Temporal Databases

P.J. McBrien

Imperial College London

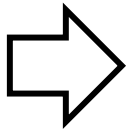
What is a 'Temporal Database'

person			
name	salary	start_time	end_time
Peter	20000	1/1/1995	31/12/1997
Peter	22000	1/1/1998	15/4/1998
Peter	23000	16/4/1998	31/8/1998
Peter	25000	1/1/1999	31/10/2002
Mary	24000	1/1/1995	30/11/1998
Mary	26000	1/12/1998	15/3/2003

- Holds data with some temporal dimension
- Provides support for query processing over that data
  - *How do you find periods a person worked for the company?*

Temporal Query Processing

```
SELECT name, start_time , end_time
FROM person
```



person		
name	start_time	end_time
Peter	1/1/1995	31/12/1997
Peter	1/1/1998	15/4/1998
Peter	16/4/1998	31/8/1998
Peter	1/1/1999	31/10/2002
Mary	1/1/1995	30/11/1998
Mary	1/12/1998	15/3/2003

- *How do we merge periods?*

person		
name	start_time	end_time
Peter	1/1/1995	31/8/1998
Peter	1/1/1999	31/10/2002
Mary	1/1/1995	15/3/2003

Two Types of Time to Store

- **valid time** time data valid in UoD.
  e.g. *time payments and withdrawal movements made in branch*
- **transaction time** time data valid in the DBMS
  e.g. *from time the clerk entered the payment or withdrawal into the DBMS*
- Each type of time can be called a 'flow of time'
- Same query language can be used on either 'flow of time'

Modelling a flow of time

Simple method	*Complex method*
Discrete fixed **chronons**, have next and previous time	**Continuous** always add a new time point between two other
Bounded first and last time	**Unbounded** all times may be represented
Linear one view of each time	**Branching** multiple views of each time

Temporal Structure

t	time	account			movement		
	time	no	type	rate	mid	no	amount
0	1/1/1999	100	'current'	NULL	0000	100	220.00
		101	'deposit'	8.50	0001	102	2520.00
		102	'deposit'	8.50			
1	2/1/1999	100	'current'	NULL	0002	101	100.00
		101	'deposit'	8.25	0003	104	1000.00
		103	'current'	NULL			
		104	'current'	NULL			
2	3/1/1999	100	'current'	NULL	0004	104	-500.00
		101	'deposit'	8.25	0005	105	500.00
		103	'current'	NULL	0006	100	100.23
		104	'current'	NULL			
		105	'deposit'	8.50			

Using RA on a Snapshop of the Temporal Structure

t	time	account			movement		
	time	no	type	rate	mid	no	amount
1	2/1/1999	100	'current'	NULL	0002	101	100.00
		101	'deposit'	8.25	0003	104	1000.00
		103	'current'	NULL			
		104	'current'	NULL			

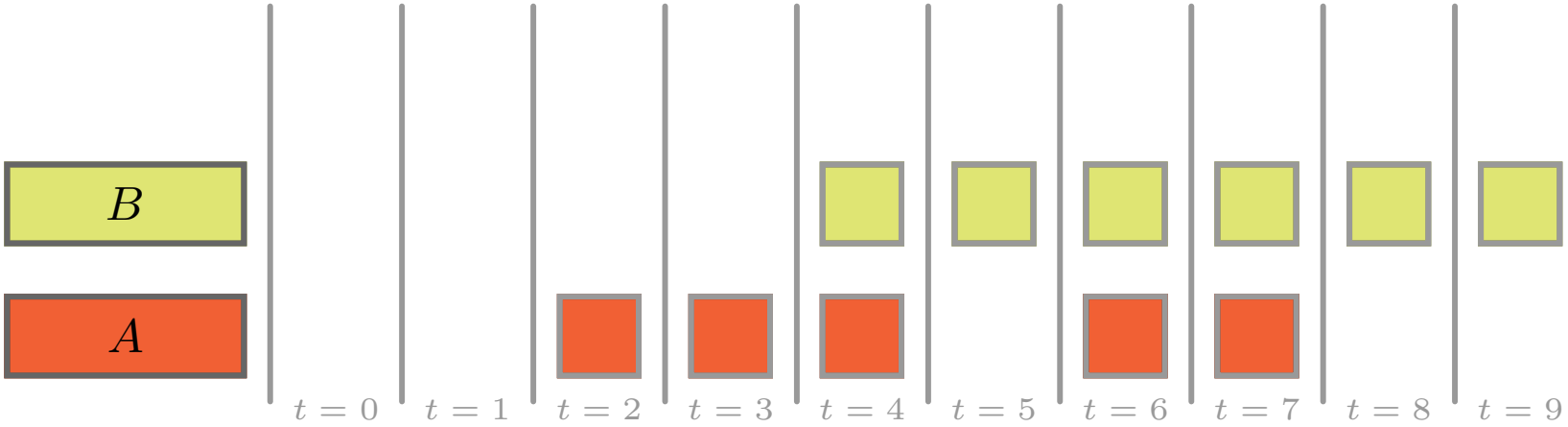
$eval(\pi_{\texttt{no}}\texttt{account}, 1)$

no
100
101
103
104

$eval(\pi_{\texttt{no}}\texttt{account} \times \pi_{\texttt{mid}}\texttt{movement}, 1)$

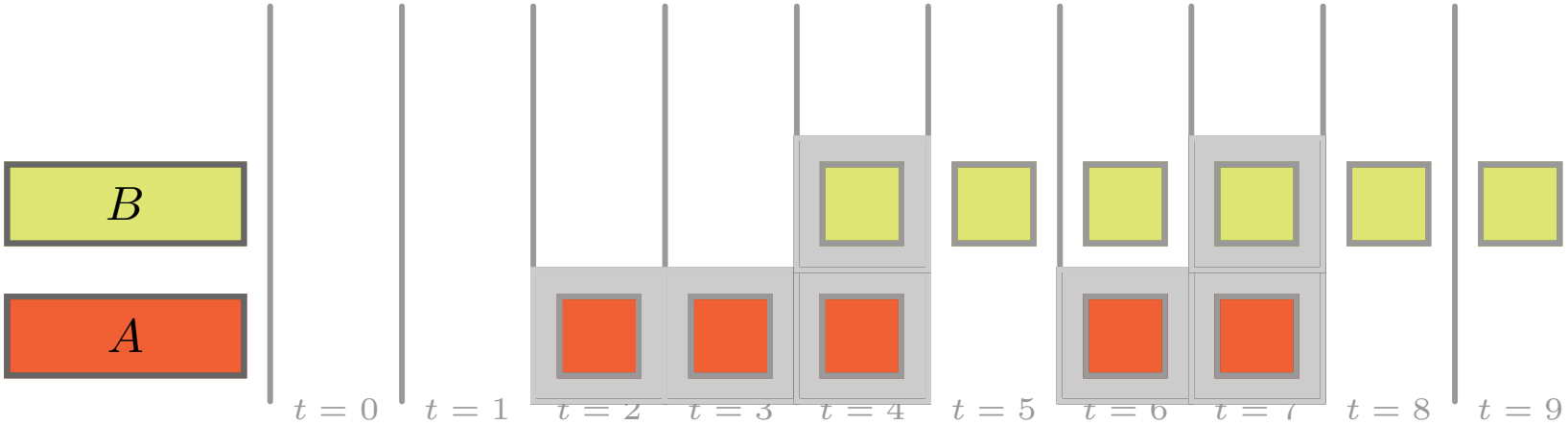
no	mid	no	mid
100	0002	103	0002
100	0003	103	0003
101	0002	104	0002
101	0003	104	0003

US-Logic: A Until B



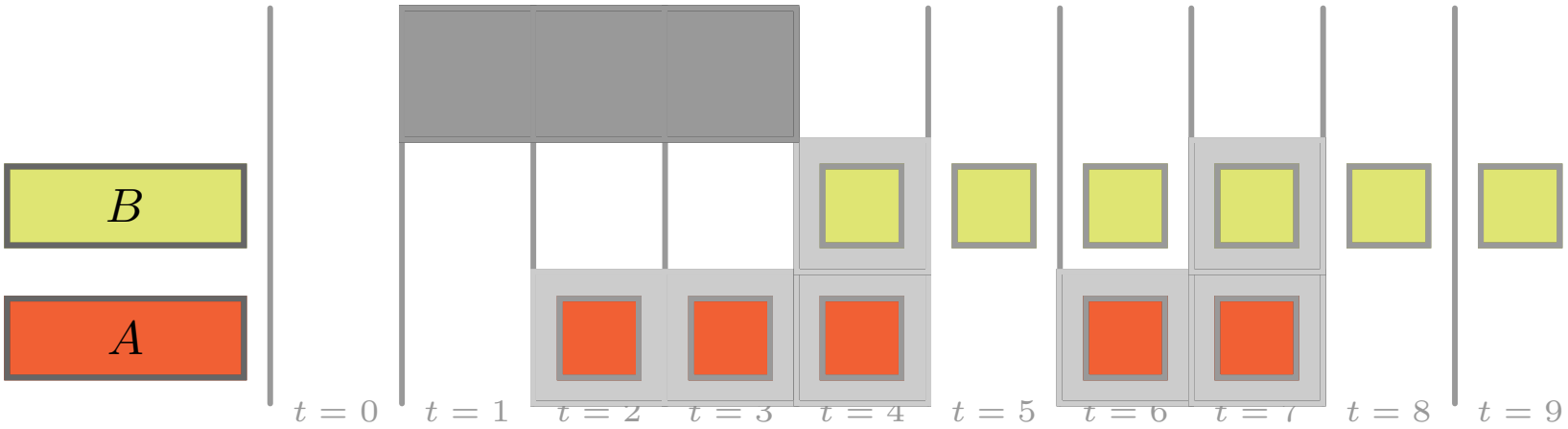
A will hold at every time up to and including the time when B holds

US-Logic: A Until B



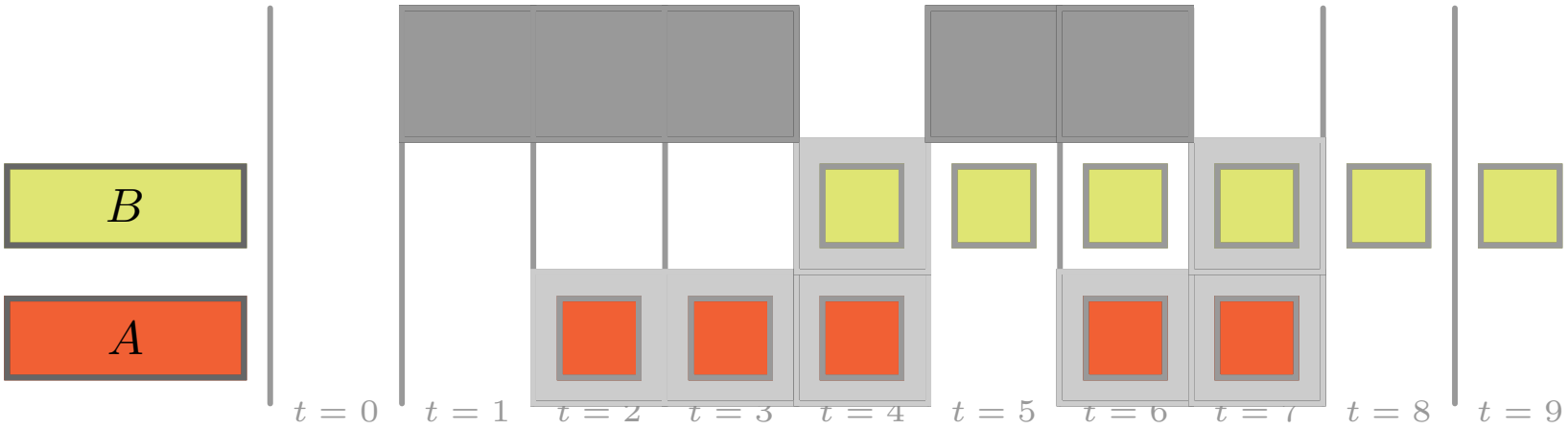
A will hold at every time up to and including the time when B holds

US-Logic: A Until B



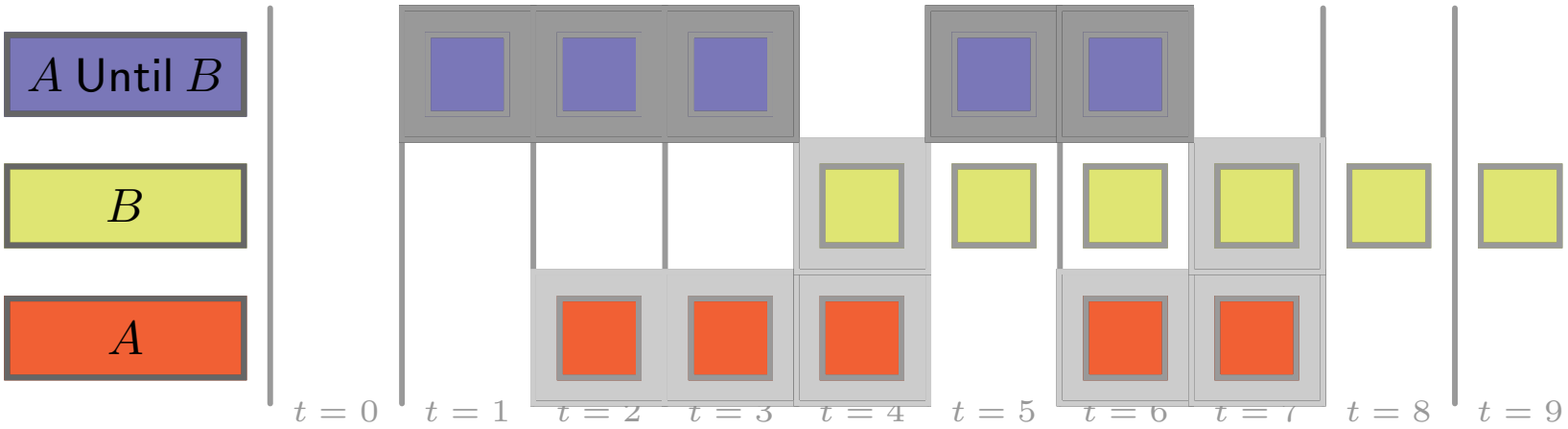
A will hold at every time up to and including the time when B holds

