Spatio-temporal model checking using `topochecker`

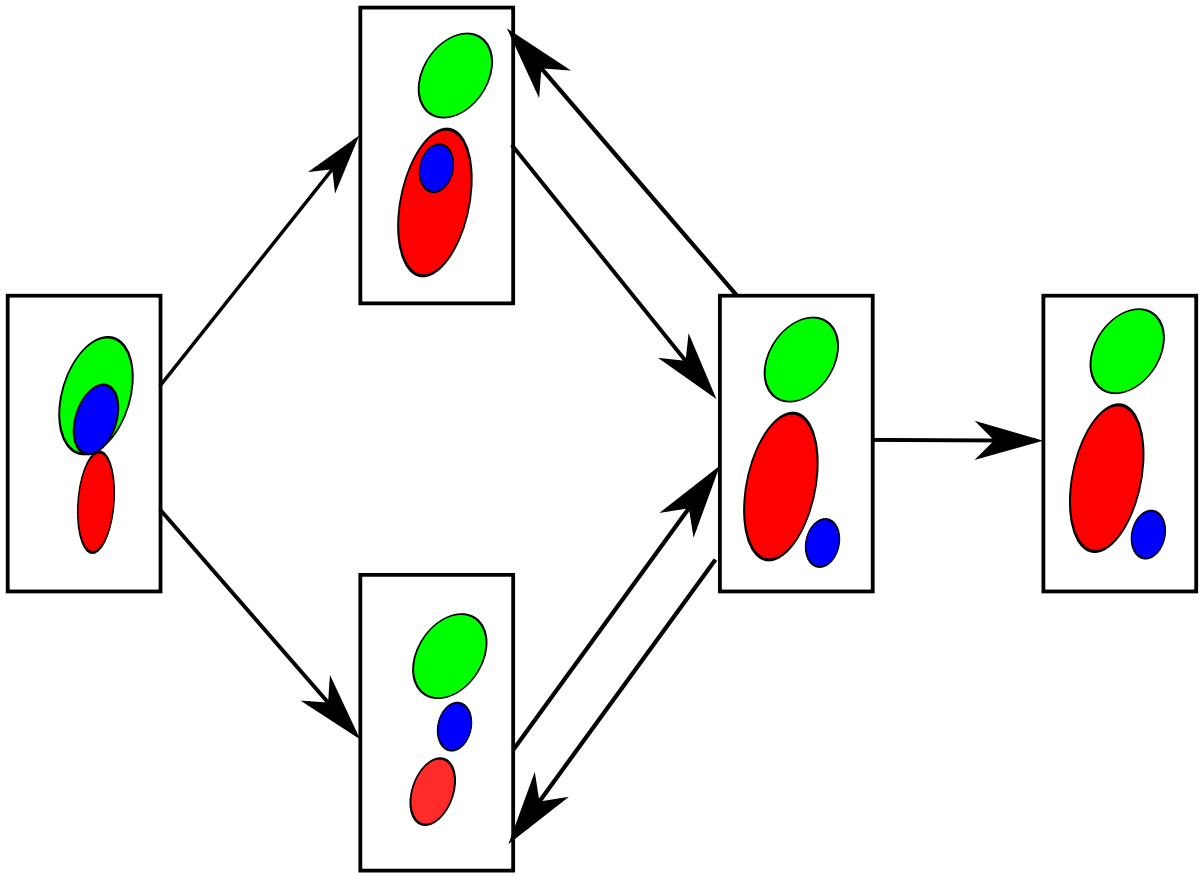
Vincenzo Ciancia, Diego Latella, Michele Loreti, Mieke Massink

Bertinoro, June 23 2015

Specification

Spatio-temporal logics

Branching time spatio-temporal models



Branching time spatio-temporal models

The product of a Kripke frame and a closure space

$\mathcal{M} = ((X, \mathcal{C}), (S, \mathcal{R}), \mathcal{V}_{s \in S})$

X points in space, S states, $\mathcal{C}, \mathcal{R}$, spatial and temporal neighbourhood/successor relations.

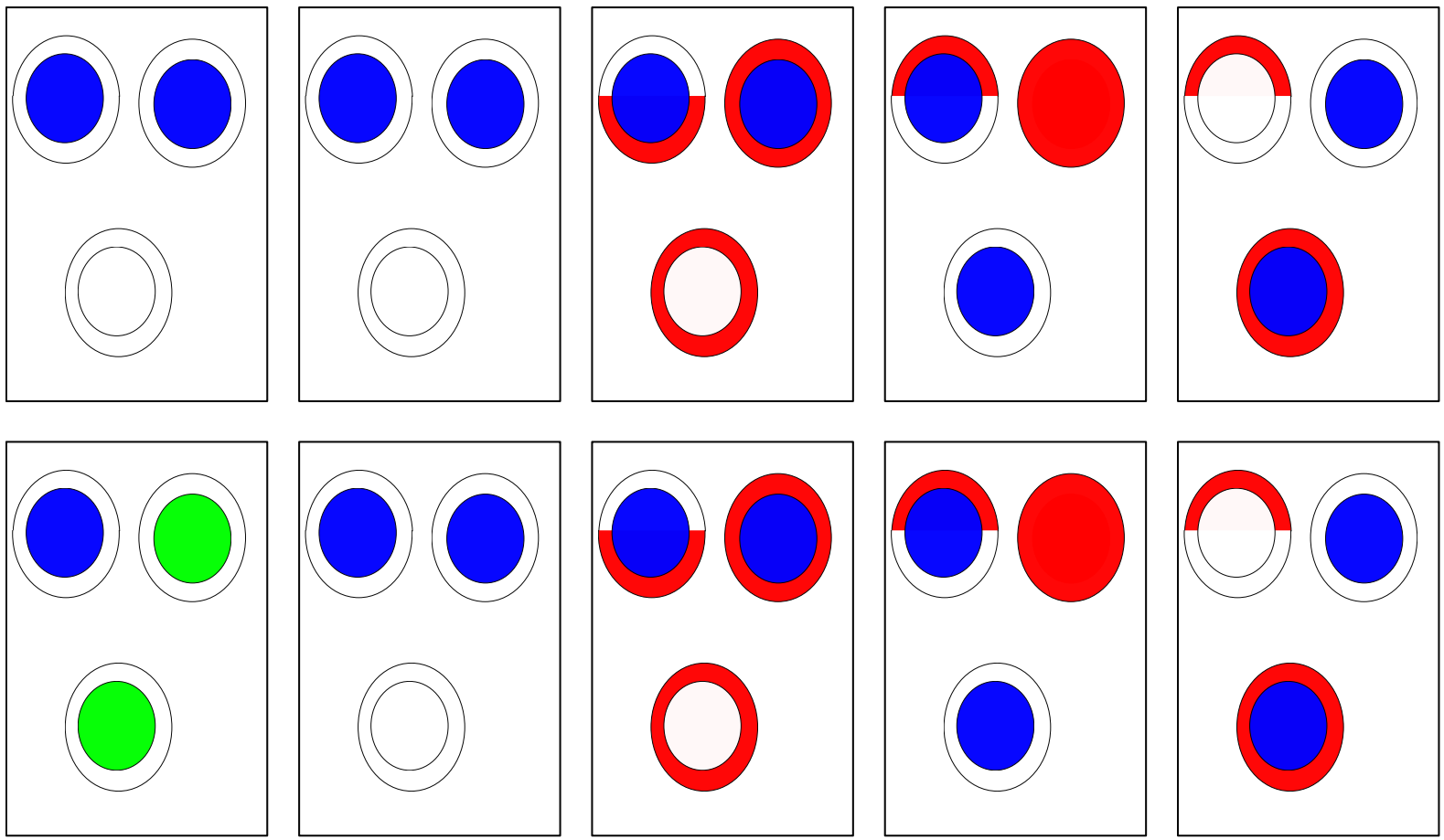
$\mathcal{V}$: assign to proposition p the pairs (x, s) where it is true

Syntax

Grammar				topochecker
Φ	$::=$	$\top$	[True]	`TT`
	$\mid$	p	[Atomic predicate]	`[p]`
	$\mid$	$\neg\Phi$	[Not]	`! phi`
	$\mid$	$\Phi \vee \Phi$	[Or]	`phi1 | phi2`
	$\mid$	$\Phi \wedge \Phi$	[And]	`phi1 & phi2`
	$\mid$	$\mathcal{N}\,\Phi$	[Near]	`N phi`
	$\mid$	$\Phi\mathcal{S}\Phi$	[Surrounded]	`phi S phi`
	$\mid$	$\mathtt{A}\,\varphi$	[All Futures]	`A ...`
	$\mid$	$\mathtt{E}\,\varphi$	[Some Future]	`E ...`
φ	$::=$	$\mathcal{X}\,\Phi$	[Next]	`X phi`
	$\mid$	$\mathtt{F}\,\Phi$	[Eventually]	`F phi`
	$\mid$	$\mathtt{G}\,\Phi$	[Globally]	`G phi`
	$\mid$	$\Phi\ \mathcal{U}\ \Phi$	[Until]	`phi1 U phi2`

