pre- κ expander

nurbin

Introduction

κ syntax

Kappa at DLah

DLab's current work Space-related simulations Timing control

lara da atau

pre-kappa expander for κ language

Héctor Urbina

October 17, 2011

κ syntax Kappa at

DLab's current work Space-related simulations Timing control pre-Kappa

- Introduction
 - What is κ
 - κ syntax
- 2 Kappa at DLab
 - DLab's current work
 - Space-related simulations
 - Timing control
 - pre-Kappa expander

pre- κ expander

hurbin

What is κ κ syntax

DLab's curred work Space-related

Space-related simulations Timing contro pre-Kappa expander κ is a formal language for defining agents as sets of sites.

Sites hold an internal state as well as a binding state

 κ also enables the expression of rules of interaction between agents.

These rules are executable, inducing a stochastic dynamics on a mixture of agents.

A κ model is a collection of rules (with rate constants) and an initial mixture of agents on which such rules begin to act.

pre- κ expander

hurbin

What is κ κ syntax Kappa at DLab

DLab's curren work Space-related simulations Timing contro pre-Kappa expander κ is a formal language for defining agents as sets of sites.

Sites hold an internal state as well as a binding state.

 κ also enables the expression of rules of interaction between agents.

These rules are executable, inducing a stochastic dynamics on a mixture of agents.

A κ model is a collection of rules (with rate constants) and an initial mixture of agents on which such rules begin to act.

pre- κ expander

hurbin

What is κ κ syntax Kappa at DLab DLab's curre

DLab's current work Space-related simulations Timing contro pre-Kappa expander κ is a formal language for defining agents as sets of sites.

Sites hold an internal state as well as a binding state.

 κ also enables the expression of rules of interaction between agents.

These rules are executable, inducing a stochastic dynamics on a mixture of agents.

A κ model is a collection of rules (with rate constants) and an initial mixture of agents on which such rules begin to act.

pre- κ expander

hurbin

What is κ κ syntax Kappa at DLab DLab's curre work

DLab

DLab's current
work

Space-related
simulations

Timing control
pre-Kappa

 κ is a formal language for defining agents as sets of sites.

Sites hold an internal state as well as a binding state.

 κ also enables the expression of rules of interaction between agents.

These rules are executable, inducing a stochastic dynamics on a mixture of agents.

A κ model is a collection of rules (with rate constants) and an initial mixture of agents on which such rules begin to act.

pre- κ expander

hurbin

What is κ κ syntax

Kappa at

DLab

DLab's curre
work

Space-related

 κ is a formal language for defining agents as sets of sites.

Sites hold an internal state as well as a binding state.

 κ also enables the expression of rules of interaction between agents.

These rules are executable, inducing a stochastic dynamics on a mixture of agents.

A κ model is a collection of rules (with rate constants) and an initial mixture of agents on which such rules begin to act.

κ Syntax short introduction

pre- κ expander

hurbin

Introductio
What is κ κ syntax

Nappa at DLab DLab's curre

DLab's curren work Space-related simulations Timing contro

Rule in English:

"Unphosphorilated Site1 of A binds to Site1 of B."

 κ Rule:

$$A(Site1^u), B(Site1) \rightarrow A(Site1^u!1), B(Site1!1)$$

- Agent Names : an identifier
- Agent Sites: an identifier
- Internal States : ~(value)
- Binding States : !(n), !_ or !?

κ Syntax short introduction

pre- κ expander

hurbin

Introduction What is κ syntax

Kappa at DLab

OLab's current work Space-related simulations Fiming contro ore-Kappa

Rule in English:

"Unphosphorilated Site1 of A binds to Site1 of B."

κ Rule:

$$\mathsf{A}(\mathsf{Site1^{\text{-}}u}), \mathsf{B}(\mathsf{Site1}) \to \mathsf{A}(\mathsf{Site1^{\text{-}}u!1}), \mathsf{B}(\mathsf{Site1!1})$$

- Agent Names: an identifier.
- Agent Sites: an identifier.
- Internal States : ~(value).
- Binding States : !\(n\), !_ or !?.

κ Syntax short introduction

pre- κ expander

hurbin

Introductio
What is κ κ syntax

Kappa at

DLab's current work Space-related simulations Timing control pre-Kappa Rule in English:

"Unphosphorilated Site1 of A binds to Site1 of B."

 κ Rule:

 $\mathsf{A}(\mathsf{Site1^{\text{-}}u}), \mathsf{B}(\mathsf{Site1}) \to \mathsf{A}(\mathsf{Site1^{\text{-}}u!1}), \mathsf{B}(\mathsf{Site1!1})$

- Agent Names : an identifier.
- Agent Sites : an identifier.
- Internal States : ~(value).
- Binding States : !⟨n⟩, !_ or !?.

