Total and Partial Well-Founded Datalog Coincide

Jörg Flum[I] Max Kubierschky[I] Bertram Ludäscher[II]

[I]Mathematische Fakultät, Universität Freiburg, Germany
[II]Institut für Informatik, Universität Freiburg, Germany

Overview

- Introduction: Open Problem by Abiteboul-Hull-Vianu [AHV95]
- Games and Well-Founded Datalog (W-Datalog)
- Main Result: W-Datalog $\overset{(1)}{\leadsto}$ Game $\overset{(2)}{\leadsto}$ Draw-Free Game
- Example for (1): Transitive Closure as a Game
- Basic Ideas of (2)

Every *W-Datalog* program P (= Datalog with *Well-Founded Semantics WFS*) defines a query:

$$q_P : \quad D \mapsto \{ \ (x_1, \ldots, x_k) \mid \mathit{WFS}(P \cup D)(answer(x_1, \ldots, x_k)) = \textbf{true} \ \}$$

- the input D of P is a finite set of facts (= database instance/EDB),
- the output is computed using a distinguished relation $answer$.

Open Problem [Abiteboul-Hull-Vianu-95]:

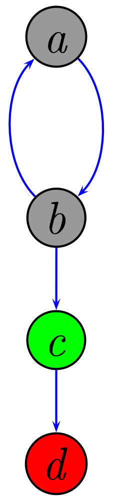
> *Can one find for every Datalog program an equivalent* **total** *program?*
> *I.e., can one find for every program P a program P' s.t.* **for all** *EDBs D:*
>
> 1. $q_{P'}(D) = q_P(D)$, and
> 2. *WFS*$(P' \cup D)$ is total (2-valued) ?

In particular:

$\mathit{WFS}(P \cup D)(answer) =$	True	Undefined	False
$\mathit{WFS}(P' \cup D)(answer) =$	True	False	

A *game* is a finite structure $G = (Positions, Moves)$. G is played with a *pebble* by two players I and II in *rounds* ($=$ 2 moves). I starts from some $x \in Positions$.

Example: $Positions = \{ a, b, c, d \}$



Moves

- A player *loses* if she cannot move; *wins* if she can move to a position in which the opponent loses.
- $x \in Positions$ is $\begin{cases} \text{won} & \text{if I can win the game starting at } x \\ \text{lost} & \text{if II can win (with I starting at } x) \\ \text{drawn} & \text{otherwise} \end{cases}$

  provided I and II play optimal.
- A game is *draw-free* if no $x \in Positions$ is drawn.

The W-Datalog program

$$P^{GAME} = \{\ win(X) \leftarrow move(X,Y),\ \neg\, win(Y)\ \}$$

represents games:

$$\boxed{\textit{WFS}(P^{GAME} \cup move)(win(X)) = \left\{ \begin{array}{c} \textbf{true} \\ \textbf{false} \\ \textbf{undef} \end{array} \right\} \iff X \text{ is } \left\{ \begin{array}{c} won \\ lost \\ drawn \end{array} \right\} \text{ in } G}$$

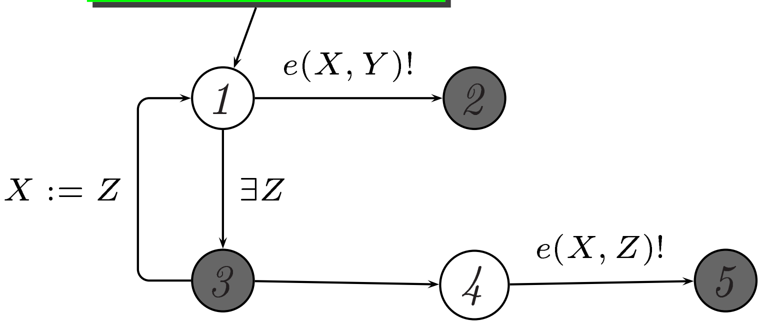
Theorem 1 (W-Datalog $\leq$ W-Datalog^{GAME}).
For every W-Datalog program P there is an equivalent program in W-Datalog^{GAME}.

