

Synchronous and Asynchronous Stream Runtime Verification

César Sánchez

IMDEA Software Institute, Spain



Outline

- ▶ What is Stream Runtime Verification
- ▶ Synchronous SRV. (Lola)
- ▶ Asynchronous SRV (Striver)
- ▶ Synchronous \leftrightarrow Asynchronous
- ▶ Conclusion

What is Stream Runtime Verification?

Stream Runtime Verification (SRV)

SRV
is
runtime verification

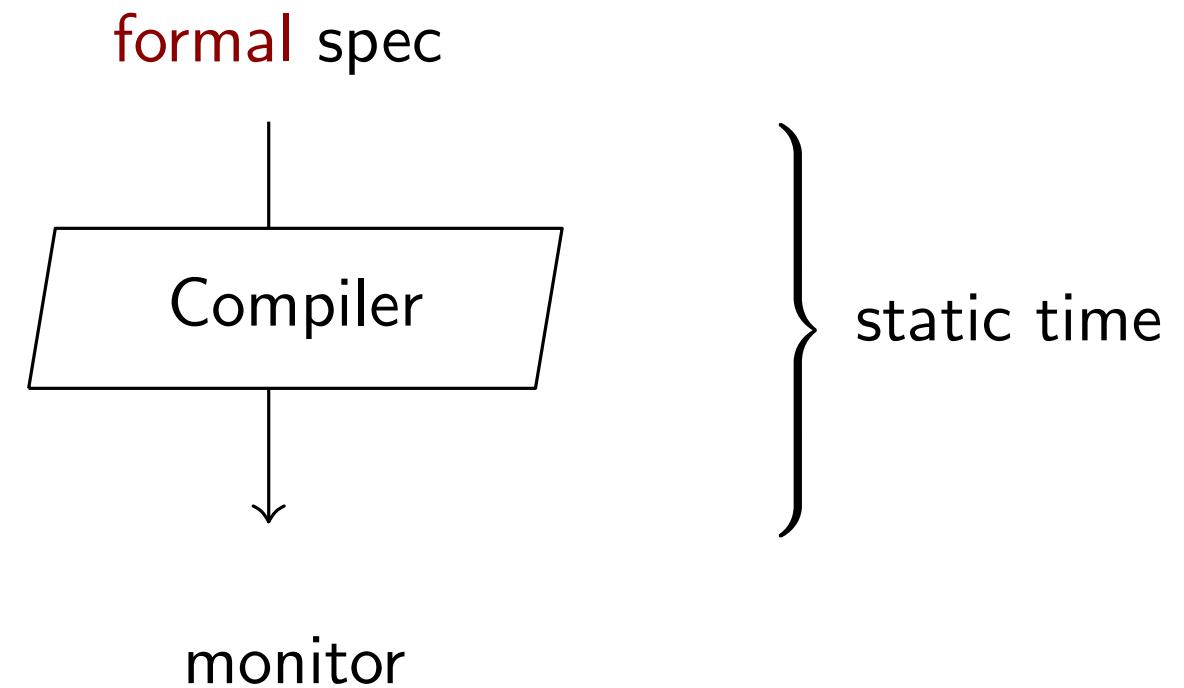
Stream Runtime Verification (SRV)

SRV
is
runtime verification

formal spec

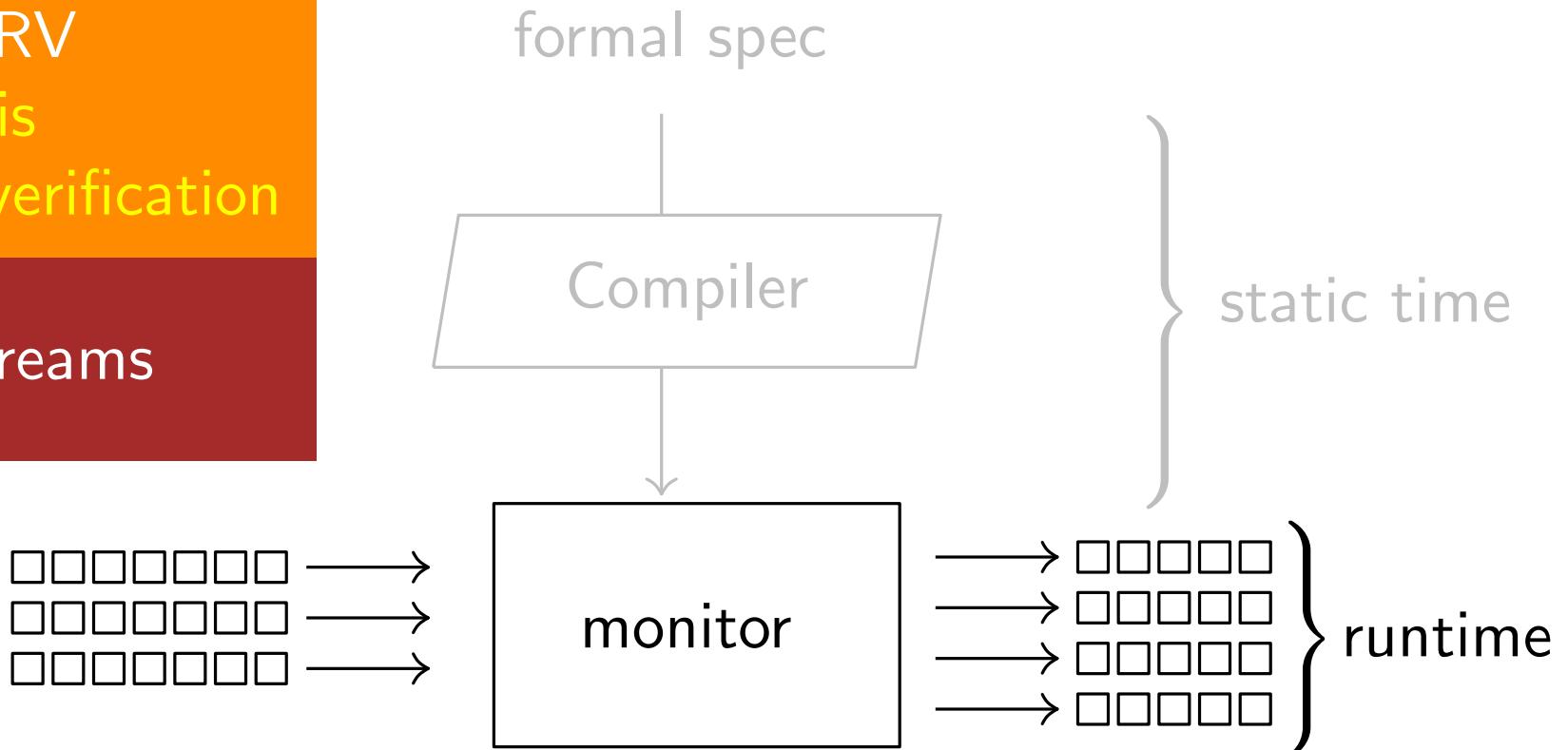
Stream Runtime Verification (SRV)

SRV
is
runtime verification



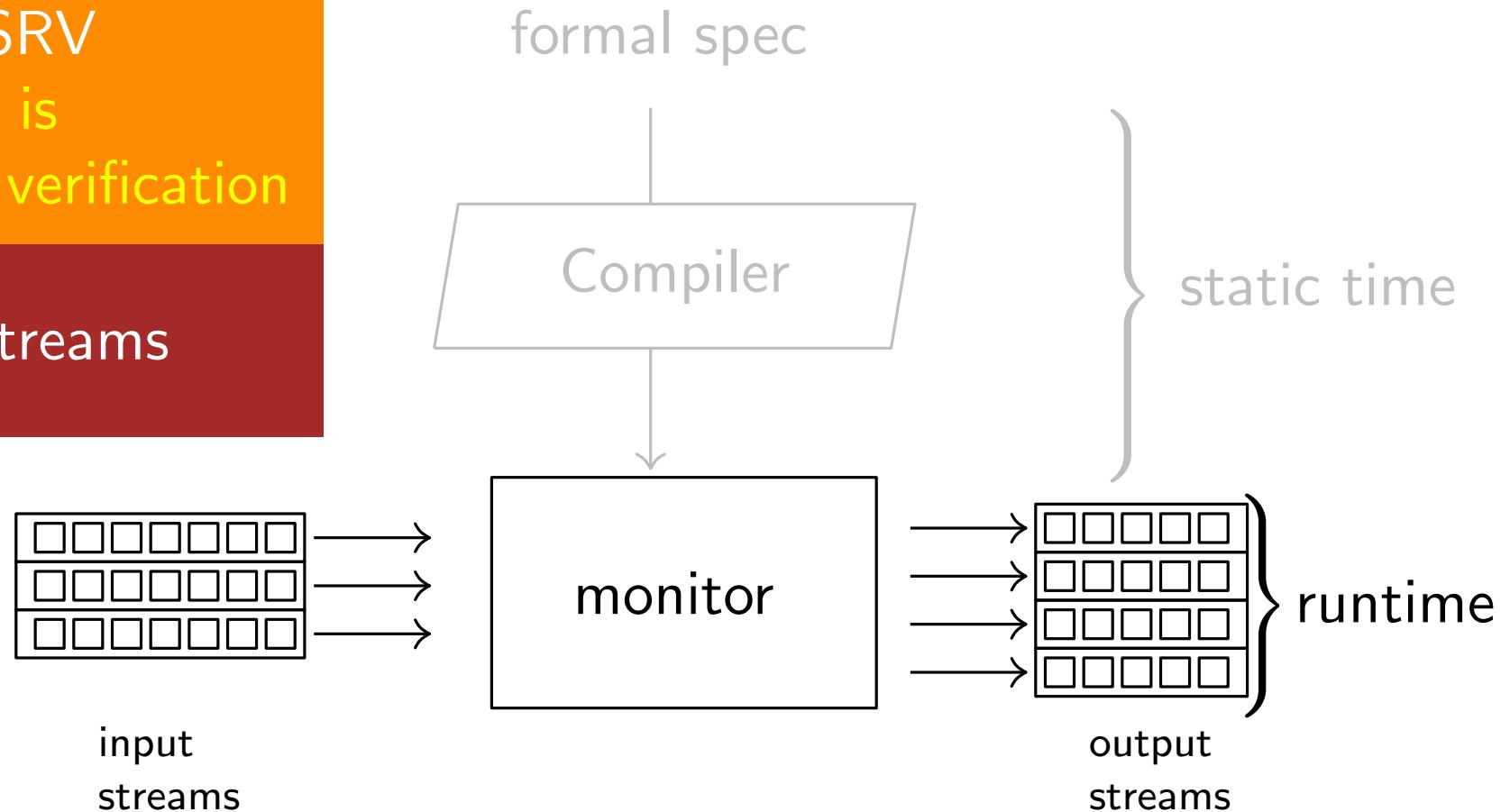
Stream Runtime Verification (SRV)

SRV
is
runtime verification
for streams



Stream Runtime Verification (SRV)

SRV
is
runtime verification
for streams



Stream Runtime Verification (SRV)

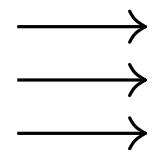
SRV
is
runtime verification
for streams

is

runtime verification

for streams

input
streams



formal spec

Compiler

monitor

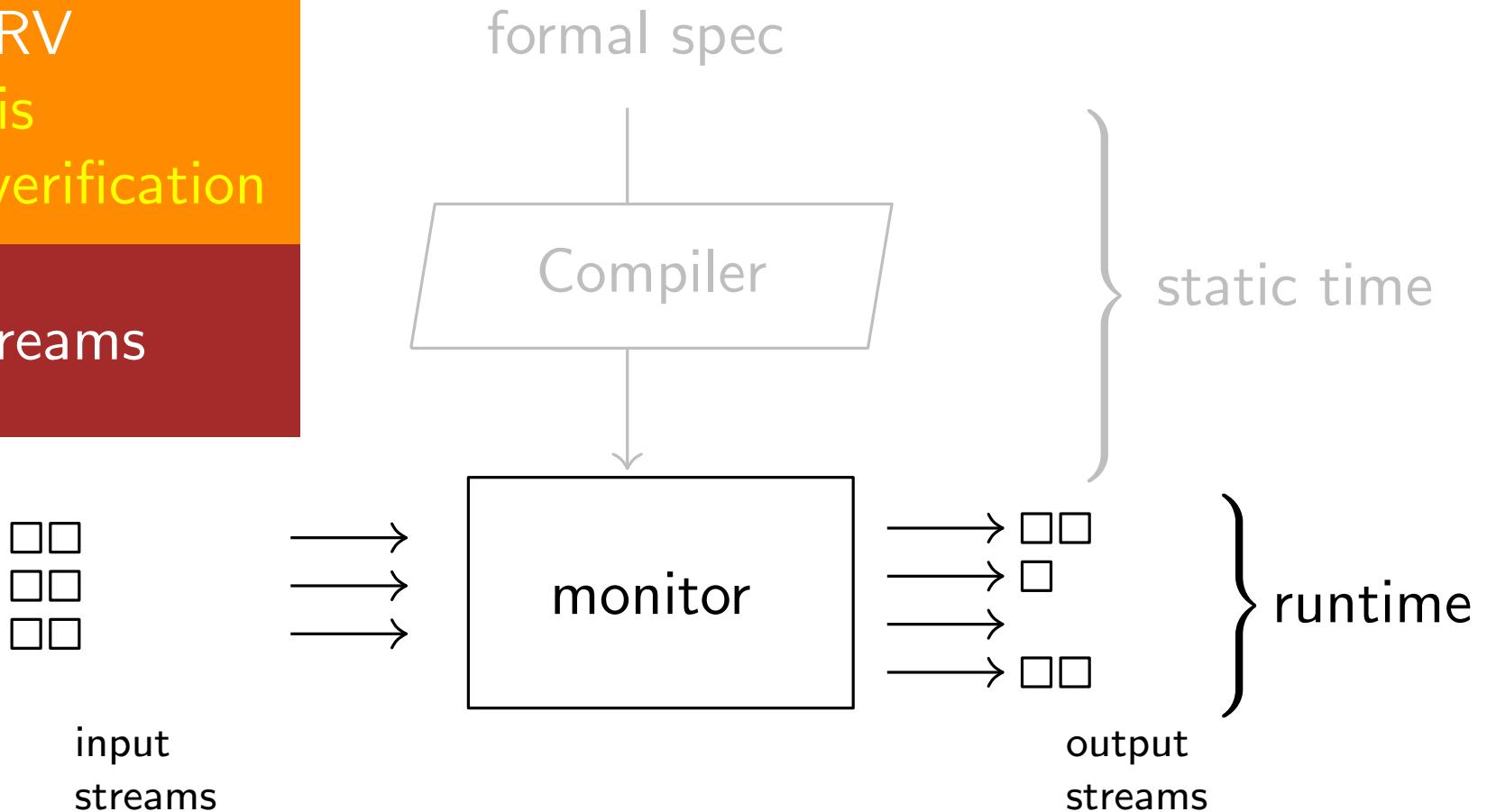
static time

runtime

output
streams

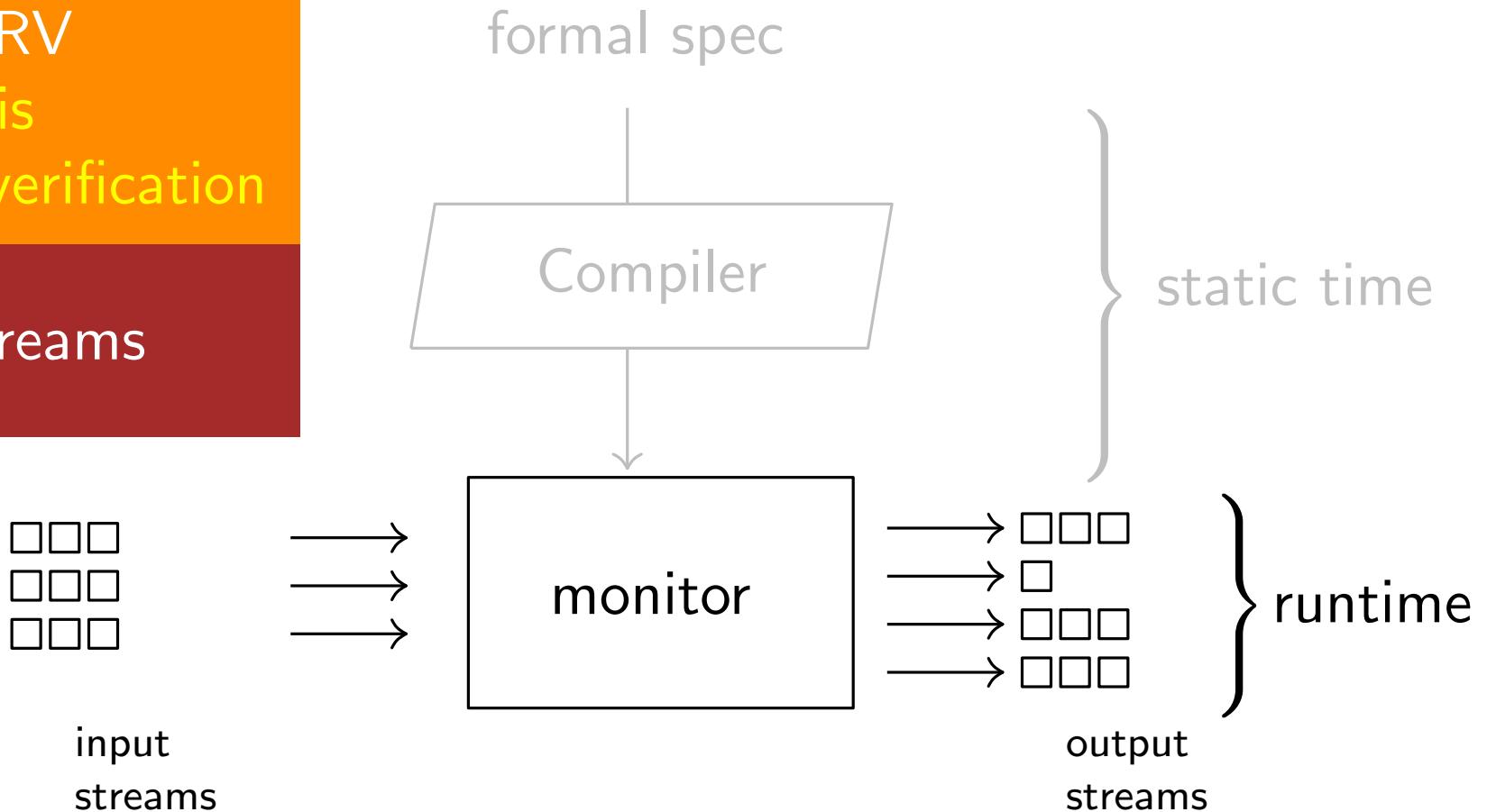
Stream Runtime Verification (SRV)

SRV
is
runtime verification
for streams



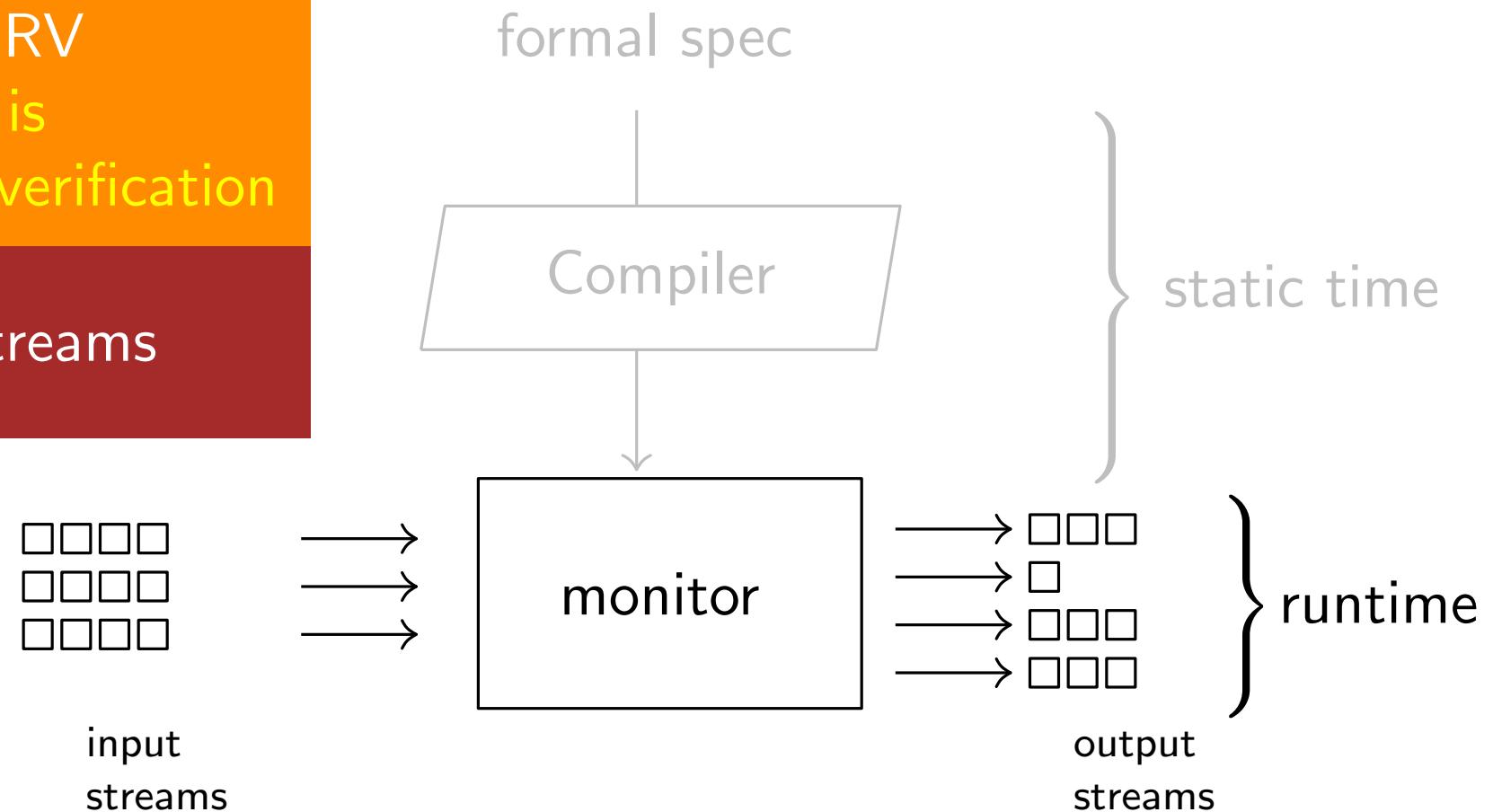
Stream Runtime Verification (SRV)

SRV
is
runtime verification
for streams



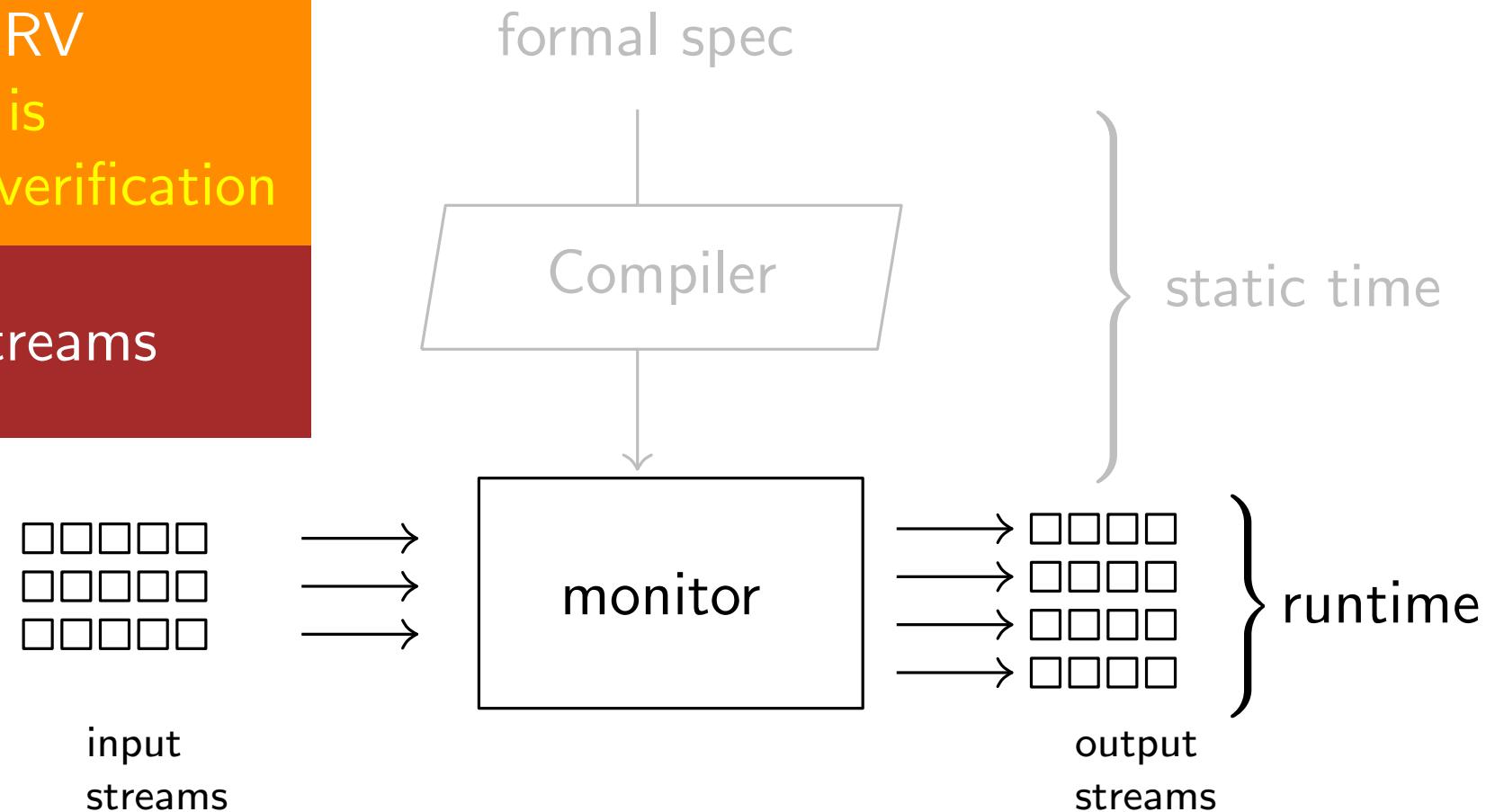
Stream Runtime Verification (SRV)

