Enablers and Inhibitors in Causal Justifications of Logic Programs

Pedro Cabalar and Jorge Fandinno

Department of Computer Science
University of Corunna (Spain)
{cabalar,fandino}@udc.es

September 28th, 2015
LPNMR'15
Lexington, KY, USA

Joint work with **Jorge Fandinno** (recent PhD)

UNIVERSIDADE DA CORUÑA

A Causal Semantics for Logic Programming
September 11th, 2015

Context: Causal Justifications

- Logic programming is an important paradigm for problem solving.
- Sometimes, is not enough to provide some conclusions, but we also are required to **provide explanations** for them.
  - In LP, true atoms must be justified.

Context: Causal Justifications

- Logic programming is an important paradigm for problem solving.
- Sometimes, is not enough to provide some conclusions, but we also are required to **provide explanations** for them.
  - In LP, true atoms must be justified.
  - Extensive **literature on justifications** and debugging in LP [Denecker & De Schreye 93; Gebser et al 08; Pontelli et al 09; Oetsch et al 10; Schulz & Toni 13]

Context: Causal Justifications

- Logic programming is an important paradigm for problem solving.
- Sometimes, is not enough to provide some conclusions, but we also are required to **provide explanations** for them.
  - In LP, true atoms must be justified.
  - Extensive **literature on justifications** and debugging in LP [Denecker & De Schreye 93; Gebser et al 08; Pontelli et al 09; Oetsch et al 10; Schulz & Toni 13]
- A pair of **multi-valued** approaches with **algebraic constructions**:
  - **Why-not Provenance** [Damásio et al 2013]:
    oriented to debugging and explanation, well-founded semantics
  - **Causal Graphs** [Cabalar and Fandinno 14]:
    causes = proof graphs from (the positive part) of an ASP program

Context: Causal Justifications

- Logic programming is an important paradigm for problem solving.
- Sometimes, is not enough to provide some conclusions, but we also are required to **provide explanations** for them.
  - In LP, true atoms must be justified.
  - Extensive **literature on justifications** and debugging in LP [Denecker & De Schreye 93; Gebser et al 08; Pontelli et al 09; Oetsch et al 10; Schulz & Toni 13]
- A pair of **multi-valued** approaches with **algebraic constructions**:
  - **Why-not Provenance** [Damásio et al 2013]:
    oriented to debugging and explanation, well-founded semantics
  - **Causal Graphs** [Cabalar and Fandinno 14]:
    causes = proof graphs from (the positive part) of an ASP program

Our proposal

We propose **Extended Causal Justifications** mixing both:

- **causal explanations** = proof graphs built with rule labels

+

- **negative terms** = related to default negation, they define **enablers** and **inhibitors** for applying causal rules

Our proposal

We propose **Extended Causal Justifications** mixing both:

- **causal explanations** = proof graphs built with rule labels

+

- **negative terms** = related to default negation, they define **enablers** and **inhibitors** for applying causal rules

Our interest: representing **(relevant) causal knowledge** rather than debugging or providing all possible explanations

Motivation Example

Example

- If James Bond drinks drug `d`, he will have paralysis `p` unless has been administered an antidote `a`.
- The MI5 daily administers Bond antidote `a`, unless he is on holiday `h`.

Motivation Example

Example

- If James Bond drinks drug `d`, he will have paralysis `p` unless has been administered an antidote `a`.
- The MI5 daily administers Bond antidote `a`, unless he is on holiday `h`.

- Le Chiffre has poured drug `d` on Bond's drink!

Example

- If James Bond drinks drug d, he will have paralysis p unless has been administered an antidote a.
- The MI5 daily administers Bond antidote a, unless he is on holiday h.

- Le Chiffre has poured drug d on Bond's drink!

$$
\begin{array}{llll}
\mathtt{p}: & \mathtt{p} & \leftarrow & \mathtt{d}, \mathtt{not}\ \mathtt{a} \\
\mathtt{a}: & \mathtt{a} & \leftarrow & \mathtt{not}\ \mathtt{h} \\
\mathtt{d}: & \mathtt{d} & &
\end{array}
$$

Example

- If James Bond drinks drug d, he will have paralysis p unless has been administered an antidote a.
- The MI5 daily administers Bond antidote a, unless he is on holiday h.

- Le Chiffre has poured drug d on Bond's drink!

$$
\begin{array}{llll}
p: & p & \leftarrow & d, \text{not } a \\
a: & a & \leftarrow & \text{not } h \\
d: & d & &
\end{array}
$$

- Bond does not have paralysis, but **why not**?

