Tao Gu, Fabio Zanasi
University College London

# Functorial Semantics as a Unifying Perspective on Logic Programming

- Background
- A string diagram perspective of PLP
- Application: transformation between PLP and BN
- Classical logic programs, Weighted logic program
- Future work

# Background

# Probabilistic logic programs (PLP)

- A PLP clause $\psi$ based on $At$ is $p :: A \leftarrow L_1, \ldots, L_m$ . , where:
  - $p \in [0,1]$
  - $A \in At$ is the head
  - $\{L_1 \ldots, L_m\} \subseteq At \cup \neg At$ is the body
- A PLP program $\mathbb{P} = \{\psi_1, \ldots, \psi_n\}$.

# PLP example

- **Example.** $\mathbb{P}_{wet}$ consists of the following 8 clauses:

$$
\begin{array}{rlll}
0.25 :: & \texttt{Winter} & \leftarrow . & (\psi_1) \\
0.2 :: & \texttt{Sprinkler} & \leftarrow \texttt{Winter}. & (\psi_2) \\
0.6 :: & \texttt{Rain} & \leftarrow \texttt{Winter}. & (\psi_3) \\
0.1 :: & \texttt{Rain} & \leftarrow \neg\texttt{Winter}. & (\psi_4)
\end{array}
\qquad
\begin{array}{rlll}
0.9 :: & \texttt{WetGrass} & \leftarrow \texttt{Sprinkler}. & (\psi_5) \\
0.8 :: & \texttt{WetGrass} & \leftarrow \texttt{Rain}. & (\psi_6) \\
0.7 :: & \texttt{SlipperyRoad} & \leftarrow \texttt{Rain}. & (\psi_7) \\
0.1 :: & \texttt{SlipperyRoad} & \leftarrow \neg\texttt{Rain}. & (\psi_8)
\end{array}
$$

# Semantics of acyclic PLP

- Idea: distribution semantics + negation as failure
- Distribution semantics
  - $\mathbb{P}$ determines a distribution $\mu_{\mathbb{P}}$ on $\mathscr{P}(\,|\,\mathbb{P}\,|\,)$
  - $Pr_{\mathbb{P}}(A) = \sum \{\mu_{\mathbb{P}}(\mathbb{Q}) \mid \mathbb{Q} \subseteq |\,\mathbb{P}\,|, \mathbb{Q} \vDash A\}$
- Negation as failure
  - $\neg A$ is true in $\mathbb{P}$ if it is *not* finitely derivable/provable in $\mathbb{P}$
- We focus on acyclic PLP:
  - Relatively simple semantics (imprecise probabilities needed).
  - Expressive power: equivalent as boolean-valued Bayesian networks.

## Functorial semantics

- F. W. Lawvere, *Functorial Semantics of Algebraic Theories*, Ph.D. thesis, Columbia University, 1963.
- Idea: algebras are functors (models) from syntax categories (e.g. Lawvere theories) to semantics categories (e.g. $\mathbf{Set}$).
- Example. Monoids $\simeq \mathrm{Prod}(\mathscr{L}_{Mon}^{op}, \mathbf{Set})$, where $\mathscr{L}_{Mon}$ has natural numbers as objects, and morphisms $n \to m$ are $\langle t_1, \ldots, t_n \rangle$ of free monoids on $\{x_1, \ldots, x_m\}$.

# Functorial semantics of PLP

# Categorical perspective of PLP

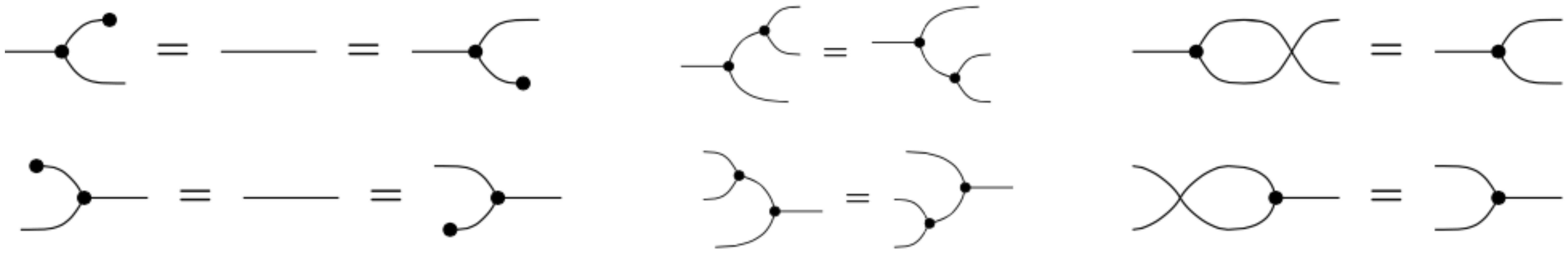
- We separate the syntax and semantics:
- The syntax describes the inferential structure
    - A definite logic program $\mathbb{L} := [\mathbb{P}]$ describes the inferential structure of $\mathbb{P}$: the inferential structure of $p :: A \leftarrow L_1, \ldots, L_m$ . is clause $A \leftarrow B_1, \ldots, B_m$ . , where $B_i$ is the atom in literal $L_i$.
    - Example. The inferential structure of $A \leftarrow B_1, \neg B_2$ . is $A \leftarrow B_1, B_2$ . ($[A \leftarrow B_1, \neg B_2.] = A \leftarrow B_1, B_2$ . )
- The semantics determines the meaning of clauses
- Both are symmetric monoidal categories with additional structure.
- *Functorial* semantics: PLPs are exactly structure-preserving functors

F : SynCat –> SemCat

# CDMU category

- Definition. Every object has copier, discarder, multiplication, unit morphisms

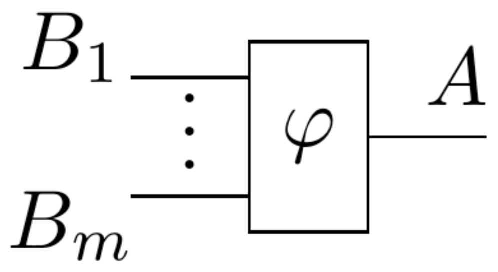
- satisfying:



- Both the syntax and semantics categories of PLP are CDMU categories.

