# Causal Graph Justifications of Stable Models

Jorge Fandinno[1]

Joint work with:
Pedro Cabalar[1] and Michael Fink[2]

1- University of Corunna, SPAIN

2- Vienna University of Technology, AUSTRIA

Logical Reasoning and computation
Corunna, Spain

February 24th, 2015

# Causality and Knowledge Representation

- For Knowledge Representation, not just deriving conclusions but sometimes we require explanations
- Causality: is a quite common concept in human daily discourse. Present in (chronologically or physically) distant cultures.
- What "$A$ has caused $B$" actually means?

# Causality and Knowledge Representation

- For Knowledge Representation, not just deriving conclusions but sometimes we require explanations
- Causality: is a quite common concept in human daily discourse. Present in (chronologically or physically) distant cultures.
- What "*A* has caused *B*" actually means?
  - Sufficient cause
  - Necessary cause
  - Actual or contributory cause

# Joint interaction

## Example

- There is a law asserts that *driving drunk* is *punishable*.
- Suppose that some person drove drunk.

Take the logic program consisting of one rule and two labelled facts

$$punish \leftarrow drive,\ drunk \qquad \mathtt{d} : drive \qquad \mathtt{k} : drunk$$

- Joint interaction of multiple events.
  The cause formed by "$\{\mathtt{d}, \mathtt{k}\}$ together has caused *punish*".

# Joint interaction

## Example

- There is a law asserts that *driving drunk* is *punishable*.
- Suppose that some person drove drunk.

Take the logic program consisting of one rule and two labelled facts

$$punish \leftarrow drive,\ drunk \qquad \mathtt{d} : drive \qquad \mathtt{k} : drunk$$

- Joint interaction of multiple events.
  The cause formed by "$\{\mathtt{d}, \mathtt{k}\}$ together has caused *punish*".
- Two kinds of causal rules:
  - Unlabelled rules: tracing them is irrelevant for causal purposes.
  - Labelled rules: keep track of possible ways to derive an effect.

# Labels

- We may want to keep track of involved rules and not only facts:

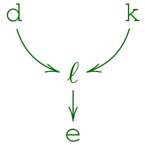
## Example

- Law $\ell$ asserts that *driving drunk* is *punishable* with imprisonment.
- The execution $\mathtt{e}$ of a sentence establishes that people who are *punished* are *imprisoned*.
- Suppose that some person drove drunk.

$$\ell : \textit{punish} \leftarrow \textit{drive}, \ \textit{drunk} \qquad \mathtt{d} : \textit{drive} \qquad \mathtt{k} : \textit{drunk}$$

$$\mathtt{e} : \textit{prison} \leftarrow \textit{punish}$$

- We get a cause in the form of a label graph



# Main ideas

- Multi-valued semantics for logic programs: each true atom will be associated to a set of justifications (causal graphs)
- Accordingly, falsity = lack of justification.
  - This coincides with the informal reading for default negation: *not p* = there is no way to derive *p*

# Main ideas

- Multi-valued semantics for logic programs: each true atom will be associated to a set of justifications (causal graphs)
- Accordingly, falsity = lack of justification.
  - This coincides with the informal reading for default negation: *not p* = there is no way to derive *p*
- Causes must be non-redundant.
  - Some causes will be stronger than others.
  - This allows us defining a lattice and algebraic operations $+$ (alternative causes), $*$ (joint causation) and $\cdot$ (rule application).

# Main ideas

- Multi-valued semantics for logic programs: each true atom will be associated to a set of justifications (causal graphs)
- Accordingly, falsity = lack of justification.
  - This coincides with the informal reading for default negation: *not p* = there is no way to derive *p*
- Causes must be non-redundant.
  - Some causes will be stronger than others.
  - This allows us defining a lattice and algebraic operations $+$ (alternative causes), $*$ (joint causation) and $\cdot$ (rule application).
- Important result: semantically obtained causal values correspond to (non-redundant) syntactic proofs using the program rules!

# Outline

# Causal Graphs

## Definition

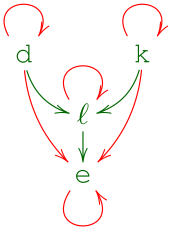
A causal graph $G$ is a transitively and reflexively closed graph of labels.

# Causal Graphs

**Definition**

A causal graph $G$ is a transitively and reflexively closed graph of labels.