Kappa file structure

%init: 1000 A,B 10000 C

%init:

 $pre-\kappa$ expander #### Signatures %agent: A(x,c) # Declaration of agent A %agent: B(x) # Declaration of B %agent: C(x1~u~p,x2~u~p) # Declaration of C with 2 modifiable sites 5 #### Rules 6 'a.b' A(x).B(x) -> A(x!1).B(x!1) @ 'on_rate' #A binds B α 'a..b' $A(x!1),B(x!1) \rightarrow A(x),B(x)$ @ 'off_rate' #AB dissociation 'ab.c' A(x!_,c),C(x1~u) ->A(x!_,c!2),C(x1~u!2) @ 'on_rate' #AB binds C 'mod x1' C(x1~u!1).A(c!1) ->C(x1~p).A(c) @ 'mod_rate' #AB modifies x1 'a.c' A(x,c),C(x1~p,x2~u) -> A(x,c!1),C(x1~p,x2~u!1) @ 'on_rate' #A binds C on x2 'mod x2' A(x,c!1),C(x1~p,x2~u!1) -> A(x,c),C(x1~p,x2~p) @ 'mod_rate' #A modifies x2 #### Variables %var: 'on_rate' 1.0E-4 # per molecule per second %var: 'off_rate' 0.1 # per second %var: 'mod_rate' 1 # per second Mobs: 'AB' A(x!x.B) %obs: 'Cuu' C(x1~u.x2~u) %obs: 'Cpu' C(x1~p.x2~u) %obs: 'Cpp' C(x1~p.x2~p) #### Initial conditions

 $pre-\kappa$ expander

DI ah's current

DLab members study complex dynamical systems.

pre- κ expander

hurbin

Introduction What is κ

Kappa a

DLab's current

Space-related simulations

Timing cont

DLab members study complex dynamical systems.

Currently, Cesar Ravello is modeling muscle contration and Felipe Nuñez is simulating massive responses to zombie attacks on human populations, whereas Ricardo Honorato is adapting Model Checking techniques to be used with systems expressed in κ language.

Without intervening the κ language, we have reached some interesting levels of abstraction!

 $pre-\kappa$ expander

DI ah's current

DLab members study complex dynamical systems.

Currently, Cesar Ravello is modeling muscle contration and Felipe Nuñez is simulating massive responses to zombie attacks on human populations, whereas Ricardo Honorato is adapting Model Checking techniques to be used with systems expressed in κ language.

Without intervening the κ language, we have reached some interesting levels of abstraction!

pre- κ expander

hurbin

Introductio

What is *i* κ syntax

Kappa at DLab

DLab's current work

Space-related simulations Timing control pre-Kappa

- Space-related simulations.
 - Compartmentalization.
 - Diffusion events.
- Timing control.
 - Polymer-driven rules to manipulate latency.

pre- κ expander

hurbin

Introduction

What is *i* κ syntax

Kappa at DLab

DLab's current

Space-related simulations Fiming control ore-Kappa

- Space-related simulations.
 - Compartmentalization.
 - Diffusion events.
- Timing control.
 - Polymer-driven rules to manipulate latency.

pre- κ expander

hurbin

Introduction

κ syntax

DLab's current

work Space-related simulations Timing contro • Space-related simulations.

- Compartmentalization.
- Diffusion events.
- Timing control.
 - Polymer-driven rules to manipulate latency.

pre- κ expander

nurbin

Introduction

What is κ κ syntax

Kappa at DLab

work

Space-related simulations

simulations Timing contro

#Signatures

%agent: $A(x,c,loc^ij^k)$ %agent: $B(x,loc^ij^k)$

#Rules

#A binds E

$$\begin{split} &\mathsf{A}(\mathsf{x},\mathsf{loc}\tilde{\ \ i}),\mathsf{B}(\mathsf{x},\mathsf{loc}\tilde{\ \ i}) \to \mathsf{A}(\mathsf{x}!1,\mathsf{loc}\tilde{\ \ i}),\mathsf{B}(\mathsf{x}!1,\mathsf{loc}\tilde{\ \ i}) \ @ \ \mathsf{'on_rate'} \\ &\mathsf{A}(\mathsf{x},\mathsf{loc}\tilde{\ \ j}),\mathsf{B}(\mathsf{x},\mathsf{loc}\tilde{\ \ j}) \to \mathsf{A}(\mathsf{x}!1,\mathsf{loc}\tilde{\ \ j}),\mathsf{B}(\mathsf{x}!1,\mathsf{loc}\tilde{\ \ j}) \ @ \ \mathsf{'on_rate'} \\ &\mathsf{A}(\mathsf{x},\mathsf{loc}\tilde{\ \ k}),\mathsf{B}(\mathsf{x},\mathsf{loc}\tilde{\ \ k}) \to \mathsf{A}(\mathsf{x}!1,\mathsf{loc}\tilde{\ \ k}),\mathsf{B}(\mathsf{x}!1,\mathsf{loc}\tilde{\ \ k}) \ @ \ \mathsf{'on_rate'} \end{split}$$

 $pre-\kappa$ expander

Space-related simulations

```
#Signatures
```

%agent: A(x,c,loc~i~j~k) %agent: B(x,loc~i~j~k)

#Rules

#A binds B

 $A(x,loc^{-}i),B(x,loc^{-}i) \rightarrow A(x!1,loc^{-}i),B(x!1,loc^{-}i)$ @ 'on_rate' $A(x,loc^{-}j),B(x,loc^{-}j) \rightarrow A(x!1,loc^{-}j),B(x!1,loc^{-}j)$ @ 'on_rate' $A(x,loc^{k}),B(x,loc^{k}) \rightarrow A(x!1,loc^{k}),B(x!1,loc^{k})$ @ 'on_rate'

pre- κ expander

hurbin

Introduction

κ syntax

DLab's curre

Space-related simulations

simulations Timing contro #Locations i, j and k have different volumen/area!

