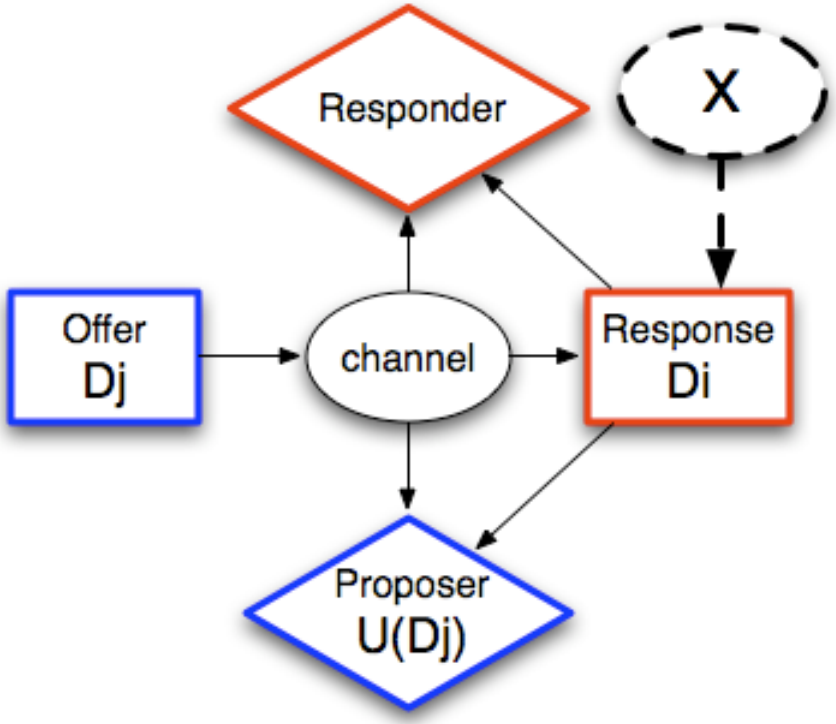
- A decision D_i is **S-Reachable** from D_j if
  - upon adding a new informational parent X to Di.
  - the path from X to $U(Dj)$ is *not* blocked given $\boldsymbol{Pa(D_j)}$, and D_j.



The parent of D_j is the channel.
The path from X to Responder is blocked by channel.
So, Offer is **not** S-reachable from Response.

S-Reachability: Graphical Criterion for Relevance

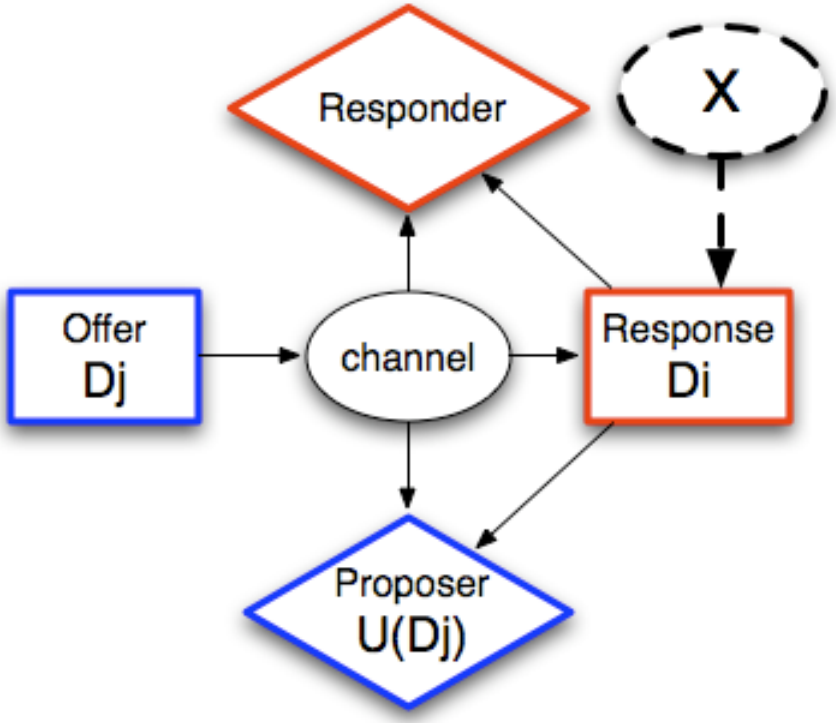
- A decision D_i is **S-Reachable** from D_j if
  - add a new informational parent X.
  - the path from X to $U(Dj)$ is *not* blocked given ***Pa(Dj)***, and D_j.



Offer does not have informational parents.
The path from X to Proposer is not blocked by Offer.
So, Response is S-Reachable from Offer.

S-Reachability is a Sound Criteria for Relevance

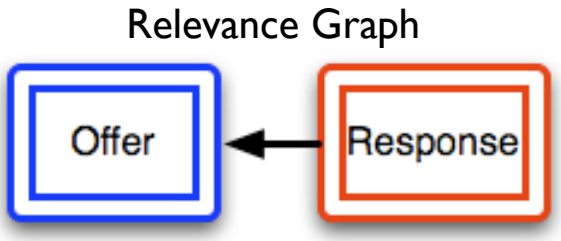
- A decision D_i is **S-Reachable** from D_j if
  - add a new informational parent X.
  - the path from X to $U(Dj)$ is *not* blocked given $\boldsymbol{Pa(D_j)}$, and D_j.

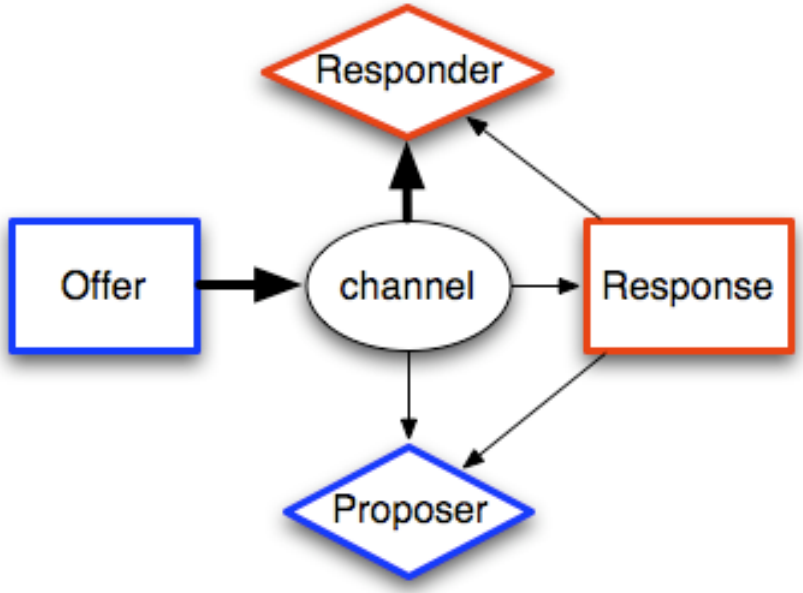


D_i is strategically relevant to D_j then ⟶ D_i is S-reachable from D_j.

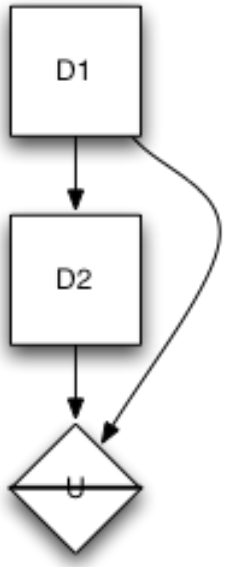
Relevance Graph

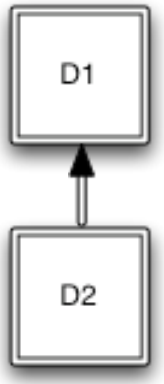
Nodes represent decisions; an edge from D1 to D2 means that D1 is beneficial to know for D2