# Syntax category

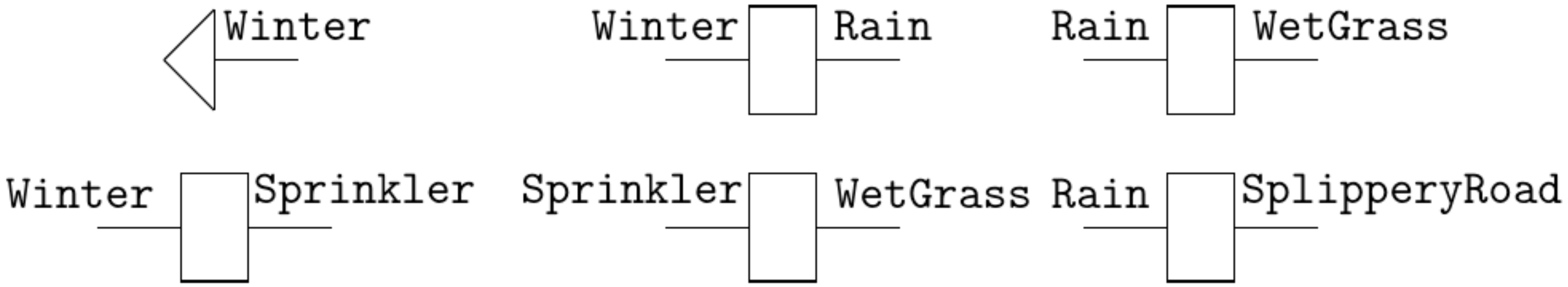
- $\mathbb{L} := [\mathbb{P}]$ determines a syntax category $\mathsf{SynPLP}_{\mathbb{L}} := \mathsf{freeCDMU}(At, \Sigma_{\mathbb{L}})$, where $\Sigma_{\mathbb{L}}$ is the set of generating morphisms, one for each $\mathbb{L}$-clause:

$$\Sigma_{\mathbb{L}} := \{$$  $$\mid \varphi \equiv A \leftarrow B_1, \ldots, B_m \text{ . is a clause in } \mathbb{L}\}$$



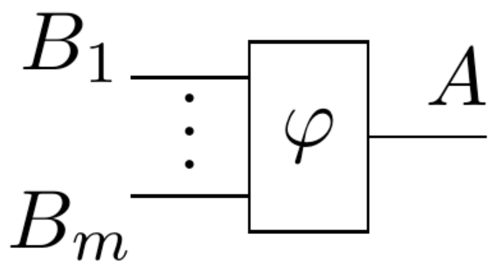
  - Objects: finite lists over $At$.
  - Morphisms: freely composed string diagrams using $\Sigma_{\mathbb{L}}$ and CDMU structure, modulo CDMU equations.

- Example. $\Sigma_{\mathbb{L}_{wet}}$ consists of 6 boxes:



## Syntax category

- $\mathbb{L} := [\mathbb{P}]$ determines a syntax category $\mathrm{SynPLP}_{\mathbb{L}} := \mathrm{freeCDMU}(At, \Sigma_{\mathbb{L}})$, where $\Sigma_{\mathbb{L}}$ is the set of generating morphisms, one for each $\mathbb{L}$-clause:

$\Sigma_{\mathbb{L}} := \{$  $\mid \varphi \equiv A \leftarrow B_1, \ldots, B_m .$ is a clause in $\mathbb{L}\}$

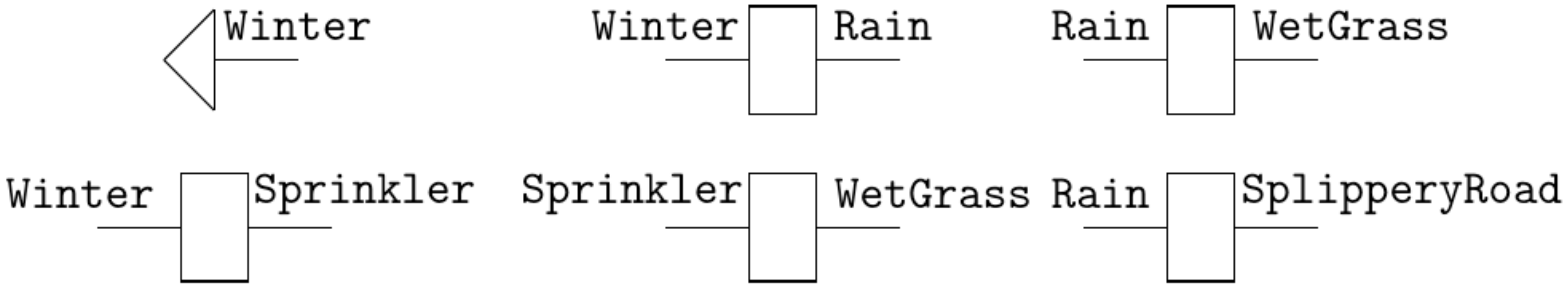


  - Objects: finite lists over $At$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.25 :: | `Winter` | $\leftarrow$ . | $(\psi_1)$ | 0.9 :: | `WetGrass` | $\leftarrow$ `Sprinkler`. | $(\psi_5)$ |
| 0.2 :: | `Sprinkler` | $\leftarrow$ `Winter`. | $(\psi_2)$ | 0.8 :: | `WetGrass` | $\leftarrow$ `Rain`. | $(\psi_6)$ |
| 0.6 :: | `Rain` | $\leftarrow$ `Winter`. | $(\psi_3)$ | 0.7 :: | `SlipperyRoad` | $\leftarrow$ `Rain`. | $(\psi_7)$ |
| 0.1 :: | `Rain` | $\leftarrow \neg$`Winter`. | $(\psi_4)$ | 0.1 :: | `SlipperyRoad` | $\leftarrow \neg$`Rain`. | $(\psi_8)$ |

  - Morphisms: freely composed string diagrams using $\Sigma_{\mathbb{L}}$ and CDMU structure, modulo CDMU equations.