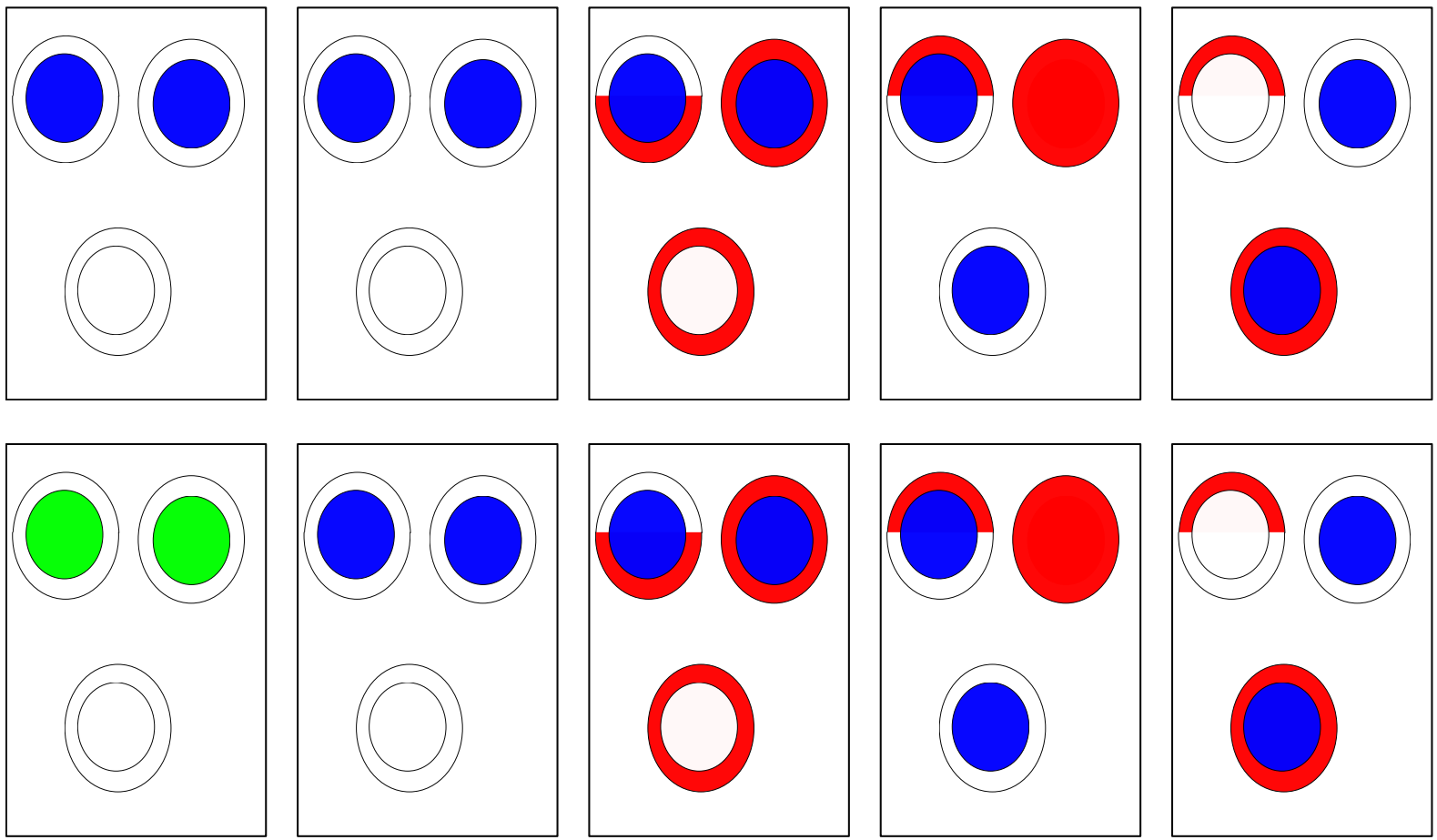
Eventually surrounded by ϕ

$\mathtt{EF}\,(blue\,\mathcal{S}\,red)$



Surrounded by eventually ϕ

$blue\,\mathcal{S}\,(\texttt{EF}\ red)$



Basics of `topochecker`

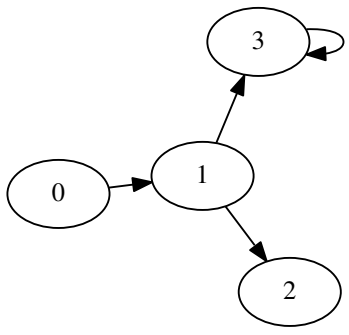
Model definitions

Graphs as models

Dot file format

```
digraph{
    0; 1; 2; 3;
    0->1; 1->2; 1->3; 3->3;
}
```

Graphs as models

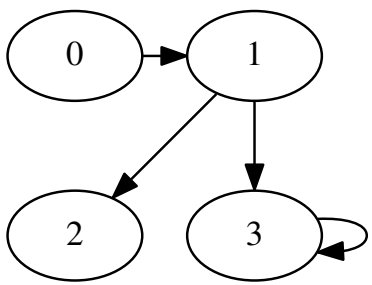


Graphs as models

Dot file format

```
digraph{
    0 [pos = "1,1!"];
    1 [pos = "2,1!"];
    2 [pos = "1,0!"];
    3 [pos = "2,0!"];
    0->1; 1->2; 1->3; 3->3;
}
```

Graphs as models



Valuation of propositional letters

CSV with three columns: Kripke state, point in space, list of propositions

```
0,0,a
0,1,b
0,2,a
0,3,a,b
```

Valuation of propositional letters

CSV with three columns: Kripke state, point in space, list of propositions, optional quantitative values

```
0,0,a=1.0
0,1,b=2.0
0,2,a=3.0
0,3,a=10.0,b=5.0
```

Atomic propositions are quantitative constraints

```
[x < 100]

[x >= 115] & [x < 190]

[x >= 115] & (! [y > 10])
```