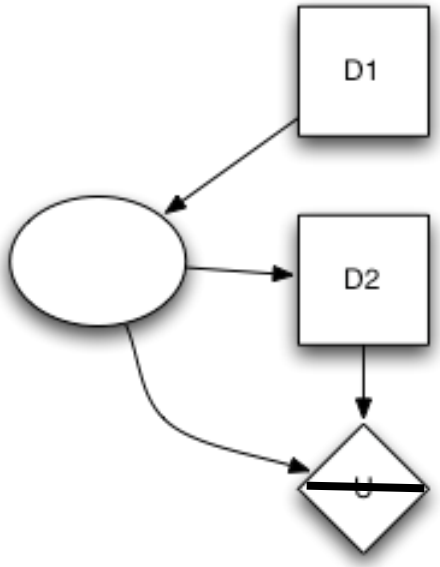


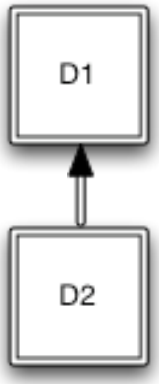


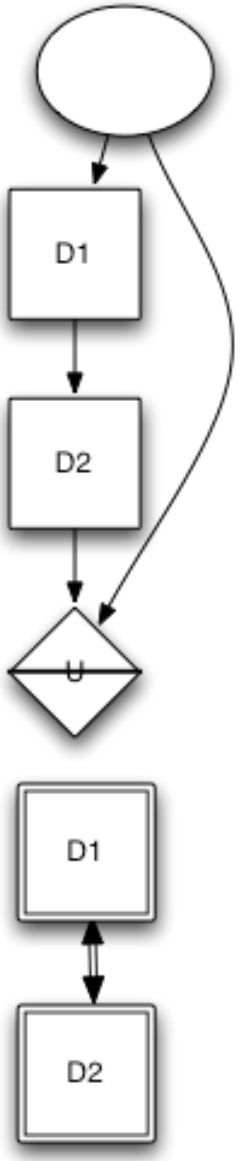
Examples





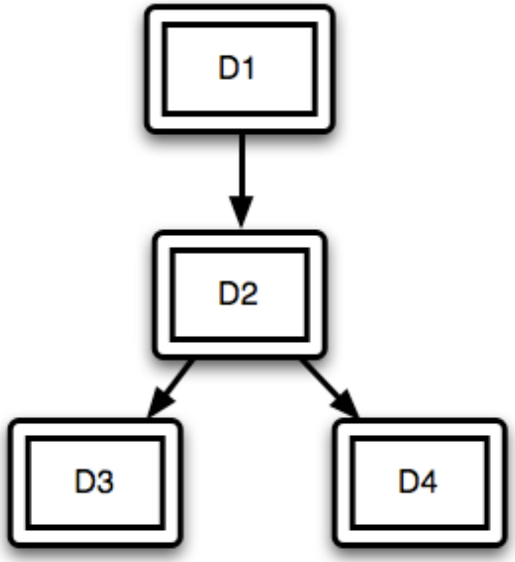






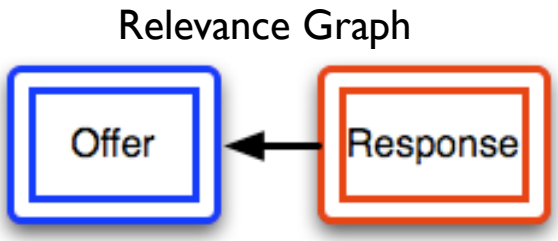
MAID Algorithm: Acyclic Relevance Graph

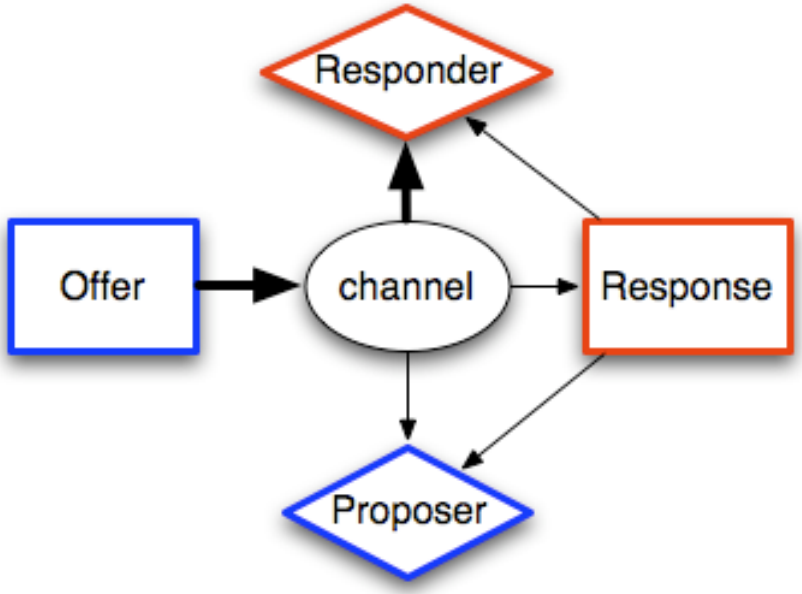
- Traverse the decisions by their topological order in the relevance graph.
- All decisions that are not relevant to current decision can be implemented by chance nodes with uniform prob. distr.
- Best-response strategies for decisions that are relevant to current decision already exist, and are implemented as chance nodes.
- Implement all utility nodes for these decisions as chance nodes.
- MAID is now ID. Can solve ID and extract the best-response strategy for the current decision.



MAID Algorithm: Acyclic Relevance Graph

- **Step 1: Solve Response**
- Begin by implementing Offer with a uniform strategy.
- The result is an ID.
- Solve ID and determine strategy for Response: (i.e., accept any split larger than zero)

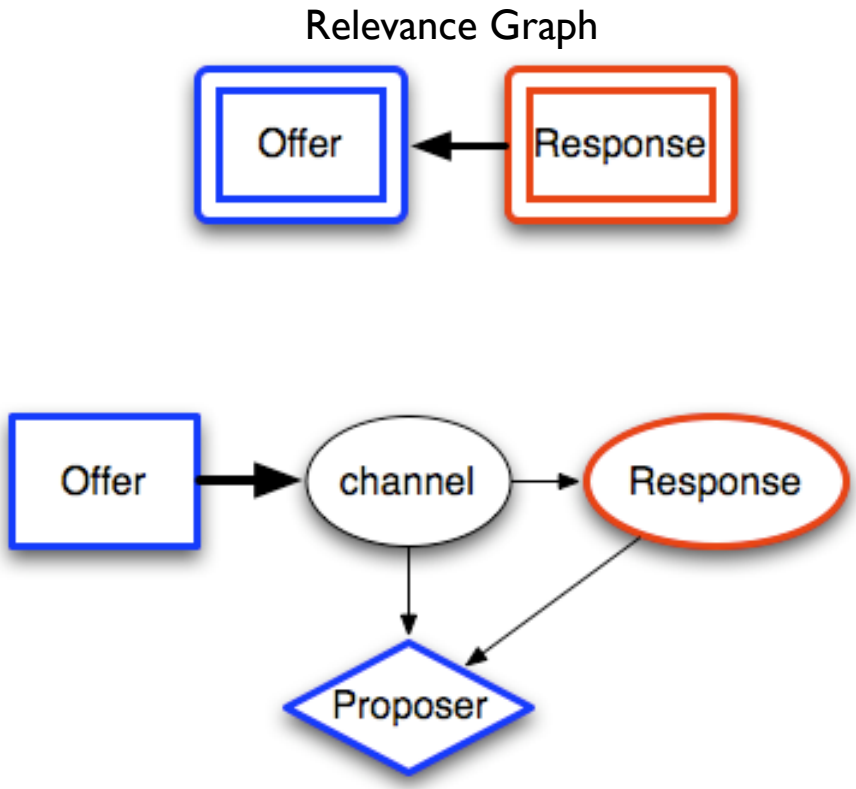




MAID Algorithm: Acyclic Relevance Graph

- **Step 2: Solve Offer**
- Responder strategy is set.
- Solve ID and determine strategy for Offer. (i.e., propose the largest split for proposer that gets accepted by responder)
- Output NE for MAID:

Offer is the largest split for proposer that offers a positive share to responder; responder accepts the offer.



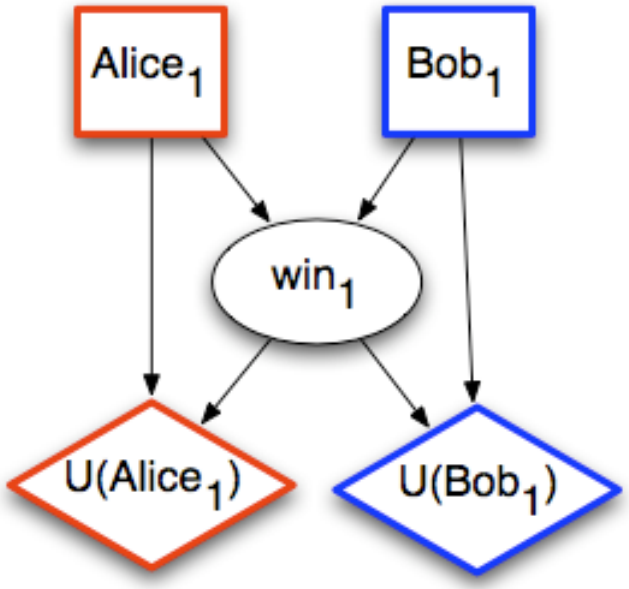
Valuations Example

- Bob and Alice participate in *n* consecutive sealed-bid auctions, each for an individual item. An agent can bid "high" or "low" at each auction.

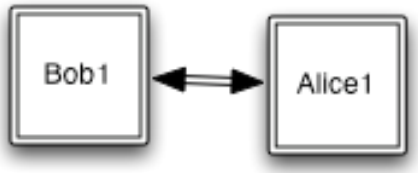
- The highest bidder wins auction if it did not win previous auction. Otherwise, the highest bidder wins auction with probability 0.6. Ties are broken randomly.

- Agents receive reward after each individual auction. The second item is more valuable then the first.

