

# CS294-248 Special Topics in Database Theory

## Unit 9: Datalog (Part 2)

Dan Suciu

University of Washington

# Announcements

- Next Tuesday, Nov. 28: office hours 2pm-4:30pm.
- Please submit a short report on your project by Wednesday, Nov. 29.
- Project presentations: Thursday, Nov. 30, 9:30am, Calvin 146. More details TBD.

# Recursion and Negation

## Recap: Datalog

- Datalog = set of rules.
- Immediate consequence operator
- Least fixpoint semantics
- Naive algorithm  $J^{(0)} \subseteq J^{(1)} \subseteq \dots$

Example:

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \end{aligned}$$

Non-example:

$$C(X) :- A(X) \wedge \neg B(X)$$

What happens if we allow negation?

# Three Examples

Transitive closure  
of the complement graph:

$$\begin{aligned} EC(X, Y) &:- V(X) \wedge V(Y) \wedge \neg E(X, Y) \\ T(X, Y) &:- EC(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge EC(Z, X) \end{aligned}$$

## Three Examples

Transitive closure  
of the complement graph:

$$\begin{aligned} EC(X, Y) &:- V(X) \wedge V(Y) \wedge \neg E(X, Y) \\ T(X, Y) &:- EC(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge EC(Z, X) \end{aligned}$$

Complement of the  
transitive closure:

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \\ \text{Answ}(X, Y) &:- V(X) \wedge V(Y) \wedge \neg T(X, Y) \end{aligned}$$

## Three Examples

Transitive closure  
of the complement graph:

$$\begin{aligned} EC(X, Y) &:- V(X) \wedge V(Y) \wedge \neg E(X, Y) \\ T(X, Y) &:- EC(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge EC(Z, X) \end{aligned}$$

Complement of the  
transitive closure:

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \\ \text{Answ}(X, Y) &:- V(X) \wedge V(Y) \wedge \neg T(X, Y) \end{aligned}$$

The Win-Move Game:

$$W(X) :- E(X, Y) \wedge \neg W(Y)$$

(will explain it later)

## But Recursion and Negation Don't Mix

EDB is  $S = \{1\}$

$$\begin{aligned} A(X) &:- S(X) \wedge \neg B(X) \\ B(X) &:- S(X) \wedge \neg A(X) \end{aligned}$$



## But Recursion and Negation Don't Mix

EDB is  $S = \{1\}$

Fixpoint 1:  $A = \{1\}, B = \emptyset$ .

$\begin{aligned} A(X) &:- S(X) \wedge \neg B(X) \\ B(X) &:- S(X) \wedge \neg A(X) \end{aligned}$
--

## But Recursion and Negation Don't Mix

$\begin{aligned} A(X) &:- S(X) \wedge \neg B(X) \\ B(X) &:- S(X) \wedge \neg A(X) \end{aligned}$
--

EDB is  $S = \{1\}$

Fixpoint 1:  $A = \{1\}, B = \emptyset$ .

Fixpoint 2:  $A = \emptyset, B = \{1\}$

## But Recursion and Negation Don't Mix

$$\begin{array}{l} A(X) :- S(X) \wedge \neg B(X) \\ B(X) :- S(X) \wedge \neg A(X) \end{array}$$

EDB is  $S = \{1\}$

Fixpoint 1:  $A = \{1\}, B = \emptyset$ .

Fixpoint 2:  $A = \emptyset, B = \{1\}$

Pre-fixpoint 3:  $A = B = \{1\}$

## But Recursion and Negation Don't Mix

$$\begin{aligned} A(X) &:- S(X) \wedge \neg B(X) \\ B(X) &:- S(X) \wedge \neg A(X) \end{aligned}$$

EDB is  $S = \{1\}$

Fixpoint 1:  $A = \{1\}, B = \emptyset$ .

Fixpoint 2:  $A = \emptyset, B = \{1\}$

Pre-fixpoint 3:  $A = B = \{1\}$

A simpler example:

$$\begin{aligned} A &:- \neg B \\ B &:- \neg A \end{aligned}$$

## But Recursion and Negation Don't Mix

$$\begin{aligned} A(X) &:- S(X) \wedge \neg B(X) \\ B(X) &:- S(X) \wedge \neg A(X) \end{aligned}$$

EDB is  $S = \{1\}$

Fixpoint 1:  $A = \{1\}, B = \emptyset$ .

Fixpoint 2:  $A = \emptyset, B = \{1\}$

Pre-fixpoint 3:  $A = B = \{1\}$

A simpler example:

$$\begin{aligned} A &:- \neg B \\ B &:- \neg A \end{aligned}$$

ICO not monotone! Need new semantics

# Outline

- Semi-positive, stratified datalog
- Semantics motivated by logic.
- Semantics motivated by computation.

Mostly based on [Abiteboul et al., 1995].

# Semi-Positive and Stratified Datalog

## Semi-positive Datalog

EDB atoms may be positive or negated.

IDB atoms can only be positive.

Example: transitive closure of the **complement graph**:

$$\begin{aligned} EC(X, Y) &:- V(X) \wedge V(Y) \wedge \neg E(X, Y) \\ T(X, Y) &:- EC(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge EC(Z, X) \end{aligned}$$



## Semi-positive Datalog

EDB atoms may be positive or negated.