Images are spatial models

A node for each pixel

An edge for each adjacent pair

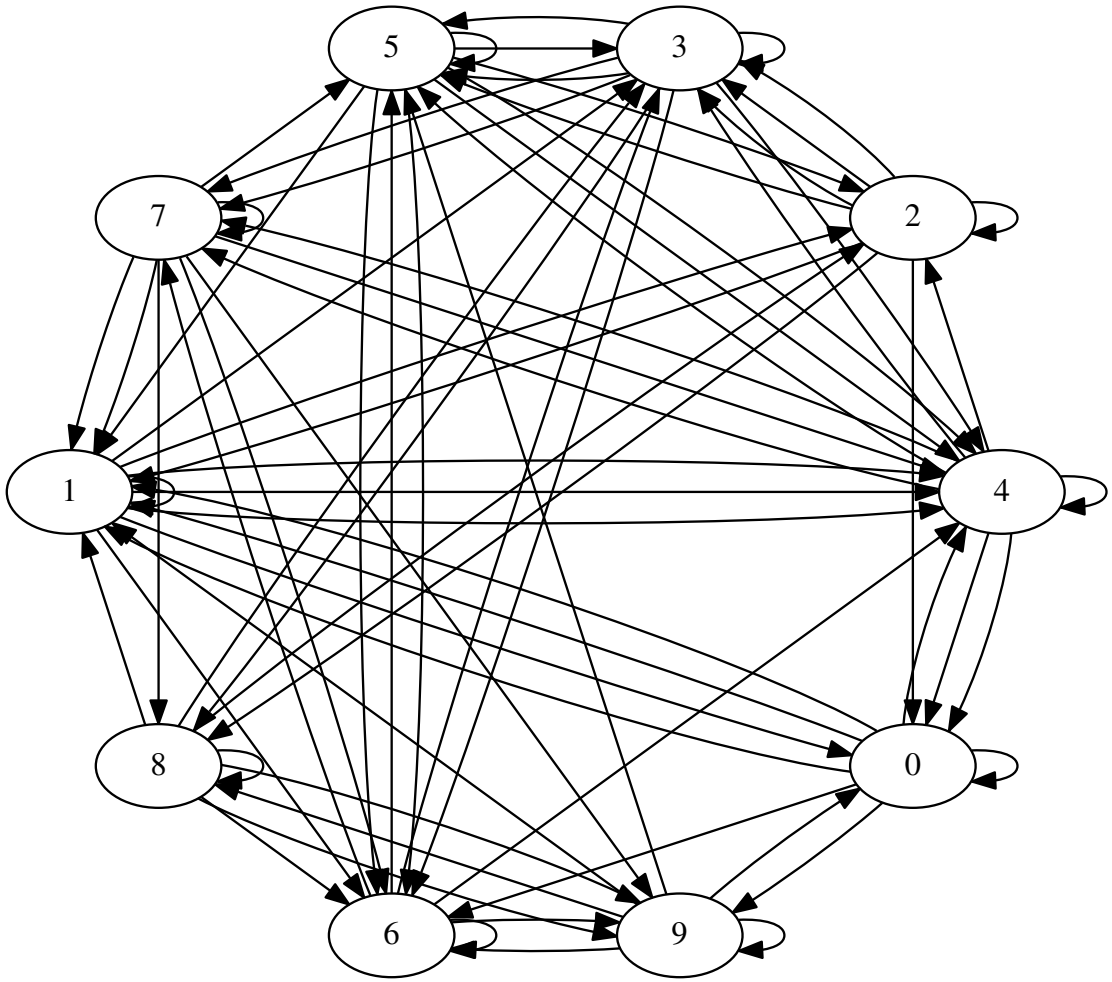
Implemented: 4-adjacency: n-s-w-e

Colours as atomic propositions

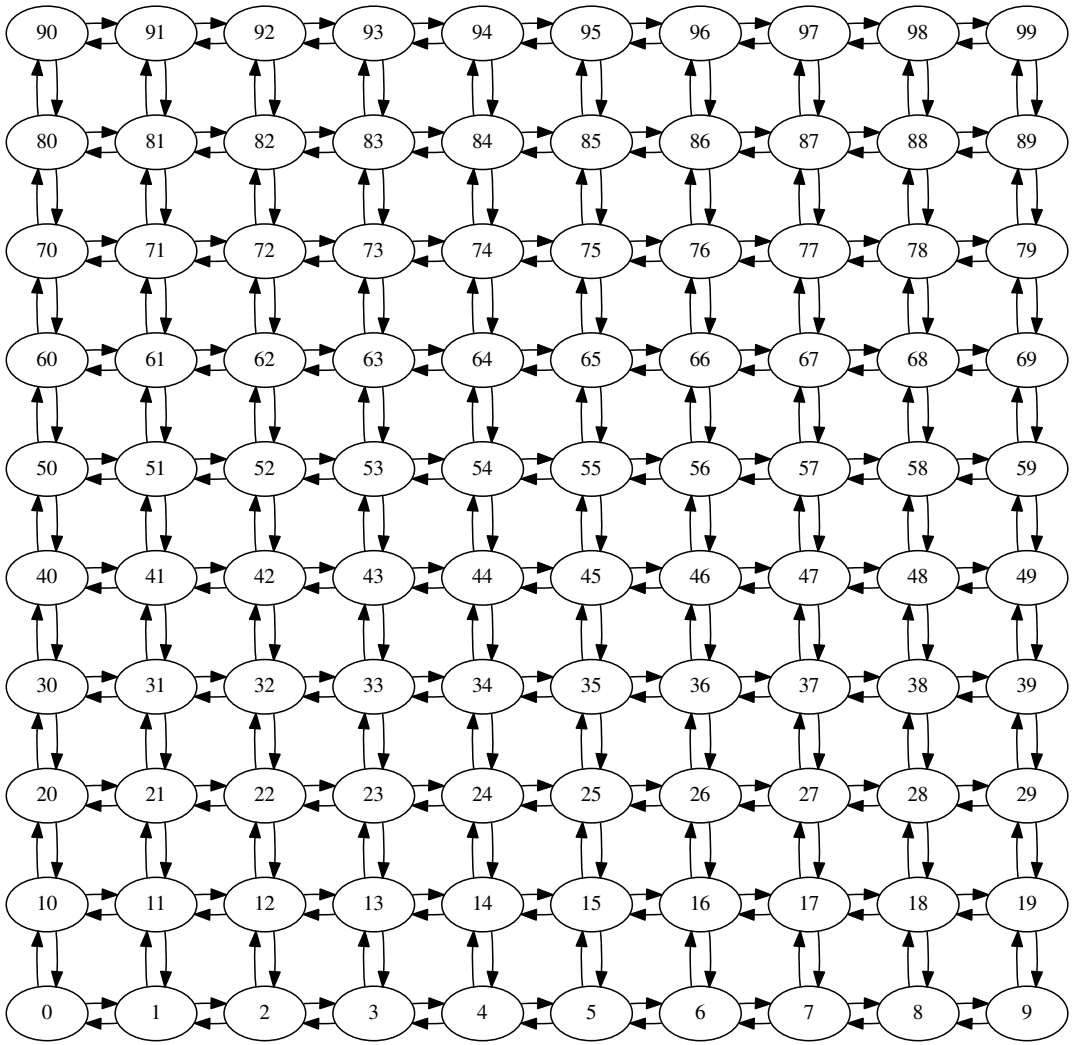
RGB components are quantitative propositions

```
   [Red = 100]
& [Blue = 100]
& [Green = 0];

[Red > 20] & [Blue = 10];

[Green > 100];
```

Kripke.dot



space.dot


90 91 92 93 94 95 96 97 98 99
80 81 82 83 84 85 86 87 88 89
70 71 72 73 74 75 76 77 78 79
60 61 62 63 64 65 66 67 68 69
50 51 52 53 54 55 56 57 58 59
40 41 42 43 44 45 46 47 48 49
30 31 32 33 34 35 36 37 38 39
20 21 22 23 24 25 26 27 28 29
10 11 12 13 14 15 16 17 18 19
0 1 2 3 4 5 6 7 8 9

eval.csv

```
...
0,95,b
0,96,b
0,97,b
0,98,b
0,99,b
1,0,b
1,1,b
...
```

Topochecker sessions

```
Kripke "kripke.dot"
Space "space.dot"
Eval "eval.csv";

Let phi = [a] S [b];

Output "output/state" "0","3";

Check "0xFF0000" A F phi;
```

Running topochecker

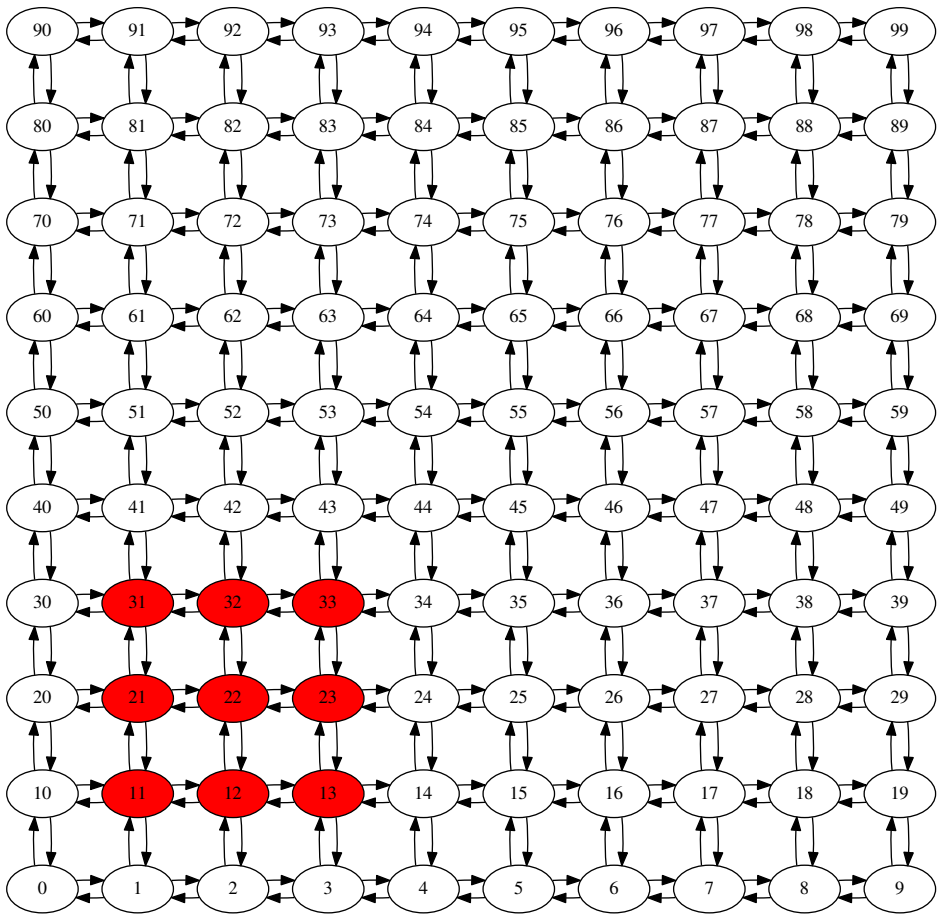
```
/path/to/topochecker foo.topochecker
```

Resulting files

```
$ ls output/
state-0.dot  state-3.dot
```

Result

```
neato -Tx11 output/state-0.dot
```



Multiple outputs

```
Kripke "kripke.dot"
Space "space.dot"
Eval "eval.csv";

Let phi = [a] S [b];

Output "output/state" "0","3";
Check "0xFF0000" A F phi;

Output "output2/state" "0";
Check "0x00FF00" E X phi;
```

Repeated execution and caching

```
$ ls
ccfad490e0c46621ef4215dec64ff0d5cdac5f9d2d2fe6fe08e
110a9fc457ec8_842662cb094d8af5bf6f92b8245fb5ef9e3ec
4272adcc2b7538b0adf4a613392.fmla

ccfad490e0c46621ef4215dec64ff0d5cdac5f9d2d2fe6fe08e
110a9fc457ec8_842662cb094d8af5bf6f92b8245fb5ef9e3ec
4272adcc2b7538b0adf4a613392.slice

...

eval.csv kripke.dot space.dot test.topochecker
```

Complexity

Global k-passes algorithm (k very small)

Complexity: $\mathcal{O}(|S_n + S_e| \cdot |X_n + X_e| \cdot |\phi|)$

Efficiency: 10000 states, 10000 points of space, 4 edges per node, ~ 3 subformulas, ~ 300 millions of cells

Computation time 28 seconds, central memory used: ~ 1.6 gigabytes

If it fits memory it'll be fine

Enemy is *free space*, not *computation time*

(after all it's spatial model checking :-))

If it fits memory it'll be fine

Enemy is *free space*, not *computation time*

(after all it's spatial model checking :-))

Break

(10 mins)