In our example, we would actually have

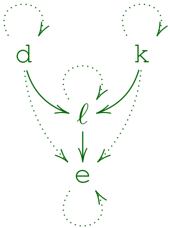


# Causal Graphs

**Definition**

A causal graph $G$ is a transitively and reflexively closed graph of labels.

In our example, we would actually have



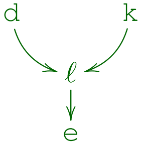
# Causal Graphs

- $G^*$ is the transitive and reflexive closure of $G$
- Product $G * G' \stackrel{def}{=} (G \cup G')^*$
- Application $G \cdot G' \stackrel{def}{=}$ graph with vertices $V \cup V'$ and edges $E \cup E' \cup \{ (x, y) \mid x \in V, y \in V' \}$
- Atomic graphs $\ell$ stands for $\langle \{\ell\}, \ \{(\ell, \ell)\} \ \rangle$

# Causal Graphs

- $G^*$ is the transitive and reflexive closure of $G$
- Product $G * G' \stackrel{def}{=} (G \cup G')^*$
- Application $G \cdot G' \stackrel{def}{=}$ graph with vertices $V \cup V'$ and edges $E \cup E' \cup \{\ (x, y) \mid x \in V, y \in V'\ \}$
- Atomic graphs $\ell$ stands for $\langle \{\ell\},\ \{(\ell,\ell)\}\ \rangle$


- Any causal graph can be built from product, application and atomic graphs. Example:




$$(\mathrm{d} * \mathrm{k}) \cdot \ell \cdot \mathrm{e}$$

# Causal Graphs

## Definition

A causal graph $G$ is sufficient for (or weaker than) another causal graph $G'$, written $G \leq G'$, when $G \supseteq G'$.

- Notice that direction is switched: the smaller the graph, the stronger the cause!

# Causal Graphs

**Definition**

A causal graph $G$ is sufficient for (or weaker than) another causal graph $G'$, written $G \leq G'$, when $G \supseteq G'$.

- Notice that direction is switched: the smaller the graph, the stronger the cause!
- The empty graph $\langle \emptyset, \emptyset \rangle$ is the top element, denoted by $1$.
  - stands for absolute truth, and assigned to $\top$.
  - $1$ is the $*$ product and $\cdot$ application identity $t * 1 = t$ and $t \cdot 1 = 1 \cdot t = t$

# Causal Graphs

## Definition

A causal graph $G$ is sufficient for (or weaker than) another causal graph $G'$, written $G \leq G'$, when $G \supseteq G'$.

- Notice that direction is switched: the smaller the graph, the stronger the cause!
- The empty graph $\langle \emptyset, \emptyset \rangle$ is the top element, denoted by $1$.
  - stands for absolute truth, and assigned to $\top$.
  - $1$ is the $*$ product and $\cdot$ application identity $t * 1 = t$ and $t \cdot 1 = 1 \cdot t = t$
- We add a bottom element $0$,
  - weaker than any causal graph $0 < G$ for all $G$,
  - stands for false,
  - $0$ is the $*$ and $\cdot$ application annihilator $t * 0 = 0$ and $t \cdot 0 = 0 \cdot t = 0$

# Outline

# Positive programs

- Syntax: as usual plus an (optional) rule label

$$t : H \leftarrow B_1, \ldots, B_n$$

with $H$, $B_i$ atoms and $t$ can be a label $t = \ell$ or $t = 1$.

# Positive programs

- Syntax: as usual plus an (optional) rule label

$$t : H \leftarrow B_1, \ldots, B_n$$

with $H$, $B_i$ atoms and $t$ can be a label $t = \ell$ or $t = 1$.