SRV
is
runtime verification
for streams



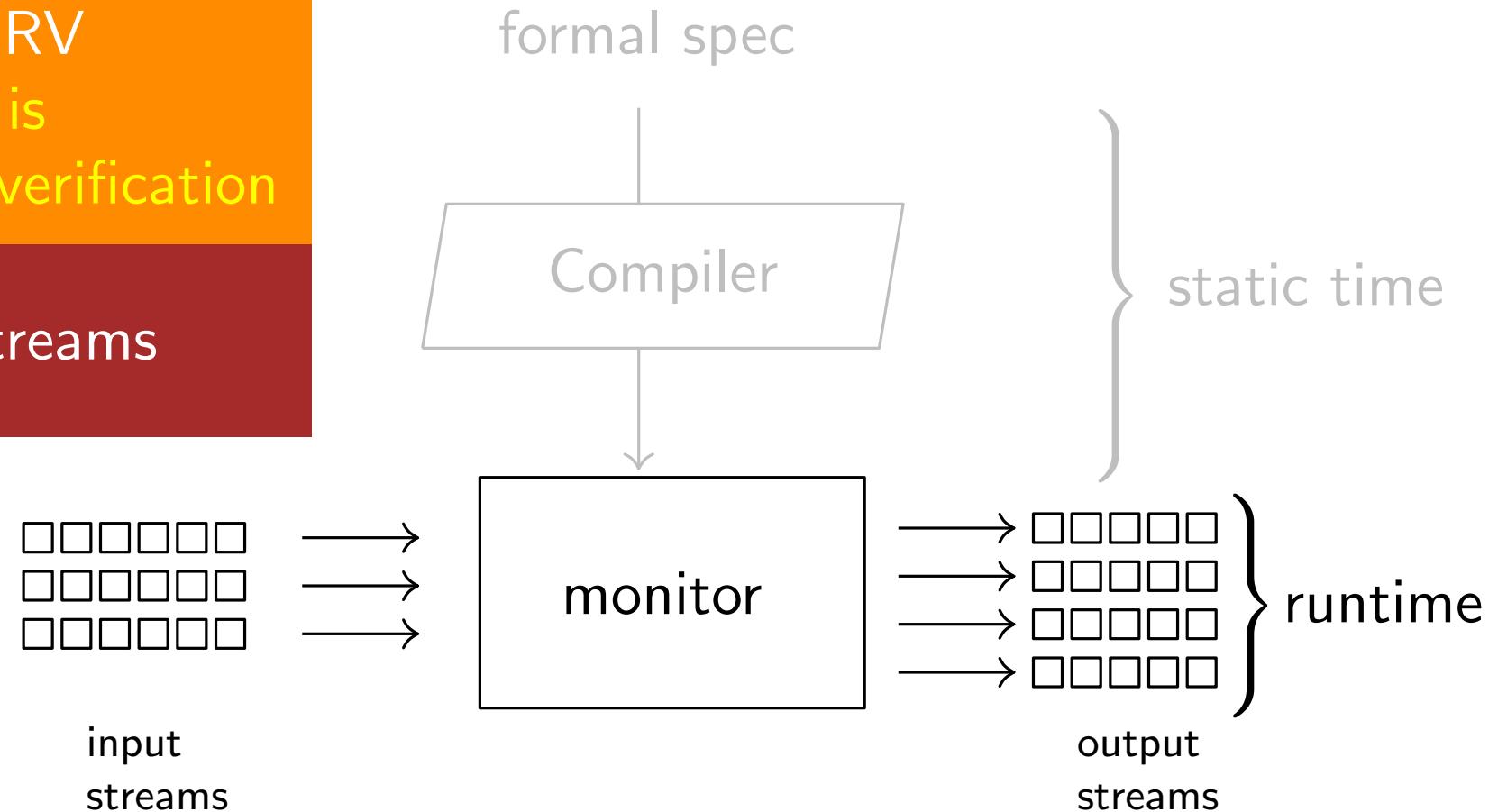
Stream Runtime Verification (SRV)

SRV
is
runtime verification
for streams



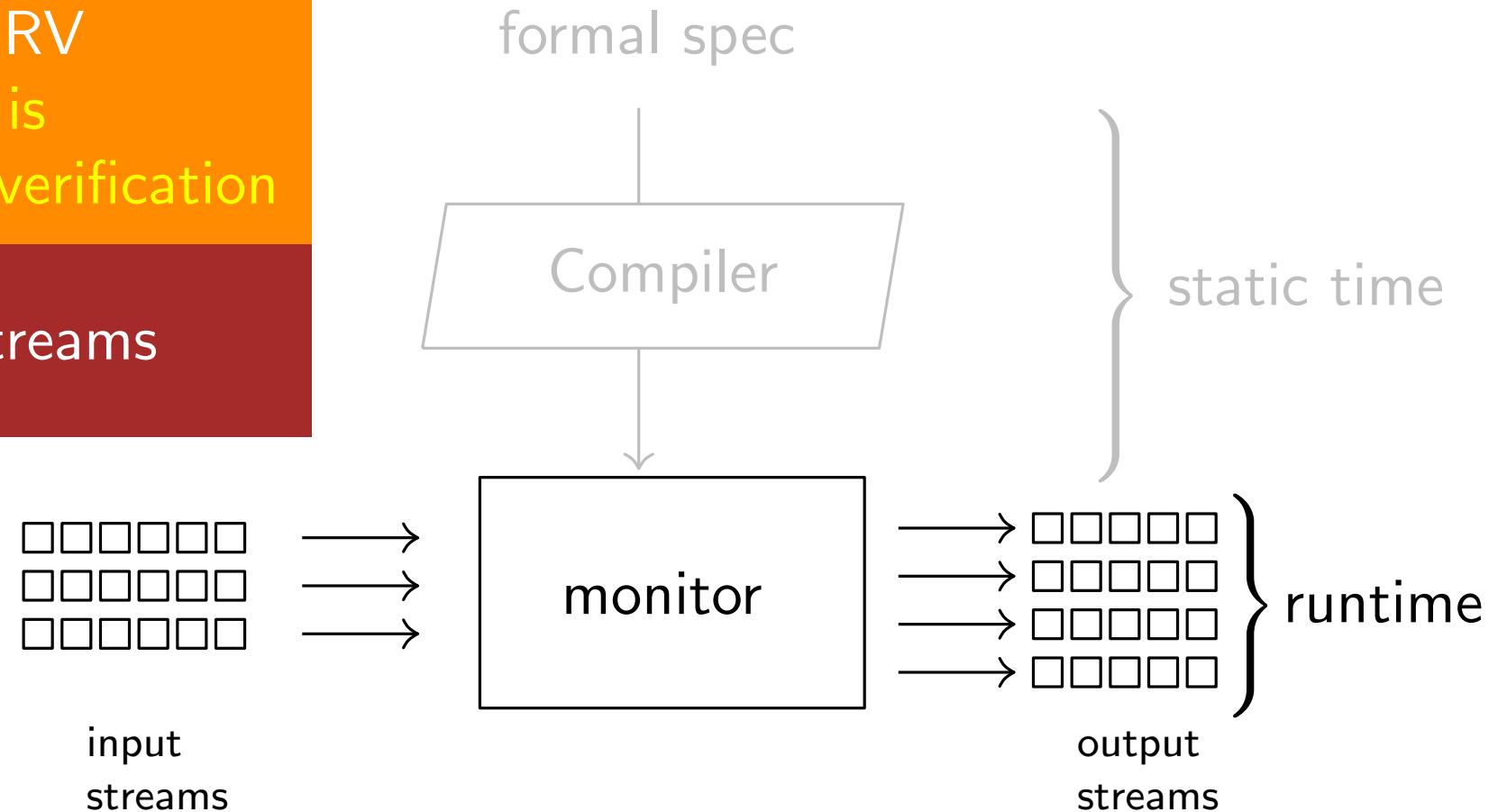
Stream Runtime Verification (SRV)

SRV
is
runtime verification
for streams



Stream Runtime Verification (SRV)

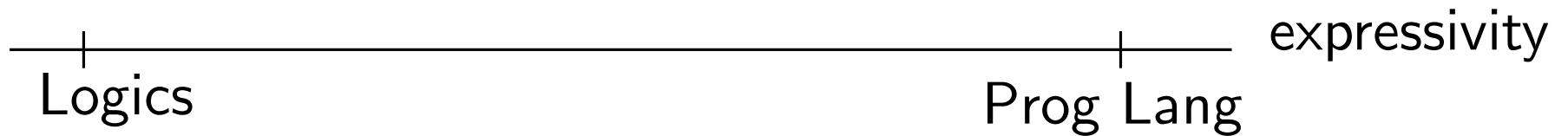
SRV
is
runtime verification
for streams



Monitor time **vs** System time

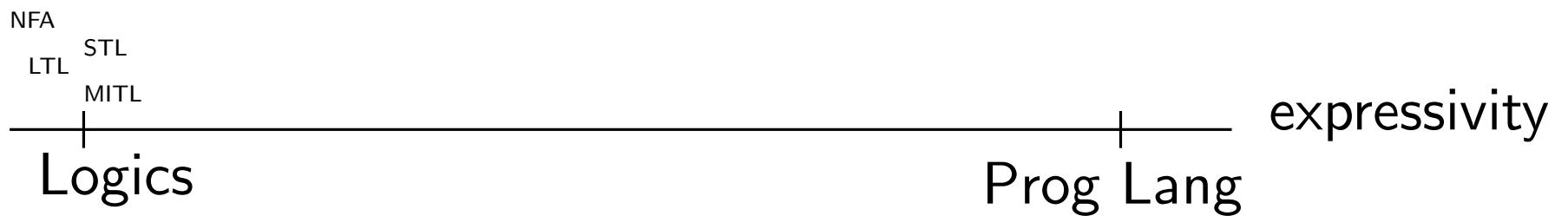
Goal (expressivity)

Expressivity Landscape



Goal (expressivity)

Expressivity Landscape



Goal (expressivity)

Expressivity Landscape



Goal (expressivity)

Expressivity Landscape



Goal (expressivity)

Expressivity Landscape



Goal (expressivity)