Proof.
$$\begin{tikzcd} \text{W-Datalog} \arrow[r, phantom, "\overset{[VanGelder]}{\equiv}"] & \text{LFP} \arrow[r, phantom, "\overset{[Grohe]}{\equiv}"] \arrow[rr, bend right, "\overset{[Paper:\ Sketch]}{\leq}"'] & \text{BSNF} \arrow[r, phantom, "\overset{[Paper]}{\leq}"] & \text{W-Datalog}^{GAME} \end{tikzcd}$$

$$
\begin{array}{lllll}
P_{tc}: & \text{tc(X,Y)} & \leftarrow & \text{e(X,Y).} \\
 & \text{tc(X,Y)} & \leftarrow & \text{e(X,Z), tc(Z,Y).}
\end{array}
$$

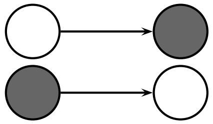
$D = \{\ e(a,b), e(b,c)\ \}$, $\quad q_{P_{tc}}(D) = \{\ (a,b), (b,c), (a,c)\ \}$ (here: $answer$ = tc)

Game-**Diagram** for P_{tc}: ☺ claims: *"tc(X, Y) holds!"*



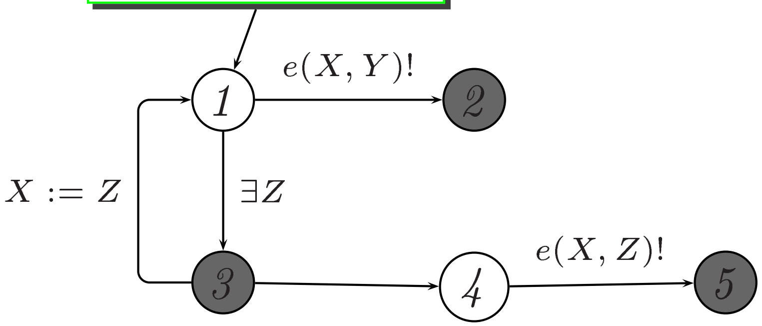
I = ☺ :

II = ☻:



Fact: ☺ can win the game $G(1, X, Y)$ iff $tc(X, Y)$ holds.

Game-**Diagram** for P_{tc}: ☺ claims: *"$tc(X, Y)$ holds!"*



Normal Form W-Datalog^{GAME} for P_{tc}:

$$\text{win}(S, X, Y, Z) \leftarrow \text{move}(S, X, Y, Z, S', X', Y', Z'), \neg\,\text{win}(S', X', Y', Z').$$

$$\text{move}(1, X, Y, Z, 2, X', Y', Z') \leftarrow e(X, Y), X' = X, Y' = Y, Z' = Z.$$
$$\text{move}(1, X, Y, Z, 3, X', Y', Z') \leftarrow \text{dom}(Z'), X' = X, Y' = Y.$$
$$\text{move}(3, X, Y, Z, 1, X', Y', Z') \leftarrow X' = Z, Y' = Y, Z' = Z.$$
$$\text{move}(3, X, Y, Z, 4, X', Y', Z') \leftarrow X' = X, Y' = Y, Z' = Z.$$
$$\text{move}(4, X, Y, Z, 5, X', Y', Z') \leftarrow e(X, Z), X' = X, Y' = Y, Z' = Z.$$

$$\text{dom}(X) \leftarrow e(X, \_).$$
$$\text{dom}(X) \leftarrow e(\_, X).$$

Main Theorem (W-Datalog $\equiv$ W-Datalog$_2^{GAME}$).
For every W-Datalog program P, there is a program $P' \equiv P$, such that

1. $win(\bar{X}) \;\leftarrow\; move(\bar{X}, \bar{Y}), \neg\, win(\bar{Y})$ *is the* **only recursive** *rule of P', and*
2. *WFS$(P' \cup D)$ is* **total** *(2-valued)* **for all** *database instances D.*

Proof. W-Datalog $\overset{Theorem1}{\leq}$ W-Datalog^{GAME} $\overset{Theorem2}{\leq}$ W-Datalog$_2^{GAME}$ $\leq$ W-Datalog

$\Rightarrow$ Every W-Datalog program P (LFP formula) corresponds to a **draw-free game** G_P between two players $\mathrm{I} =$ ☺ and $\mathrm{II} =$ ☻.

This affirmatively answers the question by Abiteboul, Hull, and Vianu:

Corollary (W-Datalog $\equiv$ W-Datalog$_2$).
For every W-Datalog program, there is an equivalent total program.

Theorem 2 (W-Datalog^{GAME} $\leq$ W-Datalog$_2^{GAME}$).
For every W-Datalog^{GAME} program there is an equivalent program in W-Datalog$_2^{GAME}$.

Problem: How to detect/avoid drawn positions ($\widehat{=}$ undefined atoms) in G? (No order available!)