**Definition (Causal model)**

A causal model of $P$ is an interpretation such that, for each rule:

$$\left( \mathcal{I}(B_1) * \ldots * \mathcal{I}(B_n) \right) \cdot t \;\; \leq \;\; \mathcal{I}(H)$$

# Alternative causes (symmetrical overdetermination)

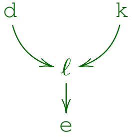
## Example

- A second law *m* specifies that *resisting* to authority is *punishable*.
- Suppose that some person drove drunk and resisted to authority.

$$\ell : \textit{punish} \leftarrow \textit{drive},\ \textit{drunk} \qquad \mathtt{d} : \textit{drive} \qquad \mathtt{k} : \textit{drunk}$$

$$\mathtt{e} : \textit{prison} \leftarrow \textit{punish} \qquad \mathtt{m} : \textit{punish} \leftarrow \textit{resist} \qquad \mathtt{r} : \textit{resist}$$

- Two equally valid alternative causes





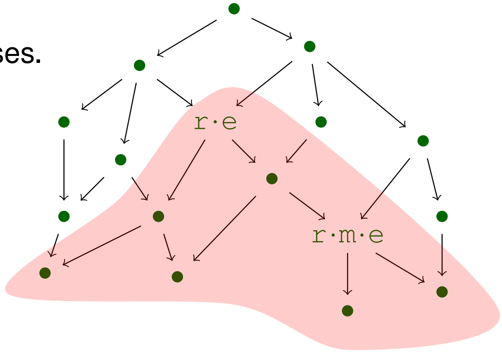
$(\mathtt{d} * \mathtt{k}) \cdot \ell \cdot \mathtt{e}$ $\qquad$ $\mathtt{r} \cdot \mathtt{m} \cdot \mathtt{e}$

# Alternative causes: Addition

- addition ($+$) represents alternative causes

$$\mathcal{I}(\mathit{punish}) \;=\; (\mathtt{d} * \mathtt{k})\cdot\ell \;+\; \mathtt{r}\cdot\mathtt{m}\cdot\mathtt{e}$$

- Causal values are ideals of causal graphs. ($+$) corresponds to the union ($\cup$) of ideals.
- Disregard redundant causes.



# Alternative causes

## Theorem

*$\langle \mathbf{V}_{Lb}, +, *, \cdot \rangle$ is the free algebra generated by labels Lb. Operations $*$ and $+$ are the meet and join of a completely distributive lattice.*

| *Associativity* | | |
|---|---|---|
| $t + (u{+}w)$ | $=$ | $(t{+}u) + w$ |
| $t * (u{*}w)$ | $=$ | $(t{*}u) * w$ |

| *Commutativity* | | |
|---|---|---|
| $t + u$ | $=$ | $u + t$ |
| $t * u$ | $=$ | $u * t$ |

| *Absorption* | | |
|---|---|---|
| $t$ | $=$ | $t + (t{*}u)$ |
| $t$ | $=$ | $t * (t{+}u)$ |

| *Distributive* | | |
|---|---|---|
| $t + (u{*}w)$ | $=$ | $(t{+}u) * (t{+}w)$ |
| $t * (u{+}w)$ | $=$ | $(t{*}u) + (t{*}w)$ |

| *Identity* | | |
|---|---|---|
| $t$ | $=$ | $t + 0$ |
| $t$ | $=$ | $t * 1$ |

| *Annihilator* | | |
|---|---|---|
| $1$ | $=$ | $1 + t$ |
| $0$ | $=$ | $0 * t$ |

# Alternative causes

- More specigic are the $(\cdot)$ application equations

$$\frac{\textit{Associativity}}{t \cdot (u{\cdot}w) \;=\; (t{\cdot}u) \cdot w} \qquad \frac{\textit{Addition distributivity}}{\begin{array}{rcl} t \cdot (u{+}w) & = & (t{\cdot}u) + (t{\cdot}w) \\ (t+u) \cdot w & = & (t{\cdot}w) + (u{\cdot}w) \end{array}}$$

$$\frac{\textit{Identity}}{\begin{array}{rcl} t & = & t \cdot 1 \\ t & = & 1 \cdot t \end{array}} \qquad \frac{\textit{Annihilator}}{\begin{array}{rcl} 0 & = & t \cdot 0 \\ 0 & = & 0 \cdot t \end{array}} \qquad \frac{\textit{Absorption}}{\begin{array}{rcl} t & = & t + u \cdot t \cdot w \\ u \cdot t \cdot w & = & t * u \cdot t \cdot w \end{array}}$$

- $l$ is a label, $c$, $d$ and $e$ terms without $(+)$

$$\frac{\textit{Label idempotence}}{\ell \cdot \ell \;=\; \ell} \qquad \frac{\textit{Product distributivity}}{\begin{array}{rcl} c \cdot (d * e) & = & (c \cdot d) * (c \cdot e) \\ (c * d) \cdot e & = & (c \cdot e) * (d \cdot e) \end{array}}$$

$$\frac{\textit{Transitivity}}{c \cdot d \cdot e \;=\; (c \cdot d) * (d \cdot e) \quad \text{with } d \neq 1}$$