Motivation Example

Example

- Suppose now that, that day, Bond was actually on a **holiday** with Vesper.

$$\begin{array}{lll} \mathtt{p}: & \mathtt{p} \leftarrow & \mathtt{d}, \mathtt{not\ a} \\ \mathtt{a}: & \mathtt{a} \leftarrow & \mathtt{not\ h} \\ \mathtt{d}: & \mathtt{d} & \\ \mathtt{h}: & \mathtt{h} & \end{array}$$

Example

- Suppose now that, that day, Bond was actually on a **holiday** with Vesper.

$$\begin{aligned} p:&\quad p \leftarrow d, \text{not } a \\ a:&\quad a \leftarrow \text{not } h \\ d:&\quad d \\ h:&\quad h \end{aligned}$$

- **Counterfactuals**:
  - Had Le Chiffre not poured the drug d, Bond would not have paralysis d.
  - Had Bond not being on holiday h, he would not have paralysis d.

Motivation Example

Example

- Suppose now that, that day, Bond was actually on a **holiday** with Vesper.

```
p :  p  ←  d, not a
a :  a  ←  not h
d :  d
h :  h
```

- **Counterfactuals**:
  - Had Le Chiffre not poured the drug *d*, Bond would not have paralysis *d*.
  - Had Bond not being on holiday *h*, he would not have paralysis *d*.

- **Causality**:
  - Is the drug *d* a cause of Bond's paralysis *d*? Yes!!
  - Is a holiday *h* a cause of Bond's paralysis *d*?

Motivation Example

Example

- Suppose now that, that day, Bond was actually on a **holiday** with Vesper.

```
p:  p  ←  d, not a
a:  a  ←  not h
d:  d
h:  h
```

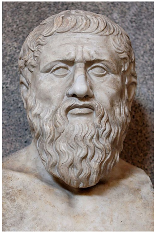
- **Counterfactuals**:
  - Had Le Chiffre not poured the drug *d*, Bond would not have paralysis *d*.
  - Had Bond not being on holiday *h*, he would not have paralysis *d*.

- **Causality**:
  - Is the drug *d* a cause of Bond's paralysis *d*? Yes!!
  - Is a holiday *h* a cause of Bond's paralysis *d*? Controversial!!!

Motivation Example

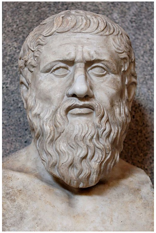
- Is a holiday *h* a cause of Bond's paralysis *d*?
  - **Yes**: [Lewis73; Halpern and Pearl2001/2005]
  - No: [Hall2004/2007; Mauldlin2004]
  - May be: [Halpern and Hitchcock2011]

- Is a holiday h a cause of Bond's paralysis d?
  - **Yes**: [Lewis73; Halpern and Pearl2001/2005]
  - No: [Hall2004/2007; Mauldlin2004]
  - May be: [Halpern and Hitchcock2011]



- Plato: "**distinguish the real cause** from that without which the cause would not be able to act as a cause."
  - The drug d is the real cause producing the paralysis.
  - The holiday h is an enabler of the cause.

- Is a holiday h a cause of Bond's paralysis d?
  - **Yes**: [Lewis73; Halpern and Pearl2001/2005]
  - No: [Hall2004/2007; Mauldlin2004]
  - May be: [Halpern and Hitchcock2011]



- Plato: "**distinguish the real cause** from that without which the cause would not be able to act as a cause."
  - The drug d is the real cause producing the paralysis.
  - The holiday h is an enabler of the cause.
- When not on holiday, the antidote a becomes an inhibitor of the potential cause (the drug) of Bond's paralysis.

In this work

- We propose a **multivalued semantics for LP**, Extended Causal Justifications (ECJ), that captures causes, enablers and inhibitors.
- **Generalizes**, under the well-founded semantics, both
  - **Causal Graph justifications** (CG): only captures real causes.
  - **Why-not Provenance** (WnP): only captures counterfactual dependence
- We also obtain a **formal comparation between CG and WnP**.

Outline

Positive programs

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

$$r_i : \quad H \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n$$

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

Positive programs

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

$$r_i : \; H \leftarrow B_1, \ldots, B_m, not\ B_{m+1}, \ldots, not\ B_n$$

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

- Labels in the following program are part of the syntax

$$\begin{array}{llll} p: & p & \leftarrow & d, not\ a \\ a: & a & \leftarrow & not\ h \\ d: & d & & \end{array}$$

Positive programs

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

$$r_i : \ H \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n$$

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

- Labels in the following program are part of the syntax

$$\begin{array}{llll} p: & p & \leftarrow & d, \text{not } a \\ a: & a & \leftarrow & \text{not } h \\ d: & d & & \end{array}$$

- We will assign the expression formed by rule labels $(\sim a * d) \cdot p$ to the atom p.

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

$$\mathtt{r_i}: \ \mathtt{H} \leftarrow \mathtt{B_1}, \ldots, \mathtt{B_m}, \mathtt{not}\ \mathtt{B_{m+1}}, \ldots, \mathtt{not}\ \mathtt{B_n}$$

with $\mathtt{H}$ and $\mathtt{B_j}$ atoms. $\mathtt{r_i}$ can be a label $\mathtt{r_i} = \ell$ or $\mathtt{t} = 1$.

