Spatial Model Checking

and its applications to Medical Image Analysis

SPIN 2021

27th International SPIN Symposium on Model Checking of Software, invited talk

Vincenzo Ciancia[1]

joint work with Gina Belmonte[2], Diego Latella[1], Mieke Massink[1]

1: **Formal Methods & Tools Laboratory**, Institute of Information Science and Technologies, CNR, Pisa

2: **Medical Physics**, San Luca Hospital, Lucca

Goal of the talk

(Modal, spatial) logic +
as a description language

Model checker =
as an interpreter

Declarative Domain-Specific Language
for analysing graphs,images,3d meshes,…

Advantages

Human-intelligible language
concise, unambiguous, collaborative,
knowledge-based, explainable, accountable…

Familiar application domain
logical specifications are easier to grasp for non-specialists

No side effects
syntactic manipulation, memoization, query optimization

Global model checking
parallel execution, GPU computation, caching, distributed execution, ...

Before we start...

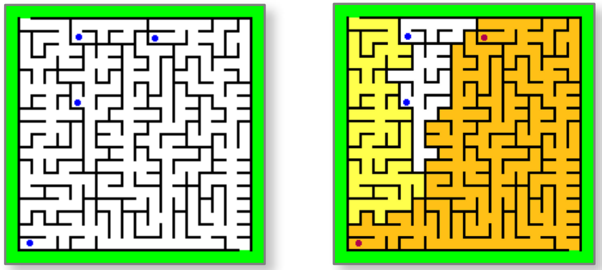
- A **spatio-temporal** notion of **system behaviour**?

  `Yes Of Course!`

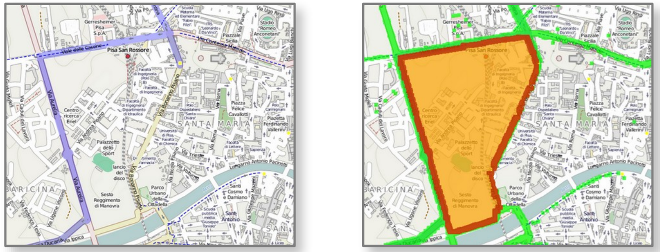
  (for instance, we used it in smart transportation case studies)

- In this talk: **purely spatial** properties of points
  main novelty in our approach

Examples

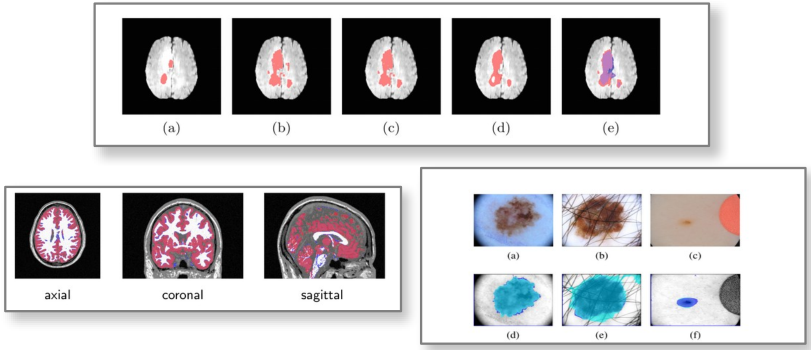


Reachability



Map annotation

GPS traces



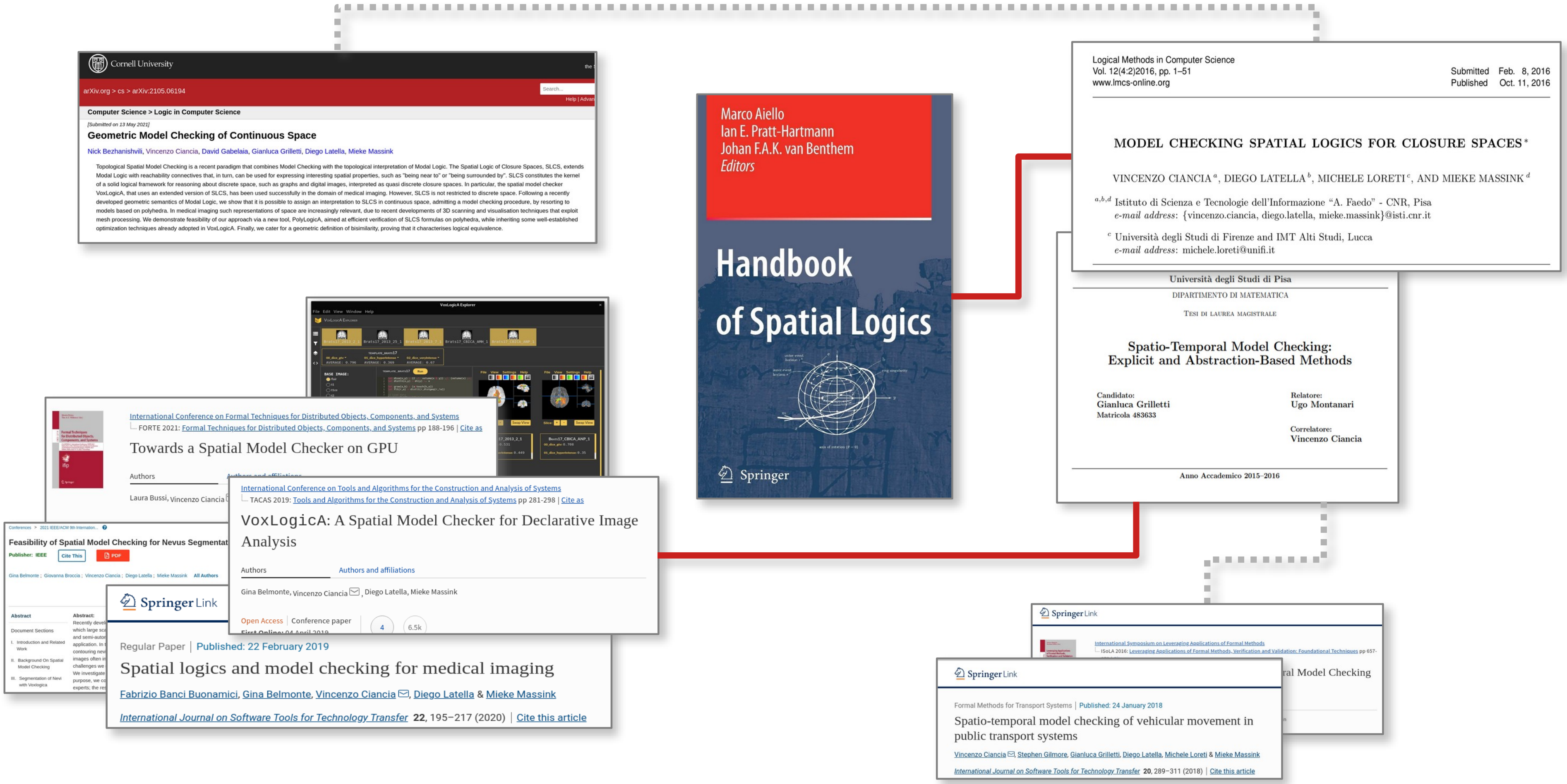
Medical Imaging

Overview

Plan of the talk:

1) Introduction to Topological Spatial Logics
2) Spatial Model Checking & VoxLogicA
3) Medical Imaging Case Study
4) Conclusions & Outlook

Outline



Cornell University
arXiv.org > cs > arXiv:2105.06194
Computer Science > Logic in Computer Science
[Submitted on 13 May 2021]
Geometric Model Checking of Continuous Space
Nick Bezhanishvili, Vincenzo Ciancia, David Gabelaia, Gianluca Grilletti, Diego Latella, Mieke Massink
International Conference on Formal Techniques for Distributed Objects, Components, and Systems
FORTE 2021: Formal Techniques for Distributed Objects, Components, and Systems pp 188-196 | Cite as
Towards a Spatial Model Checker on GPU
Authors
Laura Bussi, Vincenzo Ciancia
Feasibility of Spatial Model Checking for Nevus Segmentat
Publisher: IEEE
Cite This
PDF
International Conference on Tools and Algorithms for the Construction and Analysis of Systems
TACAS 2019: Tools and Algorithms for the Construction and Analysis of Systems pp 281-298 | Cite as
VoxLogicA: A Spatial Model Checker for Declarative Image Analysis
Authors
Authors and affiliations
Gina Belmonte, Vincenzo Ciancia, Diego Latella, Mieke Massink
Open Access
Conference paper
Springer Link
Regular Paper | Published: 22 February 2019
Spatial logics and model checking for medical imaging
Fabrizio Banci Buonamici, Gina Belmonte, Vincenzo Ciancia, Diego Latella & Mieke Massink
International Journal on Software Tools for Technology Transfer 22, 195–217 (2020) | Cite this article
Marco Aiello
Ian E. Pratt-Hartmann
Johan F.A.K. van Benthem
Editors
Handbook of Spatial Logics
Springer
Logical Methods in Computer Science
Vol. 12(4:2)2016, pp. 1–51
www.lmcs-online.org
Submitted Feb. 8, 2016
Published Oct. 11, 2016
MODEL CHECKING SPATIAL LOGICS FOR CLOSURE SPACES
VINCENZO CIANCIA, DIEGO LATELLA, MICHELE LORETI, AND MIEKE MASSINK
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" - CNR, Pisa
e-mail address: {vincenzo.ciancia, diego.latella, mieke.massink}@isti.cnr.it
Università degli Studi di Firenze and IMT Alti Studi, Lucca
e-mail address: michele.loreti@unifi.it
Università degli Studi di Pisa
DIPARTIMENTO DI MATEMATICA
Tesi di Laurea Magistrale
Spatio-Temporal Model Checking: Explicit and Abstraction-Based Methods
Candidato: Gianluca Grilletti
Matricola 483633
Relatore: Ugo Montanari
Correlatore: Vincenzo Ciancia
Anno Accademico 2015–2016
Springer Link
International Symposium on Leveraging Applications of Formal Methods
Springer Link
Formal Methods for Transport Systems | Published: 24 January 2018
Spatio-temporal model checking of vehicular movement in public transport systems
Vincenzo Ciancia, Stephen Gilmore, Gianluca Grilletti, Diego Latella, Michele Loreti & Mieke Massink
International Journal on Software Tools for Technology Transfer 20, 289–311 (2018) | Cite this article

Part 1

Introduction to Spatial Logic

- Topological models
- Interpretation of modalities
- Examples

Goal of Part 1

Topological Models

of

Modal Logic

[McKinsey-Tarski 1944]:

“Modal logic is **topological**”

[McKinsey-Tarski 1944]:

"Modal logic is **topological**"

$$\Phi ::= \top \mid p \mid \Phi \wedge \Phi \mid \neg\Phi \mid \boxed{\Diamond\Phi}$$

[McKinsey-Tarski 1944]:

"Modal logic is **topological**"

$$\Phi ::= \top \mid p \mid \Phi \wedge \Phi \mid \neg\Phi \mid \boxed{\Diamond\Phi}$$

What does that mean?

Topological Space*

Definition via open sets [edit]

A *topological space* is an ordered pair (X, τ), where X is a set and τ is a collection of subsets of X, satisfying the following axioms:[7]

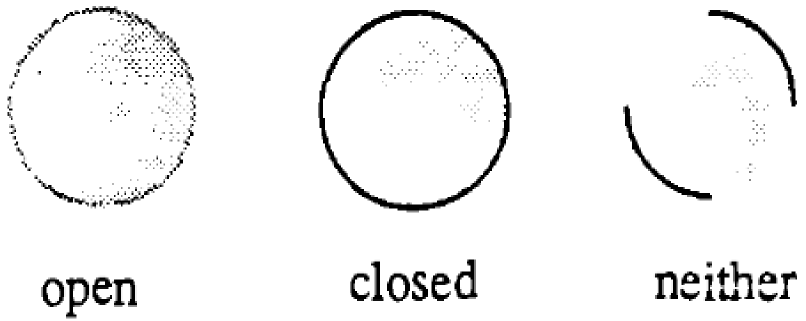
1. The empty set and X itself belong to τ.
2. Any arbitrary (finite or infinite) union of members of τ still belongs to τ.
3. The intersection of any finite number of members of τ still belongs to τ.

The elements of τ are called **open sets** and the collection τ is called a **topology** on X.