- **Example.** $\Sigma_{\mathbb{L}_{wet}}$ consists of 6 boxes:



# Semantics category

- $Kl(\mathscr{D})(\mathbf{2})$
  - objects: finite products of $\mathbf{2}_A = \{0_A, 1_A\}$.
  - morphisms $X \to Y$ are functions $X \to \mathscr{D}(Y)$.
- $Kl(\mathscr{D})(\mathbf{2})$ is a CDMU category:

$$
\begin{array}{rl}
[\![ -\!\!\bullet\!\!<_A ]\!]_{\mathbb{P}} : \mathbf{2}_A \to \mathbf{2}_A \times \mathbf{2}_A \\
x \mapsto 1|(x,x)\rangle
\end{array}
\quad
\begin{array}{rl}
[\![ -\!\!\bullet_A ]\!]_{\mathbb{P}} : \mathbf{2}_A \to \mathbf{1} \\
x \mapsto 1|*\rangle
\end{array}
\quad
\begin{array}{rl}
[\![ >\!\!\bullet\!\!-_A ]\!]_{\mathbb{P}} : \mathbf{2}_A \times \mathbf{2}_A \to \mathbf{2}_A \\
(x,y) \mapsto 1|x \vee y\rangle
\end{array}
\quad
\begin{array}{rl}
[\![ \bullet\!\!-_A ]\!]_{\mathbb{P}} : \mathbf{1} \to \mathbf{2}_A \\
* \mapsto 1|\neg A\rangle
\end{array}
$$

# Functorial semantics of PLP

- **Proposition.** PLP programs based on $\mathbb{L} \simeq$ structure-preserving functors $[[ - ]]_{\mathbb{P}} : \mathrm{SynPLP}_{\mathbb{L}} \to \mathcal{K}l(\mathscr{D})(\mathbf{2})$ mapping generating objects to generating objects.

- Idea: $\mathbb{P}$ determines the functor $[[ - ]]_{\mathbb{P}}$'s action on the generating morphisms/boxes, namely the probability labels of the clauses.

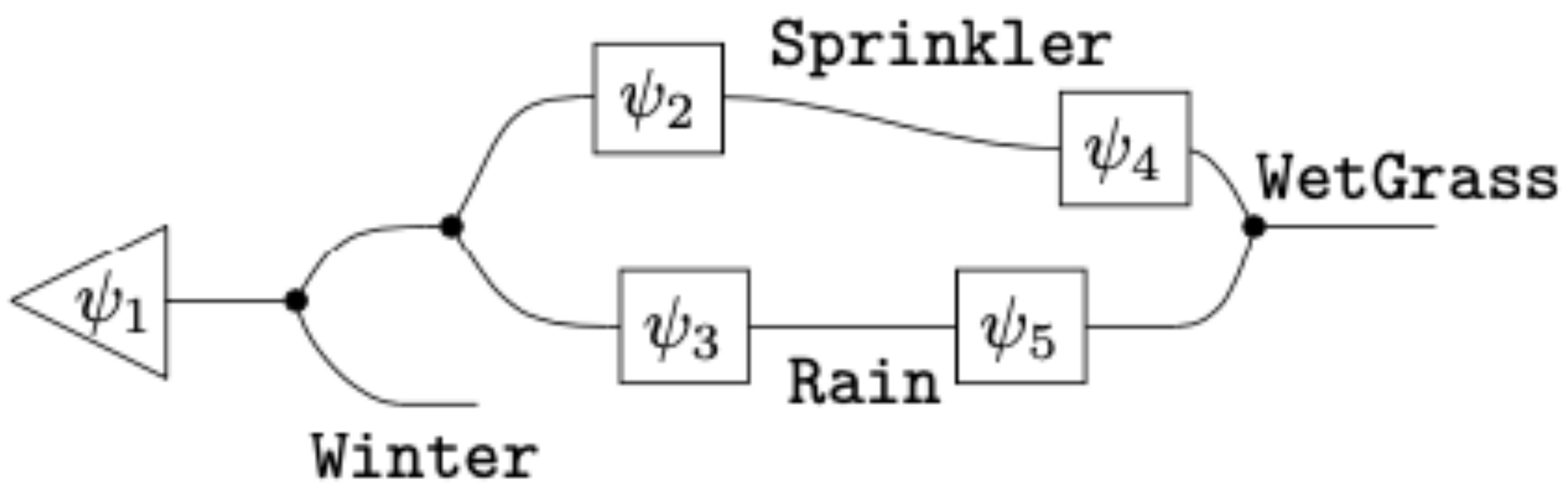
- **Example.** $\mathbb{P}_{wet}$ defines the functor $[[ - ]]_{\mathbb{P}_{wet}}$ such that:


$\xmapsto{[[ - ]]_{\mathbb{P}_{wet}}}$

$$0_{Rain} \mapsto 1\,|\,0_{WetGrass}\rangle$$
$$1_{Rain} \mapsto 0.8\,|\,1_{WetGrass}\rangle + 0.2\,|\,0_{WetGrass}\rangle$$

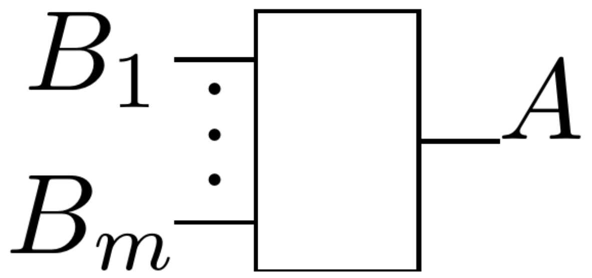
# Atom probability

- A compositional diagrammatic presentation of atom probabilities.
- Example. the $[[ - ]]_{\mathbb{P}_{wet}}$-image of the following string diagram is $Pr(\mathtt{Winter}, \mathtt{WetGrass})$ in $\mathscr{D}(\mathbf{2}_{\mathtt{Winter}} \times \mathbf{2}_{\mathtt{WetGrass}})$.