- Labels in the following program are part of the syntax

$$\begin{array}{llll} \mathtt{p}: & \mathtt{p} & \leftarrow & \mathtt{d}, \mathtt{not}\ \mathtt{a} \\ \mathtt{a}: & \mathtt{a} & \leftarrow & \mathtt{not}\ \mathtt{h} \\ \mathtt{d}: & \mathtt{d} & & \end{array}$$

- We will assign the expression formed by rule labels $(\sim\mathtt{a} * \mathtt{d}) \cdot \mathtt{p}$ to the atom $\mathtt{p}$.
  - $\mathtt{d}$ (negated label) is an inhibitor of the drug $\mathtt{d}$.
  - '$\cdot$' captures the order of rule application: $\mathtt{p}$ would have been applied to $\mathtt{d}$ if $\mathtt{a}$ had not hold.

Causal Values

- **Causal terms** are expressions of the form

$$t \quad ::= \quad \ell \;\;|\;\; \prod S \;\;|\;\; \sum S \;\;|\;\; t_1 \cdot t_2 \;\;|\;\; {\sim} t_1$$

where ℓ is a rule label, S is a set of causal terms and t_1 and t_2 are causal terms.

Causal Values

- **Causal terms** are expressions of the form
$$t \quad ::= \quad \ell \;\mid\; \prod S \;\mid\; \sum S \;\mid\; t_1 \cdot t_2 \;\mid\; {\sim} t_1$$
where ℓ is a rule label, S is a set of causal terms and t_1 and t_2 are causal terms.
- By 1 and 0 we denote the empty product $\prod \emptyset$ and sum $\sum \emptyset$, respectively.

- **Causal terms** are expressions of the form
$$t \ ::= \ \ell \ | \ \prod S \ | \ \sum S \ | \ t_1 \cdot t_2 \ | \ \sim t_1$$
where ℓ is a rule label, S is a set of causal terms and t_1 and t_2 are causal terms.
- By 1 and 0 we denote the empty product $\prod \emptyset$ and sum $\sum \emptyset$, respectively.
- **Causal values** are the equivalence classes of causal terms under completely distributive (complete) lattice

Associativity	*Commutativity*	*Absorption*
$t + (u+w) = (t+u) + w$	$t + u = u + t$	$t = t + (t * u)$
$t * (u * w) = (t * u) * w$	$t * u = u * t$	$t = t * (t+u)$

Distributive	*Identity*	*Annihilator*
$t + (u * w) = (t+u) * (t+w)$	$t = t + 0$	$1 = 1 + t$
$t * (u+w) = (t * u) + (t * w)$	$t = t * 1$	$0 = 0 * t$

plus ...

Causal Values

- plus the following axioms for '·'

Associativity

$$t \cdot (u{\cdot}w) \;=\; (t{\cdot}u) \cdot w$$

Product Distributivity

$$\begin{aligned} c \cdot (d * e) &= (c \cdot d) * (c \cdot e) \\ (c * d) \cdot e &= (c \cdot e) * (d \cdot e) \end{aligned}$$

Identity

$$\begin{aligned} t &= t \cdot 1 \\ t &= 1 \cdot t \end{aligned}$$

Addition distributivity

$$\begin{aligned} t \cdot (u{+}w) &= (t{\cdot}u) + (t{\cdot}w) \\ (t + u) \cdot w &= (t{\cdot}w) + (u{\cdot}w) \end{aligned}$$

Absorption

$$\begin{aligned} t + u{\cdot}t{\cdot}w &= t \\ t * u{\cdot}t{\cdot}w &= u{\cdot}t{\cdot}w \end{aligned}$$

Annihilator

$$\begin{aligned} 0 &= t \cdot 0 \\ 0 &= 0 \cdot t \end{aligned}$$

Label idempotence

$$\ell \cdot \ell \;=\; \ell$$

Graph Representation

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

Causal Values

- plus the following axioms for '$\cdot$'

Associativity

$$t \cdot (u \cdot w) = (t \cdot u) \cdot w$$

Product Distributivity

$$c \cdot (d * e) = (c \cdot d) * (c \cdot e)$$
$$(c * d) \cdot e = (c \cdot e) * (d \cdot e)$$

Identity

$$t = t \cdot 1$$
$$t = 1 \cdot t$$

Addition distributivity

$$t \cdot (u + w) = (t \cdot u) + (t \cdot w)$$
$$(t + u) \cdot w = (t \cdot w) + (u \cdot w)$$

Absorption

$$t + u \cdot t \cdot w = t$$
$$t * u \cdot t \cdot w = u \cdot t \cdot w$$

Annihilator

$$0 = t \cdot 0$$
$$0 = 0 \cdot t$$

Label idempotence

$$\ell \cdot \ell = \ell$$

Graph Representation

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

- plus the following axioms for '$\sim$'

pseudo-compl.