*https://en.wikipedia.org/w/index.php?title=Topological_space

Enter Topological Spaces

Closed sets: complements of open sets



Closure: a set **plus** its border

Interior: a set **without** its border

Image courtesy of https://ncatlab.org

Kuratowski definition of Topological Spaces

A **topological** space is a pair $(X, \mathcal{C})$

- X is a set
- $\mathcal{C} : \mathcal{P}(X) \to \mathcal{P}(X)$

- $\mathcal{C}(\emptyset) = \emptyset$
- $A \subseteq \mathcal{C}(A)$
- $\mathcal{C}(A \cup B) = \mathcal{C}(A) \cup \mathcal{C}(B)$
- $\mathcal{C}(\mathcal{C}(A)) = \mathcal{C}(A)$

Continuous function

$$f : X_1 \to X_2 \text{ with } f(\mathcal{C}_1(A)) \subseteq \mathcal{C}_2(f(A))$$

McKinsey & Tarski's proposal:

$$\Diamond \Phi$$

Interpret **diamond** as closure

(and box as interior)

Topological model

$$\mathcal{M} = (\underbrace{(X, \mathcal{C})}_{\text{Topological space (Kuratowski)}}, \mathcal{V})$$

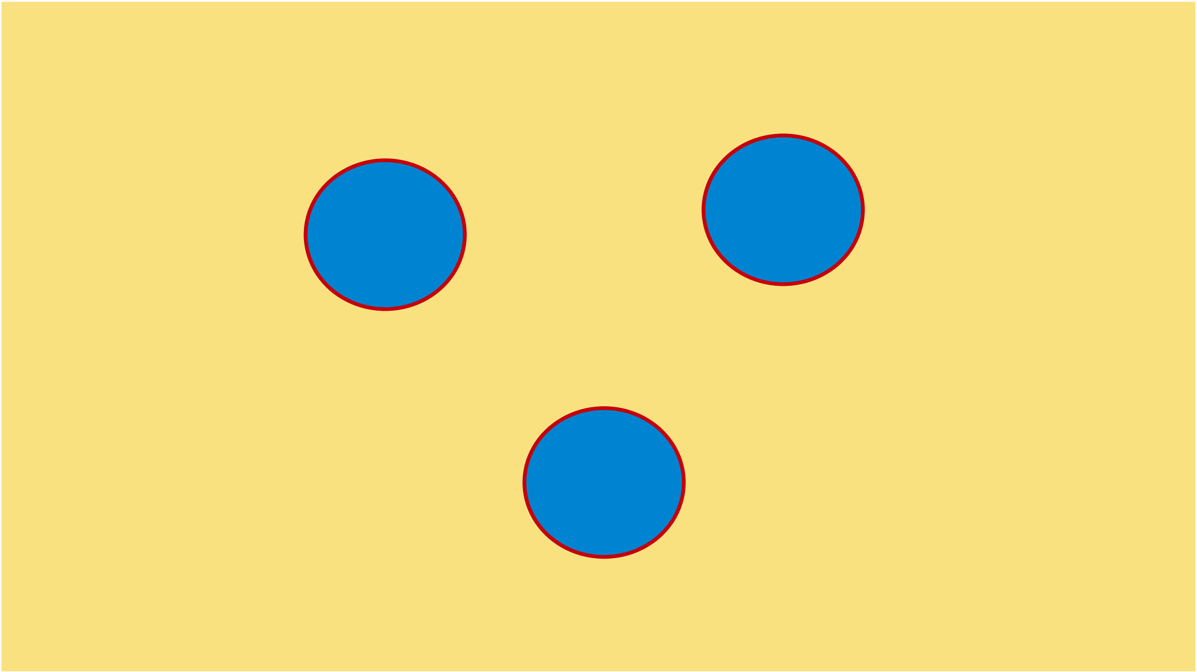
Topological space

(Kuratowski)

Valuation of atomic propositions

$\mathcal{V}(p) \subseteq X$

Topological model: a “coloured” topological space



Satisfaction relation

$$\boxed{\mathcal{M}, x \in X \models \phi}$$

point of the space

set of points **("region"?)**

Satisfaction relation

See J. van Benthem, G. Bezhanishvili, Modal Logics of Space, Handbook of Spatial Logics Chapter 5

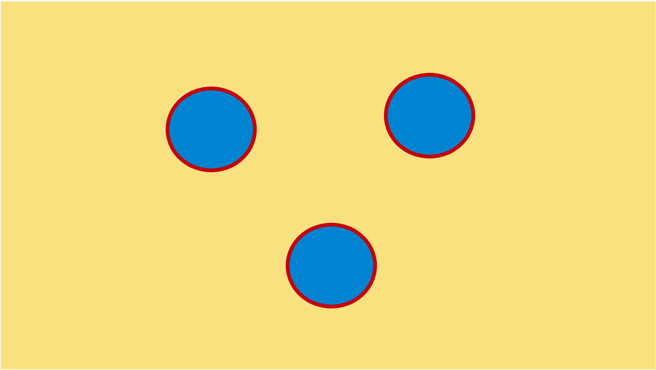
$$
\begin{array}{rcl}
[\![p]\!] & = & v(p) \\
[\![\top]\!] & = & X \\
[\![\neg\phi]\!] & = & \overline{[\![\phi]\!]} \\
[\![\phi \wedge \psi]\!] & = & [\![\phi]\!] \cap [\![\psi]\!] \\
[\![\diamond\phi]\!] & = & \mathcal{C}([\![\phi]\!]) \\
[\![\Box\phi]\!] & = & \mathcal{I}([\![\phi]\!])
\end{array}
$$

Closure

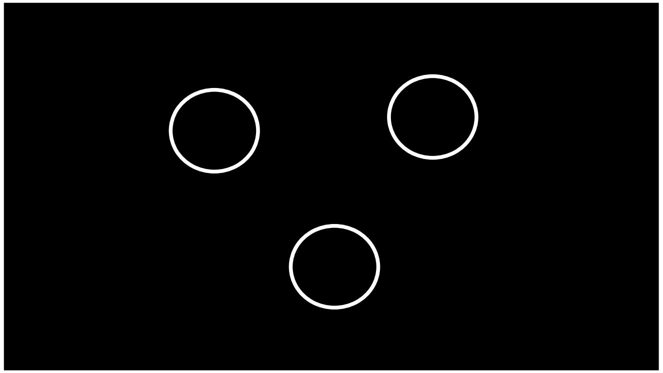
Interior (dual)

$\mathcal{I}(A) = \overline{\mathcal{C}(\overline{A})}$

Topological model: a “coloured” topological space

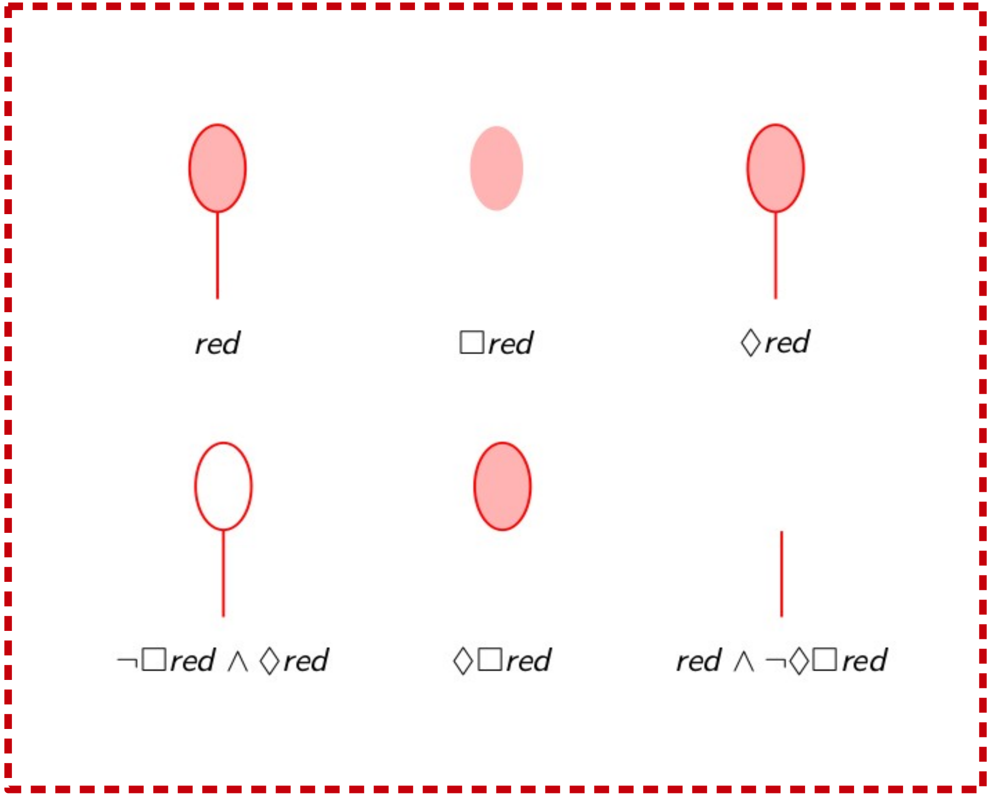


Model

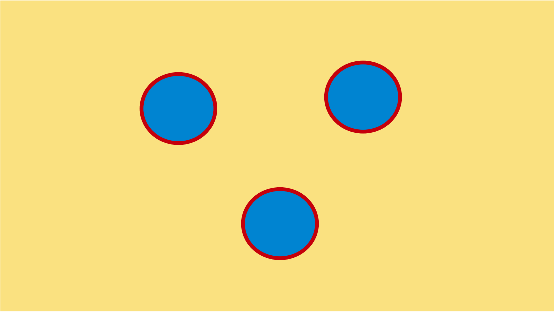


$\diamond blue \wedge \neg blue$

Example (Ch. 5, Handbook of Spatial Logics)



Logical equivalence, bisimilarity*



Model based on Euclidean Plane

Model based on Euclidean Line

Closed interval

* Hennessy-Milner theorem! (See Handbook of Spatial Logics, Chapter 5).

Logics & Computer Science Matters

Decidability of validity
(what formulas are theorems?)

Axioms
(how to prove theorems in the logic?)

Completeness
(are all theorems provable?)

Finite(ly representable) models
(how to run algorithms?)

Model checking
(how to check satisfaction?)

Case studies
(when can one use the logic?)

Eduard Čech

ELSEVIER
Theoretical Computer Science 151 (1995) 257–276
Theoretical Computer Science

Semi-metrics, closure spaces and digital topology

M.B. Smyth
Department of Computing, Imperial College, London SW7 2BZ

Abstract

that they
to digital
ppropriate

ch closure
ch can be
it interval

ELSEVIER
Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®
Theoretical Computer Science 305 (2003) 111 – 134
Theoretical Computer Science
www.elsevier.com/locate/tcs

A generalized topological view of motion in discrete space

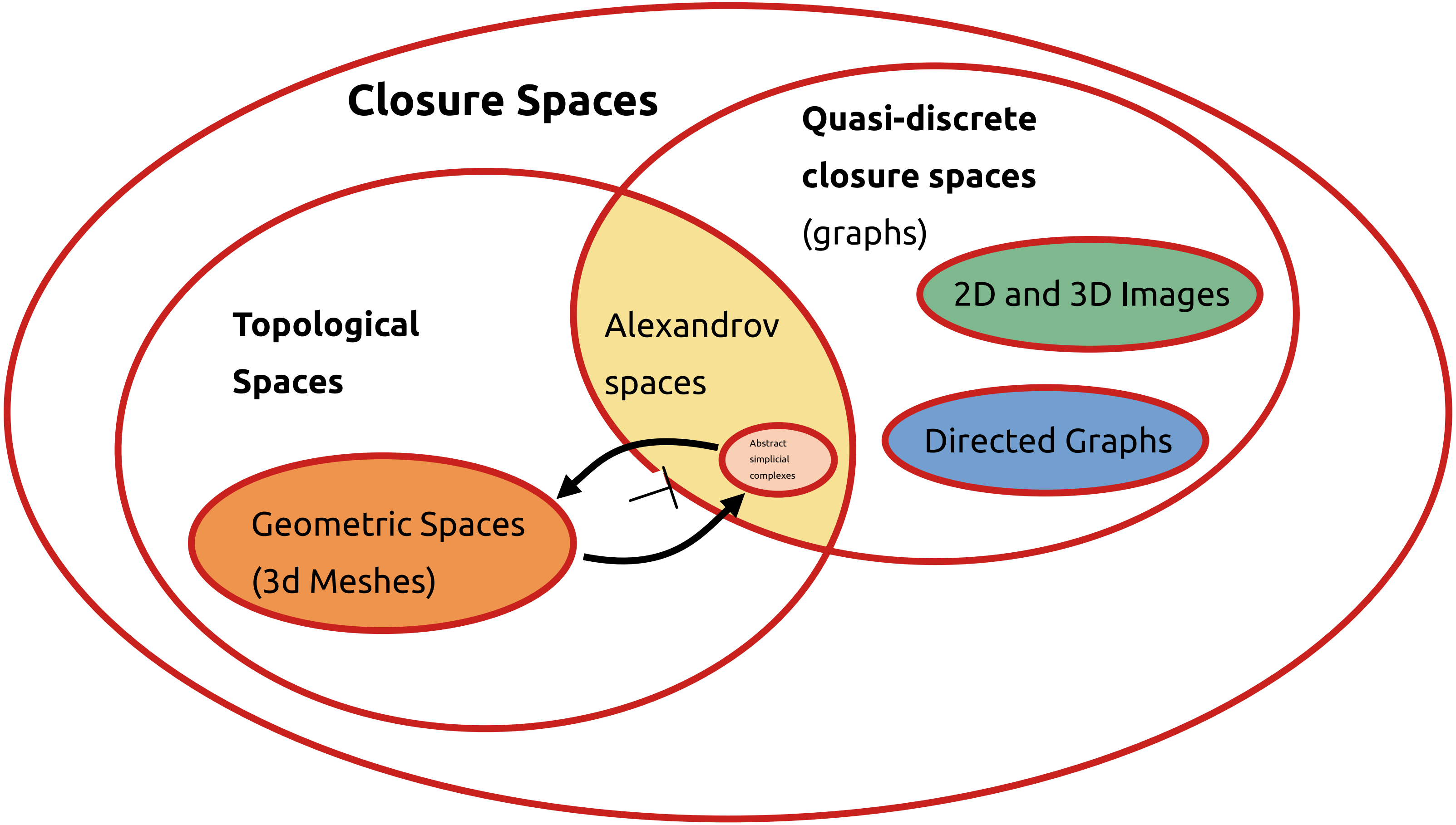
Antony Galton
School of Engineering and Computer Science, University of Exeter, Exeter, Devon EX4 4PT, UK