Theory

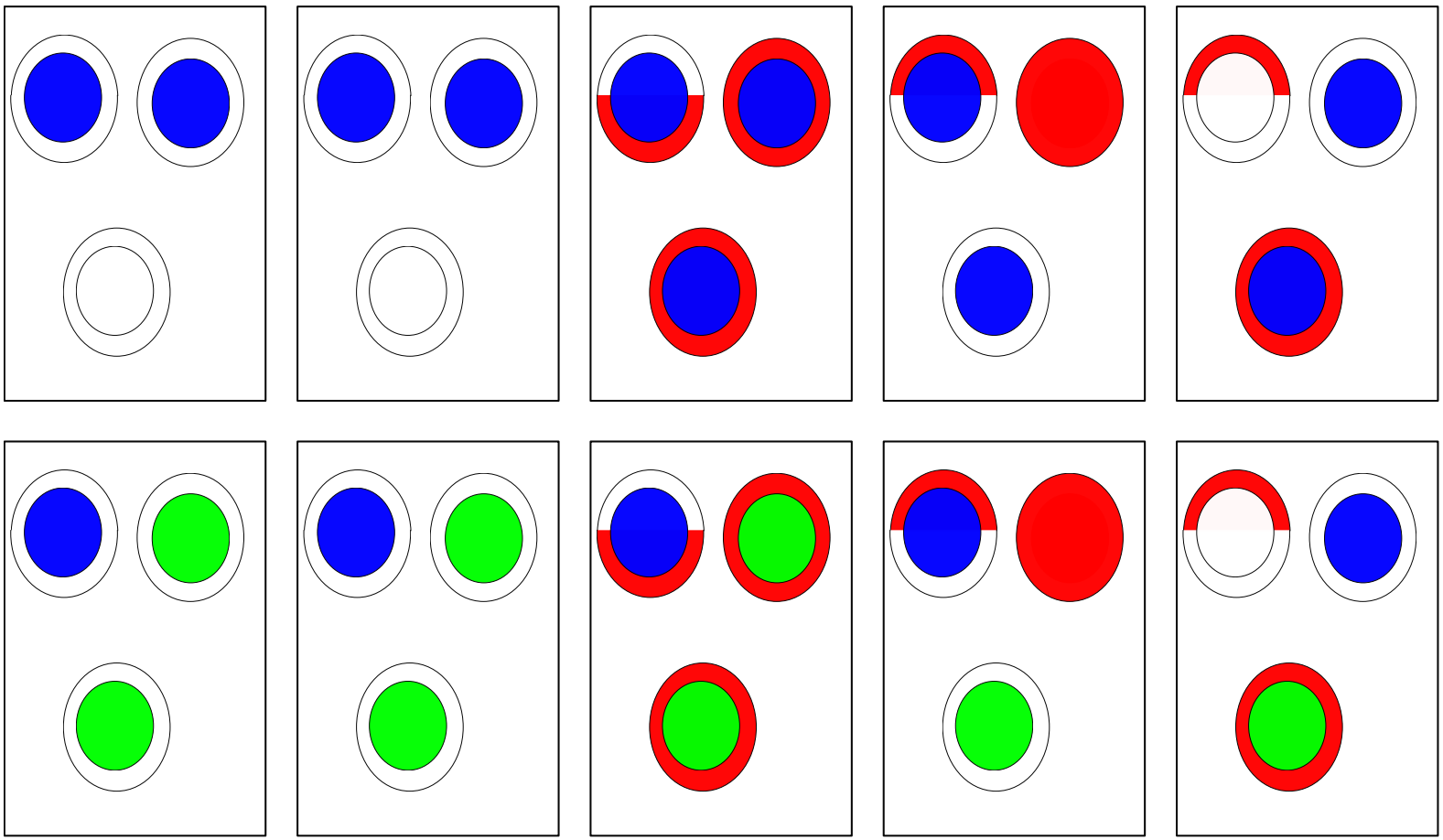
Formal semantics

Satisfaction

For each point x (of space) and state s:

$$(x, s) \models \phi$$

$\mathtt{EF}\,(blue\,\mathcal{S}\,red)$



Syntax

Grammar				topochecker
Φ	$::=$	$\top$	[True]	`TT`
	$\mid$	p	[Atomic predicate]	`[p]`
	$\mid$	$\neg\Phi$	[Not]	`! phi`
	$\mid$	$\Phi \vee \Phi$	[Or]	`phi1 | phi2`
	$\mid$	$\Phi \wedge \Phi$	[And]	`phi1 & phi2`
	$\mid$	$\mathcal{N}\,\Phi$	[Near]	`N phi`
	$\mid$	$\Phi\mathcal{S}\Phi$	[Surrounded]	`phi S phi`
	$\mid$	$\mathtt{A}\,\varphi$	[All Futures]	`A ...`
	$\mid$	$\mathtt{E}\,\varphi$	[Some Future]	`E ...`
φ	$::=$	$\mathcal{X}\,\Phi$	[Next]	`X phi`
	$\mid$	$\mathtt{F}\,\Phi$	[Eventually]	`F phi`
	$\mid$	$\mathtt{G}\,\Phi$	[Globally]	`G phi`
	$\mid$	$\Phi\ \mathcal{U}\ \Phi$	[Until]	`phi1 U phi2`

Interpretation

$\mathcal{M} = ((X, \mathcal{C}), (S, \mathcal{R}), \mathcal{V}_{s \in S})$

$$
\begin{array}{lcll}
\mathcal{M}, x, s & \models & \top & \\
\mathcal{M}, x, s & \models & p & \iff x \in \mathcal{V}_s(p) \\
\mathcal{M}, x, s & \models & \neg\Phi & \iff \mathcal{M}, x, s \not\models \Phi \\
\mathcal{M}, x, s & \models & \Phi \vee \Psi & \iff \mathcal{M}, x, s \models \Phi \text{ or } \mathcal{M}, x, s \models \Psi \\
\mathcal{M}, x, s & \models & \mathcal{N}\Phi & \iff x \in \mathcal{C}(\{y \in X | \mathcal{M}, y, s \models \Phi\}) \\
\mathcal{M}, x, s & \models & \Phi \,\mathcal{S}\, \Psi & \iff \exists A \subseteq X. x \in A \wedge \\
 & & & \quad \forall y \in A. \mathcal{M}, y, s \models \Phi \wedge \\
 & & & \quad \forall z \in \mathcal{B}^+(A). \mathcal{M}, z, s \models \Psi
\end{array}
$$

Interpretation

$\mathcal{M} = ((X, \mathcal{C}), (S, \mathcal{R}), \mathcal{V}_{s \in S})$

$$
\begin{array}{lcll}
\mathcal{M}, x, s & \models & \mathtt{A}\,\varphi & \iff \forall \sigma \in \mathcal{P}_s.\, \mathcal{M}, x, \sigma \models \varphi \\
\mathcal{M}, x, s & \models & \mathtt{E}\,\varphi & \iff \exists \sigma \in \mathcal{P}_s.\, \mathcal{M}, x, \sigma \models \varphi \\
 & & & \\
\mathcal{M}, x, \sigma & \models & \mathcal{X}\Phi & \iff \mathcal{M}, x, \sigma(1) \models \Phi \\
\mathcal{M}, x, \sigma & \models & \Phi\,\mathcal{U}\,\Psi & \iff \exists n.\, \mathcal{M}, x, \sigma(n) \models \Psi \text{ and} \\
 & & & \quad \forall n' \in [0, n). \mathcal{M}, x, \sigma(n') \models \Phi
\end{array}
$$

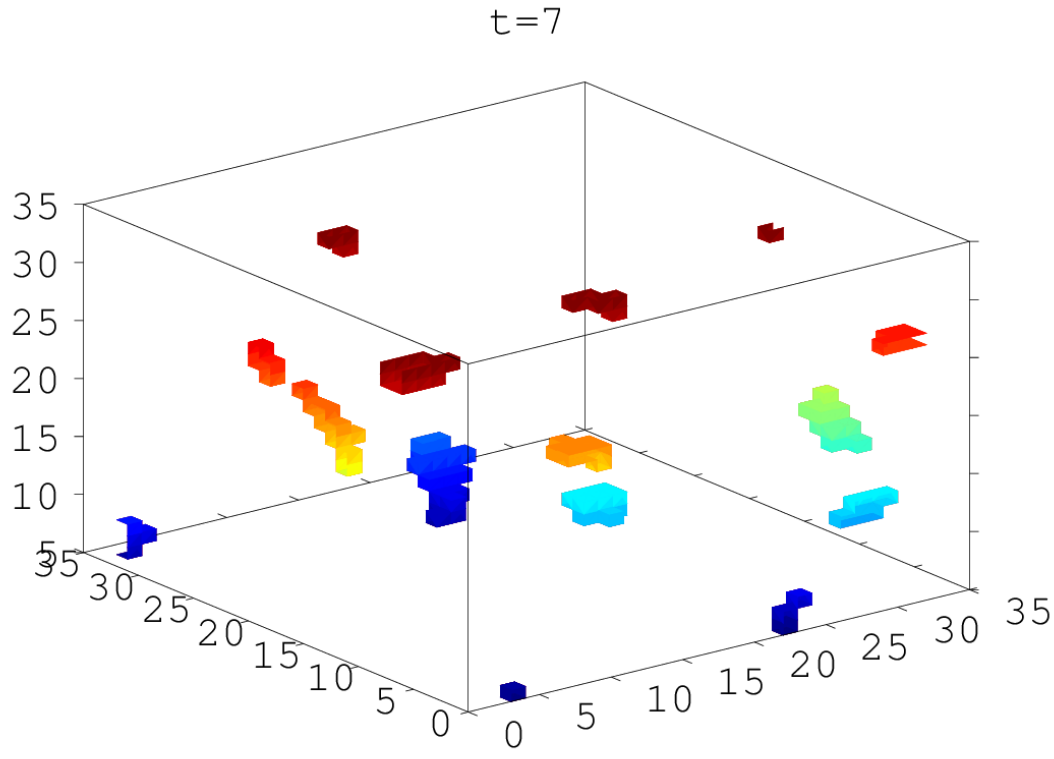
where $\mathcal{P}_s$ are the paths rooted in s in the Kripke structure

Example: stability of patterns

```
pattern2steps = pattern & A X pattern

pattern3steps = pattern & A X pattern2steps

...

pattern10steps = pattern9steps & ...
```

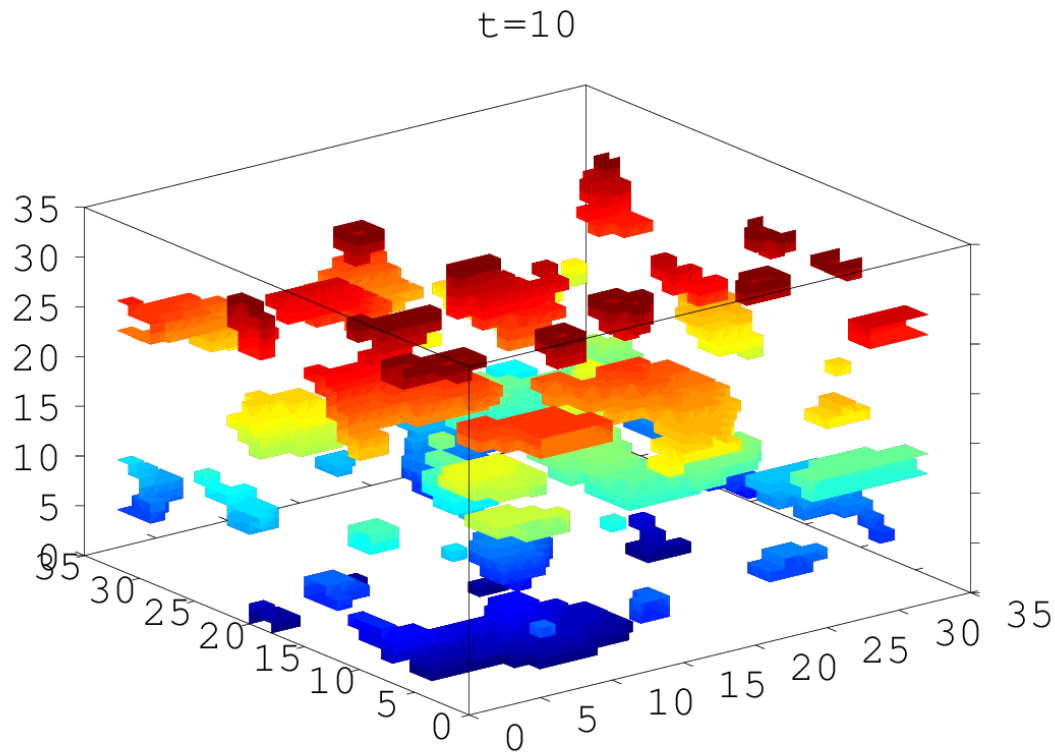
Areas of low concentration of A in 3D over time

