KRR with Uncertainty

LP^{MLN} Relationships to Other Languages

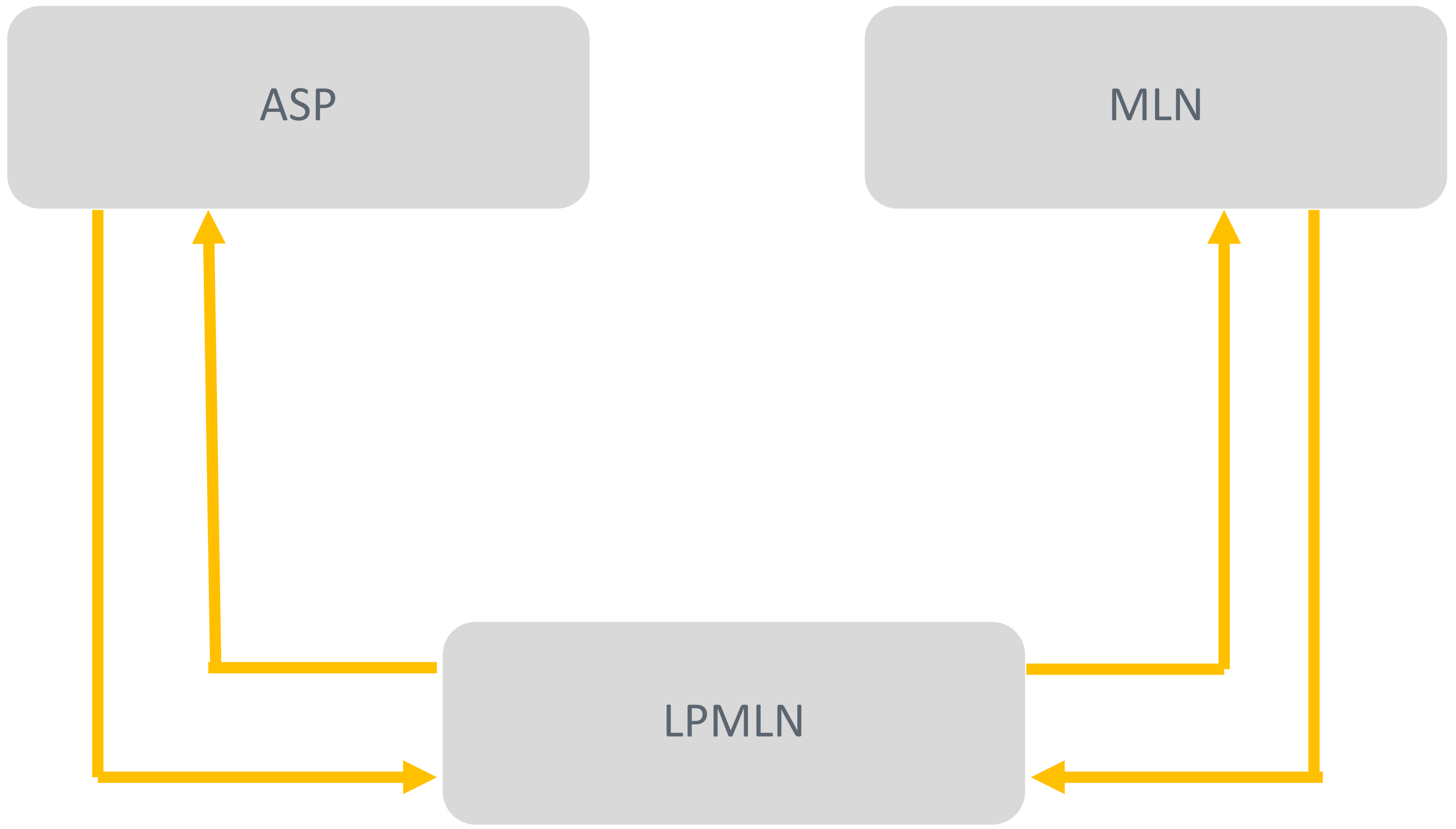
Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

