Equivalent Stream Reasoning Programs

Harald Beck Minh Dao-Tran Thomas Eiter

Institute of Information Systems, TU Vienna

IJCAI, July 13, 2016

- Austrian Science Fund (FWF) grants P26471, W1255-N23

Equivalent Stream Reasoning Programs

Equivalent Stream Reasoning Programs

- Logic-oriented processing of continuously streaming data

Equivalent Stream Reasoning Programs

- Logic-oriented processing of continuously streaming data
- "What is true *now*?"

Equivalent Stream Reasoning Programs

- Logic-oriented processing of continuously streaming data
- “What is true *now*?”
- Various aspects are emphasized, e.g., incremental reasoning (Stream Reasoning Workshop)

Equivalent Stream Reasoning Programs

- Logic-oriented processing of continuously streaming data
- "What is true *now*?"
- Various aspects are emphasized, e.g., incremental reasoning (Stream Reasoning Workshop)
- Streaming data
  - Data comes and goes - only recent data matters
  - Need to cope with data volume

Equivalent Stream Reasoning Programs

- Logic-oriented processing of continuously streaming data
- "What is true *now*?"
- Various aspects are emphasized, e.g., incremental reasoning (Stream Reasoning Workshop)
- Streaming data
  - Data comes and goes - only recent data matters
  - Need to cope with data volume
- ⇒ Limit reasoning to small **windows**
  - Time-based windows
  - Tuple-based windows . . .

Equivalent Stream Reasoning Programs

- Logic-oriented processing of continuously streaming data
- "What is true *now*?"
- Various aspects are emphasized, e.g., incremental reasoning (Stream Reasoning Workshop)
- Streaming data
  - Data comes and goes - only recent data matters
  - Need to cope with data volume
- $\Rightarrow$ Limit reasoning to small **windows**
  - Time-based windows
  - Tuple-based windows ...
- Stream $S = (T, \upsilon)$
  - **Timeline** T closed interval in $\mathbb{N}$, $t \in T$ **time point**
  - **Evaluation** function $\upsilon : T \rightarrow 2^{\mathcal{A}}$ (sets of atoms)

Equivalent Stream Reasoning Programs

- Logic-oriented processing of continuously streaming data
- "What is true *now*?"
- Various aspects are emphasized, e.g., incremental reasoning (Stream Reasoning Workshop)
- Streaming data
  - Data comes and goes - only recent data matters
  - Need to cope with data volume
- $\Rightarrow$ Limit reasoning to small **windows**
  - Time-based windows
  - Tuple-based windows ...
- Stream $S = (T, \upsilon)$
  - **Timeline** T closed interval in $\mathbb{N}$, $t \in T$ **time point**
  - **Evaluation** function $\upsilon : T \rightarrow 2^{\mathcal{A}}$ (sets of atoms)
- Window function w yields window $w(S, t) \subseteq S$

Equivalent Stream Reasoning Programs

Equivalent Stream Reasoning Programs

"If you are in a hurry,

$$hurry,$$

Equivalent Stream Reasoning Programs

"If you are in a hurry, take a cab,

$$takeCab \leftarrow hurry,$$

Equivalent Stream Reasoning Programs

"If you are in a hurry, take a cab, unless there was a traffic jam in the last 10 minutes."

$$\mathit{takeCab} \leftarrow \mathit{hurry}, \text{not } \mathit{trafficJamLast10Min}$$

Equivalent Stream Reasoning LARS Programs

“If you are in a hurry, take a cab, unless,

$$takeCab \leftarrow hurry,\ \text{not}$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window (⊞) of the last 10 minutes,

$$takeCab \leftarrow hurry,\ \text{not}\ \boxplus^{10}$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window ($\boxplus$) of the last 10 minutes, there is a time point ($\Diamond$)

$$\mathit{takeCab} \leftarrow \mathit{hurry},\ \text{not}\ \boxplus^{10}\Diamond$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window ($\boxplus$) of the last 10 minutes, there is a time point ($\Diamond$) where a traffic jam was reported."

$$takeCab \leftarrow hurry,\ \text{not}\ \boxplus^{10}\Diamond trafficJam$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window ($\boxplus$) of the last 10 minutes, there is a time point ($\Diamond$) where a traffic jam was reported."

$$\mathit{takeCab} \leftarrow \mathit{hurry},\ \mathrm{not}\ \boxplus^{10} \Diamond \mathit{trafficJam}$$

"If no train arrived in the last 20 minutes, do not take the subway."