Valuations Example (*n*=1): MAID

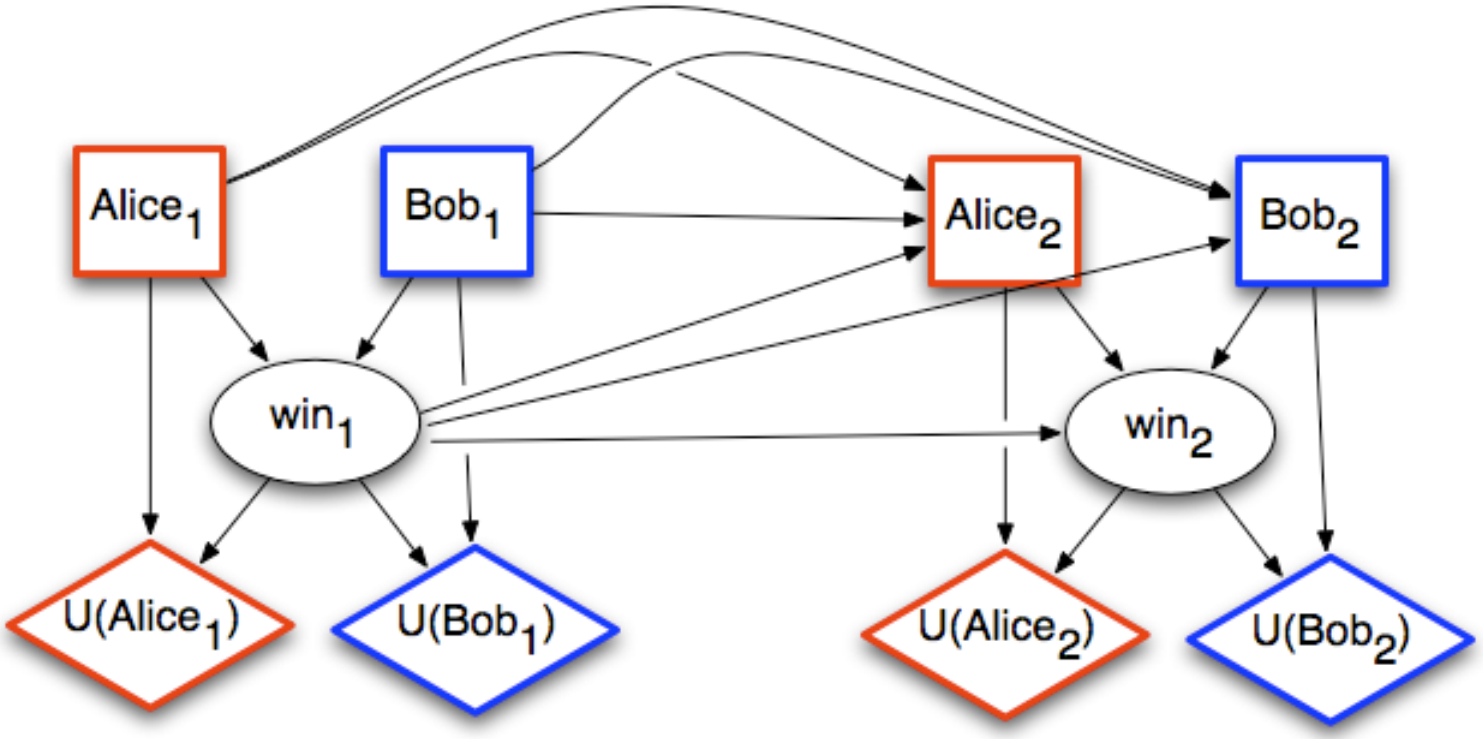


relevance graph



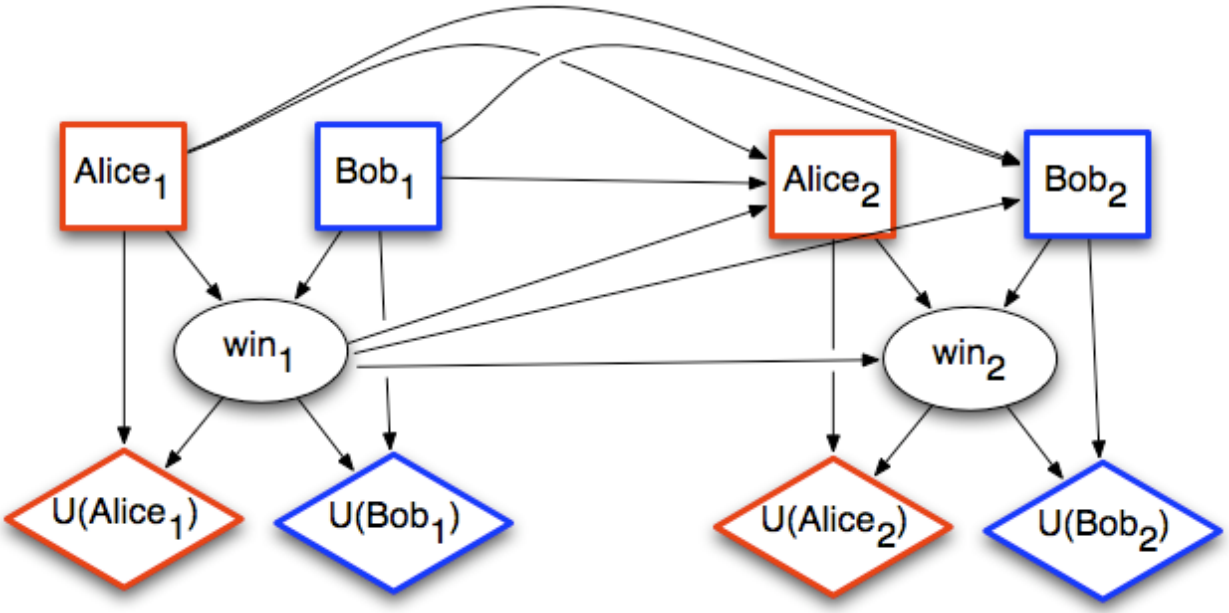
Valuations MAID *(n=2)*

- Alice and Bob are interacting for two rounds. Agents observe the history of past bids.



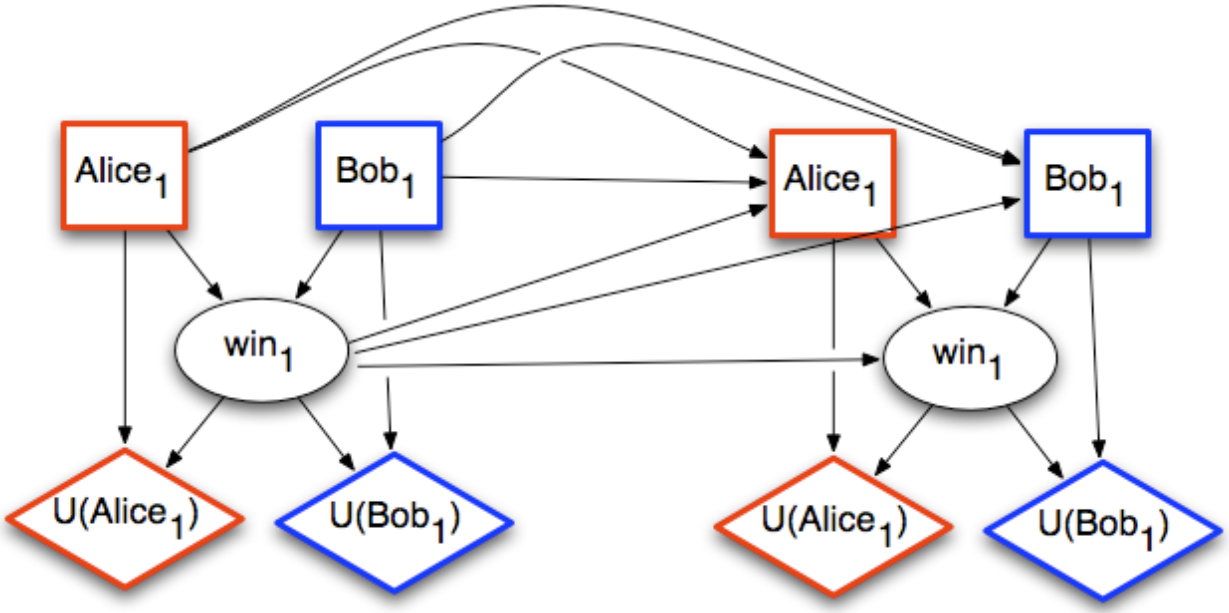
Valuations MAID

- In round 2, Bob can observe Alice's past bid. He doesn't care about the strategy that generated that bid.
  - Past decisions are not strategically relevant to future decisions.

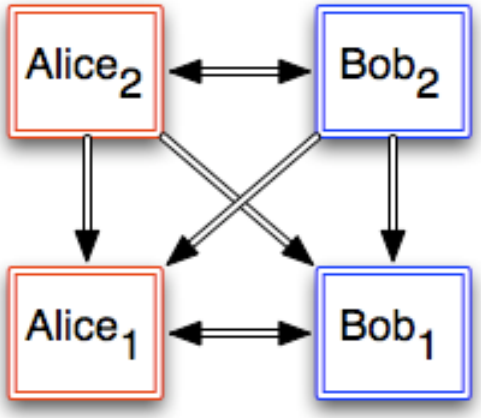


Valuations MAID

- If Alice knows that Bob will bid low for the second item, she should bid low for the first item to increase her chances of winning the second item if she bids high.
  - Future decisions may be strategically relevant to current decisions.

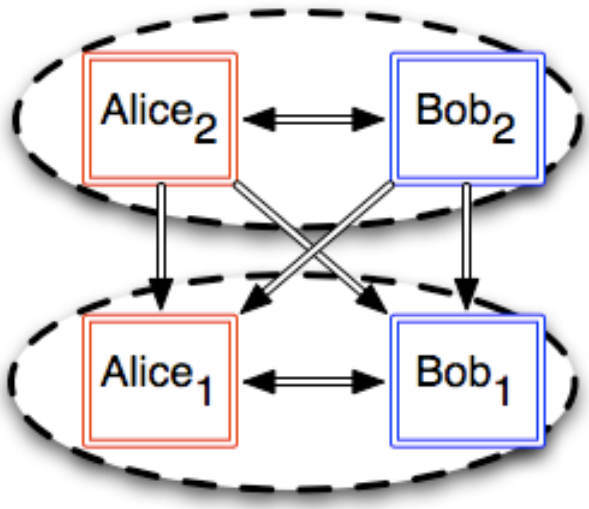


Relevance Graph for Valuations Game



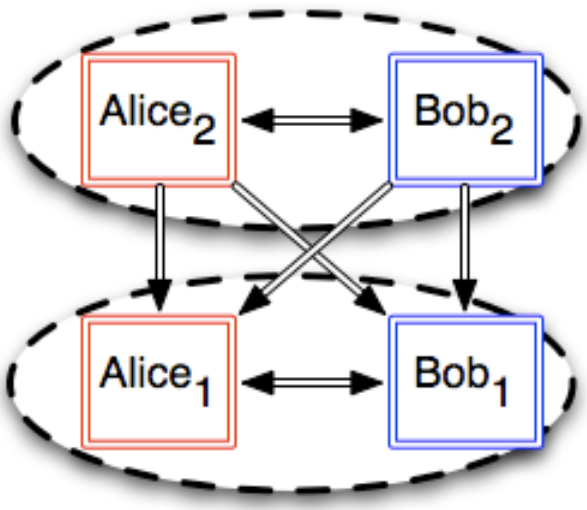
Strongly Connected Components

- A graph is **strongly connected** if there is a path from each node to every other node. The **strongly connected components (SCC)** of a graph are its largest strongly connected sub-graphs.