#Signatures

%agent: A(x,c,loc~i~j~k)

%agent: B(x,loc"i"j"k)

#Rules

#A binds B

 $\mathsf{A}(\mathsf{x}.\mathsf{loc}\tilde{\ }i), \mathsf{B}(\mathsf{x}.\mathsf{loc}\tilde{\ }i) \to \mathsf{A}(\mathsf{x}!1,\mathsf{loc}\tilde{\ }i), \mathsf{B}(\mathsf{x}!1,\mathsf{loc}\tilde{\ }i) \ @ \ \mathsf{'on_rate_loc}(i)\mathsf{'}$

 $\mathsf{A}(\mathsf{x},\mathsf{loc}\tilde{\ }\mathsf{j}),\mathsf{B}(\mathsf{x},\mathsf{loc}\tilde{\ }\mathsf{j})\to\mathsf{A}(\mathsf{x}!1,\mathsf{loc}\tilde{\ }\mathsf{j}),\mathsf{B}(\mathsf{X}!1,\mathsf{loc}\tilde{\ }\mathsf{j})\; @\; \mathsf{'on_rate_loc}(\mathsf{j})$

 $A(x,loc^k),B(x,loc^k) \rightarrow A(x!1,loc^k),B(X!1,loc^k) @ 'on_rate_loc(k)$

#AB dissociation

 $A(x!1,loc^{\sim}i),B(x!1,loc^{\sim}i) \rightarrow A(x,loc^{\sim}i),B(x,loc^{\sim}i)$ @ 'off_rate'

 $A(x!1,loc^{-}j),B(x!1,loc^{-}j) \rightarrow A(x,loc^{-}j),B(x,loc^{-}j)$ @ 'off_rate'

 $A(x!1,loc^k),B(x!1,loc^k) \rightarrow A(x,loc^k),B(x,loc^k) @ 'off_rate'$

```
pre-\kappa
expander
```

Space-related

simulations

```
#Locations i, j and k have different volumen/area!
#Signatures
%agent: A(x,c,loc~i~i~k)
%agent: B(x,loc~i~j~k)
#Rules
#A binds B
```

 $A(x,loc^{-i}),B(x,loc^{-i}) \rightarrow A(x!1,loc^{-i}),B(x!1,loc^{-i})$ @ 'on_rate_loc(i)' $A(x,loc^*j),B(x,loc^*j) \rightarrow A(x!1,loc^*j),B(X!1,loc^*j)$ @ 'on_rate_loc(j)' $A(x,loc^k),B(x,loc^k) \rightarrow A(x!1,loc^k),B(X!1,loc^k) @ 'on_rate_loc(k)'$

```
pre-\kappa expander
```

hurbin

Introduction What is κ

κ syntax Kappa at

DLab's currer work

Space-related simulations

simulations
Timing contro

```
#Locations i, j and k have different volumen/area!
#Signatures
%agent: A(x,c,loc~i~i~k)
%agent: B(x,loc^i^j^k)
#Rules
#A binds B
A(x,loc^{-i}),B(x,loc^{-i}) \rightarrow A(x!1,loc^{-i}),B(x!1,loc^{-i}) @ 'on_rate_loc(i)'
A(x,loc^*i),B(x,loc^*i) \rightarrow A(x!1,loc^*i),B(X!1,loc^*i) @ 'on_rate_loc(i)'
A(x,loc^k),B(x,loc^k) \rightarrow A(x!1,loc^k),B(X!1,loc^k) @ 'on_rate_loc(k)'
#AB dissociation
A(x!1,loc^{-}i),B(x!1,loc^{-}i) \rightarrow A(x,loc^{-}i),B(x,loc^{-}i) @ 'off_rate'
A(x!1,loc^*i),B(x!1,loc^*j) \rightarrow A(x,loc^*j),B(x,loc^*j) @ 'off_rate'
```

 $A(x!1,loc^k),B(x!1,loc^k) \rightarrow A(x,loc^k),B(x,loc^k)$ @ 'off_rate'

Diffusion events

 $pre-\kappa$ expander

Space-related

simulations

#Signatures

%agent: A(x,c,loc~i~j~k)

%agent: $B(x,loc^i^j^k)$

%agent: T(s,org~i~j~k,dst~i~j~k)

Diffusion events

pre- κ expander

hurbin

Introductio

Kappa at

DLab's curre

Space-related simulations

simulations
Timing contro
pre-Kappa
expander

#Rules

#A diffusions

$$\begin{split} &A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}j) \rightarrow A(loc^{\sim}j,x,c),T(org^{\sim}i,dst^{\sim}j) \ @ \ 'Adiff_ij' \\ &A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}k) \rightarrow A(loc^{\sim}k,x,c),T(org^{\sim}i,dst^{\sim}k) \ @ \ 'Adiff_ik' \\ &A(loc^{\sim}j,x,c),T(org^{\sim}j,dst^{\sim}i) \rightarrow A(loc^{\sim}i,x,c),T(org^{\sim}j,dst^{\sim}i) \ @ \ 'Adiff_ji' \\ &A(loc^{\sim}j,x,c),T(org^{\sim}j,dst^{\sim}k) \rightarrow A(loc^{\sim}k,x,c),T(org^{\sim}j,dst^{\sim}k) \ @ \ 'Adiff_jk' \\ &A(loc^{\sim}k,x,c),T(org^{\sim}k,dst^{\sim}i) \rightarrow A(loc^{\sim}i,x,c),T(org^{\sim}k,dst^{\sim}i) \ @ \ 'Adiff_ki' \\ &A(loc^{\sim}k,x,c),T(org^{\sim}k,dst^{\sim}j) \rightarrow A(loc^{\sim}j,x,c),T(org^{\sim}k,dst^{\sim}j) \ @ \ 'Adiff_kj' \\ \end{split}$$