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window (⊞) of the last 10 minutes, there is a time point (◇) where a traffic jam was reported."

$$takeCab \leftarrow hurry,\ \text{not}\ \boxplus^{10} \Diamond trafficJam$$

"If no train arrived in the last 20 minutes, do not take the subway."

$$@_T someTrain \quad \leftarrow \quad \boxplus^{20} @_T arrivalOfTrain(Id)$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window (⊞) of the last 10 minutes, there is a time point (◇) where a traffic jam was reported."

$$takeCab \leftarrow hurry,\ \text{not}\ \boxplus^{10} \Diamond trafficJam$$

"If no train arrived in the last 20 minutes, do not take the subway."

$$@_T someTrain \quad \leftarrow \quad \boxplus^{20} @_T arrivalOfTrain(Id)$$

$$\boxplus^{20}$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window ($\boxplus$) of the last 10 minutes, there is a time point ($\Diamond$) where a traffic jam was reported."

$$takeCab \leftarrow hurry, \text{ not } \boxplus^{10} \Diamond trafficJam$$

"If no train arrived in the last 20 minutes, do not take the subway."

$$@_T someTrain \quad \leftarrow \quad \boxplus^{20} @_T arrivalOfTrain(Id)$$
$$\boxplus^{20} \Box$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window ($\boxplus$) of the last 10 minutes, there is a time point ($\Diamond$) where a traffic jam was reported."

$$takeCab \leftarrow hurry, \text{ not } \boxplus^{10} \Diamond trafficJam$$

"If no train arrived in the last 20 minutes, do not take the subway."

$$@_T someTrain \quad \leftarrow \quad \boxplus^{20} @_T arrivalOfTrain(Id)$$
$$\boxplus^{20} \Box \neg$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window (⊞) of the last 10 minutes, there is a time point (◇) where a traffic jam was reported."

$$takeCab \leftarrow hurry, \; \text{not } \boxplus^{10} \Diamond trafficJam$$

"If no train arrived in the last 20 minutes, do not take the subway."

$$\begin{aligned} @_T someTrain \quad \leftarrow \quad & \boxplus^{20} @_T arrivalOfTrain(Id) \\ & \boxplus^{20} \Box \neg someTrain \end{aligned}$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window ($\boxplus$) of the last 10 minutes, there is a time point ($\Diamond$) where a traffic jam was reported."

$$takeCab \leftarrow hurry,\ \text{not}\ \boxplus^{10}\Diamond trafficJam$$

"If no train arrived in the last 20 minutes, do not take the subway."

$$\begin{aligned} @_T someTrain &\leftarrow \boxplus^{20} @_T arrivalOfTrain(Id) \\ \neg takeSub &\leftarrow \boxplus^{20}\Box\neg someTrain \end{aligned}$$

Equivalent Stream Reasoning LARS Programs

"If you are in a hurry, take a cab, unless, within a window (⊞) of the last 10 minutes, there is a time point (◇) where a traffic jam was reported."

$$takeCab \leftarrow hurry,\ \text{not}\ \boxplus^{10}\Diamond trafficJam$$

"If no train arrived in the last 20 minutes, do not take the subway."

$$\begin{aligned} @_T someTrain &\leftarrow \boxplus^{20}@_T arrivalOfTrain(Id) \\ \neg takeSub &\leftarrow \boxplus^{20}\Box\neg someTrain \\ &\Downarrow \\ \neg takeSub &\leftarrow \text{not}\ \boxplus^{20}\Diamond someTrain \end{aligned}$$

Equivalent Stream Reasoning Programs

Goal

- Towards optimization:

When are two LARS programs equivalent?