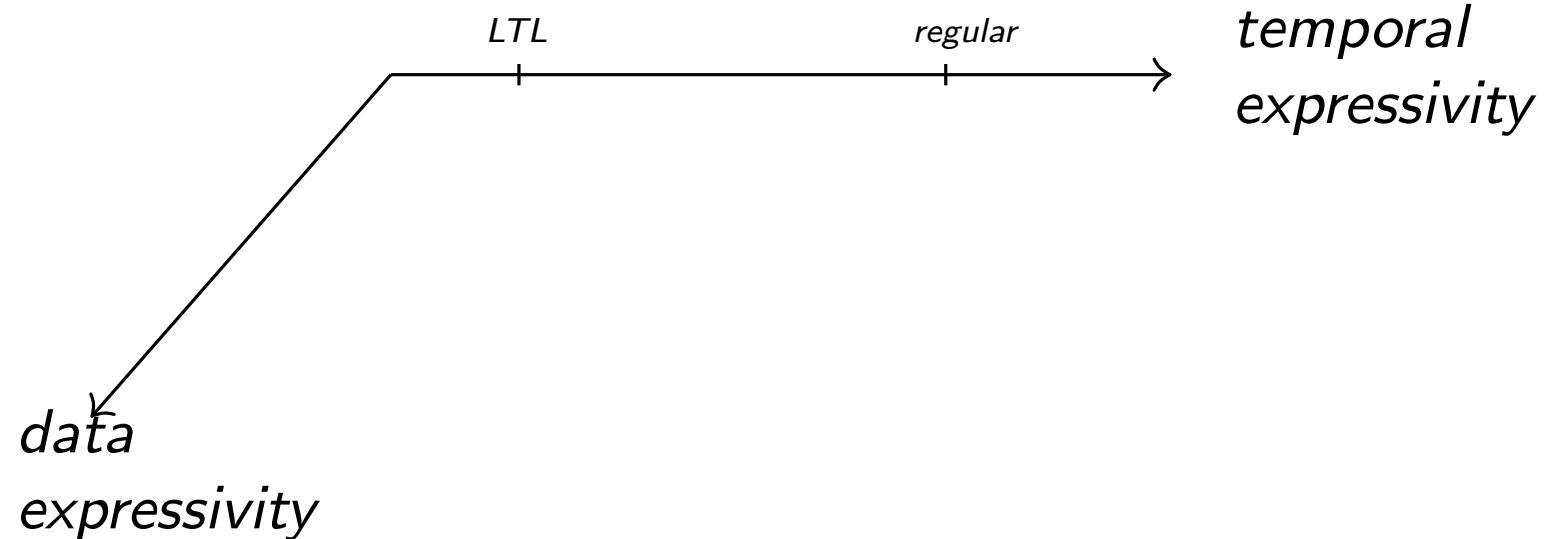
Expressivity Landscape



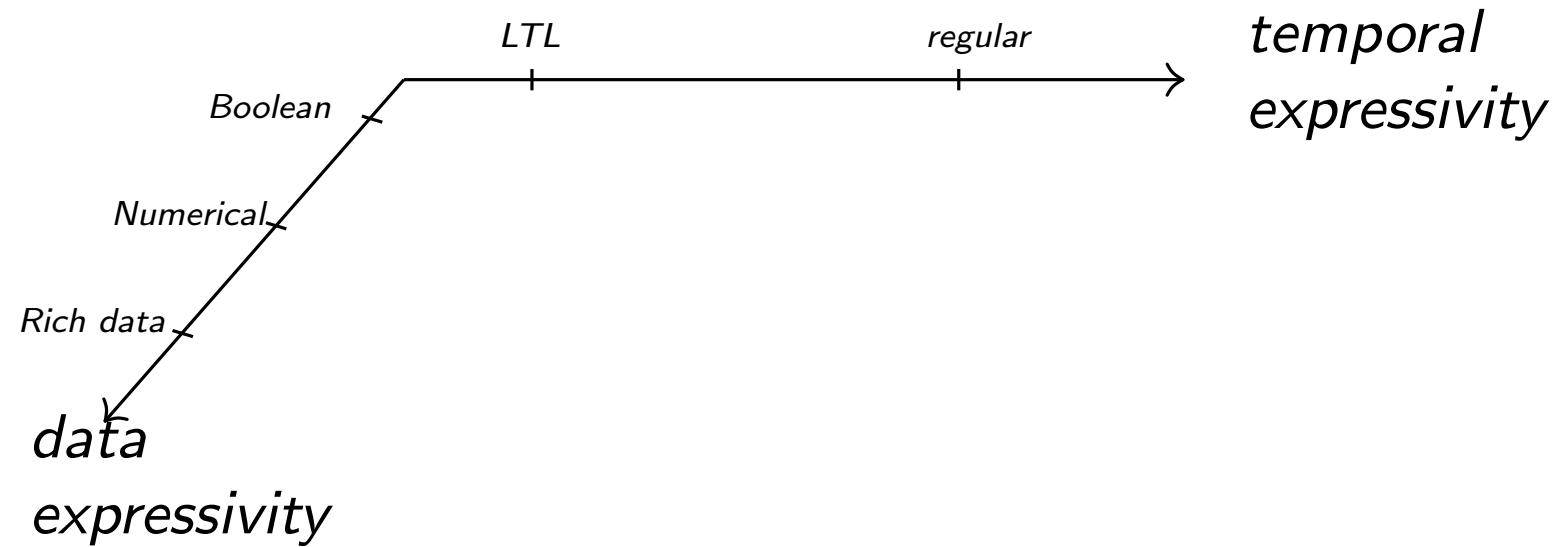
Landscape of SRV



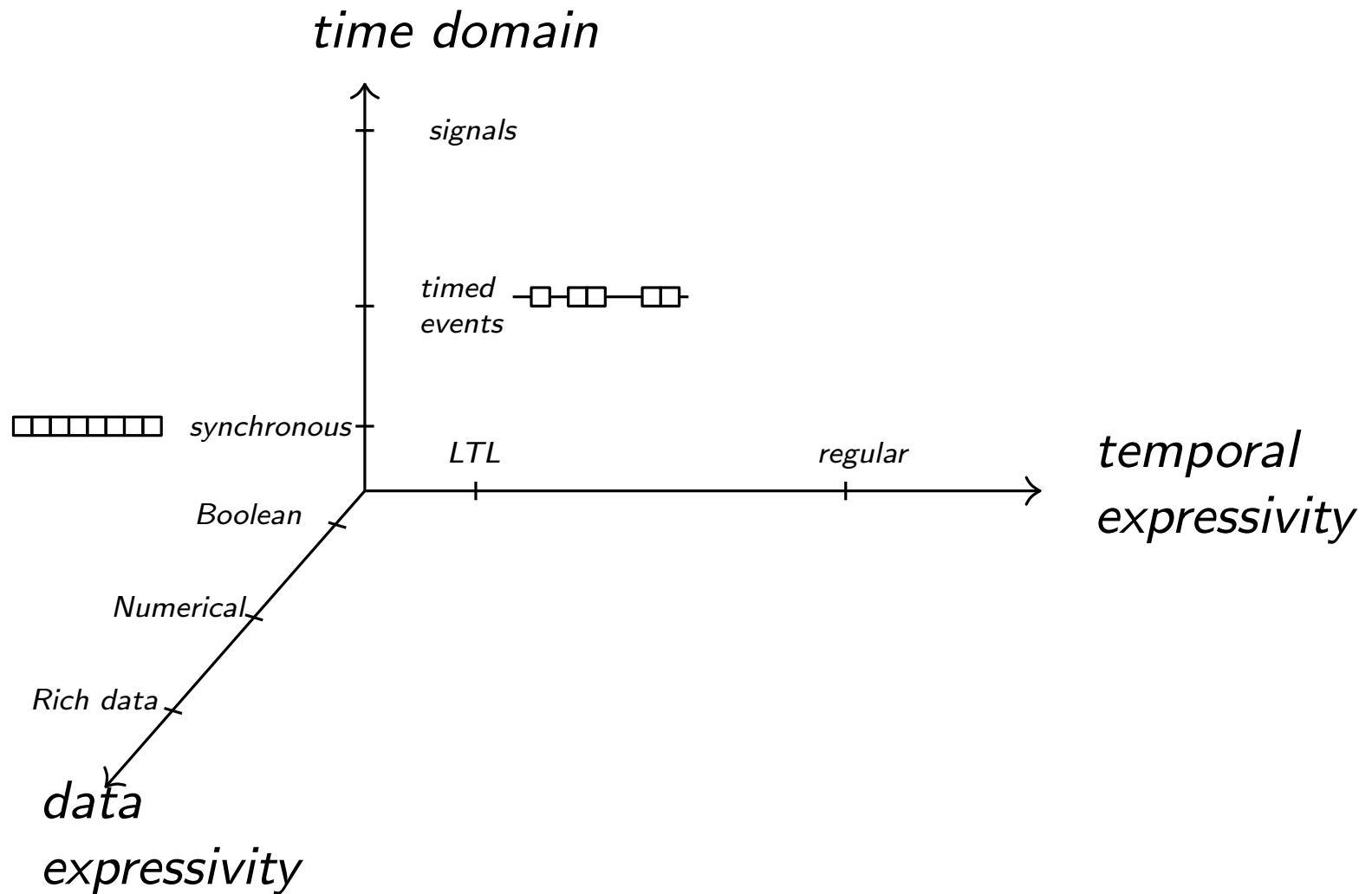
Landscape of SRV



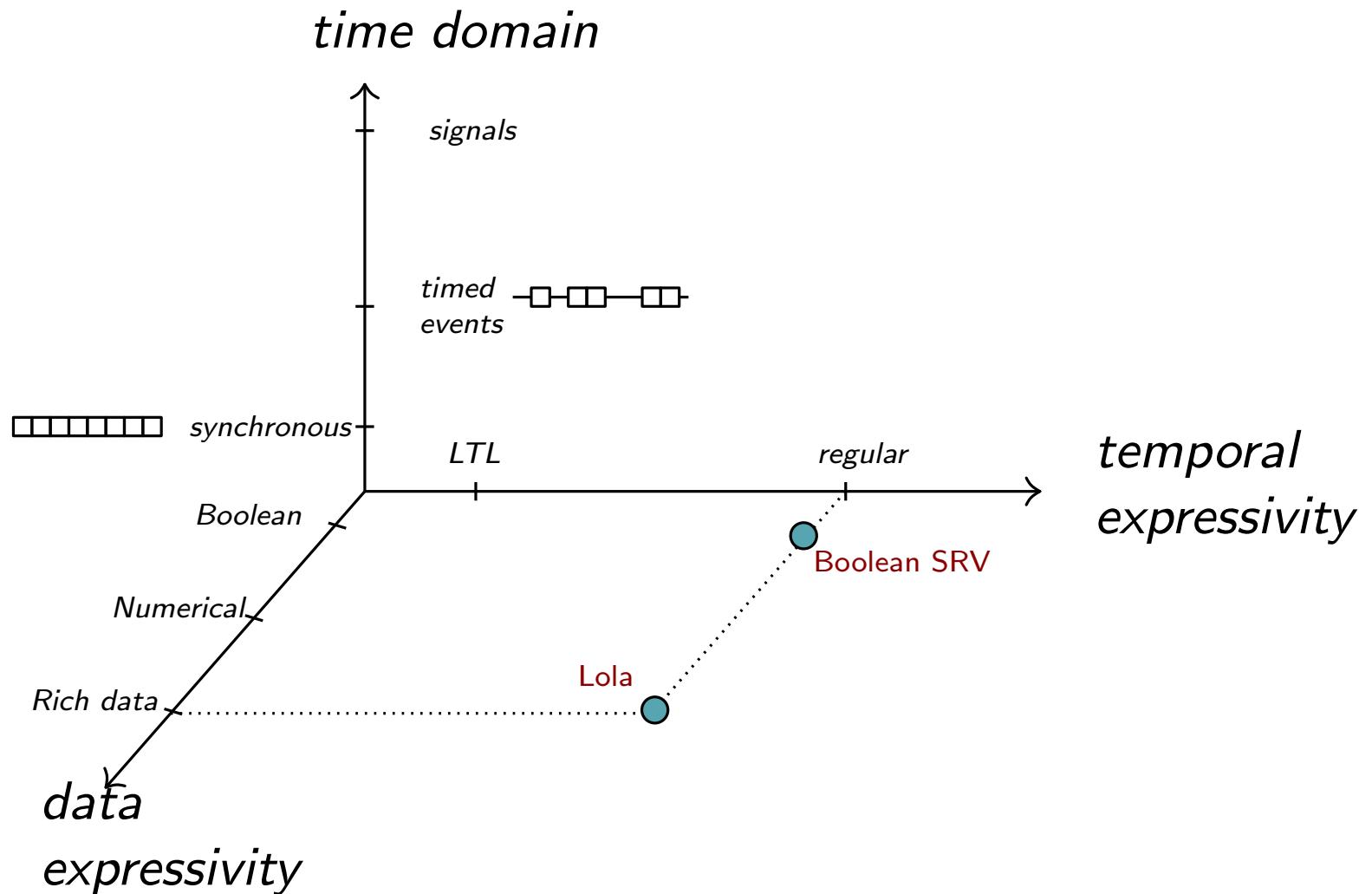
Landscape of SRV



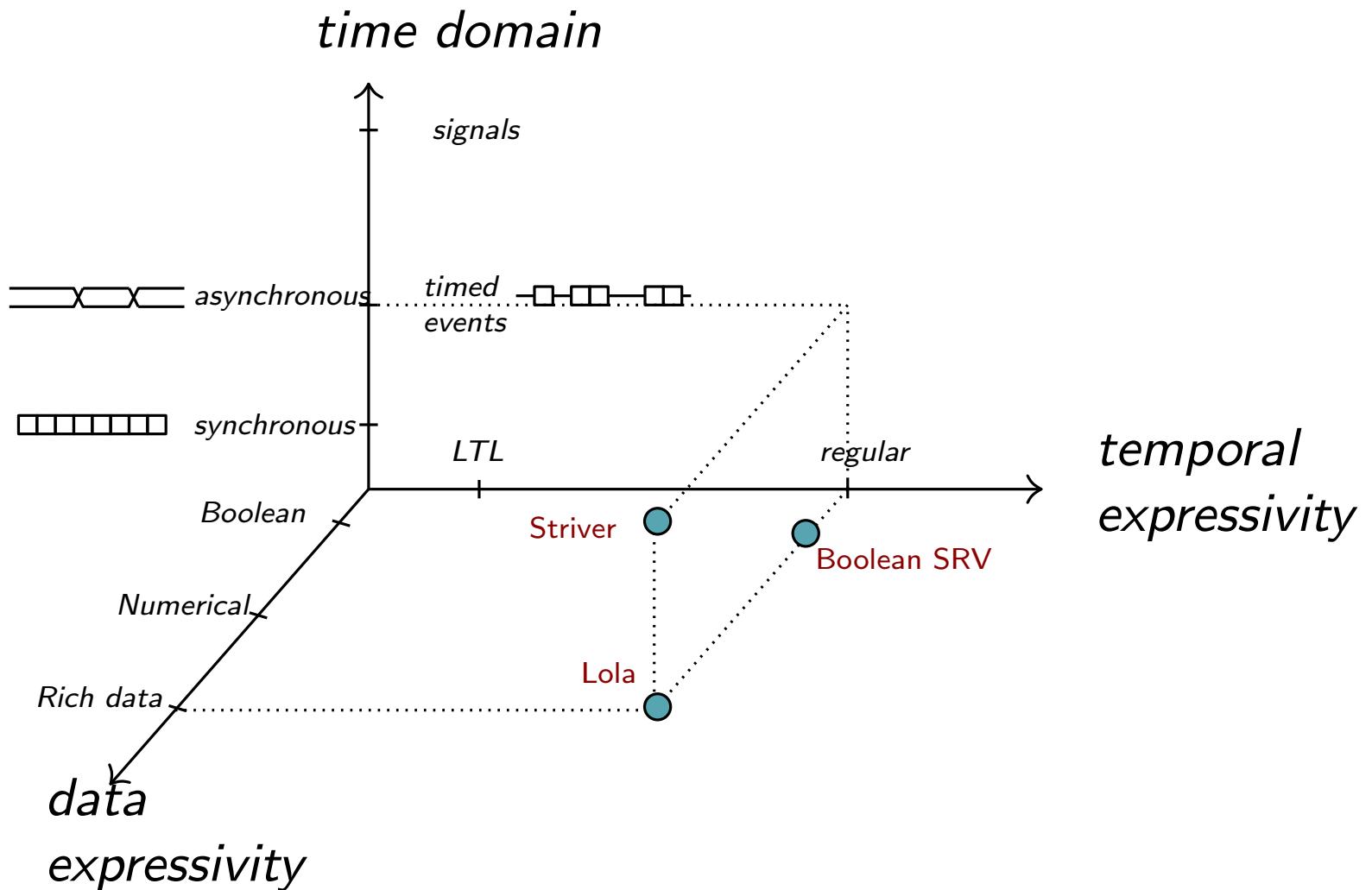
Landscape of SRV



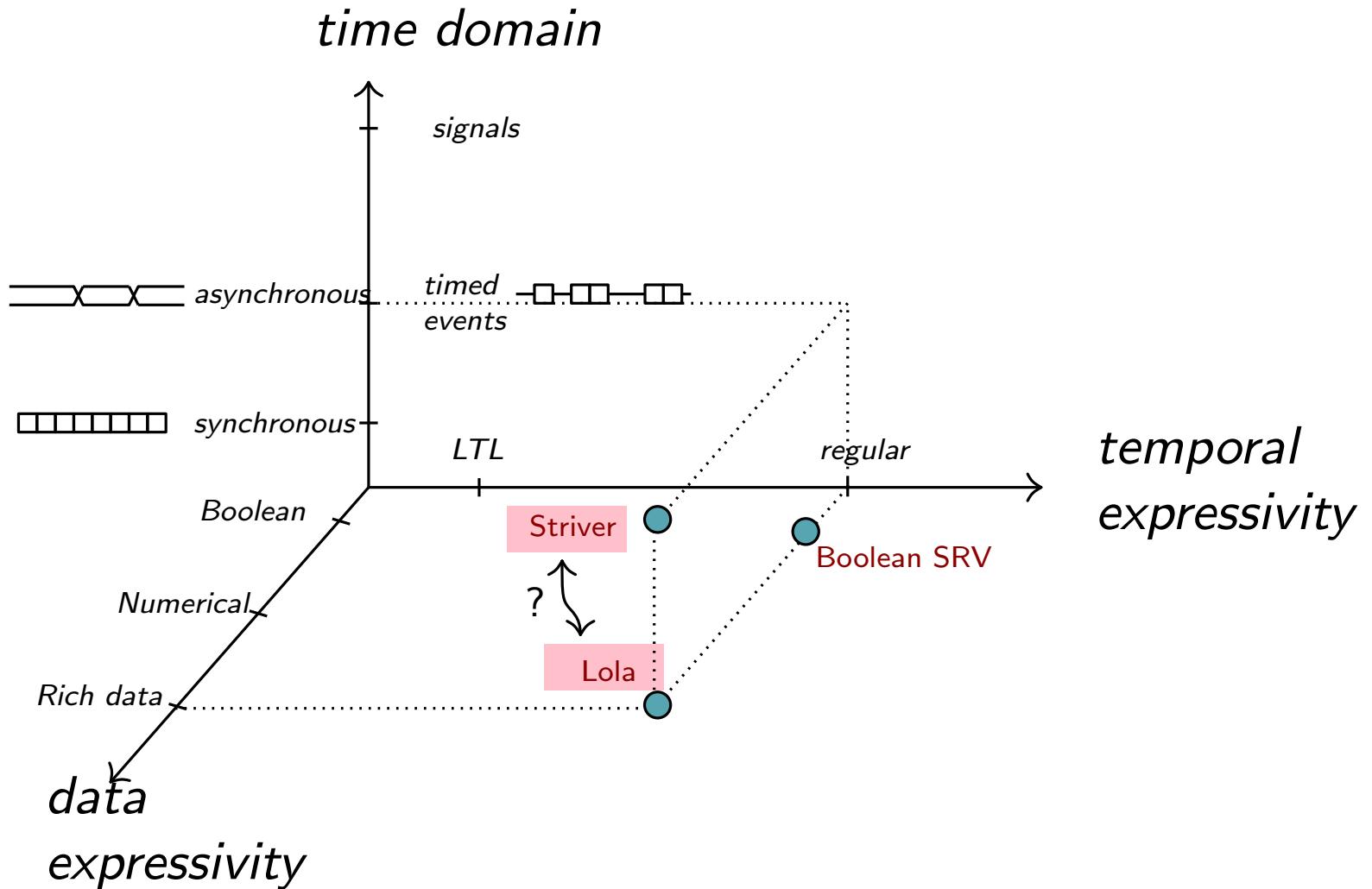
Landscape of SRV



Landscape of SRV



Landscape of SRV



History of Stream Runtime Verification

- ▶ B. D'Angelo, S. Sankaranarayanan, César Sánchez, W. Robinson, B. Finkbeiner, H. Sipma, S. Mehrotra, Z. Manna: *LOLA: Runtime Monitoring of Synchronous Systems*. TIME 2005
- ▶ F. Gorostiaga, C. Sánchez: *Striver: Stream Runtime Verification for Real-Time Signals and Event-Streams* RV'2018
- ▶ L. Pike, A. Goodloe, R. Morisset, S. Niller: *Copilot: A Hard Real-Time Runtime Monitor*. RV 2010
- ▶ L.M. Danielsson, C. Sánchez: *Decentralized Stream Runtime Verification* RV'2019.
- ▶ T. Reinbacher, K. Rozier, J. Schumann: *Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems*. TACAS 2014
- ▶ M. Leucker, C. Sánchez, T. Scheffel, M. Schmitz, D. Thoma: *Runtime Verification for Timed Event Streams with Partial Information*. RV 2019
- ▶ P. Faymonville, B. Finkbeiner, S. Schirmer, H. Torfah: *A Stream-Based Specification Language for Network Monitoring*. RV 2016
- ▶ F. Gorostiaga, L.M. Danielsson, C. Sánchez: *Unifying the Time-Event Spectrum for Stream Runtime Verification* RV'2020.
- ▶ F. Adolf, P. Faymonville, B. Finkbeiner, S. Schirmer, C. Torens: *Stream Runtime Monitoring on UAS*. RV 2017
- ▶ M. Ceresa, F. Gorostiaga, C. Sánchez: *Declarative Stream Runtime Verification (hLola)* APLAS'2020.
- ▶ L. Bozzelli, C. Sánchez: *Foundations of Boolean Stream Runtime Verification* RV 2014
- ▶ F. Gorostiaga, C. Sánchez: *Stream Runtime Verification of Real-Time Event-Streams with the Striver Language* STTT 2021.
- ▶ L. Convent, S. Hungerecker, M. Leucker, T. Scheffel, M. Schmitz, D. Thoma: *TeSSLa: Temporal Stream-Based Specification Language*. BSMF 2018
- ▶ F. Gorostiaga, C. Sánchez: *HLola: a Very Functional Tool for Extensible Stream Runtime Verification* TACAS'2021.
- ▶ M. Leucker, C. Sánchez, T. Scheffel, M. Schmitz, A. Schramm: *TeSSLa: Runtime Verification of Non-synchronized Real-Time Streams*. SAC 2018
- ▶ S. Zudaire, F. Gorostiaga, C. Sánchez, G. Schneider, S. Uchitel *Assumption Monitoring Using Runtime Verification for UAV Temporal Task Plan Executions*, ICRA'2021.

History of Stream Runtime Verification

- B. D'Angelo, S. Sankaranarayanan, César Sánchez, W. Robinson, B. Finkbeiner, H. Sipma, S. Mehrotra, Z. Manna: *LOLA: Runtime Monitoring of Synchronous Systems*. TIME 2005

- F. Gorostiaga, C. Sánchez: *Striver: Stream Runtime Verification for Real-Time Signals and Event-Streams* RV'2018

- L. Pike, A. Goodloe, R. Morisset, S. Niller: *Copilot: A Hard Real-Time Runtime Monitor*. RV 2010.



Sánchez: *Decentralized Stream Runtime Verification*
zier, J. Schumann: *Temporal-Logic Based Runtime Verification*
CAS 2014

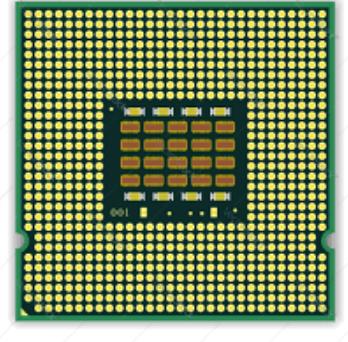


nent of

Partial

- Information.* RV 2019

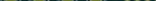
- P. Faymonville, B. Finkbeiner, S. Schirmer, H. 2016



Spectrum for Stream Runtime Verification RV'2020.

- F. Goerke, S. Sankaranarayanan, S. Srivastava, and M. S. Sooriyakumaran
► F. Adel, S. Sankaranarayanan, S. Srivastava, and M. S. Sooriyakumaran



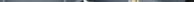
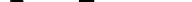
chez:  Spectrum for Stream Runtime Verification RV'2020.

- F. Adl-Post, S. Söderström, J. M. Rasmussen / Runtime Monitoring on UAS, RV 2017

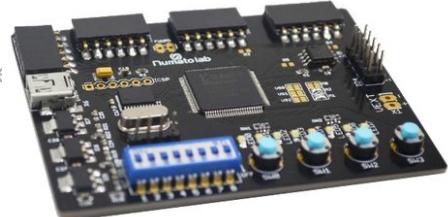
- ▶ M_Ce Declassified 1/2) API AS'2020

- Page 14 —

- © E. G. S. 2023. All rights reserved. This material may not be published, reproduced, displayed or distributed without the express written permission of the author.

- Network  T-SQL   

- Language. BSMF 2018



- S. Zudaire, F. Gorostiaga, C. Sánchez, G. Schneider, S. Uchitel *Assumption Monitoring Using Runtime Verification for UAV Temporal Task Plan Executions*, ICRA'2021.

Goal (expressivity)

Consider the following LTL specs:

$$\square p$$

Goal (expressivity)

Consider the following LTL specs:

 $\square p$

```
define bool s := p /\ s[ 1,true]
```

Goal (expressivity)

Consider the following LTL specs:

 $\Box p$

```
define bool s := p /\ s[ 1,true]
```

 $\neg p$ $\Diamond p$ $\lozenge p$ $p \mathcal{U} q$ $p \mathcal{W} q$ $\bigcirc p$

Goal (expressivity)

Consider the following LTL specs:

 $\Box p$

```
define bool s := p /\ s[ 1,true]
```

 $\neg p$

```
define bool s := p /\ s[-1,true]
```

 $\Diamond p$

```
define bool s := p \vee s[ 1,false]
```

 $\lozenge p$

```
define bool s := p \vee s[-1,false]
```

 $p \mathcal{U} q$

```
define bool s := q \vee (p /\ s[1|false])
```

 $p \mathcal{W} q$

```
define bool s := q \vee (p /\ s[1,true])
```

 $\bigcirc p$

```
define bool s := p[1,false]
```

Goal (expressivity)

Consider the following LTL specs:

Why restrict to Booleans?

 $\Box p$

```
define bool s := p /\ s[ 1,true]
```

 $\neg p$

```
define bool s := p /\ s[-1,true]
```

 $\Diamond p$

```
define bool s := p \vee s[ 1,false]
```

 $\lozenge p$

```
define bool s := p \vee s[-1,false]
```

 $p \mathcal{U} q$

```
define bool s := q \vee (p /\ s[1|false])
```

 $p \mathcal{W} q$

```
define bool s := q \vee (p /\ s[1,true])
```

 $\bigcirc p$

```
define bool s := p[1,false]
```

Separation of concerns

A **runtime verification** algorithm deals with two aspects:

- ▶ the ***when***: a sequence of temporal data dependencies necessary to compute the verdict (**time**)
- ▶ the ***what***: the details of how to compute each action (**the data**)

Separation of concerns

A **runtime verification** algorithm deals with two aspects:

- ▶ the ***when***: a sequence of temporal data dependencies necessary to compute the verdict (**time**)
- ▶ the ***what***: the details of how to compute each action (**the data**)

Stream Runtime Verification Moto:

Monitoring = **temporal dependencies** +
data manipulation

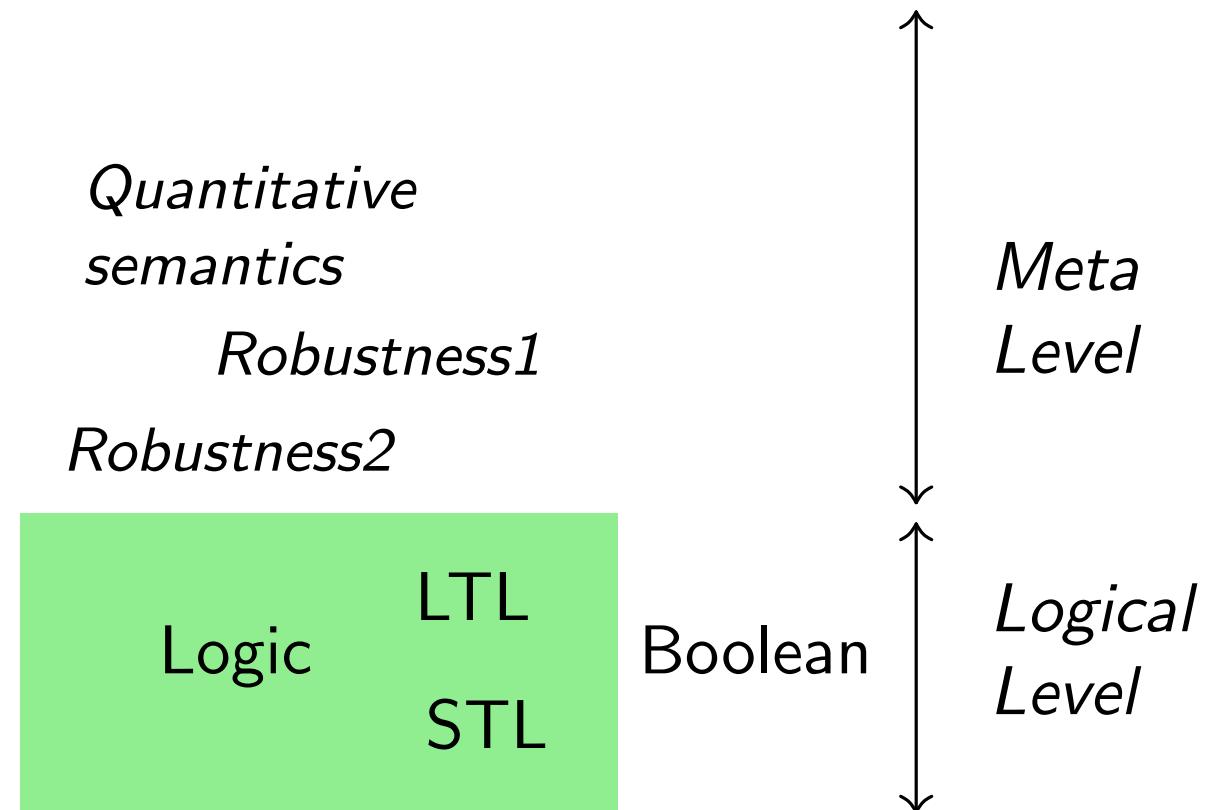
Goal (expressivity)



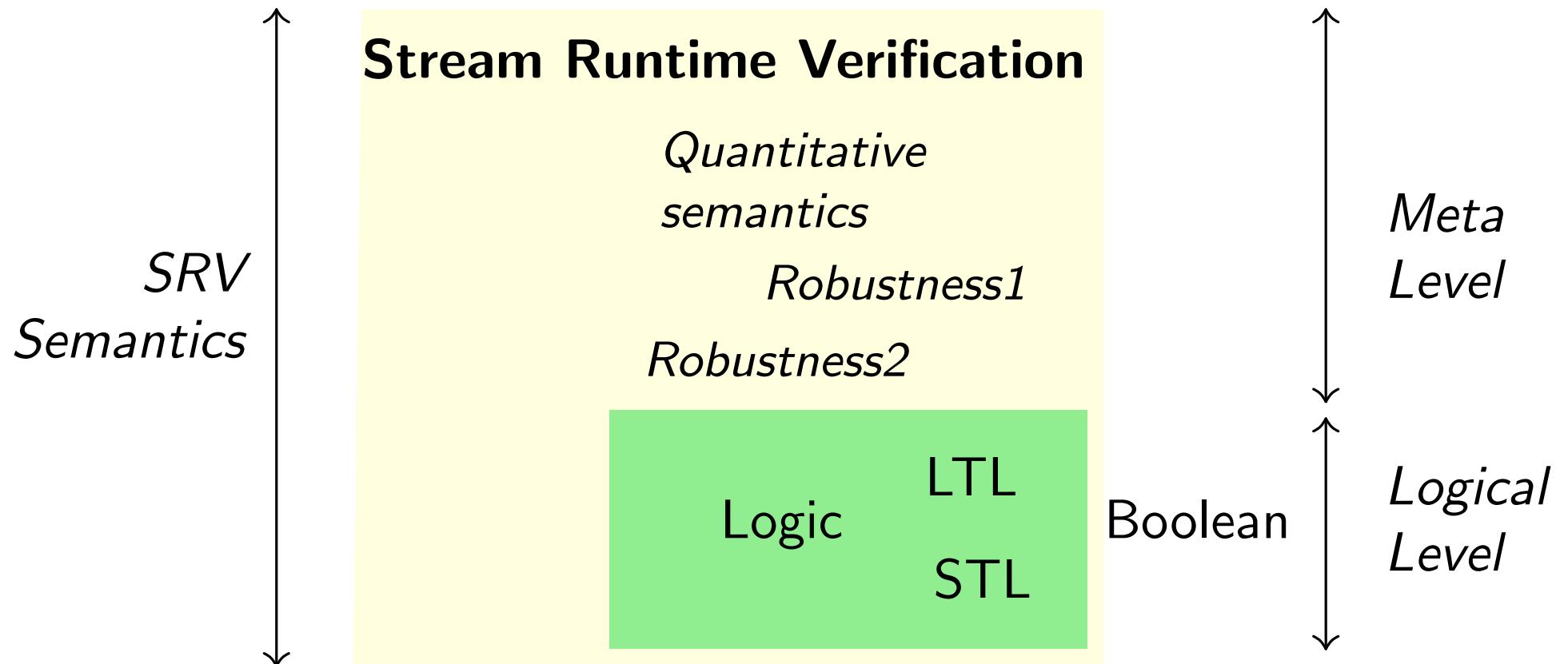
Goal (expressivity)



Goal (expressivity)



Goal (expressivity)



Synchronous SRV **Lola**

Data Domains

Stream Runtime Verification Moto:

Monitoring = +
 temporal dependencies
 data manipulation

Example

Every request has a response

Example

Every request has a response

$$\square(\textcolor{blue}{req} \rightarrow \diamond \textcolor{blue}{resp})$$

Example

Every request has a response

$$\square(\text{req} \rightarrow \diamond \text{resp})$$

With Booleans:

```
input  bool req
input  bool resp

define bool evresp  := resp \vee evresp[1|false]
define bool granted := req -> evresp
output bool ok      := granted /\ ok[1|true]
```

Example

Every request has a response

$$\square(\text{req} \rightarrow \diamond \text{resp})$$

With Integers:

```
input  bool  req
input  bool  resp

define int  nreq  := nreq[-1|0] + if req  then 1 else 0
define int  nresp := nres[-1|0] + if resp then 1 else 0
output bool  ok    := last -> nreq=nresp
```

Example

Every request has a response

$$\square(\text{req} \rightarrow \diamond \text{resp})$$

With Integers:

```
input  bool  req
input  bool  resp

define int  nreq  := count(req)
define int  nresp := count(resp)
output bool  ok    := last -> nreq=nresp
```

where

```
output last := false[1,true]
```

Example

Every request has a response

$$\square(\text{req} \rightarrow \diamond \text{resp})$$

With Integers:

```
input  bool  req
input  bool  resp

define int  nreq  := nreq[-1|0] + if req  then 1 else 0
define int  nresp := nres[-1|0] + if resp then 1 else 0
output bool  ok    := last -> nreq=nresp
```

An additional sanity check:

```
output bool  good   := nresp <= nreq
output bool  G_good := G_good[-1,true] /\ good
```

Example

Every request has a response

$$\square(\text{req} \rightarrow \diamond \text{resp})$$

With Sets:

```
input  bool req
input  bool resp
input  ID reqid, respid

define set pend := pend[-1|emp] union (if req then [reqid] else emp )
                                \ (if resp then [respid] else emp)
output bool ok      := pending = emp
```

Example using Bool3 and Bool4

 $\Box p$

```
define bool s := p /\ s[ 1,true]
```

 $\neg p$

```
define bool s := p /\ s[-1,true]
```

 $\Diamond p$

```
define bool s := p \vee s[ 1,false]
```

 $\lozenge p$

```
define bool s := p \vee s[-1,false]
```

 $p \mathcal{U} q$

```
define bool s := q \vee (p /\ s[1|false])
```

 $p \mathcal{W} q$

```
define bool s := q \vee (p /\ s[1,true])
```

 $\bigcirc p$

```
define bool s := p[1,false]
```

Example using Bool3 and Bool4

 $\Box p$

```
define bool3 s := p /\ s[ 1,?]
```

 $\neg p$

```
define bool3 s := p /\ s[-1,true]
```

 $\Diamond p$

```
define bool3 s := p \vee s[ 1,?]
```

 $\lozenge p$

```
define bool3 s := p \vee s[-1,false]
```

 $p \mathcal{U} q$

```
define bool3 s := q \vee (p /\ s[1,?])
```

 $p \mathcal{W} q$

```
define bool3 s := a \vee (p /\ s[1,?])
```

 $\bigcirc p$

```
define bool3 s := p[1,?]
```

Example using Bool3 and Bool4

 $\Box p$

```
define bool4 s := p /\ s[ 1,true?]
```

 $\neg p$

```
define bool4 s := p /\ s[-1,true]
```

 $\Diamond p$

```
define bool4 s := p \/\ s[ 1,false?]
```

 $\lozenge p$

```
define bool4 s := p \/\ s[-1,false]
```

 $p \mathcal{U} q$

```
define bool4 s := q \/\ (p /\ s[1,false?])
```

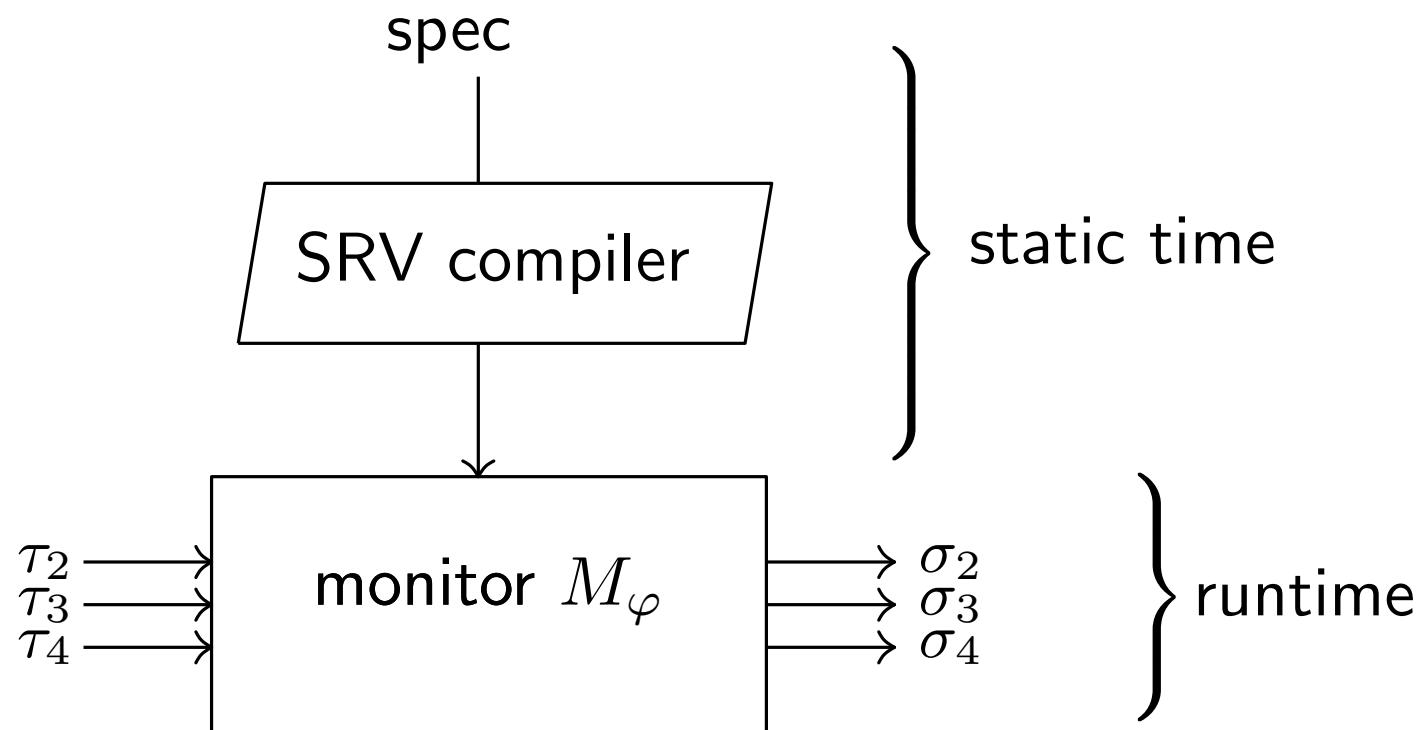
 $p \mathcal{W} q$

```
define bool4 s := a \/\ (p /\ s[1,true?])
```

 $\bigcirc p$

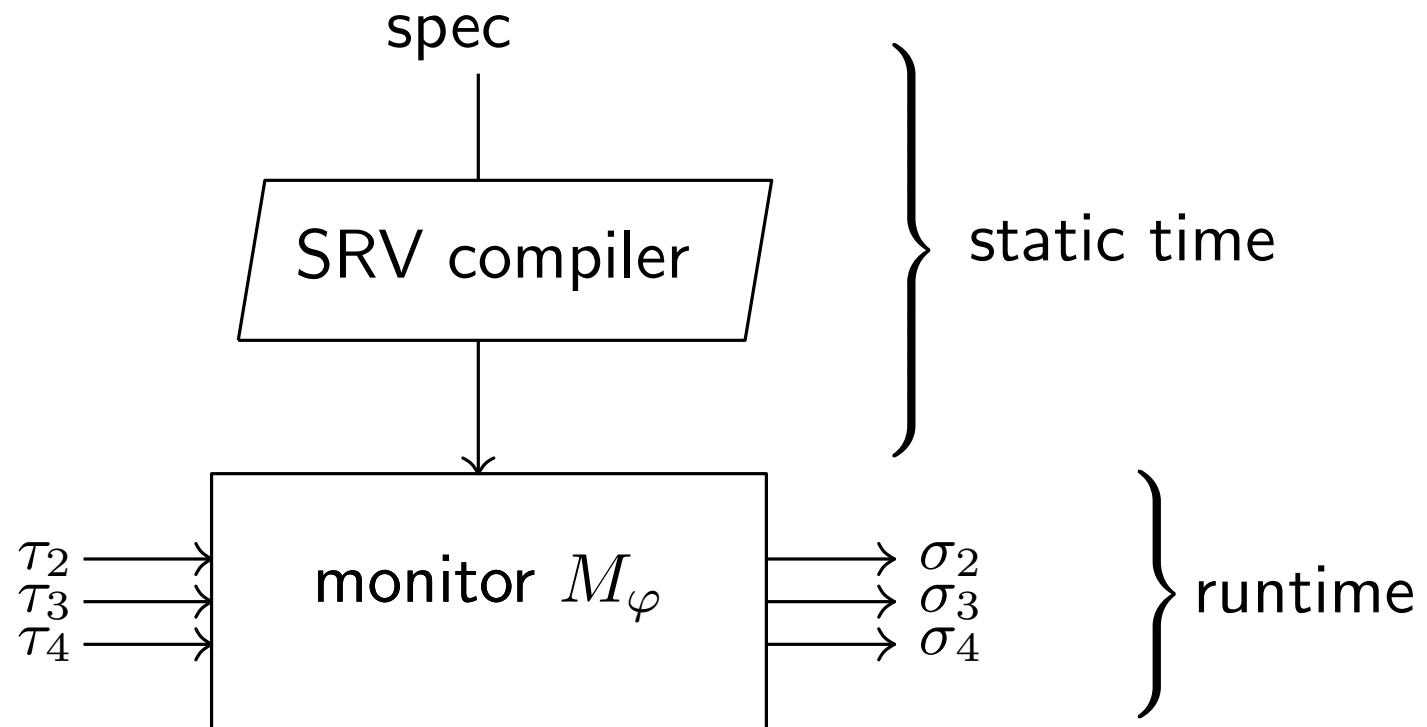
```
define bool4 s := p[1,false?]
```

Semantics (intention)