IDB atoms can only be positive.

Example: transitive closure of the **complement graph**:

$$\begin{aligned} EC(X, Y) &:- V(X) \wedge V(Y) \wedge \neg E(X, Y) \\ T(X, Y) &:- EC(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge EC(Z, X) \end{aligned}$$

The Immediate Consequence Operator is monotone.

**Semantics**: least fixpoint of the ICO.

# Stratified Datalog: Definition

- **Stratification**: assign to each IDB predicate a **stratum**  $s(R) \in \mathbb{N}$ .

# Stratified Datalog: Definition

- **Stratification**: assign to each IDB predicate a **stratum**  $s(R) \in \mathbb{N}$ .
- A program  $P$  is **stratified** if there exists a stratification such that:
  - ▶ For positive atoms  $A(\mathbf{X}) :- \dots \wedge B(\mathbf{Y}) \wedge \dots$ :  $s(A) \geq s(B)$ .
  - ▶ For any negative atoms  $A(\mathbf{X}) :- \dots \wedge \neg B(\mathbf{Y}) \wedge \dots$ :  $s(A) > s(B)$ .

# Stratified Datalog: Definition

- **Stratification**: assign to each IDB predicate a **stratum**  $s(R) \in \mathbb{N}$ .
- A program  $P$  is **stratified** if there exists a stratification such that:
  - ▶ For positive atoms  $\boxed{A(\mathbf{X}) \text{ :- } \dots \wedge B(\mathbf{Y}) \wedge \dots}$ :  $s(A) \geq s(B)$ .
  - ▶ For any negative atoms  $\boxed{A(\mathbf{X}) \text{ :- } \dots \wedge \neg B(\mathbf{Y}) \wedge \dots}$ :  $s(A) > s(B)$ .
- **Semantics**: for each stratum  $s = 1, 2, \dots$ , view it as a semi-positive datalog program, compute its fixpoint.

# Stratified Datalog: Definition

- **Stratification**: assign to each IDB predicate a **stratum**  $s(R) \in \mathbb{N}$ .
- A program  $P$  is **stratified** if there exists a stratification such that:
  - ▶ For positive atoms  $A(\mathbf{X}) :- \dots \wedge B(\mathbf{Y}) \wedge \dots$ :  $s(A) \geq s(B)$ .
  - ▶ For any negative atoms  $A(\mathbf{X}) :- \dots \wedge \neg B(\mathbf{Y}) \wedge \dots$ :  $s(A) > s(B)$ .
- **Semantics**: for each stratum  $s = 1, 2, \dots$ , view it as a semi-positive datalog program, compute its fixpoint.
- The output is called **perfect model**; it is not a minimal model!

## Example

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \\ \text{Answ}(X, Y) &:- V(X) \wedge V(Y) \wedge \neg T(X, Y) \end{aligned}$$

Stratum 1:  $T$

Stratum 2:  $\text{Answ}$

---

<sup>1</sup>Assuming no isolated nodes

## Example

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \\ \text{Answ}(X, Y) &:- V(X) \wedge V(Y) \wedge \neg T(X, Y) \end{aligned}$$

Stratum 1:  $T$

Stratum 2: Answ

Semantics:

$T$  = transitive closure, Answ = its complement

This is **not** the least fixpoint (minimal model) **why??**

---

<sup>1</sup>Assuming no isolated nodes

## Example

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \\ \text{Answ}(X, Y) &:- V(X) \wedge V(Y) \wedge \neg T(X, Y) \end{aligned}$$

Stratum 1:  $T$

Stratum 2:  $\text{Answ}$

Semantics:

$T$  = transitive closure,  $\text{Answ}$  = its complement

This is **not** the least fixpoint (minimal model) **why??**

The following is also a fixpoint:<sup>1</sup>

$T = V \times V, \text{Answ} = \emptyset$

---

<sup>1</sup>Assuming no isolated nodes



# Discussion

- Stratified datalog is by far the most popular extension of datalog with negation.
- It is limited: it completely prevents the interleaving of recursion and negation. The following is not allowed:

$$A \text{ :- } \neg B$$
$$B \text{ :- } \neg A$$

# Logic-Based Extensions

# Logic-Based Extensions

- Stable Models
- Well Founded Model

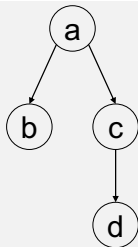
Representative example: the [Win-Move Game](#) (next)

# The Win-Move Game

- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?

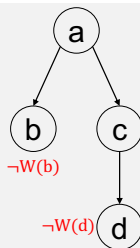
# The Win-Move Game

- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



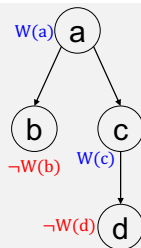
# The Win-Move Game

- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



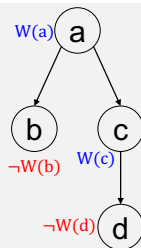
# The Win-Move Game

- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



# The Win-Move Game

- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?