Equivalent Stream Reasoning Programs

Goal

- Towards optimization:

  When are two LARS programs equivalent?

- Semantic characterizations for suitable notions of equivalence

Equivalent Stream Reasoning Programs

Goal

- Towards optimization:

  When are two LARS programs equivalent?

- Semantic characterizations for suitable notions of equivalence
- $\Rightarrow$ Do techniques from ASP carry over to LARS?

Strong Equivalence: Recall from ASP

Strong Equivalence: Recall from ASP

- $\mathcal{AS}(P)$: set of answer sets for program P

Strong Equivalence: Recall from ASP

- $\mathcal{AS}(P)$: set of answer sets for program P
- Ordinary equivalence of P and Q: $\mathcal{AS}(P) = \mathcal{AS}(Q)$

Strong Equivalence: Recall from ASP

- $\mathcal{AS}(P)$: set of answer sets for program P
- Ordinary equivalence of P and Q: $\mathcal{AS}(P) = \mathcal{AS}(Q)$
- Strong Equivalence:

$$P \equiv_s Q \text{ iff for all programs } R,\ \mathcal{AS}(P \cup R) = \mathcal{AS}(Q \cup R)$$

Strong Equivalence: Recall from ASP

- $\mathcal{AS}(P)$: set of answer sets for program P
- Ordinary equivalence of P and Q: $\mathcal{AS}(P) = \mathcal{AS}(Q)$
- Strong Equivalence:

$$P \equiv_s Q \text{ iff for all programs } R,\ \mathcal{AS}(P \cup R) = \mathcal{AS}(Q \cup R)$$

- Characterization with logic of Here-and-There models HT:

$$P \equiv_s Q \quad \text{iff} \quad HT(P) = HT(Q)$$

Strong Equivalence: Recall from ASP

- $\mathcal{AS}(P)$: set of answer sets for program P
- Ordinary equivalence of P and Q: $\mathcal{AS}(P) = \mathcal{AS}(Q)$
- Strong Equivalence:

$$P \equiv_s Q \text{ iff for all programs } R,\ \mathcal{AS}(P \cup R) = \mathcal{AS}(Q \cup R)$$

- Characterization with logic of Here-and-There models HT:

$$P \equiv_s Q \quad \text{iff} \quad HT(P) = HT(Q)$$

- Abstraction: Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\ \ X \subseteq Y \qquad (ii)\ \ Y \models P \qquad (iii)\ \ X \models P^Y$$

Strong Equivalence: Recall from ASP

- $\mathcal{AS}(P)$: set of answer sets for program P
- Ordinary equivalence of P and Q: $\mathcal{AS}(P) = \mathcal{AS}(Q)$
- Strong Equivalence:

$$P \equiv_s Q \text{ iff for all programs } R,\ \mathcal{AS}(P \cup R) = \mathcal{AS}(Q \cup R)$$

- Characterization with logic of Here-and-There models HT:

$$P \equiv_s Q \quad \text{iff} \quad HT(P) = HT(Q)$$

- Abstraction: Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\ \ X \subseteq Y \qquad (ii)\ \ Y \models P \qquad (iii)\ \ X \models P^Y$$

- $\Rightarrow$ Alternative characterization with SE-models SE:

$$P \equiv_s Q \quad \text{iff} \quad SE(P) = SE(Q)$$

Strong Equivalence: Recall from ASP

- $\mathcal{AS}(P)$: set of answer sets for program P
- Ordinary equivalence of P and Q: $\mathcal{AS}(P) = \mathcal{AS}(Q)$
- Strong Equivalence:

$$P \equiv_s Q \text{ iff for all programs } R,\ \mathcal{AS}(P \cup R) = \mathcal{AS}(Q \cup R)$$