Points where pattern persists for at least 9 steps



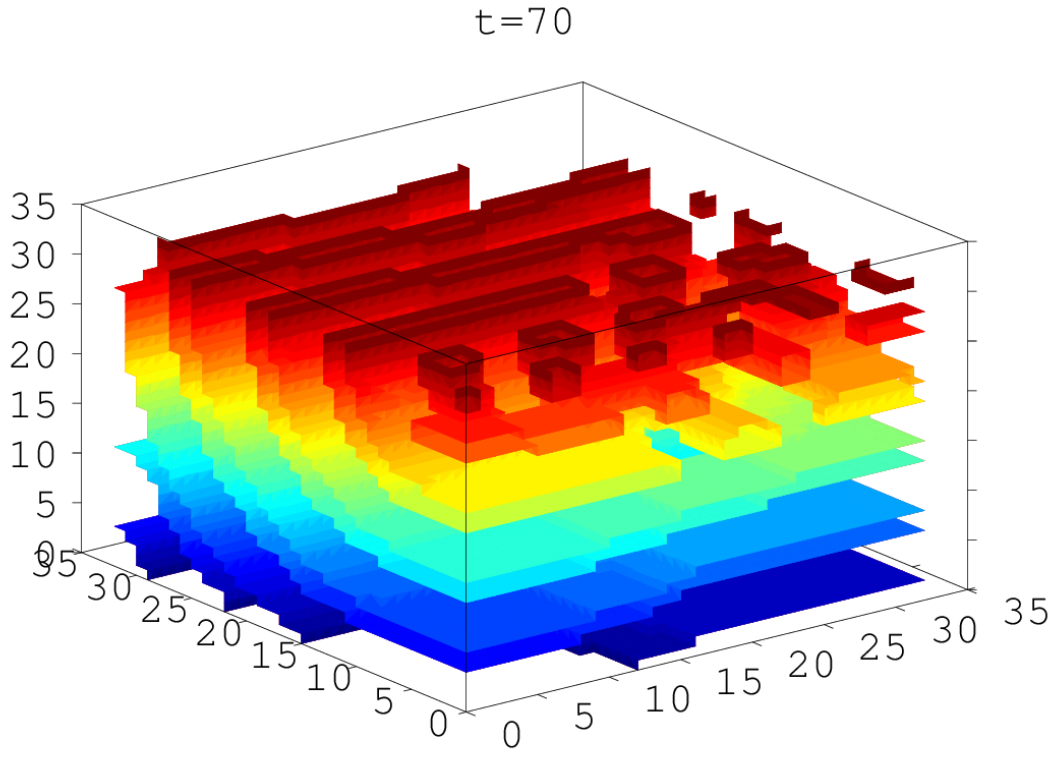
Areas of low concentration of A in 3D over time

Points where pattern persists for at least 9 steps



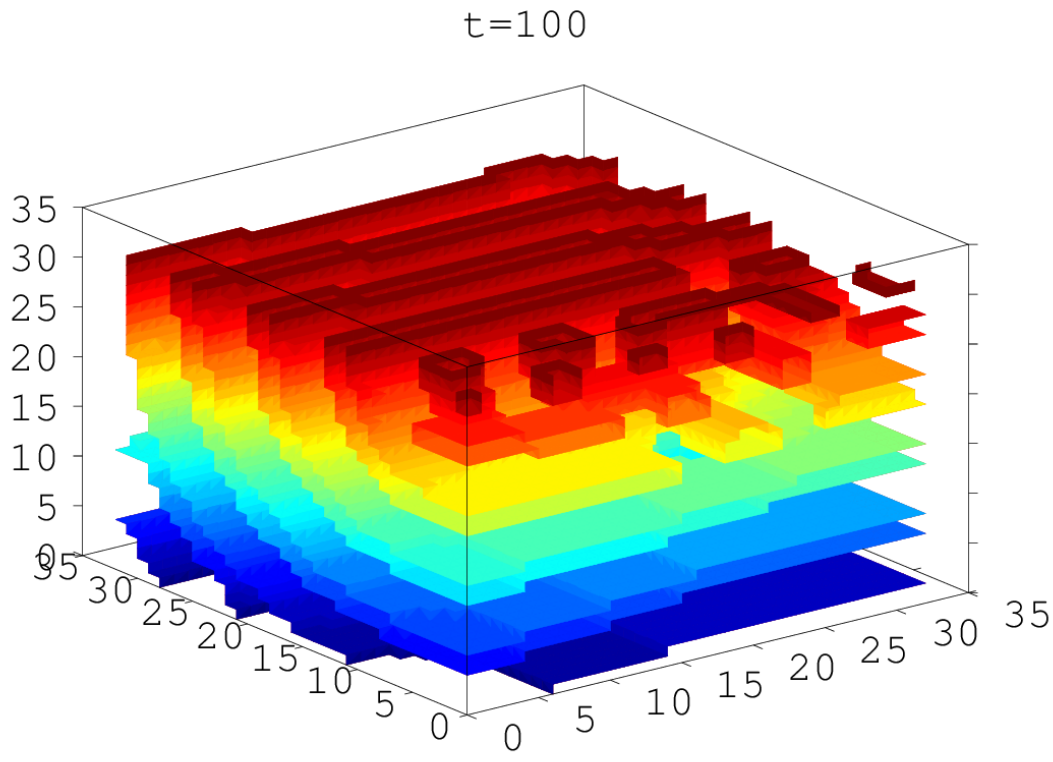
Areas of low concentration of A in 3D over time

Points where pattern persists for at least 9 steps



Areas of low concentration of A in 3D over time

Points where pattern persists for at least 9 steps



Case study

Bike sharing systems

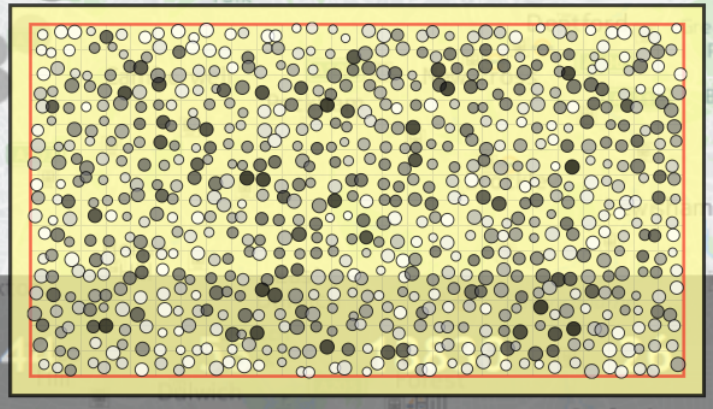
[Ciancia, Latella, Massink, Paskauskas - SASOWS2015]

Bike sharing

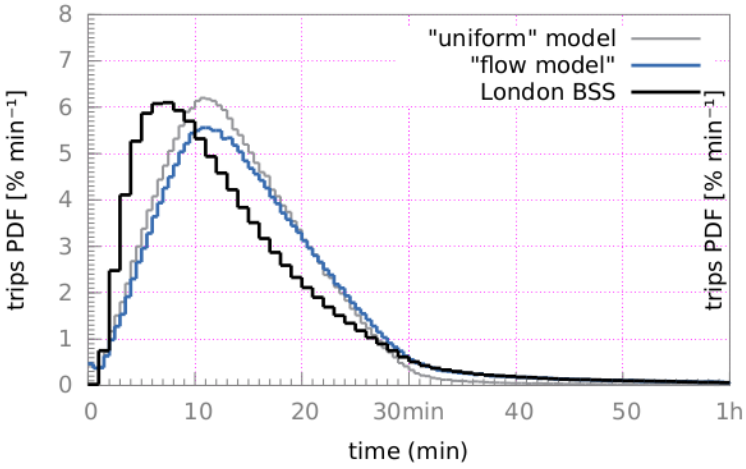
Key	Value
Stations	742
Capacity	19,000
Bike Fleet	11,500
Trips·h^{-1}	1,120
Area (km^2)	90

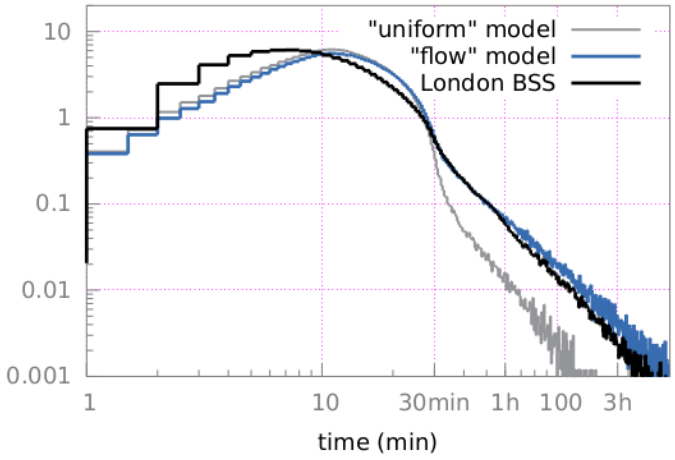
- Rectangular map
- Randomly distributed stations
- Bird's flight itineraries

Background img.: `http://bikes.oobrien.com/london`



Bike sharing





Uniform Multi-agent, uniform OD

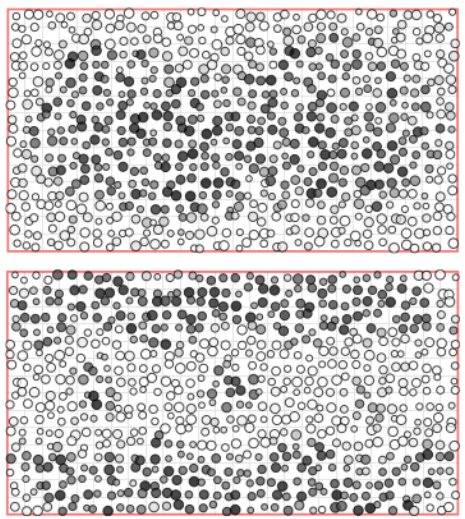
- Trips(> 30)min$= 2\%$

Flow Multi-agent, non-uniform OD

- Trips(> 30)min$= 7.7\%$ Bingo!