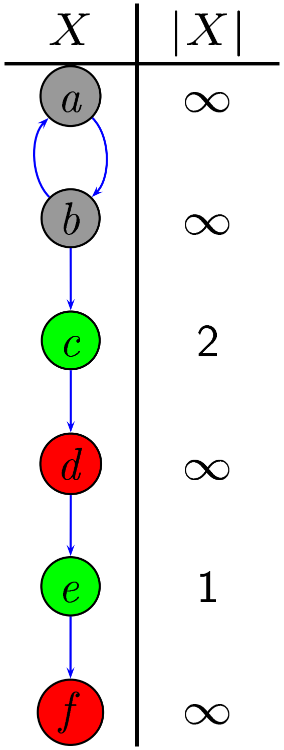
Basic Idea: Limit the **length** of the game $G(X)$ starting at X by comparing it with a **won** game $G(Y)$ of **maximal length**. Play both games $G(X)$ and $G(Y)$ simultaneously:

$\boxed{\text{If ☻ can win } G(Y) \text{ before ☺ wins } G(X)\text{, then } X \text{ is drawn! ☻ ☹}}$

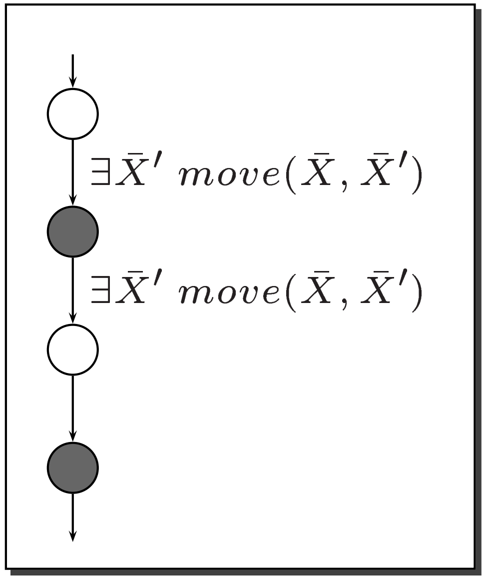
This idea can be encoded as a game $2G(X, Y)$! [Kubierschky95]

The **length** $|\boldsymbol{X}|$ of a won position X is the maximal number of rounds needed to win $G(X)$. $|X| := \infty$ if X is lost or drawn.

Example:

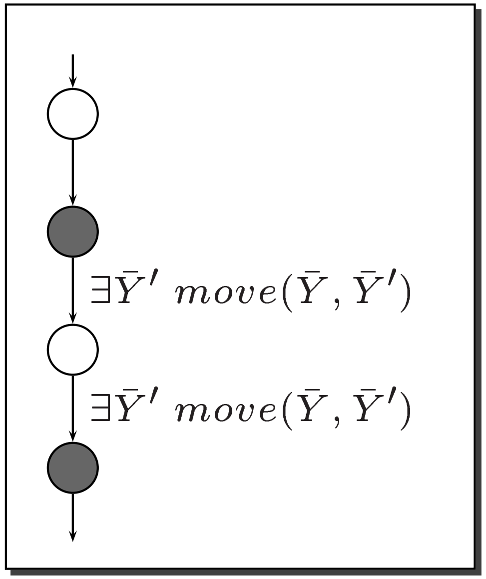


$\boxed{\text{1 round}(\bar{X})}$:=



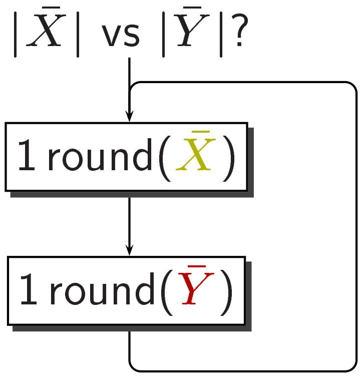
☺ moves first in the game with $\bar{X}$

$\boxed{\text{1 round}(\bar{Y})}$:=



☻ moves first in the game with $\bar{Y}$

Comparison-Loop for $|\bar{X}|, |\bar{Y}|$:



☺ can win iff $|\bar{X}| \leq |\bar{Y}|$
☻ can win iff $|\bar{X}| > |\bar{Y}|$

☺ $s_0 \longrightarrow a$: *"$\bar{X}$ is won!"*

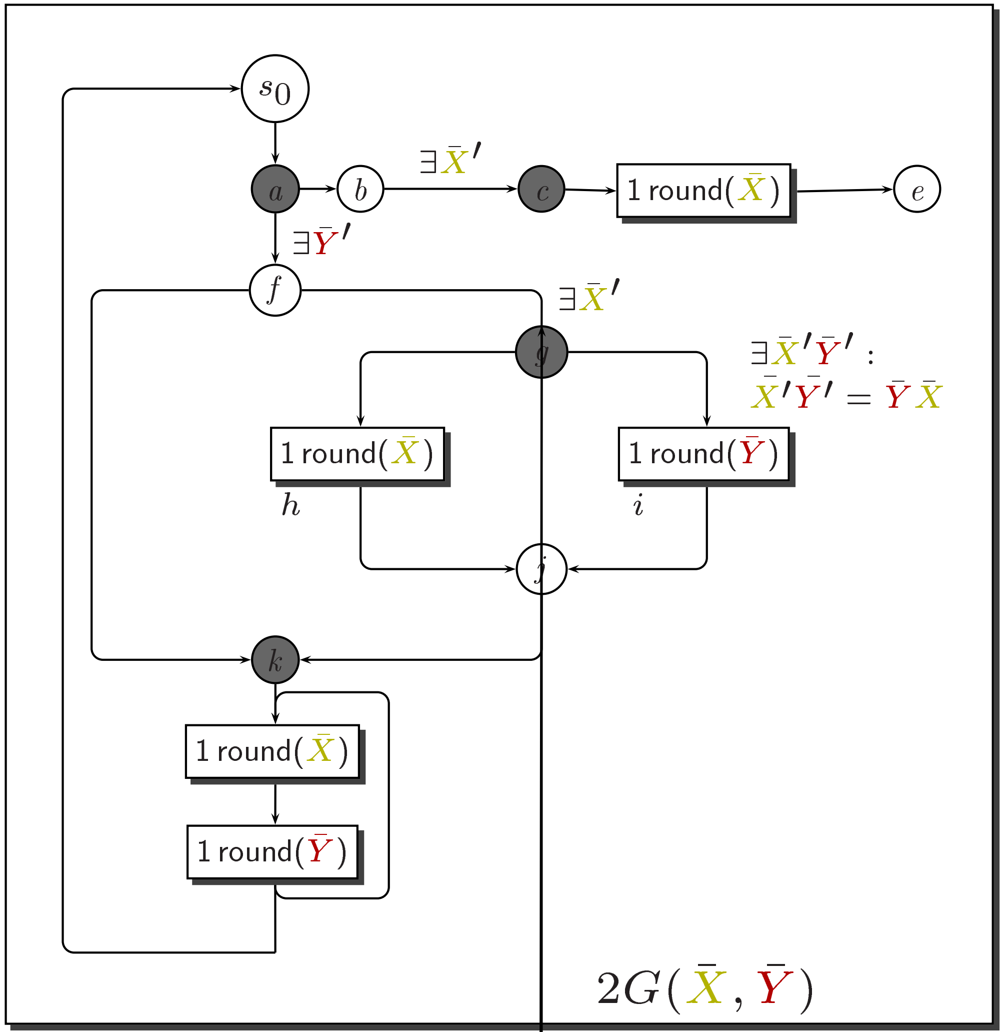
☻ $a \longrightarrow f$: *"$\bar{Y}$ is won and maximal!"*

☺ $f \longrightarrow k$: *"I win $\bar{X}$ before you win $\bar{Y}$!"*

$\quad\searrow g$: *"Your $\bar{Y}$ is too small (not maximal)! I show you a new $\bar{X}$, which is 1 round longer!"*

☻ $g \longrightarrow h$: *"Your $\bar{X}$ is more than 1 longer. I give you a lead of 1 round($\bar{X}$) and you lose!"*

$\quad\searrow i$: *"Your $\bar{X}$ is not longer than my $\bar{Y}$! Let's swap $\bar{X}$ and $\bar{Y}$, and give me a lead of 1 round($\bar{Y}$). Then I will win before you!"*



$G(\bar{X}) \mathrel{\widehat{=}}$ original game (☺ moves first)

$G(\bar{Y}) \mathrel{\widehat{=}}$ "clock game" (☻ moves first)

The reduction is constructed such that:

- ☺ wins $(s_0, \bar{X}, \bar{Y})$ in $2G$ iff $\bar{X}$ is won in G, and
- ☻ wins $(a, \bar{X}, \bar{Y})$ in $2G$ iff $\bar{X}$ is lost or drawn in G.
- No positions $(S, \bar{X}, \bar{Y})$ in $2G$ are drawn.

Summary

- For every W-Datalog program (and LFP- or Fixpoint-Query) there is an equivalent program in the form of a game between two players ☺ and ☻ (Theorem 1: W-Datalog$\leq$W-Datalog^{GAME}).
- Games can be reduced to draw-free games ☹ ☻ (Theorem 2: W-Datalog$^{GAME}\leq$W-Datalog$_2^{GAME}$).

$\Rightarrow$ **Main Theorem**: W-Datalog $\equiv$ W-Datalog$_2^{GAME}$

$\Rightarrow$ **Corollary**: W-Datalog $\equiv$ W-Datalog$_2$, ie for every W-Datalog program, there is an equivalent total program.