- Characterization with logic of Here-and-There models HT:

$$P \equiv_s Q \quad \text{iff} \quad HT(P) = HT(Q)$$

- Abstraction: Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\ \ X \subseteq Y \qquad (ii)\ \ Y \models P \qquad (iii)\ \ X \models P^Y$$

- $\Rightarrow$ Alternative characterization with SE-models SE:

$$P \equiv_s Q \quad \text{iff} \quad SE(P) = SE(Q)$$

- Answer sets characterized by equilibrium models (X, X), where no smaller (X', X) is an SE-model

Cornerstone: Characterizing Answer Sets/Streams

ASP

- Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\;\; X \subseteq Y \qquad (ii)\;\; Y \models P \qquad (iii)\;\; X \models P^Y$$

- Characterization:

$$P \equiv_s Q \quad \text{iff} \quad SE(P) = SE(Q)$$

Cornerstone: Characterizing Answer Sets/Streams

ASP

- Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\ \ X \subseteq Y \qquad (ii)\ \ Y \models P \qquad (iii)\ \ X \models P^Y$$

- Characterization:

$$P \equiv_s Q \quad \text{iff} \quad SE(P) = SE(Q)$$

- Can we find similar characterizations for LARS?

Cornerstone: Characterizing Answer Sets/Streams

ASP

- Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\ \ X \subseteq Y \qquad (ii)\ \ Y \models P \qquad (iii)\ \ X \models P^Y$$

- Characterization:

$$P \equiv_s Q \quad \text{iff} \quad SE(P) = SE(Q)$$

- Can we find similar characterizations for LARS?

LARS (Informally)

- Model of a program P is (essentially) a stream S at a time point t

$$S, t \models P$$

Cornerstone: Characterizing Answer Sets/Streams

ASP

- Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\ \ X \subseteq Y \qquad (ii)\ \ Y \models P \qquad (iii)\ \ X \models P^Y$$

- Characterization:

$$P \equiv_s Q \quad \text{iff} \quad SE(P) = SE(Q)$$

- Can we find similar characterizations for LARS?

LARS (Informally)

- Model of a program P is (essentially) a stream S at a time point t

$$S, t \models P$$

- Answer sets $\rightsquigarrow$ **answer streams**

Cornerstone: Characterizing Answer Sets/Streams

ASP

- Pair (X, Y) of atoms is an SE-model for program P, if

$$(i)\ \ X \subseteq Y \qquad (ii)\ \ Y \models P \qquad (iii)\ \ X \models P^Y$$

- Characterization:

$$P \equiv_s Q \quad \text{iff} \quad SE(P) = SE(Q)$$

- Can we find similar characterizations for LARS?

LARS (Informally)

- Model of a program P is (essentially) a stream S at a time point t

$$S, t \models P$$

- Answer sets $\rightsquigarrow$ **answer streams**
- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ L \subseteq R \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

Central Challenge: Window behaviour

- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

Central Challenge: Window behaviour

- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L, t) \subseteq w(R, t)$$

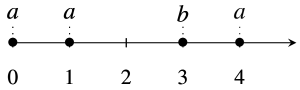
Central Challenge: Window behaviour

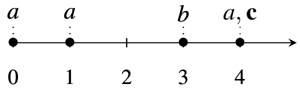
- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L, t) \subseteq w(R, t)$$





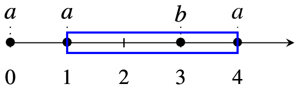
Central Challenge: Window behaviour

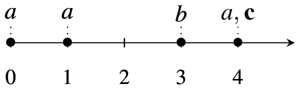
- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L, t) \subseteq w(R, t)$$