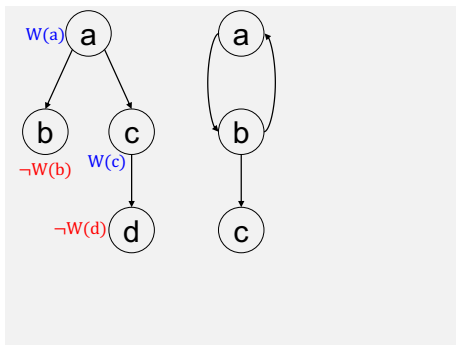


$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$



# The Win-Move Game

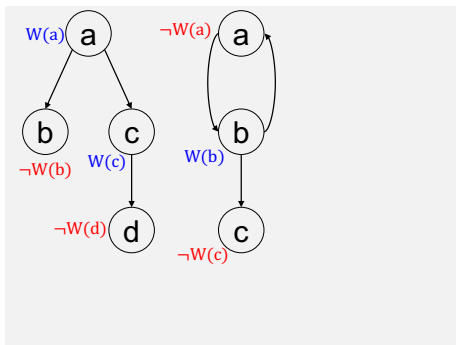
- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

# The Win-Move Game

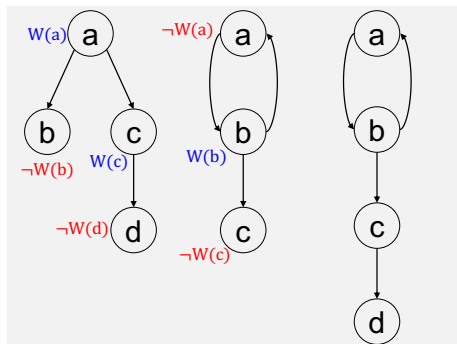
- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

# The Win-Move Game

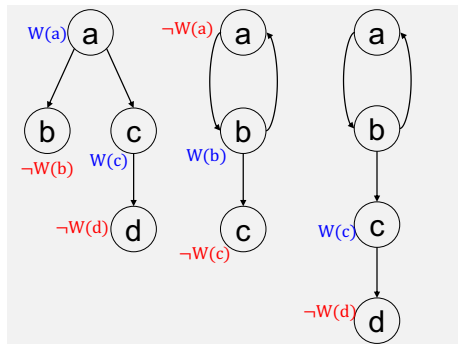
- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

# The Win-Move Game

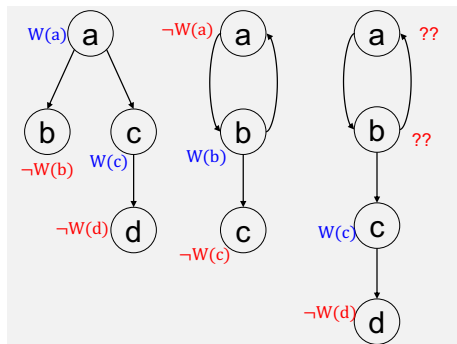
- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

# The Win-Move Game

- Players I, II take turns moving a pebble in a graph.
- Player who cannot move loses.
- For each node  $X$ , does Player I have a winning strategy?



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

## Discussion

- **Least Fixpoint Logic (LFP)** is FO extended with monotone fixpoint.  
E.g. the win-move game:

$$\text{lf}_{p_{W(x)}}(\exists y(E(x, y) \Rightarrow \forall z(E(y, z) \wedge W(z))))$$

## Discussion

- **Least Fixpoint Logic (LFP)** is FO extended with monotone fixpoint.  
E.g. the win-move game:

$$\text{lf}_{p_{W(x)}}(\exists y(E(x, y) \Rightarrow \forall z(E(y, z) \wedge W(z))))$$

- [Chandra and Harel, 1985] claimed that LFP = stratified datalog.

## Discussion

- **Least Fixpoint Logic (LFP)** is FO extended with monotone fixpoint.  
E.g. the win-move game:

$$\text{lfp}_{W(x)}(\exists y(E(x, y) \Rightarrow \forall z(E(y, z) \wedge W(z))))$$

- [Chandra and Harel, 1985] claimed that LFP = stratified datalog.
- [Kolaitis, 1991] disproved it: the win-move game  $\notin$  stratified datalog.



# Discussion

- **Least Fixpoint Logic (LFP)** is FO extended with monotone fixpoint.  
E.g. the win-move game:

$$\text{lfp}_{W(x)}(\exists y(E(x, y) \Rightarrow \forall z(E(y, z) \wedge W(z))))$$

- [Chandra and Harel, 1985] claimed that LFP = stratified datalog.
- [Kolaitis, 1991] disproved it: the win-move game  $\notin$  stratified datalog.
- Hence: need datalog extensions beyond stratified datalog.

## Discussion

- **Least Fixpoint Logic (LFP)** is FO extended with monotone fixpoint.  
E.g. the win-move game:

$$\text{lfp}_{W(x)}(\exists y(E(x, y) \Rightarrow \forall z(E(y, z) \wedge W(z))))$$

- [Chandra and Harel, 1985] claimed that LFP = stratified datalog.
- [Kolaitis, 1991] disproved it: the win-move game  $\notin$  stratified datalog.
- Hence: need datalog extensions beyond stratified datalog.