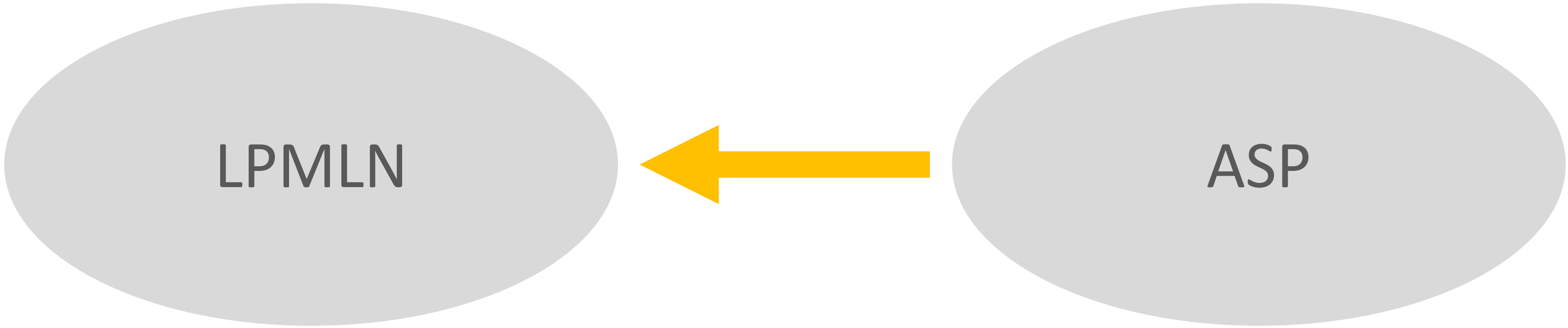
Objective

Explain the relationships between LPMLN and other languages

LPMLN vs. ASP vs. MLN



From ASP to LP^{MLN}



ASP as a Special Case of LP^{MLN}

- **Any answer set program Π can be viewed as a special case of an LP^{MLN} program P_Π by assigning the infinite weight to each rule**

Π	$p \leftarrow \text{not } q$	P_Π	$\alpha:\ p \leftarrow \text{not } q$
	$q \leftarrow \text{not } p$		$\alpha:\ q \leftarrow \text{not } p$

$\{p\}$
$\{q\}$

P_Π

I	$W(I)$	$P(I)$
$\emptyset$	e^0	0
$\{p\}$	$e^{2\alpha}$	$\frac{1}{2}$
$\{q\}$	$e^{2\alpha}$	$\frac{1}{2}$
$\{p, q\}$	0	0

- **Theorem:** **For any answer set program Π, the (deterministic) stable models of Π are exactly the (probabilistic) stable models of LP^{MLN} program P_Π whose weight is $e^{k\alpha}$, where k is the number of all ground rules in Π**

Example

- If Π has at least one (deterministic) stable model, then all (probabilistic) stable models of P_Π have the same probability, and are thus the stable models of Π as well

- **Q**: What if Π has no stable models?