# Positive programs

**Definition (Direct consequences)**

$$T_P(\mathcal{I})(p) \stackrel{def}{=} \sum \{ \; ( \, \mathcal{I}(B_1) * \ldots * \mathcal{I}(B_n) \, ) \cdot t \;\mid\; (t : p \leftarrow B_1, \ldots, B_n) \in P \; \}$$

**Theorem (Analogous to standard LP)**

*Let $P$ be a (possibly infinite) positive logic program with $n$ causal rules.*

(i) *$lfp(T_P)$ is the least model of $P$,*

(ii) *$lfp(T_P) = T_P\!\uparrow^{\omega}(\mathbf{0})$, and*

(iii) *iteration ends in finite steps when $P$ is finite $lfp(T_P) = T_P\!\uparrow^{n}(\mathbf{0})$.*

**Theorem**

*Removing all labels we get the traditional (two-valued) least model.*

# Positive programs

- Positive programs have a least model.

$$\mathcal{I}(\mathit{prison}) = (\mathtt{d} * \mathtt{k}) \cdot \ell \cdot \mathtt{e} + \mathtt{r} \cdot \mathtt{m} \cdot \mathtt{e}$$

- If we remove all labels, then it corresponds to the standard least model.

$$\mathcal{I}(\mathit{prison}) = 1$$

# Positive programs

- Positive programs have a least model.

$$\mathcal{I}(\mathit{prison}) = (\mathtt{d} * \mathtt{k}) \cdot \ell \cdot \mathtt{e} + \mathtt{r} \cdot \mathtt{m} \cdot \mathtt{e}$$

- If we remove all labels, then it corresponds to the standard least model.

$$\mathcal{I}(\mathit{prison}) = 1$$

- Each subterm with no sums is a cause. But what do causal values really capture?

# Positive programs

- Positive programs have a least model.

$$\mathcal{I}(\textit{prison}) = (\mathtt{d} * \mathtt{k}) \cdot \ell \cdot \mathtt{e} + \mathtt{r} \cdot \mathtt{m} \cdot \mathtt{e}$$

- If we remove all labels, then it corresponds to the standard least model.

$$\mathcal{I}(\textit{prison}) = 1$$

- Each subterm with no sums is a cause. But what do causal values really capture?
  - syntactic proofs?

# Positive programs

- Positive programs have a least model.

$$\mathcal{I}(\mathit{prison}) = (\mathtt{d} * \mathtt{k}) \cdot \ell \cdot \mathtt{e} + \mathtt{r} \cdot \mathtt{m} \cdot \mathtt{e}$$

- If we remove all labels, then it corresponds to the standard least model.

$$\mathcal{I}(\mathit{prison}) = 1$$

- Each subterm with no sums is a cause. But what do causal values really capture?
  - syntactic proofs?
  - some proofs? all proofs?

- Notice we have not used syntactic information!

# Positive programs

**Theorem**

*The causal value of an atom in the least model exactly corresponds to all its possible (non-redundant) proofs.*

$$(\mathtt{d} * \mathtt{k}) \cdot \ell \cdot \mathtt{e} \qquad\qquad \mathtt{r} \cdot \mathtt{m} \cdot \mathtt{e}$$

$$\cfrac{\cfrac{\cfrac{\top}{\mathit{drive}}\,(\mathtt{d}) \qquad \cfrac{\top}{\mathit{drunk}}\,(\mathtt{k})}{\mathit{punish}}\,(\ell)}{\mathit{prison}}\,(\mathtt{e}) \qquad\qquad \cfrac{\cfrac{\cfrac{\top}{\mathit{resist}}\,(\mathtt{r})}{\mathit{punish}}\,(\mathtt{m})}{\mathit{prison}}\,(\mathtt{e})$$

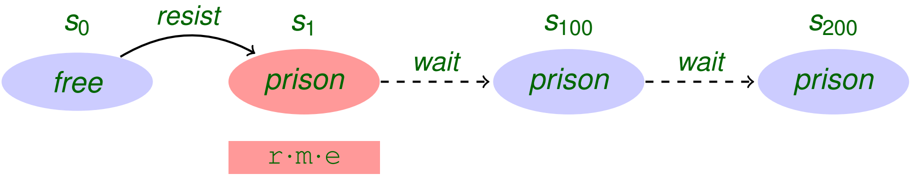
# Outline

# Default negation

- Negation will be used for representing defaults.
  - Inertia laws are an example of dynamic defaults.