US-Logic: A Until B



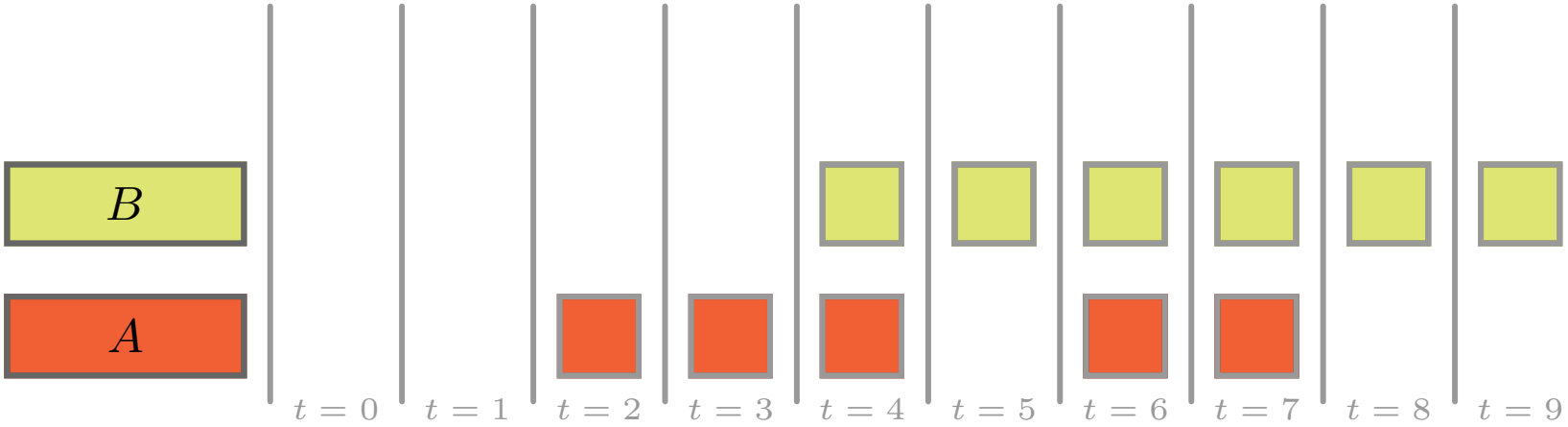
A will hold at every time up to and including the time when B holds

US-Logic: A Until B



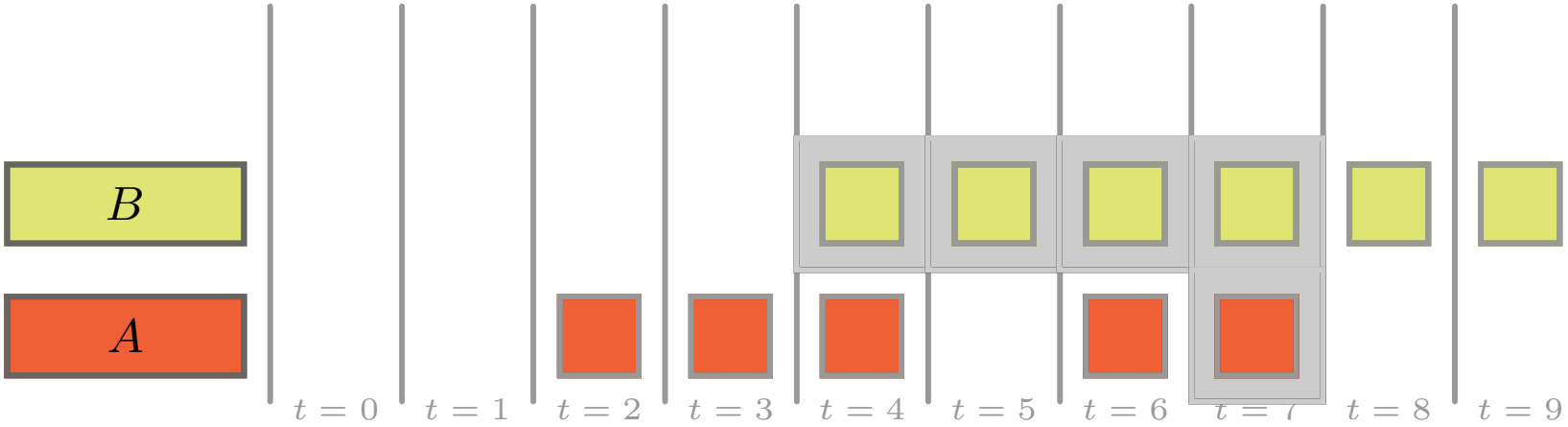
A will hold at every time up to and including the time when B holds

US-Logic: B Until A



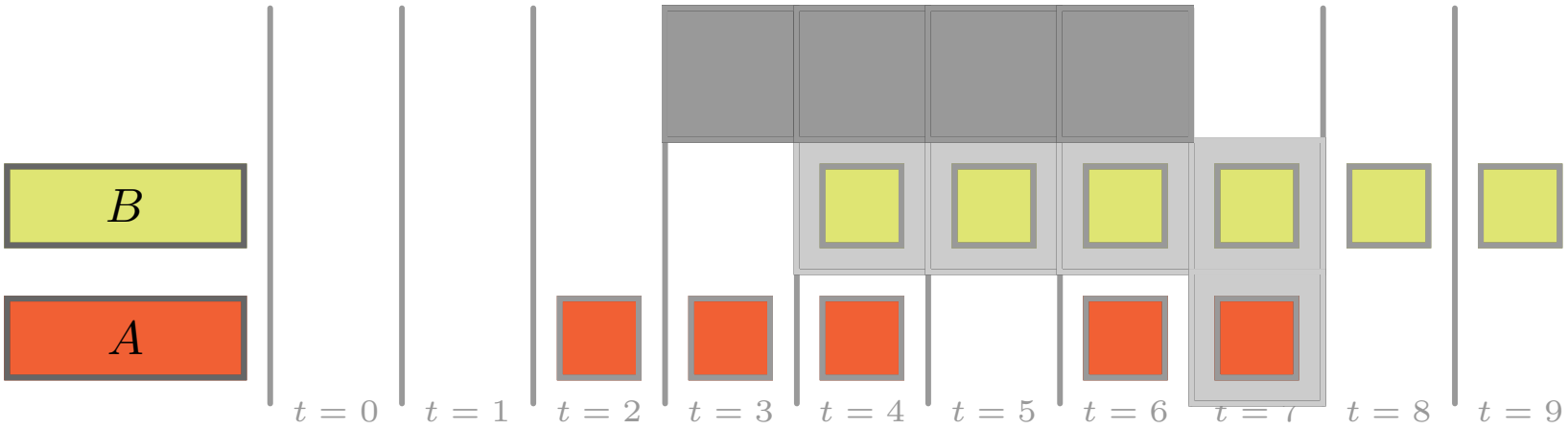
B will hold at every time up to and including the time when A holds

US-Logic: B Until A



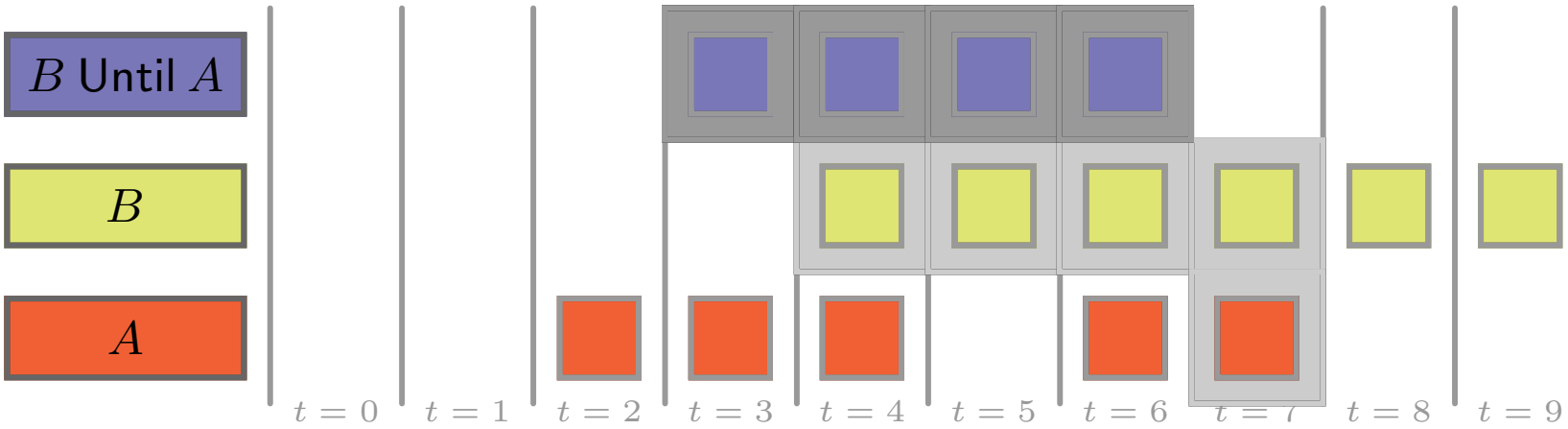
B will hold at every time up to and including the time when A holds

US-Logic: B Until A



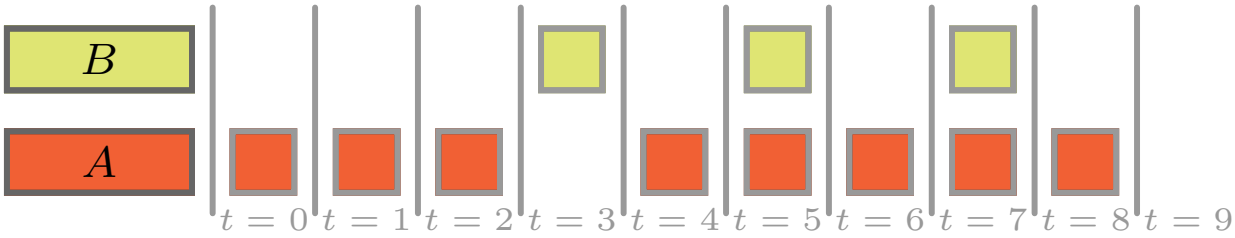
B will hold at every time up to and including the time when A holds

US-Logic: B Until A



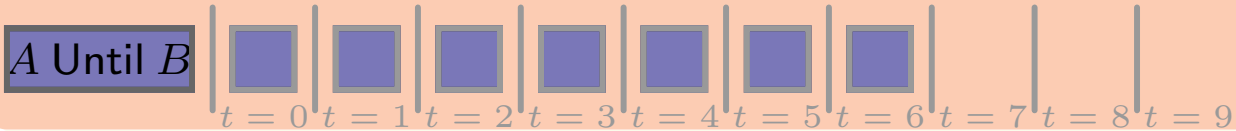
B will hold at every time up to and including the time when A holds

Quiz 1: Until modal logic operator

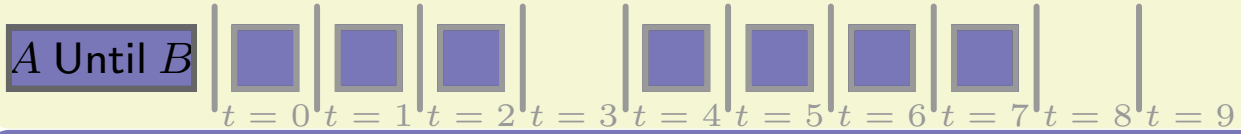


When does A Until B hold?