Part 2

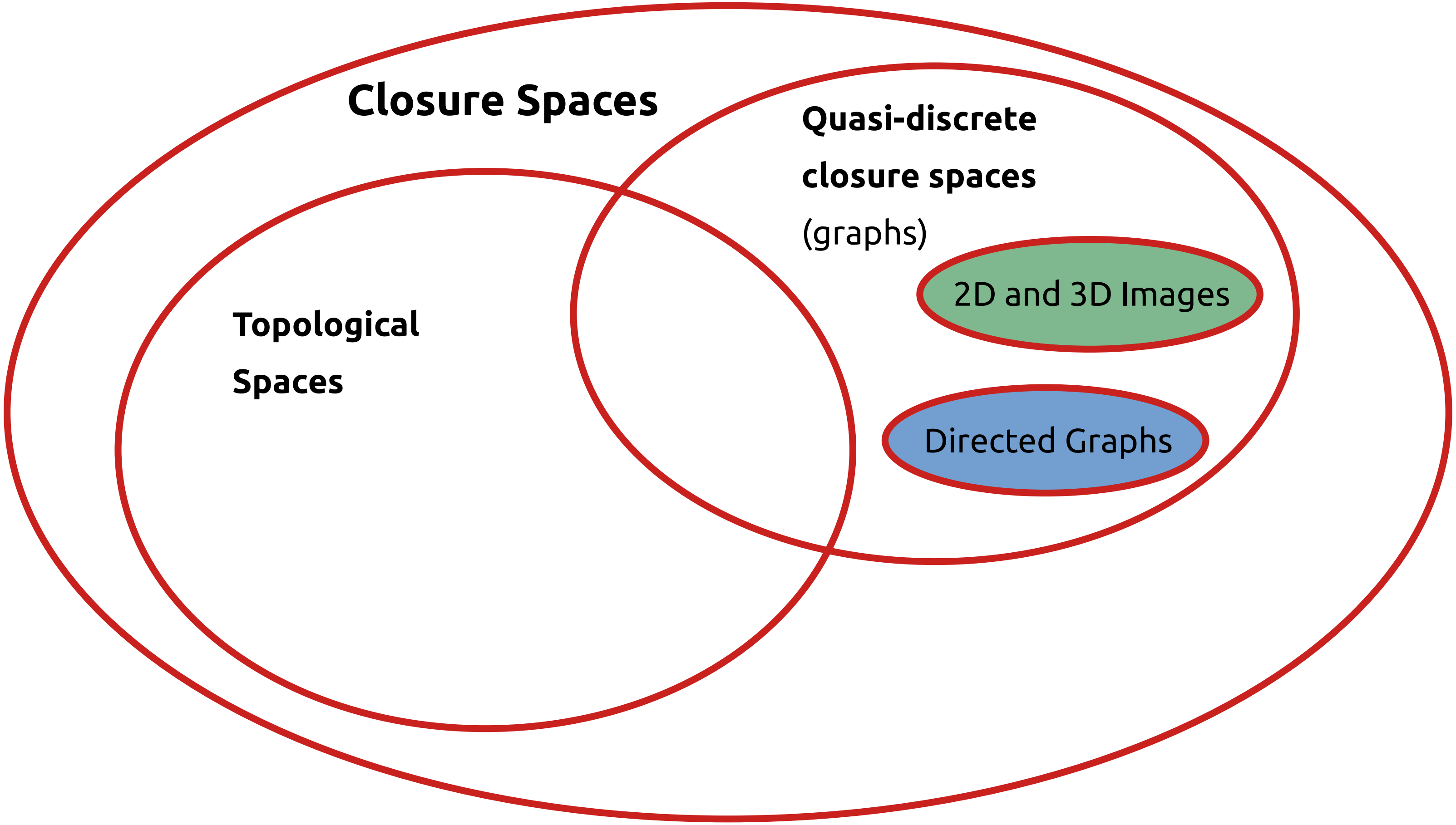
Spatial Model Checking

- Čhech closure spaces
- SLCS: the Spatial Logic of Closure spaces (reachability, surrounded, distance…)
- Model checkers: Topochecker & VoxLogicA
- Optimizations

Goal of Part 2



Goal of Part 2, simplified



Čhech Closure Spaces

A **closure** space is a pair $(X, \mathcal{C})$

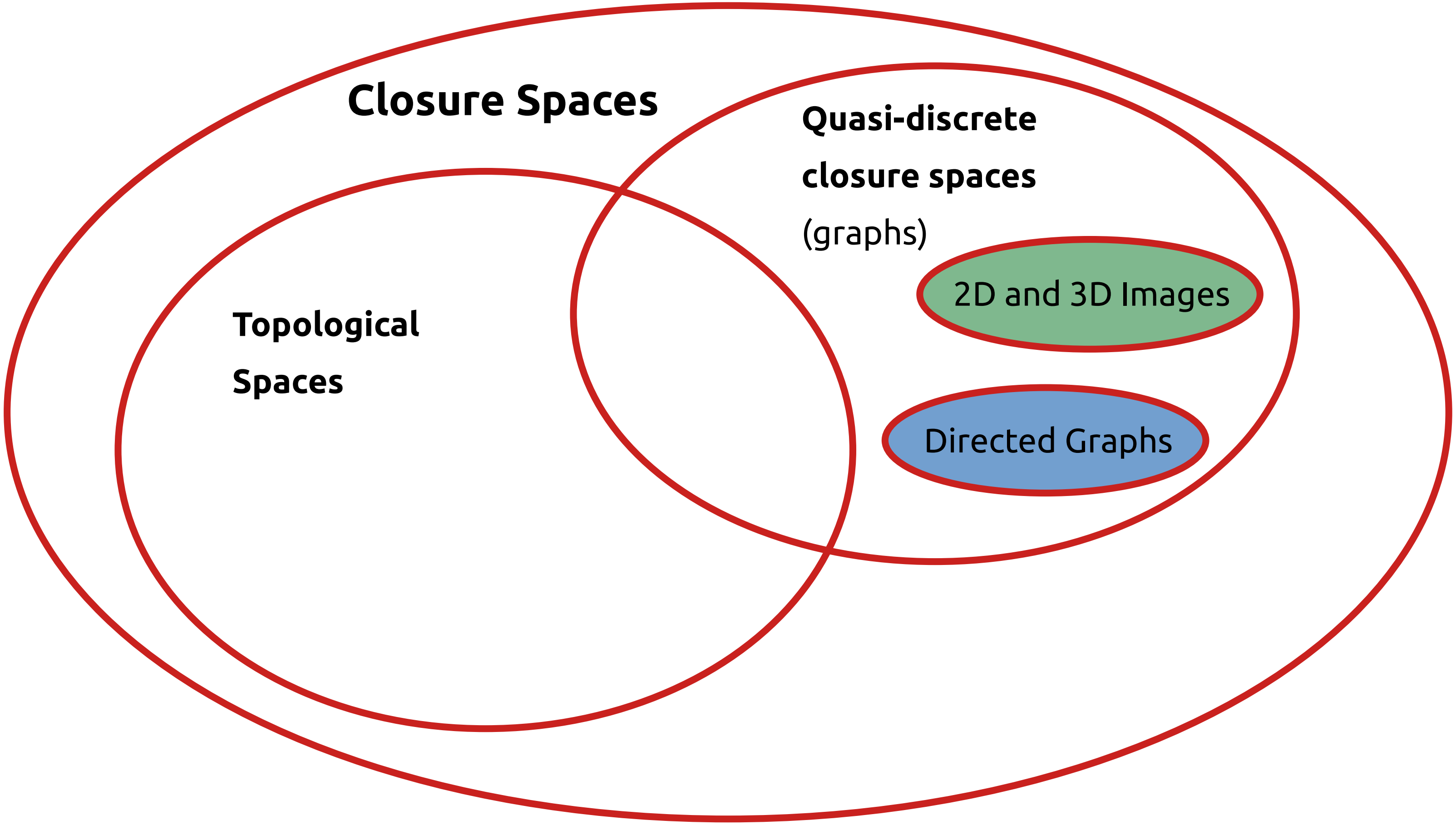
- X is a set
- $\mathcal{C} : \mathcal{P}(X) \to \mathcal{P}(X)$

- $\mathcal{C}(\emptyset) = \emptyset$
- $A \subseteq \mathcal{C}(A)$
- $\mathcal{C}(A \cup B) = \mathcal{C}(A) \cup \mathcal{C}(B)$
- $\cancel{\mathcal{C}(\mathcal{C}(A)) = \mathcal{C}(A)}$

Continuous function

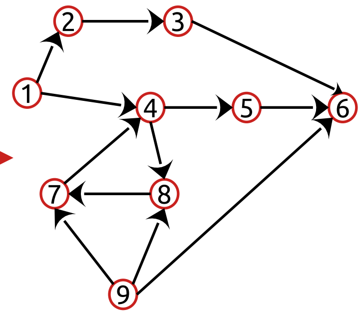
$$f : X_1 \to X_2 \text{ with } f(\mathcal{C}_1(A)) \subseteq \mathcal{C}_2(f(A))$$

Goal of Part 2, simplified



Quasi-discrete closure spaces

Closure is **generated by a relation**



$$R \subseteq X \times X$$

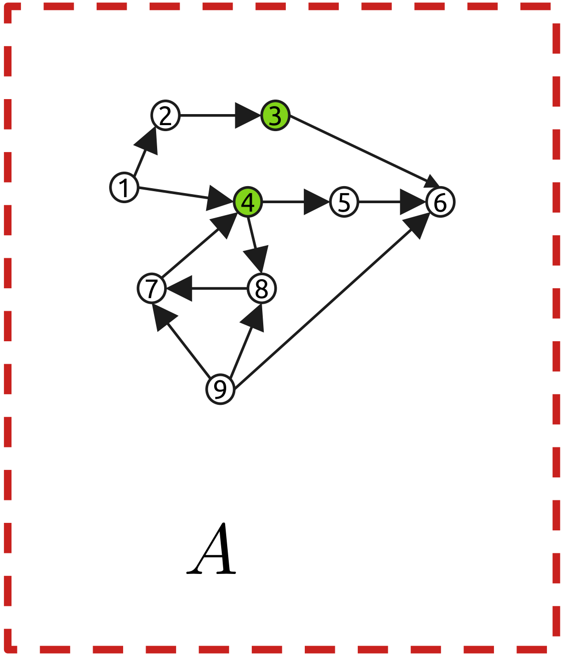
$$\mathcal{C}_R(A) = A \cup \{x \in X \mid \exists a \in A.(a, x) \in R\}$$