$$t * {\sim} t = 0$$
$${\sim}{\sim}{\sim} t = {\sim} t$$

De Morgan

$${\sim}(t + u) = ({\sim} t * {\sim} u)$$
$${\sim}(t * u) = ({\sim} t + {\sim} u)$$

excluded middle

$${\sim} t + {\sim}{\sim} t = 1$$

appl. negation

$${\sim}(t \cdot u) = {\sim}(t * u)$$

- Every causal value can be represented by a term in **disjunctive normal form**
  - sum '$+$' is not in the scope of another connective: it separates **alternative causes**
  - negation '$\sim$' is only applied to labels or negated labels

$$(\sim a * d) \cdot p \;+\; (\sim\sim h * d) \cdot p$$

- Every causal value can be represented by a term in **disjunctive normal form**
  - sum '$+$' is not in the scope of another connective: it separates **alternative causes**
  - negation '$\sim$' is only applied to labels or negated labels

$$(\sim a * d) \cdot p \ + \ (\sim\sim h * d) \cdot p$$

- $\sim$a means that a is an **inhibitor** of d.

- Every causal value can be represented by a term in **disjunctive normal form**
  - sum '$+$' is not in the scope of another connective: it separates **alternative causes**
  - negation '$\sim$' is only applied to labels or negated labels

$$(\sim \mathtt{a} * \mathtt{d}) \cdot \mathtt{p} \; + \; (\sim\sim \mathtt{h} * \mathtt{d}) \cdot \mathtt{p}$$

- $\sim \mathtt{a}$ means that $\mathtt{a}$ is an **inhibitor** of $\mathtt{d}$.
- Negation is not classical, $\sim\sim \mathtt{h} \neq \mathtt{h}$ allows to **distinguish between an enabler $\mathtt{h}$ and the real cause $\mathtt{d}$** . . .

Definition (Causal model)

A **causal model** of P is an interpretation satisfying, for each rule:

$$\big(\ I(B_1) * \ldots * I(B_n) * I(\text{not } B_{m+1}) * \ldots * I(\text{not } B_n)\ \big) \cdot t \ \leq\ I(H)$$

Definition (Causal model)

A **causal model** of $\mathtt{P}$ is an interpretation satisfying, for each rule:

$$\Big(\ \mathtt{I(B_1)} * \ldots * \mathtt{I(B_n)} * \mathtt{I(not\ B_{m+1})} * \ldots * \mathtt{I(not\ B_n)}\ \Big) \cdot \mathtt{t} \ \leq\ \mathtt{I(H)}$$

- Labelling a rule with $\mathtt{t} = 1$ is used to **ignore** the trace of a rule.

$$\Big(\ \mathtt{I(B_1)} * \ldots * \mathtt{I(B_m)} * \mathtt{I(not\ B_{m+1})} * \ldots * \mathtt{I(not\ B_n)}\ \Big) \cdot 1 \ \leq\ \mathtt{I(H)}$$

Definition (Causal model)

A **causal model** of P is an interpretation satisfying, for each rule:

$$\big(\ I(B_1) * \ldots * I(B_n) * I(not\ B_{m+1}) * \ldots * I(not\ B_n)\ \big) \cdot t \ \leq\ I(H)$$

- Labelling a rule with $t = 1$ is used to **ignore** the trace of a rule.

$$\big(\ I(B_1) * \ldots * I(B_m) * I(not\ B_{m+1}) * \ldots * I(not\ B_n)\ \big) \cdot \not{1} \ \leq\ I(H)$$

Definition (Causal model)

A **causal model** of P is an interpretation satisfying, for each rule:

$$\Big(\ I(B_1) * \ldots * I(B_n) * I(not\ B_{m+1}) * \ldots * I(not\ B_n)\ \Big) \cdot t\ \ \leq\ \ I(H)$$

- Labelling a rule with $t = 1$ is used to **ignore** the trace of a rule.

$$\Big(\ I(B_1) * \ldots * I(B_m) * I(not\ B_{m+1}) * \ldots * I(not\ B_n)\ \Big)\ \ \leq\ \ I(H)$$

Definition (Causal model)

A **causal model** of $\mathtt{P}$ is an interpretation satisfying, for each rule:

$$\Big(\ \mathtt{I(B_1)} * \ldots * \mathtt{I(B_n)} * \mathtt{I(not\ B_{m+1})} * \ldots * \mathtt{I(not\ B_n)}\ \Big) \cdot \mathtt{t} \ \leq\ \mathtt{I(H)}$$

- Labelling a rule with $\mathtt{t} = 1$ is used to **ignore** the trace of a rule.

$$\Big(\ \mathtt{I(B_1)} * \ldots * \mathtt{I(B_m)} * \mathtt{I(not\ B_{m+1})} * \ldots * \mathtt{I(not\ B_n)}\ \Big) \ \leq\ \mathtt{I(H)}$$

Theorem

A positive program has a least model, which can be computed by iteration of a direct consequences operator from the bottom interpretation.