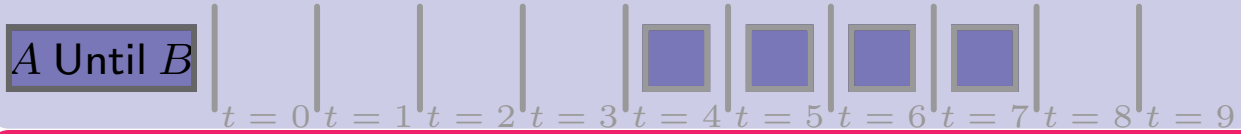
A



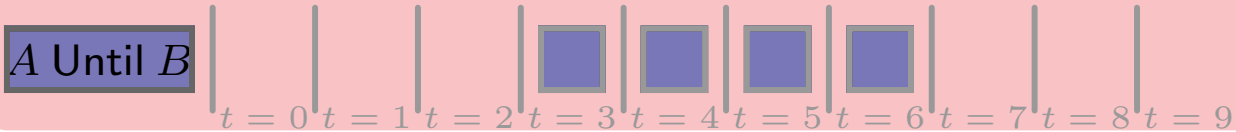
B



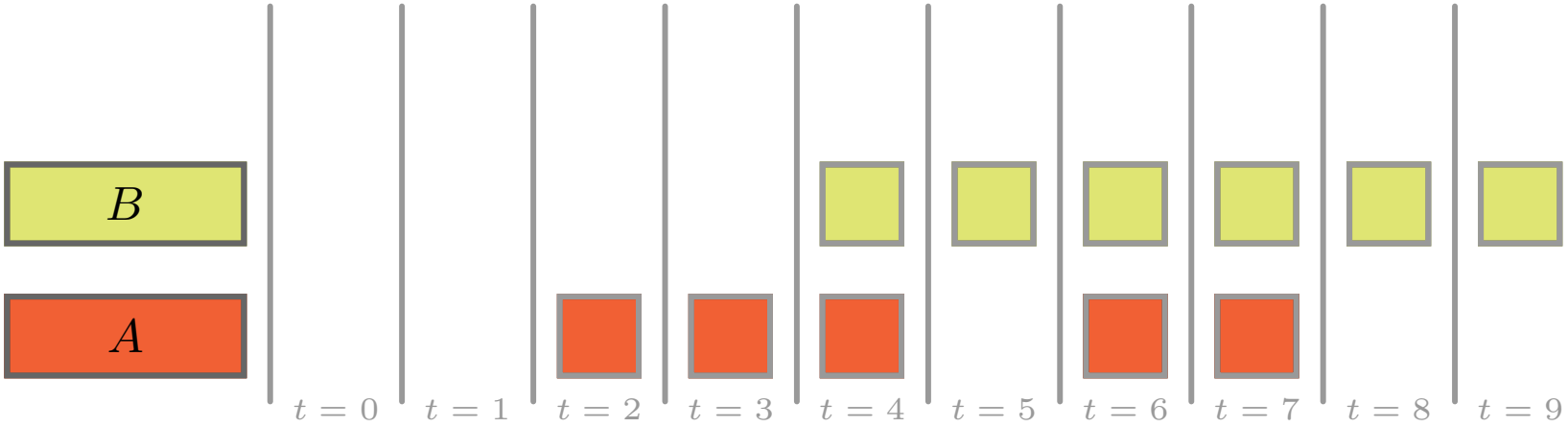
C



D

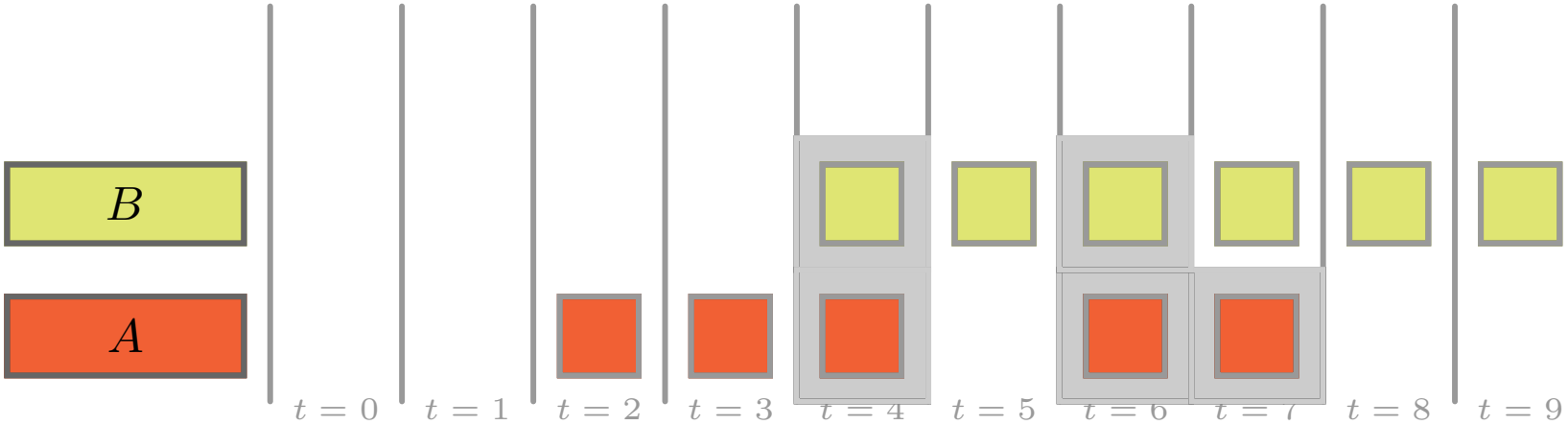


US-Logic: A Since B



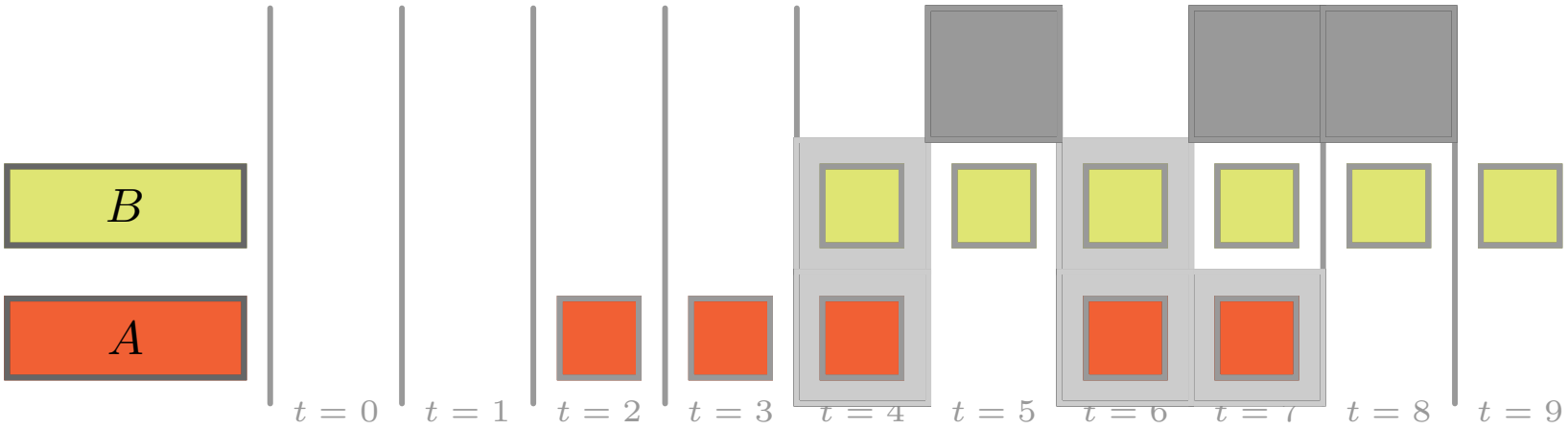
A has held at every time since and including the time when B held

US-Logic: A Since B



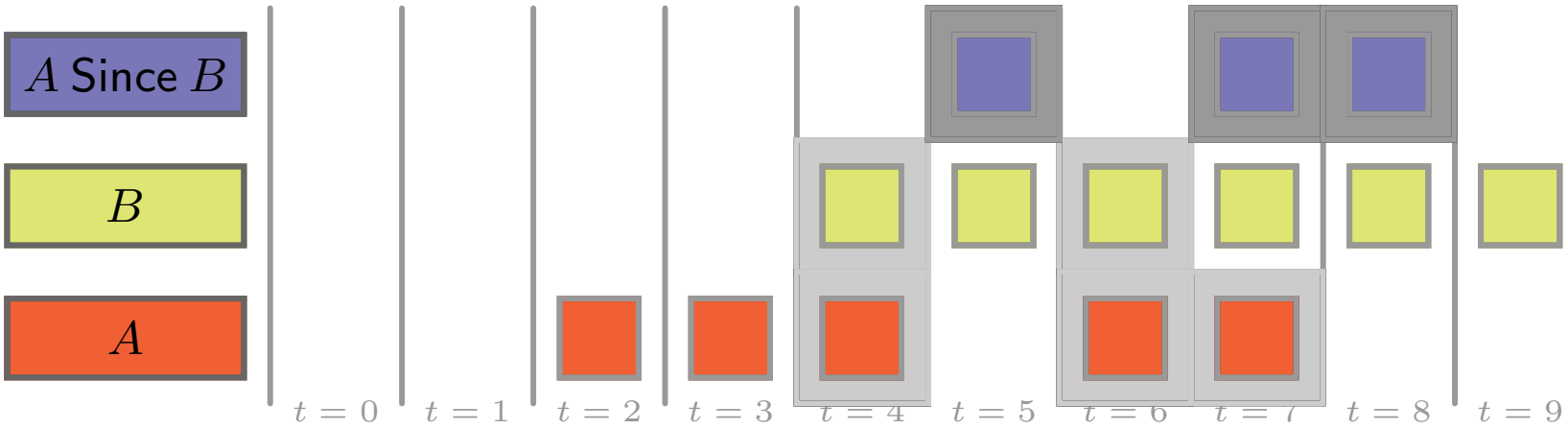
A has held at every time since and including the time when B held

US-Logic: A Since B



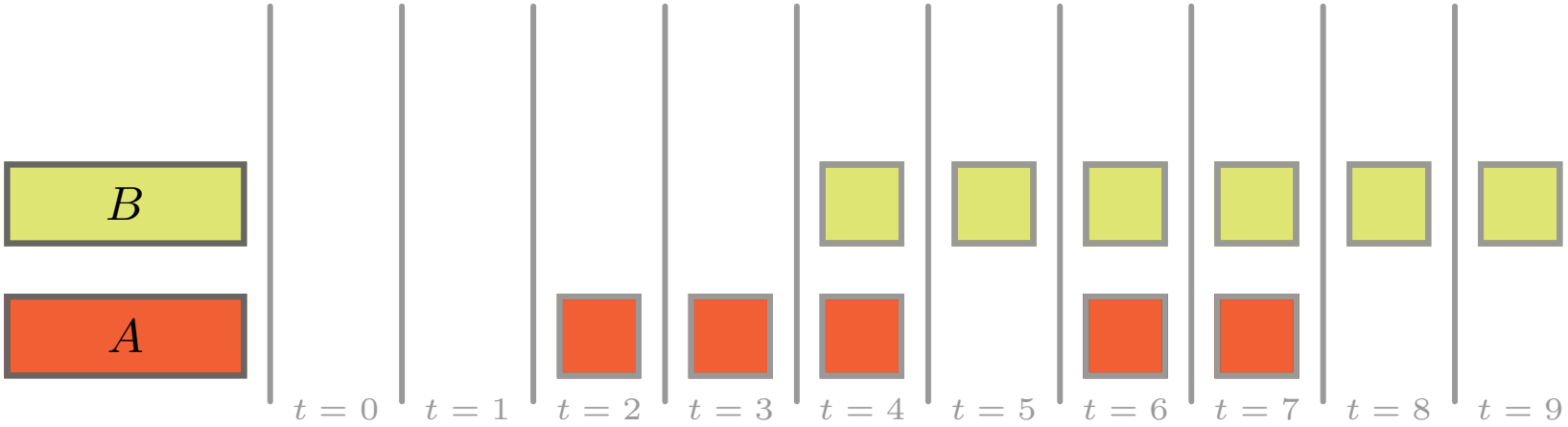
A has held at every time since and including the time when B held

US-Logic: A Since B



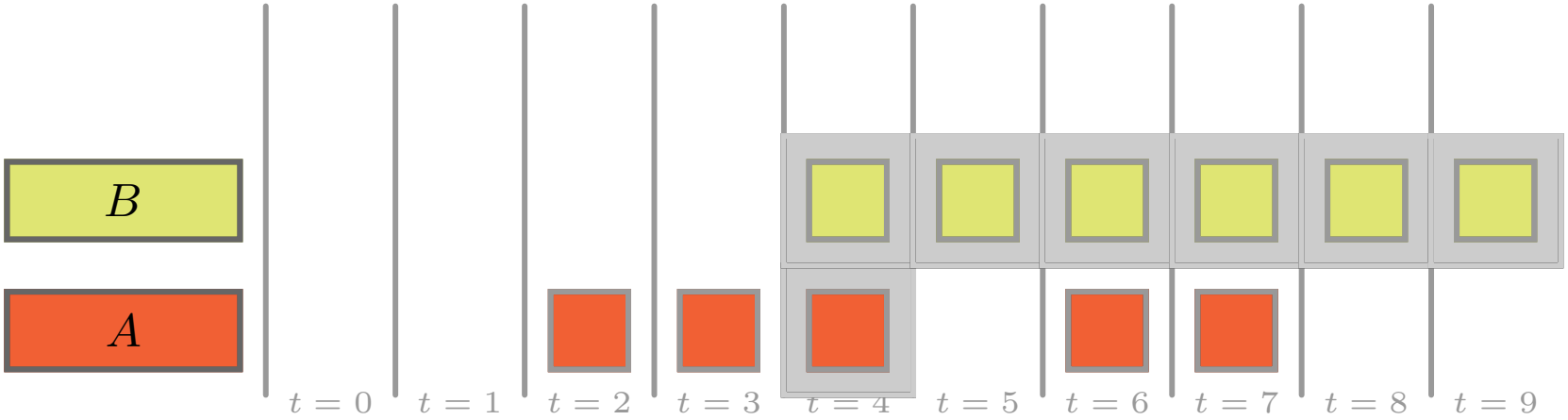
A has held at every time since and including the time when B held