# Default negation

- Negation will be used for representing defaults.
  - Inertia laws are an example of dynamic defaults.
  - Suppose now that we add time to our running example and we are imprisoned by *resist* at situation $s_1$, then

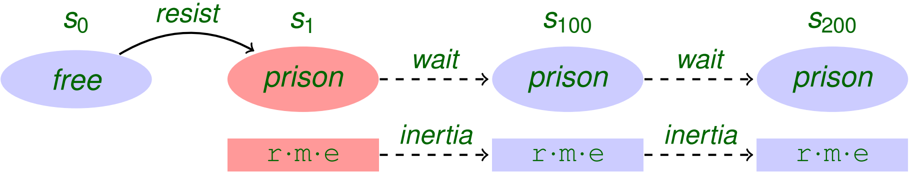



- Inertia law

$$prison(T+1) \leftarrow prison(T), \ not \ free(T+1)$$

# Default negation

- Negation will be used for representing defaults.
  - Inertia laws are an example of dynamic defaults.
  - Suppose now that we add time to our running example and we are imprisoned by *resist* at situation $s_1$, then

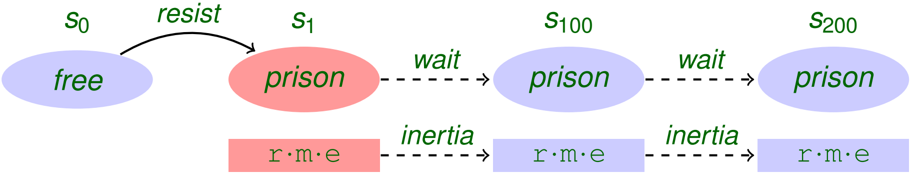



- Inertia law

$$prison(T+1) \leftarrow prison(T),\ not\ free(T+1)$$

# Default negation

- Negation will be used for representing defaults.
  - Inertia laws are an example of dynamic defaults.
  - Suppose now that we add time to our running example and we are imprisoned by *resist* at situation $s_1$, then




- Inertia law

$$prison(T+1) \leftarrow prison(T),\ not\ free(T+1)$$

- Causal values persist by inertia. We disregard explanations for *not* being *free* along that period!

# Default negation

- *not free*$(T+1)$ is the default (or expected) behaviour
  - if this happens, no cause is propagated (*not free*$(T+1)$ becomes 1).

# Default negation

- $not\ free(T+1)$ is the default (or expected) behaviour
  - if this happens, no cause is propagated ($not\ free(T+1)$ becomes 1).
- Program reduct.
  Static default: punished people normally goes to prison

$$
\begin{aligned}
\ell &: \ punish \leftarrow drive, drunk & \quad d &: \ drive \\
m &: \ punish \leftarrow resist & \quad k &: \ drunk \\
e &: \ prison \leftarrow punish, not\ abnormal & \quad r &: \ resist
\end{aligned}
$$

# Default negation

- *not free*$(T + 1)$ is the default (or expected) behaviour
  - if this happens, no cause is propagated (*not free*$(T+1)$ becomes 1).

- Program reduct.
  Static default: punished people normally goes to prison

$$
\begin{array}{llll}
\ell: & punish \leftarrow drive, drunk & \mathtt{d}: & drive \\
\mathtt{m}: & punish \leftarrow resist & \mathtt{k}: & drunk \\
\mathtt{e}: & prison \leftarrow punish, not\ abnormal & \mathtt{r}: & resist
\end{array}
$$

- If we assume $\mathcal{I}(\ abnormal\ ) = 0$ (false).

$$
\begin{array}{llll}
\ell: & punish \leftarrow drive, drunk & \mathtt{d}: & drive \\
\mathtt{m}: & punish \leftarrow resist & \mathtt{k}: & drunk \\
\mathtt{e}: & prison \leftarrow punish, \xcancel{not\ abnormal} & \mathtt{r}: & resist
\end{array}
$$