Before that we discuss two simple technical constructs:

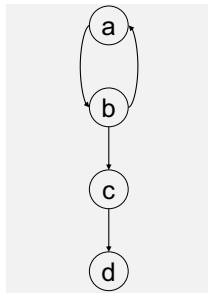
**grounded program** and **reduct**

# The Grounded Datalog Program

A **grounded** atom, or a **fact**, is an atom without variables

A **grounded rule** is a rule whose atoms are grounded.

The **grounding** of a program  $P$  consists of all possible groundings of its rules



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$

# The Reduct

$P \stackrel{\text{def}}{=} \text{the grounded program, } J = \text{any set of grounded atoms;}$

The **reduct**,  $P_J$  is obtained as follows:

- Remove all rules with a negated atom in  $J$ .
- Remove all remaining negated atoms.

$P_J$  is monotone;       $\text{lfp}(P_J)$  exists;       $J_1 \subseteq J_2$  implies  $\text{lfp}(P_{J_1}) \supseteq \text{lfp}(P_{J_2})$

# The Reduct

$P \stackrel{\text{def}}{=}$  the grounded program,  $J$  = any set of grounded atoms;

The **reduct**,  $P_J$  is obtained as follows:

- Remove all rules with a negated atom in  $J$ .
- Remove all remaining negated atoms.

$P_J$  is monotone;       $\text{lfp}(P_J)$  exists;       $J_1 \subseteq J_2$  implies  $\text{lfp}(P_{J_1}) \supseteq \text{lfp}(P_{J_2})$

$W(a) :- E(a, b) \wedge \neg W(b)$

$W(b) :- E(b, a) \wedge \neg W(a)$

$W(b) :- E(b, c) \wedge \neg W(c)$

$W(c) :- E(c, d) \wedge \neg W(d)$

# The Reduct

$P \stackrel{\text{def}}{=}$  the grounded program,  $J =$  any set of grounded atoms;

The **reduct**,  $P_J$  is obtained as follows:

- Remove all rules with a negated atom in  $J$ .
- Remove all remaining negated atoms.

$P_J$  is monotone;       $\text{lfp}(P_J)$  exists;       $J_1 \subseteq J_2$  implies  $\text{lfp}(P_{J_1}) \supseteq \text{lfp}(P_{J_2})$   
 $J = \{W(a), W(d)\};$

$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$   
 $W(b) \text{ :- } E(b, a) \wedge \neg W(a)$   
 $W(b) \text{ :- } E(b, c) \wedge \neg W(c)$   
 $W(c) \text{ :- } E(c, d) \wedge \neg W(d)$

$W(a) \text{ :- } E(a, b)$	$\wedge \neg W(b)$
$W(b) \text{ :- } E(b, a)$	$\wedge \neg W(a)$
$W(b) \text{ :- } E(b, c)$	$\wedge \neg W(c)$
$W(c) \text{ :- } E(c, d)$	$\wedge \neg W(d)$

$\text{lfp}(P_J) = \{W(a), W(b)\}$

# The Reduct

$P \stackrel{\text{def}}{=}$  the grounded program,  $J$  = any set of grounded atoms;

The **reduct**,  $P_J$  is obtained as follows:

- Remove all rules with a negated atom in  $J$ .
- Remove all remaining negated atoms.

$P_J$  is monotone;       $\text{lfp}(P_J)$  exists;       $J_1 \subseteq J_2$  implies  $\text{lfp}(P_{J_1}) \supseteq \text{lfp}(P_{J_2})$

$$J = \{W(a), W(d)\};$$

$$J = \{W(a), W(b), W(d)\};$$

$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$   
 $W(b) \text{ :- } E(b, a) \wedge \neg W(a)$   
 $W(b) \text{ :- } E(b, c) \wedge \neg W(c)$   
 $W(c) \text{ :- } E(c, d) \wedge \neg W(d)$

$\boxed{W(a) \text{ :- } E(a, b)} \wedge \neg W(b)$   
 $W(b) \text{ :- } E(b, a) \wedge \neg W(a)$   
 $\boxed{W(b) \text{ :- } E(b, c)} \wedge \neg W(c)$   
 $W(c) \text{ :- } E(c, d) \wedge \neg W(d)$

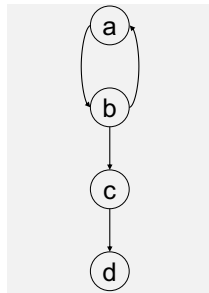
$$\text{lfp}(P_J) = \{W(a), W(b)\}$$

$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$   
 $W(b) \text{ :- } E(b, a) \wedge \neg W(a)$   
 $\boxed{W(b) \text{ :- } E(b, c)} \wedge \neg W(c)$   
 $W(c) \text{ :- } E(c, d) \wedge \neg W(d)$

$$\text{lfp}(P_J) = \{W(b)\}$$

# Stable Models

$J$  is a **stable model** if  $J = \text{lfp}(P_J)$



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

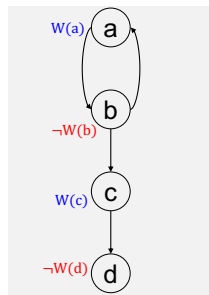
$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$