US-Logic: B Since A



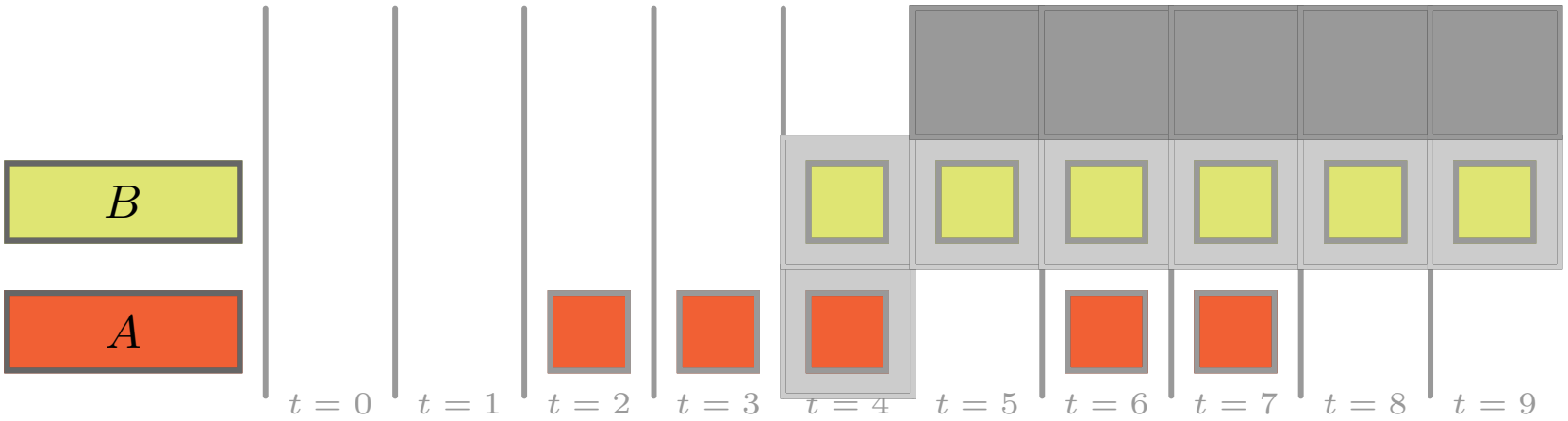
B has held at every time since and including the time when A held

US-Logic: B Since A



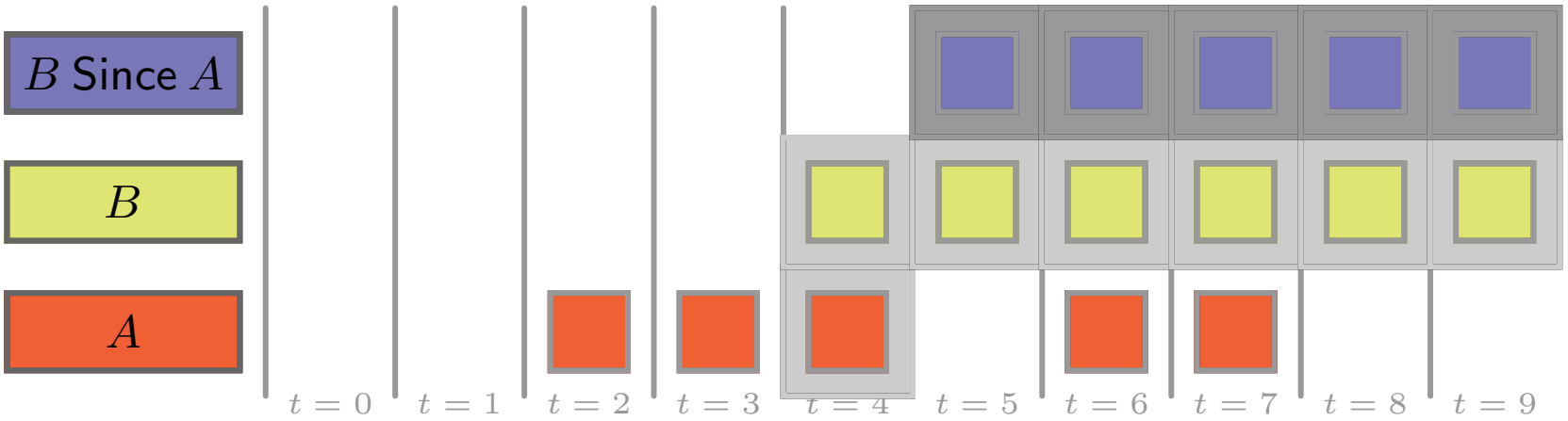
B has held at every time since and including the time when A held

US-Logic: B Since A



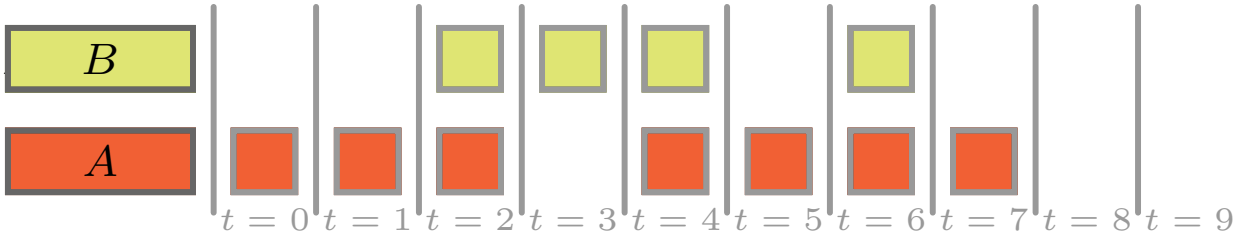
B has held at every time since and including the time when A held

US-Logic: B Since A



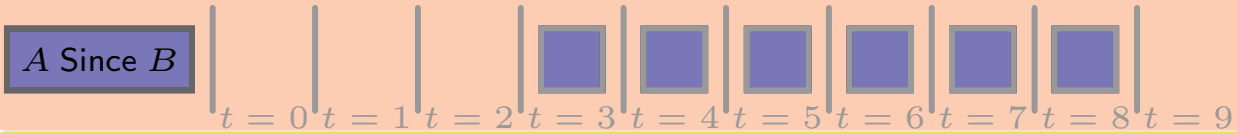
B has held at every time since and including the time when A held

Quiz 2: Since modal logic operator

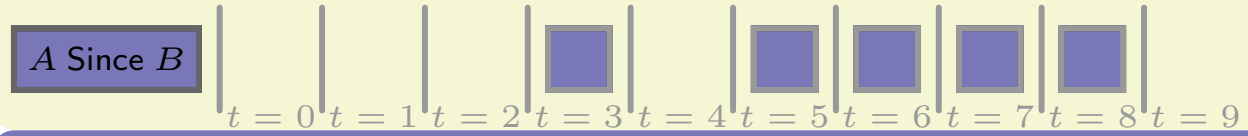


When does A Since B hold?