# Application: transformations between PLP and BN

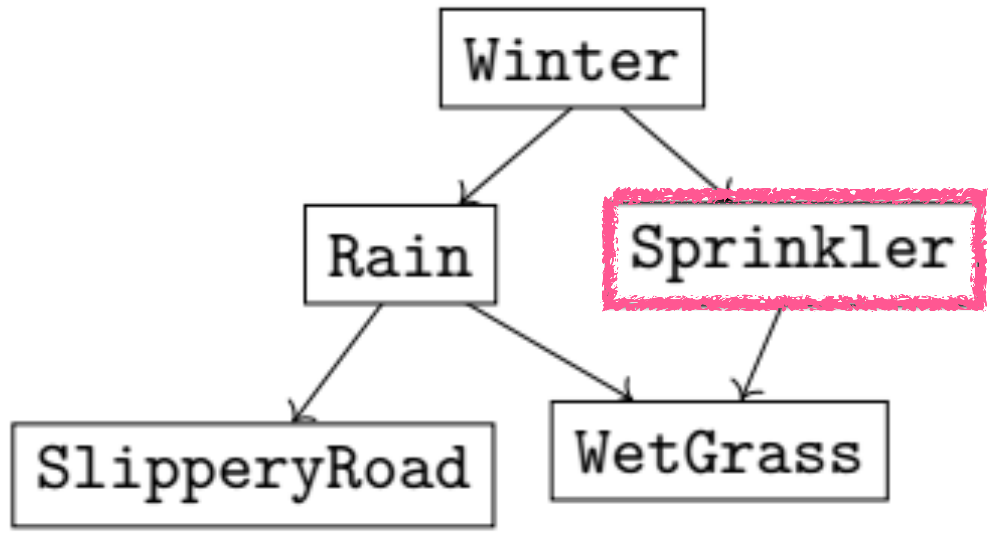
## Functorial semantics of BN

- Idea:
  - Syntax: DAGs –> string diagrams. $\mathsf{SynBN}_G$ for a DAG $G$ is $\mathsf{FreeCD}(V_G, \Sigma_G)$, where $\Sigma_G = \{$ 

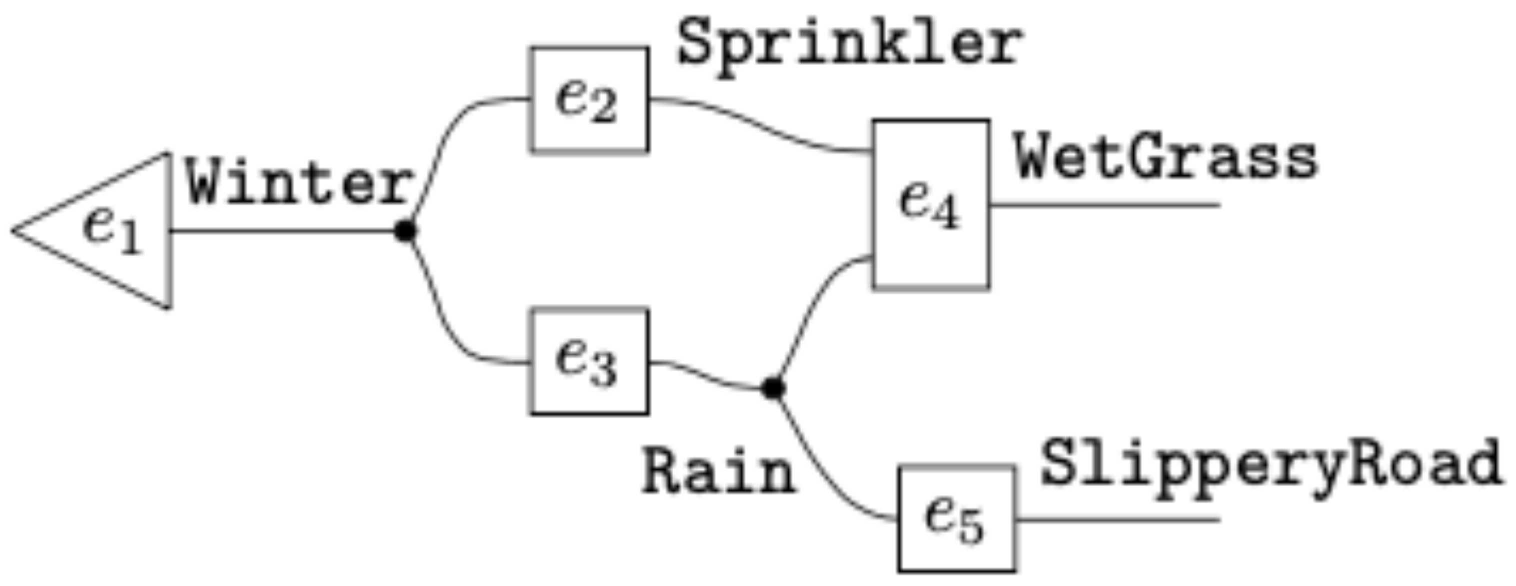
$\mid pa(A) = \{B_1, \ldots, B_m\}\}$
  - Semantics: category of stochastic processes.
- Proposition ([JKZ2019]). Boolean-valued Bayesian networks based on $G \simeq$ structure-preserving functors $[[\,-\,]]_{\mathbb{B}} : \mathsf{SynBN}_G \to \mathcal{Kl}(\mathcal{D})(\mathbf{2})$.

# BN example

- BN $\mathbb{B}_{wet}$:




| | | Pr(WetGrass) |
|---|---|---|
| ¬Sprinkler | ¬Rain | 0 |
| ¬Sprinkler | Rain | 0.8 |
| Sprinkler | ¬Rain | 0.9 |
| Sprinkler | Rain | 0.98 |




$G_{wet}$ as string diagram

$\mathbf{2}_{Sprinkler} \times \mathbf{2}_{Rain} \to \mathscr{D}(\mathbf{2}_{WetGrass})$ such that:

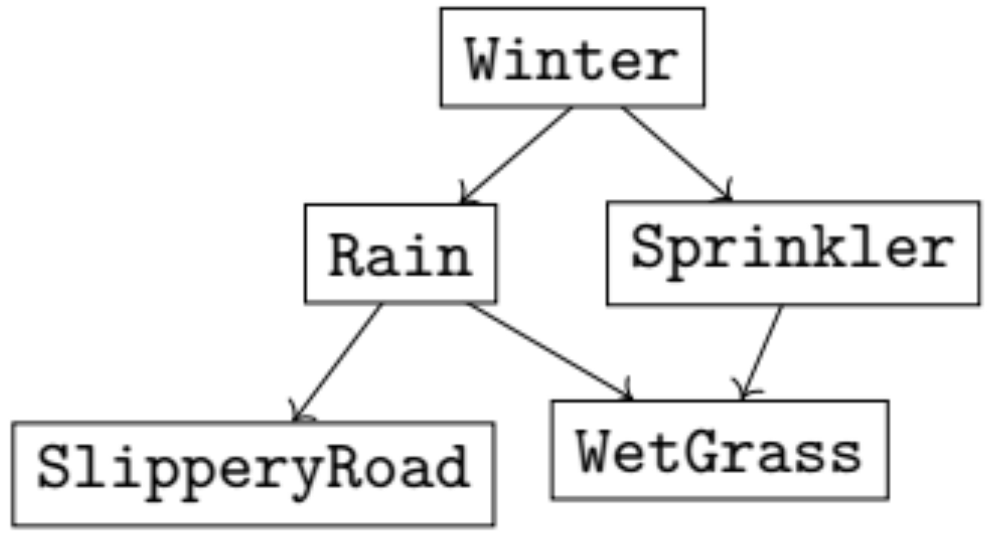
$$(0_S,0_R) \mapsto 1\,|\,0_W\rangle$$
$$(0_S,1_R) \mapsto 0.8\,|\,0_W\rangle + 0.2\,|\,0_W\rangle$$
$$(1_S,0_R) \mapsto 0.9\,|\,0_W\rangle + 0.1\,|\,0_W\rangle$$
$$(1_S,1_R) \mapsto 0.98\,|\,0_W\rangle + 0.02\,|\,0_W\rangle$$

# PLP and BN

- Fact. Every boolean-valued Bayesian network can be encoded as an acyclic ground PLP, and vice versa.
- Categorically,

$$
\begin{tikzcd}
\mathsf{SynLP}_{\mathbb{L}} \arrow[dr, "{[[-]]_{\mathbb{P}}}"'] & \mathsf{SynBN}_G \arrow[l, "\mathscr{F}"'] \arrow[r, "\iota"] \arrow[d, "{[[-]]_{\mathbb{B}}}"'] \arrow[dl, phantom, "(1)"] \arrow[dr, phantom, "(2)"] & \mathsf{SynLP}_{\mathbb{L}'} \arrow[ll, bend right, "(3)"'] \arrow[d, "{[[-]]_{\mathbb{P}'}}"] \\
 & \mathbf{Stoch}(2) \arrow[r, equal] & \mathbf{Stoch}(2)
\end{tikzcd}
$$

# From BN to PLP




| | | Pr(WetGrass) |
|---|---|---|
| ¬Sprinkler | ¬Rain | 0 |
| ¬Sprinkler | Rain | 0.8 |
| Sprinkler | ¬Rain | 0.9 |
| Sprinkler | Rain | 0.98 |

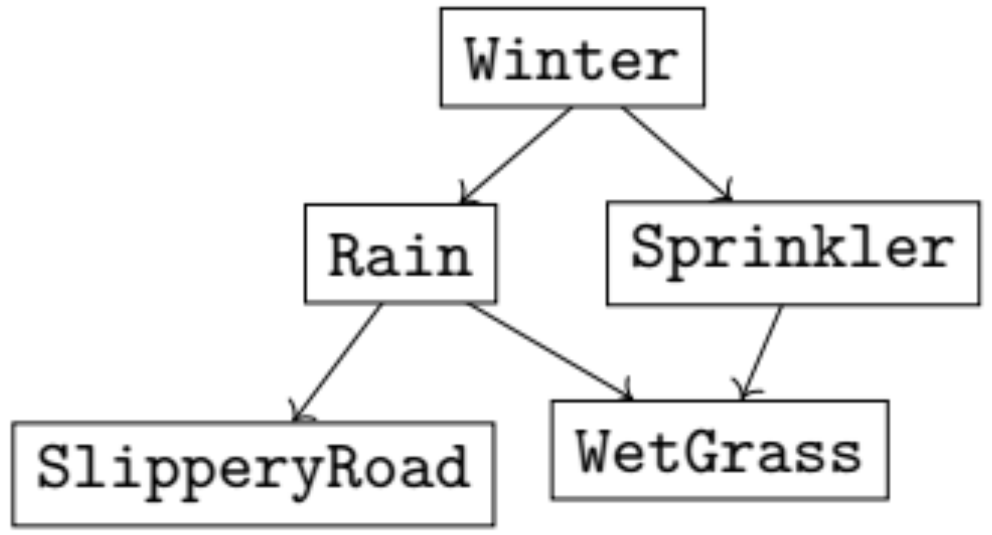
$$\begin{aligned} 0 &:: \mathsf{WetGrass} \leftarrow \neg\mathsf{Sprinkler}, \neg\mathsf{Rain}\,. \\ 0.8 &:: \mathsf{WetGrass} \leftarrow \neg\mathsf{Sprinkler}, \mathsf{Rain}\,. \\ 0.9 &:: \mathsf{WetGrass} \leftarrow \mathsf{Sprinkler}, \neg\mathsf{Rain}\,. \\ 0.98 &:: \mathsf{WetGrass} \leftarrow \mathsf{Sprinkler}, \mathsf{Rain}\,. \end{aligned}$$

- Start from $\mathsf{SynBN}_G = \mathsf{freeCD}(V_G, \Sigma_G)$, define $\mathsf{SynPLP}_{\mathbb{L}}$ as $\mathsf{freeCDMU}(V_G, \Sigma_G)$.
- $[[-]]_{\mathbb{P}}$ is $[[-]]_{\mathbb{B}}$ plus (routine) interpretation of multiplication and unit.
-

$$\begin{tikzcd} \mathsf{SynBN}_G \arrow[r, "\iota"] \arrow[d, "{[[-]]_{\mathbb{B}}}"'] \arrow[dr, phantom, "(2)"] & \mathsf{SynLP}_{\mathbb{L}'} \arrow[d, "{[[-]]_{\mathbb{P}'}}"] \\ \mathbf{Stoch(2)} \arrow[r, equal] & \mathbf{Stoch(2)} \end{tikzcd}$$