# Stable Models

$J$  is a **stable model** if  $J = \text{lfp}(P_J)$

Example:  $J = \{W(a), W(c)\}$



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

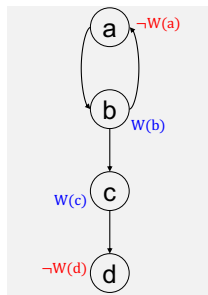
$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$

# Stable Models

$J$  is a **stable model** if  $J = \text{lfp}(P_J)$

Example:  $J = \{W(a), W(c)\}$

Example:  $J = \{W(b), W(c)\}$



$$W(X) :- E(X, Y) \wedge \neg W(Y)$$

$$W(a) :- E(a, b) \wedge \neg W(b)$$

$$W(b) :- E(b, a) \wedge \neg W(a)$$

$$W(b) :- E(b, c) \wedge \neg W(c)$$

$$W(c) :- E(c, d) \wedge \neg W(d)$$

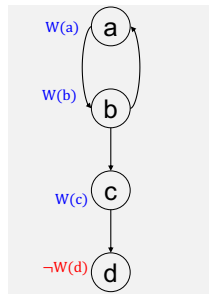
# Stable Models

$J$  is a **stable model** if  $J = \text{lfp}(P_J)$

Example:  $J = \{W(a), W(c)\}$

Example:  $J = \{W(b), W(c)\}$

Non-example:  $J = \{W(a), W(b), W(c)\}$  **why??**



$W(X) :- E(X, Y) \wedge \neg W(Y)$

$W(a) :- E(a, b) \wedge \neg W(b)$

$W(b) :- E(b, a) \wedge \neg W(a)$

$W(b) :- E(b, c) \wedge \neg W(c)$

$W(c) :- E(c, d) \wedge \neg W(d)$

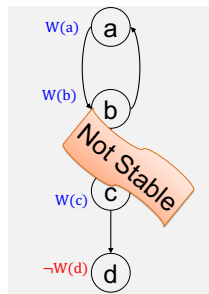
# Stable Models

$J$  is a **stable model** if  $J = \text{lfp}(P_J)$

Example:  $J = \{W(a), W(c)\}$

Example:  $J = \{W(b), W(c)\}$

Non-example:  $J = \{W(a), W(b), W(c)\}$  **why??**



$$W(X) :- E(X, Y) \wedge \neg W(Y)$$

$$W(a) :- E(a, b) \wedge \neg W(b)$$

$$W(b) :- E(b, a) \wedge \neg W(a)$$

$$W(b) :- E(b, c) \wedge \neg W(c)$$

$$W(c) :- E(c, d) \wedge \neg W(d)$$

# Stable Models

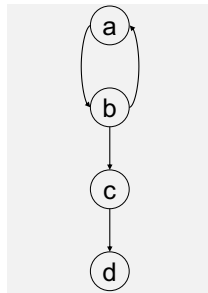
$J$  is a **stable model** if  $J = \text{lfp}(P_J)$

Example:  $J = \{W(a), W(c)\}$

Example:  $J = \{W(b), W(c)\}$

Non-example:  $J = \{W(a), W(b), W(c)\}$  why??

Non-example:  $J = \{W(c)\}$  why??



$$W(X) \text{ :- } E(X, Y) \wedge \neg W(Y)$$

$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$

# Stable Models

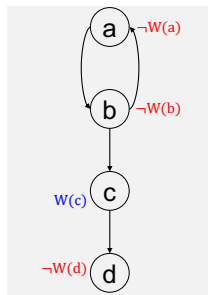
$J$  is a **stable model** if  $J = \text{Ifp}(P_J)$

Example:  $J = \{W(a), W(c)\}$

Example:  $J = \{W(b), W(c)\}$

Non-example:  $J = \{W(a), W(b), W(c)\}$  why??

Non-example:  $J = \{W(c)\}$  why??



$$W(X) :- E(X, Y) \wedge \neg W(Y)$$

$$W(a) :- E(a, b) \wedge \neg W(b)$$

$$W(b) :- E(b, a) \wedge \neg W(a)$$

$$W(b) :- E(b, c) \wedge \neg W(c)$$

$$W(c) :- E(c, d) \wedge \neg W(d)$$

## Discussion

- Stable models introduced by [Gelfond and Lifschitz, 1988]
- Elegant, principled definition.
- But: NP-hard to check if there exists any stable model.

A stratified program has a unique stable model, which is the perfect model.