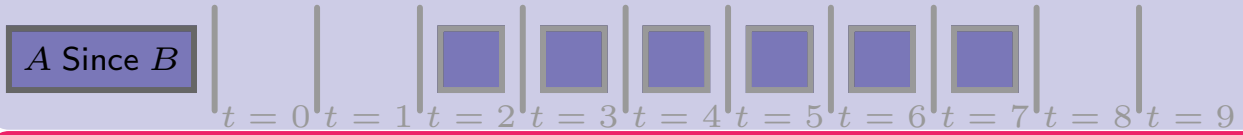
A



B

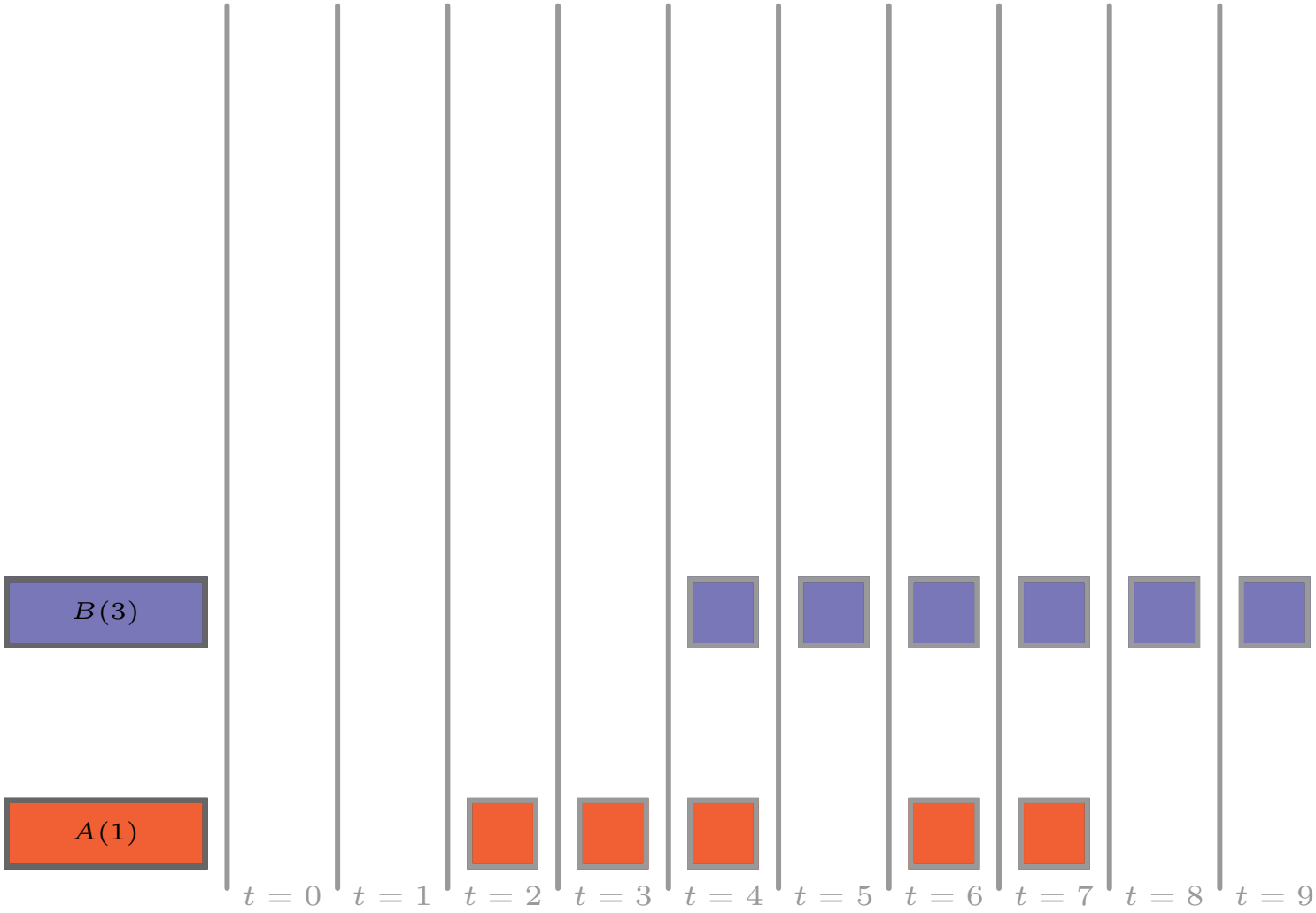


C

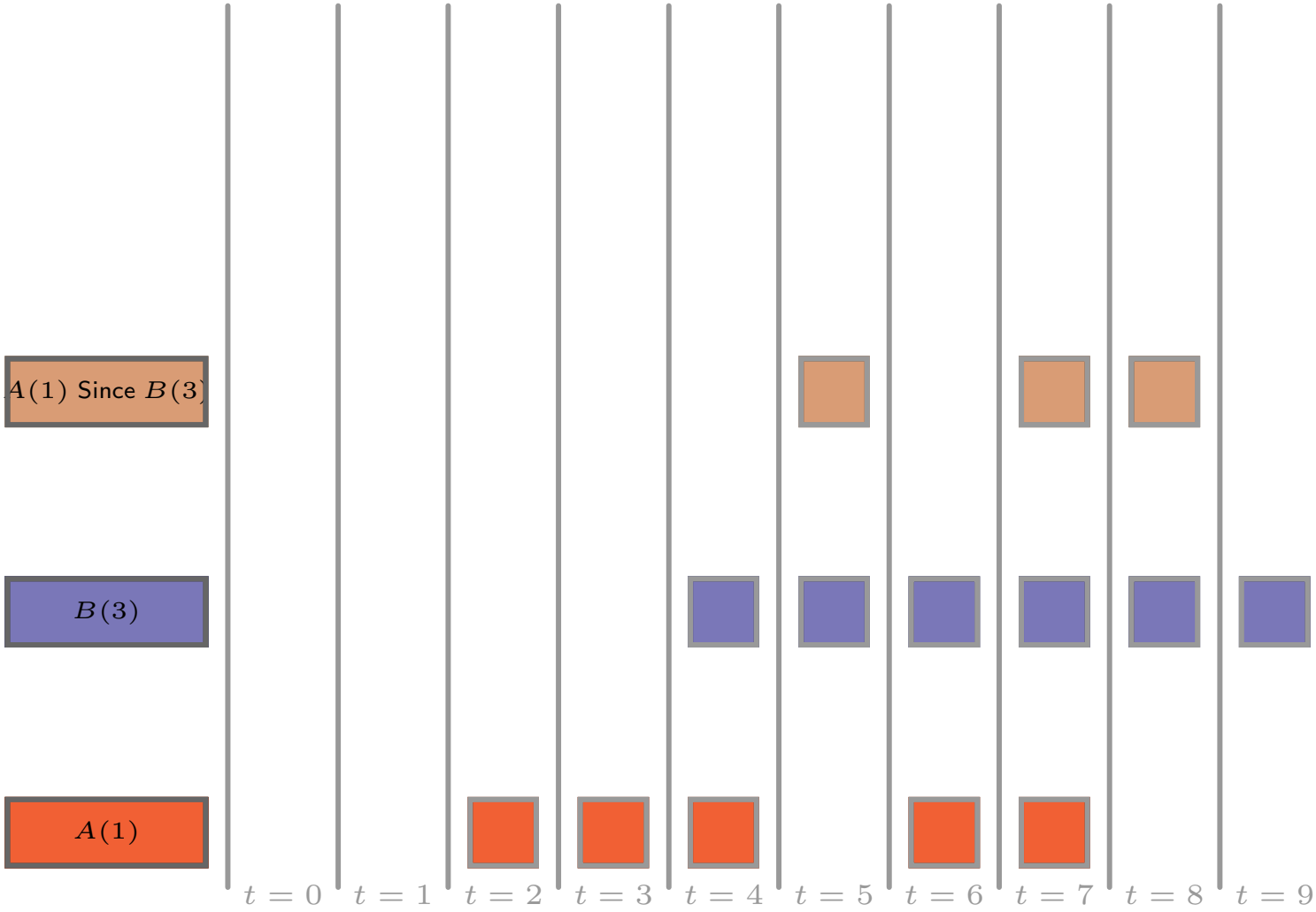


D

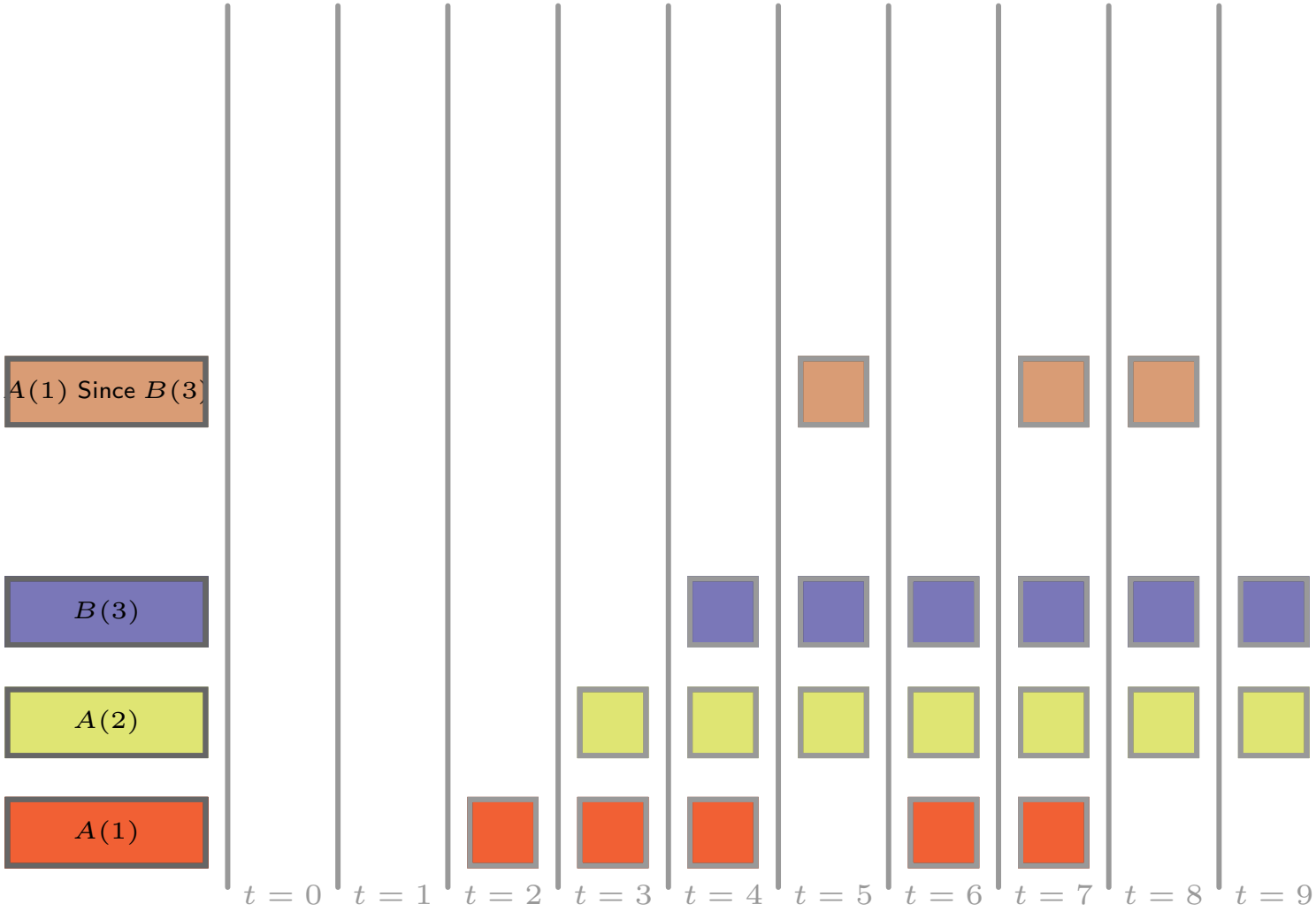
Predicate US Logic



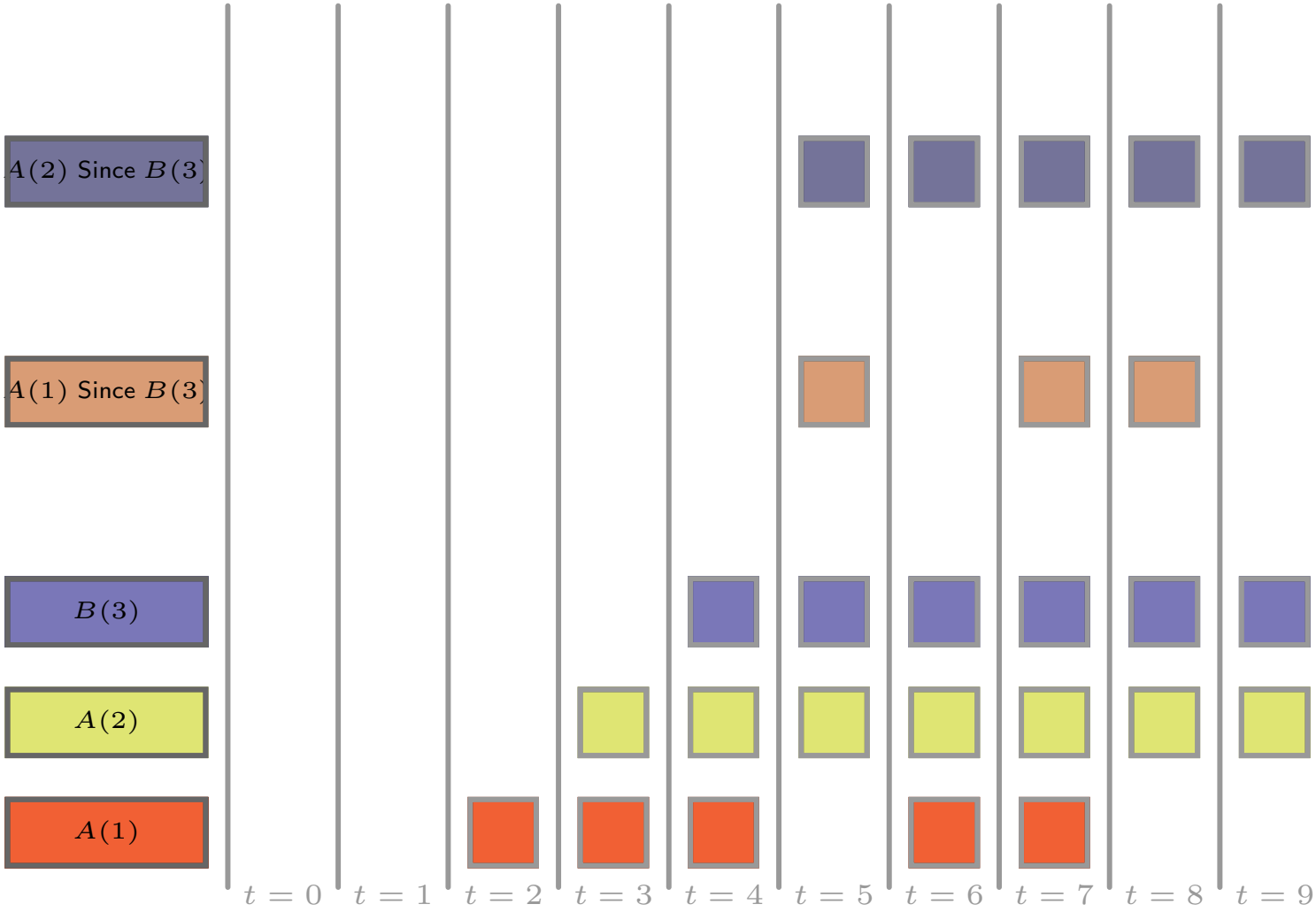
Predicate US Logic



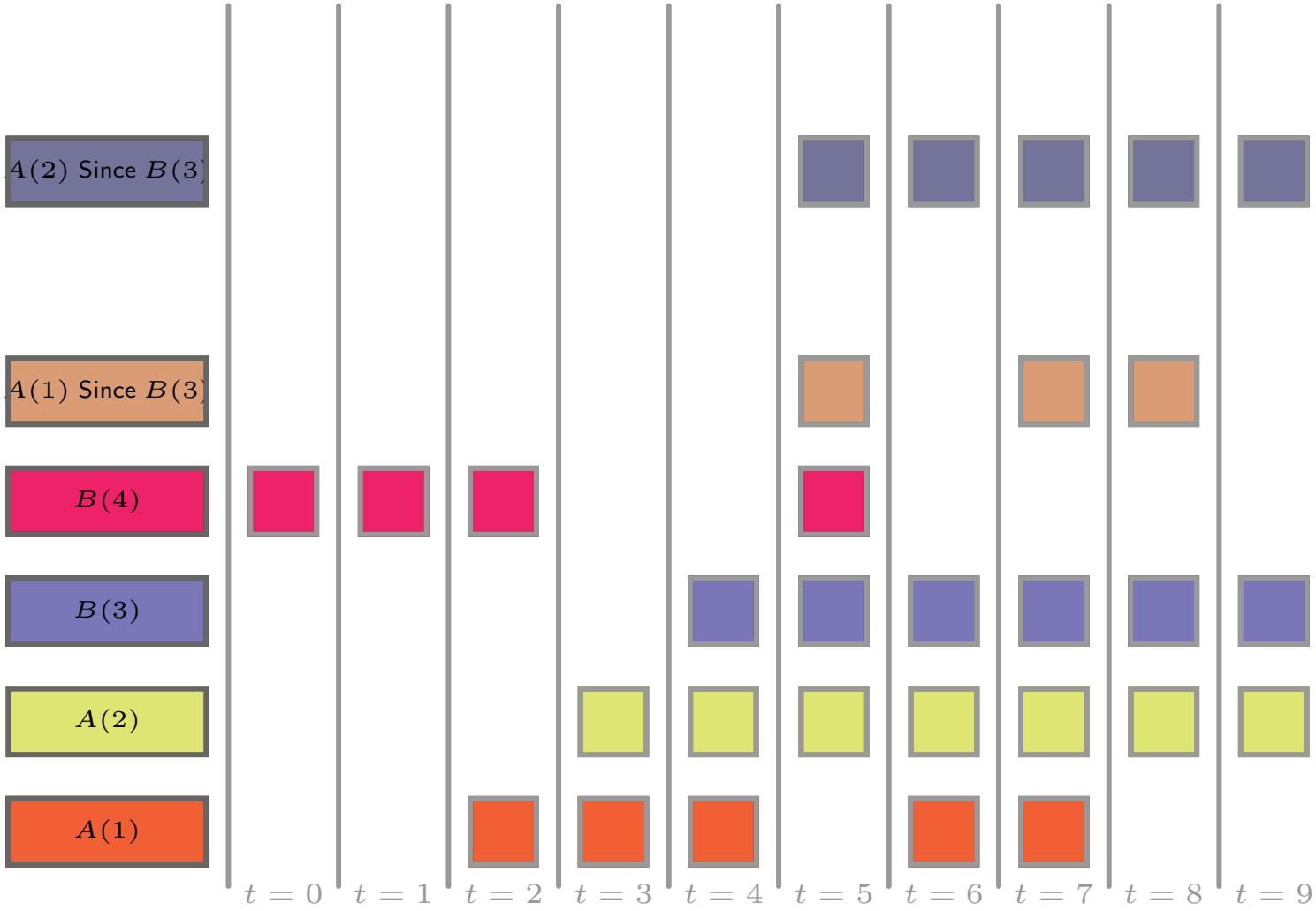
Predicate US Logic



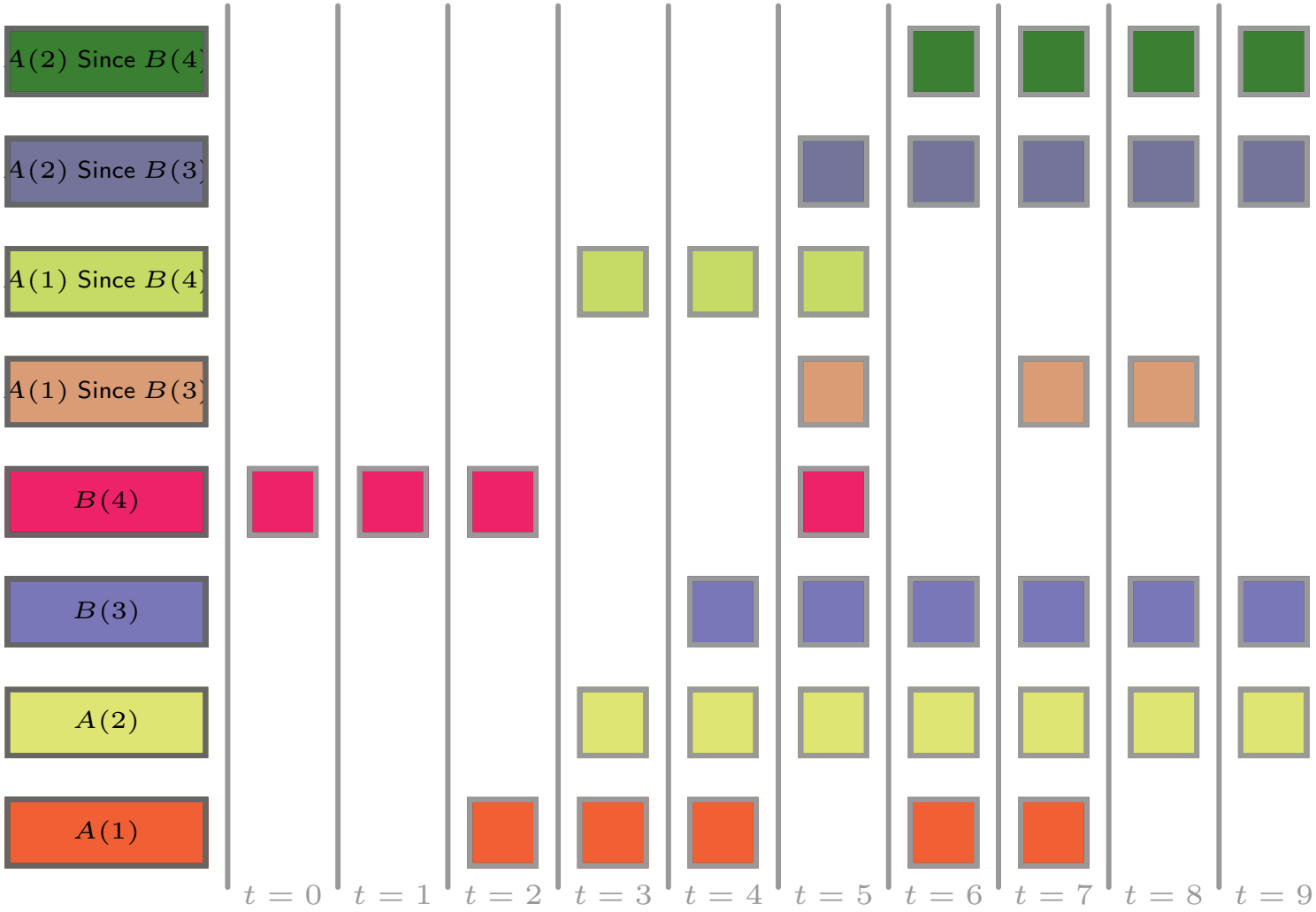
Predicate US Logic



Predicate US Logic



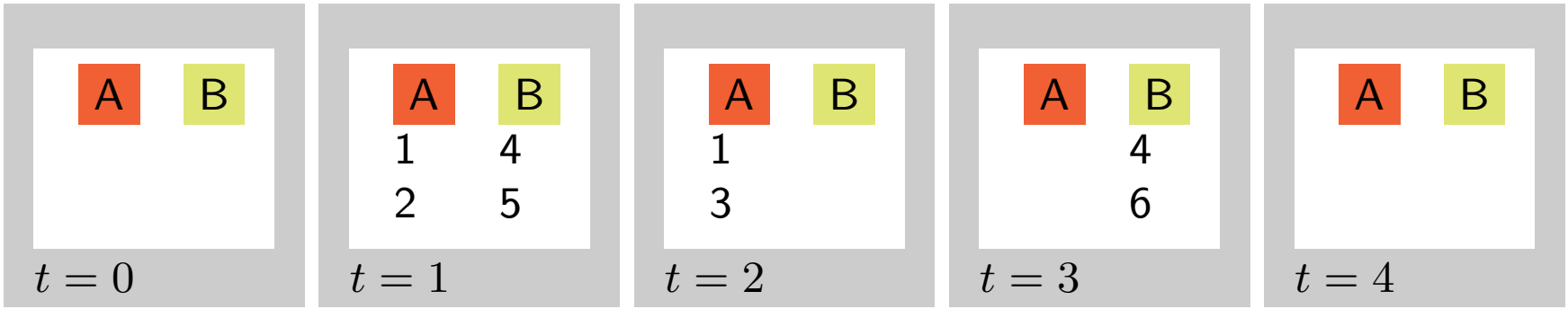
Predicate US Logic



Relational form of US Logic

Since Product

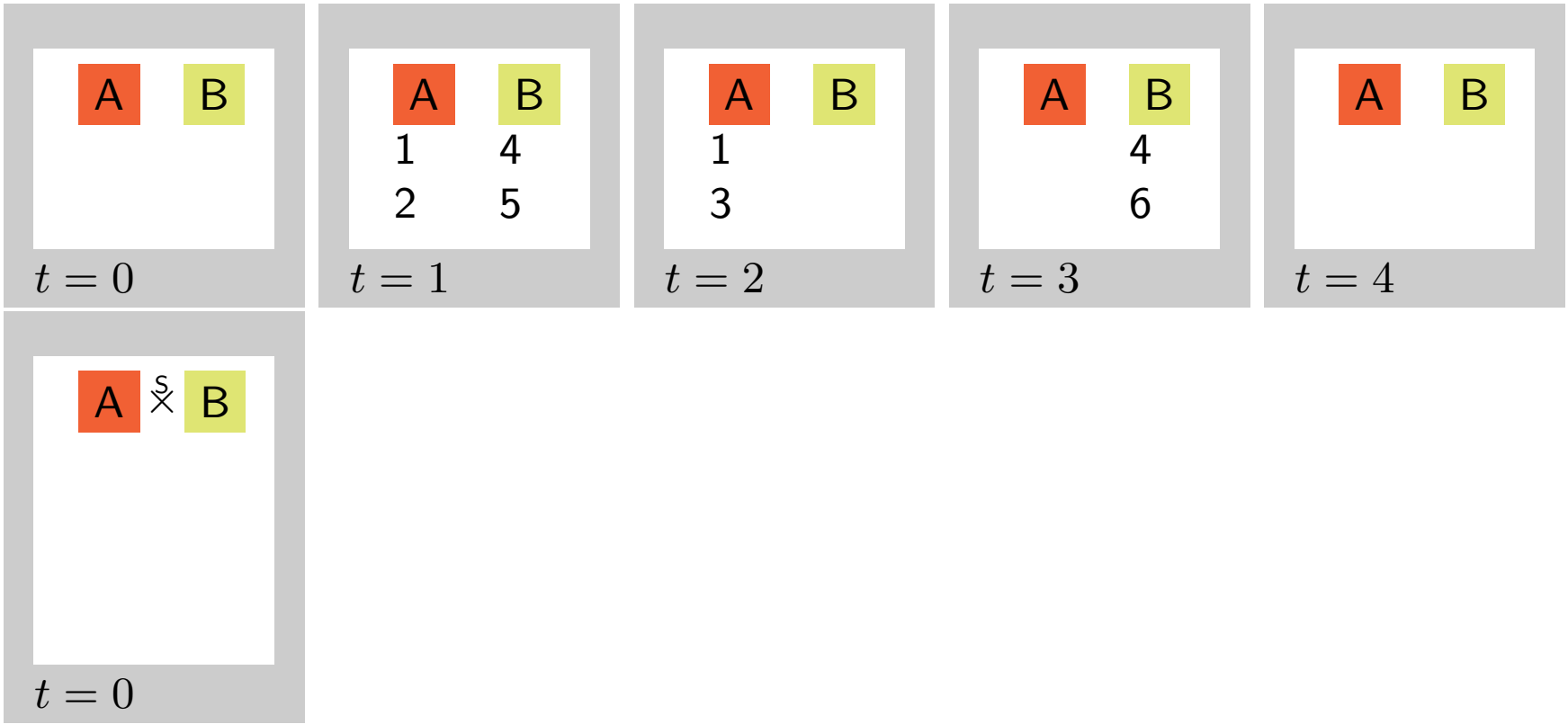
$$eval(R \overset{S}{\times} S, t) = eval((R \times S) \cup ((R \overset{S}{\times} S) \ltimes R), t-1)$$



Relational form of US Logic

Since Product

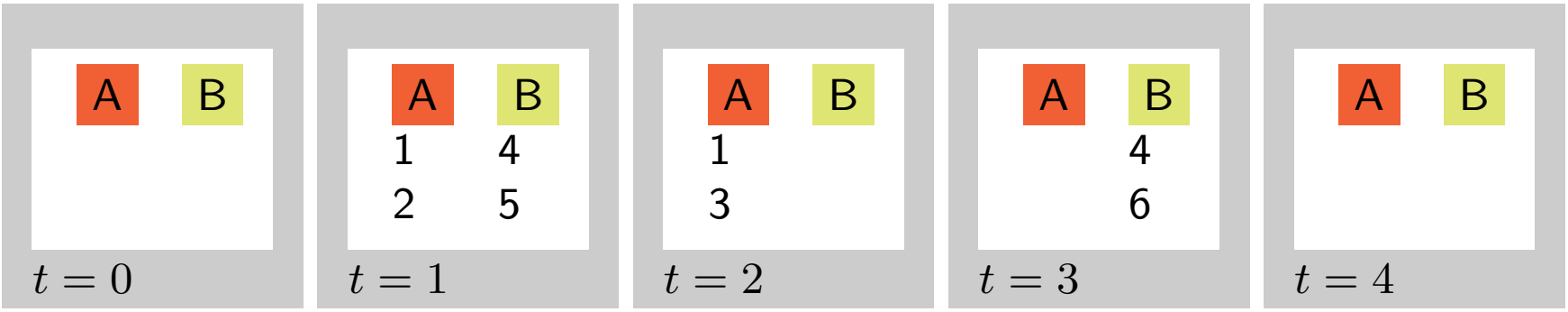
$$eval(R \overset{S}{\times} S, t) = eval((R \times S) \cup ((R \overset{S}{\times} S) \ltimes R), t-1)$$

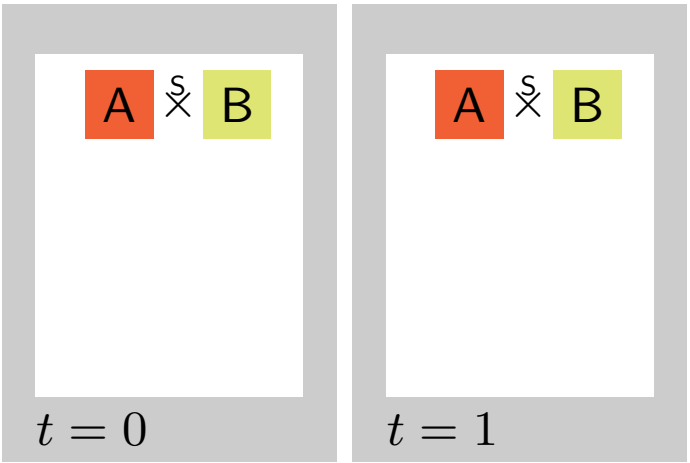


Relational form of US Logic

Since Product

$$eval(R \overset{S}{\times} S, t) = eval((R \times S) \cup ((R \overset{S}{\times} S) \ltimes R), t-1)$$





Relational form of US Logic

Since Product

$$eval(R \overset{S}{\times} S, t) = eval((R \times S) \cup ((R \overset{S}{\times} S) \ltimes R), t-1)$$

A	B