Expected Trips(> 30)min$= 0\%$

Soft control: dissolve clusters



Model checking

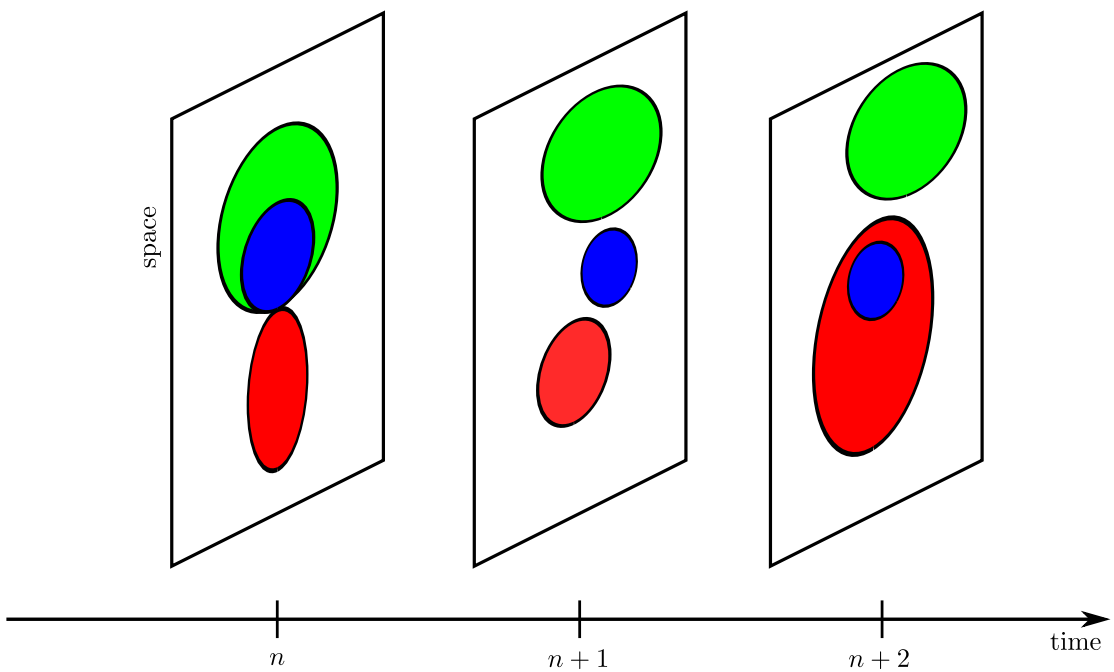
Use a simulator to produce possible spatio-temporal traces

Analyse traces using STLCS

(ongoing: *statistical* model checking)

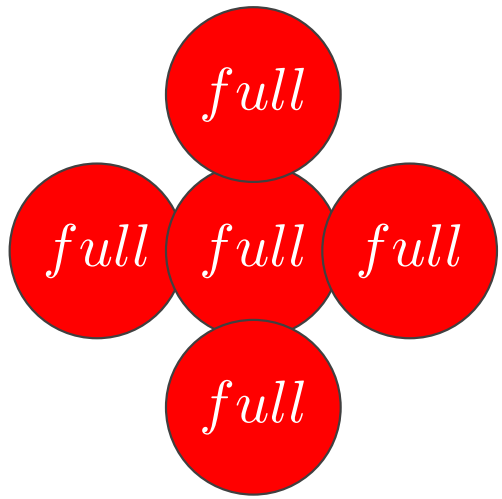
Model

Kripke model: linear



Spatial patterns

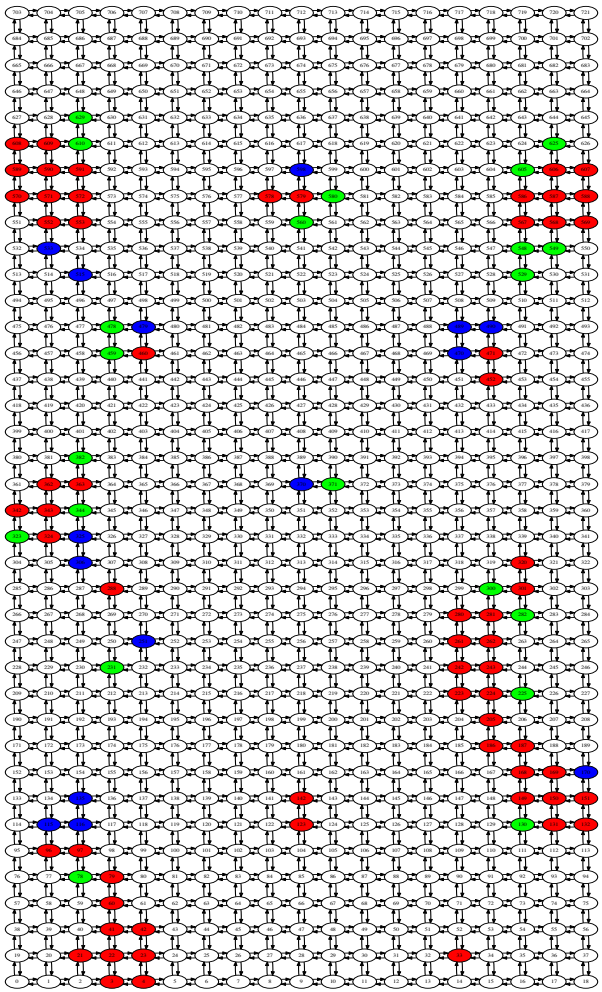
Cluster of full stations



Center of a cluster

```
Let full = [vacant == 0];
Let cluster = I full;
```

Persistent clusters



cluster $\triangleq$ `I full`

cluster2steps $\triangleq$ `cluster & (A X cluster)`

cluster3steps $\triangleq$ `cluster & (A X cluster2steps)`

eventually, cluster (green)

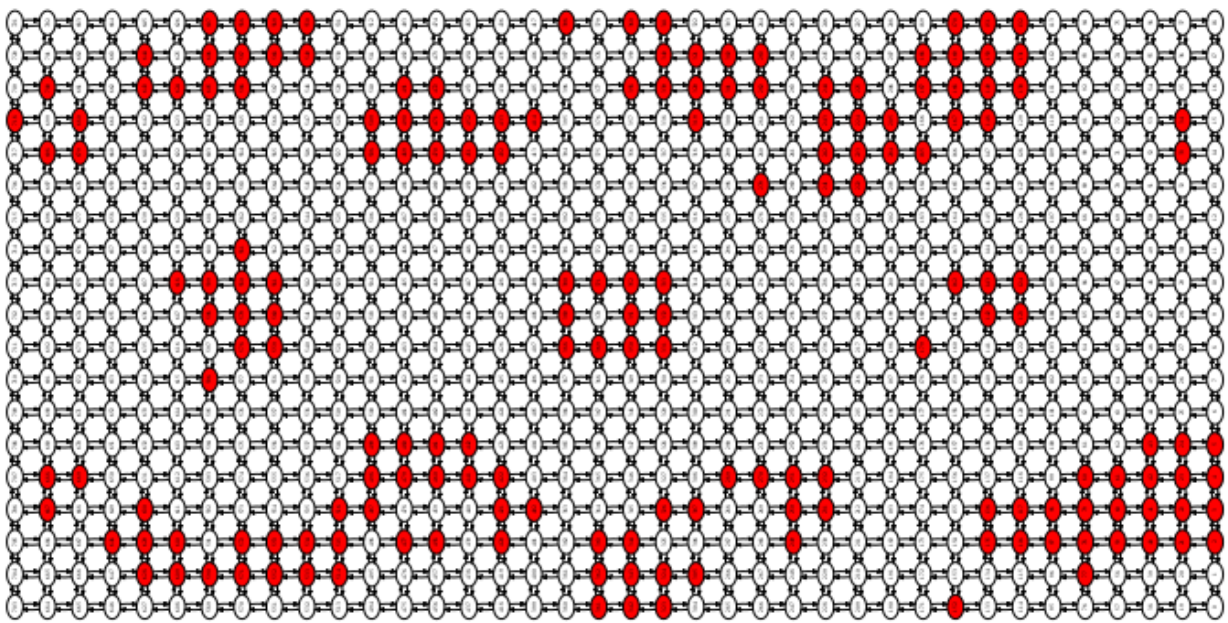
eventually, cluster that persists for two steps (blue)

eventually, cluster that persists for three steps (red)