# From PLP to BN

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.25 :: | Winter | $\leftarrow$ . | ($\psi_1$) | 0.9 :: | WetGrass | $\leftarrow$ Sprinkler. | ($\psi_5$) |
| 0.2 :: | Sprinkler | $\leftarrow$ Winter. | ($\psi_2$) | 0.8 :: | WetGrass | $\leftarrow$ Rain. | ($\psi_6$) |
| 0.6 :: | Rain | $\leftarrow$ Winter. | ($\psi_3$) | 0.7 :: | SlipperyRoad | $\leftarrow$ Rain. | ($\psi_7$) |
| 0.1 :: | Rain | $\leftarrow$ ¬Winter. | ($\psi_4$) | 0.1 :: | SlipperyRoad | $\leftarrow$ ¬Rain. | ($\psi_8$) |



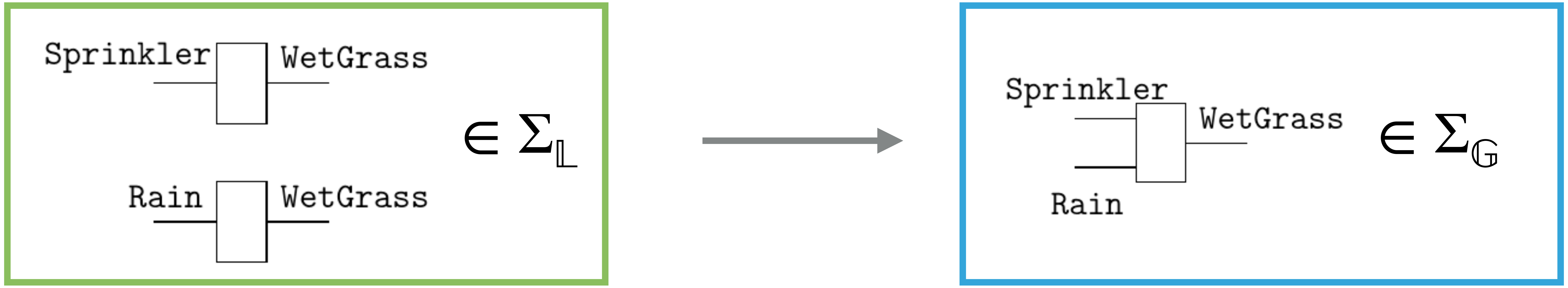

| | | Pr(WetGrass) |
|---|---|---|
| ¬Sprinkler | ¬Rain | 0 |
| ¬Sprinkler | Rain | 0.8 |
| Sprinkler | ¬Rain | 0.9 |
| Sprinkler | Rain | 0.98 |

# From PLP to BN

- Observation. The PLP-to-BN transformation is syntactical.
- In words, to obtain the corresponding BN model (functor) of a PLP model (functor), it suffices to define a suitable functor between the two syntax categories.

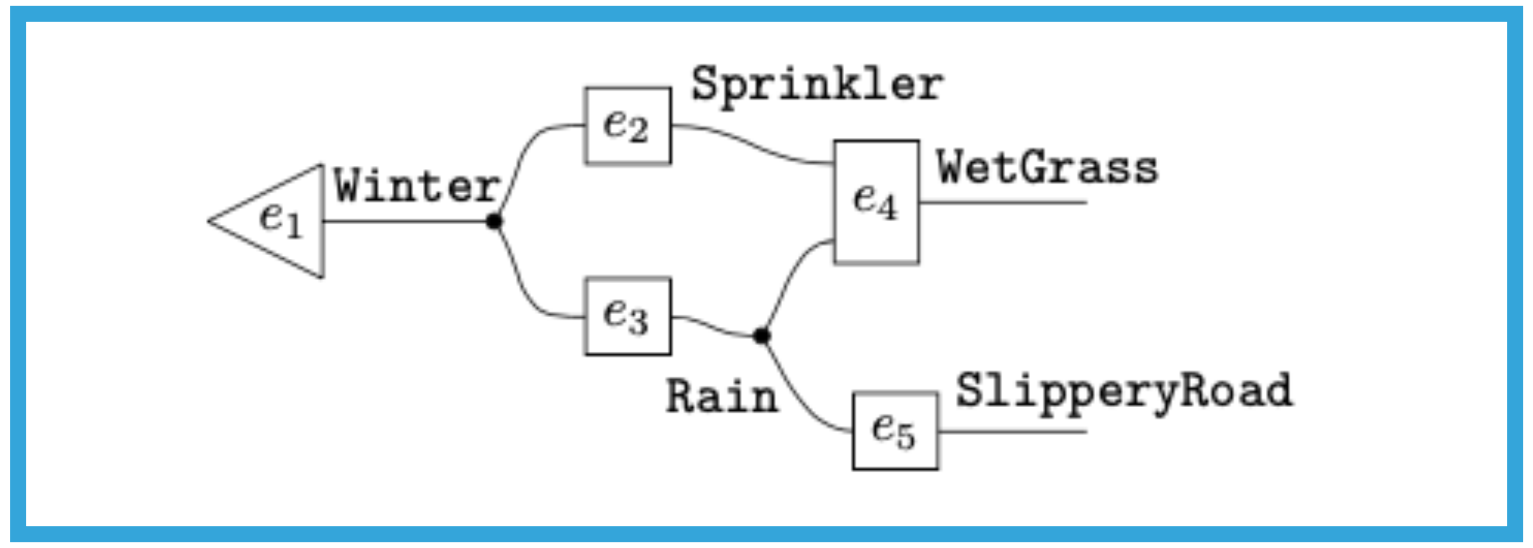
## Step 1

- We define a BN-syntax category from the PLP-syntax category, by combining all generating boxes in the latter with the same codomain into a single generating box in the former.
- Example.