Solving Cyclic MAIDs

- Iterate backwards on SCCs of relevance graph. For each component
  - Create a MAID that implements uniform strategies for decisions in earlier SCCs.
  - Obtain Nash equilibrium strategy profile for decisions in the component by constructing an extensive form game.
  - The game will only split on the decisions in the current SCC.
  - **May be much smaller than the tree for the full MAID and easier to solve – savings can be exponential**



Reasoning Patterns

- Informally, a reasoning pattern is a form of argument that leads to and explains a decision
  - e.g.
    - modus ponens in logic
    - explaining away in Bayes nets

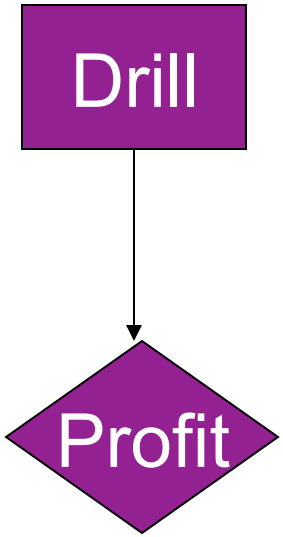
- **What reasoning patterns can agents use in games?**

Characterization of Reasoning Patterns

- Four basic reasoning patterns, each characterized by paths in a MAID

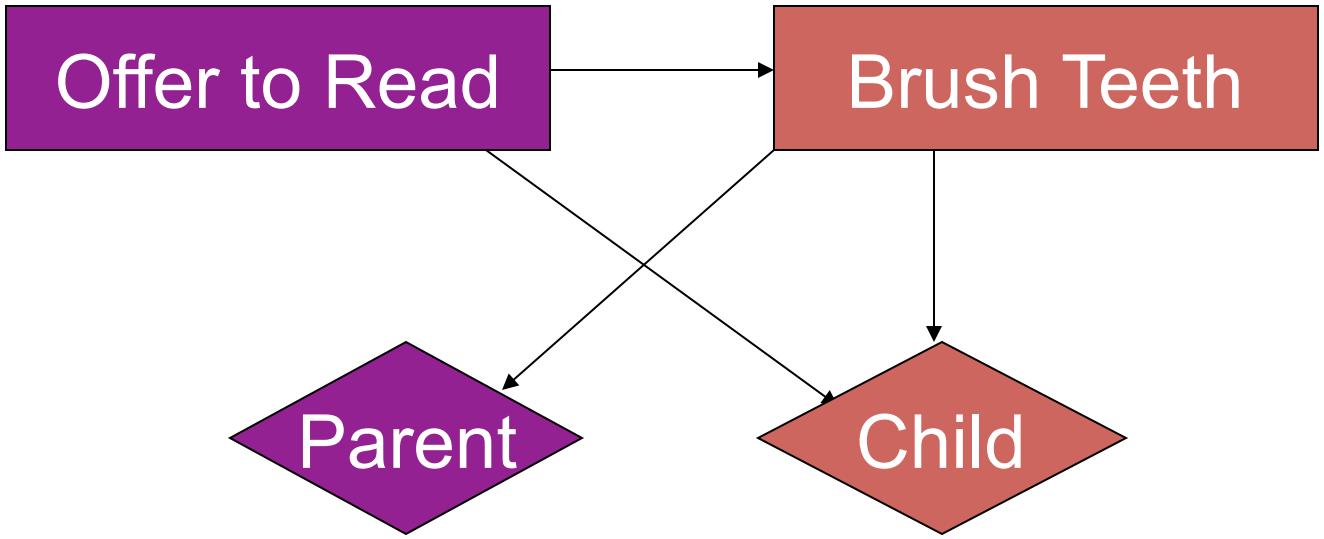
- Characterization based on graphical criteria only
  - could further refine characterization based on numerical parameters

Reasoning Pattern #1: Direct Effect



- An agent takes a decision because of its direct effect on its utility
  - without being mediated by other agents' actions

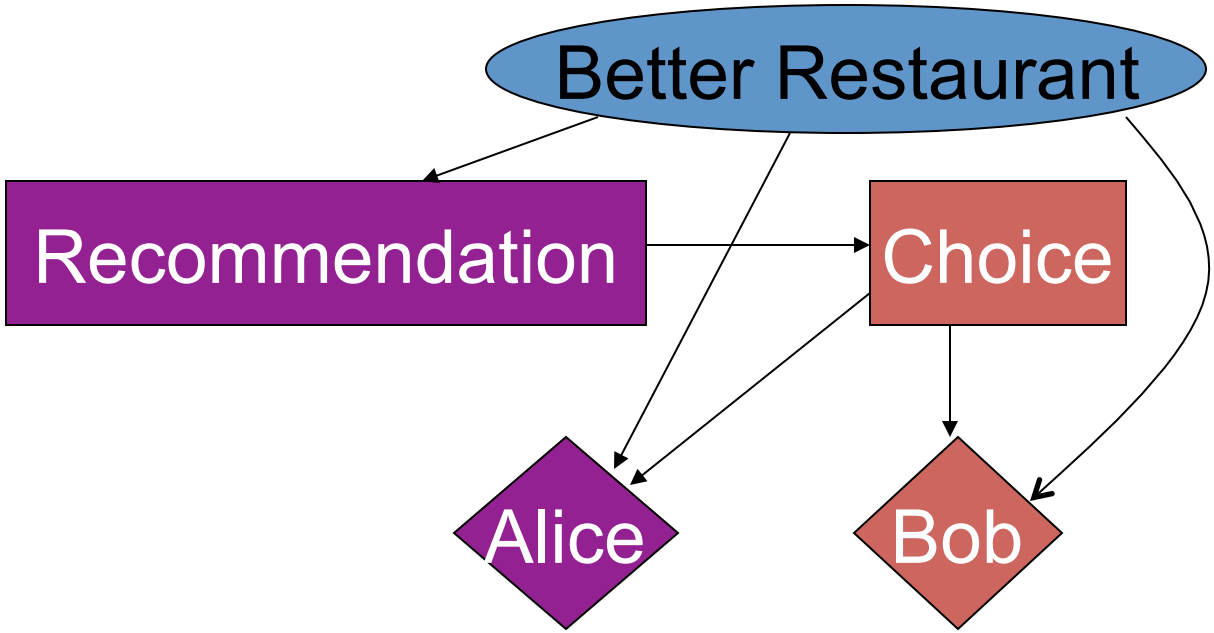
Reasoning Pattern #2: Manipulation



- Child knows about parent's action
- Parent does not care about reading, but wants child to brush teeth
- Child dislikes brushing teeth but likes being read to