Definition (Reduct)

The reduct P^J of a program P w.r.t. an interpretation J contains

$$r_i : \quad H \leftarrow B_1, \ldots, B_m, \; J(\text{not } B_{m+1}), \ldots, \; J(\text{not } B_n)$$

per each rule in P.

- By $\Gamma_P(J)$, we denote the least model I of P^J.
- $\Gamma_P(J)$ is antimonotonic and, thus $\Gamma_P^2(J)$ is monotonic.

Definition (Causal well-founded model)

The **causal well-founded model** $\mathbb{W}_P$ is a mapping such that

$$\mathbb{W}_P(A) \stackrel{def}{=} \mathrm{lfp}(\Gamma_P^2)(A)$$

$$\mathbb{W}_P(\text{not } A) \stackrel{def}{=} {\sim}\mathrm{gfp}(\Gamma_P^2)(A)$$

$$\mathbb{W}_P(\text{undef } A) \stackrel{def}{=} {\sim}\mathbb{W}_P(A) * {\sim}\mathbb{W}_P(\text{not } A)$$

- For instance, the least and greatest fixpoint of

$$\begin{array}{llll} \mathtt{p}: & \mathtt{p} \leftarrow \mathtt{d}, \mathtt{not}\ \mathtt{a} & \mathtt{d}: & \mathtt{d} \\ \mathtt{a}: & \mathtt{a} \leftarrow \mathtt{not}\ \mathtt{h} & \mathtt{h}: & \mathtt{h} \end{array}$$

coincide and satisfy

$$\mathrm{lfp}(\Gamma_{\mathtt{P}}^2)(\mathtt{a}) \quad = \quad {\sim}\mathtt{h}\cdot\mathtt{a} \qquad\qquad \mathrm{lfp}(\Gamma_{\mathtt{P}}^2)(\mathtt{p}) \quad = \quad ({\sim}\mathtt{a} * \mathtt{d})\cdot\mathtt{p} + ({\sim\sim}\mathtt{h} * \mathtt{d})\cdot\mathtt{p}$$

- Each addend is a justification: ${\sim}\mathtt{h}\cdot\mathtt{a}$, $({\sim}\mathtt{a} * \mathtt{d})\cdot\mathtt{p}$ and $({\sim\sim}\mathtt{h} * \mathtt{d})\cdot\mathtt{p}$.

Well-founded causal model

- For instance, the least and greatest fixpoint of

$$\begin{array}{llll} \mathtt{p}: & \mathtt{p} \leftarrow \mathtt{d}, \mathtt{not}\ \mathtt{a} & \qquad \mathtt{d}: & \mathtt{d} \\ \mathtt{a}: & \mathtt{a} \leftarrow \mathtt{not}\ \mathtt{h} & \qquad \mathtt{h}: & \mathtt{h} \end{array}$$

  coincide and satisfy

$$\mathsf{lfp}(\Gamma_P^2)(\mathtt{a}) = \sim\mathtt{h}\cdot\mathtt{a} \qquad \mathsf{lfp}(\Gamma_P^2)(\mathtt{p}) = (\sim\mathtt{a} * \mathtt{d})\cdot\mathtt{p} + (\sim\sim\mathtt{h} * \mathtt{d})\cdot\mathtt{p}$$

- Each addend is a justification: $\sim\mathtt{h}\cdot\mathtt{a}$, $(\sim\mathtt{a} * \mathtt{d})\cdot\mathtt{p}$ and $(\sim\sim\mathtt{h} * \mathtt{d})\cdot\mathtt{p}$.
- If a justification contains a negated label (odd num. of times) is said to be inhibited.
  - The antidote a has not been administered because Bond was on a holiday h.

Well-founded causal model

- For instance, the least and greatest fixpoint of

$$\begin{array}{llll} \mathtt{p}: & \mathtt{p} \leftarrow \mathtt{d},\ \mathtt{not\ a} & \mathtt{d}: & \mathtt{d} \\ \mathtt{a}: & \mathtt{a} \leftarrow \mathtt{not\ h} & \mathtt{h}: & \mathtt{h} \end{array}$$

  coincide and satisfy

$$\text{lfp}(\Gamma_{\mathtt{P}}^2)(\mathtt{a}) \quad = \quad {\sim}\mathtt{h}{\cdot}\mathtt{a} \qquad\qquad \text{lfp}(\Gamma_{\mathtt{P}}^2)(\mathtt{p}) \quad = \quad ({\sim}\mathtt{a} * \mathtt{d}){\cdot}\mathtt{p} + ({\sim\sim}\mathtt{h} * \mathtt{d}){\cdot}\mathtt{p}$$

- Each addend is a justification: ${\sim}\mathtt{h}{\cdot}\mathtt{a}$, $({\sim}\mathtt{a} * \mathtt{d}){\cdot}\mathtt{p}$ and $({\sim\sim}\mathtt{h} * \mathtt{d}){\cdot}\mathtt{p}$.
- If a justification contains a negated label (odd num. of times) is said to be inhibited.
  - The antidote $\mathtt{a}$ has not been administered because Bond was on a holiday $\mathtt{h}$.
- Otherwise, a justification is said to be enabled.
  - Bond is paralyzed because $\mathtt{h}$ has enabled $\mathtt{d}$ to cause $\mathtt{p}$: $({\sim\sim}\mathtt{h} * \mathtt{d}){\cdot}\mathtt{p}$.