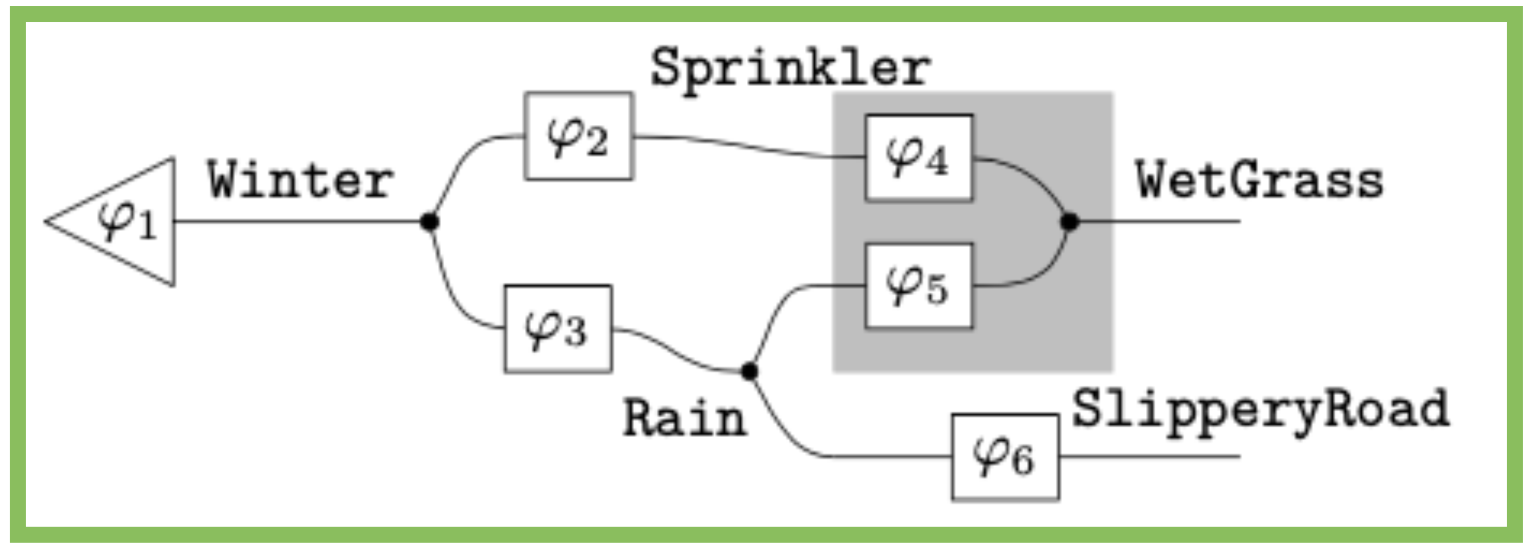


# Step 2

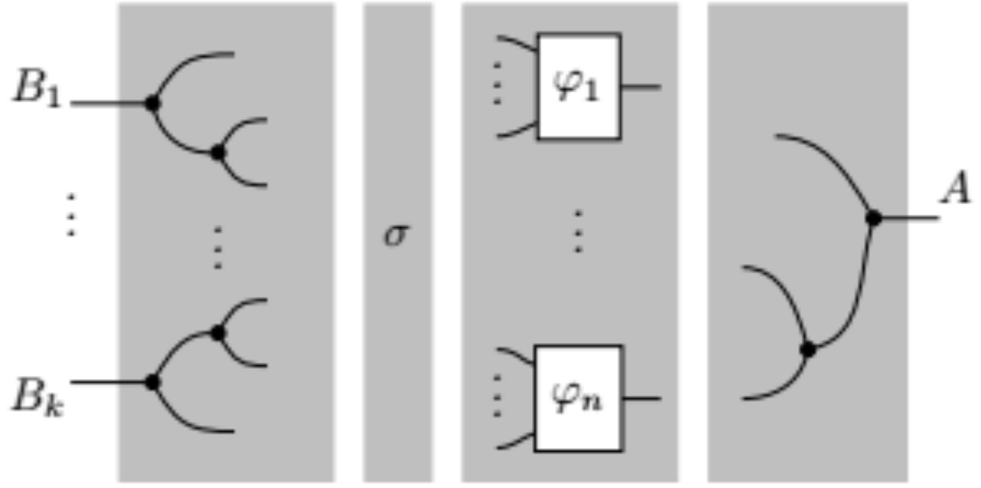
- We inductively define a functor $\mathscr{F}$ between the two syntax categories.
- Example.




$\xmapsto{\mathscr{F}}$



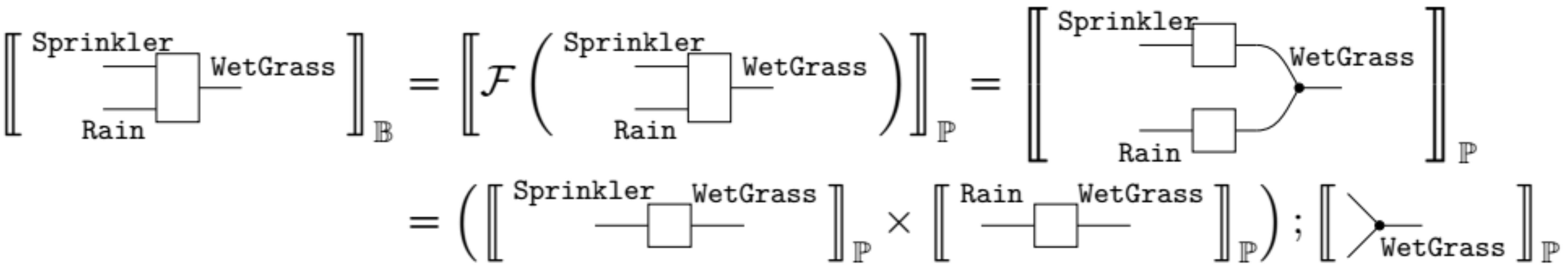

- In general, $pa(A) \to A$ is mapped to $comp(A)$:



*comp(A)*

# Step 3

- We obtain the BN model by composition: $[[\,-\,]]_{\mathbb{B}} = [[\,-\,]]_{\mathbb{P}} \circ \mathscr{F}$
- Example.