Intention: M_φ is a “*function*” from inputs to outputs

Semantics (intention)



Intention: M_φ is a “*function*” from inputs to outputs

Online monitoring: M_φ computes incrementally

Dependency Graph and Evaluation Graph

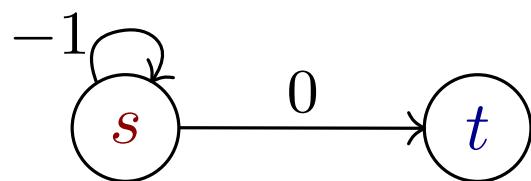
The *dependency graph* G captures the temporal dependencies

Dependency Graph and Evaluation Graph

The *dependency graph* G captures the temporal dependencies

Example $\exists t$

```
input  bool  t
output bool s := s[-1,true] \vee t[now]
```

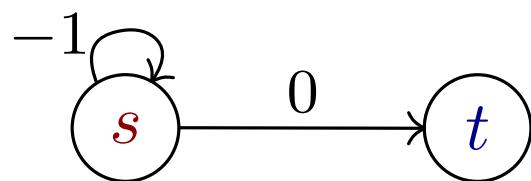


Dependency Graph and Evaluation Graph

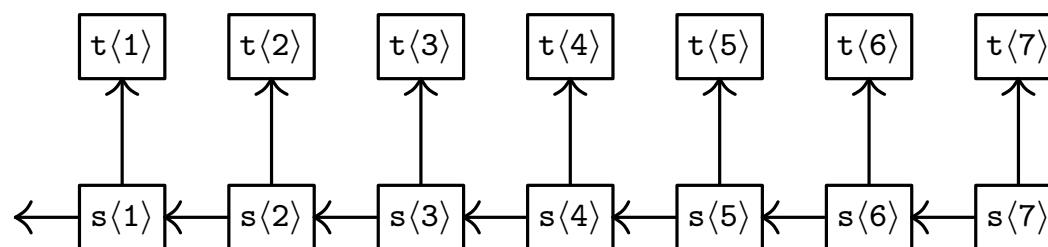
The *dependency graph* G captures the temporal dependencies

Example $\exists t$

```
input  bool  t
output bool s := s[-1,true] \vee t[now]
```



For length 7 the *evaluation graph* is:

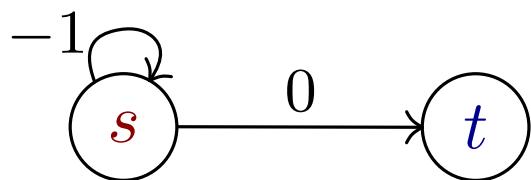


Dependency Graph and Evaluation Graph

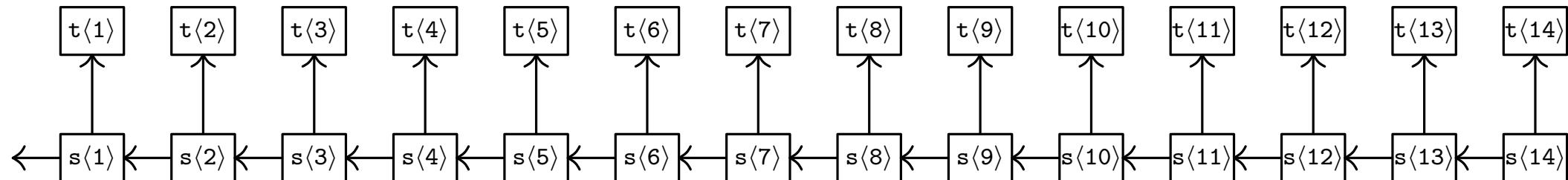
The *dependency graph* G captures the temporal dependencies

Example $\exists t$

```
input  bool  t
output bool s := s[-1,true] \vee t[now]
```



For length 14 the *evaluation graph* is:



Realistic Example (Havesine for geographic distance)

```
input  double lat, lon, ug, vg, wg, time

define double diff_lon := lon[-1,0.0] - lon
define double diff_lat := lat[-1,0.0] - lat
define double a := sin((diff_lat/2.0))^2.0 +
                  cos(lat[-1,0.0]) * cos(lat) * (sin(diff_lon/2.0))^2.0
define double gps_distance := R * 2.0 * atan2(sqrt(a), sqrt(1.0-a))

define double velocity := sqrt(ug * ug + vg * vg + wg * wg)
define double elapse    := time - time[-1,0.0]

output bool jump := !first /\ (gps_distance - velocity * elapse) > 1.0
```

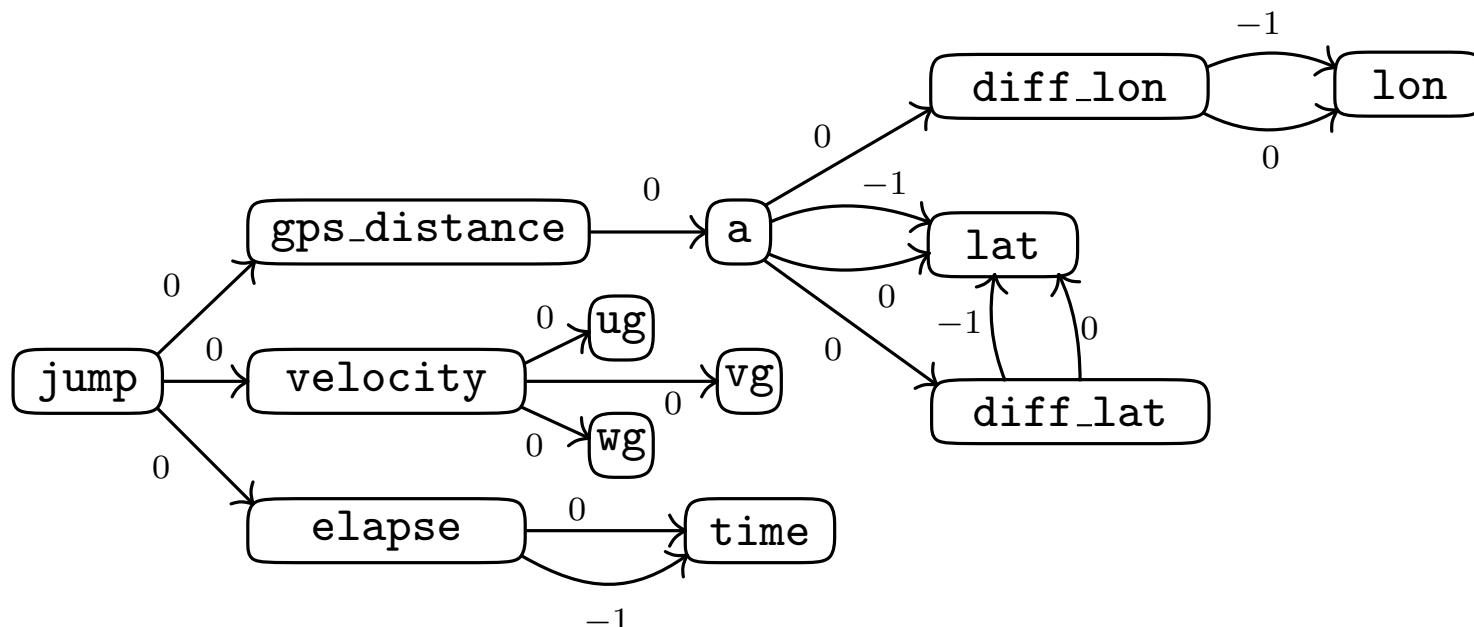
Realistic Example (Haversine for geographic distance)

```
input  double lat, lon, ug, vg, wg, time

define double diff_lon := lon[-1,0.0] - lon
define double diff_lat := lat[-1,0.0] - lat
define double a := sin((diff_lat/2.0))^2.0 +
                  cos(lat[-1,0.0]) * cos(lat) * (sin(diff_lon/2.0))^2.0
define double gps_distance := R * 2.0 * atan2(sqrt(a), sqrt(1.0-a))

define double velocity := sqrt(ug * ug + vg * vg + wg * wg)
define double elapse   := time - time[-1,0.0]

output bool jump := !first /\ (gps_distance - velocity * elapse) > 1.0
```



Asynchronous SRV Striver

Striver

Striver extends SRV to real-time **asynchronous** streams

Striver

Striver extends SRV to real-time **asynchronous** streams

F. Gorostiaga, C. Sánchez: *Striver: Stream Runtime Verification for Real-Time Signals and Event-Streams* RV'2018

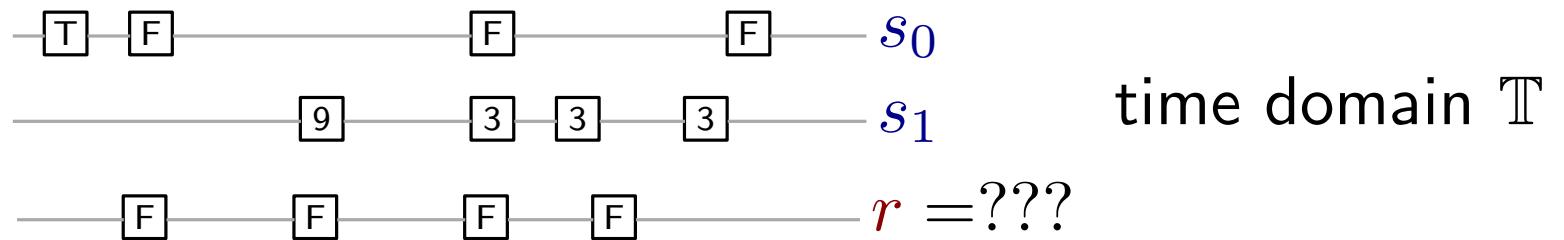
F. Gorostiaga, C. Sánchez: *Stream Runtime Verification of Real-Time Event-Streams with the Striver Language* STTT 2021.

Striver

Striver extends SRV to real-time **asynchronous** streams

F. Gorostiaga, C. Sánchez: *Striver: Stream Runtime Verification for Real-Time Signals and Event-Streams* RV'2018

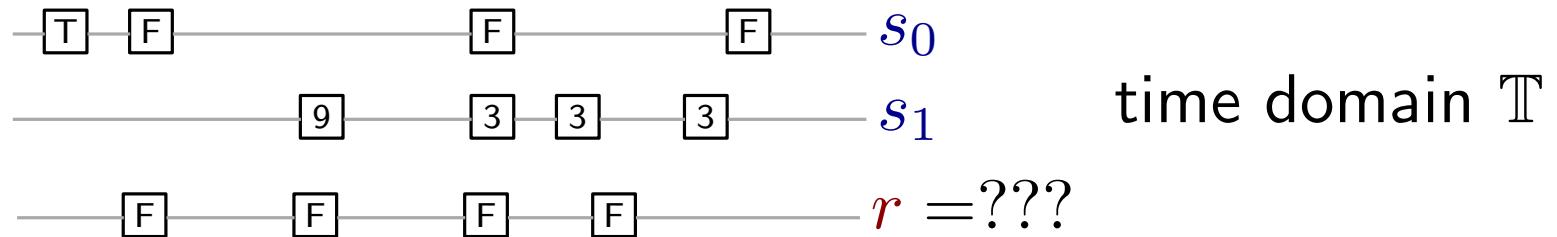
F. Gorostiaga, C. Sánchez: *Stream Runtime Verification of Real-Time Event-Streams with the Striver Language* STTT 2021.



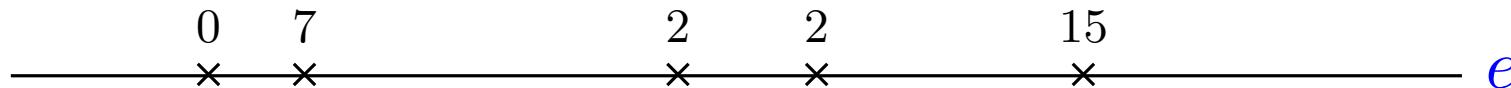
Striver extends SRV to real-time **asynchronous** streams

F. Gorostiaga, C. Sánchez: *Striver: Stream Runtime Verification for Real-Time Signals and Event-Streams* RV'2018

F. Gorostiaga, C. Sánchez: *Stream Runtime Verification of Real-Time Event-Streams with the Striver Language* STTT 2021.



► Event streams



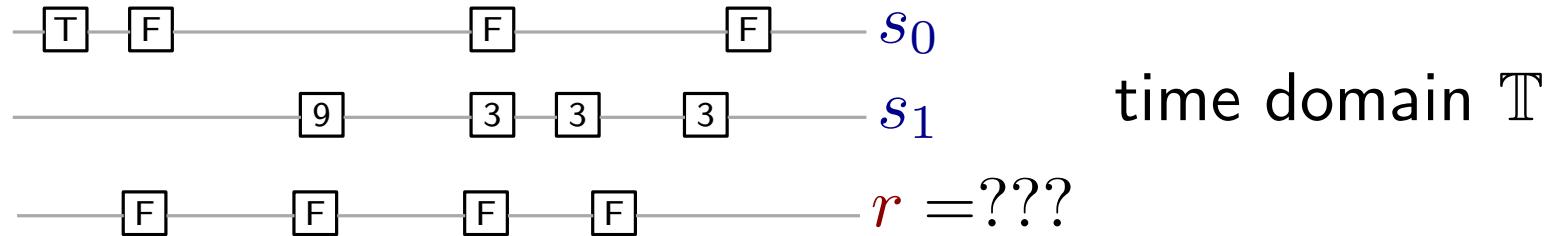
$$e : \mathbb{T} \rightarrow D$$

Striver

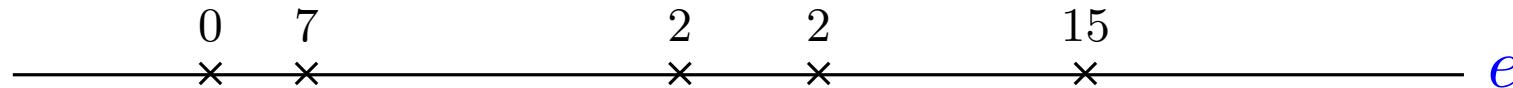
Striver extends SRV to real-time **asynchronous** streams

F. Gorostiaga, C. Sánchez: *Striver: Stream Runtime Verification for Real-Time Signals and Event-Streams* RV'2018

F. Gorostiaga, C. Sánchez: *Stream Runtime Verification of Real-Time Event-Streams with the Striver Language* STTT 2021.



► Event streams



$e : \mathbb{T} \rightarrow D$ $dom(e)$ no bounded infinite subsets (non-zeno) 19/38


```
input int sale
```

```
input int arrival
```

ticks stock := whenever sale *or* arrival tick

define int stock := *last value of stock or 0 +*
the number of products arriving –
the number of products being sold

```
input int sale  
input int arrival
```

```
ticks stock := whenever sale or arrival tick
```

```
define int stock := last value of stock or 0 +  
the number of products arriving -  
the number of products being sold
```

ticks equation



```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

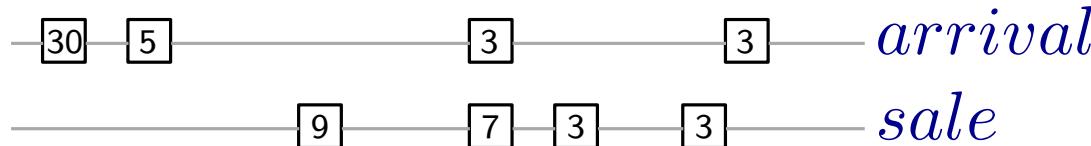
define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold

ticks equation
value equation

```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

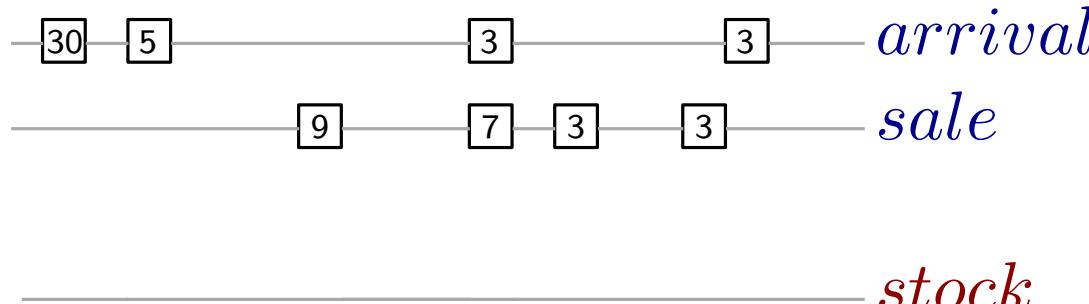
define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold



```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

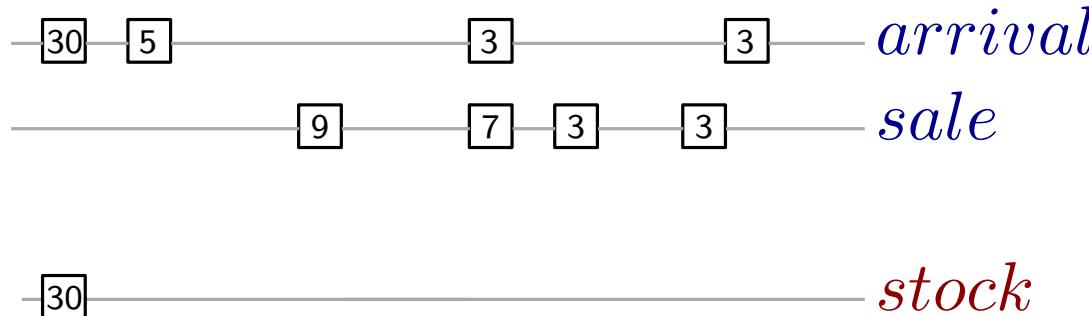
define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold



```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

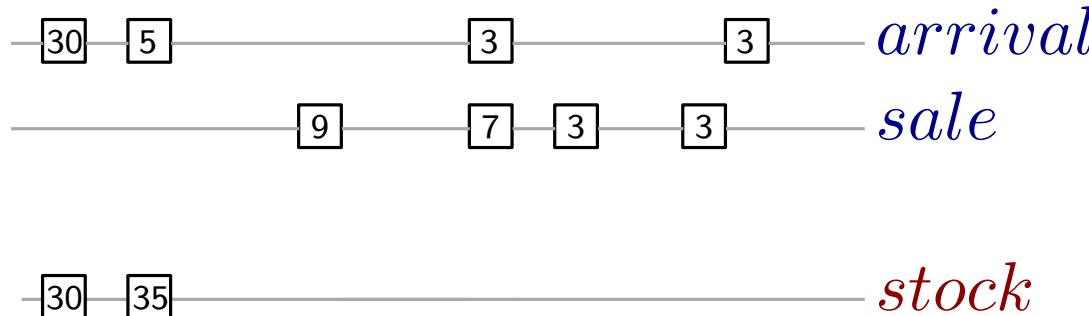
define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold



```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

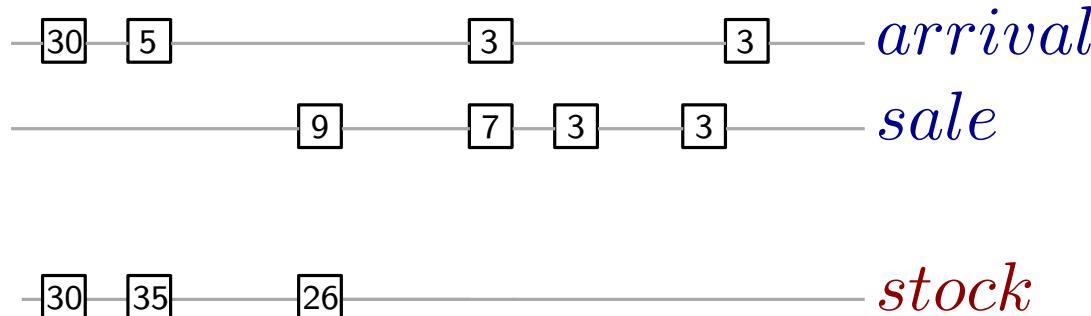
define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold



```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

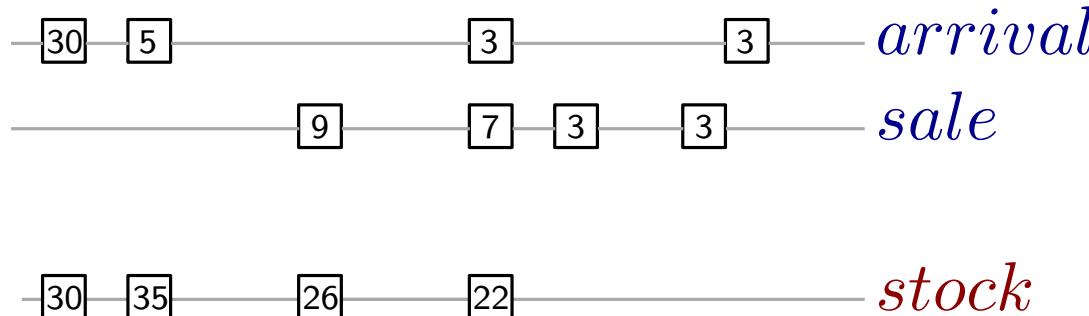
define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold



```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

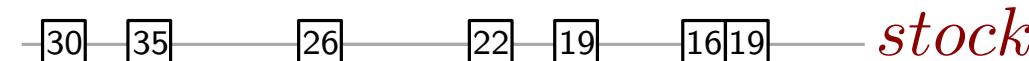
define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold



```
input int sale  
input int arrival
```

ticks stock := whenever sale *or* arrival tick

define int stock := *last value of stock or 0 +*
the number of products arriving -
the number of products being sold



ticks equation

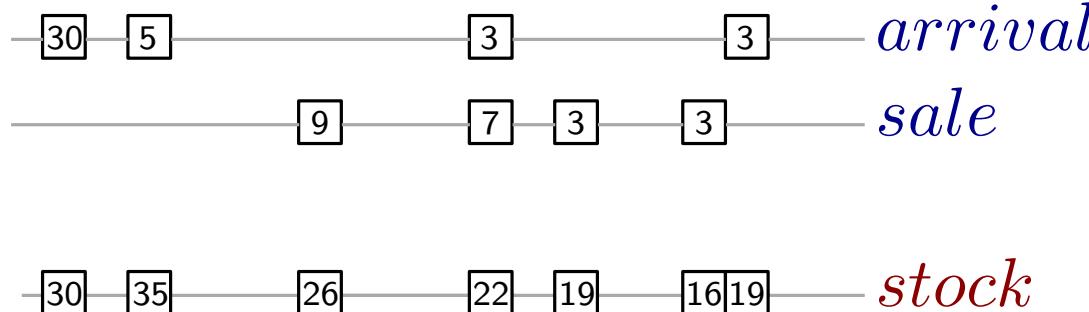
value equation

-

```
input int sale  
input int arrival
```

```
ticks stock := sale.ticks U arrival.ticks
```

define int stock := *last value of stock or 0* +
the number of products arriving -
the number of products being sold



```
input int sale
input int arrival
```

```
ticks stock := sale.ticks U arrival.ticks
```

```
define int stock := stock(<t,0)
```

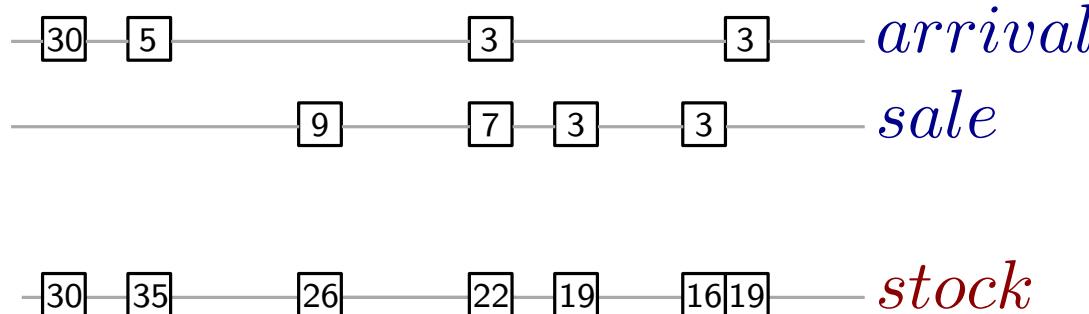
the number of products arriving
the number of products being sold

ticks equation

value equation

+

-

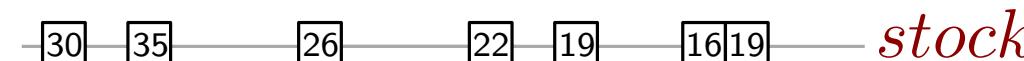


```
input int sale
input int arrival
```

```
ticks stock := sale.ticks U arrival.ticks
```

```
define int stock := stock(<t,0)
```

(if isticking(arrival) then arrival($\sim t$) else 0) -
the number of products being sold



ticks equation

value equation

+

```
input int sale
input int arrival
```

```
ticks stock := sale.ticks U arrival.ticks
```

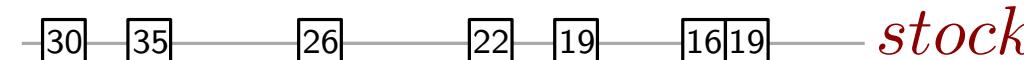
```
define int stock := stock(<t,0)
  (if isticking(arrival) then arrival(~t) else 0) +
  (if isticking(sale) then sale(~t) else 0)
```



arrival



sale



stock

ticks equation

value equation

+

Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

ticks $s := \{24\}$

define unit $s := ()$

s _____

Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

ticks $s := \{24\}$

define unit $s := ()$



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

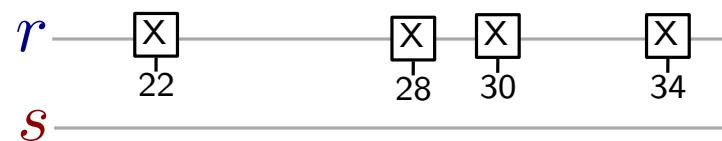
delay w