# Default negation

- we can flexibly add exceptions

$$\begin{aligned} \textit{abnormal} &\leftarrow \textit{pardon} \\ \textit{abnormal} &\leftarrow \textit{revoke} \\ \textit{abnormal} &\leftarrow \textit{diplomat} \end{aligned}$$

- If we assume to be a *diplomat*, then $\mathcal{I}(\textit{abnormal}) = 1$ (true).

$$\begin{aligned} \ell :&\quad \textit{punish} \leftarrow \textit{drive}, \textit{drunk} & \mathtt{d} :&\quad \textit{drive} \\ \mathtt{m} :&\quad \textit{punish} \leftarrow \textit{resist} & \mathtt{k} :&\quad \textit{drunk} \\ \sout{\mathtt{e} :}&\quad \sout{\textit{prison} \leftarrow \textit{punish}, \textit{not abnormal}} & \mathtt{r} :&\quad \textit{resist} \end{aligned}$$

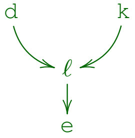
**Theorem**

*For each (standard) two-valued stable model there is (exactly one) corresponding causal stable model and vice versa.*

# Outline

# Sufficient Cause

- Why are we in prison?
  - $\mathit{sufficient}(X, \mathit{prison})?$, $X$ should be a minimimal explanation

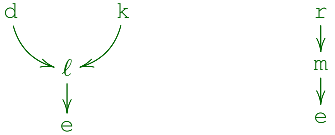




$(\mathtt{d} * \mathtt{k}) \cdot \ell \cdot \mathtt{e}$ $\qquad$ $\mathtt{r} \cdot \mathtt{m} \cdot \mathtt{e}$

# Sufficient Cause

- Why are we in prison?
  - *sufficient*$(X, prison)$?, $X$ should be a minimimal explanation




$(d * k) \cdot \ell \cdot e$ $\quad$ $r \cdot m \cdot e$

  - Was $d * k * chew$ sufficient to cause it?
  - $sufficient(d * k * chew, prison)$ should holds, despite of lack of minimality

# Sufficient Cause

- Given a causal graph $G$
  - $G$ is a sufficient explanation for $p$ iff $G \leq I(p)$
  - $G$ is a sufficient cause for $p$ iff $G$ is a subgraph-minimal sufficient explation for $p$

# Sufficient Cause

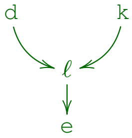
- Given a causal graph $G$
  - $G$ is a sufficient explanation for $p$ iff $G \leq I(p)$
  - $G$ is a sufficient cause for $p$ iff $G$ is a subgraph-minimal sufficient explation for $p$
- Complexity (complete results)

| | positive | well founded | answer set (brave) | answer set (cautions) |
|---|---|---|---|---|
| entailment | P | P | NP | coNP |
| explanation | P | P | NP | coNP |
| cause | P | P | NP | coNP |

  - same complexity than entailment in standard LP

# Necessary Cause

- Why are we in prison?
  - What has been necessary to cause it?

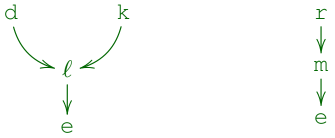



$(d * k)\cdot \ell \cdot e$


$r \cdot m \cdot e$

# Necessary Cause

- Why are we in prison?
  - What has been necessary to cause it?

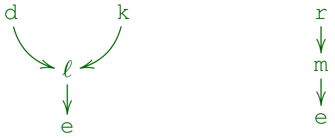



$(\mathrm{d} * \mathrm{k}) \cdot \ell \cdot \mathrm{e}$ $\qquad$ $\mathrm{r} \cdot \mathrm{m} \cdot \mathrm{e}$

  - Only the rule *e* has been necessary.

# Necessary Cause

- Why are we in prison?
  - What has been necessary to cause it?




$(d * k) \cdot \ell \cdot e$ $\quad$ $r \cdot m \cdot e$

  - Only the rule *e* has been necessary.
  - Suppose we do not resit. Then *drive* and *drunk* would have been necessary causes.
  - Suppose we were not drunk. Then *resit* would have been a necessary cause.