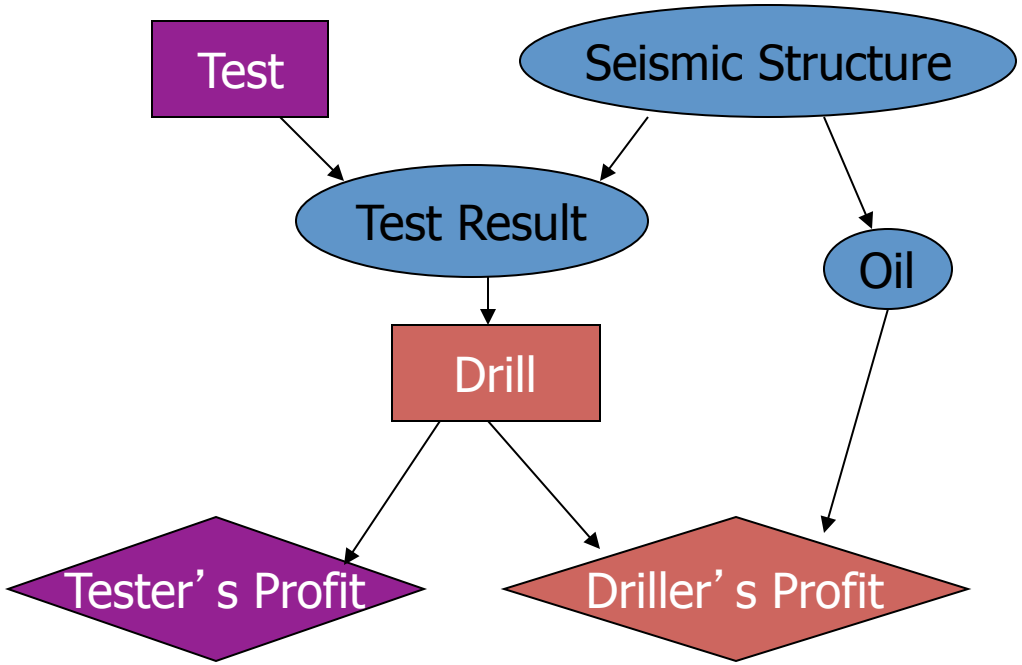
⇒ Parent can manipulate child

Reasoning Pattern #3: Signaling



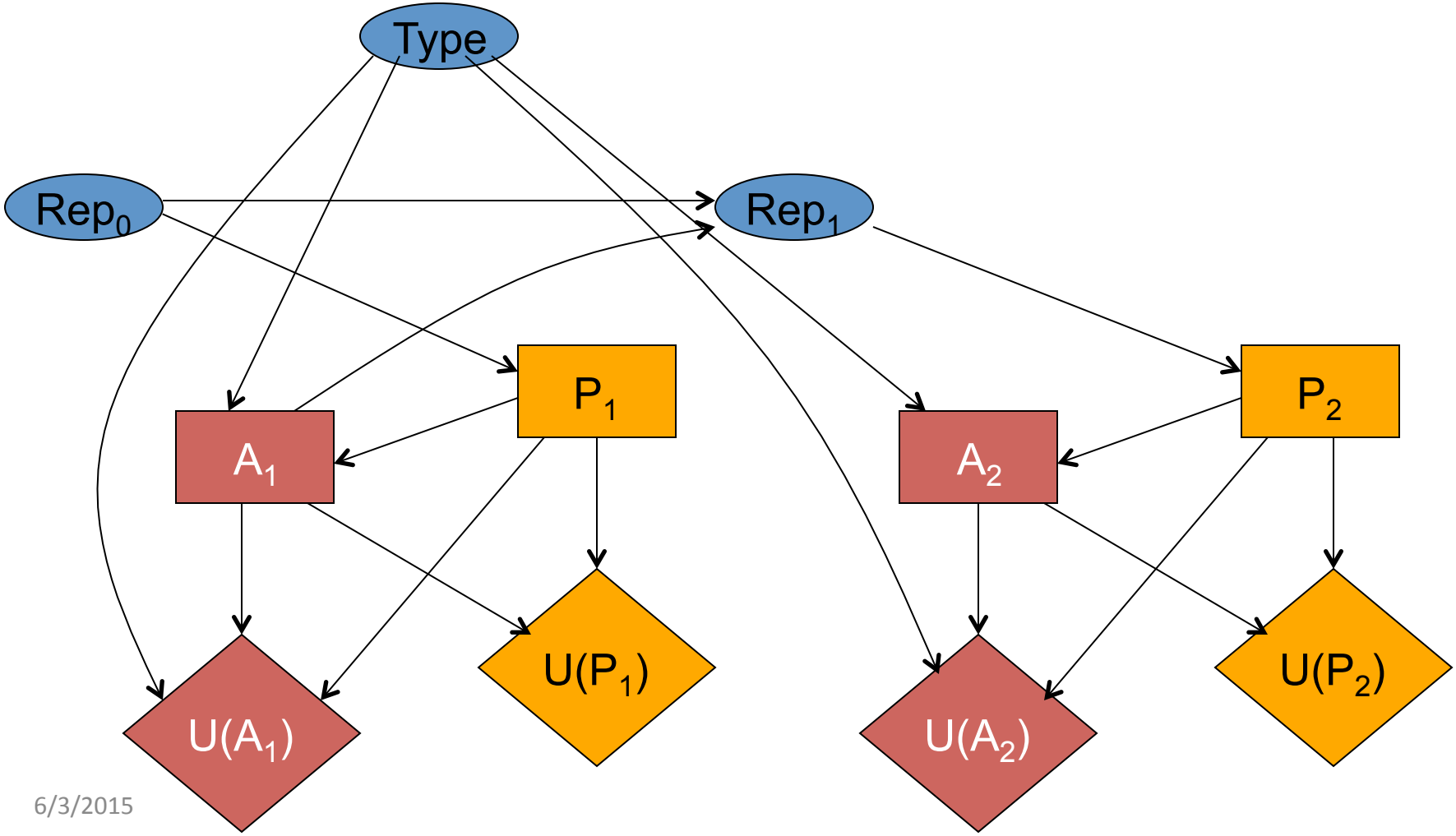
- A communicates something that she knows to B, thus influencing B's behavior

Reasoning Pattern #4: Revealing/Denying

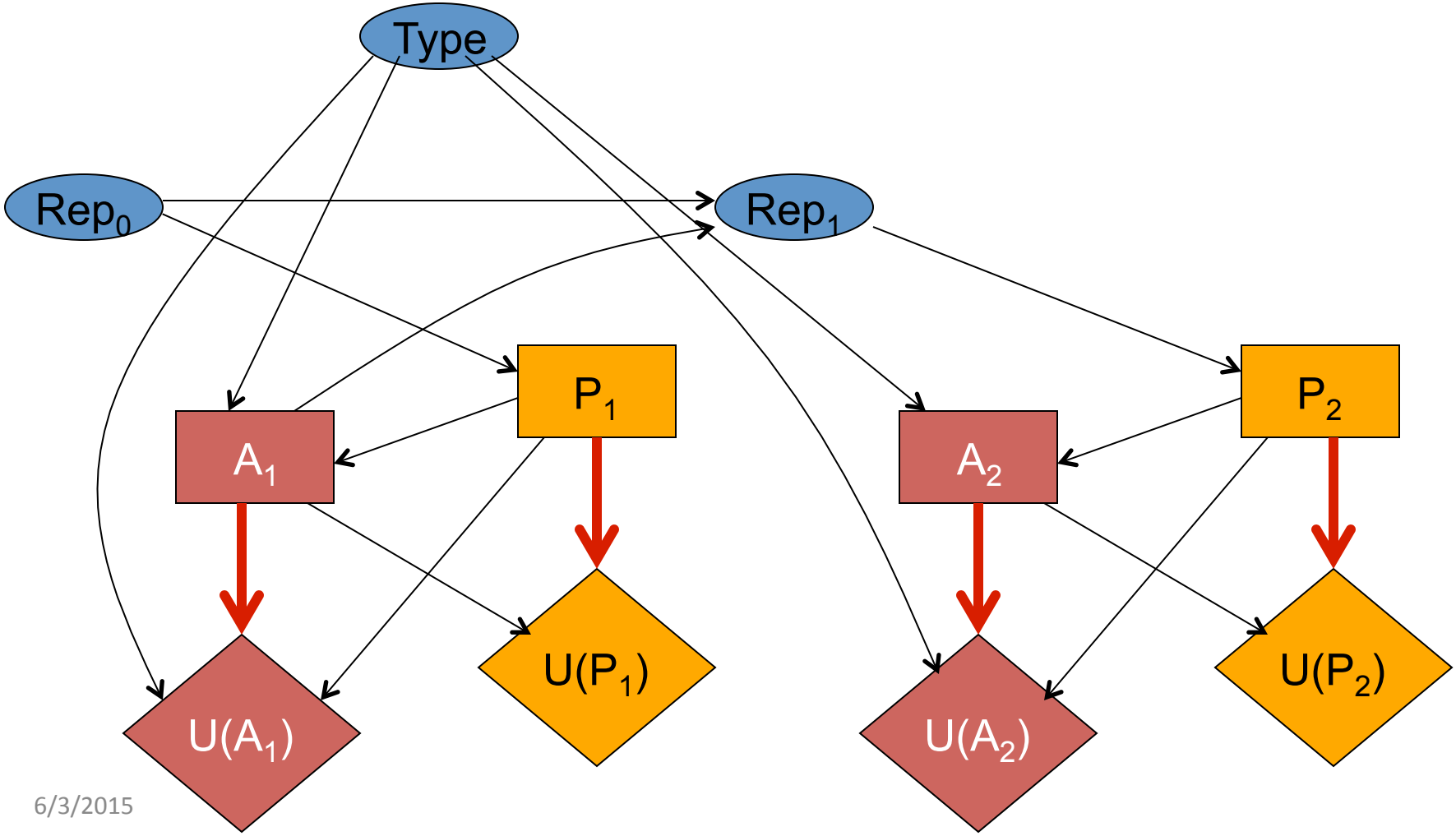


- Driller cares about oil
- Tester receives fee if driller drills
- Tester causes driller to find out (or not) about information tester herself does *not* know