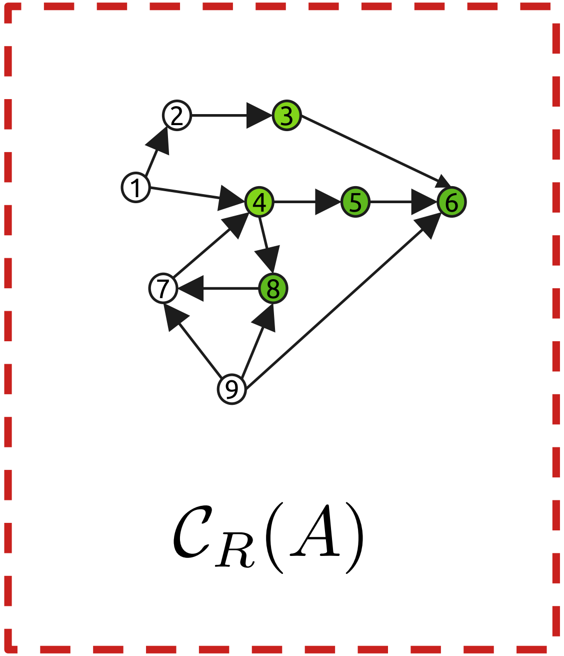
EQUIVALENT: closure of a set **determined by the singletons**: $\mathcal{C}(A) = \bigcup_{a \in A} \mathcal{C}(\{a\})$

[EQUIVALENT: **minimal neighbourhoods exist**], cf. *Alexandrov* spaces

Closure in quasi-discrete closure spaces

$$\mathcal{C}_R(A) = A \cup \{x \in X \mid \exists a \in A.(a, x) \in R\}$$





Clearly not idempotent

Example: graph paths as a continuous function from a Q.D.C.S.

$$[0,1) \to (X, \mathcal{C}_R)$$

Continuous path

$$(\mathbb{N}, \mathcal{C}_{Succ}) \to (X, \mathcal{C}_R)$$

Graph theoretical path

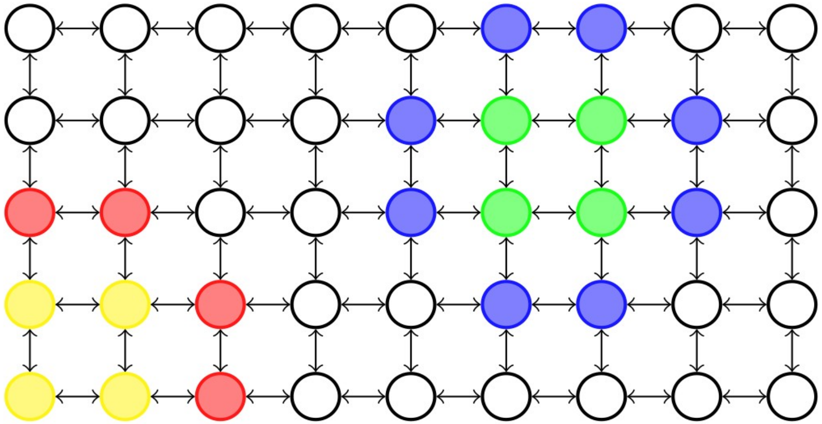
Closure model

$$\mathcal{M} = (\underbrace{(X, \mathcal{C})}_{\text{Closure space}}, \mathcal{V})$$

Valuation of atomic propositions

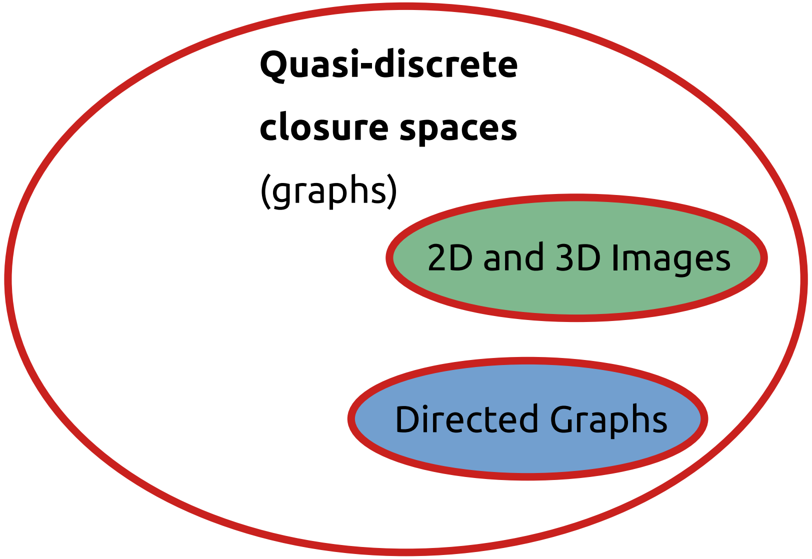
$\mathcal{V}(p) \subseteq X$

Closure model: a “coloured” closure space



images are graphs

regular grids with chosen adjacency



Syntax of the **Spatial Logic of Closure Spaces (SLCS)**

See V. Ciancia, D. Latella, M. Loreti, M. Massink, Model Checking Spatial Logics for Closure Spaces, LMCS 2016

$$
\begin{array}{rcll}
\Phi & ::= & p & \text{atomic proposition} \\
 & \mid & \top & \text{true} \\
 & \mid & \neg\Phi & \text{negation} \\
 & \mid & \Phi \wedge \Phi & \text{conjunction} \\
 & \mid & \mathcal{N}\Phi & \text{closure} \\
 & \mid & \Phi\mathcal{S}\Phi & \text{surrounded}
\end{array}
$$

Satisfaction relation

$$
\begin{aligned}
[\![p]\!] &= v(p) \\
[\![\top]\!] &= X \\
[\![\neg\phi]\!] &= \overline{[\![\phi]\!]} \\
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \cap [\![\psi]\!] \\
[\![\mathcal{N}\phi]\!] &= \mathcal{C}([\![\phi]\!])
\end{aligned}
$$

Surrounded operator

A spatial version of

"Until" operator

$$\mathcal{M}, x \models \phi \, \mathcal{S} \, \psi$$

"To get out of ϕ, one must first pass by ψ"

Satisfaction relation for surrounded

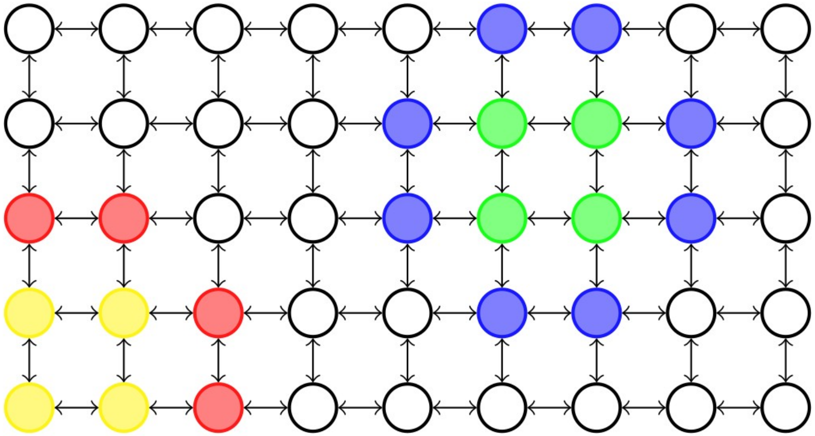
$$\mathcal{M}, x \models \phi \mathcal{S} \psi$$

if and only if

$$\forall p : path(x). \forall \ell.$$

$$\boxed{\mathcal{M}, p_\ell \models \neg\phi \implies \exists k \in (0, \ell]. \mathcal{M}, p_k \models \psi}$$

Example



$$yellow \; \mathcal{S} \; red$$

$$yellow \wedge \mathcal{N} \; red$$

$$green \; \mathcal{S} \; violet$$

$$white \; \mathcal{S} \; red$$

Derived operators

$$
\begin{aligned}
\bot &\triangleq \neg\top \\
\mathcal{I}\phi &\triangleq \neg(\mathcal{N}\neg\phi) \\
\delta^{-}\phi &\triangleq \phi \wedge (\neg\mathcal{I}\phi) \\
\phi_1 \,\mathcal{R}\, \phi_2 &\triangleq \neg((\neg\phi_2)\,\mathcal{S}(\neg\phi_1))
\end{aligned}
\qquad
\begin{aligned}
\phi_1 \vee \phi_2 &\triangleq \neg(\neg\phi_1 \wedge \neg\phi_2) \\
\delta\phi &\triangleq (\mathcal{N}\phi) \wedge (\neg\mathcal{I}\phi) \\
\delta^{+}\phi &\triangleq (\mathcal{N}\phi) \wedge (\neg\phi) \\
\mathcal{E}\,\phi &\triangleq \phi\,\mathcal{S}\,\bot
\end{aligned}
$$

Duality between "Reachable" and "Surrounded"

$$\phi_1 \mathcal{R} \phi_2 \triangleq \neg((\neg\phi_2)\mathcal{S}(\neg\phi_1))$$

Theorem:

$$\mathcal{M}, x \models \phi_1 \mathcal{R} \phi_2 \iff$$

$$\exists p : path(x). \exists \ell. \mathcal{M}, p_\ell \models \phi_2$$

$$\wedge \forall j. 0 < j \leq \ell \implies \mathcal{M}, p_j \models \phi_1$$

Note: paths could also be all reversed, starting from a "far" point, and ending in **x**

A “Modernised” Syntax*

$$\Phi ::= p \mid \top \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathcal{N}\Phi \mid \underbrace{\overrightarrow{\rho}\,\Phi[\Phi] \mid \overleftarrow{\rho}\,\Phi[\Phi]}_{\text{Two reachability operators (forward \& backward)}} \mid \underbrace{\mathcal{D}_c\Phi}_{\text{Distance modality}}$$

*See also E. Bartocci, L. Bortolussi, M. Loreti, L. Nenzi, “Monitoring mobile and spatially distributed cyber-physical systems” MEMOCODE 2017

Distance operator

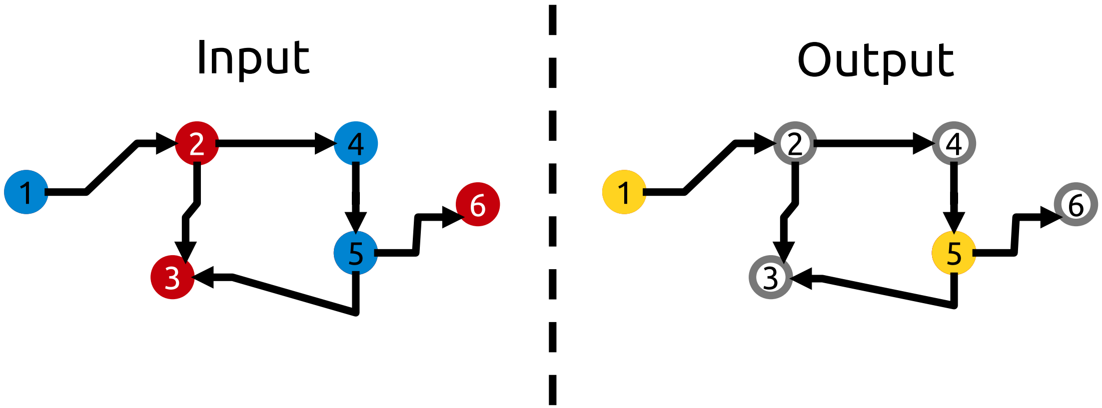
$$\boxed{\mathcal{M}, x \models \mathcal{D}_c\phi \iff c(distance(x, \{y \in X \mid y \models \phi\}))}$$

c is a **constraint**

e.g. $\mathcal{D}_{\leq 3}(\mathcal{N} black)$

"The points distant **at most 3** from the points **near** to a black point"