$A(1) :-$

$B(1) :- \neg A(1)$

$C(1) :- A(1)$

$C(1) :- C(1) \wedge \neg B(1)$

Perfect model:  $J = \{A(1), C(1)\}$

Not stable:  $J = \{A(1), B(1), C(1)\}$  **why?**

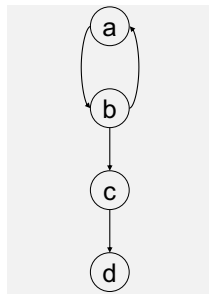
# Well-Founded Model

Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$



$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$



# Well-Founded Model

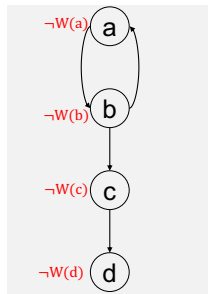
## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$J^{(0)} = \emptyset$$



$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$

# Well-Founded Model

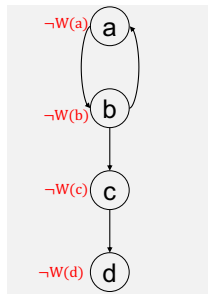
## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$J^{(0)} = \emptyset$$



$$W(a) \text{ :- } E(a, b) \quad \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \quad \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \quad \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \quad \wedge \neg W(d)$$

# Well-Founded Model

## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$J^{(0)} = \emptyset$$

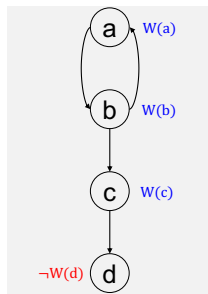
$$J^{(1)} = \{W(a), W(b), W(c)\}$$

$$W(a) \text{ :- } E(a, b) \quad \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \quad \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \quad \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \quad \wedge \neg W(d)$$



# Well-Founded Model

## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$J^{(0)} = \emptyset$$

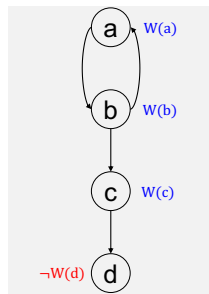
$$J^{(1)} = \{W(a), W(b), W(c)\}$$

$$W(a) :- E(a, b) \wedge \neg W(b)$$

$$W(b) :- E(b, a) \wedge \neg W(a)$$

$$W(b) :- E(b, c) \wedge \neg W(c)$$

$$W(c) :- E(c, d) \wedge \neg W(d)$$



# Well-Founded Model

## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-facts} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$J^{(0)} = \emptyset$$

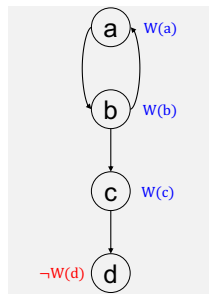
$$J^{(1)} = \{W(a), W(b), W(c)\}$$

$$W(a) :- E(a, b) \wedge \neg W(b)$$

$$W(b) :- E(b, a) \wedge \neg W(a)$$

$$W(b) :- E(b, c) \wedge \neg W(c)$$

$$W(c) :- E(c, d) \wedge \neg W(d)$$



# Well-Founded Model

## Alternating Fixpoint:

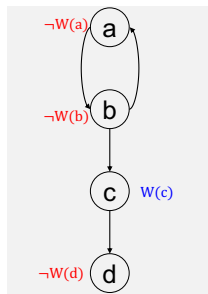
$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-facts} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$\begin{aligned} J^{(0)} &= \emptyset \\ J^{(2)} &= \{W(c)\} \end{aligned} \quad \begin{aligned} J^{(1)} &= \{W(a), W(b), W(c)\} \end{aligned}$$

$$\begin{aligned} W(a) &:- E(a, b) \wedge \neg W(b) \\ W(b) &:- E(b, a) \wedge \neg W(a) \\ W(b) &:- E(b, c) \wedge \neg W(c) \\ W(c) &:- E(c, d) \wedge \neg W(d) \end{aligned}$$



# Well-Founded Model

## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-facts} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

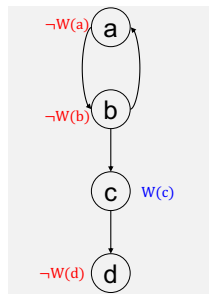
$$\begin{aligned} J^{(0)} &= \emptyset \\ J^{(2)} &= \{W(c)\} \end{aligned} \quad \begin{aligned} J^{(1)} &= \{W(a), W(b), W(c)\} \end{aligned}$$

$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$



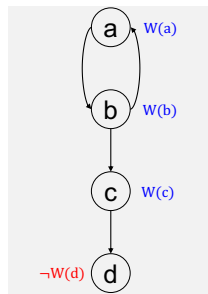
# Well-Founded Model

## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$



$$\begin{array}{ll} J^{(0)} = \emptyset & J^{(1)} = \{W(a), W(b), W(c)\} \\ J^{(2)} = \{W(c)\} & J^{(3)} = \{W(a), W(b), W(c)\} \end{array}$$

$$W(a) \text{ :- } E(a, b) \wedge \neg W(b)$$

$$W(b) \text{ :- } E(b, a) \wedge \neg W(a)$$

$$W(b) \text{ :- } E(b, c) \wedge \neg W(c)$$

$$W(c) \text{ :- } E(c, d) \wedge \neg W(d)$$



# Well-Founded Model

## Alternating Fixpoint:

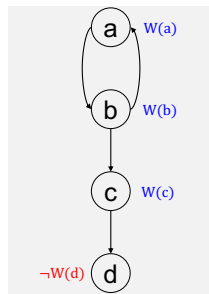
$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, \quad J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$\begin{aligned} J^{(0)} &= \emptyset & J^{(1)} &= \{W(a), W(b), W(c)\} \\ J^{(2)} &= \{W(c)\} & J^{(3)} &= \{W(a), W(b), W(c)\} \end{aligned}$$

$$\begin{aligned} W(a) &:- E(a, b) \wedge \neg W(b) \\ W(b) &:- E(b, a) \wedge \neg W(a) \\ W(b) &:- E(b, c) \wedge \neg W(c) \\ W(c) &:- E(c, d) \wedge \neg W(d) \end{aligned}$$