 $pre-\kappa$ expander

Timing control

#Signatures

%agent: S(x)%agent: Z()

%agent: V(p,n)

```
pre-\kappa expander
```

.....

Introduction

What is κ

Kappa at

DLab's currei

work Space-related

simulations

Timing control pre-Kappa

#Signatures

%agent: S(x) %agent: Z()

%agent: V(p,n)

#Rules

'Infection' $Z(),S(x) \rightarrow Z(),S(x!1),V(p!1,n)$ @ 'infection_rate' 'Polymerization' $V(n) \rightarrow V(n!1),V(p!1,n)$ @ 'polymer_rate' 'Expression' S(x!1),V(p!1,n!2),V(p!2,n!3),V(p!3,n!4), V(p!4,n!5),V(p!5,n!6),V(p!6,n!7),V(p!7,n!8),V(p!8,n!9), $V(p!9,n!10),V(p!10,n) \rightarrow Z()$ @ [inf]

```
pre-\kappa expander
```

nurbina

Introduction

What is κ κ syntax

Kappa at

DLab's curre

work

Space-related simulations

Timing control

```
#Signatures
```

%agent: S(x)

%agent: Z()

%agent: V(p,n)

#Rules

'Infection' $Z(),S(x) \rightarrow Z(),S(x!1),V(p!1,n)$ @ 'infection_rate' 'Polymerization' $V(n) \rightarrow V(n!1),V(p!1,n)$ @ 'polymer_rate'

'Expression' $S(x|1),V(p|1,n|2),V(p|2,n|3),V(p|3,n|4), V(p|4,n|5),V(p|5,n|6),V(p|6,n|7),V(p|7,n|8),V(p|8,n|9), V(p|9,n|10),V(p|10,n) <math>\rightarrow 7()$ @ [inf]

```
pre-\kappa expander
```

hurbin

Introductio What is κ

κ syntax

Kappa at DLab

DLab's currei work

Space-related simulations

Timing control

```
#Signatures
```

%agent: S(x)

%agent: Z()

%agent: V(p,n)

#Rules

'Infection' Z(),S(x) \rightarrow Z(),S(x!1),V(p!1,n) @ 'infection_rate' 'Polymerization' V(n) \rightarrow V(n!1),V(p!1,n) @ 'polymer_rate' 'Expression' S(x!1),V(p!1,n!2),V(p!2,n!3),V(p!3,n!4), \ V(p!4,n!5),V(p!5,n!6),V(p!6,n!7),V(p!7,n!8),V(p!8,n!9), \ V(p!9,n!10),V(p!10,n) \rightarrow Z() @ [inf]

pre-Kappa expander

pre- κ expander

hurbin

Introduction

κ syntax

DLab's cur

expander

work
Space-related
simulations
Timing control
pre-Kappa

A Python (V2) script that takes as input a (built in-house) pre- κ file and outputs a kappa file which can subsequently be used with KaSim.

This is done using Lexer & Parser techniques, available in Python through ply library.

It facilitates κ abstraction while reducing error-proneness

pre-Kappa expander

pre- κ expander

hurbin

Introductio

Kappa at

DLab DLab's curi

expander

work
Space-related
simulations
Timing control
pre-Kappa

A Python (V2) script that takes as input a (built in-house) pre- κ file and outputs a kappa file which can subsequently be used with KaSim.

This is done using Lexer & Parser techniques, available in Python through ply library.

It facilitates κ abstraction while reducing error-proneness

pre-Kappa expander

pre- κ expander

hurbin

Introductio What is κ

Kappa at

expander

DLab's current work Space-related simulations Timing control pre-Kappa A Python (V2) script that takes as input a (built in-house) pre- κ file and outputs a kappa file which can subsequently be used with KaSim.

This is done using Lexer & Parser techniques, available in Python through ply library.