- The axiom $\sim(t \cdot u) = \sim(t * u)$ allow us to break justifications into inhibited and enabled:

$$
\begin{aligned}
(\sim(\sim h \cdot a) * d) \cdot p &= (\sim(\sim h * a) * d) \cdot p \\
&= ((\sim\sim h + \sim a) * d) \cdot p \\
&= (\sim\sim h * d) \cdot p \; + \; (\sim a * d) \cdot p
\end{aligned}
$$

Theorem

An atom is true, false of undefined in the standard well-founded model iff there is some ***enabled justification*** *for it.*

Theorem

Inhibited justifications become enabled justifications when the inhibitors are removed.

- For instance, $\sim\mathtt{h}\cdot\mathtt{a}$ is an inhibited justification for $\mathtt{a}$ in

$\mathtt{p}: \ \mathtt{p} \leftarrow \mathtt{d},\ \mathtt{not}\ \mathtt{a}$	$\mathtt{d}: \ \mathtt{d}$
$\mathtt{a}: \ \mathtt{a} \leftarrow \mathtt{not}\ \mathtt{h}$	$\mathtt{h}: \ \mathtt{h}$

and rule $\mathtt{a}$ is an enabled justification for $\mathtt{a}$ in

$\mathtt{p}: \ \mathtt{p} \leftarrow \mathtt{d},\ \mathtt{not}\ \mathtt{a}$	$\mathtt{d}: \ \mathtt{d}$
$\mathtt{a}: \ \mathtt{a} \leftarrow \mathtt{not}\ \mathtt{h}$	~~$\mathtt{h}: \ \mathtt{h}$~~

Undefined literals

- Consider the usual cycle

$$\mathtt{r_1}: \quad \mathtt{a} \leftarrow \mathtt{not\ b} \qquad\qquad \mathtt{r_2}: \quad \mathtt{b} \leftarrow \mathtt{not\ a}$$

Undefined literals

- Consider the usual cycle

$$r_1: \quad a \leftarrow not\ b \qquad\qquad r_2: \quad b \leftarrow not\ a$$

- The least and greatest fixpoint do not coincide.

$$\mathrm{lfp}(\Gamma_P^2)(a) \quad = \quad {\sim}r_2 \cdot r_1 \qquad\qquad \mathrm{gfp}(\Gamma_P^2)(a) \quad = \quad r_1$$

Undefined literals

- Consider the usual cycle

$$\mathtt{r}_1: \quad \mathtt{a} \leftarrow \mathtt{not\ b} \qquad\qquad \mathtt{r}_2: \quad \mathtt{b} \leftarrow \mathtt{not\ a}$$

- The least and greatest fixpoint do not coincide.

$$\mathrm{lfp}(\Gamma_P^2)(\mathtt{a}) = {\sim}\mathtt{r}_2 \cdot \mathtt{r}_1 \qquad \mathrm{gfp}(\Gamma_P^2)(\mathtt{a}) = \mathtt{r}_1$$

$$\mathbb{W}_P(\mathtt{a}) = {\sim}\mathtt{r}_2 \cdot \mathtt{r}_1 \qquad \mathbb{W}_P(\mathtt{not\ a}) = {\sim}\mathtt{r}_1$$

- Consider the usual cycle

$$\mathtt{r}_1: \quad \mathtt{a} \leftarrow \mathtt{not\ b} \qquad\qquad \mathtt{r}_2: \quad \mathtt{b} \leftarrow \mathtt{not\ a}$$

- The least and greatest fixpoint do not coincide.

$$\mathrm{lfp}(\Gamma_P^2)(\mathtt{a}) = {\sim}\mathtt{r}_2 \cdot \mathtt{r}_1 \qquad \mathrm{gfp}(\Gamma_P^2)(\mathtt{a}) = \mathtt{r}_1$$

$$\mathbb{W}_P(\mathtt{a}) = {\sim}\mathtt{r}_2 \cdot \mathtt{r}_1 \qquad \mathbb{W}_P(\mathtt{not\ a}) = {\sim}\mathtt{r}_1$$

- $\mathtt{a}$ is not true because of $\mathtt{r}_2$ and it is not false because of $\mathtt{r}_1$.