# Well-Founded Model

## Alternating Fixpoint:

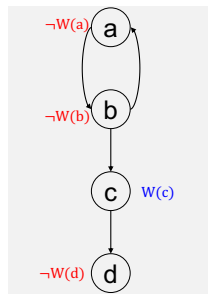
$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-facts} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$\begin{array}{ll} J^{(0)} = \emptyset & J^{(1)} = \{W(a), W(b), W(c)\} \\ J^{(2)} = \{W(c)\} & J^{(3)} = \{W(a), W(b), W(c)\} \\ J^{(4)} = \{W(c)\} & \dots \end{array}$$

$$\begin{array}{l} W(a) :- E(a, b) \wedge \neg W(b) \\ W(b) :- E(b, a) \wedge \neg W(a) \\ W(b) :- E(b, c) \wedge \neg W(c) \\ \boxed{W(c) :- E(c, d)} \wedge \neg W(d) \end{array}$$



# Well-Founded Model

## Alternating Fixpoint:

$$J^{(0)} \stackrel{\text{def}}{=} \emptyset, J^{(t+1)} \stackrel{\text{def}}{=} \text{lfp}(P_{J^{(t)}})$$

$$J^{(0)} \subseteq J^{(2)} \subseteq J^{(4)} \subseteq \dots \subseteq J^{(5)} \subseteq J^{(3)} \subseteq J^{(1)}.$$

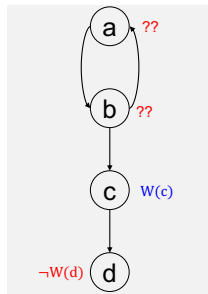
$$\bigcup_t J^{(2t)} \stackrel{\text{def}}{=} \text{certain-} \quad \bigcap_t J^{(2t+1)} \stackrel{\text{def}}{=} \text{possible-facts}$$

$$\begin{aligned} J^{(0)} &= \emptyset & J^{(1)} &= \{W(a), W(b), W(c)\} \\ J^{(2)} &= \{W(c)\} & J^{(3)} &= \{W(a), W(b), W(c)\} \\ J^{(4)} &= \{W(c)\} & \dots & \end{aligned}$$

Certain facts:  $W(c)$ ;

possible facts:  $W(a), W(b)$ .

$$\begin{aligned} W(a) &:- E(a, b) \wedge \neg W(b) \\ W(b) &:- E(b, a) \wedge \neg W(a) \\ W(b) &:- E(b, c) \wedge \neg W(c) \\ W(c) &:- E(c, d) \wedge \neg W(d) \end{aligned}$$



# Discussion

- Well-founded models can be computed in PTIME.
- Yet, I don't know of any system that supports it.  
Maybe because of the 3-valued logic?

Next: two other semantics motivated by computation.

# Computation-Based Extensions

# Computation-Based Extensions

- Datalog with inflationary fixpoint semantics.
- Datalog with partial fixpoint semantics.

# Inflationary Fixpoint

Let  $P$  be a datalog<sup>+</sup> program,  $T_P$  its ICO.

The **inflationary fixpoint** is  $\boxed{\text{ifp}(P) \stackrel{\text{def}}{=} \bigcup_{t \geq 0} J_t}$ , where:

$$J_0 \stackrel{\text{def}}{=} \emptyset, \quad J_{t+1} \stackrel{\text{def}}{=} J_t \cup T_P(J_t)$$

## Fact

*ifp(P) can be computed in PTIME in the size of the EDB I.*

why?

# Inflationary Fixpoint

Let  $P$  be a datalog<sup>+</sup> program,  $T_P$  its ICO.

The **inflationary fixpoint** is  $\boxed{\text{ifp}(P) \stackrel{\text{def}}{=} \bigcup_{t \geq 0} J_t}$ , where:

$$J_0 \stackrel{\text{def}}{=} \emptyset, \quad J_{t+1} \stackrel{\text{def}}{=} J_t \cup T_P(J_t)$$

## Fact

*ifp(P) can be computed in PTIME in the size of the EDB I.*

**why?** Because  $J_0 \subseteq J_1 \subseteq \dots \subseteq (\text{ADom}(I))^k$



# Partial Fixpoint

The **partial fixpoint** is:

$$\text{pfp}(P) \stackrel{\text{def}}{=} \begin{cases} J_{t_0} & \text{if } J_{t_0} = J_{t_0+1} \\ \emptyset & \text{if } J_t \neq J_{t+1}, \forall t \end{cases}$$

where

$$J_0 \stackrel{\text{def}}{=} \emptyset, \quad J_{t+1} \stackrel{\text{def}}{=} T_P(J_t)$$

## Fact

*pfp(P) can be computed in PSPACE in the size of the EDB I.*

why?