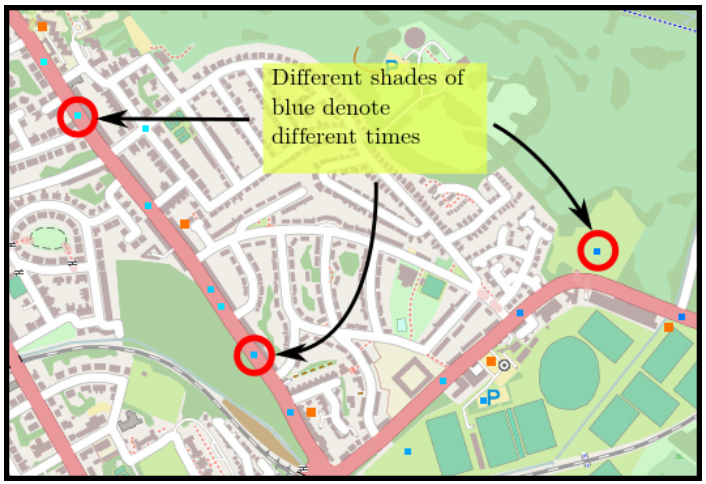
User dissatisfaction

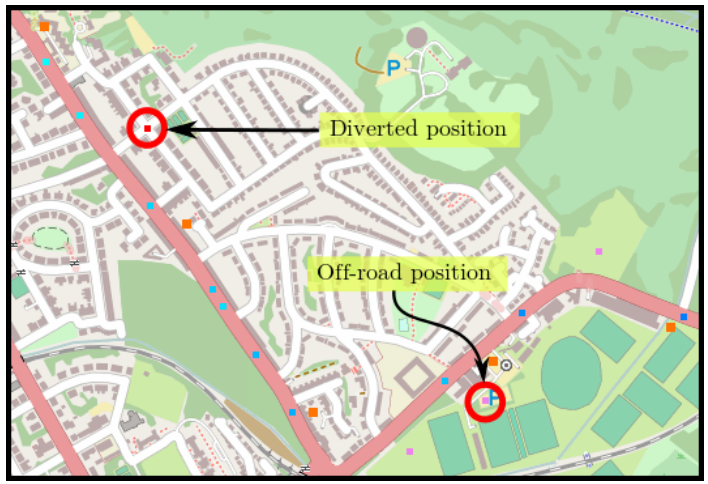
Upon bumping on a full station, cycle to a nearby one or wait
Rare event: User bumps to a full station, followed by bumping to a full station, followed by bumping to a full station,...
Formula to express this event:

$$\texttt{unlucky3steps} = \texttt{full} \,\&\, (\texttt{N}\,(\texttt{A}\,\texttt{X}\,(\texttt{full} \,\&\, (\texttt{N}\,(\texttt{A}\,\texttt{X}\,(\texttt{full} \,\&\, \texttt{N}\,(\texttt{A}\,\texttt{X}\ \texttt{full})))))))$$



Example: GPS traces





see tutorial

Outlook

Current research

Spatial propagation

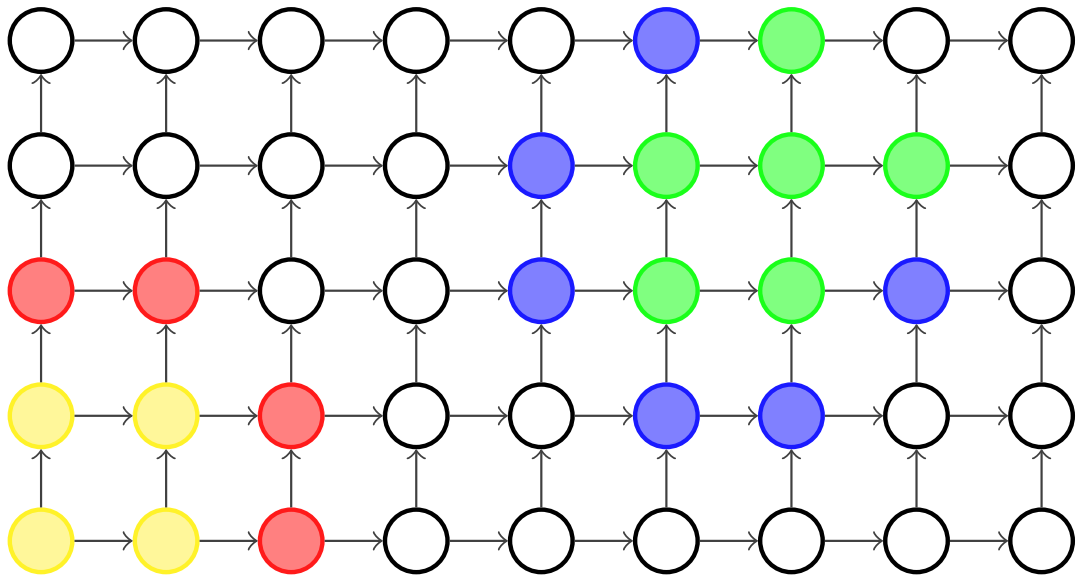
Directed graphs distinguish between “reaching” and “being reached”

Using a (spatial-)path based semantics, operators may be defined in both directions

Redefine $\phi_1 \mathcal{S} \phi_2$ using paths (currently a theorem)

Define operators that explore space in the opposite direction

Spatial propagation



Each yellow point satisfies $yellow \mathcal{S} red$

Each purple point satisfies satisfies $red \mathcal{P} white$

Statistical spatio-temporal model checking

V. Ciancia, D. Latella, M. Massink, A. Vandin

Models coming from probabilistic simulations

Boolean spatio-temporal model checking

Turn boolean values into probabilities!

Statistical spatio-temporal model checking

Repeat experiment several times

count occurrences

get a probability and a confidence interval for each point of spacetime

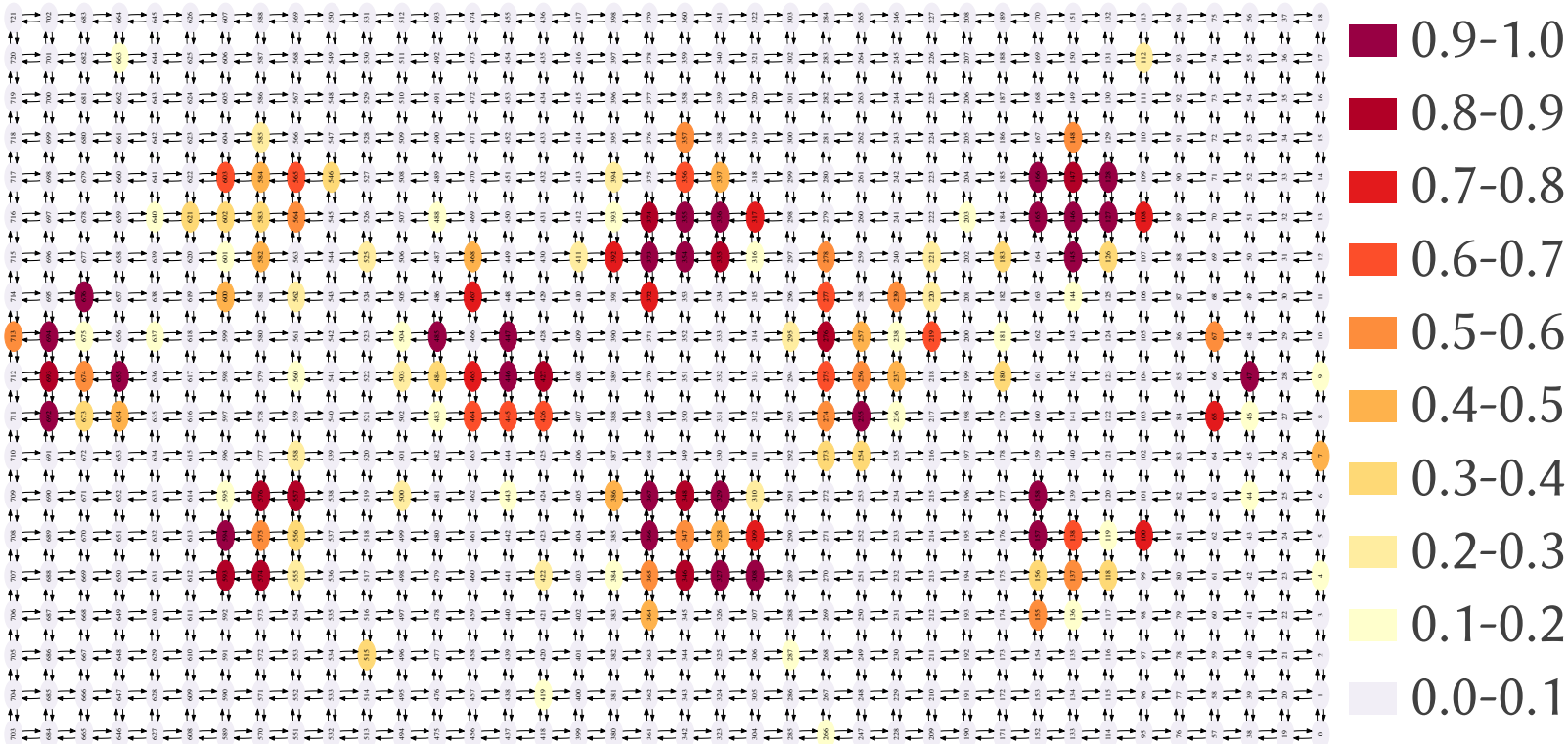
Tool support

A “server mode” in `topochecker`

Several queries for the same model

MultiVESTA: multiple (thousands!) observables, automatic orchestration of the simulator and model checker

Heat map (confidence intervals not shown)



Medical images as models

G. Belmonte, V. Ciancia, D. Latella, M. Massink

Just like images, but k-dimensional!