$t = 0$

A	B
1	4
2	5

$t = 1$

A	B
1	
3	

$t = 2$

A	B
	4
	6

$t = 3$

A	B

$t = 4$

A $\overset{S}{\times}$	B

$t = 0$

A $\overset{S}{\times}$	B

$t = 1$

A $\overset{S}{\times}$	B
1	4
1	5
2	4
2	5

$t = 2$

Relational form of US Logic

Since Product

$$eval(R \overset{S}{\times} S, t) = eval((R \times S) \cup ((R \overset{S}{\times} S) \ltimes R), t-1)$$

A	B

$t = 0$

A	B
1	4
2	5

$t = 1$

A	B
1	
3	

$t = 2$

A	B
	4
	6

$t = 3$

A	B

$t = 4$

A $\overset{S}{\times}$	B

$t = 0$

A $\overset{S}{\times}$	B

$t = 1$

A $\overset{S}{\times}$	B
1	4
1	5
2	4
2	5

$t = 2$

A $\overset{S}{\times}$	B
1	4
1	5

$t = 3$

Relational form of US Logic

Since Product

$$eval(R \overset{S}{\times} S, t) = eval((R \times S) \cup ((R \overset{S}{\times} S) \ltimes R), t-1)$$

A	B

$t = 0$

A	B
1	4
2	5

$t = 1$

A	B
1	
3	

$t = 2$

A	B
	4
	6

$t = 3$

A	B

$t = 4$

A $\overset{S}{\times}$	B

$t = 0$

A $\overset{S}{\times}$	B

$t = 1$

A $\overset{S}{\times}$	B
1	4
1	5
2	4
2	5

$t = 2$

A $\overset{S}{\times}$	B
1	4
1	5

$t = 3$

A $\overset{S}{\times}$	B

$t = 4$

Relational form of US Logic

Since Product

$$eval(R \overset{S}{\times} S, t) = eval((R \times S) \cup ((R \overset{S}{\times} S) \ltimes R), t-1)$$