- For instance, $(0_{\texttt{Sprinkler}}, 1_{\texttt{Rain}}) \mapsto 0.8\,|\,1_{\texttt{WetGrass}}\rangle + 0.2\,|\,0_{\texttt{WetGrass}}\rangle$

$$\begin{tikzcd}
\mathrm{SynLP}_{\mathbb{L}} \arrow[dr, "{[[\,-\,]]_{\mathbb{P}}}"'] & & \mathrm{SynBN}_{G} \arrow[ll, "\mathscr{F}"'] \arrow[dl, "{[[\,-\,]]_{\mathbb{B}}}"] \\
& \mathbf{Stoch(2)} \arrow[u, phantom, "(1)"] &
\end{tikzcd}$$

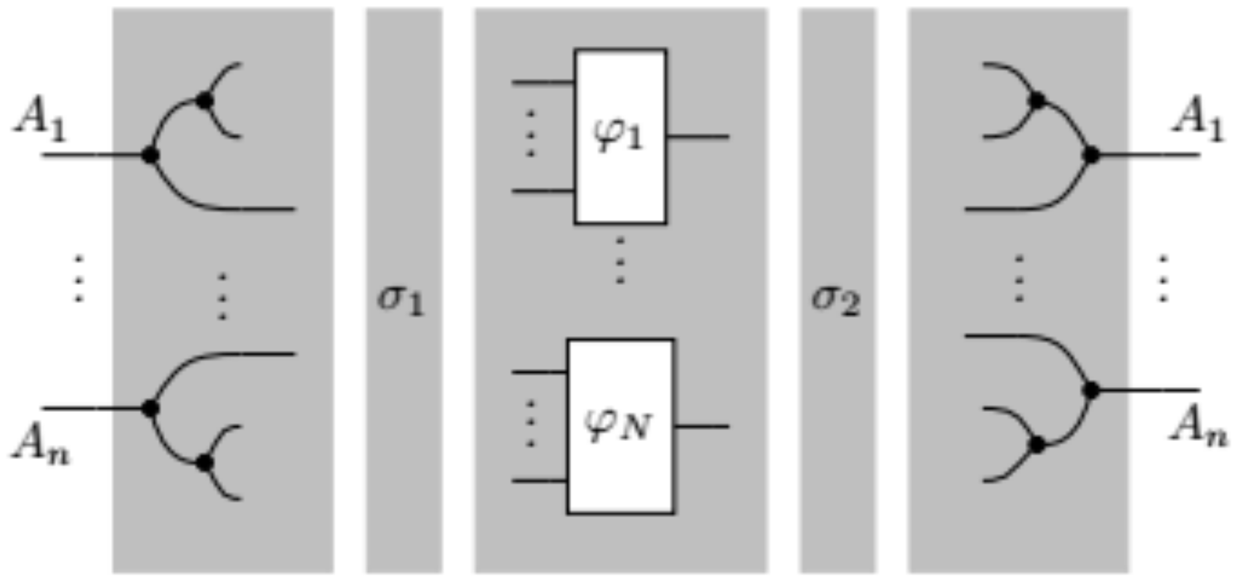
# Classical, weighted LP

# Idea

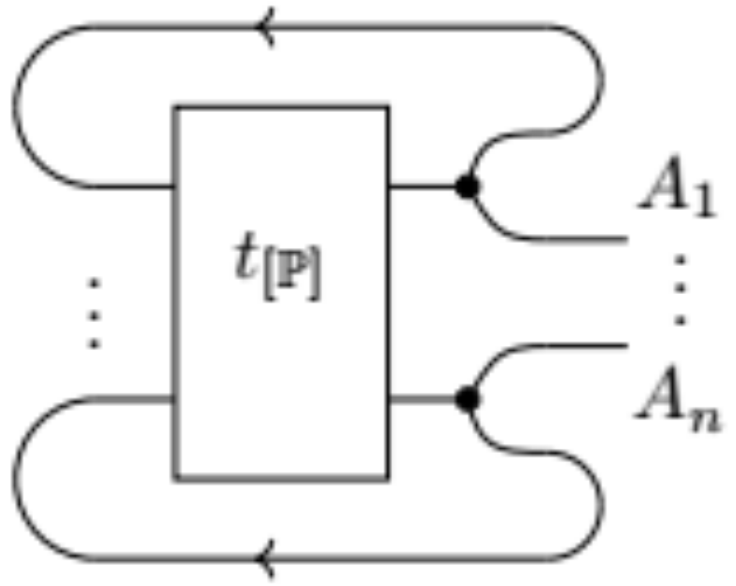
- Syntax categories: remains the same.
- Semantics categories:
  - Classical LP: $\mathbf{Set}(\mathbf{2})$
  - Probabilistic LP: $\mathcal{Kl}(\mathscr{D})(\mathbf{2})$
  - Weighted LP: $\mathbf{Set}(\mathscr{K})$
- Interestingly, $\mathcal{Kl}(\mathscr{D})(\mathbf{2})$ and $\mathbf{Set}(\mathscr{K})$ are both variants of $\mathbf{Set}(\mathbf{2})$, but in different flavour: $\mathcal{Kl}(\mathscr{D})(\mathbf{2})$ is some 'Kleisli category' (morphisms are different), $\mathbf{Set}(\mathscr{K})$ changes the 'base' (objects are different).

# Fixed-point style semantics

- We enrich the syntax categories to include ‘feedback wires’.
- Consequently the semantics categories move from functions to relations.
- Example.




immediate consequence operator $T_{\mathbb{P}}$




supported models

# Future work

- General case: variables may occur.
- Other logic-programming formalisms, e.g. CP-logic, LPAD.
- Diagrammatically represent PLP inference tasks, e.g. maximum a posteriori (MAP), most likely explanation (MEP), probabilistic inductive logic programming (PILP).

# Thank you!

## Reference


- F. W. Lawvere, *Functorial Semantics of Algebraic Theories*, Ph.D. thesis, Columbia University, 1963.
- Brendon Fong. Causal theories: A categorical perspective of on bayesian networks. 2013. arXiv:1301.6201.
- Bart Jacobs, Aleks Kissinger, and Fabio Zanasi. Causal inference by string diagram surgery. CoRR, abs/1811.08338, 2018.