It facilitates κ abstraction while reducing error-proneness.

```
pre-\kappa expander
```

hurbin

Introduction

What is ε κ syntax

Kappa at DLab

DLab's current work Space-related

simulations
Timing control

Timing contro pre-Kappa expander

```
#Locations
```

%loc: i 100 10000 1201 %loc: j 1000 20000 3902

%loc: k 500 30000 2890 #Location list %loc!: all i i k

%locl: all i j k #Signatures

%expand-agent: all A(x,c)%expand-agent: all B(x)

gives:

%agent: A(x,c,loc~i~j~k) %agent: B(x loc~i~i~k)

```
pre-\kappa expander
```

hurbin

Introduction
What is κ

Kappa at DLab DLab's currer

work
Space-related
simulations
Timing control

Timing contro pre-Kappa expander

```
#Locations
```

%loc: i 100 10000 1201 %loc: j 1000 20000 3902 %loc: k 500 30000 2890

#Location list %locl: all i j k

%expand-agent: all A(x,c)%expand-agent: all B(x)

gives:

%agent: A(x,c,loc~i~j~k) %agent: B(x,loc~i~j~k)

```
pre-\kappa
expander
```

pre-Kappa expander

```
#Locations
```

%loc: i 100 10000 1201 %loc: i 1000 20000 3902 %loc: k 500 30000 2890

#Location list %locl: all i j k **#Signatures**

%expand-agent: all A(x,c)%expand-agent: all B(x)

```
pre-\kappa expander
```

hurbin

Introduction What is κ

Kappa at DLab

DLab's current work Space-related

simulations
Timing control
pre-Kappa

expander

```
#Locations
```

%loc: i 100 10000 1201 %loc: j 1000 20000 3902 %loc: k 500 30000 2890

#Location list %locl: all i j k #Signatures

%expand-agent: all A(x,c) %expand-agent: all B(x)

gives:

%agent: A(x,c,loc~i~j~k)
%agent: B(x,loc~i~j~k)

```
pre-\kappa
              #Locations
 expander
              %loc: i 100 10000 1201
              %loc: j 1000 20000 3902
              %loc: k 500 30000 2890
              #Location list
             %locl: all i j k
              #Initializations
              %expand-init: all %loc[0] A(x,c)
              %expand-init: all %loc[1] B(x)
pre-Kappa
expander
```

```
pre-\kappa
             #Locations
 expander
             %loc: i 100 10000 1201
             %loc: j 1000 20000 3902
             %loc: k 500 30000 2890
             #Location list
             %locl: all i j k
             #Initializations
             %expand-init: all \%loc[0] A(x,c)
             %expand-init: all %loc[1] B(x)
pre-Kappa
expander
             gives:
             %init: 100 A(x,c,loc~i)
             %init: 1000 A(x,c,loc~j)
             %init: 500 A(x,c,loc^k)
```

```
pre-\kappa
             #Locations
 expander
             %loc: i 100 10000 1201
             %loc: j 1000 20000 3902
             %loc: k 500 30000 2890
             #Location list
             %locl: all i j k
             #Initializations
             %expand-init: all %loc[0] A(x,c)
             %expand-init: all %loc[1] B(x)
pre-Kappa
expander
             gives:
             %init: 100 A(x,c,loc~i)
             %init: 1000 A(x,c,loc~j)
             %init: 500 A(x,c,loc^k)
             %init: 10000 B(x,loc~i)
             %init: 20000 B(x,loc~j)
             %init: 30000 B(x,loc^{k})
```

 $pre-\kappa$ expander

pre-Kappa expander

A bimolecular stochastic rate constant γ , expressed in s^{-1} molecule⁻¹, is related to its deterministic counterpart k, expressed in $s^{-1}M^{-1}$ as

$$\gamma = \frac{k}{AV},\tag{1}$$

where A is Avogadro's number.

Krivine et. al. Programs as models: Execution. Unpublised work.

```
pre-\kappa expander
```

hurbin

Introduction What is κ

Kappa at

DLab

work
Space-related

simulations Timing control

pre-Kappa expander

```
#Locations
```

%loc: i 100 10000 1.201 %loc: j 1000 20000 3.902 %loc: k 500 30000 2.89

#Location list %locl: all i j k #A binds B

%expand-rule: all A(x),B(x) \rightarrow A(x!1),B(x!1) @ %loc[2]

gives

$$\begin{split} &A(x,loc^{-}i),B(x,loc^{-}i) \to A(x!1,loc^{-}i),B(x!1,loc^{-}i) @ 1.201 \\ &A(x,loc^{-}j),B(x,loc^{-}j) \to A(x!1,loc^{-}j),B(x!1,loc^{-}j) @ 3.901 \\ &A(x,loc^{-}k),B(x,loc^{-}k) \to A(x!1,loc^{-}k),B(x!1,loc^{-}k) @ 2.81 \\ &A(x,loc^{-}k),B(x,loc^{-}k) \to A(x!1,loc^{-}k),B(x!1,loc^{-}k) & 2.81 \\ &A(x,loc^{-}k),B(x,loc^{-}k) \to A(x!1,loc^{-}k) \\ &A(x,loc^{-}k),B(x,loc^{-}k) & 2.81 \\ &A(x,loc^{-}k),B(x,lo$$

```
pre-\kappa expander
```

hurbin

```
Introduction
What is \kappa
```

Kappa at

DLab's curren work

Space-related simulations

Timing cont pre-Kappa expander

```
#Locations %loc: i 100 10000 1.201 %loc: j 1000 20000 3.902 %loc: k 500 30000 2.89 #Location list %locl: all i j k #A binds B %expand-rule: all A(x),B(x) \rightarrow A(x!1),B(x!1) @ %loc[2] gives: A(x,loc^*i),B(x,loc^*i) \rightarrow A(x!1,loc^*i),B(x!1,loc^*i) @ 1.201 A(x,loc^*j),B(x,loc^*j) \rightarrow A(x!1,loc^*j),B(x!1,loc^*j) @ 3.902 A(x,loc^*k),B(x,loc^*k) \rightarrow A(x!1,loc^*k),B(x!1,loc^*k) @ 2.89
```

pre- κ expander

hurbin

Introduction What is κ

K syntax Kappa at

DLab's currer

work

simulations Timing contro

pre-Kappa expander

#Location matrices

%locm:

TM	i	j	k
i	0	0.5	1.5
j	2.0	0	1.8
k	1.0	1.1	0

#A diffusions

%expand-rule: TM A(x,c),T() ightarrow A(%,x,c),T() @ %cell

$$\begin{array}{l} A(loc^*i,x,c),T(org^*i,dst^*j) \rightarrow A(loc^*j,x,c),T(org^*i,dst^*j) @ 0.5 \\ A(loc^*i,x,c),T(org^*i,dst^*k) \rightarrow A(loc^*k,x,c),T(org^*i,dst^*k) @ 1.5 \\ A(loc^*j,x,c),T(org^*j,dst^*i) \rightarrow A(loc^*i,x,c),T(org^*j,dst^*i) @ 2.0 \\ A(loc^*j,x,c),T(org^*j,dst^*k) \rightarrow A(loc^*k,x,c),T(org^*j,dst^*k) @ 1.6 \\ A(loc^*k,x,c),T(org^*k,dst^*i) \rightarrow A(loc^*i,x,c),T(org^*k,dst^*i) @ 1.0 \\ A(loc^*k,x,c),T(org^*k,dst^*i) \rightarrow A(loc^*i,x,c),T(org^*k,dst^*i) @ 1.1 \\ A(loc^*k,x,c),T(org^*k,dst^*i) \rightarrow A(loc^*k,x,c),T(org^*k,dst^*i) @ 1.1 \\ A(loc^*k,x,c),T(org^*k,dst^*i) & A(loc^*k,x$$

pre- κ expander

hurbin

Introduction What is κ

κ syntax Kappa at

DLab's curren

work

space-related simulations Timing contr

pre-Kappa expander

```
#Location matrices
```

%locm:

TM	i	j	k
i	0	0.5	1.5
j	2.0	0	1.8
k	1.0	1.1	0

#A diffusions

%expand-rule: TM A(x,c),T() \rightarrow A(%,x,c),T() @ %cell

$$A(loc^*i,x,c),T(org^*i,dst^*j) \rightarrow A(loc^*j,x,c),T(org^*i,dst^*j) @ 0.5 \\ A(loc^*i,x,c),T(org^*i,dst^*k) \rightarrow A(loc^*k,x,c),T(org^*i,dst^*k) @ 1.5 \\ A(loc^*j,x,c),T(org^*j,dst^*i) \rightarrow A(loc^*i,x,c),T(org^*j,dst^*i) @ 2.0 \\ A(loc^*j,x,c),T(org^*j,dst^*k) \rightarrow A(loc^*k,x,c),T(org^*j,dst^*k) @ 1.8 \\ A(loc^*k,x,c),T(org^*k,dst^*i) \rightarrow A(loc^*i,x,c),T(org^*k,dst^*i) @ 1.0 \\ A(loc^*k,x,c),T(org^*k,dst^*i) \rightarrow A(loc^*i,x,c),T(org^*k,dst^*i) @ 1.1 \\ A(loc^*k,x,c),T(org^*k,dst^*i) \rightarrow A(loc^*k,x,c),T(org^*k,dst^*i) @ 1.1 \\ A(loc^*k,x,c),T(org^*k,dst^*i) & A(loc^*k,$$

pre- κ expander

hurbin

Introductio
What is κ

Kappa at

DLab's curren

work Space-related

space-related simulations Fiming contro

pre-Kappa expander

```
#Location matrices
```

%locm:

TM	i	j	k
i	0	0.5	1.5
j	2.0	0	1.8
k	1.0	1.1	0

#A diffusions

%expand-rule: TM A(x,c),T() \rightarrow A(%,x,c),T() @ %cell

$$A(loc~i,x,c),T(org~i,dst~j) \rightarrow A(loc~j,x,c),T(org~i,dst~j) @ 0.5$$

$$A(loc^{\sim}j,x,c),T(org^{\sim}j,dst^{\sim}i) \rightarrow A(loc^{\sim}i,x,c),T(org^{\sim}j,dst^{\sim}i) @ 2.0$$

$$A(loc^{\tilde{}}j,x,c),T(org^{\tilde{}}j,dst^{\tilde{}}k) \rightarrow A(loc^{\tilde{}}k,x,c),T(org^{\tilde{}}j,dst^{\tilde{}}k) \ @ \ 1.8$$

$$A(loc^{\kappa},x,c),T(org^{\kappa},dst^{\tilde{\kappa}}) \rightarrow A(loc^{\tilde{\kappa}},x,c),T(org^{\kappa},dst^{\tilde{\kappa}}) \otimes 1.0$$

$$A(loc^*k,x,c),T(org^*k,dst^*j) \rightarrow A(loc^*j,x,c),T(org^*k,dst^*j) @ 1.1$$

pre- κ expander

hurbin

What is κ

Kappa at

DLab's currer

work Space-related

space-related simulations Timing contr

Timing contr pre-Kappa expander

```
#Location matrices
```

%locm:

TM	i	j	k
i	0	0.5	1.5
j	2.0	0	1.8
k	1.0	1.1	0

#A diffusions

%expand-rule: TM A(x,c),T() \rightarrow A(%,x,c),T() @ %cell

$$A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}j) \rightarrow A(loc^{\sim}j,x,c),T(org^{\sim}i,dst^{\sim}j) \ @ \ 0.5$$

$$A(loc^*i,x,c),T(org^*i,dst^*k) \rightarrow A(loc^*k,x,c),T(org^*i,dst^*k) \ @ \ 1.5$$

$$A(loc^*j,x,c),T(org^*j,dst^*i) \rightarrow A(loc^*i,x,c),T(org^*j,dst^*i) \ @ \ 2.0$$

$$A(loc\ \tilde{\ }j,x,c),T(org\ \tilde{\ }j,dst\ \tilde{\ }k)\rightarrow A(loc\ \tilde{\ }k,x,c),T(org\ \tilde{\ }j,dst\ \tilde{\ }k)\ @\ 1.8$$

$$A(loc^{\sim}k,x,c),T(org^{\sim}k,dst^{\sim}i) \rightarrow A(loc^{\sim}i,x,c),T(org^{\sim}k,dst^{\sim}i) \otimes 1.0$$

$$A(loc^*k,x,c),T(org^*k,dst^*j) \rightarrow A(loc^*j,x,c),T(org^*k,dst^*j) @ 1.1$$

```
pre-\kappa expander
```

hurbin

Introductio
What is κ

Kappa at

DLab's curren

Space-related simulations

simulations
Timing contro
pre-Kappa
expander

#Location matrices

%locm:

```
TM i j k i 0 0.5 1.5 j 2.0 0 1.8 k 1.0 1.1 0
```

#A diffusions

%expand-rule: TM $A(x,c),T() \rightarrow A(\%,x,c),T()$ @ %cell

$$\begin{array}{l} A(\text{loc}^{\sim}i,x,c), T(\text{org}^{\sim}i,\text{dst}^{\sim}j) \rightarrow A(\text{loc}^{\sim}j,x,c), T(\text{org}^{\sim}i,\text{dst}^{\sim}j) @ 0.5 \\ A(\text{loc}^{\sim}i,x,c), T(\text{org}^{\sim}i,\text{dst}^{\sim}k) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}i,\text{dst}^{\sim}k) @ 1.5 \\ A(\text{loc}^{\sim}j,x,c), T(\text{org}^{\sim}j,\text{dst}^{\sim}i) \rightarrow A(\text{loc}^{\sim}i,x,c), T(\text{org}^{\sim}j,\text{dst}^{\sim}i) @ 2.0 \\ A(\text{loc}^{\sim}j,x,c), T(\text{org}^{\sim}j,\text{dst}^{\sim}k) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}j,\text{dst}^{\sim}k) @ 1.8 \\ A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}k,\text{dst}^{\sim}i) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}k,\text{dst}^{\sim}i) @ 1.0 \\ A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}k,\text{dst}^{\sim}i) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}k,\text{dst}^{\sim}i) @ 1.0 \\ A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}k,\text{dst}^{\sim}i) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{org}^{\sim}k,\text{dst}^{\sim}i) @ 1.0 \\ A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,\text{dst}^{\sim}i) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,\text{dst}^{\sim}i) @ 1.0 \\ A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,\text{dst}^{\sim}i) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,\text{dst}^{\sim}i) @ 1.0 \\ A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,\text{dst}^{\sim}k) \rightarrow A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,\text{dst}^{\sim}k) & A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,\text{dst}^{\sim}k) & A(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,x,c), T(\text{loc}^{\sim}k,x,c),$$

```
pre-\kappa expander
```

hurbin

Introductio

Kappa at DLab

DLab's curren

work Space-related simulations

simulations Timing contro pre-Kappa expander

#Location matrices

%locm:

TM	i	j	k
i	0	0.5	1.5
j	2.0	0	1.8
k	1.0	1.1	0

#A diffusions

```
%expand-rule: TM A(x,c),T() \rightarrow A(\%,x,c),T() @ %cell
```

$$A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}j) \rightarrow A(loc^{\sim}j,x,c),T(org^{\sim}i,dst^{\sim}j) @ 0.5$$

$$A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}k) \rightarrow A(loc^{\sim}k,x,c),T(org^{\sim}i,dst^{\sim}k) @ 1.5$$

$$A(loc^{\sim}j,x,c),T(org^{\sim}j,dst^{\sim}i) \rightarrow A(loc^{\sim}i,x,c),T(org^{\sim}j,dst^{\sim}i) @ 2.0$$

$$A(loc^{\sim}j,x,c),T(org^{\sim}j,dst^{\sim}k) \rightarrow A(loc^{\sim}k,x,c),T(org^{\sim}j,dst^{\sim}k) @ 1.8$$

$$A(loc^{\kappa},x,c),T(org^{\kappa},dst^{\kappa}) \rightarrow A(loc^{\kappa},x,c),T(org^{\kappa},dst^{\kappa}) \otimes 1.1$$

```
pre-\kappa
expander
```

pre-Kappa expander

#Location matrices

%locm:

TM	i	j	k
i	0	0.5	1.5
j	2.0	0	1.8
k	1.0	1.1	0

#A diffusions

%expand-rule: TM $A(x,c),T() \rightarrow A(\%,x,c),T()$ @ %cell

$$A(loc\~i,x,c),T(org\~i,dst\~j) \rightarrow A(loc\~j,x,c),T(org\~i,dst\~j) \ @ \ 0.5$$

$$A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}k) \rightarrow A(loc^{\sim}k,x,c),T(org^{\sim}i,dst^{\sim}k) \ @ \ 1.5$$

$$A(loc~j,x,c),T(org~j,dst~i) \rightarrow A(loc~i,x,c),T(org~j,dst~i) @ 2.0$$

$$A(loc^{\tilde{}}j,x,c),T(org^{\tilde{}}j,dst^{\tilde{}}k) \rightarrow A(loc^{\tilde{}}k,x,c),T(org^{\tilde{}}j,dst^{\tilde{}}k) \otimes 1.8$$

$$A(loc^{k},x,c),T(org^{k},dst^{i}) \rightarrow A(loc^{i},x,c),T(org^{k},dst^{i}) @ 1.0$$

```
pre-\kappa
expander
```

pre-Kappa expander

#Location matrices

%locm:

TM	i	j	k
i	0	0.5	1.5
j	2.0	0	1.8
k	1.0	1.1	0

#A diffusions

%expand-rule: TM $A(x,c),T() \rightarrow A(\%,x,c),T()$ @ %cell

$$\begin{split} &A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}j) \rightarrow A(loc^{\sim}j,x,c),T(org^{\sim}i,dst^{\sim}j) \ @ \ 0.5 \\ &A(loc^{\sim}i,x,c),T(org^{\sim}i,dst^{\sim}k) \rightarrow A(loc^{\sim}k,x,c),T(org^{\sim}i,dst^{\sim}k) \ @ \ 1.5 \end{split}$$

$$\Delta(\log^2 i \times c) T(\arg^2 i \det^2 i) \rightarrow \Delta(\log^2 i \times c) T(\arg^2 i \det^2 i) @ 20$$

$$A(loc\ \ j,x,c),T(org\ \ \ j,dst\ \ \)\rightarrow A(loc\ \ \ i,x,c),T(org\ \ \ j,dst\ \ \ \) \ @\ 2.0$$

$$A(loc^{\tilde{}}j,x,c),T(org^{\tilde{}}j,dst^{\tilde{}}k) \rightarrow A(loc^{\tilde{}}k,x,c),T(org^{\tilde{}}j,dst^{\tilde{}}k) @ 1.8$$

$$A(loc^{k},x,c),T(org^{k},dst^{i}) \rightarrow A(loc^{i},x,c),T(org^{k},dst^{i}) @ 1.0$$

$$A(loc^k,x,c),T(org^k,dst^j) \rightarrow A(loc^j,x,c),T(org^k,dst^j) @ 1.1$$

```
pre-\kappa
expander
```

pre-Kappa expander

```
#Location matrices
```

%locm:

TM k 0.5 15 0 20 0 18 1.0 1.1

#Observing transporters

%expand-obs: TM 'Transporter(%org,%dst)' T()

```
pre-\kappa expander
```

hurbin

Introductio
What is κ

Kappa at DLab

DLab's currer work

Space-related simulations

Timing contro pre-Kappa expander

```
#Location matrices
```

%locm:

```
TM i j k i 0 0.5 1.5 j 2.0 0 1.8 k 1.0 1.1 0
```

#Observing transporters

%expand-obs: TM 'Transporter(%org,%dst)' T()

```
%obs: 'Transporter(i,j)' T(org~i,dst~j)
%obs: 'Transporter(i,k)' T(org~i,dst~k)
%obs: 'Transporter(j,i)' T(org~j,dst~i)
%obs: 'Transporter(j,k)' T(org~j,dst~k)
```

pre-Kappa syntax: Chains

pre- κ expander

hurbin

Introductio

What is I

Kappa at

DLab's curren

vork Space-related simulations

símulations Fiming contro

pre-Kappa expander

#Rules

 $\text{`Expression' S(x!1),V(p!1,n!2),V(p!2,n!3),...,V(p!10,n)} \rightarrow \text{Z() @ [inf]}$

gives:

'Expression' $S(x!1),V(p!1,n!2),V(p!2,n!3),V(p!3,n!4), V(p!4,n!5),V(p!5,n!6),V(p!6,n!7),V(p!7,n!8),V(p!8,n!9), V(p!9,n!10),V(p!10,n) <math>\rightarrow Z()$ @ [inf]

pre-Kappa syntax: Chains

pre- κ expander

hurbin

Introductio

What is i κ syntax

Kappa at

OLab's current vork Space-related

simulations Timing contro pre-Kappa

expander

#Rules

 $\text{`Expression' S(x!1),V(p!1,n!2),V(p!2,n!3),...,V(p!10,n)} \rightarrow \text{Z() @ [inf]}$

gives:

'Expression' $S(x|1),V(p|1,n|2),V(p|2,n|3),V(p|3,n|4), V(p|4,n|5),V(p|5,n|6),V(p|6,n|7),V(p|7,n|8),V(p|8,n|9), V(p|9,n|10),V(p|10,n) <math>\rightarrow$ Z() @ [inf]