```
ticks s := {24}
```

```
define unit s := ()
```

```
ticks s := r.ticks
```

```
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

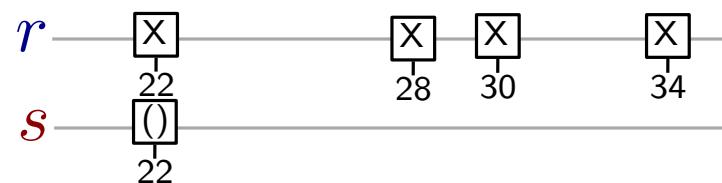
delay w

```
ticks s := {24}
```

```
define unit s := ()
```

```
ticks s := r.ticks
```

```
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

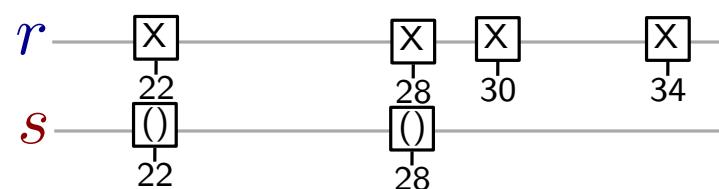
delay w

```
ticks s := {24}
```

```
define unit s := ()
```

```
ticks s := r.ticks
```

```
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

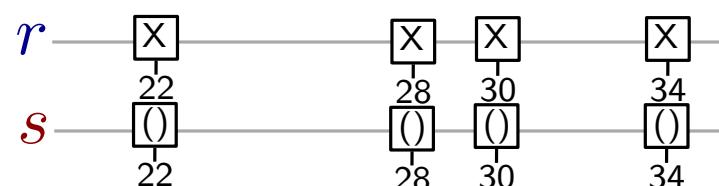
delay w

```
ticks s := {24}
```

```
define unit s := ()
```

```
ticks s := r.ticks
```

```
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

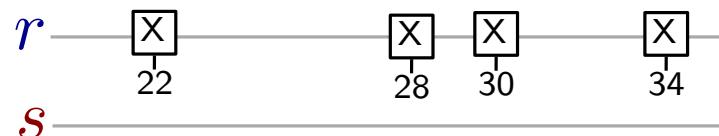
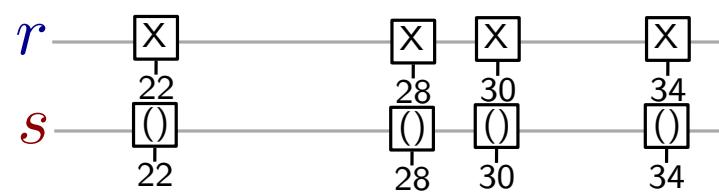
$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := {24}
define unit s := ()
```

```
ticks s := r.ticks
define unit s := ()
```

```
ticks s := r.ticks U {24}
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

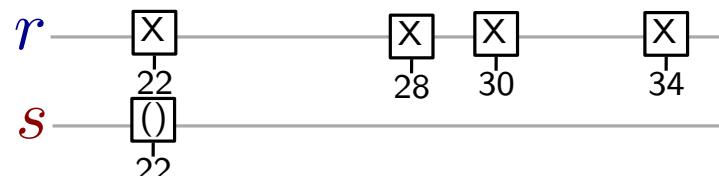
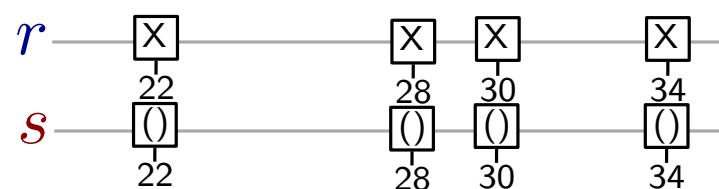
$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := {24}
define unit s := ()
```

```
ticks s := r.ticks
define unit s := ()
```

```
ticks s := r.ticks U {24}
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

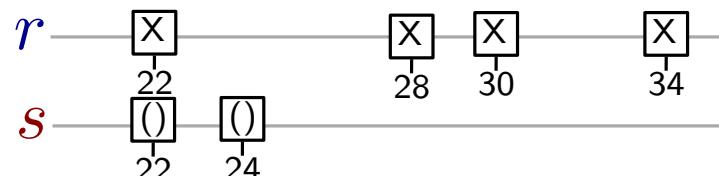
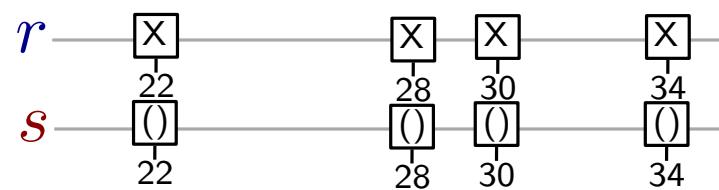
$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := {24}
define unit s := ()
```

```
ticks s := r.ticks
define unit s := ()
```

```
ticks s := r.ticks U {24}
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

$\{c\}$

$v.\text{ticks}$

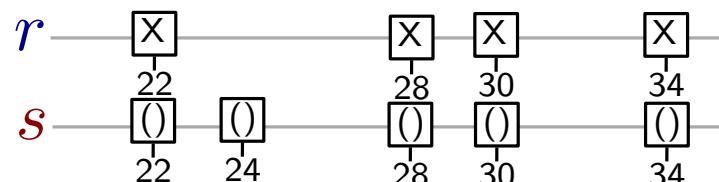
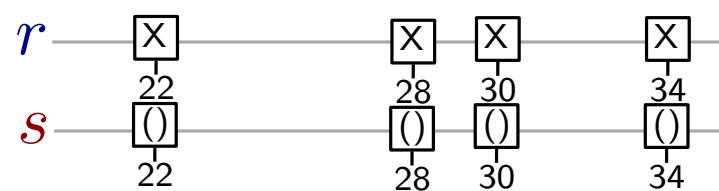
$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := {24}
define unit s := ()
```

```
ticks s := r.ticks
define unit s := ()
```

```
ticks s := r.ticks U {24}
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

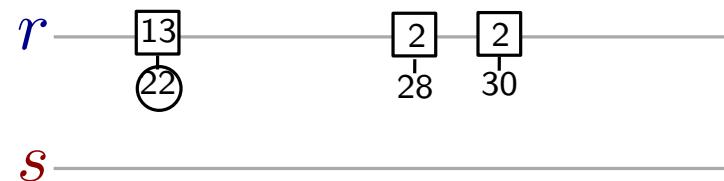
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

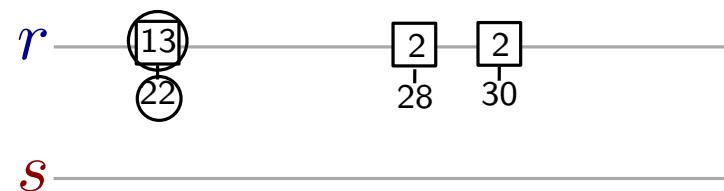
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

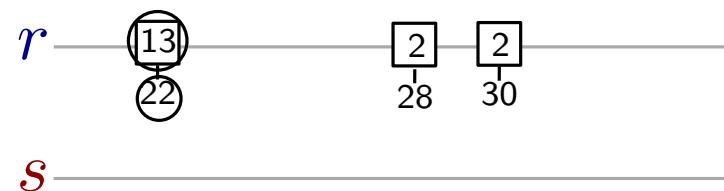
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

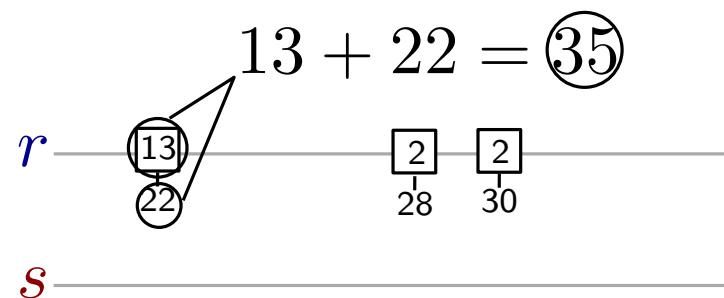
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

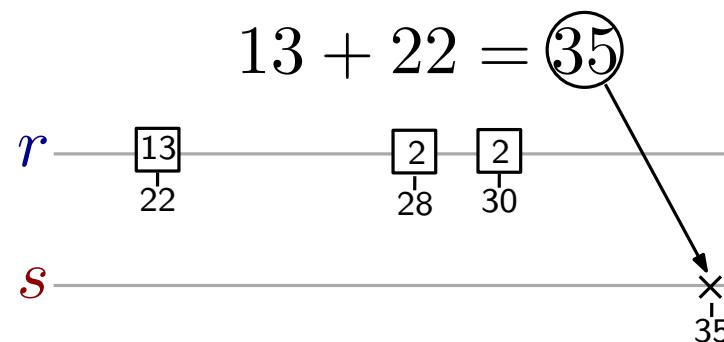
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

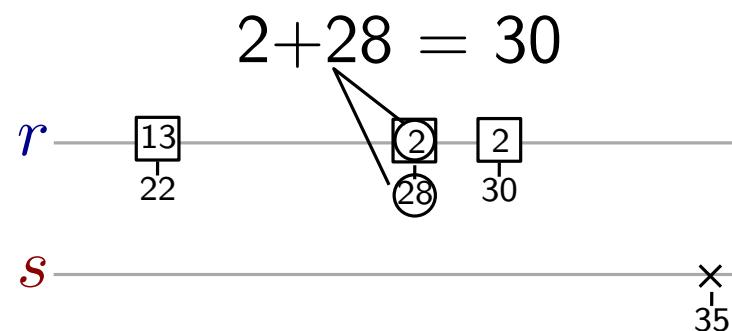
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

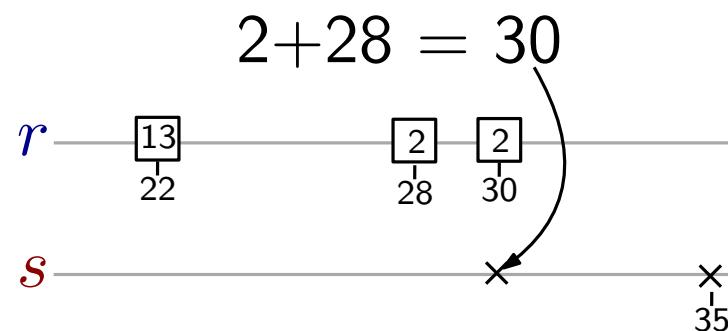
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

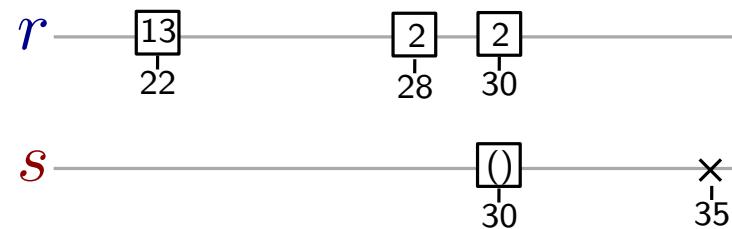
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Syntax of Striver

ticks

Each output stream s is associated with:

- ▶ a **ticking equation** instants at which s may have an event
- ▶ a **value equation**

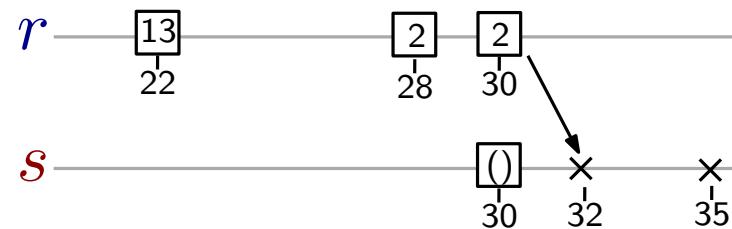
$\{c\}$

$v.\text{ticks}$

$\alpha_1 \cup \alpha_2$

delay w

```
ticks s := delay r  
define unit s := ()
```



Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$\mathbf{t} \quad x <^{\sim} t \quad x << t$$

Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$\mathbf{t} \quad x <^{\sim} t \quad x << t$$

Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

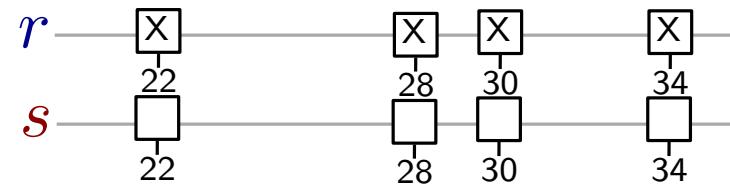
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$\mathbf{t} \quad x <^{\sim} t \quad x << t$$

```
ticks s := r.ticks  
define unit s := ()
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

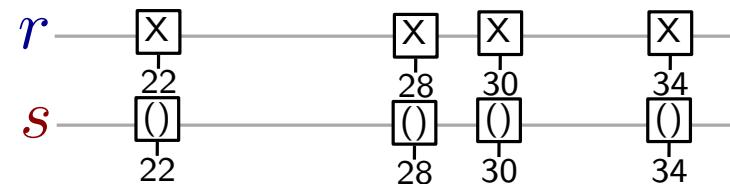
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$\mathbf{t} \quad x <^{\sim} t \quad x << t$$

```
ticks s := r.ticks  
define unit s := ()
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

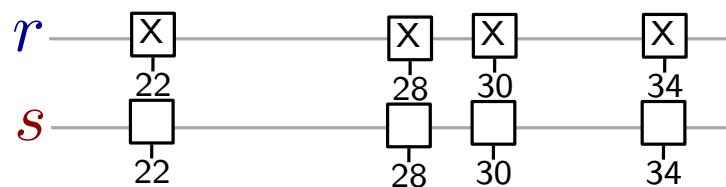
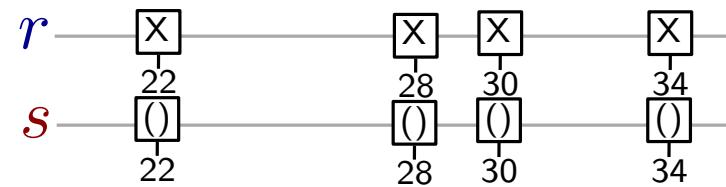
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$\mathbf{t} \quad x <^{\sim} t \quad x << t$$

```
ticks s := r.ticks  
define unit s := ()
```

```
ticks s := r.ticks  
define Time s := t
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

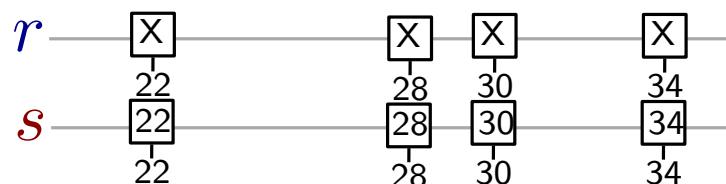
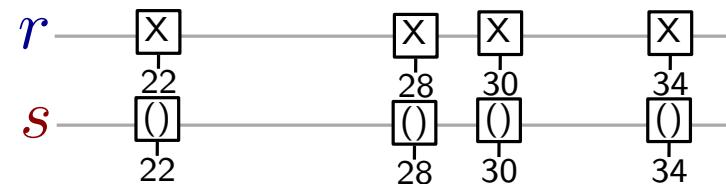
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$\mathbf{t} \quad x <^{\sim} t \quad x << t$$

```
ticks s := r.ticks  
define unit s := ()
```

```
ticks s := r.ticks  
define Time s := t
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

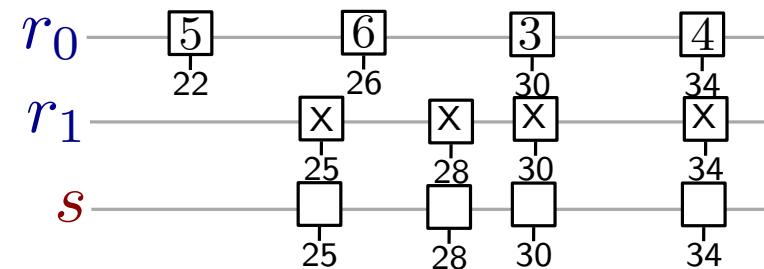
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

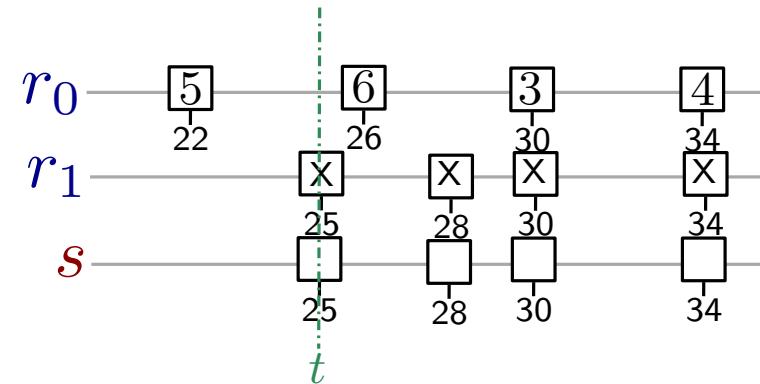
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

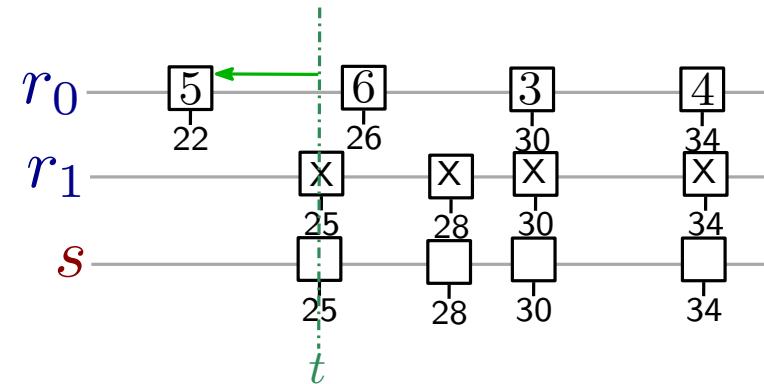
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

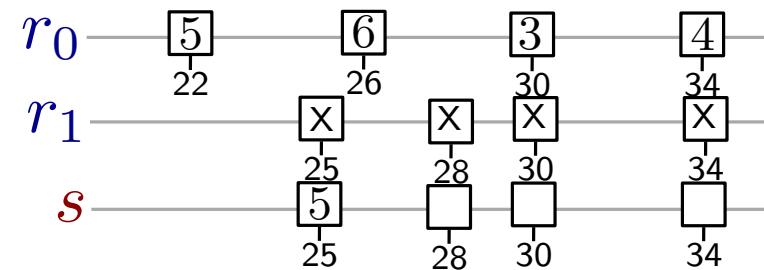
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

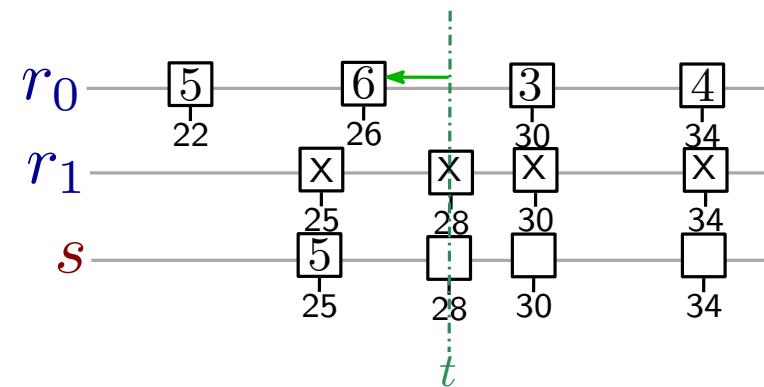
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

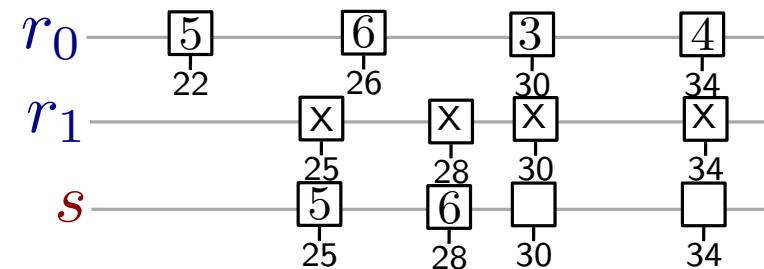
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

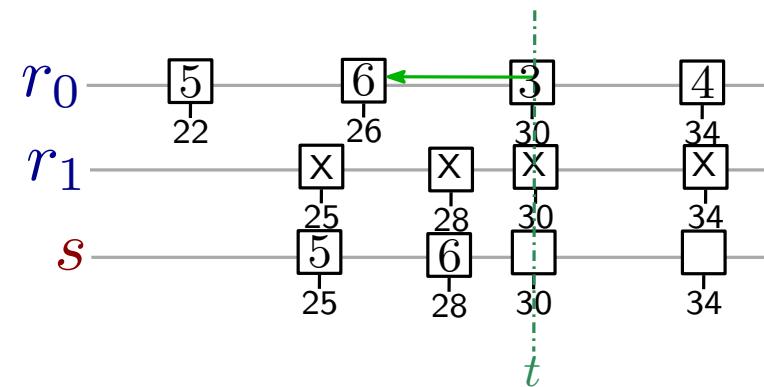
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

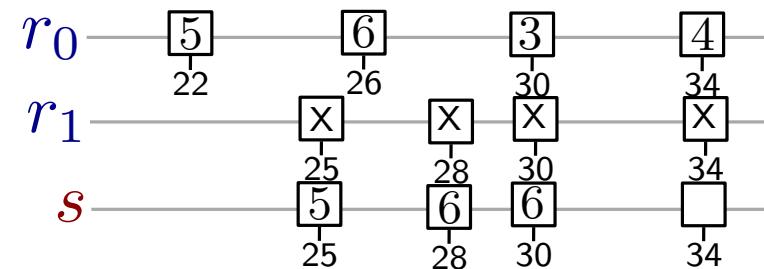
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

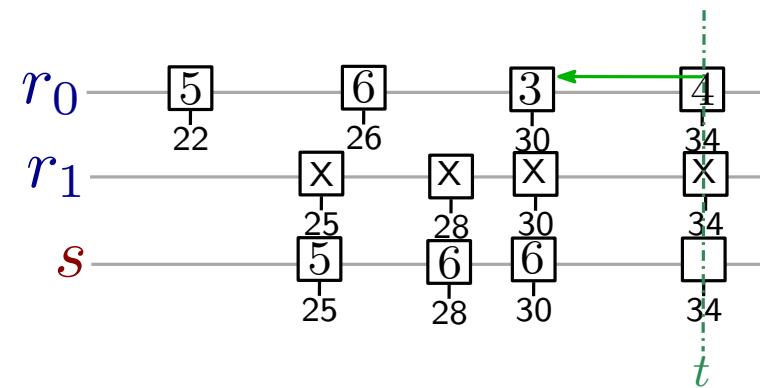
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

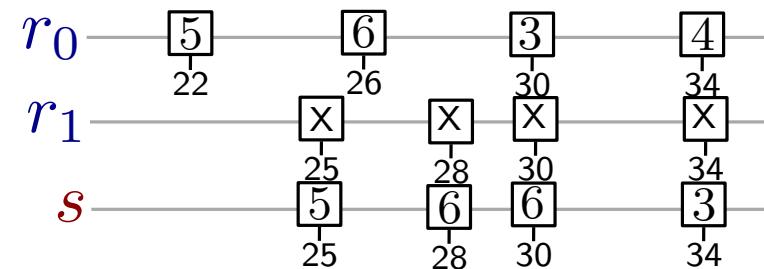
- ▶ *Value Expressions:*

$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x \sim t \quad x \ll t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

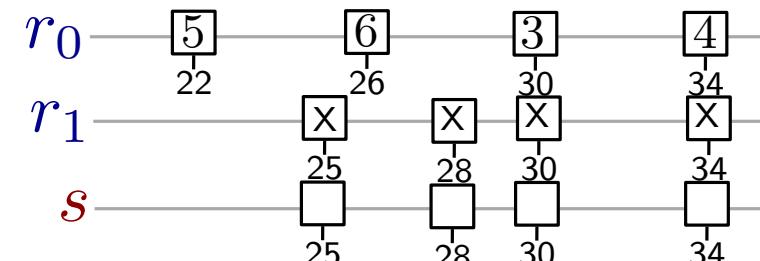
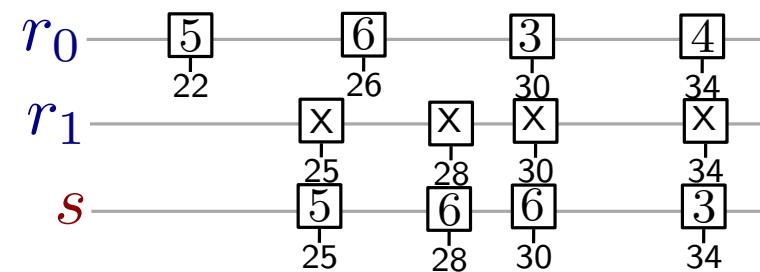
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks  
define int s := r0(r0 <^{\sim} t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

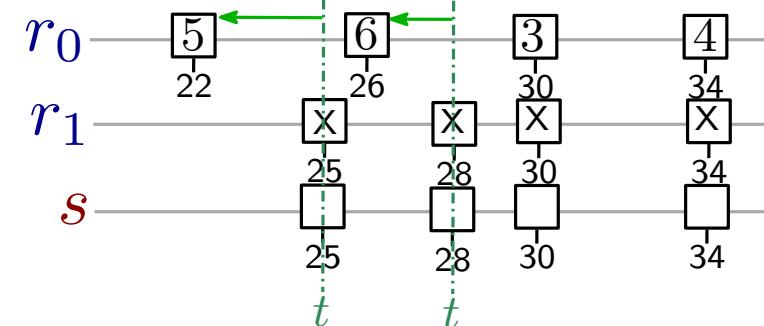
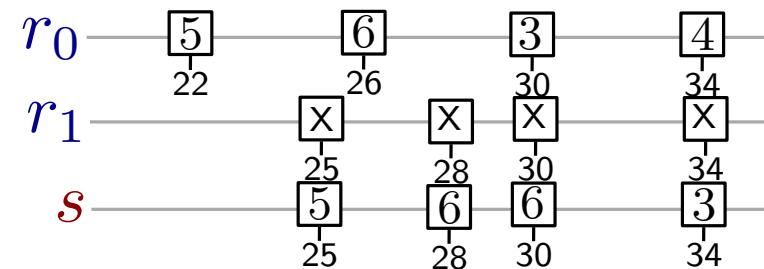
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks  
define int s := r0(r0 <^ t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

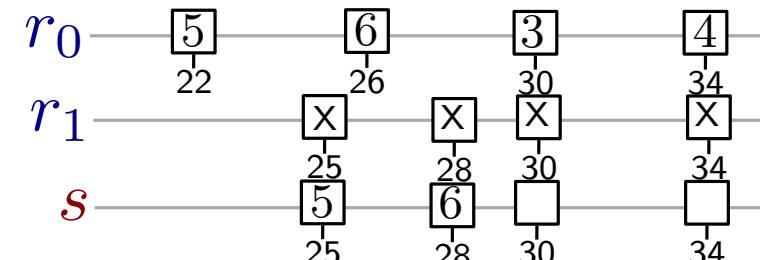
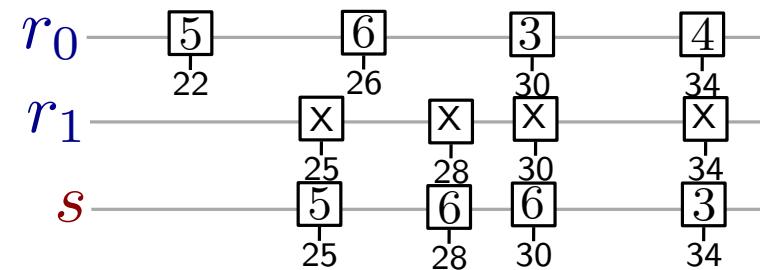
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks
define int s := r0(r0 <^ t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

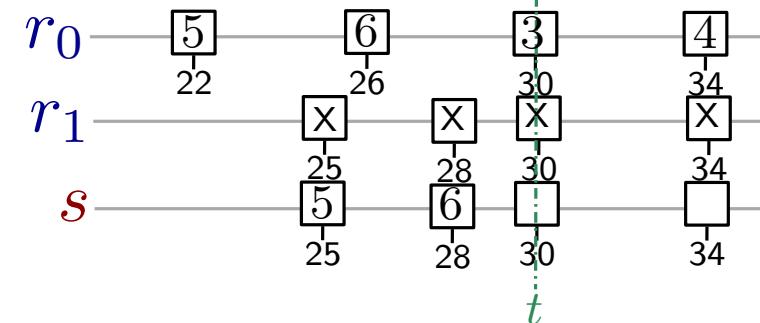
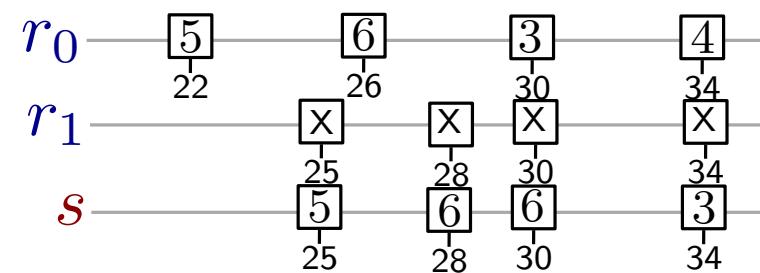
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks  
define int s := r0(r0 <^{\sim} t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

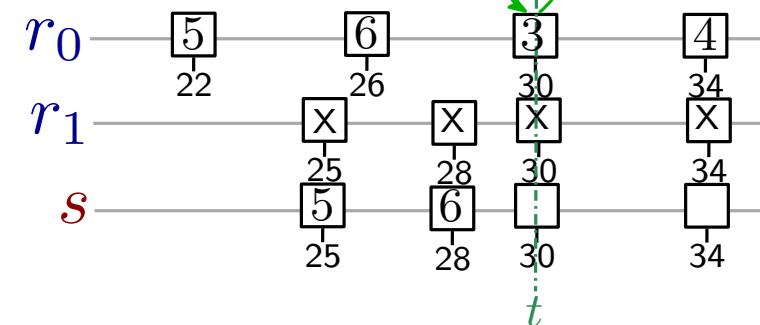
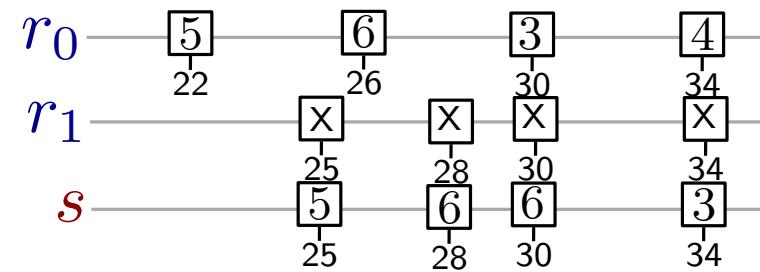
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks  
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks  
define int s := r0(r0 <^ t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

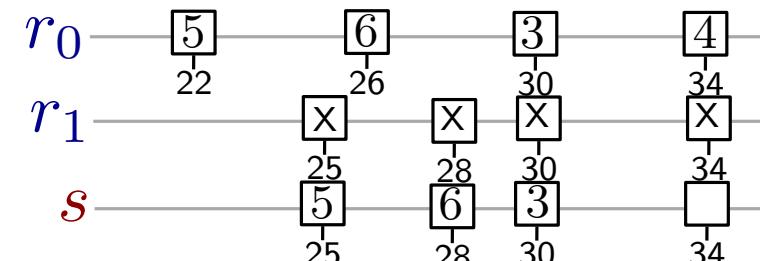
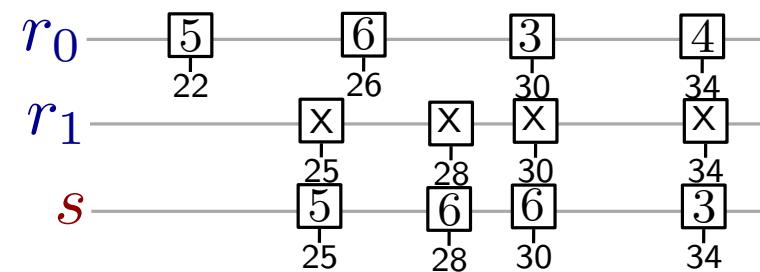
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks
define int s := r0(r0 <^ t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

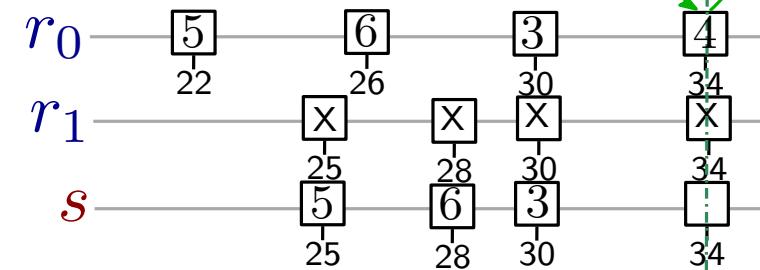
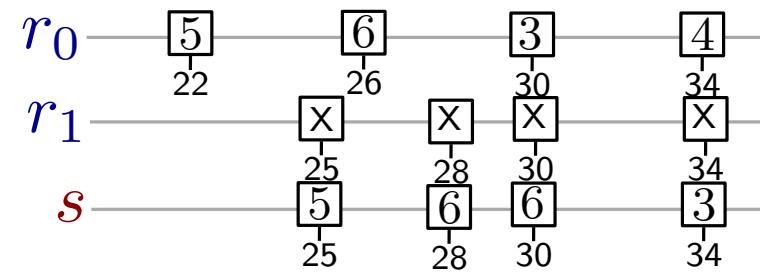
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks
define int s := r0(r0 <^{\sim} t)
```