Undefined literals

- Consider the usual cycle

$$r_1: \quad a \leftarrow \text{not } b \qquad\qquad r_2: \quad b \leftarrow \text{not } a$$

- The least and greatest fixpoint do not coincide.

$$\text{lfp}(\Gamma_P^2)(a) = \sim r_2 \cdot r_1 \qquad \text{gfp}(\Gamma_P^2)(a) = r_1$$

$$\mathbb{W}_P(a) = \sim r_2 \cdot r_1 \qquad \mathbb{W}_P(\text{not } a) = \sim r_1$$

- a is not true because of r_2 and it is not false because of r_1.
- a is undefined because of rules r_1 and r_2.

$$\begin{aligned} \mathbb{W}_P(\text{undef } a) &= \sim\mathbb{W}_P(a) * \mathbb{W}_P(\text{not } a) \\ &= \sim(\sim r_2 \cdot r_1) * \sim\sim r_1 = \sim\sim r_1 * \sim\sim r_2 \end{aligned}$$

Outline

Causal Graph Justifications

- Causal Graph Justifications (CG) is an extension of the **stable model semantics** whereas ECJ is an extension of the **well-founded semantics**.

- Causal Graph Justifications (CG) is an extension of the **stable model semantics** whereas ECJ is an extension of the **well-founded semantics**.
- CG does **not capture enablers** nor **inhibited justifications**.

$$(\sim a * d) \cdot p \xrightarrow{\lambda^c} 0 \qquad\qquad (\sim\sim h * d) \cdot p \xrightarrow{\lambda^c} d \cdot p$$

- ECJ justifications can be mapped into CG justifications.
  - Removing all inhibited justifications and
  - Removing all enablers for the remaining ones.

CG stable models can be related to the Γ_P^2 fixpoints.

Theorem

For any enabled justification, there is a CG-justification w.r.t all stable models obtained by removing all enablers.

- The converse does not hold in general as happened with the standard well-founded and stable model semantics.

$r_1: \ a \leftarrow not\ b \qquad r_2: \ b \leftarrow not\ a, not\ c \qquad r_4: \ d \leftarrow b, not\ d$

$r_3: \ c \leftarrow a \qquad c: \ c$

- Two-valued standard well-founded model $\{a, c\}$. $\mathbb{W}_P(c) = c$. Unique standard stable model $\{a, c\}$. $I(c) = c + r_1 \cdot r_3$.

Outline

Why-not Provenance Justifications

- Why-not Provenance (WnP) Justifications
  - is also defined as an extension of **well-founded semantics**.

Why-not Provenance Justifications

- Why-not Provenance (WnP) Justifications
  - is also defined as an extension of **well-founded semantics**.
  - are just sets: it does not capture the causal structure (graph).

Why-not Provenance Justifications

- Why-not Provenance (WnP) Justifications
  - is also defined as an extension of **well-founded semantics**.
  - are just sets: it does not capture the causal structure (graph).
  - it **does not distinguish** between **enablers** and **real causes**.

Why-not Provenance Justifications

- Why-not Provenance (WnP) Justifications
  - is also defined as an extension of **well-founded semantics**.
  - are just sets: it does not capture the causal structure (graph).
  - it **does not distinguish** between **enablers** and **real causes**.
  - it **may require extra labels** asserting which facts cannot hold.

Why-not Provenance Justifications

- Why-not Provenance (WnP) Justifications
  - is also defined as an extension of **well-founded semantics**.
  - are just sets: it does not capture the causal structure (graph).
  - it **does not distinguish** between **enablers** and **real causes**.
  - it **may require extra labels** asserting which facts cannot hold.

$$(\sim\mathtt{a} * \mathtt{d}) \cdot \mathtt{p} \xrightarrow{\lambda^p} \neg\mathtt{a} \wedge \mathtt{d} \wedge \mathtt{p} \qquad (\sim\sim\mathtt{h} * \mathtt{d}) \cdot \mathtt{p} \xrightarrow{\lambda^p} \mathtt{h} \wedge \mathtt{d} \wedge \mathtt{p} \qquad \mathtt{d} \cdot \mathtt{p} \xrightarrow{\lambda^p} \mathit{not}(\mathtt{a}) \wedge \mathtt{d} \wedge \mathtt{p}$$

- ECJ justifications can be mapped into WnP justifications.
  - Replacing both, '$*$' and '$\cdot$', by '$\wedge$'.
  - Replacing double negated labels by positive ones $\sim\sim\mathtt{h} \Longrightarrow \mathtt{h}$
  - Adding $\mathit{not}(\mathtt{a})$ for each negative literal `not a` in the body of some rule in the justification.

Why-not Provenance Justifications

- WnP may contain more justifications that we call **hypothetical**.