$w^3_{time}(L, 4)$

Central Challenge: Window behaviour

- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L,t) \subseteq w(R,t)$$

$w^3_{time}(L, 4)$ $\qquad$ $w^3_{time}(R, 4)$

Central Challenge: Window behaviour

- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ R, t \models P \qquad (iii)\ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L, t) \subseteq w(R, t)$$

$$w^3_{time}(L, 4) \subseteq w^3_{time}(R, 4)$$

Central Challenge: Window behaviour

- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L,t) \subseteq w(R,t)$$

$$w^3_{time}(L,4) \subseteq w^3_{time}(R,4)$$

- **Tuple-based** windows are not:

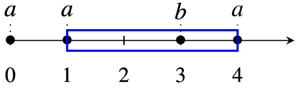
Central Challenge: Window behaviour

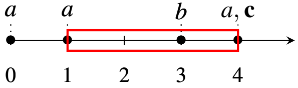
- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

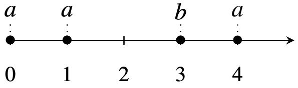
$$L \subseteq R \quad \text{implies} \quad w(L, t) \subseteq w(R, t)$$

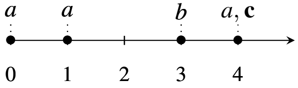




$$w^3_{time}(L, 4) \subseteq w^3_{time}(R, 4)$$

- **Tuple-based** windows are not:





Central Challenge: Window behaviour

- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\;\; \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\;\; R, t \models P \qquad (iii)\;\; L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L,t) \subseteq w(R,t)$$

$$w^3_{time}(L, 4) \subseteq w^3_{time}(R, 4)$$

- **Tuple-based** windows are not:

$$w^3_{tuple}(L, 4)$$

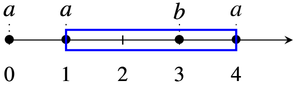
Central Challenge: Window behaviour

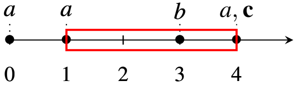
- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L} \subseteq \mathbf{R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

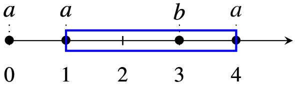
$$L \subseteq R \quad \text{implies} \quad w(L, t) \subseteq w(R, t)$$

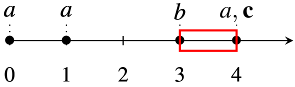




$$w^3_{time}(L, 4) \subseteq w^3_{time}(R, 4)$$

- **Tuple-based** windows are not:





$$w^3_{tuple}(L, 4) \qquad w^3_{tuple}(R, 4)$$

Central Challenge: Window behaviour

- Bi-LARS: Evaluate pair of left/right stream (L, R) s.t.

$$(i)\ \ \mathbf{L \subseteq R} \qquad (ii)\ \ R, t \models P \qquad (iii)\ \ L, t \models P^{R,t}$$

- **Time-based** window functions w are monotone:

$$L \subseteq R \quad \text{implies} \quad w(L, t) \subseteq w(R, t)$$

$$w^{3}_{time}(L, 4) \subseteq w^{3}_{time}(R, 4)$$

- **Tuple-based** windows are not:

$$w^{3}_{tuple}(L, 4) \not\subseteq w^{3}_{tuple}(R, 4)$$

Results

- Semantic characterizations of equivalences by means of models in Bi-LARS (left/right stream)
  - Monotone fragment: Generalization of logic of Here-and-There
  - Non-monotone fragement more involved

Results

- Semantic characterizations of equivalences by means of models in Bi-LARS (left/right stream)
  - Monotone fragment: Generalization of logic of Here-and-There
  - Non-monotone fragement more involved
- Notions of equivalence for stream reasoning
  - Strong / Uniform / Data Equivalence

Results

- Semantic characterizations of equivalences by means of models in Bi-LARS (left/right stream)
  - Monotone fragment: Generalization of logic of Here-and-There
  - Non-monotone fragement more involved
- Notions of equivalence for stream reasoning
  - Strong / Uniform / Data Equivalence
- Complexity of deciding eq.: similar to ASP (mostly coNP-c.)