Syntax of Striver

values

Each output stream s is associated with:

- ▶ a **ticking** equation
- ▶ a **value** equation

- ▶ *Value Expressions:*

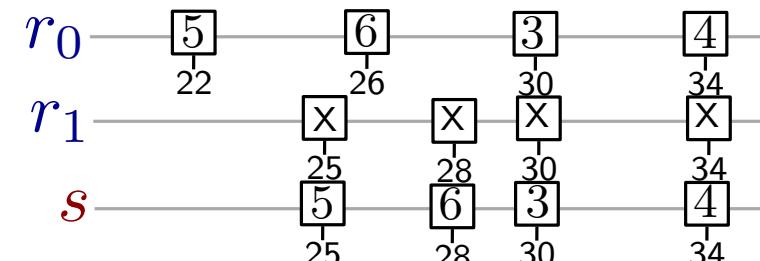
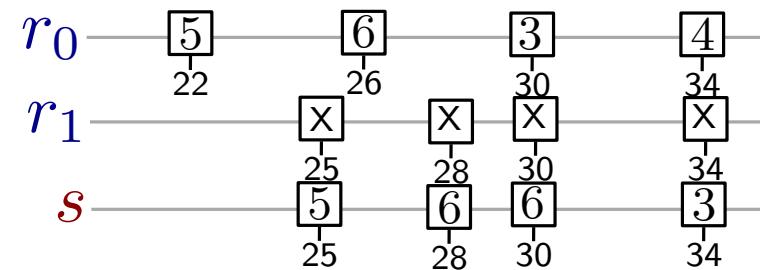
$$d \quad x(\tau) \quad \mathbf{f}(e_1, \dots, e_k) \quad \tau$$

- ▶ *Offset Expressions:*

$$t \quad x <^{\sim} t \quad x << t$$

```
ticks s := r1.ticks
define int s := r0(r0 << t)
```

```
ticks s := r1.ticks
define int s := r0(r0 <^{\sim} t)
```



Synchronous ↔ Asynchronous

Synchronous vs. asynchronous streams

Synchronous vs. asynchronous streams

Synchronous streams

Lola

Asynchronous streams

Striver

Synchronous vs. asynchronous streams

Synchronous streams

Lola

- ▶ Streams are sequences of values

p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p
q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q

Asynchronous streams

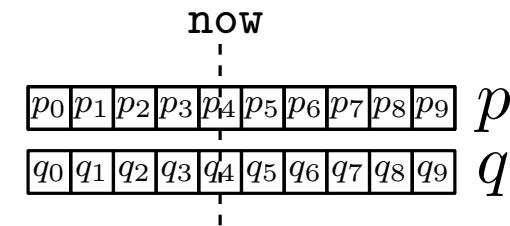
Striver

Synchronous vs. asynchronous streams

Synchronous streams

- ▶ Streams are sequences of values
- ▶ One position is active

Lola



Asynchronous streams

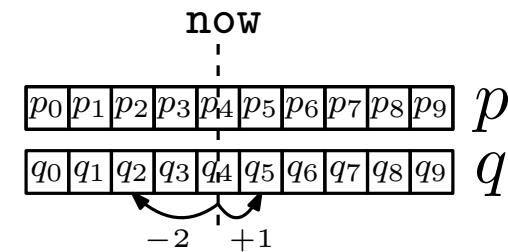
Striver

Synchronous vs. asynchronous streams

Synchronous streams

- ▶ Streams are sequences of values
- ▶ One position is active
- ▶ Offsets are relative to “now”

Lola



Asynchronous streams

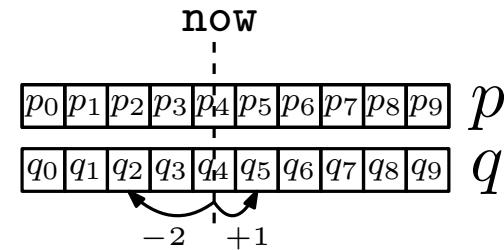
Striver

Synchronous vs. asynchronous streams

Synchronous streams

- ▶ Streams are sequences of values
- ▶ One position is active
- ▶ Offsets are relative to “now”

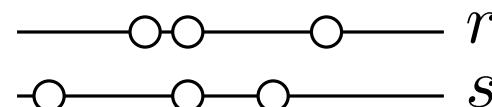
Lola



Asynchronous streams

- ▶ Streams are timestamped sequences of values

Striver

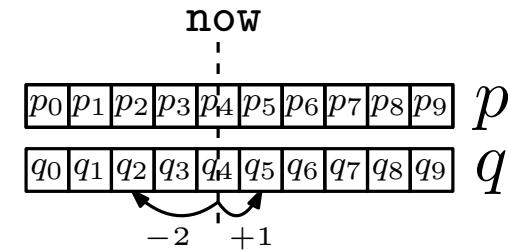


Synchronous vs. asynchronous streams

Synchronous streams

- ▶ Streams are sequences of values
- ▶ One position is active
- ▶ Offsets are relative to “now”

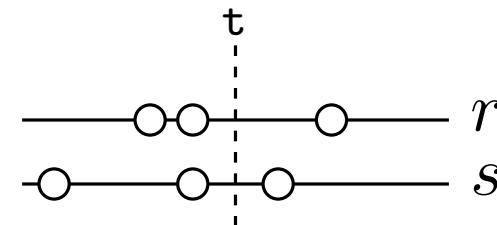
Lola



Asynchronous streams

- ▶ Streams are timestamped sequences of values
- ▶ There is a current time

Striver

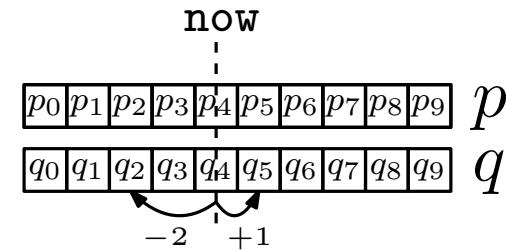


Synchronous vs. asynchronous streams

Synchronous streams

- ▶ Streams are sequences of values
- ▶ One position is active
- ▶ Offsets are relative to “now”

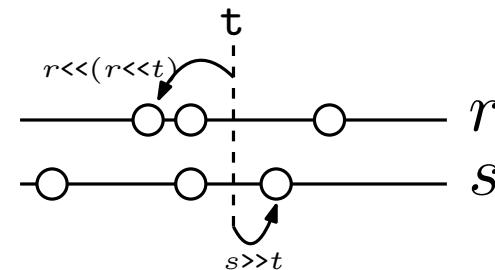
Lola



Asynchronous streams

- ▶ Streams are timestamped sequences of values
- ▶ There is a current time
- ▶ Refer to instants jumping around over events

Striver



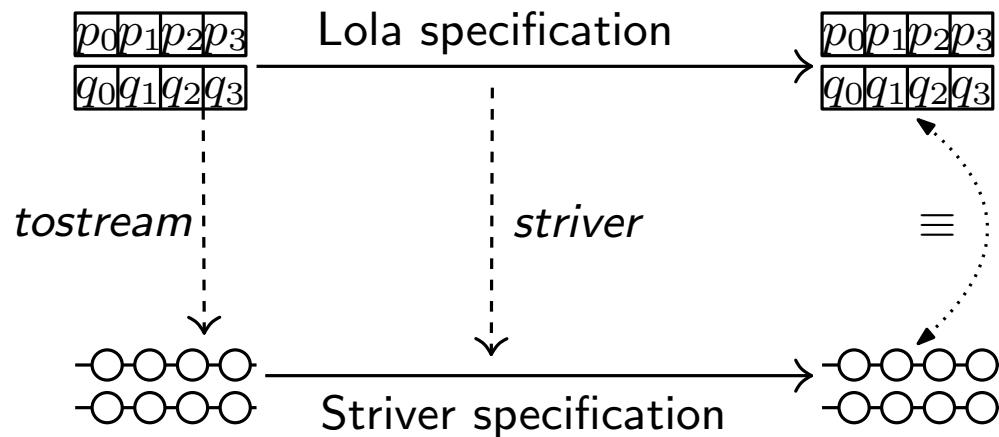
Synchronous vs. asynchronous streams

Synchronous → asynchronous

Asynchronous → synchronous

Synchronous vs. asynchronous streams

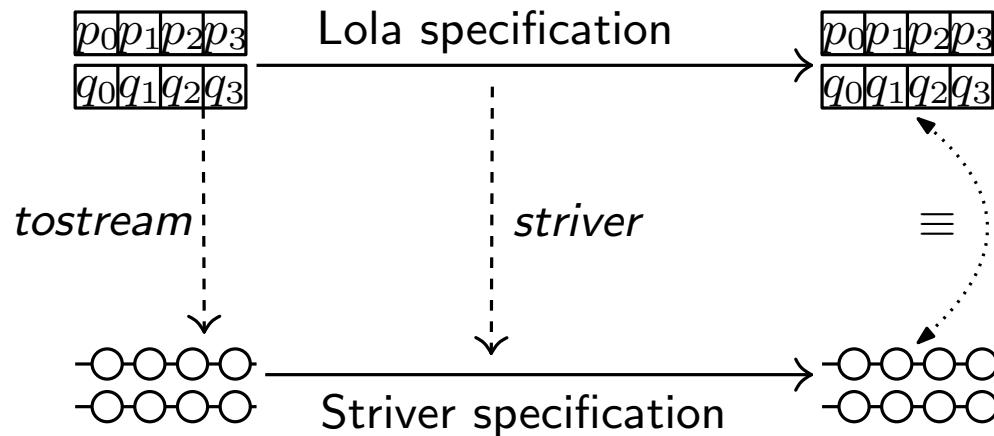
Synchronous → asynchronous



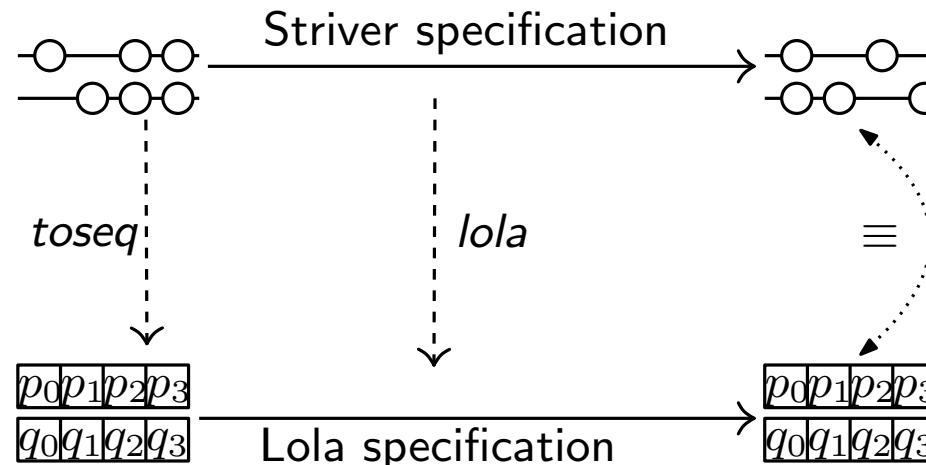
Asynchronous → synchronous

Synchronous vs. asynchronous streams

Synchronous → asynchronous



Asynchronous → synchronous



Lola to Striver: the problem

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```

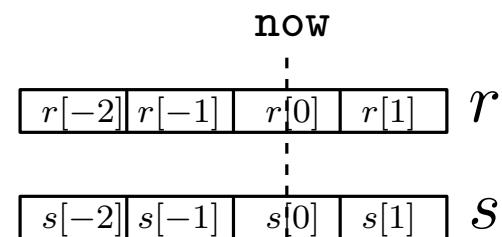
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



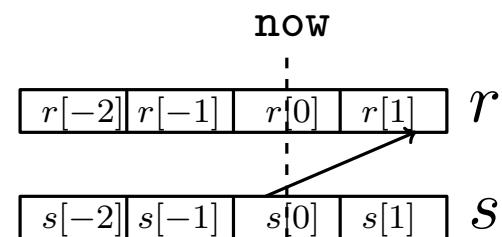
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



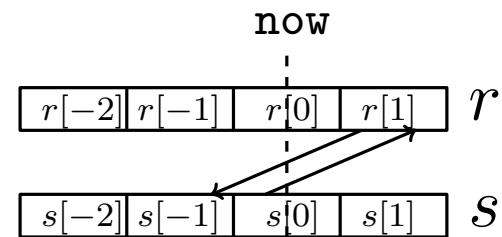
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



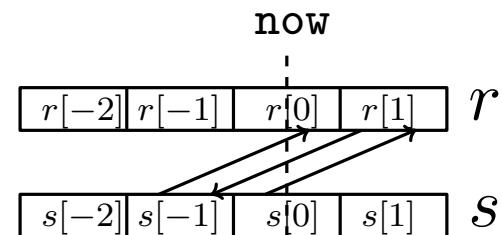
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



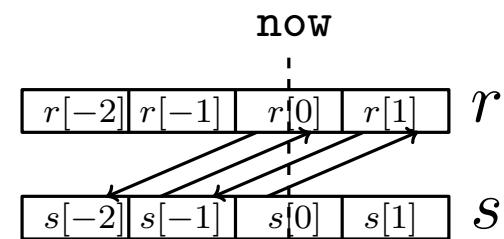
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



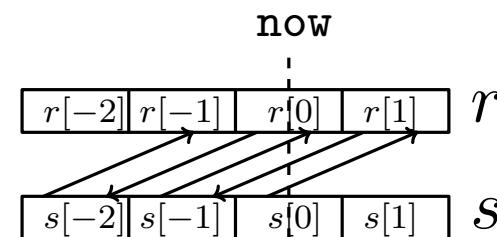
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

Lola to Striver: the problem

Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



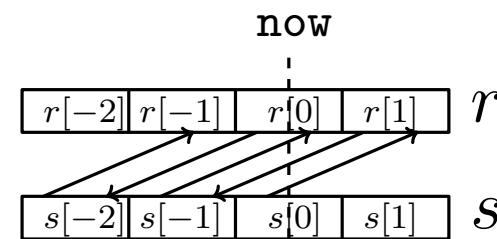
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

Lola to Striver: the problem

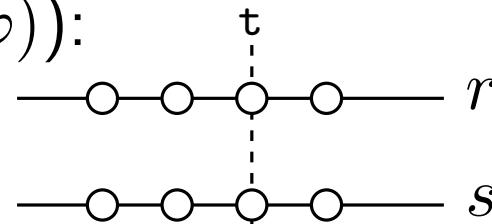
Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Translated Striver specification ($striver(\varphi)$):

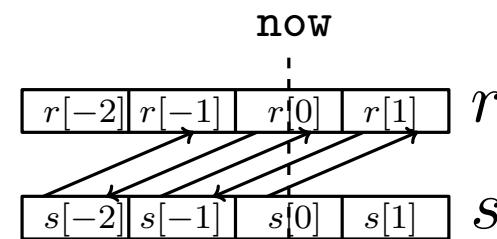
```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```



Lola to Striver: the problem

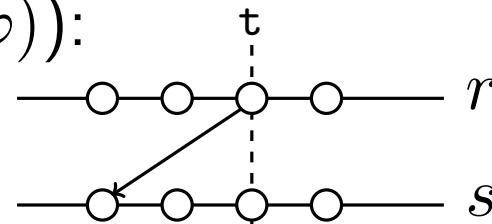
Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Translated Striver specification ($striver(\varphi)$):

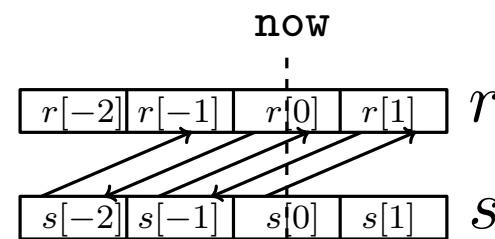
```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```



Lola to Striver: the problem

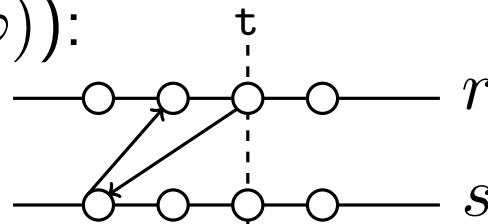
Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Translated Striver specification ($striver(\varphi)$):

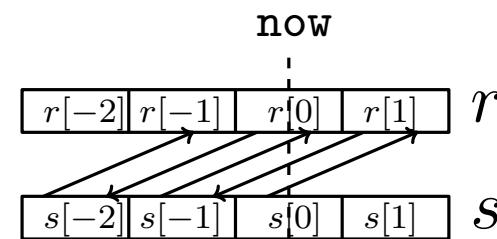
```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```



Lola to Striver: the problem

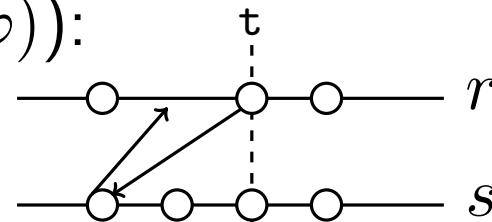
Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Translated Striver specification ($striver(\varphi)$):

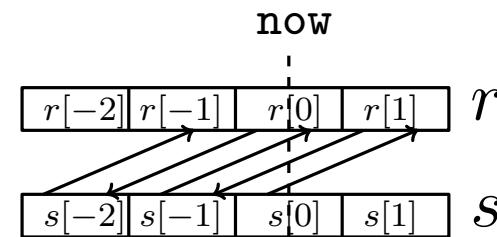
```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```



Lola to Striver: the problem

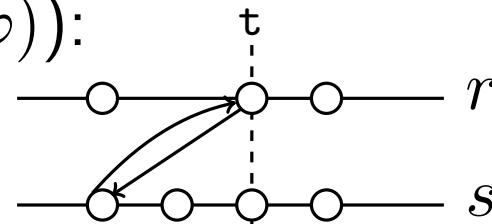
Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Translated Striver specification ($striver(\varphi)$):

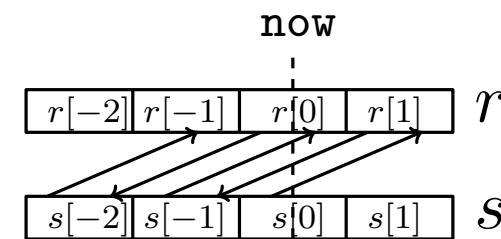
```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```



Lola to Striver: the problem

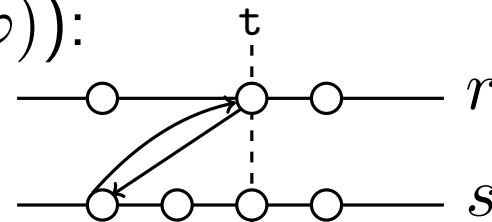
Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```

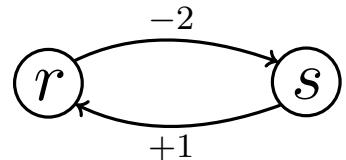


Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```



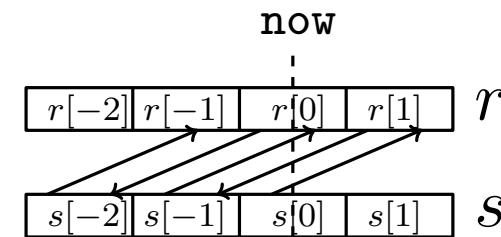
Dependency graph in Lola:



Lola to Striver: the problem

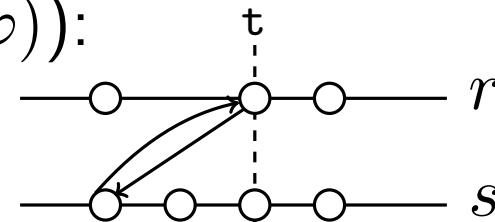
Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```

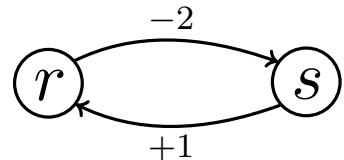


Translated Striver specification ($striver(\varphi)$):

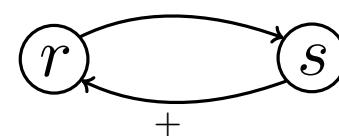
```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```