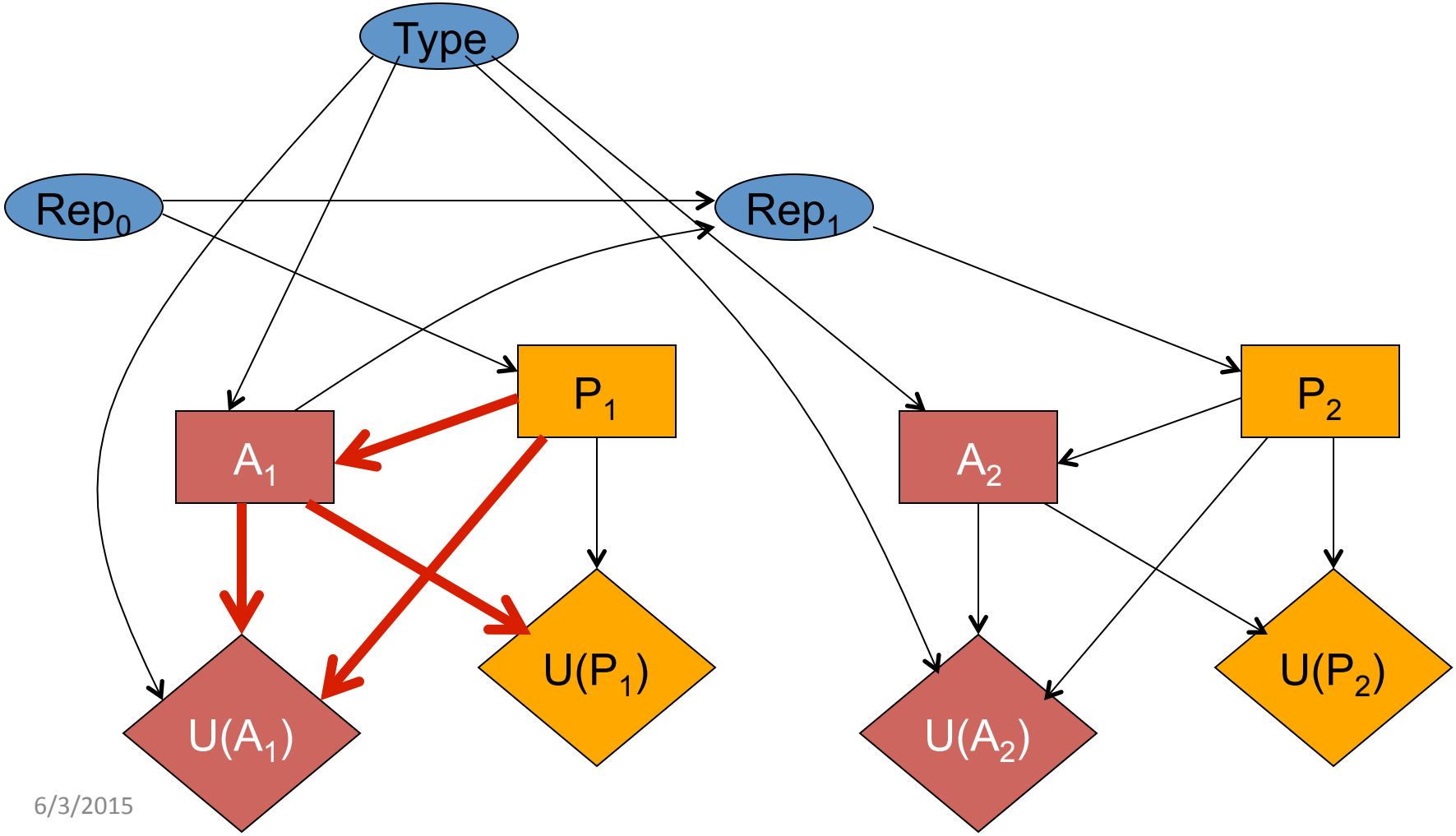
Example: Two Stage Principal-Agent Game



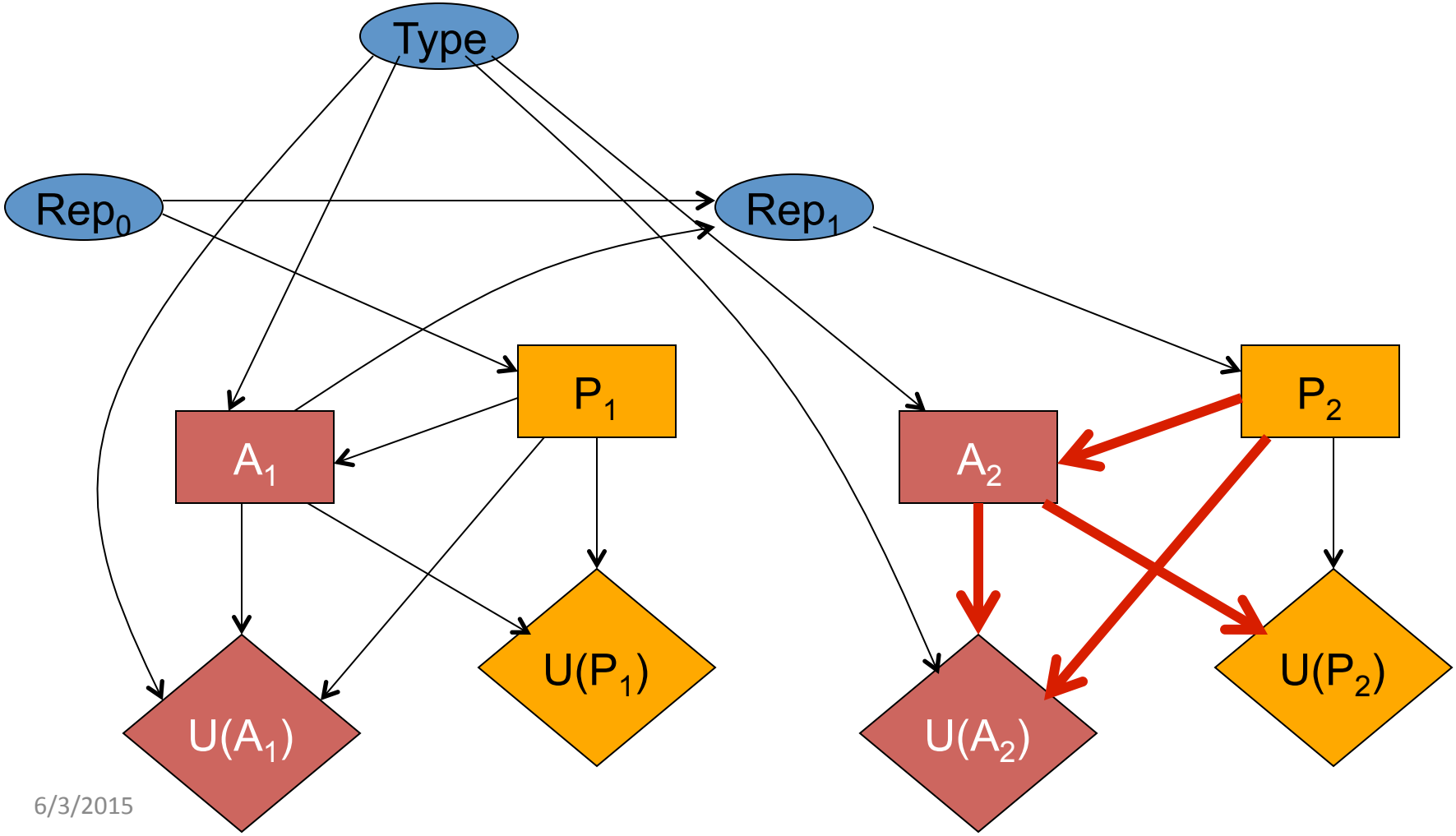
Direct Effect For All Four Decisions



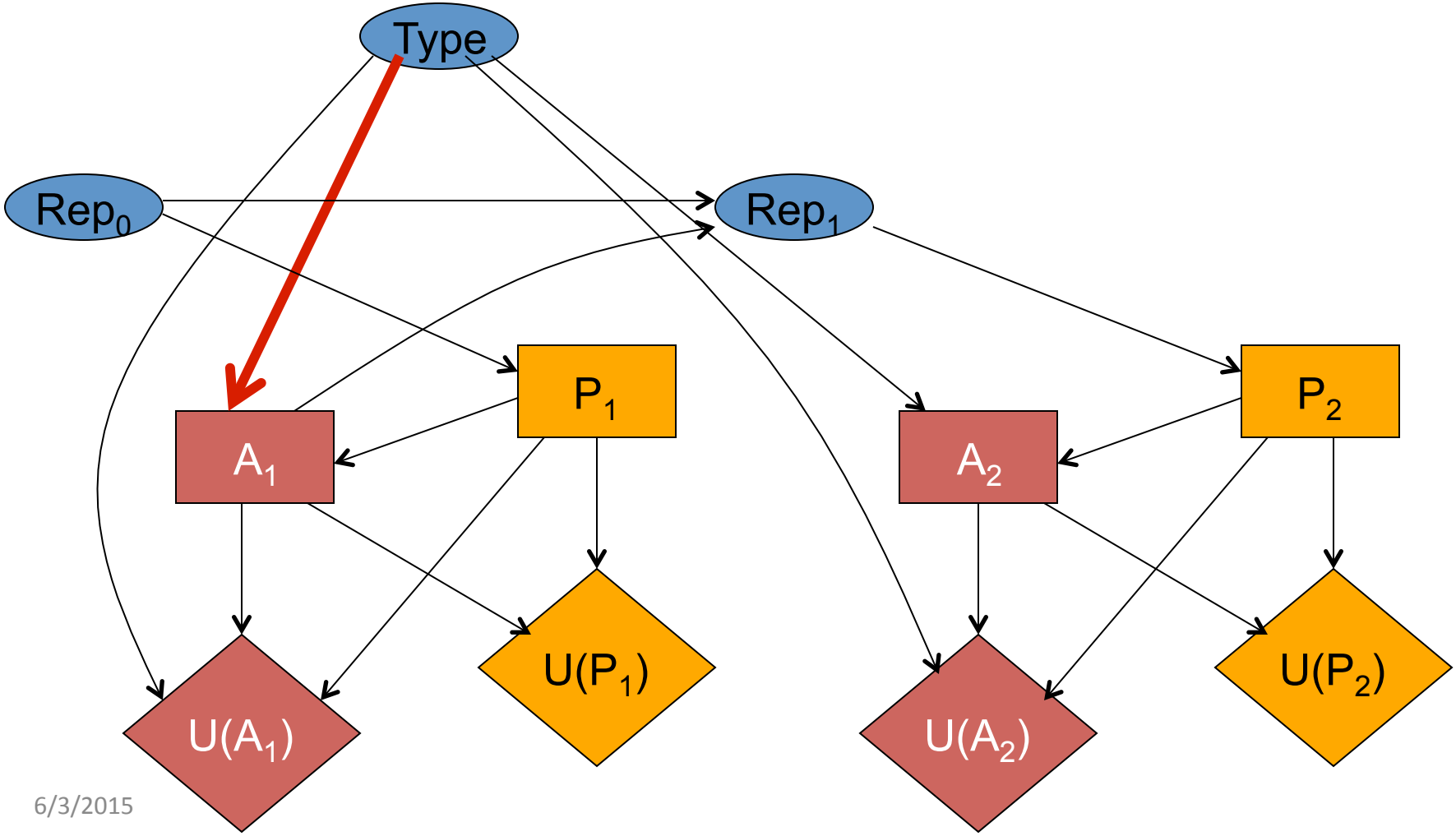
Manipulation ($P_1 \rightarrow A_1$)