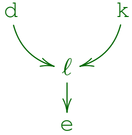
# Necessary Cause

- Given a causal graph $G$
  - $G$ is a necessary cause for $p$ iff $G$ subgraph of all sufficient causes for $p$ and $I(p) \neq 0$
  - $G$ is a necessary cause for $p$ iff $G \geq I(p)$ and $I(p) \neq 0$
- Complexity (complete results)

| | positive | well founded | answer set (brave) | answer set (cautions) |
|---|---|---|---|---|
| entailment | P | P | NP | coNP |
| necessary | coNP | coNP | $\Sigma_2^P$ | coNP |

# Actual and Contributory Cause

- Why are we in prison?
  - Actual Cause $\approx$ contingency necessary cause.
  - There exists a possible world where *G* is a necessary cause [Pearl 2000, Halpern & Pearl 2001 and 2005].

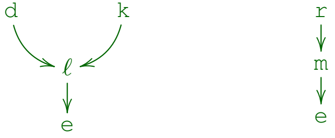



$(d * k) \cdot \ell \cdot e$


$r \cdot m \cdot e$

# Actual and Contributory Cause

- Why are we in prison?
  - Actual Cause ≈ contingency necessary cause.
  - There exists a possible world where *G* is a necessary cause [Pearl 2000, Halpern & Pearl 2001 and 2005].




$(\mathtt{d} * \mathtt{k})\cdot\ell\cdot\mathtt{e}$ $\mathtt{r}\cdot\mathtt{m}\cdot\mathtt{e}$

  - Contributory cause: Necessary condition in a sufficient cause [Mackie 1965, Wright 1988]

# Actual and Contributory Cause

- Given a causal graph $G$
  - $G$ is a actual cause for $p$ iff there exists a sufficient cause $G'$ for $p$ such that $G \subseteq G'$
- Complexity

| | positive | well founded | answer set (brave) | answer set (cautions) |
|---|---|---|---|---|
| entailment | P | P | NP | coNP |
| actual | $\leq$ NP | $\leq$ NP | $\leq$ NP | $\leq \Pi_2^P$ |
| HP 2001 | NP / $\Sigma_2^P$ | | | |
| HP 2005 | $D_2^P$ | | | |

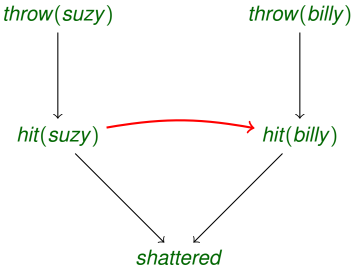
  - [Eiter & Lukasiewicz 2001, Aleksandrowicz et. al. 2014]
  - $\Sigma_2^P \leq D_2^P \leq \Delta_3^P \leq \Sigma_3^P$
  - $\Pi_2^P \leq D_2^P \leq \Delta_3^P \leq \Pi_3^P$

# Causality and Knowledge Representation

## Example (Lewis2000)

Suzy throws a rock at a bottle. The rock hits the bottle, shattering it. Suzy's friend Billy throws a rock at the bottle a couple of seconds later. Who has caused the bottle to shattered?

$$\begin{aligned} hit(suzy) &= throw(suzy) \\ hit(billy) &= throw(billy) \wedge \neg hit(suzy) \\ shattered &= hit(suzy) \vee hit(billy) \end{aligned}$$




- Actual Cause in structural equations [Halpern&Pearl2005, Hall2007, Halpern2008, Halpern2014]

# Causality and Knowledge Representation

- Suppose that John has also thrown after Billy.

$$
\begin{aligned}
hit(suzy) &= throw(suzy)\\
hit(billy) &= throw(billy) \wedge \neg hit(suzy)\\
hit(john) &= throw(john) \wedge \neg hit(suzy) \wedge \neg hit(billy)\\
shattered &= hit(suzy) \vee hit(billy) \vee hit(john)
\end{aligned}
$$

# Causality and Knowledge Representation

- Suppose that John has also thrown after Billy.

$$
\begin{aligned}
hit(suzy) &= throw(suzy)\\
hit(billy) &= throw(billy) \wedge \neg hit(suzy)\\
hit(john) &= throw(john) \wedge \neg hit(suzy) \wedge \neg hit(billy)\\
shattered &= hit(suzy) \vee hit(billy) \vee hit(john)
\end{aligned}
$$

- Change: John has thrown before Suzy.