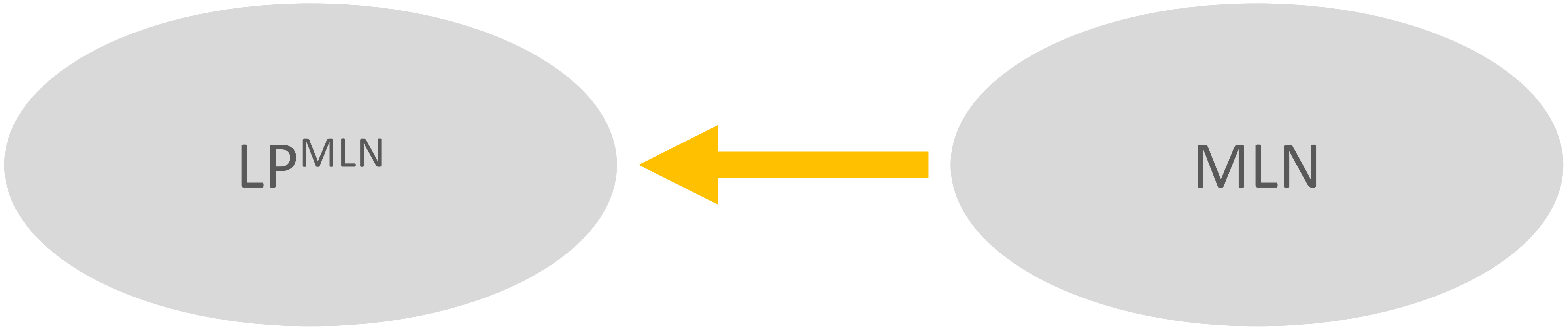
$$
\begin{array}{ll}
\Pi & \text{Bird(Jo)} \leftarrow \text{ResidentBird(Jo)} \\
 & \text{Bird(Jo)} \leftarrow \text{MigratoryBird(Jo)} \\
 & \bot \leftarrow \text{ResidentBird(Jo), MigratoryBird(Jo)} \\
 & \text{ResidentBird(Jo)} \\
 & \text{MigratoryBird(Jo)}
\end{array}
\qquad
\begin{array}{ll}
P_\Pi & \alpha:\ \text{Bird(Jo)} \leftarrow \text{ResidentBird(Jo)} \\
 & \alpha:\ \text{Bird(Jo)} \leftarrow \text{MigratoryBird(Jo)} \\
 & \alpha:\ \bot \leftarrow \text{ResidentBird(Jo), MigratoryBird(Jo)} \\
 & \alpha:\ \text{ResidentBird(Jo)} \\
 & \alpha:\ \text{MigratoryBird(Jo)}
\end{array}
$$

Correction note: There is a typo in the annotation below; the last stable model should be "{B(Jo), R(Jo), M(Jo)}".

- **Q**: What are the stable models P_Π?

{B(Jo), R(Jo)}, {B(Jo), M(Jo)}, {B(Jo)}

From MLN to LP^{MLN}



Embedding Propositional Logic in ASP

- **Theorem. For any propositional formula F of a finite signature σ, X is a model of $\boxed{\text{F}}$ iff X is a stable model of $\boxed{F \wedge Ch}$ where Ch is the conjunction of the choice rules $\{\sigma\}^{ch}$.**
  - The effect of adding the choice rules is to exempt A from minimization under the stable model semantics

$$F \;=\; p \leftarrow \neg q$$

models of F: $\boxed{\{p\},\quad \{q\},\quad \{p, q\}}$

stable models of F: $\{p\}$

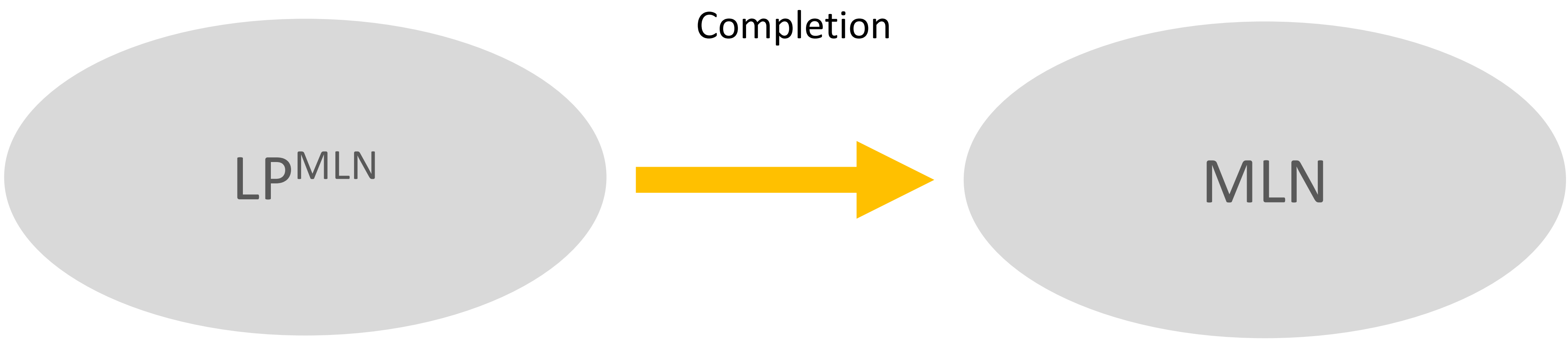
stable models of $F \wedge \{p; q\}^{ch}$: $\boxed{\{p\},\ \{q\},\ \{p, q\}}$