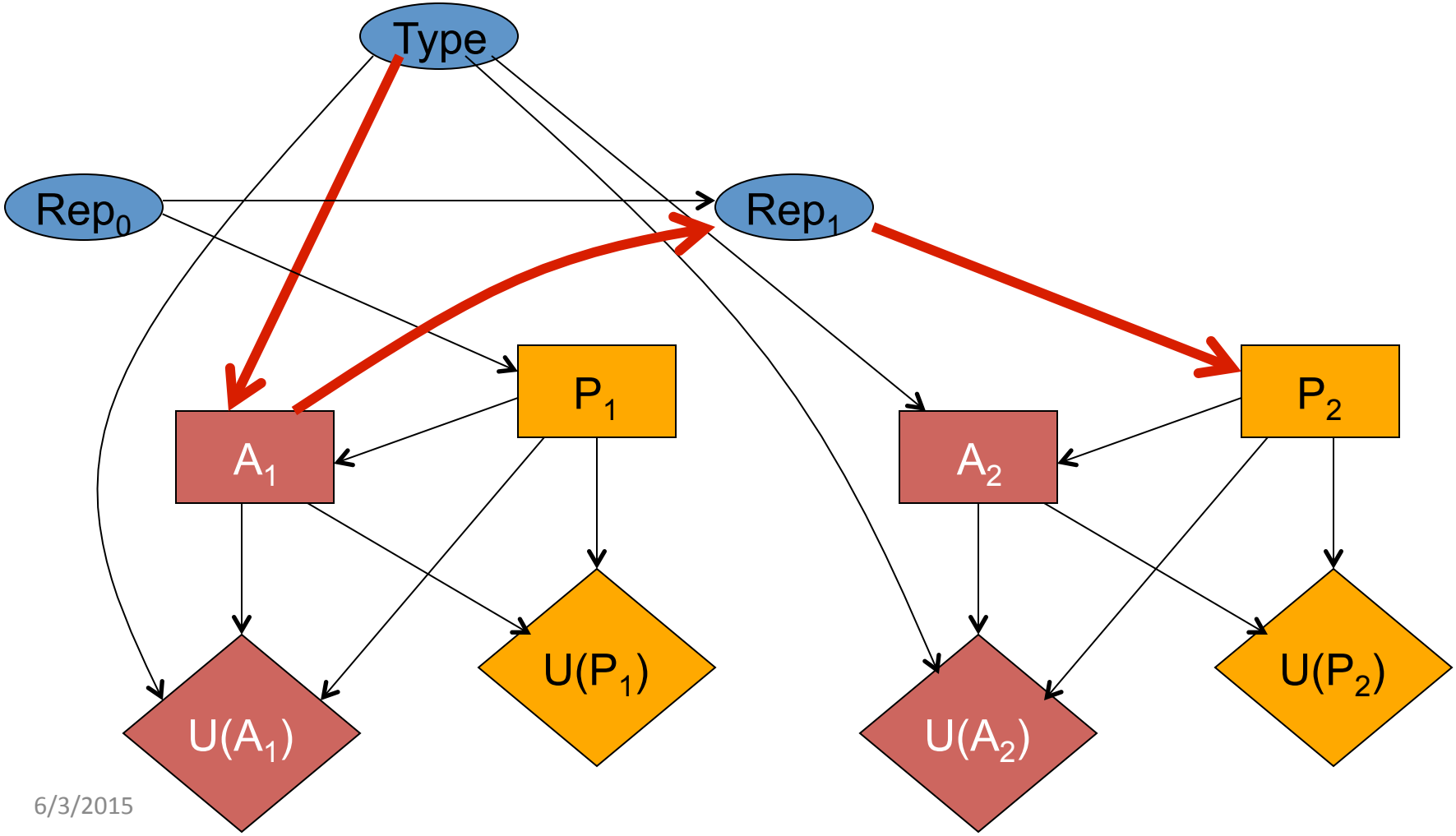
Manipulation ($P_2 \rightarrow A_2$)



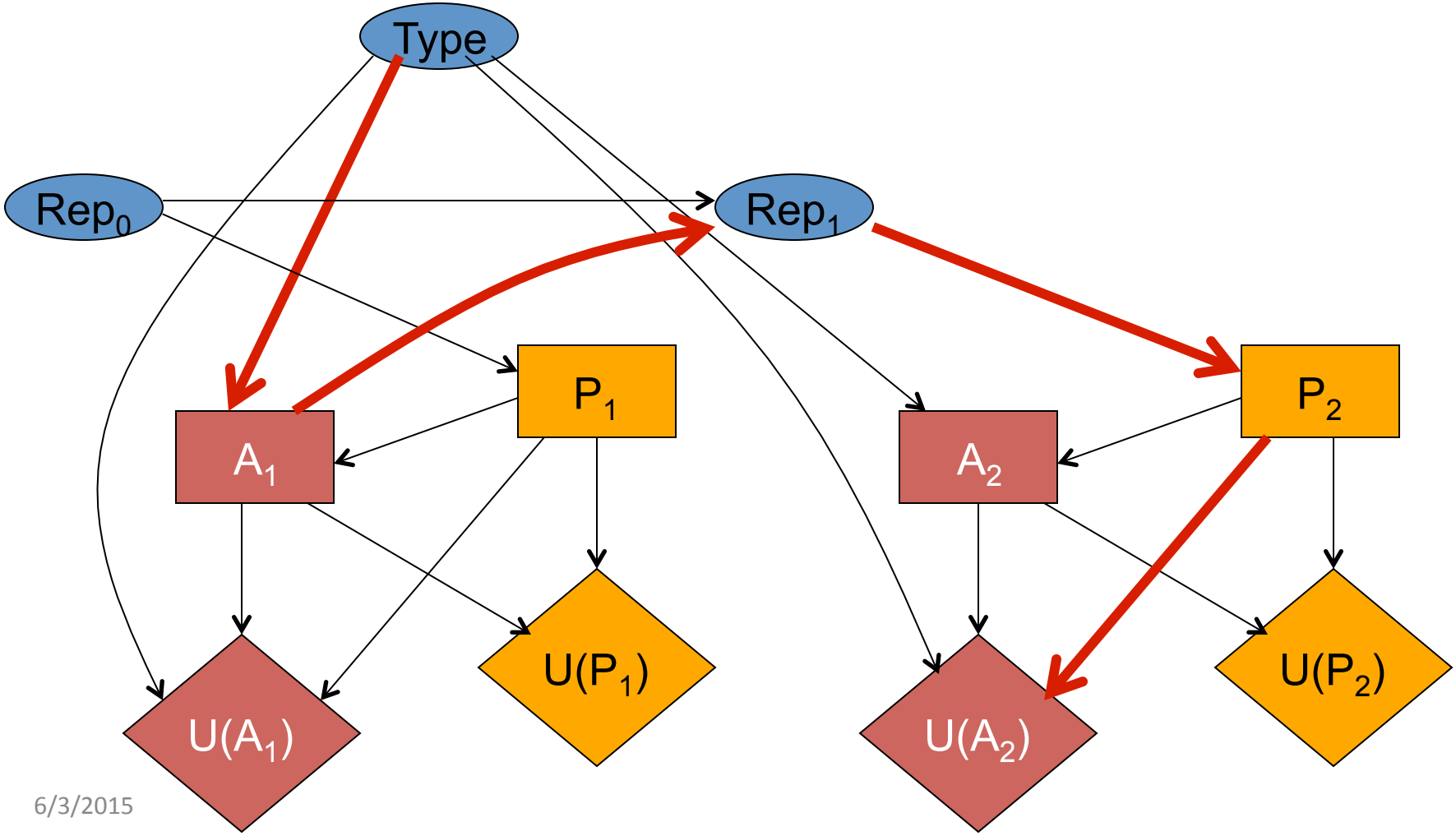
Signaling (A_1 signals Type to P_2)



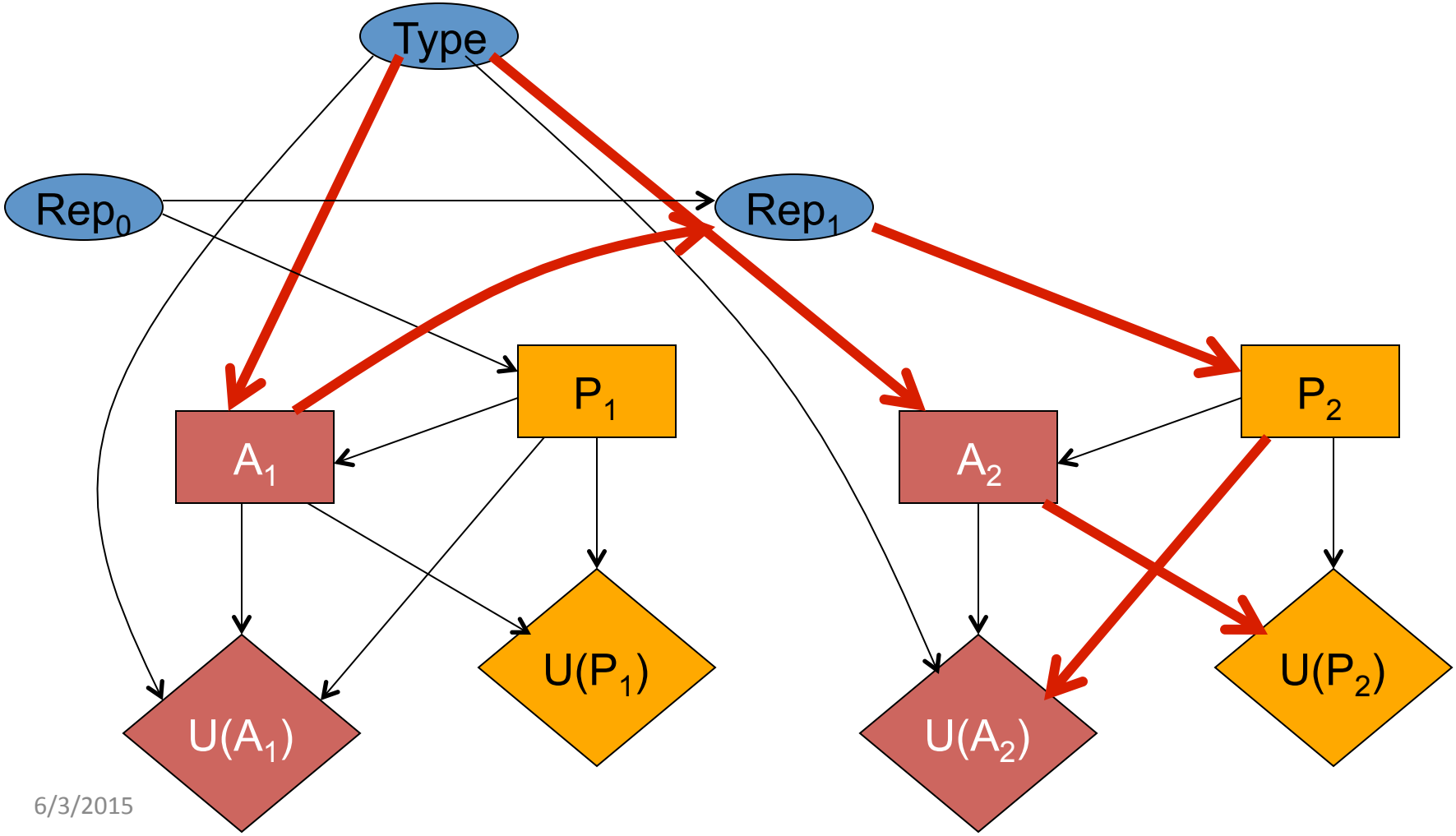
Signaling (A_1 signals Type to P_2)