A	B

$t = 0$

A	B
1	4
2	5

$t = 1$

A	B
1	
3	

$t = 2$

A	B
	4
	6

$t = 3$

A	B

$t = 4$

A $\overset{S}{\times}$	B

$t = 0$

A $\overset{S}{\times}$	B

$t = 1$

A $\overset{S}{\times}$	B
1	4
1	5
2	4
2	5

$t = 2$

A $\overset{S}{\times}$	B
1	4
1	5

$t = 3$

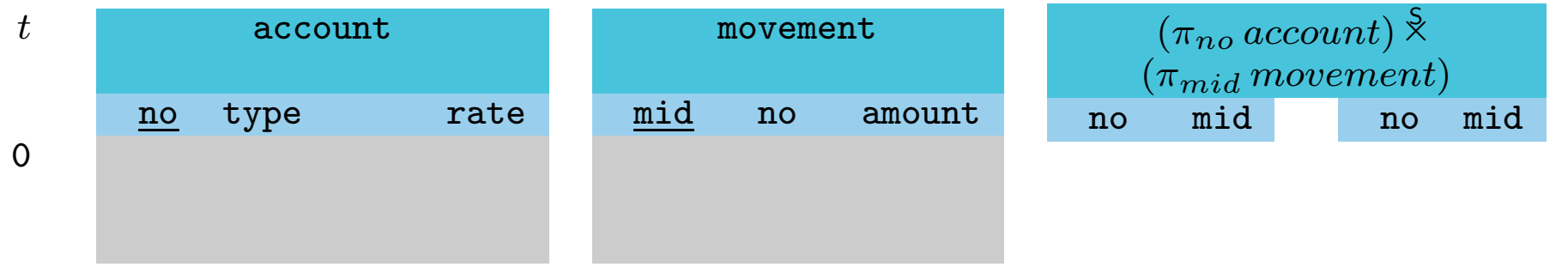
A $\overset{S}{\times}$	B

$t = 4$

Until Product

$$eval(R \overset{U}{\times} S, t) = eval((R \times S) \cup ((R \overset{U}{\times} S) \ltimes R), t+1)$$

Example of since-product



Example of since-product

t

0

account

no	type	rate
100	'current'	NULL
101	'deposit'	8.50
102	'deposit'	8.50

movement

mid	no	amount
0000	100	220.00
0001	102	2520.00

$(\pi_{no}\, account) \overset{S}{\times} (\pi_{mid}\, movement)$

no	mid	no	mid

1

100	0000	101	0001
100	0001	102	0000
101	0000	102	0001

Example of since-product

t	account			movement			$(\pi_{\texttt{no}}\ \texttt{account}) \overset{S}{\times} (\pi_{\texttt{mid}}\ \texttt{movement})$			
	no	type	rate	mid	no	amount	no	mid	no	mid
0	100	'current'	NULL	0000	100	220.00				
	101	'deposit'	8.50	0001	102	2520.00				
	102	'deposit'	8.50							
1	100	'current'	NULL	0002	101	100.00	100	0000	101	0001
	101	'deposit'	8.25	0003	104	1000.00	100	0001	102	0000
	103	'current'	NULL				101	0000	102	0001
	104	'current'	NULL							
2							100	0002	104	0002
							100	0003	104	0003
							101	0002	100	0000
							101	0003	100	0001
							103	0002	101	0000
							103	0003	101	0001

Worksheet: The Since-Product Operator

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

Worksheet: The Since-Product Operator

$t = 0$

loc_a	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

loc_b	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

time
0

$t = 1$

loc_a	
name	town
Peter	London
Jim	Edinburgh

loc_b	
name	town
Peter	London
Jim	Edinburgh

time
1

$t = 2$

loc_a	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

loc_b	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

time
2

$t = 3$

loc_a	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

loc_b	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

time
3

Worksheet: The Since-Product Operator

loc

name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc

name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc

name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc

name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

$\text{eval}(\text{loc}_a \overset{S}{\times} \text{loc}_b, 2)$

loc_a.name	loc_a.town	loc_b.name	loc_b.town
Peter	London	Peter	London
Jim	Edinburgh	Peter	London
Peter	London	Jim	Edinburgh
Jim	Edinburgh	Jim	Edinburgh
Peter	London	Fred	Manchester
Jim	Edinburgh	Fred	Manchester

Worksheet: The Since-Product Operator

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

eval(loc $\overset{S}{\times}$ ($\sigma_{\text{time}=0}$ time), 3)

name	town	time
Peter	London	0

Find who has been working in the same place since time 0

Worksheet: The Since-Product Operator

loc

name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc

name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc

name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc

name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

eval$((\pi_{\text{name}}\ \text{loc}) \overset{\text{S}}{\times} \sigma_{\text{time}=0}\ \text{time}, 3)$

name	time
Peter	0
Jim	0

Find who has been working since time 0

Worksheet: The Since-Product Operator

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

eval$(\pi_{\mathsf{loc_b}.\mathsf{town}}((\pi_{\mathsf{town}}\ \sigma_{\mathsf{town}=\text{`}Glasgow\text{'}}\ \mathsf{loc_a}) \overset{\mathsf{S}}{\times} \mathsf{loc_b}), 4)$

$\mathsf{loc_b}.\mathsf{town}$
London
Glasgow
Edinburgh
Bristol

Find towns that have had people working in them for any period during the period Glasgow has had people working in it

Deriving other operators

- Rather like Cartesian product, since-product and until-product not very useful alone
- Form basis for defining other operators

Deriving other operators

- Rather like Cartesian product, since-product and until-product not very useful alone
- Form basis for defining other operators

since-join ($\overset{S}{\bowtie}$)

$$eval(R \overset{S}{\bowtie} S, t) = eval(\sigma_{R.A=S.A}(R \overset{S}{\times} S), t)$$

Deriving other operators

- Rather like Cartesian product, since-product and until-product not very useful alone
- Form basis for defining other operators

since-join ($\overset{S}{\bowtie}$)

$$eval(R \overset{S}{\bowtie} S, t) = eval(\sigma_{R.A=S.A}(R \overset{S}{\times} S), t)$$

Find movements of accounts still open until yesterday

$$\pi_{\mathsf{no,mid,amount}}(\pi_{\mathsf{no}}\mathsf{account} \overset{S}{\bowtie} \mathsf{movement})$$

Deriving other operators

- Rather like Cartesian product, since-product and until-product not very useful alone
- Form basis for defining other operators

since-join ($\overset{S}{\bowtie}$)

$$eval(R \overset{S}{\bowtie} S, t) = eval(\sigma_{R.A=S.A}(R \overset{S}{\times} S), t)$$

Find movements of accounts still open until yesterday

$$\pi_{\mathsf{no,mid,amount}}(\pi_{\mathsf{no}}\mathsf{account} \overset{S}{\bowtie} \mathsf{movement})$$

until-join ($\overset{U}{\bowtie}$)

$$eval(R \overset{U}{\bowtie} S, t) = eval(\sigma_{R.A=S.A}(R \overset{U}{\times} S), t)$$

Example of since-join

t	account			movement			$\pi_{no,mid,amount}(\pi_{no}\, account \overset{S}{\bowtie} movement)$		
	no	type	rate	mid	no	amount	no	mid	amount
0	100	'current'	NULL	0000	100	220.00			
	101	'deposit'	8.50	0001	102	2520.00			
	102	'deposit'	8.50						
1	100	'current'	NULL	0002	101	100.00	100	0000	220.00
	101	'deposit'	8.25	0003	104	1000.00	102	0001	2520.00
	103	'current'	NULL						
	104	'current'	NULL						
2	100	'current'	NULL	0004	104	-500.00	101	0002	100.00
	101	'deposit'	8.25	0005	105	500.00	104	0003	1000.00
	103	'current'	NULL	0006	100	100.23	100	0000	220.00
	104	'current'	NULL						
	105	'deposit'	8.50						

Quiz 3: Using $\overset{S}{\times}$

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

What is the result of $Eval(\pi_{loc_b.name}(\pi_{name}\, \sigma_{name=\text{'John'}}\, loc_a \overset{S}{\times} loc_b), 4)$?

A

name
Peter
John
Fred
Mary
Jim

B

name
Peter
Fred
Mary
Jim

C

name
Peter
John
Mary

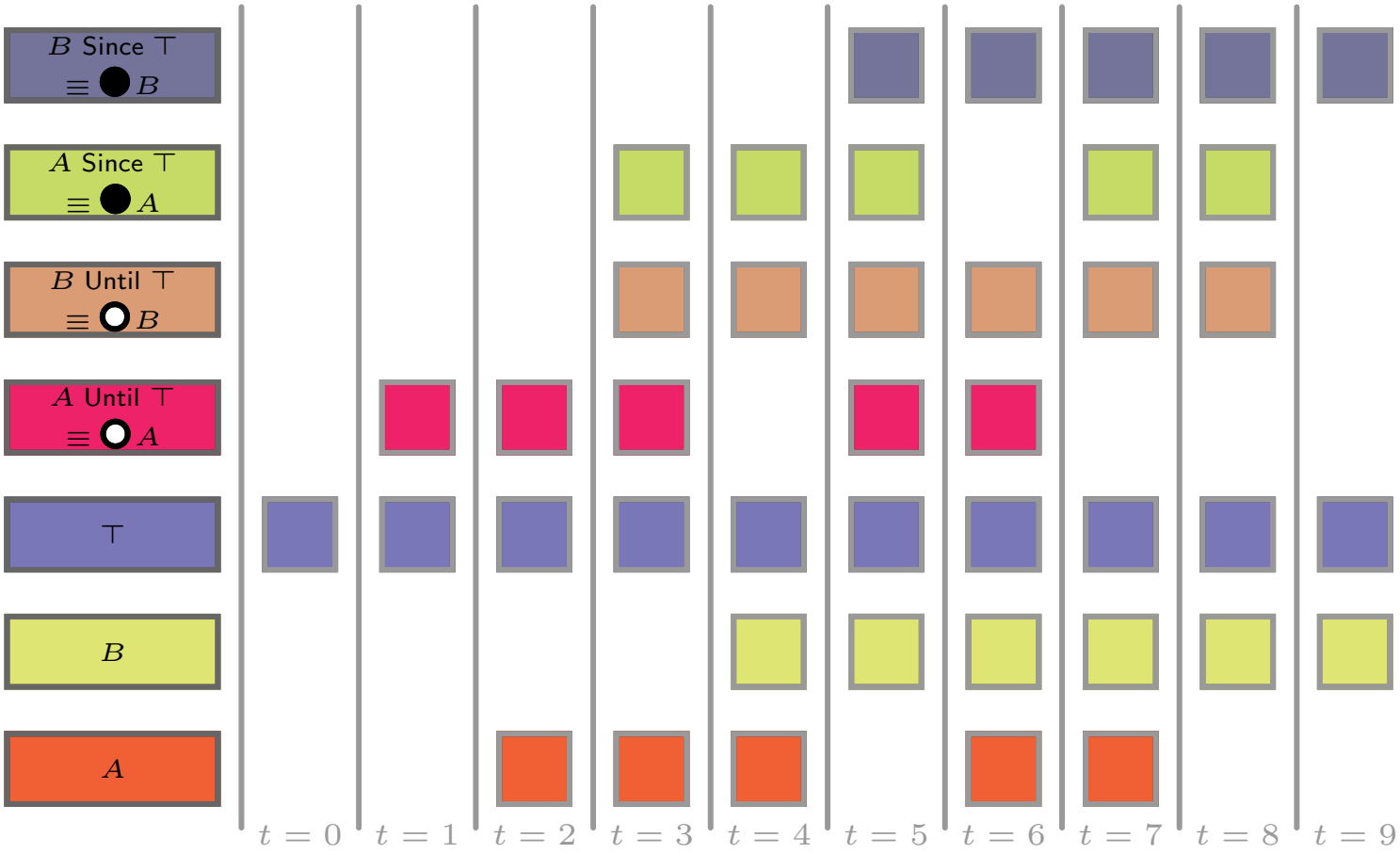
D

name
Peter

US-Logic: Derived Operators ○ and ●

- ○$A \equiv A$ Until $\top$
- Means 'is A true at the next time'

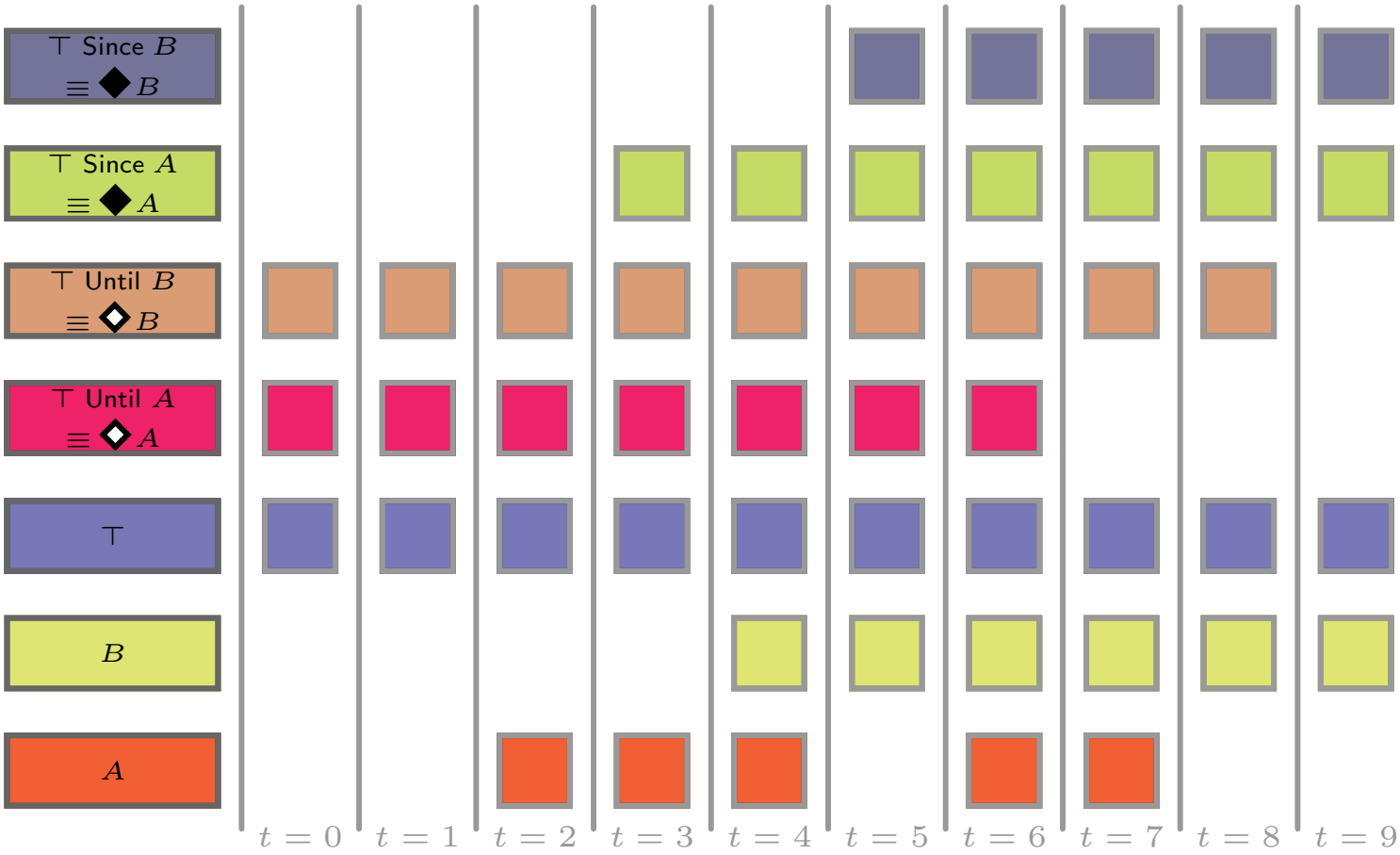
- ●$A \equiv A$ Since $\top$
- Means 'is A true at the previous time'