Global model checking algorithm: **Dynamic Programming**

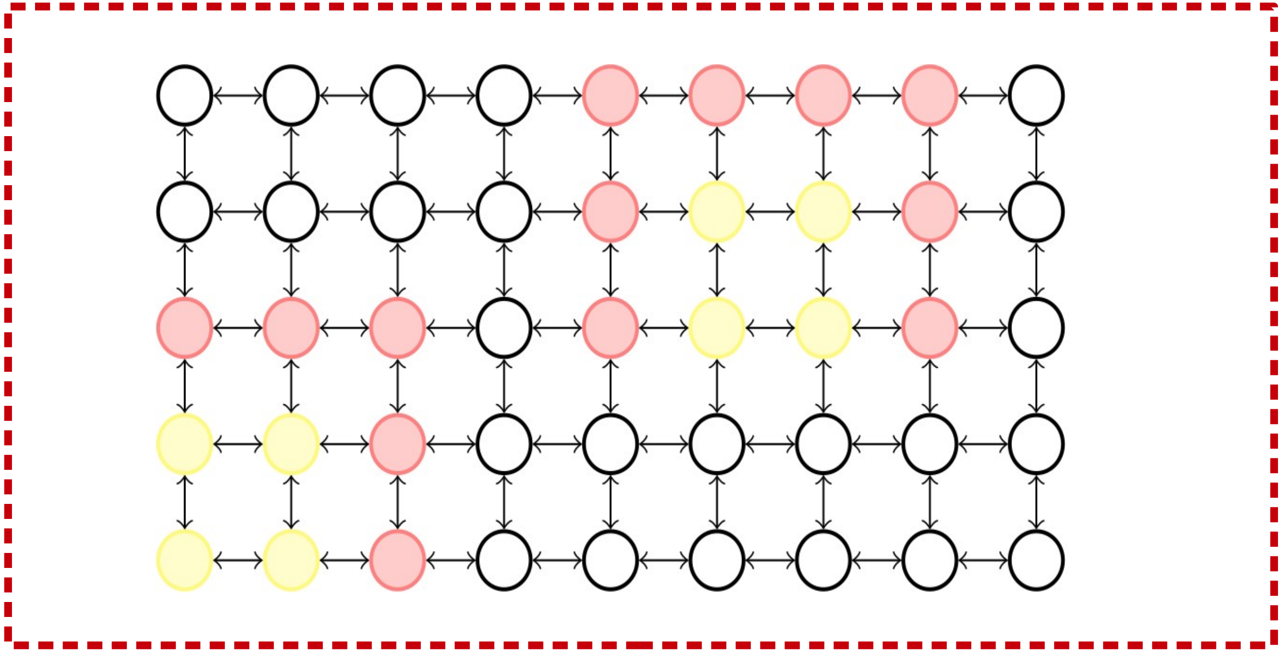


$\boxed{blue \wedge \neg \mathcal{N} red}$

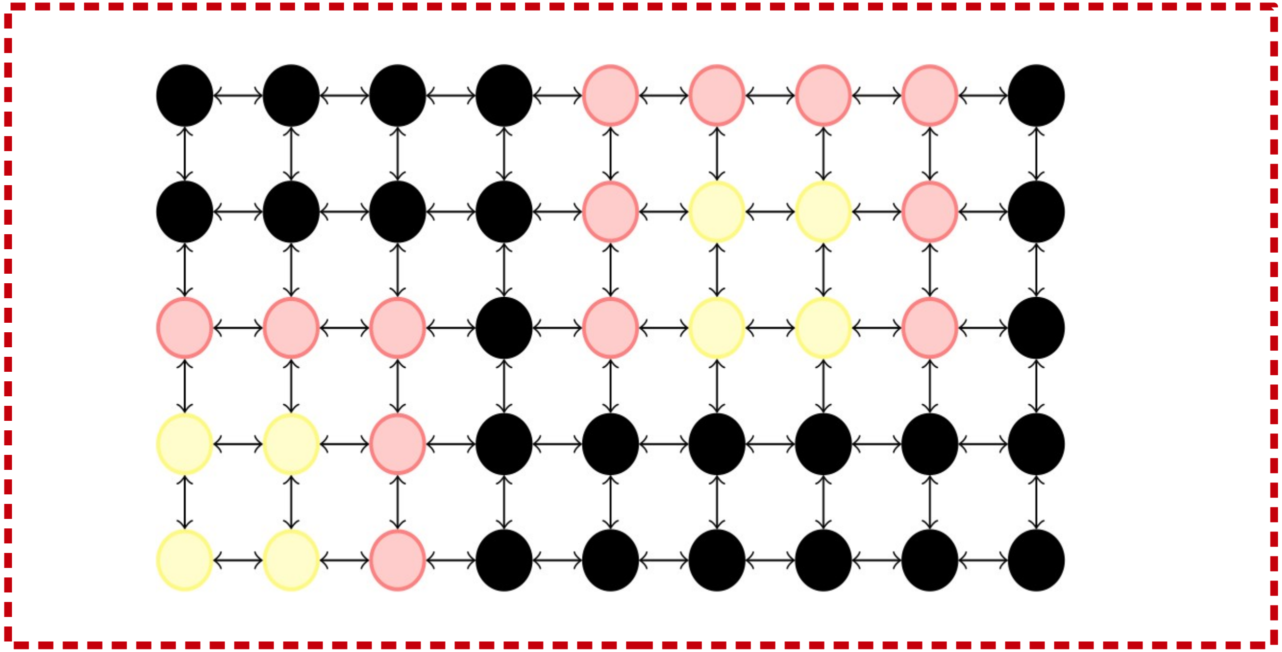
	1	2	3	4	5	6
$blue \wedge \neg \mathcal{N} red$	T	F	F	F	T	F
$\neg \mathcal{N} red$	T	F	F	F	T	T
$\mathcal{N} red$	F	T	T	T	F	F
red	F	T	T	F	F	T
$blue$	T	F	F	T	T	F

Each line is a graph with boolean labelled nodes, or a binary image ("mask")