Anisotropic

Only one atomic proposition (intensity)

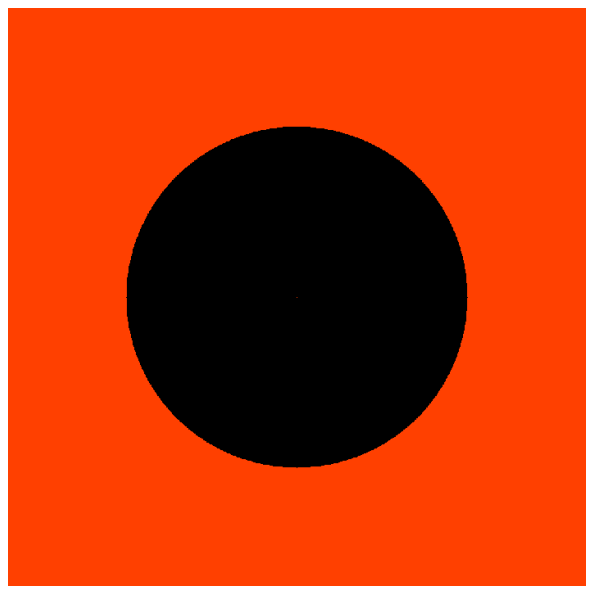
Distance operators

At most distant k, distance between k_1 and k_2
[Handbook of Spatial Logics]
[Nenzi, Bortolussi - VALUETOOLS 2014]

Bounded surrounded operator
[Nenzi, Bortolussi, Ciancia, Loreti, Massink - RV2015]

Distance transform

Use *distance transforms* for linear or quasi-linear model checking



Euclidean and shortest-path distance

Graphs: naturally augmented with *weights*, and *shortest-path* distance

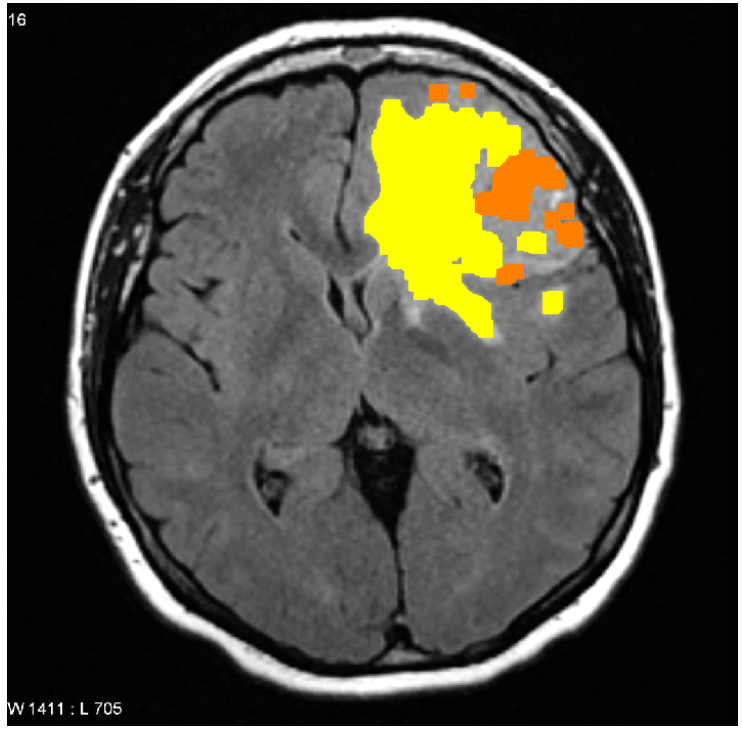
Images: naturally augmented with *positions*, and *Euclidean* distance

What models?

Topological spaces abstract metric spaces

Closure spaces abstract <?>

Task: detect tumor and oedema



Filtering noise

```
Let lt = [x > 90] & [x < 120];

Let tumor = N (N (N
        (I (I ((I (I (I lt))))))));
```

Detecting closeness

```
Let ut = [x >= 115] & [x < 190];
Let lt = [x > 90] & [x < 120];

Let lt1 = N (N (N (I (I (I (I (I lt)))))));
Let ut1 = N (N (N (N (I (I (I (I ut)))))));

Let utdst = MDDT (ut1,<=50.0);

Let reach(a,b) = !( (!b) S (!a) );

Let tum1 = lt1 & reach(lt1,utdst);
Let oed1 = ut1 & reach(utdst,tum1);

...
```

Ugly, but on the right track!

Medical image analysis *is* logically structured

Just better than the human eye!

Parameter finding [R. Grosu, S. A. Smolka, F. Corradini, A. Wasilewska, E. Entcheva, E. Bartocci - CACM 2009]

Collective logic

Collective formulas

$$
\begin{array}{lcll}
\Psi & ::= & \top & [\textsc{True}] \\
 & | & \neg\Psi & [\textsc{Not}] \\
 & | & \Psi \wedge \Psi & [\textsc{And}] \\
 & | & \Phi \prec \Psi & [\textsc{Share}] \\
 & | & \mathcal{G}\Phi & [\textsc{Group}]
\end{array}
$$

Individual formulas

$$
\begin{array}{lcll}
\Phi & ::= & p & [\textsc{Atomic prop.}] \\
 & | & \top & [\textsc{True}] \\
 & | & \neg\Phi & [\textsc{Not}] \\
 & | & \Phi \wedge \Phi & [\textsc{And}] \\
 & | & \mathcal{N}\Phi & [\textsc{Near}] \\
 & | & \Phi\mathcal{P}\Phi & [\textsc{Propagation}] \\
 & | & \Phi\mathcal{S}\Phi & [\textsc{Surrounded}]
\end{array}
$$

Collective satisfaction?

$\mathcal{M}, A \models \Psi$

A is a *set* of points!

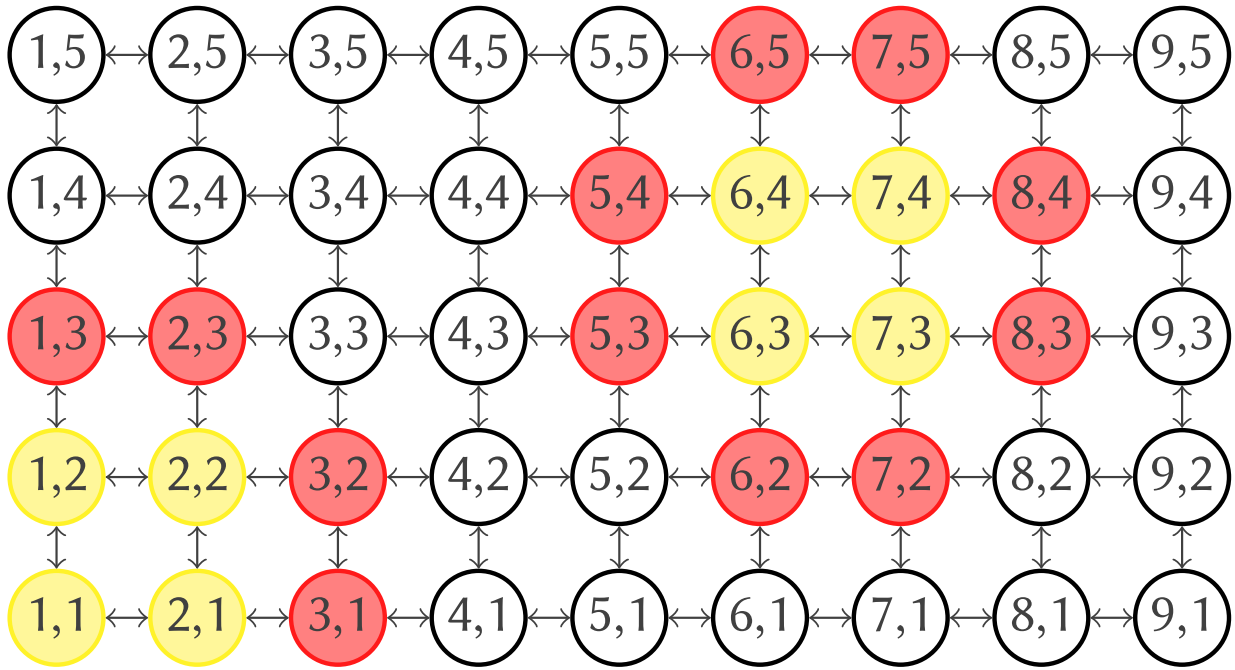
Semantics

$$
\begin{array}{lll}
\mathcal{M}, A \models_C \top & & \\
\mathcal{M}, A \models_C p & \iff & A \subseteq \mathcal{V}(p) \\
\mathcal{M}, A \models_C \neg\psi & \iff & \mathcal{M}, A \not\models_C \psi \\
\mathcal{M}, A \models_C \psi_1 \wedge \psi_2 & \iff & \mathcal{M}, A \models_C \psi_1 \text{ and } \mathcal{M}, A \models_C \psi_2 \\
 & & \\
\mathcal{M}, A \models_C \phi \prec \psi & \iff & \mathcal{M}, \{x \in A \mid \mathcal{M}, x \models \phi\} \models_C \psi \\
 & & \\
\mathcal{M}, A \models_C \mathcal{G}\phi & \iff & \exists B \subseteq X. A \subseteq B \wedge \\
 & & \wedge B \text{ is path-connected} \wedge \\
 & & \wedge \forall z \in B. \mathcal{M}, z \models \phi
\end{array}
$$

Derived operators

$$
\begin{array}{lcll}
\forall\phi & \triangleq & \neg\phi \prec \mathcal{G}\bot & [\textsc{Forall, Individually}] \\
\exists\phi & \triangleq & \neg(\forall\neg\phi) & [\textsc{Exists}] \\
\emptyset & \triangleq & \forall\bot & [\textsc{Empty}] \\
 & & & \\
\phi_1 \,\mathtt{CS}\, \phi_2 & \triangleq & \mathcal{G}(\neg\phi_2 \wedge (\phi_1 \mathcal{S} \phi_2)) & [\textsc{Collectively surrounded}]
\end{array}
$$

Collectively surrounded



$$M, \{(6,4),(7,3))\} \models \mathit{yellow}\ \texttt{CS}\ \mathit{red}$$
$$M, \{(1,1),(2,1),(2,2),(1,2))\} \models \mathit{yellow}\ \texttt{CS}\ \mathit{red}$$
$$M, \{(6,4),(2,1)))\} \not\models \mathit{yellow}\ \texttt{CS}\ \mathit{red}$$

Thanks for listening!