# Causality and Knowledge Representation

- Suppose that John has also thrown after Billy.

$$\begin{aligned}
hit(suzy) &= throw(suzy)\\
hit(billy) &= throw(billy) \wedge \neg hit(suzy)\\
hit(john) &= throw(john) \wedge \neg hit(suzy) \wedge \neg hit(billy)\\
shattered &= hit(suzy) \vee hit(billy) \vee hit(john)
\end{aligned}$$

- Change: John has thrown before Suzy.

$$\begin{aligned}
hit(suzy) &= throw(suzy) \wedge \neg hit(john)\\
hit(billy) &= throw(billy) \wedge \neg hit(suzy) \wedge \neg hit(john)\\
hit(john) &= throw(john)\\
shattered &= hit(suzy) \vee hit(billy) \vee hit(john)
\end{aligned}$$

# Causality and Knowledge Representation

- Suppose that John has also thrown after Billy.

$$
\begin{aligned}
hit(suzy) &= throw(suzy)\\
hit(billy) &= throw(billy) \wedge \neg hit(suzy)\\
hit(john) &= throw(john) \wedge \neg hit(suzy) \wedge \neg hit(billy)\\
shattered &= hit(suzy) \vee hit(billy) \vee hit(john)
\end{aligned}
$$

- Change: John has thrown before Suzy.

$$
\begin{aligned}
hit(suzy) &= throw(suzy) \wedge \neg hit(john)\\
hit(billy) &= throw(billy) \wedge \neg hit(suzy) \wedge \neg hit(john)\\
hit(john) &= throw(john)\\
shattered &= hit(suzy) \vee hit(billy) \vee hit(john)
\end{aligned}
$$

- Small changes implies revise the entire model. Problem of tolerance to the elaboration [McCarthy1998]

### Example (Lewis2000)

Suzy throws a rock at a bottle. The rock hits the bottle, shattering it. Suzy's friend Billy throws a rock at the bottle a couple of seconds later. Who has caused the bottle to shattered?

$$shattered(T+1) \leftarrow throws(X,T),\ not\ shattered(T)$$
$$throw(suzy, 2)$$
$$throw(billy, 4)$$

**Example (Lewis2000)**

Suzy throws a rock at a bottle. The rock hits the bottle, shattering it. Suzy's friend Billy throws a rock at the bottle a couple of seconds later. Who has caused the bottle to shattered?

$$\begin{aligned}&\mathit{shattered}(T+1) \leftarrow \mathit{throws}(X,T),\ \mathit{not\ shattered}(T)\\&\mathit{throw}(\mathit{suzy},2)\\&\mathit{throw}(\mathit{billy},4)\end{aligned}$$

- Inertia axiom

$$\mathit{shattered}(T+1) \leftarrow \mathit{shattered}(T)$$

- We may conclude that the bottle is *shattered* at 3, but not who caused it.

# Causality and Knowledge Representation

$$
\begin{aligned}
r_1 &: \mathit{shattered}(T+1) \leftarrow \mathit{throws}(X,T),\ \mathit{not}\ \mathit{shattered}(T)\\
\mathit{suzy} &: \mathit{throw}(\mathit{suzy},2)\\
\mathit{billy} &: \mathit{throw}(\mathit{billy},4)
\end{aligned}
$$

# Causality and Knowledge Representation

$$
\begin{aligned}
r_1 &: shattered(T+1) \leftarrow throws(X,T),\ not\ shattered(T)\\
suzy &: throw(suzy,2)\\
billy &: throw(billy,4)
\end{aligned}
$$

- We may conclude that the bottle is *shattered* at 3 because

$$
\begin{array}{c}
suzy\\
\downarrow\\
r_1
\end{array}
$$

$$suzy \cdot r_1$$

- Note that rule $r_1$ for $T = 4$ is not in the reduct of the program

# Conclusions

- Multi-valued semantics based on (ideals of) causal graphs
- Values capture non-redundant proofs, but with semantic, algebraic operations
- Default negation = absence of cause.
  - Reduct definition allows defining causal stable models
  - Allows expressing dynamic defaults (ex: inertia laws)
- Ongoing work:
  - Studding actual causation.
  - Adding this causal operators on rule bodies.

# Causal Graph Justifications of Stable Models

Jorge Fandinno

Thanks for your attention!

Logical Reasoning and computation
Corunna, Spain

February 24th, 2015