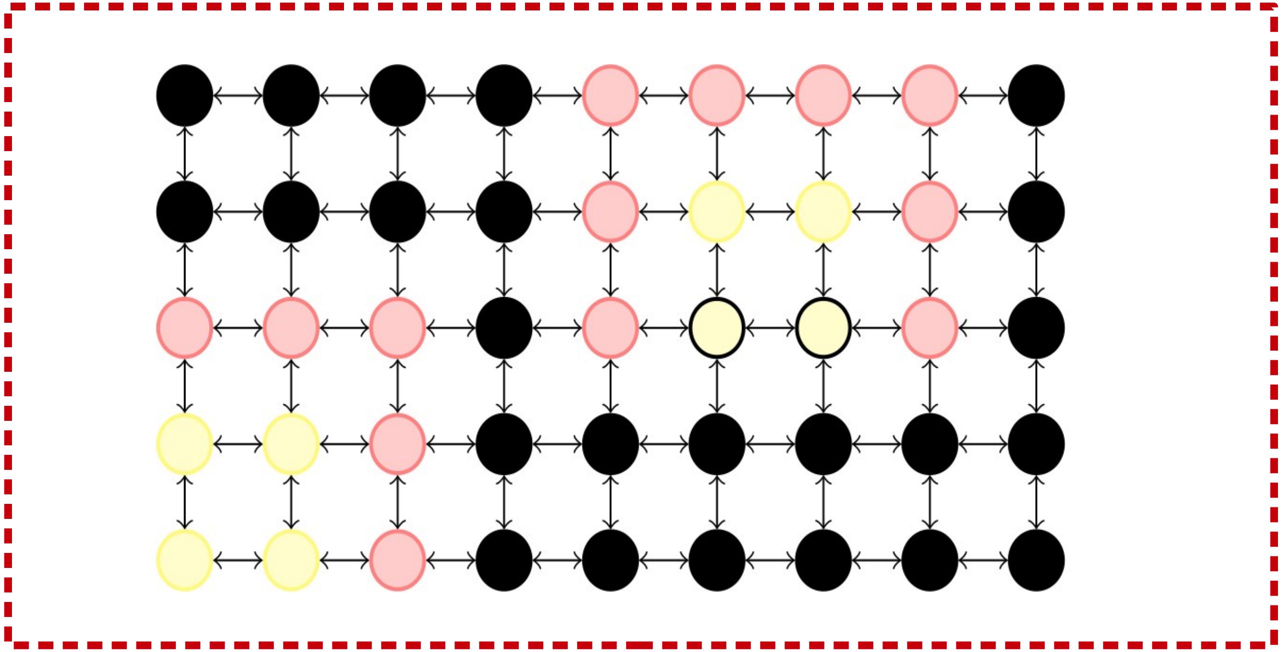
Flooding algorithm for Surrounded



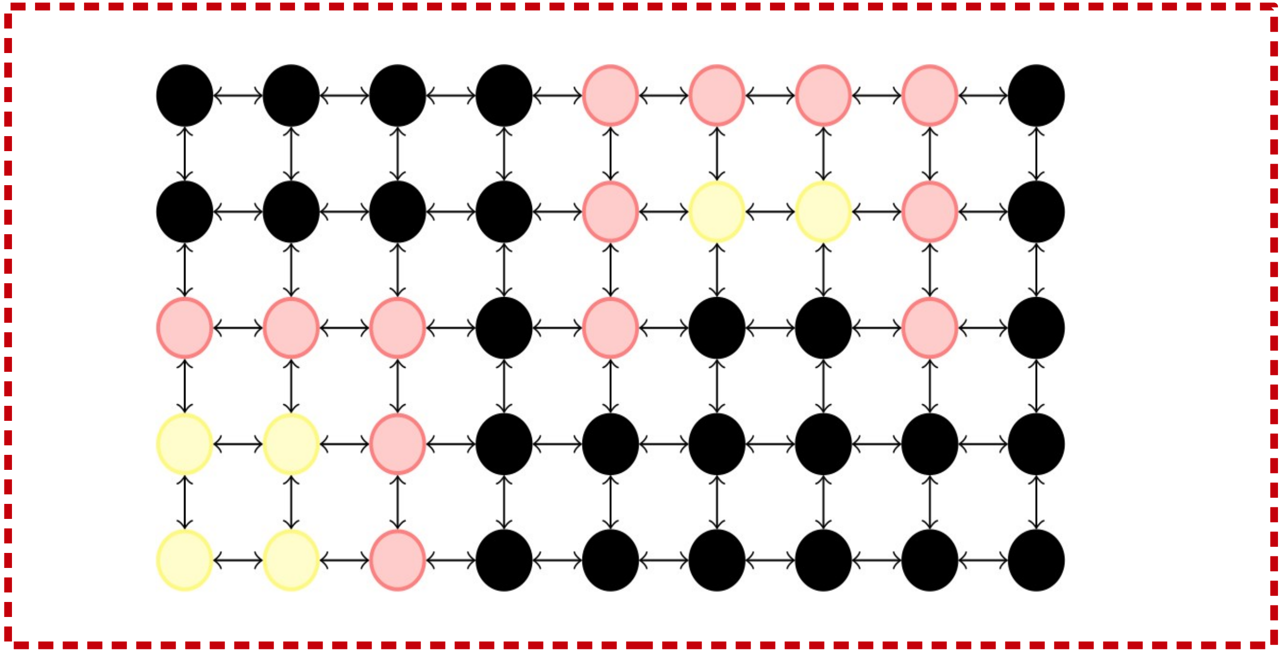
Flooding algorithm for Surrounded



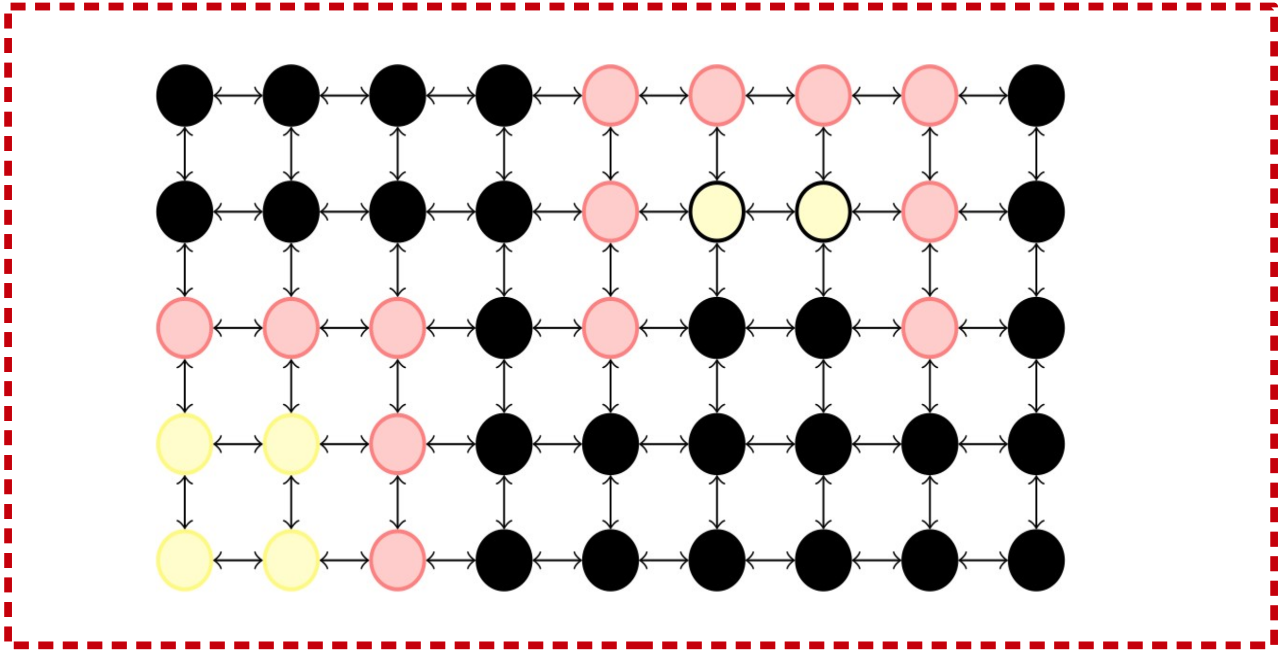
Flooding algorithm for Surrounded



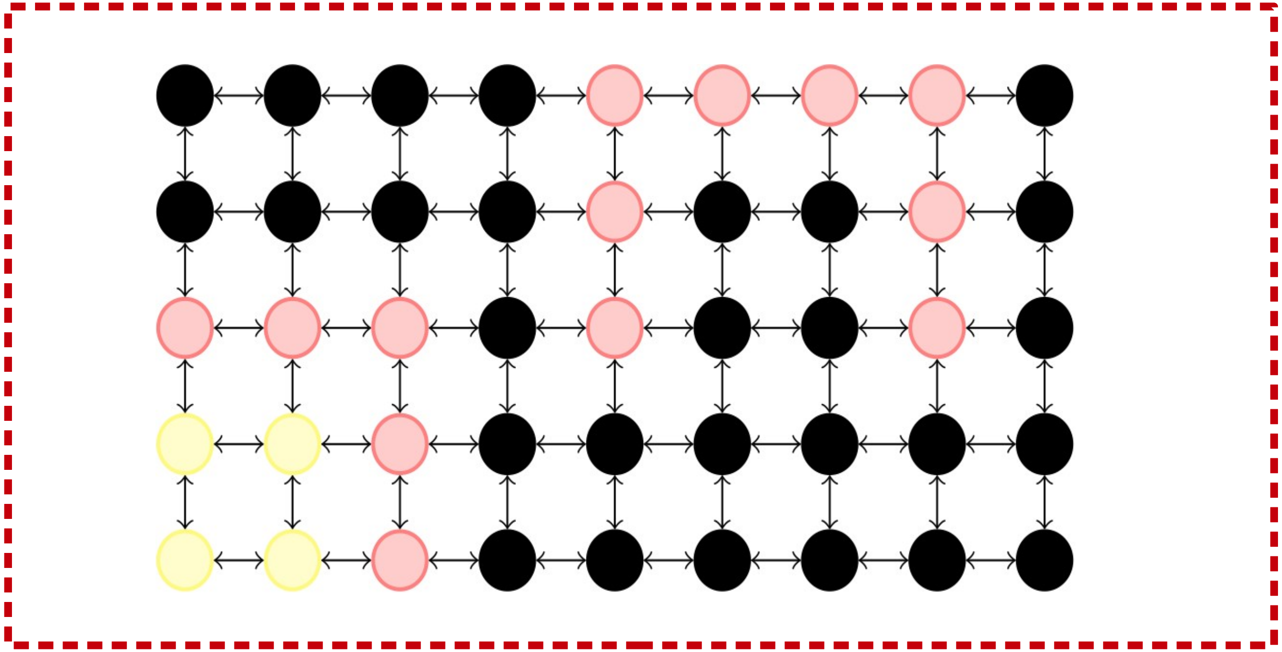
Flooding algorithm for Surrounded



Flooding algorithm for Surrounded

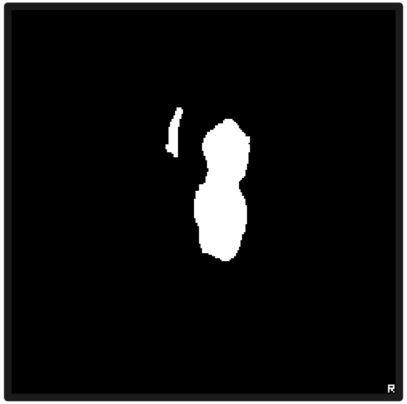


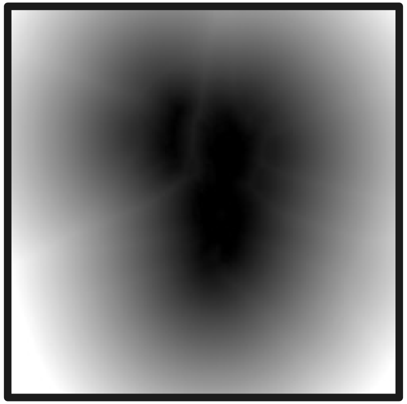
Flooding algorithm for Surrounded



Implementation: distance transform

See G. Belmonte, V. Ciancia, D. Latella, M. Massink, VoxLogicA: A Spatial Model Checker for Declarative Image Analysis, TACAS19





Images: Euclidean Distance Transform **O(n)** (Maurer, 2003)

Graphs: generalised Dijkstra shortest path algorithm **O(n log(n))**

Model checkers

Free and Open Source! See https://github.com/vincenzoml/topochecker & https://github.com/vincenzoml/voxlogica

- **Topochecker**

  graphs, images, spatio-temporal (CTL x SLCS), Ocaml, prototype,
  interface with MultiVeStA* (statistical m.c.),

- **VoxLogicA**

  2D and 3D images, spatial, Fsharp/.Net Core, user-oriented
  state-of-the-art ITK library, multi-core, orders of magnitude faster

*“MultiVeStA: Statistical Model Checking for Discrete Event Simulators”, S. Sebastio, A. Vandin, VALUETOOLS’13 (extended version)
“A framework for quantitative modeling and analysis of highly (re)configurable systems”, M. ter Beek, A. Legay, A. Lluch Lafuente, A. Vandin, IEEE TSE 2020

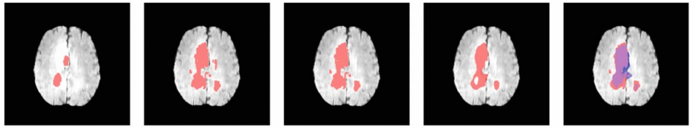
Optimizations in spatial model checking?

	Memoization	Parallel Evaluation	Specialised Libraries	GPU Computing	Caching	Distributed execution
Topochecker	Yes	No	No	No	Yes	No
VoxLogicA	Yes	Yes	Yes	In progress	Planned	Future work

VoxLogicA: some applications

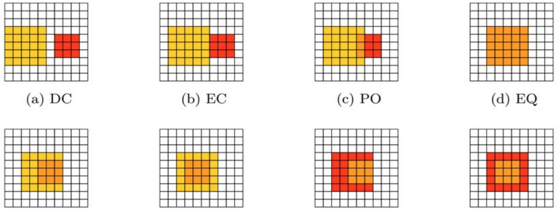
Segmentation of brain tumours

TACAS 2019 -- STTT 2020 -- ESMRMB 2017, 2019



(a) (b) (c) (d) (e)

Region Calculi – LNCS 11665



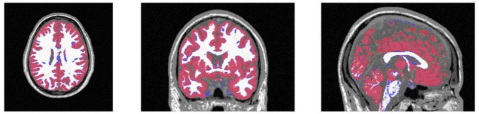
(a) DC (b) EC (c) PO (d) EQ

(e) TPP (f) NTPP (g) TPPi (h) NTPPi

Fig. 6: The eight RCC8D relations.

Segmentation of white and grey matter

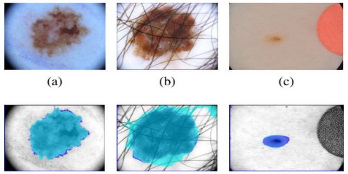
LNCS 11865



axial coronal sagittal

Segmentation of nevi - FORMALISE@ ICSE 2021

joint work with G. Belmonte, G. Broccia, D. Latella, M. Massink



(a) (b) (c)

(d) (e) (f)

Domain-specific language **ImgQL**

joint work with Gina Belmonte, Diego Latella, Mieke Massink (TACAS 2019)

Logical core

+

Imaging primitives

2D and 3D images

(png,bmp,nifti, ...)

+

Function definition

(no recursion)

Domain-specific language **ImgQL**

joint work with Gina Belmonte, Diego Latella, Mieke Massink (TACAS 2019)

$$\Phi ::= p \mid \neg \Phi \mid \Phi_1 \wedge \Phi_2 \mid$$

$$\mid \mathcal{N}\Phi \mid \rho\, \Phi_1[\Phi_2] \mid \mathcal{D}^I \Phi$$

Logical core

+

Imaging primitives

2D and 3D images

(png,bmp,nifti, ...)

+

Function definition

(no recursion)

Domain-specific language **ImgQL**

joint work with Gina Belmonte, Diego Latella, Mieke Massink (TACAS 2019)

$$\Phi ::= p \mid \neg \Phi \mid \Phi_1 \wedge \Phi_2 \mid$$
$$\mid \mathcal{N}\Phi \mid \rho\, \Phi_1[\Phi_2] \mid \mathcal{D}^I \Phi$$

Logical core

+

Imaging primitives

2D and 3D images

(png,bmp,nifti, ...)

+

Function definition

(no recursion)

Colour & thresholds

Normalization

Texture analysis

Volume, max, min (global)

...

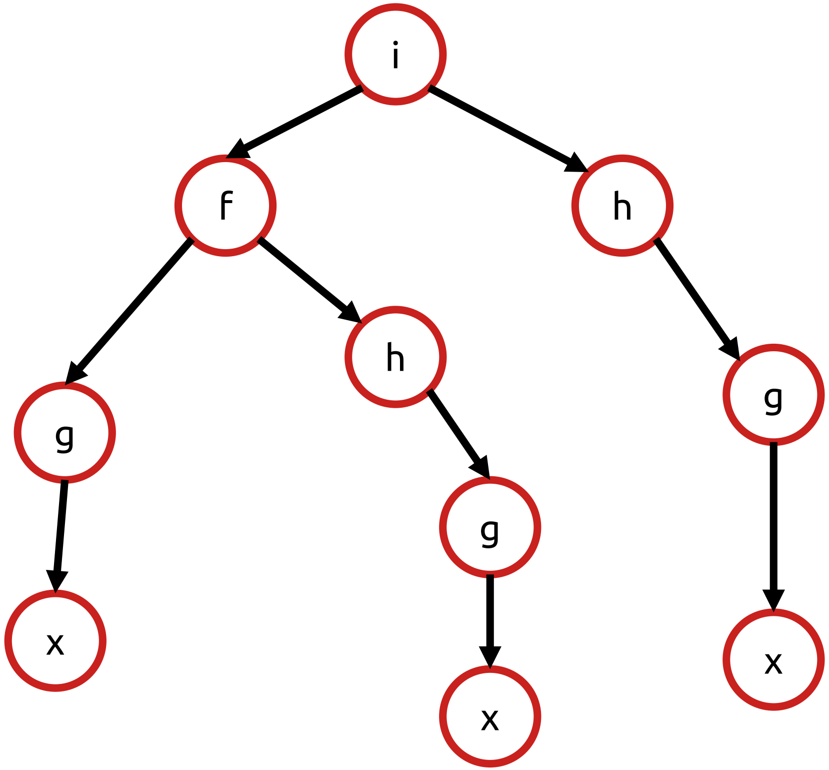
Memoization: never compute anything twice

Syntactic approach:

- **“Pre-interpretation”** (~ abstract interpretation)
- The syntax tree is **progressively hashed** at node creation time
- Semantic domain: directed acyclic graphs

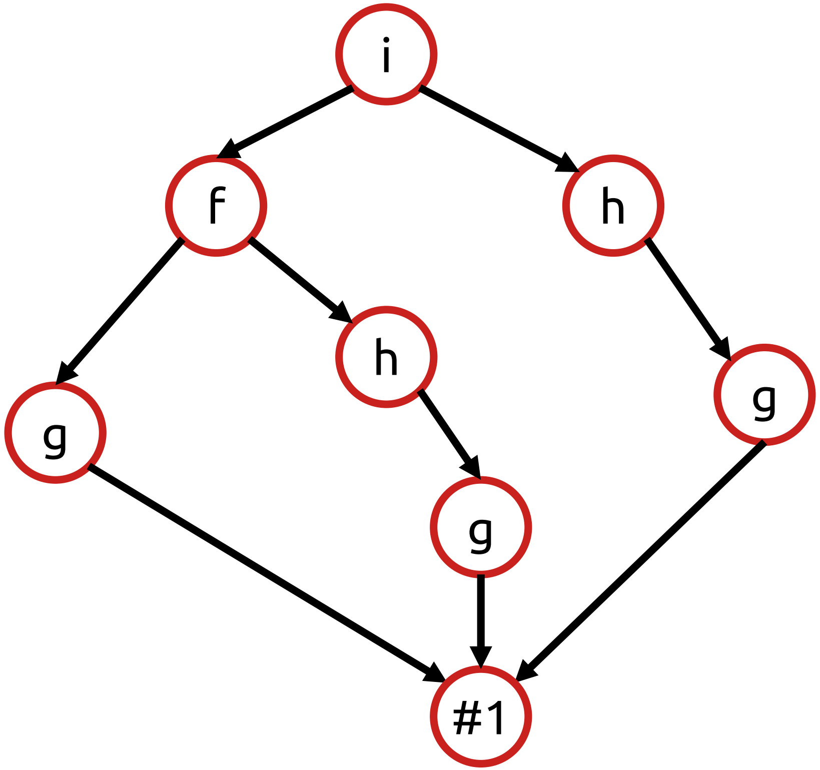
Directed Acyclic Graph with maximal sharing.