US-Logic: Derived Operators $\Diamond$ and $\blacklozenge$

- $\Diamond A \equiv \top$ Until A
- Means 'is A true at some point in the future'

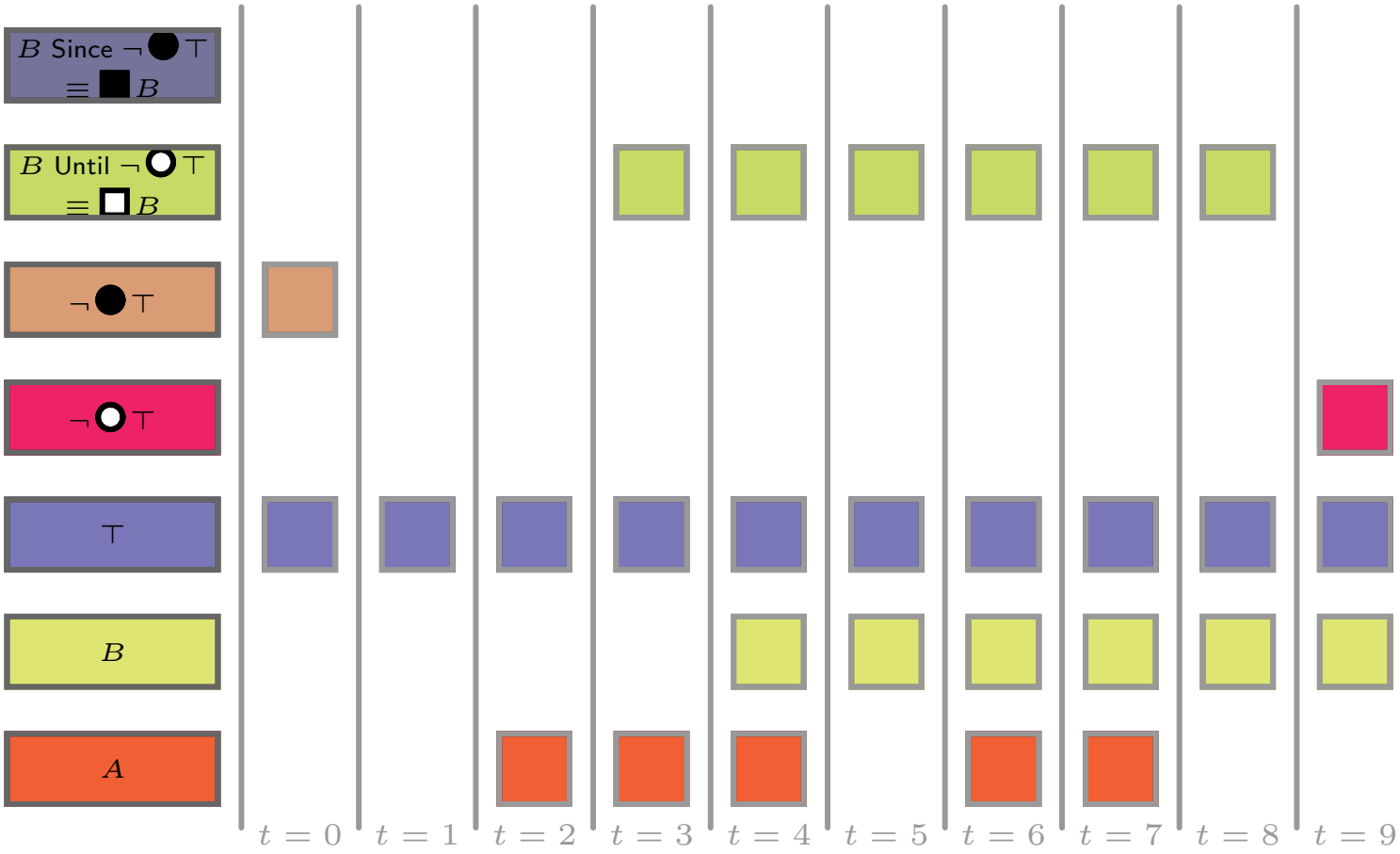
- $\blacklozenge A \equiv \top$ Since A
- Means 'is A true at some point in the past'



US-Logic: Derived Operators $\Box$ and $\blacksquare$

- $\Box A \equiv A$ Until $\neg \bigcirc \top$
- Means 'is A always true in the future'

- $\blacksquare A \equiv A$ Since $\neg \bullet \top$
- Means 'is A always true in the past'



Derived relational modal operators

temporal operator	name	equivalent using $\overset{S}{\times}$ and $\overset{U}{\times}$
$\bullet R$	**previous**	$R \overset{S}{\times} I$
$\blacklozenge R$	**sometime in past**	$I \overset{S}{\times} R$
$\blacksquare R$	**always in past**	$\pi_R(R \overset{S}{\times} \sigma_{time=0}\, time)$
$\circ R$	**next**	$R \overset{U}{\times} I$
$\lozenge R$	**sometime in future**	$I \overset{U}{\times} R$
$\square R$	**always in future**	$\pi_R(R \overset{U}{\times} \sigma_{time=n}\, time)$

Modal operator: Sometime in past ◆

t	account			◆account			◆π_{no}account
	no	type	rate	no	type	rate	no
0	100	'current'	NULL				
	101	'deposit'	8.50				
	102	'deposit'	8.50				
1	100	'current'	NULL	100	'current'	NULL	100
	101	'deposit'	8.25	101	'deposit'	8.50	101
	103	'current'	NULL	102	'deposit'	8.50	102
	104	'current'	NULL				
2	100	'current'	NULL	100	'current'	NULL	100
	101	'deposit'	8.25	101	'deposit'	8.25	101
	103	'current'	NULL	101	'deposit'	8.50	102
	104	'current'	NULL	102	'deposit'	8.50	103
	105	'deposit'	8.50	103	'current'	NULL	104
				104	'current'	NULL	

Modal operator: Previous ●

t	account			●account		
	no	type	rate	no	type	rate
0	100	'current'	NULL			
	101	'deposit'	8.50			
	102	'deposit'	8.50			
1	100	'current'	NULL	100	'current'	NULL
	101	'deposit'	8.25	101	'deposit'	8.50
	103	'current'	NULL	102	'deposit'	8.50
	104	'current'	NULL			
2	100	'current'	NULL	100	'current'	NULL
	101	'deposit'	8.25	101	'deposit'	8.25
	103	'current'	NULL	103	'current'	NULL
	104	'current'	NULL	104	'current'	NULL
	105	'deposit'	8.50			

Modal operator: Always in past ■

t	account			■account			■$\pi_{\text{no,type}}$account	
	no	type	rate	no	type	rate	no	type
0	100	'current'	NULL					
	101	'deposit'	8.50					
	102	'deposit'	8.50					
1	100	'current'	NULL	100	'current'	NULL	100	'current'
	101	'deposit'	8.25	101	'deposit'	8.50	101	'deposit'
	103	'current'	NULL	102	'deposit'	8.50	102	'deposit'
	104	'current'	NULL					
2	100	'current'	NULL	100	'current'	NULL	100	'current'
	101	'deposit'	8.25				101	'deposit'
	103	'current'	NULL					
	104	'current'	NULL					
	105	'deposit'	8.50					

Quiz 4: ■ modal logic operator

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

What is the result of $Eval(\blacksquare loc, 3)$?

A

name	town
Peter	London

B

name	town
Peter	London
Jim	Edinburgh

C

name	town
Peter	London
Jim	Edinburgh
Jim	Glasgow

D

name	town
Fred	Manchester
Peter	London
Jim	Edinburgh
John	London
Jim	Glasgow
Mary	Edinburgh

Quiz 5: ◆ modal logic operator

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

What is the result of $Eval(\blacklozenge loc, 3)$?

A

name	town
Peter	London

B

name	town
Peter	London
Jim	Edinburgh

C

name	town
Peter	London
Jim	Edinburgh
Jim	Glasgow

D

name	town
Fred	Manchester
Peter	London
Jim	Edinburgh
John	London
Jim	Glasgow
Mary	Edinburgh

Using the Derived Operators

Use ■ to find things always true

- *Find accounts that have always existed*
  ■ π_{no} account

Using the Derived Operators

Use ■ to find things always true

- *Find accounts that have always existed*
  ■π_{no} account

Use ◆ to search the past for data

- *Find accounts that have made a withdrawal in the past*
  ◆$\pi_{\text{no}}\ \sigma_{\text{amount}<0}$ movement
- *List the open accounts which have not made a withdrawal*
  π_{no} account $-$ ◆$\pi_{\text{no}}\ \sigma_{\text{amount}<0}$ movement
- *List accounts which have opened for the first time*
  π_{no} account $-$ ◆π_{no} account

Using the Derived Operators

Use ■ to find things always true

- *Find accounts that have always existed*
  $\blacksquare \pi_{\text{no}}\ \text{account}$

Use ◆ to search the past for data

- *Find accounts that have made a withdrawal in the past*
  $\blacklozenge \pi_{\text{no}}\ \sigma_{\text{amount}<0}\ \text{movement}$
- *List the open accounts which have not made a withdrawal*
  $\pi_{\text{no}}\ \text{account} - \blacklozenge \pi_{\text{no}}\ \sigma_{\text{amount}<0}\ \text{movement}$
- *List accounts which have opened for the first time*
  $\pi_{\text{no}}\ \text{account} - \blacklozenge \pi_{\text{no}}\ \text{account}$

Use ● to check what has just occurred

- *List accounts which have just opened or reopened*
  $\pi_{\text{no}}\ \text{account} - \bullet \pi_{\text{no}}\ \text{account}$

Worksheet: Derived TRA Operators

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

Worksheet: Derived TRA Operators

loc

name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc

name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc

name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc

name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

eval(◆loc, 3)

name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh
Jim	Edinburgh
Fred	Manchester

Worksheet: Derived TRA Operators

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

eval([redacted] π_{town} loc, 3)

town
London
Edinburgh

Worksheet: Derived TRA Operators

loc	
name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc	
name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc	
name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc	
name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

$\mathsf{eval}(\pi_{\mathsf{loc_a.name}}\, \sigma_{\mathsf{loc_a.town} \neq \mathsf{loc_b.town} \wedge \mathsf{loc_a.name} = \mathsf{loc_b.name}}\, \blacklozenge(\mathsf{loc_a} \times \blacklozenge \mathsf{loc_b}), 4)$

$\mathsf{loc_a.name}$
Jim
Mary
John
Fred

Worksheet: Derived TRA Operators

loc

name	town
Fred	Manchester
Peter	London
Jim	Edinburgh

$t = 0$

loc

name	town
Peter	London
Jim	Edinburgh

$t = 1$

loc

name	town
Peter	London
John	London
Jim	Glasgow
Mary	Edinburgh

$t = 2$

loc

name	town
Peter	London
John	Bristol
Fred	Glasgow
Mary	London

$t = 3$

pairs of towns that a person has transferred directly between

$$\pi_{\mathsf{loc_a.town},\mathsf{loc_b.town}}\ \sigma_{\mathsf{loc_a.town}\neq\mathsf{loc_b.town}\wedge\mathsf{loc_a.name}=\mathsf{loc_b.name}}\ \blacklozenge(\mathsf{loc_a}\times\blacklozenge\mathsf{loc_b})$$

people who have always worked in the same town

$$(\pi_{\mathsf{name}}\blacklozenge\mathsf{loc})-(\pi_{\mathsf{loc_a.name}}\ \sigma_{\mathsf{loc_a.town}\neq\mathsf{loc_b.town}\wedge\mathsf{loc_a.name}=\mathsf{loc_b.name}}\ \blacklozenge(\mathsf{loc_a}\times\blacklozenge\mathsf{loc_b}))$$

THE END

Revision

- Review worksheets, coursework and exercises in notes
- Exam similar to exams of previous years
- Content of the course is what was presented in the lectures

Going Further ...

- ISO
- Project (Object-Relational Mapping, Data and Knowledge Integration, Managing BigData)
- PhD