Dependency graph in Lola:



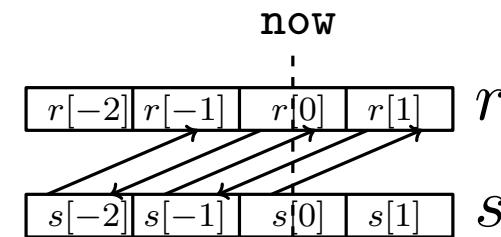
Dependency graph in Striver:



Lola to Striver: the problem

Lola specification (φ):

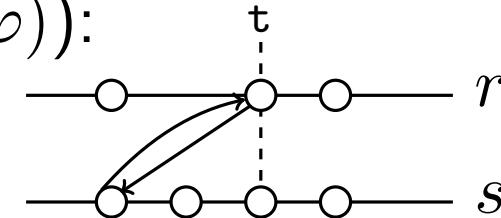
```
output int r := s[-2|7]
output int s := r[+1|0]
```



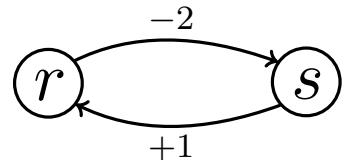
Translated Striver specification ($striver(\varphi)$):

```
define int r := s(<(s<<t),7)
define int s := r(>t,0)
```

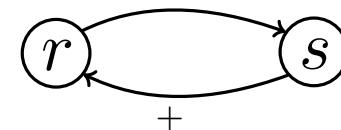
ILLEGAL



Dependency graph in Lola:



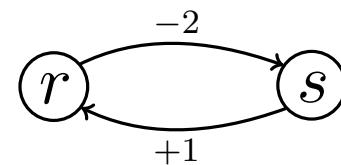
Dependency graph in Striver:



Lola to Striver: the solution

Original Lola specification (φ):

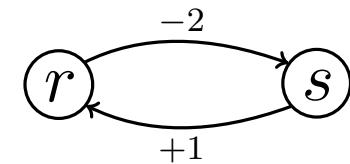
```
output int r := s[-2|7]
output int s := r[+1|0]
```



Lola to Striver: the solution

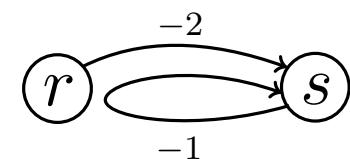
Original Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Equivalent Lola specification ($\text{expand}(\varphi)$):

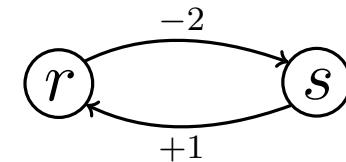
```
output int r := s[-2|7]
output int s := if true[1|false]
               then s[-1|7] else 0
```



Lola to Striver: the solution

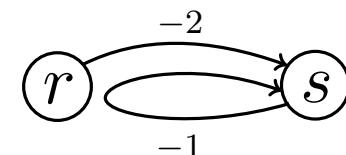
Original Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Equivalent Lola specification ($\text{expand}(\varphi)$):

```
output int r := s[-2|7]
output int s := if true[1|false]
    then s[-1|7] else 0
```



Translated Striver specification ($\text{striver}(\text{expand}(\varphi))$):

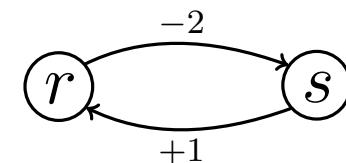
```
define int r := s(<(s<<t),7)
define int s := if true(>t,false)
    then s(<t,7) else 0
```



Lola to Striver: the solution

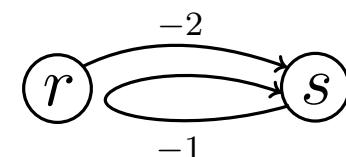
Original Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Equivalent Lola specification ($\text{expand}(\varphi)$):

```
output int r := s[-2|7]
output int s := if true[1|false]
    then s[-1|7] else 0
```



Translated Striver specification ($\text{striver}(\text{expand}(\varphi))$):

```
define int r := s(<(s<<t),7)
define int s := if true(>t,false)
    then s(<t,7) else 0
```

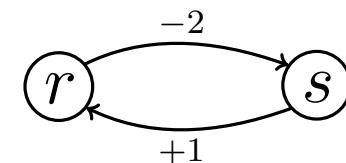


LEGAL

Lola to Striver: the solution

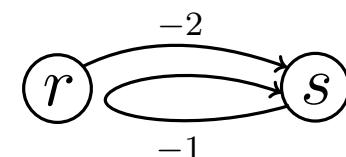
Original Lola specification (φ):

```
output int r := s[-2|7]
output int s := r[+1|0]
```



Equivalent Lola specification ($expand(\varphi)$):

```
output int r := s[-2|7]
output int s := if true[1|false]
               then s[-1|7] else 0
```



Translated Striver specification ($striver(expand(\varphi))$):

```
define int r := s(<(s<<t),7)
define int s := if true(>t,false)
               then s(<t,7) else 0
```

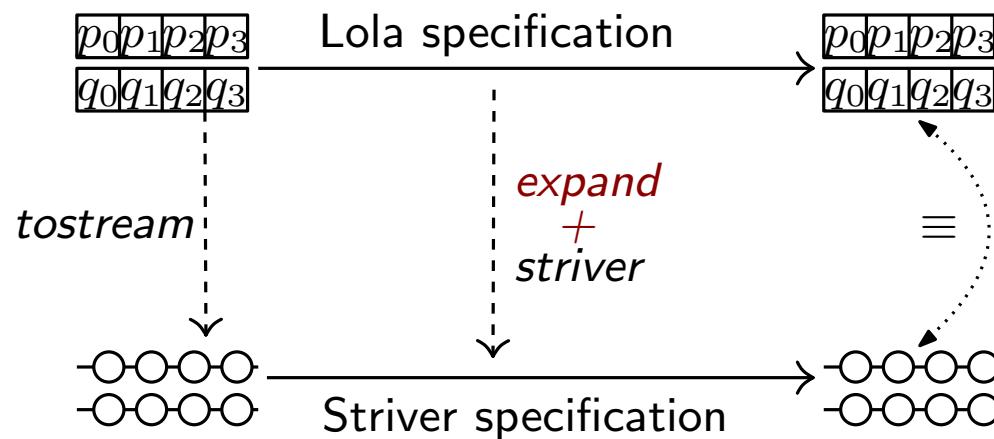


LEGAL

Lemma $\varphi \equiv expand(\varphi)$

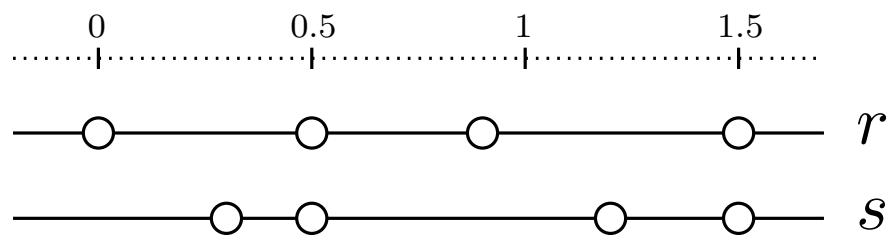
Lola to Striver

Theorem

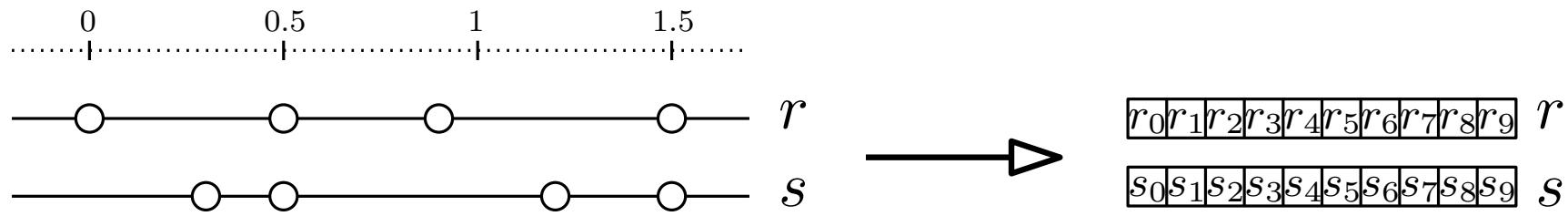


Striver to Lola

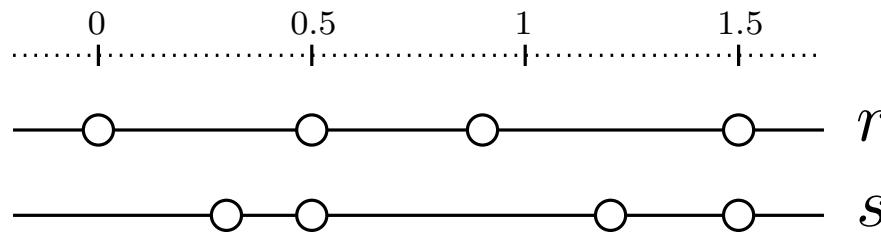
Striver to Lola



Striver to Lola

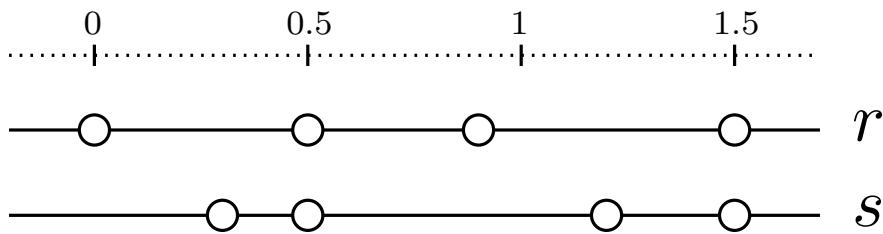


Striver to Lola

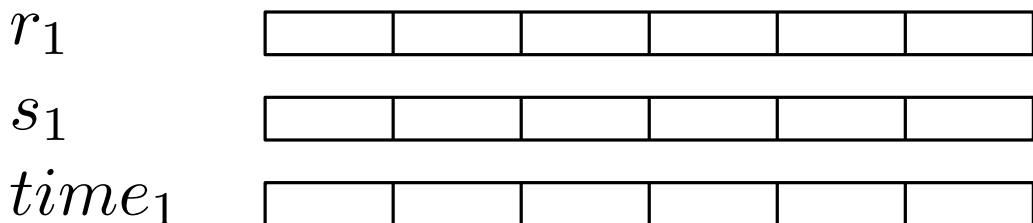


$r_0 | r_1 | r_2 | r_3 | r_4 | r_5 | r_6 | r_7 | r_8 | r_9 \quad r$
 $s_0 | s_1 | s_2 | s_3 | s_4 | s_5 | s_6 | s_7 | s_8 | s_9 \quad s$
 $t_0 | t_1 | t_2 | t_3 | t_4 | t_5 | t_6 | t_7 | t_8 | t_9 \quad time$

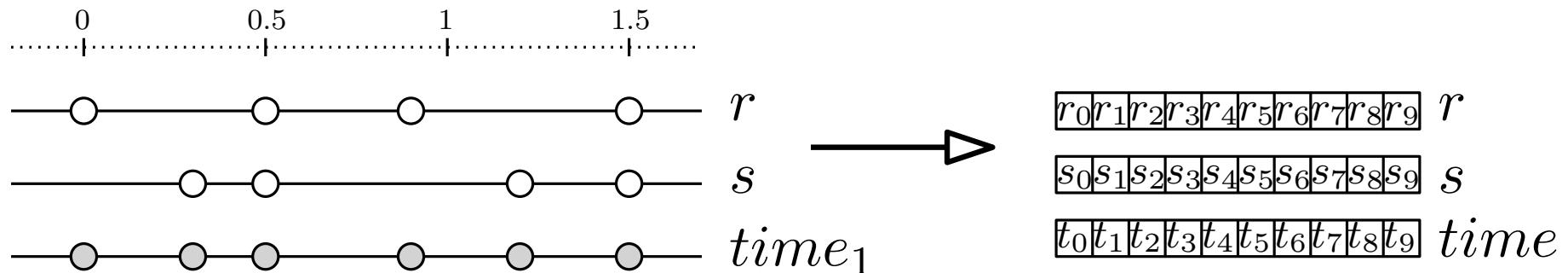
Striver to Lola



$r_0 \boxed{r_1} \boxed{r_2} \boxed{r_3} \boxed{r_4} \boxed{r_5} \boxed{r_6} \boxed{r_7} \boxed{r_8} \boxed{r_9} \quad r$
 $s_0 \boxed{s_1} \boxed{s_2} \boxed{s_3} \boxed{s_4} \boxed{s_5} \boxed{s_6} \boxed{s_7} \boxed{s_8} \boxed{s_9} \quad s$
 $t_0 \boxed{t_1} \boxed{t_2} \boxed{t_3} \boxed{t_4} \boxed{t_5} \boxed{t_6} \boxed{t_7} \boxed{t_8} \boxed{t_9} \quad time$

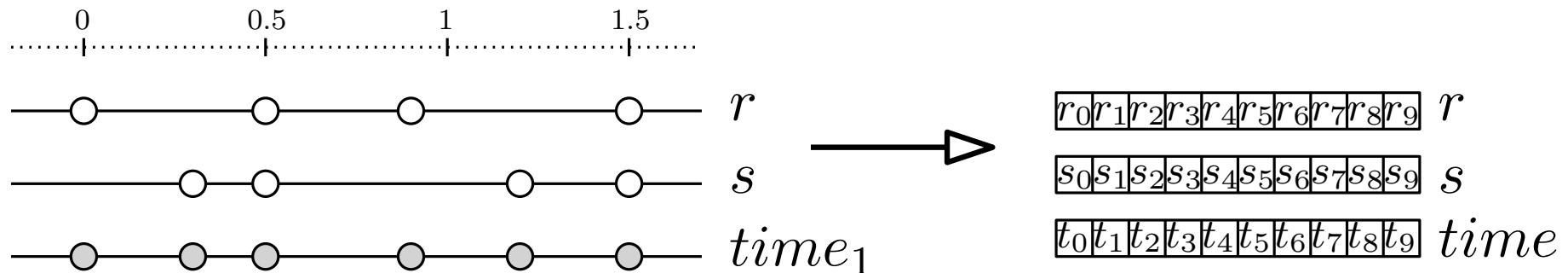


Striver to Lola



r_1	[] [] [] [] [] []
s_1	[] [] [] [] [] []
$time_1$	0 0.3 0.5 0.9 1.2 1.5

Striver to Lola

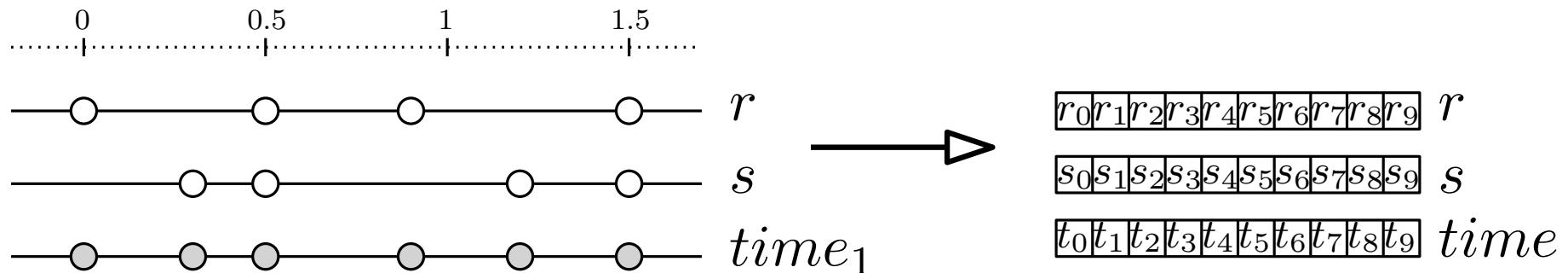


r_1	$r(0)$		$r(0.5)$	$r(0.9)$		$r(1.5)$
-------	--------	--	----------	----------	--	----------

s_1		$s(0.3)$	$s(0.5)$		$s(1.2)$	$s(1.5)$
-------	--	----------	----------	--	----------	----------

$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

Striver to Lola

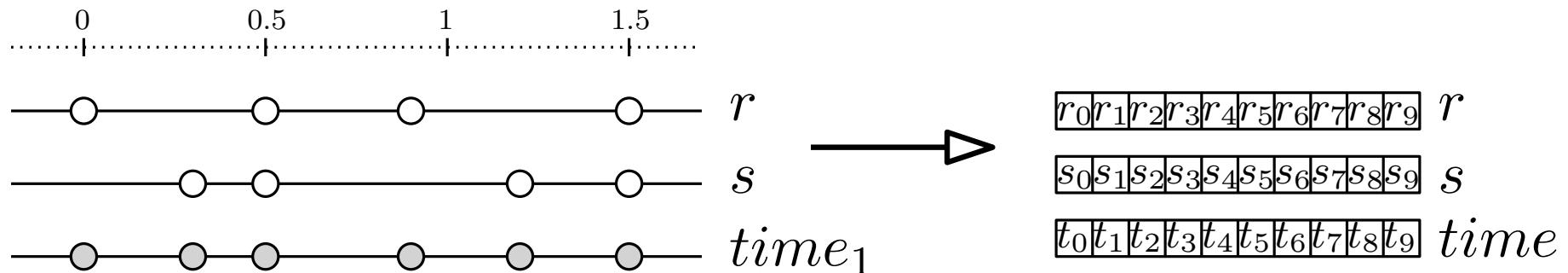


r_1 $\boxed{r(0)} \boxed{\text{notick}} \boxed{r(0.5)} \boxed{r(0.9)} \boxed{\text{notick}} \boxed{r(1.5)}$

s_1 $\boxed{\text{notick}} \boxed{s(0.3)} \boxed{s(0.5)} \boxed{\text{notick}} \boxed{s(1.2)} \boxed{s(1.5)}$

$time_1$ $\boxed{0} \boxed{0.3} \boxed{0.5} \boxed{0.9} \boxed{1.2} \boxed{1.5}$

Striver to Lola

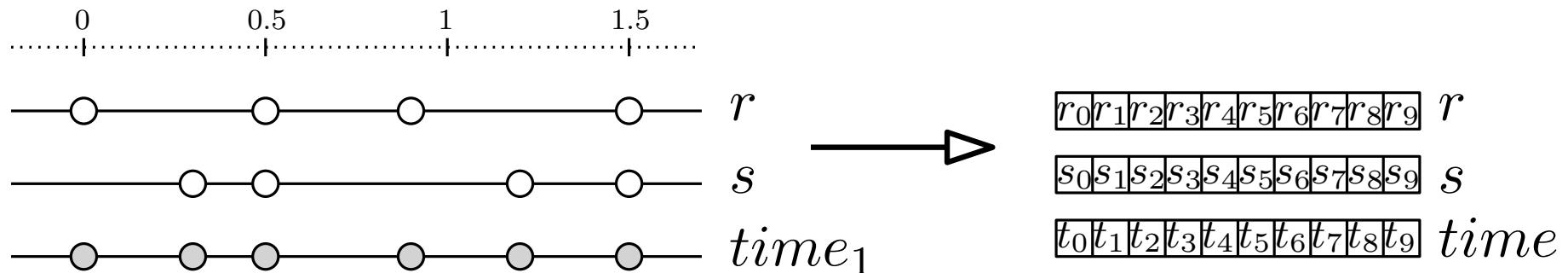


r_1	$r(0) \perp r(0.5) r(0.9) \perp r(1.5)$
-------	---

s_1	$\perp s(0.3) s(0.5) \perp s(1.2) s(1.5)$
-------	---

$time_1$	$0 0.3 0.5 0.9 1.2 1.5$
----------	-----------------------------------

Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

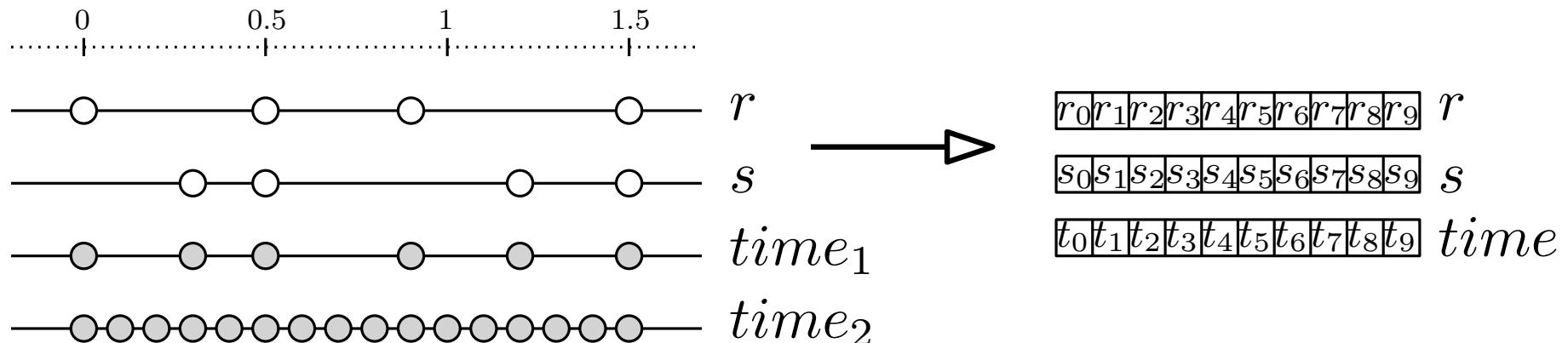
$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

r_2											
-------	--	--	--	--	--	--	--	--	--	--	--

s_2											
-------	--	--	--	--	--	--	--	--	--	--	--

$time_2$											
----------	--	--	--	--	--	--	--	--	--	--	--

Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

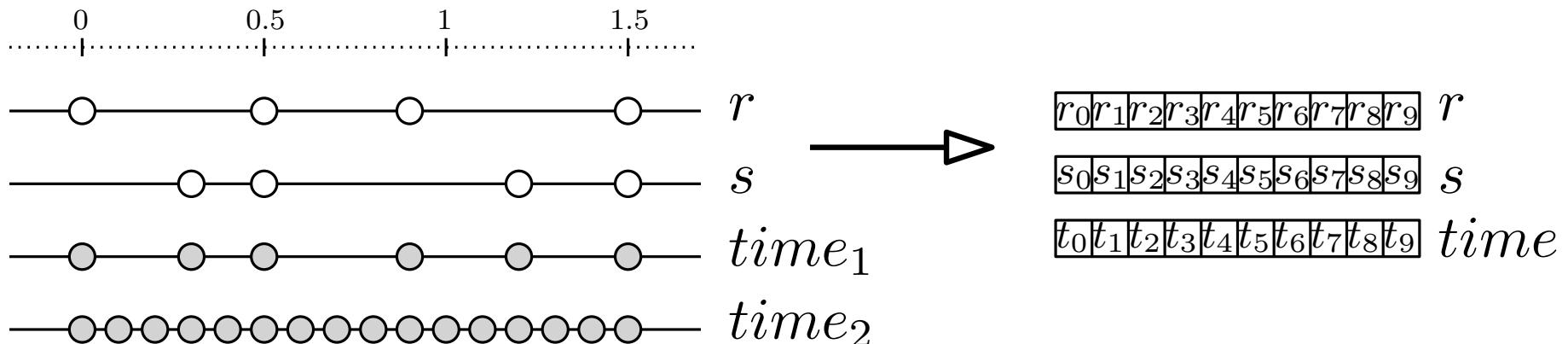
$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

r_2													
-------	--	--	--	--	--	--	--	--	--	--	--	--	--

s_2													
-------	--	--	--	--	--	--	--	--	--	--	--	--	--

$time_2$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----

Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

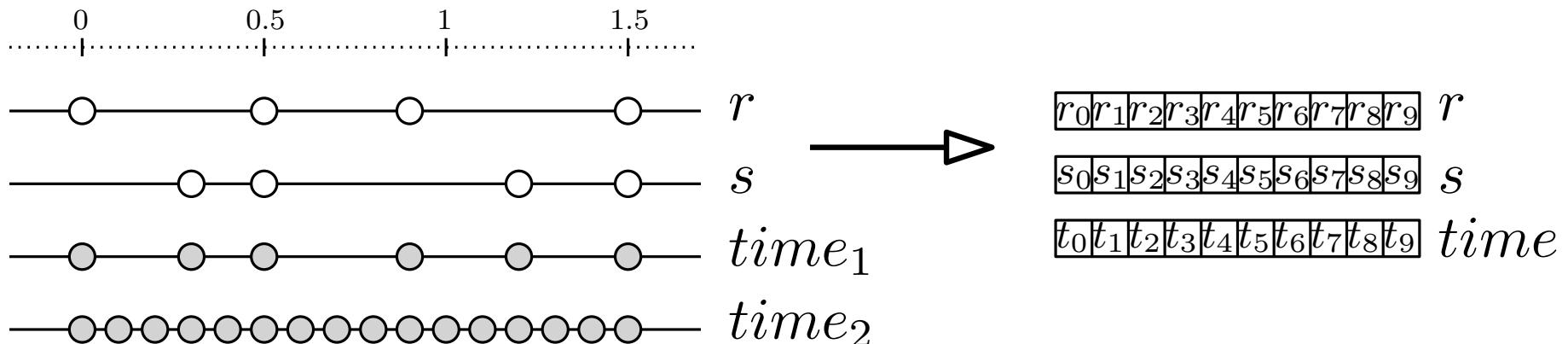
$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

r_2	$r(0)$	\perp	\perp	\perp	$r(0.5)$	\perp	\perp	$r(0.9)$	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	---------	---------	----------	---------	---------	----------	---------	---------	---------	----------

s_2	\perp	\perp	\perp	$s(0.3)$	$s(0.5)$	\perp	\perp	\perp	\perp	$s(1.2)$	\perp	\perp
-------	---------	---------	---------	----------	----------	---------	---------	---------	---------	----------	---------	---------