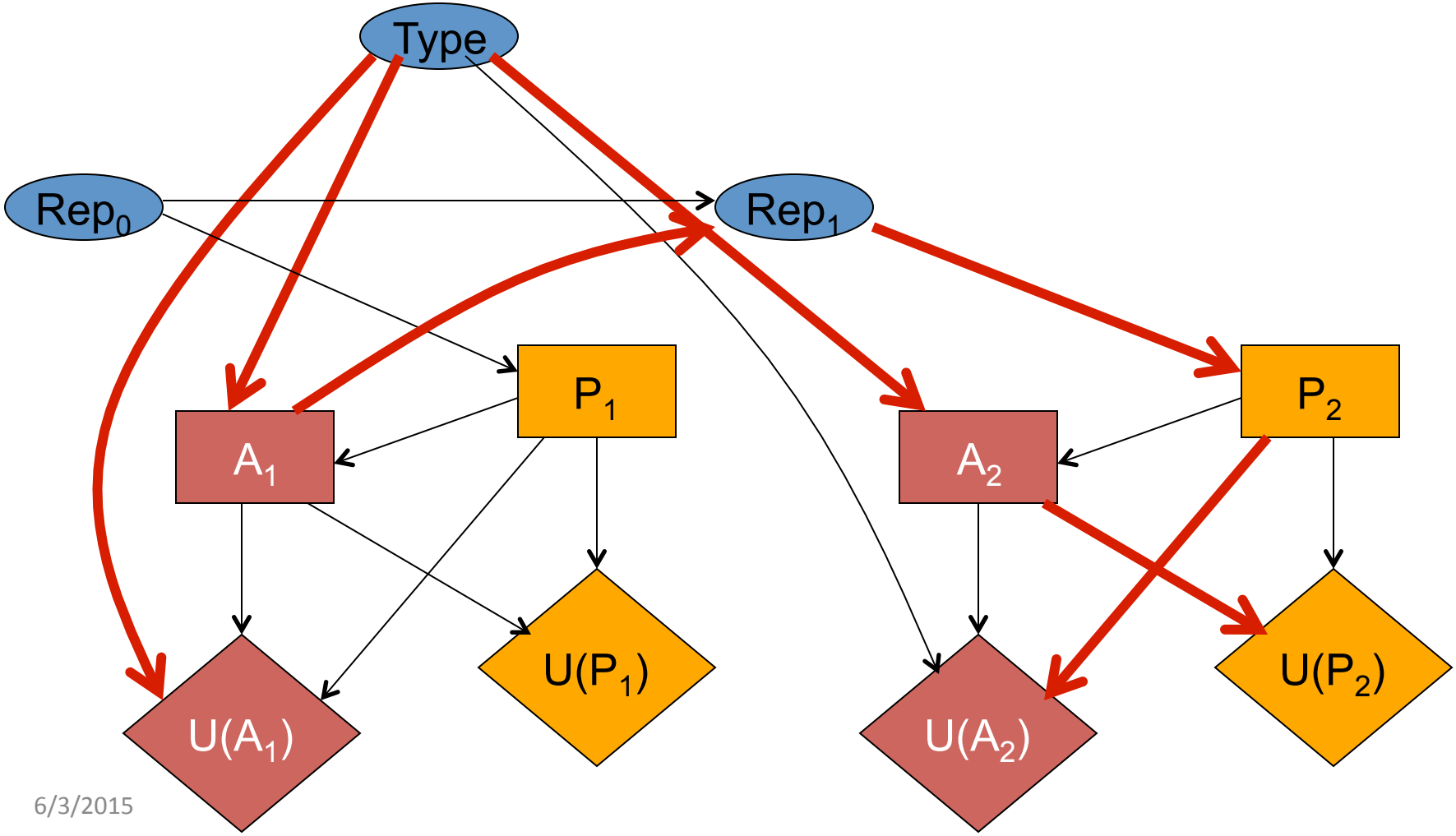
Signaling (A_1 signals Type to P_2)



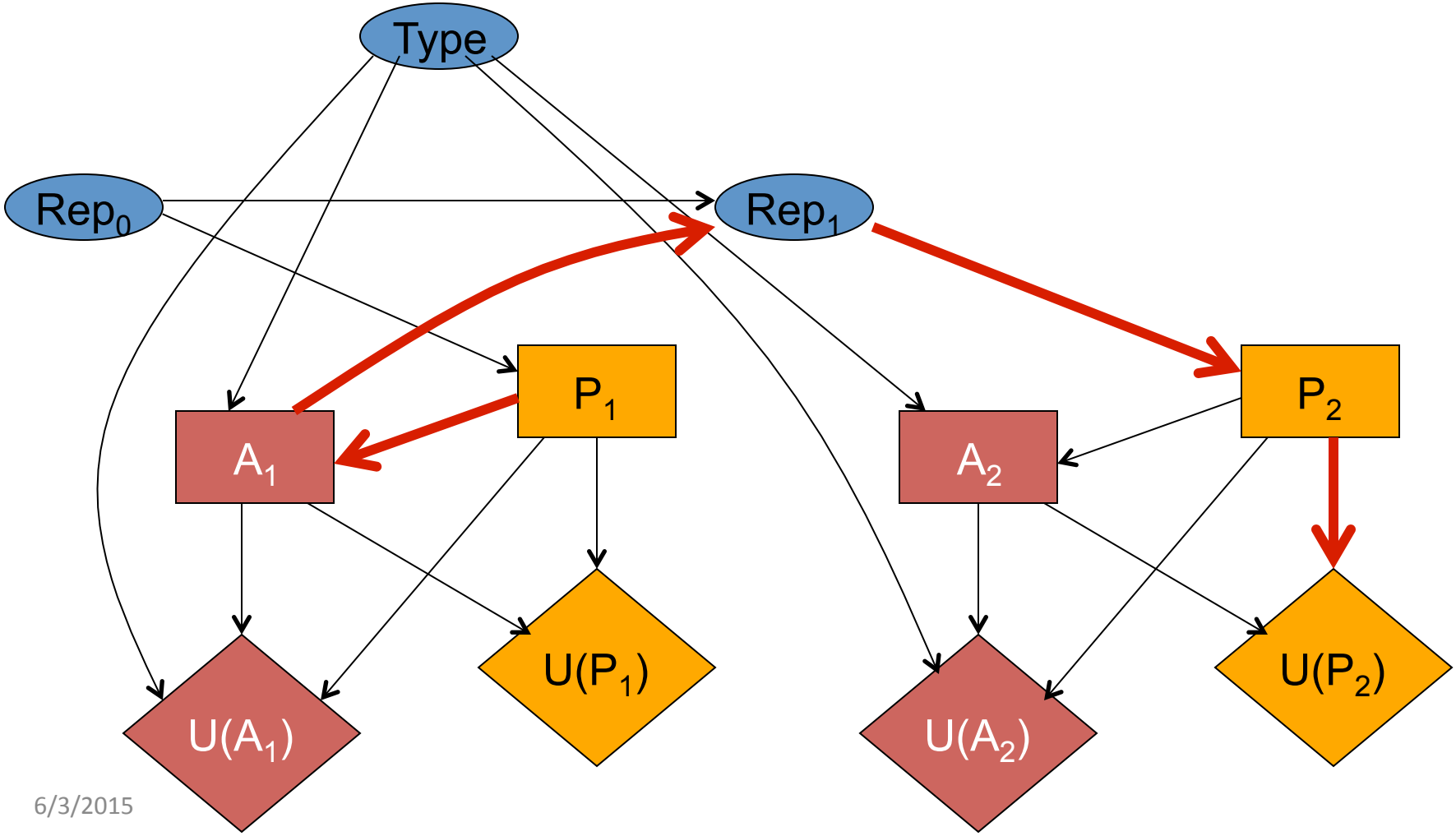
Signaling (A_1 signals Type to P_2)



Signaling (A_1 signals Type to P_2)



Revealing/Denying (P_1 reveals Type to P_2)



Revealing/Denying (P_1 reveals Type to P_2)