- WnP may contain more justifications that we call **hypothetical**.
  - Suppose that Le Chiffre did not pour the drug and Bond is not on holidays.

$$\mathtt{p}: \quad \mathtt{p} \leftarrow \mathtt{d}, \mathtt{not\ a}$$
$$\mathtt{a}: \quad \mathtt{a} \leftarrow \mathtt{not\ h}$$

- WnP may contain more justifications that we call **hypothetical**.
  - Suppose that Le Chiffre did not pour the drug and Bond is not on holidays.

$$\texttt{p}: \ \texttt{p} \leftarrow \texttt{d}, \texttt{not a}$$
$$\texttt{a}: \ \texttt{a} \leftarrow \texttt{not h}$$

- Clearly, Bond is not paralyzed, but there is an hypothetical WnP justification $\neg \mathit{not}(\texttt{d}) \wedge \neg \mathit{not}(\texttt{h}) \wedge \mathit{not}(\texttt{a}) \wedge \texttt{p}$ meaning that p would have been true if we had added facts d and h to the program while not adding fact a.

Theorem

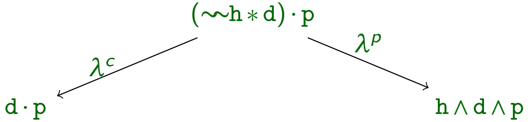
For any non-hypothetical WnP justification, there is a corresponding ECJ justification, and vice-versa.

- Hypothetical WnP justifications can be captured augmenting the program with facts

$$\sim\!\mathit{not}(\mathtt{a}) : \quad \mathtt{a}$$

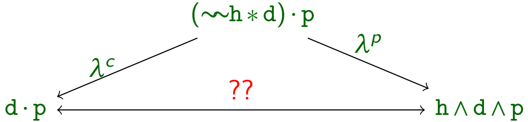
for every atom a which is not already a fact.

- ECJ justifications can be mapped into both, CG and WnP Justifications.



Why-not Provenance Justifications

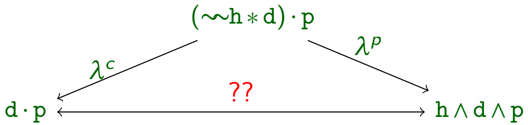
- ECJ justifications can be mapped into both, CG and WnP Justifications.



- Can we establish a direct correspondence between CJ and WnP?

Why-not Provenance Justifications

- ECJ justifications can be mapped into both, CG and WnP Justifications.

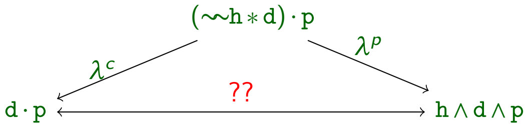


- Can we establish a direct correspondence between CJ and WnP?

Theorem

For any non-hypothetical and enabled WnP-justification, there is some CG-justification, w.r.t. to all causal stable models, such that the former contains all the vertices of the later (may contain more).

- ECJ justifications can be mapped into both, CG and WnP Justifications.



- Can we establish a direct correspondence between CJ and WnP?

Theorem

For any non-hypothetical and enabled WnP-justification, there is some CG-justification, w.r.t. to all causal stable models, such that the former contains all the vertices of the later (may contain more).

- As usual the converse does not hold.

Conclusions

- A **multivalued** extension of **well-founded semantics** that captures inhibitors, enablers and causes by introducing '∼' in the CG algebra.
- Allows distinguishing between **enablers** '∼∼a' and real **causes** 'a'.
- The existence of enabled justifications is a **sufficient and necessary condition** for the truth value of an atom.
- It **captures both WnP and CJ** justification.
- We established a **formal relation between WnP and CJ**.

Conclusions

- A **multivalued** extension of **well-founded semantics** that captures inhibitors, enablers and causes by introducing '∼' in the CG algebra.
- Allows distinguishing between **enablers** '∼∼a' and real **causes** 'a'.
- The existence of enabled justifications is a **sufficient and necessary condition** for the truth value of an atom.
- It **captures both WnP and CJ** justification.
- We established a **formal relation between WnP and CJ**.

Ongoing work

- Incorporate enablers and inhibitors to the **stable model semantics**.
- **Deriving new conclusions** from the cause-effect relations (ASPOCP 2015)
- **Non deterministic** causal laws: disjunctive and probabilistic LP

Enablers and Inhibitors in Causal Justifications of Logic Programs

Pedro Cabalar and Jorge Fandinno

Thanks for your attention!

September 28th, 2015
LPNMR'15
Lexington, KY, USA

- Programs with odd negative cycles may have standard stable model, but no Γ_P fixpoint

$$\begin{aligned} r_1 : p &\leftarrow \\ r_2 : p &\leftarrow \ \text{not}\ p \end{aligned}$$

- ... although they have Γ_P^2 fixpoints.

$$\text{lfp}(\Gamma_P^2)(p) \ = \ r_1 \qquad \qquad \text{gfp}(\Gamma_P^2)(p) \ = \ r_1 + {\sim} r_1 \cdot r_2$$

- p is clearly true because of r_1, but what about r_2?