$time_2$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----

Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

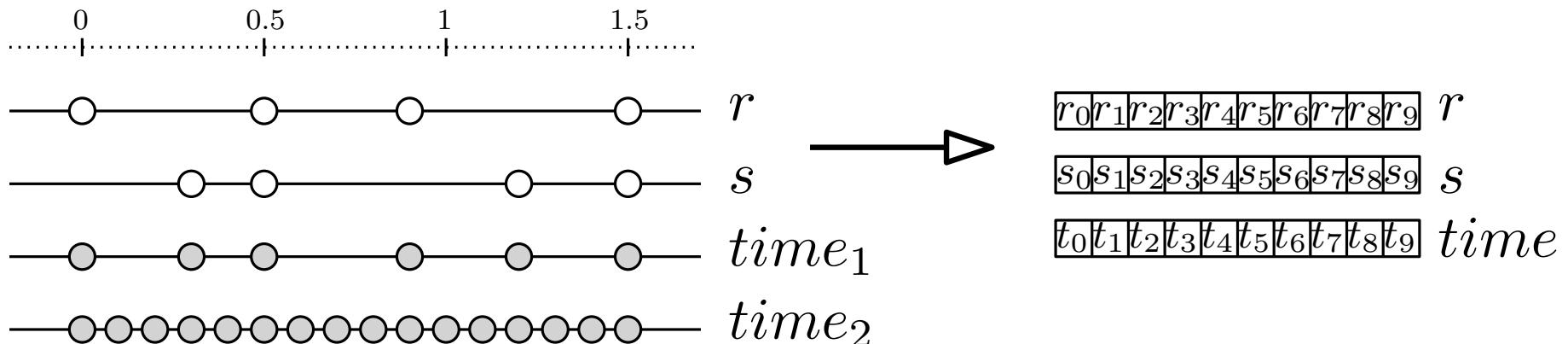
$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

r_2	$r(0)$	\perp	\perp	\perp	\perp	$r(0.5)$	\perp	\perp	\perp	$r(0.9)$	\perp	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	---------	---------	---------	----------	---------	---------	---------	----------	---------	---------	---------	---------	----------

s_2	\perp	\perp	\perp	$s(0.3)$	\perp	$s(0.5)$	\perp	\perp	\perp	\perp	\perp	$s(1.2)$	\perp	\perp	\perp
-------	---------	---------	---------	----------	---------	----------	---------	---------	---------	---------	---------	----------	---------	---------	---------

$time_2$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----

Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

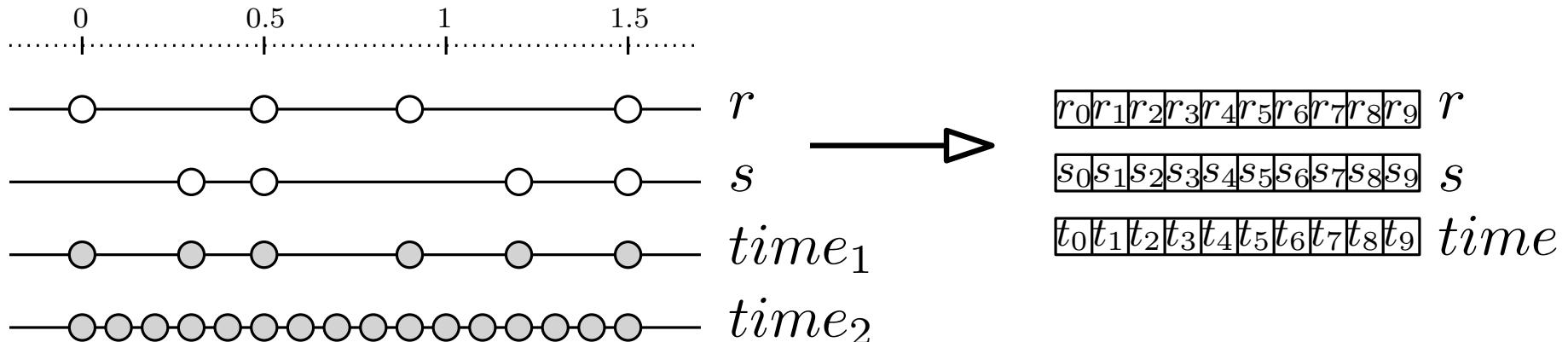
$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

r_2	$r(0)$	\perp	\perp	\perp	\perp	$r(0.5)$	\perp	\perp	\perp	$r(0.9)$	\perp	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	---------	---------	---------	----------	---------	---------	---------	----------	---------	---------	---------	---------	----------

s_2	\perp	\perp	\perp	$s(0.3)$	\perp	$s(0.5)$	\perp	\perp	\perp	\perp	\perp	\perp	$s(1.2)$	\perp	\perp
-------	---------	---------	---------	----------	---------	----------	---------	---------	---------	---------	---------	---------	----------	---------	---------

$time_2$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----

Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

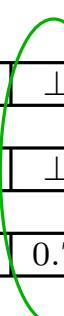
$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

void instants

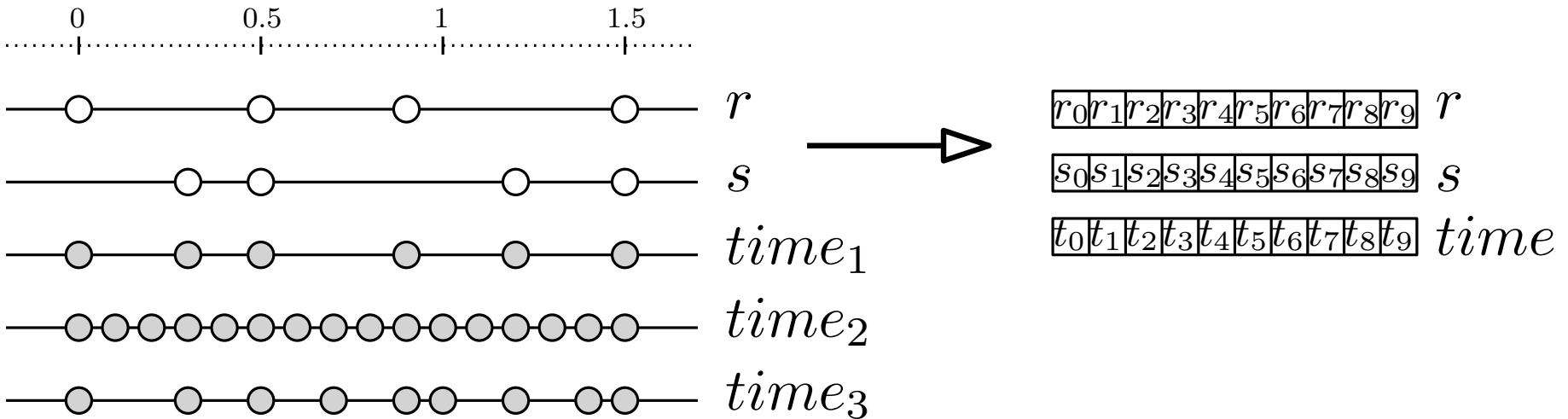
r_2	$r(0)$	\perp	\perp	\perp	\perp	$r(0.5)$	\perp	\perp	\perp	$r(0.9)$	\perp	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	---------	---------	---------	----------	---------	---------	---------	----------	---------	---------	---------	---------	----------

s_2	\perp	\perp	\perp	$s(0.3)$	\perp	$s(0.5)$	\perp	\perp	\perp	\perp	\perp	\perp	$s(1.2)$	\perp	\perp
-------	---------	---------	---------	----------	---------	----------	---------	---------	---------	---------	---------	---------	----------	---------	---------

$time_2$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----



Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

r_2	$r(0)$	\perp	\perp	\perp	\perp	$r(0.5)$	\perp	\perp	\perp	$r(0.9)$	\perp	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	---------	---------	---------	----------	---------	---------	---------	----------	---------	---------	---------	---------	----------

s_2	\perp	\perp	\perp	$s(0.3)$	\perp	$s(0.5)$	\perp	\perp	\perp	\perp	\perp	$s(1.2)$	\perp	\perp	\perp
-------	---------	---------	---------	----------	---------	----------	---------	---------	---------	---------	---------	----------	---------	---------	---------

$time_2$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----

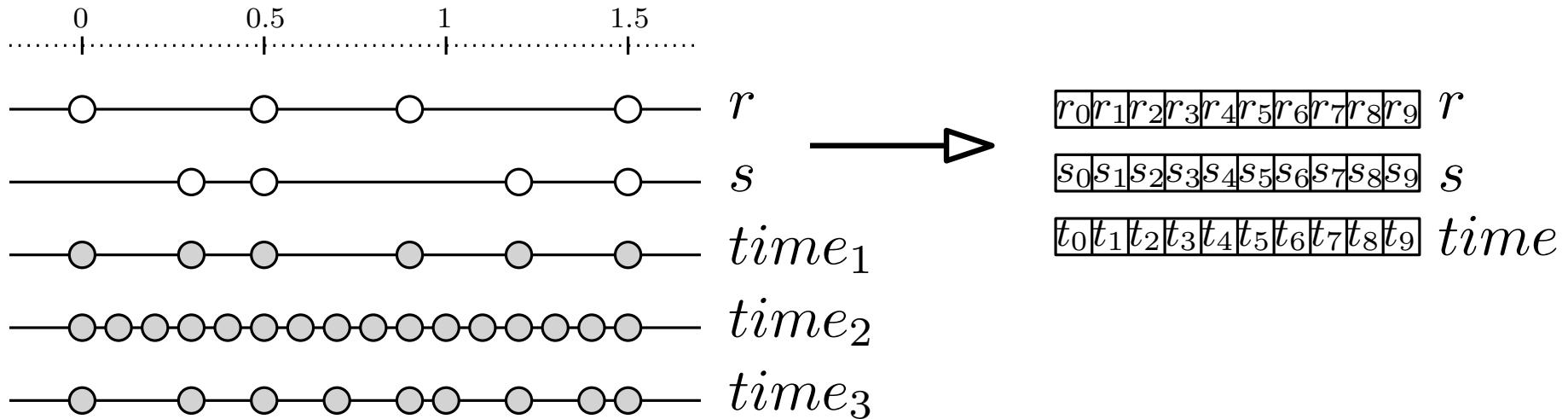
void instants

r_3	$r(0)$	\perp	$r(0.5)$	\perp	$r(0.9)$	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	----------	---------	----------	---------	---------	---------	----------

s_3	\perp	$s(0.3)$	$s(0.5)$	\perp	\perp	\perp	$s(1.2)$	\perp	\perp
-------	---------	----------	----------	---------	---------	---------	----------	---------	---------

$time_3$	0	0.3	0.5	0.7	0.9	1	1.2	1.4	1.5
----------	---	-----	-----	-----	-----	---	-----	-----	-----

Striver to Lola



r_1	$r(0) \perp r(0.5) r(0.9) \perp r(1.5)$
-------	---

s_1	$\perp s(0.3) s(0.5) \perp s(1.2) s(1.5)$
-------	---

$time_1$	$0 0.3 0.5 0.9 1.2 1.5$
----------	-----------------------------------

void instants

r_2	$r(0) \perp \perp \perp \perp r(0.5) \perp \perp \perp r(0.9) \perp \perp \perp \perp \perp r(1.5)$
-------	---

s_2	$\perp \perp \perp s(0.3) \perp s(0.5) \perp \perp \perp \perp \perp \perp \perp s(1.2) \perp \perp \perp$
-------	--

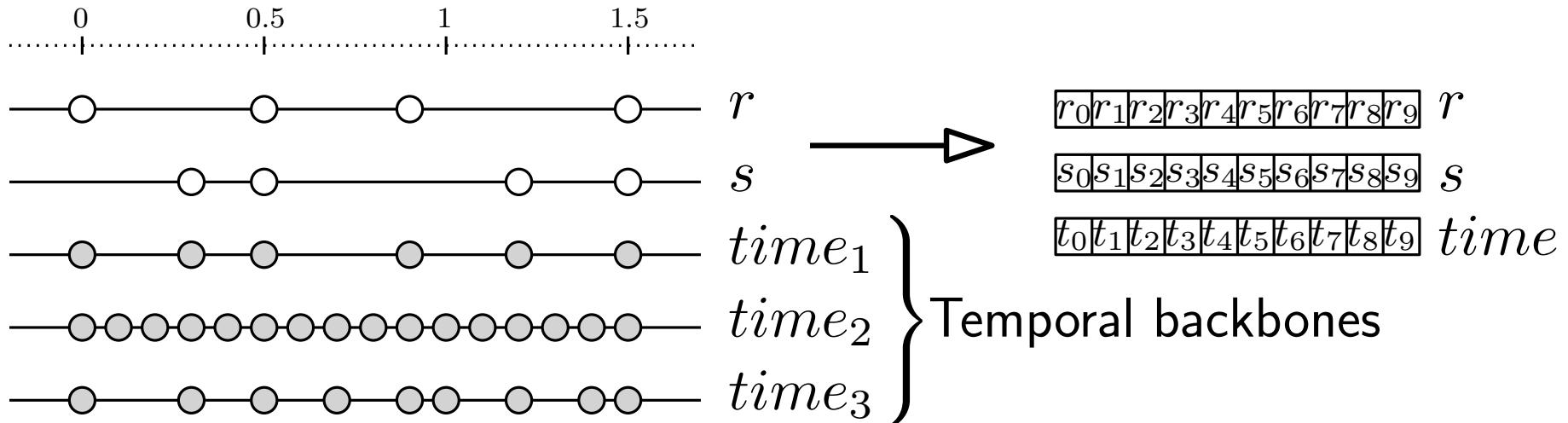
$time_2$	$0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5$
----------	---

r_3	$r(0) \perp r(0.5) \perp r(0.9) \perp \perp \perp \perp r(1.5)$
-------	---

s_3	$\perp s(0.3) s(0.5) \perp \perp \perp s(1.2) \perp \perp$
-------	--

$time_3$	$0 0.3 0.5 0.7 0.9 1 1.2 1.4 1.5$
----------	---

Striver to Lola



r_1	$r(0)$	\perp	$r(0.5)$	$r(0.9)$	\perp	$r(1.5)$
-------	--------	---------	----------	----------	---------	----------

s_1	\perp	$s(0.3)$	$s(0.5)$	\perp	$s(1.2)$	$s(1.5)$
-------	---------	----------	----------	---------	----------	----------

$time_1$	0	0.3	0.5	0.9	1.2	1.5
----------	---	-----	-----	-----	-----	-----

void instants

r_2	$r(0)$	\perp	\perp	\perp	\perp	$r(0.5)$	\perp	\perp	\perp	$r(0.9)$	\perp	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	---------	---------	---------	----------	---------	---------	---------	----------	---------	---------	---------	---------	----------

s_2	\perp	\perp	\perp	$s(0.3)$	\perp	$s(0.5)$	\perp	\perp	\perp	\perp	\perp	$s(1.2)$	\perp	\perp	\perp
-------	---------	---------	---------	----------	---------	----------	---------	---------	---------	---------	---------	----------	---------	---------	---------

$time_2$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
----------	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----

r_3	$r(0)$	\perp	$r(0.5)$	\perp	$r(0.9)$	\perp	\perp	\perp	$r(1.5)$
-------	--------	---------	----------	---------	----------	---------	---------	---------	----------

s_3	\perp	$s(0.3)$	$s(0.5)$	\perp	\perp	\perp	$s(1.2)$	\perp	\perp
-------	---------	----------	----------	---------	---------	---------	----------	---------	---------

$time_3$	0	0.3	0.5	0.7	0.9	1	1.2	1.4	1.5
----------	---	-----	-----	-----	-----	---	-----	-----	-----

Striver to Lola

Striver specification (φ):

```
ticks r      := s.ticks
define int r := s(s<t|7)
```

Striver to Lola

Striver specification (φ):

```
ticks r      := s.ticks
define int r := s(s<t|7)
```

Equivalent Lola specification ($lola(\varphi)$):

```
define int r := if s[now] == notick
                  then notick
                  else prev_s[0|7]
```

Striver to Lola

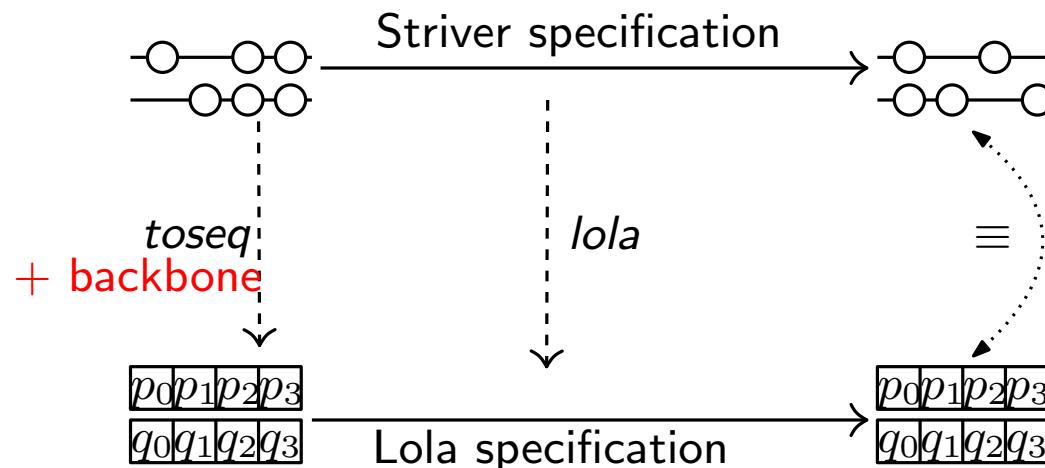
Striver specification (φ):

```
ticks r      := s.ticks
define int r := s(s<t|7)
```

Equivalent Lola specification ($lola(\varphi)$):

```
define int r := if s[now] == notick
                  then notick
                  else prev_s[0|7]
```

Theorem 2.



Empirical evaluation

Two Haskell programs: *HLola* and *HStriver* (available on GitHub)

Empirical evaluation

Two Haskell programs: *HLola* and *HStriver* (available on GitHub)
Dataset: Orange4Home

Empirical evaluation

Two Haskell programs: *HLola* and *HStriver* (available on GitHub)

Dataset: Orange4Home

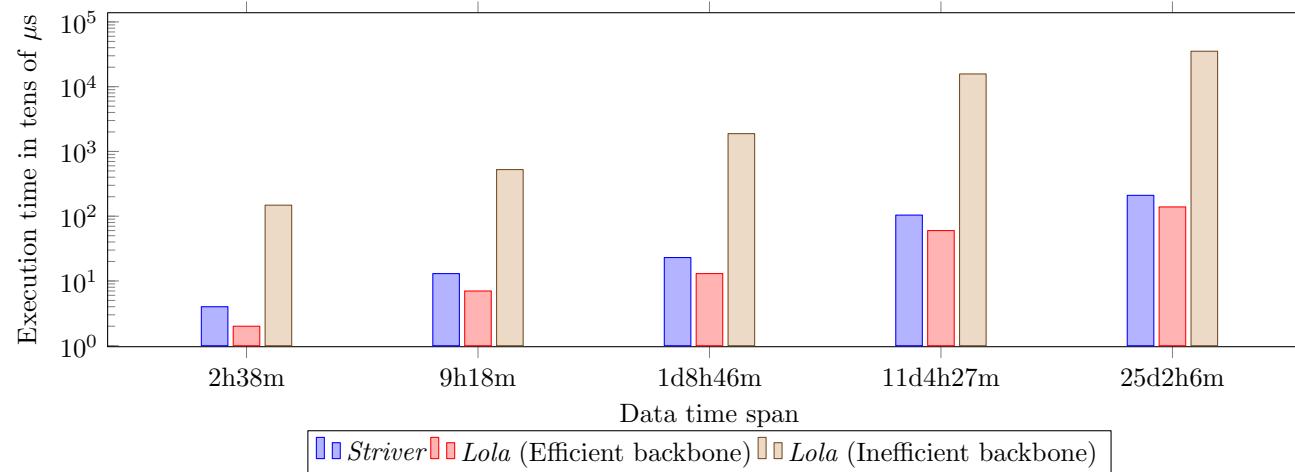
HLola with two different backbones

Empirical evaluation

Two Haskell programs: *HLola* and *HStriver* (available on GitHub)

Dataset: Orange4Home

HLola with two different backbones

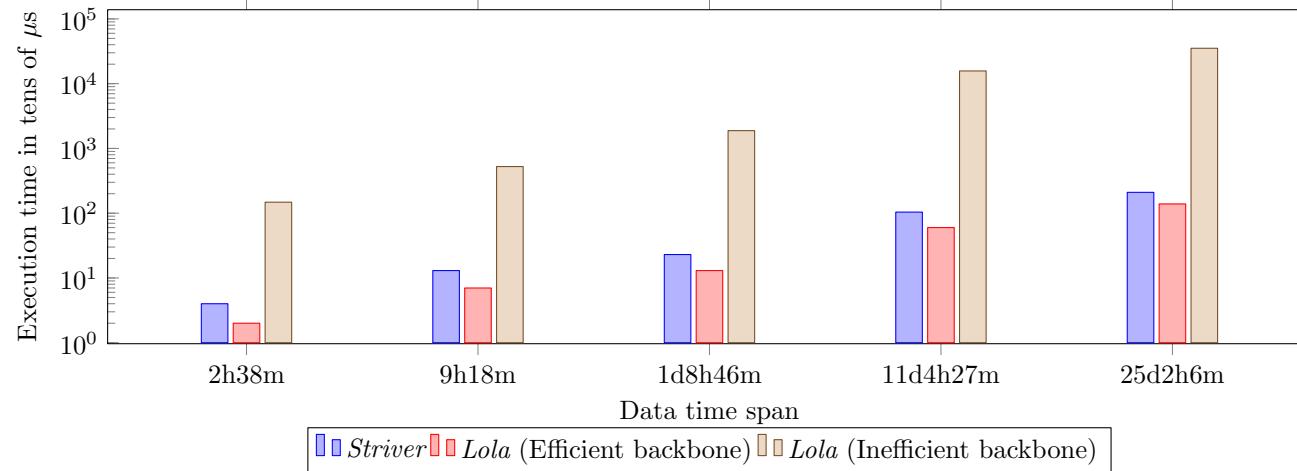


Empirical evaluation

Two Haskell programs: *HLola* and *HStriver* (available on GitHub)

Dataset: Orange4Home

HLola with two different backbones



Asynchronous → Synchronous works for three backbones:

- ▶ atomic clock
- ▶ event-based systems
- ▶ event-based system + predictable clocks

Conclusion 1: SRV

SRV extends logics for RV

- ▶ **easy to use** and **expressive**
- ▶ for rich verdicts
- ▶ formal semantics
- ▶ analysis of bounded resources (formal calculations)

Conclusion 1: SRV

SRV extends logics for RV

- ▶ **easy to use** and **expressive**
- ▶ for rich verdicts
- ▶ formal semantics
- ▶ analysis of bounded resources (formal calculations)

- ▶ Take home #1:
 - ▶ separation between **temporal algorithm** and **data**

Stream Runtime Verification **Moto**:

Monitoring = **temporal dependencies** +
data manipulation

Conclusion 2: Synchronous vs Asynchronous

- ▶ Two models of time:
 - ▶ Synchronous: Lola
 - value equation
 - ▶ Asynchronous: Striver
 - ticking equation
 - value equation

Conclusion 2: Synchronous vs Asynchronous

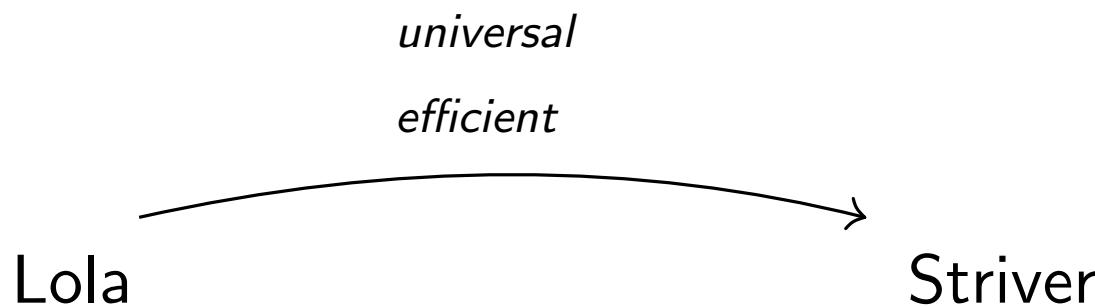
- ▶ Two models of time:
 - ▶ Synchronous: Lola
 - value equation
 - ▶ Asynchronous: Striver
 - ticking equation
 - value equation
- ▶ Take home #2
(synchronous \leftrightarrow asynchronous):

Lola

Striver

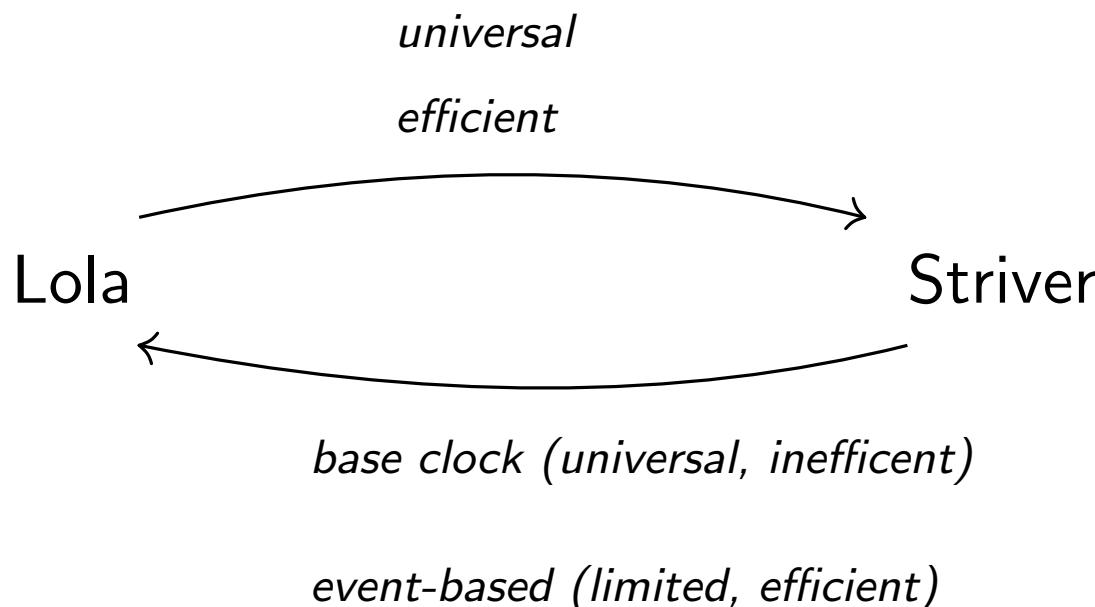
Conclusion 2: Synchronous vs Asynchronous

- ▶ Two models of time:
 - ▶ Synchronous: Lola
 - value equation
 - ▶ Asynchronous: Striver
 - ticking equation
 - value equation
- ▶ Take home #2
(synchronous \leftrightarrow asynchronous):



Conclusion 2: Synchronous vs Asynchronous

- ▶ Two models of time:
 - ▶ Synchronous: Lola
 - value equation
 - ▶ Asynchronous: Striver
 - ticking equation
 - value equation
- ▶ Take home #2
(synchronous \leftrightarrow asynchronous):



Current and Future work

- ▶ Decentralized SRV
- ▶ Distributed SRV
- ▶ Uncertainty and errors
- ▶ Systems

- HLola (TACAS'21)



<https://github.com/imdea-software/hlola>

<https://software.imdea.org/hlola/>

- HStriver

<https://github.com/imdea-software/hstriver>

- ▶ Applications

- UAV planning
 - AI shielding