Embedding MLN in LP$^{\text{MLN}}$

- For any MLN L, LP$^{\text{MLN}}$ program Π_L is obtained from L by adding w : $\{A\}^{ch}$ for every ground atom A of σ and any weight w

- Theorem: Any MLN L and its LP$^{\text{MLN}}$ counterpart Π_L have the same probability distribution over all interpretations

From LP^{MLN} to MLN



Turning LP^{MLN} into MLN (1 of 2)

- We first consider how to turn an ASP program into a propositional formula.

- **Completion** is a process that turns an ASP program Π into a propositional formula F so that the stable models of Π are precisely the models of F.

Turning LP^{MLN} into MLN (2 of 2)

- The process works only for "tight" ASP programs (defined later).

- The method can be generalized to turning an LP^{MLN} program Π into an MLN program L so that the probabilistic answer sets of Π are precisely the models of L with the same probability distribution.

Completion

For any ground ASP program Π that consists of rules of the form

- $A \leftarrow Body$
- where A is an atom and $Body$ is a formula,

The completion of Π is defined as the union of Π and

$$A \rightarrow \bigvee_{A \leftarrow Body \in \Pi} Body$$

for each ground atom A

Theorem: For any "tight" answer set program Π, the stable models of Π are exactly the models of the completion of Π .

Example 1

Stable models of

$$p \leftarrow \neg q$$
$$q \leftarrow \neg p$$

$\{p\}$
$\{q\}$

$A \rightarrow \bigvee$ Body

$A \leftarrow$ Body

Models of completion

$p \leftarrow \neg q$
$q \leftarrow \neg p$
$p \rightarrow \neg q$
$q \rightarrow \neg p$

$\boxed{p \leftrightarrow \neg q}$

$\{p\}$
$\{q\}$

Example 2

Stable models of

$$p \leftarrow \neg q$$
$$q \leftarrow \neg r$$

$\{q\}$

$A \rightarrow \bigvee \text{Body}$

$A \leftarrow \text{Body}$

Models of completion

$p \leftarrow \neg q$
$q \leftarrow \neg r$
$p \rightarrow \neg q$
$q \rightarrow \neg r$
$r \rightarrow \bot$

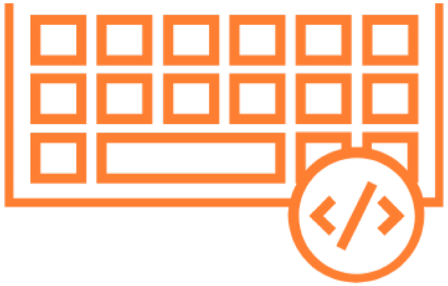
$p \leftrightarrow \neg q$
$q \leftrightarrow \neg r$
$\neg r$

$\{q\}$

Tight Programs

Theorem: For any "tight" answer set program Π, the answer sets of Π are exactly the models of the completion of Π

What would go wrong if Π is non-tight?