# Partial Fixpoint

The **partial fixpoint** is:

$$\text{pfp}(P) \stackrel{\text{def}}{=} \begin{cases} J_{t_0} & \text{if } J_{t_0} = J_{t_0+1} \\ \emptyset & \text{if } J_t \neq J_{t+1}, \forall t \end{cases}$$

where

$$J_0 \stackrel{\text{def}}{=} \emptyset, \quad J_{t+1} \stackrel{\text{def}}{=} T_P(J_t)$$

## Fact

*pfp(P) can be computed in PSPACE in the size of the EDB I.*

**why?** each  $|J_t|$  has size polynomial in  $\text{ADom}(I)$ .

Detect non-termination using a counter.

# How To Express Negation

It's harder than one may think!

Complement of the TC:

$$T(X, Y) :- E(X, Y)$$

$$T(X, Y) :- T(X, Z) \wedge E(Z, X)$$

$$\text{Answ}(X, Y) :- V(X) \wedge V(Y)$$

$$\wedge \neg T(X, Y)$$

ifp( $P$ ) is incorrect!

# How To Express Negation

It's harder than one may think!

Complement of the TC:

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \\ \text{Answ}(X, Y) &:- V(X) \wedge V(Y) \\ &\quad \wedge \neg T(X, Y) \end{aligned}$$

ifp( $P$ ) is incorrect!

Detect the last step:

$$\begin{aligned} T(X, Y) &:- E(X, Y) \\ T(X, Y) &:- T(X, Z) \wedge E(Z, X) \\ T_{\text{prev}}(X, Y) &:- T(X, Y) \\ C() &:- T(U, V) \wedge \neg T_{\text{prev}}(U, V) \\ D() &:- T(U, V) \wedge \neg C() \\ \text{Answ}(X, Y) &:- V(X) \wedge V(Y) \\ &\quad \wedge \neg T(X, Y) \wedge D() \end{aligned}$$

## Descriptive Complexity

- Datalog<sup>¬</sup> cannot express **parity**, no matter which semantics we adopt.

---

<sup>2</sup>**Exercise:** express  $\text{succ}(X, Y)$ ,  $\text{min}(X)$ ,  $\text{max}(Y)$  using  $<$ .

## Descriptive Complexity

- Datalog<sup>¬</sup> cannot express **parity**, no matter which semantics we adopt.
- If we have access to an order relation  $<$  then we can express parity as:<sup>2</sup>

---

<sup>2</sup>**Exercise:** express  $\text{succ}(X, Y)$ ,  $\text{min}(X)$ ,  $\text{max}(Y)$  using  $<$ .

## Descriptive Complexity

- Datalog<sup>¬</sup> cannot express **parity**, no matter which semantics we adopt.
- If we have access to an order relation  $<$  then we can express parity as:<sup>2</sup>

$$E(X, Y) \text{ :- succ}(X, Z) \wedge \text{succ}(Z, Y)$$

$$E(X, Y) \text{ :- } E(X, Z) \wedge E(Z, Y) \quad // \text{ even-length distance}$$

$$\text{Even}() \text{ :- } R(X) \wedge \text{min}(X) \wedge E(X, Y) \wedge \text{max}(Y) \wedge R(Y)$$

### Theorem (Descriptive Complexity [Vardi, 1982, Immerman, 1986])

- Datalog<sup>¬</sup>( $<$ , ifp) expresses precisely queries in PTIME.
- Datalog<sup>¬</sup>( $<$ , pfp) expresses precisely queries in PSPACE.

<sup>2</sup>**Exercise:** express succ( $X, Y$ ), min( $X$ ), max( $Y$ ) using  $<$ .

# Discussion

- Datalog: simple, elegant, appealing. New resurgence after a 40 years history.
- Stratified datalog<sup>¬</sup> is a simple and practical extension.
- Beyond that, it becomes questionable.
- But the theory is beautiful. A famous result:

Theorem ([Abiteboul et al., 1992])

$datalog^{\neg}(ifp) = datalog^{\neg}(pfp)$  iff  $PTIME = PSPACE$ .





Abiteboul, S., Hull, R., and Vianu, V. (1995).

*Foundations of Databases.*

Addison-Wesley.



Abiteboul, S., Vardi, M. Y., and Vianu, V. (1992).

Fixpoint logics, relational machines, and computational complexity.

In *Proceedings of the Seventh Annual Structure in Complexity Theory Conference, Boston, Massachusetts, USA, June 22-25, 1992*, pages 156–168. IEEE Computer Society.



Chandra, A. K. and Harel, D. (1985).

Horn clauses queries and generalizations.

*J. Log. Program.*, 2(1):1–15.



Gelfond, M. and Lifschitz, V. (1988).

The stable model semantics for logic programming.

In Kowalski, R. A. and Bowen, K. A., editors, *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*, pages 1070–1080. MIT Press.



Immerman, N. (1986).

Relational queries computable in polynomial time.

*Inf. Control.*, 68(1-3):86–104.



Kolaitis, P. G. (1991).

The expressive power of stratified programs.

*Inf. Comput.*, 90(1):50–66.



Vardi, M. Y. (1982).

The complexity of relational query languages (extended abstract).

In Lewis, H. R., Simons, B. B., Burkhard, W. A., and Landweber, L. H., editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 137–146. ACM.