Syntax tree (never represented in memory!)



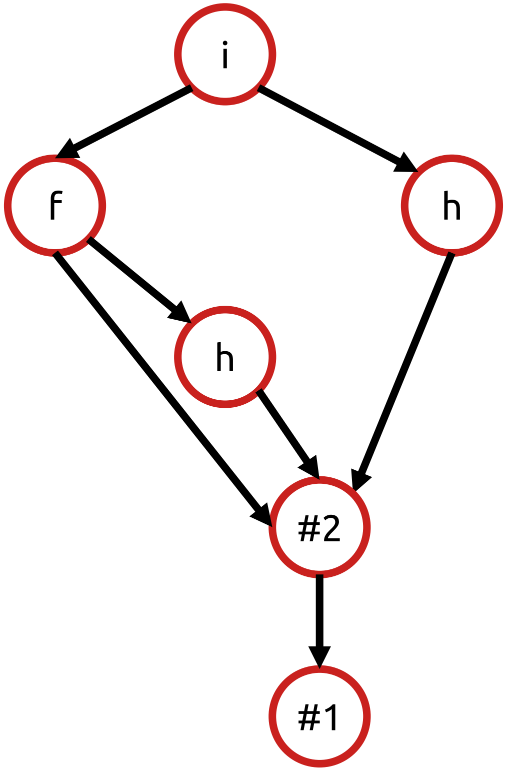
i(f(g(x),h(g(x)),h(g(x)))))

Simplification



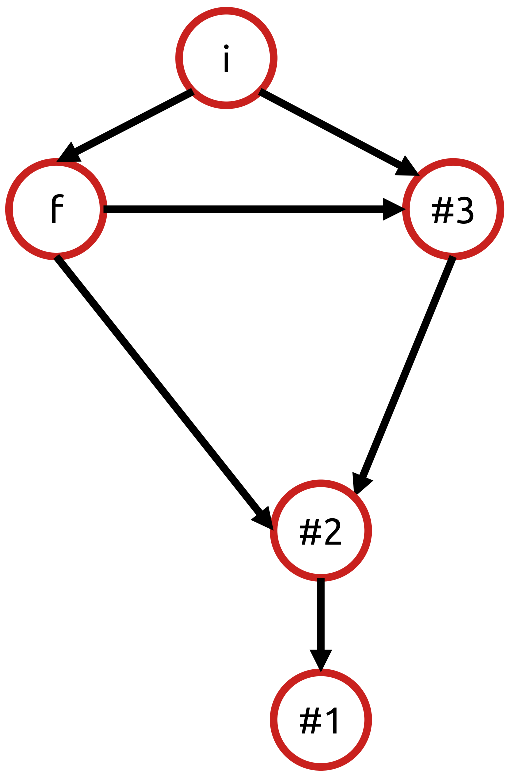
i(f(g(#1),h(g(#1)),h(g(#1))))

Simplification



i(f(#2,h(#2),h(#2))))

Maximal Sharing



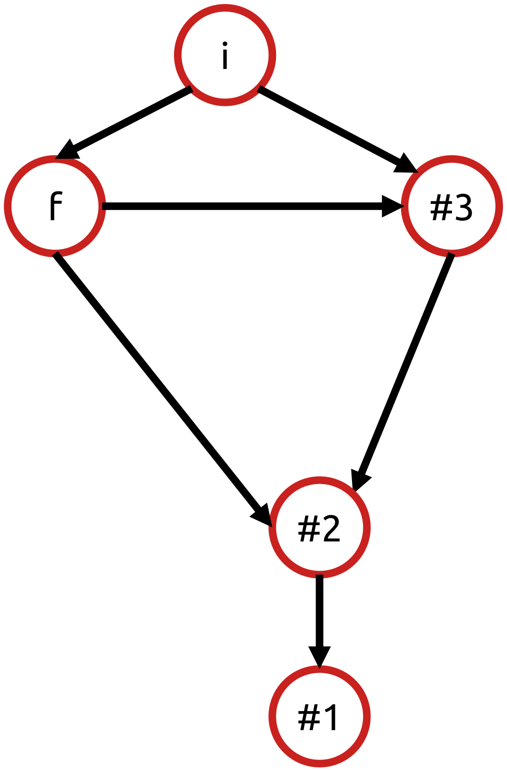
i(f(#2,#3,#3))

Parallel execution

Run the tasks in **parallel** on multiple cores

Each task is **computationally intensive**
(thus, the approach is advantageous!)

Dynamic scheduling
state-of-the-art FSharp library HOPAC



VoxLogicA-GPU:

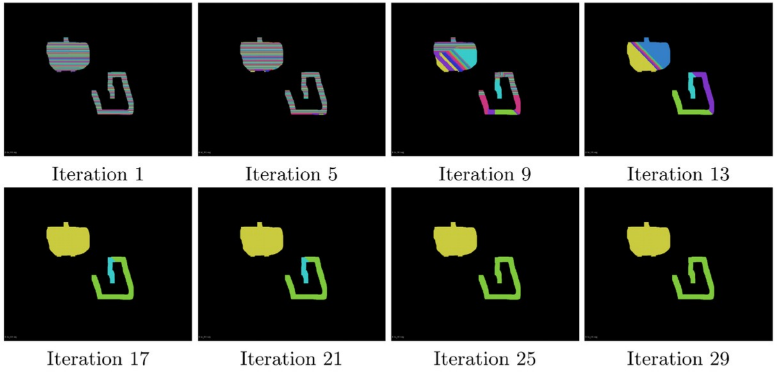
joint work with Laura Bussi, Fabio Gadducci (FORTE 2021)

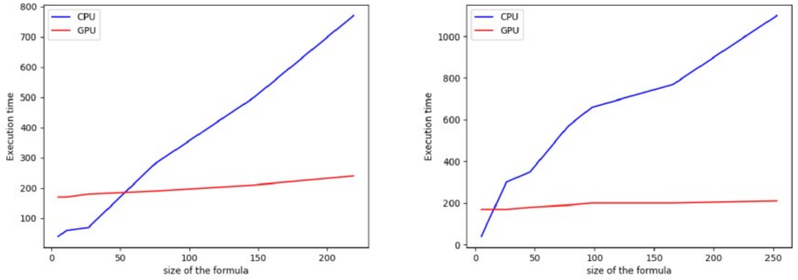
Massively parallel implementation.

Reachability via Connected Components.

Consistent speed up.

Garbage collection.





VoxLogicA-GPU: challenges

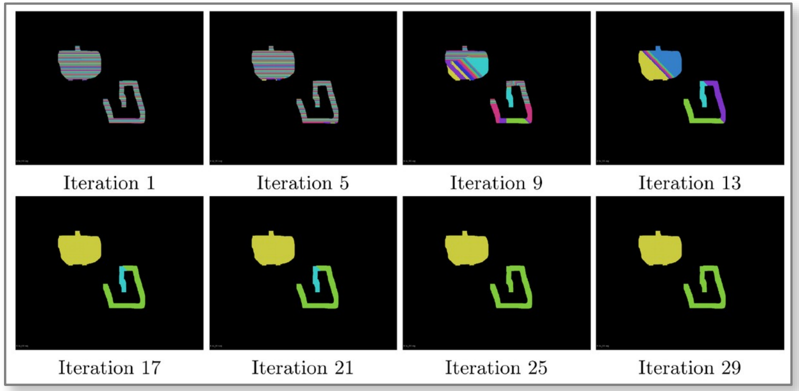
- Minimize transfers between CPU and GPU
- Minimize on-CPU operations (extremely expensive compared to CPU)
- Find pixel-parallel implementations of primitive operations
- Specialised Garbage Collection methods

Performance, garbage collection

No. of Tasks	CPU	GPU		GPU-GC	
	Time	Time	Speed-up	Time	Speed-up
11	410ms	190ms	2.15	200ms	2.05
35	1470ms	190ms	7.73	230ms	6.39
67	1800ms	190ms	9.47	230ms	7.82
195	8200ms	200ms	41.00	320ms	25.62
259	10900ms	210ms	51.90	360ms	30.27
1027	43600ms	350ms	124.57	980ms	44.48
4099	174600ms	Out of memory	-	4100ms	42.58
8195	479000ms	Out of memory	-	12000ms	39.91

Connected components in GPU

- **Pointer-jumping** algorithm
  + “reconciliation step”
- ~ **log(n)** parallel iterations
- Balances **simplicity & efficiency**
  10-20x speedup



`Termination in a few iterations on a 2048x2048 image (4Mpixels!)`

JOURNAL OF ALGORITHMS 3, 57–67 (1982)

An $O(\log n)$ Parallel Connectivity Algorithm

YOSSI SHILOACH
IBM Israel Scientific Center, Haifa, Israel
AND
UZI VISHKIN
Computer Science Department, Technion–Israel Institute of Technology, Haifa, Israel

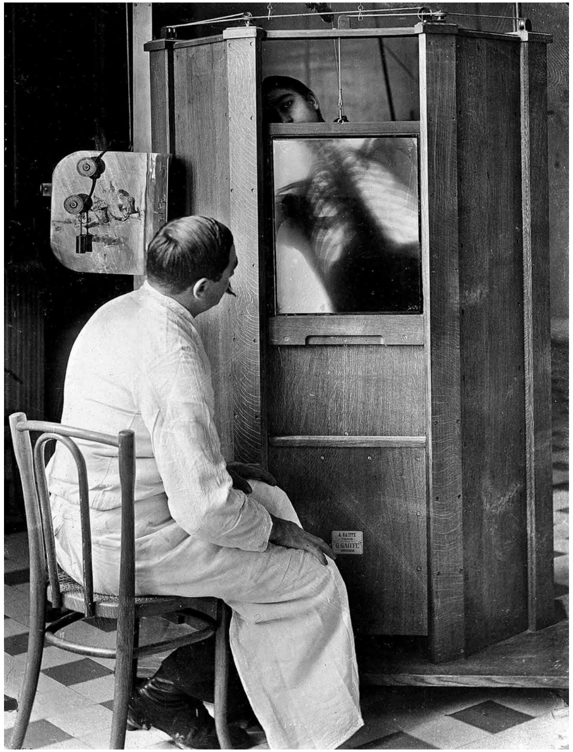
Algorithm 1: Pseudocode for connected components labelling

```
initialization(start: image of bool, output: image of int × int)
    // parallel for on GPU
    for (i, j) ∈ Coords do
        if start(i,j) then
            output(i, j) = (i, j) // null otherwise
mainIteration(start: image of bool, input, output: image of int × int)
    // parallel for on GPU
    for (i, j) ∈ Coords do
        if start(i,j) then
            (i', j') = input(i, j) // pointer jumping
            output(i, j) = maxNeighbour(input, i', j')
reconnect(start: image of bool, input, output: image of int × int)
    // parallel for on GPU
    for (i, j) ∈ Coords do
        if start(i,j) then
            (i', j') = input(i, j)
            (a, b) = maxNeighbour(input, i, j)
            (c, d) = input(i', j')
            if (a, b) > (c, d) then
                output(i', j') = (a, b) // Requires atomic write
```

Part 3

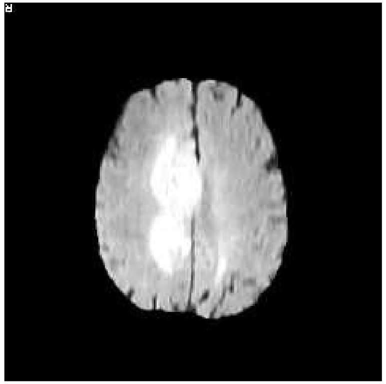
Medical Imaging & VoxLogicA

- A specialised model checker for images: VoxLogicA
- Example: brain tumour segmentation
- More case studies: nevi, white/grey matter