Completion and Non-tight programs

$$p \leftarrow q$$
$$q \leftarrow p$$

Models: $\emptyset$, $\{p, q\}$

Stable models: $\emptyset$

Completion:

$$\left.\begin{array}{l} p \leftarrow q \\ q \leftarrow p \\ p \rightarrow q \\ q \rightarrow p \end{array}\right\} \Leftrightarrow \quad p \Leftrightarrow q$$

Models of Comp: $\emptyset$, $\{p, q\}$

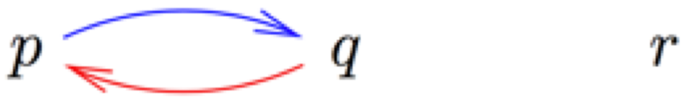
Positive Dependency Graph

- A program is a finite set of rules of the form

$$a \leftarrow \underbrace{a_1, \ldots, a_m,}_{P} \underbrace{\textit{not } a_{m+1}, \ldots, \textit{not } a_n}_{N} .$$

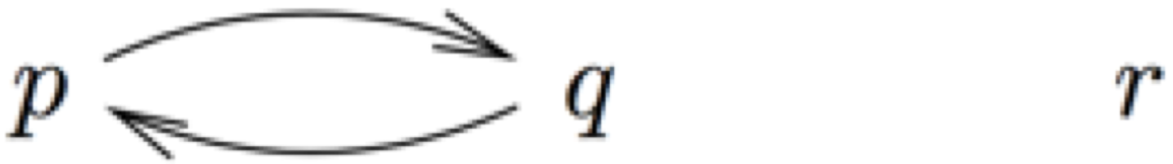
- The positive dependency graph of Π is the directed graph such that
  - its vertices are the atoms occurring in Π, and
  - for each $a \leftarrow P, N$ in Π, its edges go from a to each atom in P.

$$\begin{aligned} \Pi_1 : \quad & p \leftarrow q \\ & q \leftarrow p \\ & r \leftarrow \textit{not } p \end{aligned}$$



Loop

- A nonempty set L of atoms is called a loop of Π if, for every pair a_1, a_2 of atoms in L, there exists a path of non-zero length from a_1 to a_2 in the positive dependency graph of Π such that all vertices in this path belong to L.



- Π_1 has only one loop: $\{p, q\}$.

- A program is called **tight** if it has no loops.

Which of these Examples is a Tight Program?

A.

$$\begin{aligned} p &\leftarrow \neg q \\ q &\leftarrow \neg p \end{aligned}$$

p q

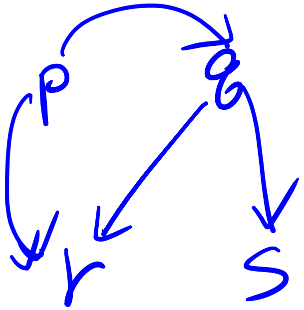
B.

$$\begin{aligned} p &\leftarrow \neg q \\ q &\leftarrow \neg r \end{aligned}$$

p q r

C.

$$\begin{aligned} p &\leftarrow q, r \\ q &\leftarrow r, s \\ s &\leftarrow \neg p \end{aligned}$$



~~**D.**~~

$$\begin{aligned} p &\leftarrow q \\ q &\leftarrow p \\ p &\leftarrow \neg r \end{aligned}$$

Completion: Turning LP$^{\text{MLN}}$ to MLN

- For any ground LPMLN program that consists of rules of the form
  - w: $A \leftarrow Body$
  - where A is an atom and $Body$ is a formula,
- The completion of Π is defined as the union of Π and hard rules

$$\alpha: \quad A \rightarrow \bigvee_{w:A\leftarrow Body\in\Pi} Body$$

for each ground atom A

- **Theorem:** "Tight" LP$^{\text{MLN}}$ program Π under the stable model semantic has the same probability distribution over all interpretations with the completion of Π under the MLN semantics

Example

- Π: under LP^{MLN}

  2: $p \leftarrow \neg q$

  1: $q \leftarrow \neg p$

- Comp(Π): under MLN

  2: $p \leftarrow \neg q$

  1: $q \leftarrow \neg p$

  α: $p \rightarrow \neg q$

  α: $q \rightarrow \neg p$

Wrap-Up