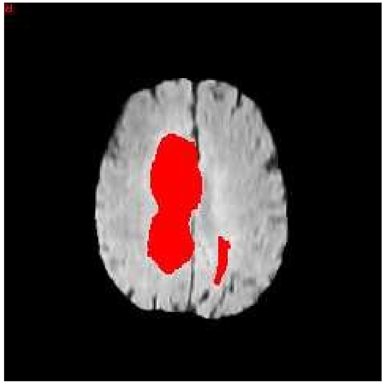


Dr. Maxime Menard’s radiology department,Hôpital Cochin, Paris 1914

Example from Radiotherapy: **Brain tumour segmentation**



Input

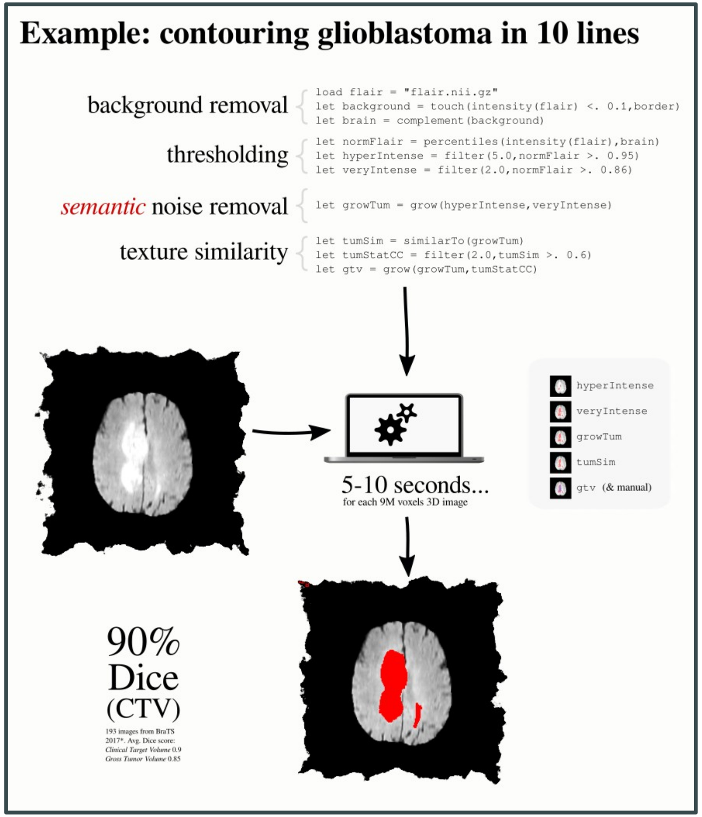


Output

Usually 3D images
(multiple slices)

- Highly specialised working time (hours, per case)
- Bottleneck in radiotherapy planning
- Precise guidelines / protocols to follow

Tumour delineation for radiotherapy using VoxLogicA



Accuracy:

in par with humans

and best-in-class machine learning

Validation:

circa 200 cases, ground truth avaliable

MICCAI-BraTS Challenge 2017 dataset

Speed:

<5 seconds for each 3d image (9 milion voxels)

on a intel Core-I7 desktop computer

Can we automate contouring?

Yes! State of the art is promising!

(hint: uses machine learning)

But rarely used in clinical & research workflows

Why?

A few reasons why

- Accountability.
- Compliance to protocols and guidelines (a case for Formal Methods!).
- Quality assurance.
- Innovative ideas do not have training data.

State of the art

Menze, B.H.e.a.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Transactions on Medical Imaging 2015

Not so easy

Yearly MICCAI-BraTS challenge

Since 2013

2017: **circa 50 papers**, all machine learning

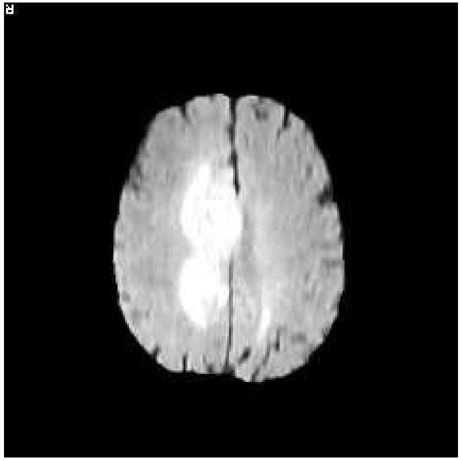
Some excellent results; but also **bad results**

We can do it in three slides...

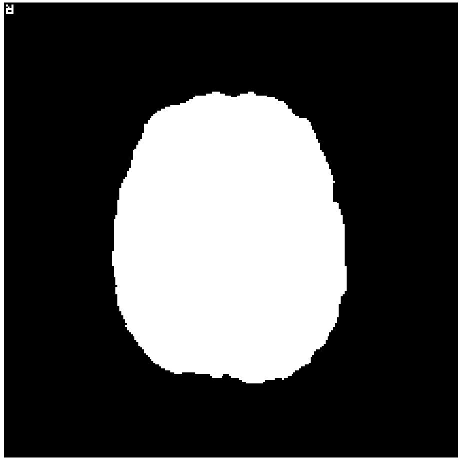
Step 1: Background removal

The background is the dark area that **touches the border**

The brain is the rest of the image



intensity

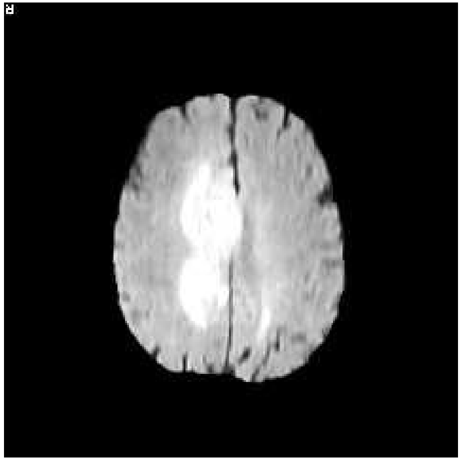


brain

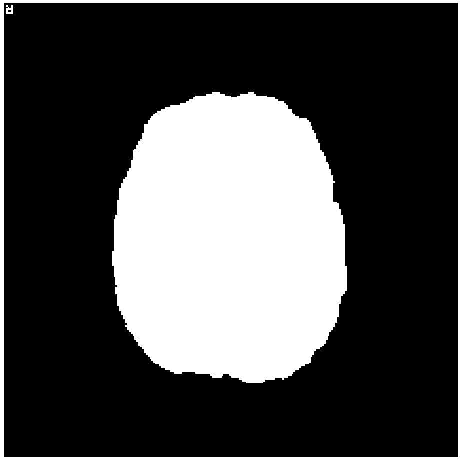
Step 1: Background removal

```
let background = touch(intensity <. 0.1, border)

let brain = !background
```



intensity

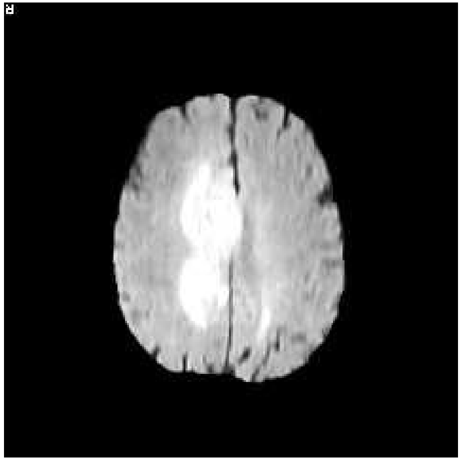


brain

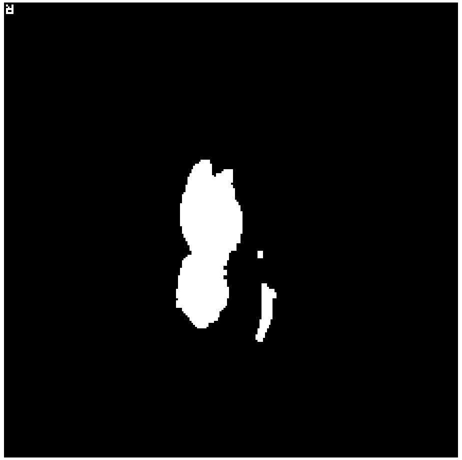
Step 2: Thresholding

Hyperintense areas of the brain belong to the tumour

Very intense areas sometimes belong to the tumour



intensity



veryIntense

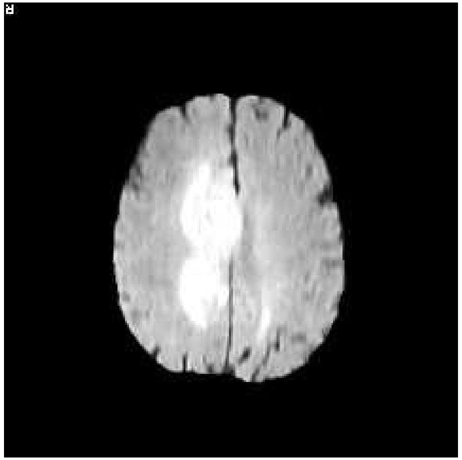
hyperIntense

Step 2: Thresholding

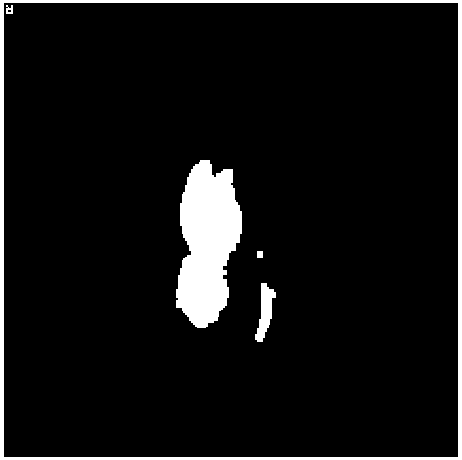
```
let normIntensity = percentiles(intensity,brain)

let hyperIntense = filter(5.0,intensity >. 0.95)

let veryIntense = filter(2.0,intensity >. 0.85)
```



intensity



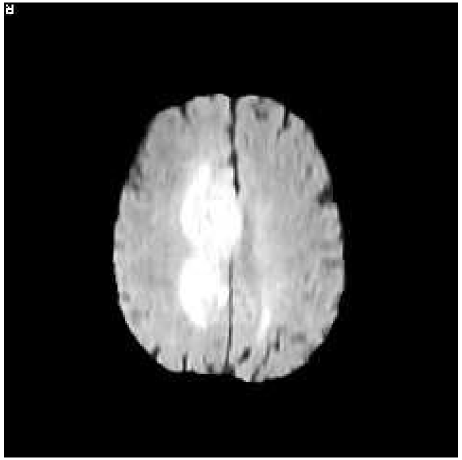
veryIntense

hyperIntense

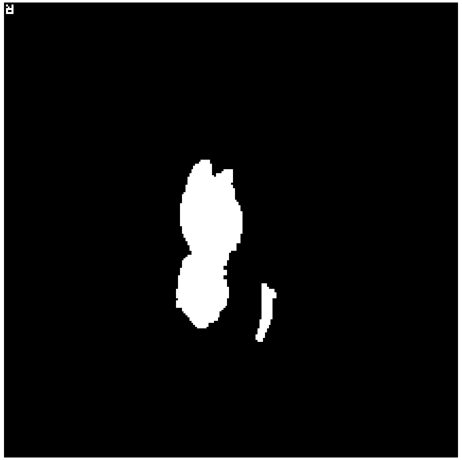
Step 3: Noise removal, semantically

Hyperintense areas + **Very intense** touching them ~= **tumour**

The obtained area is refined using **texture similarity** (non-logical)



intensity



growTum

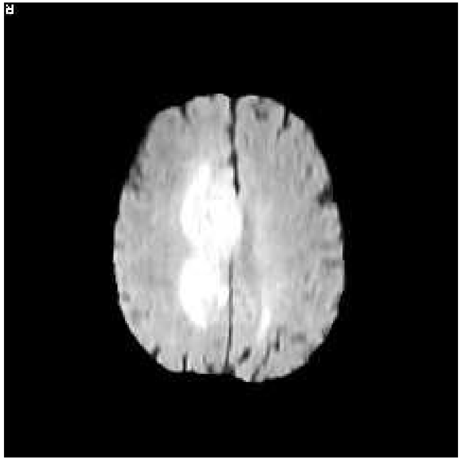
tumSim

Step 3: Noise removal, semantically

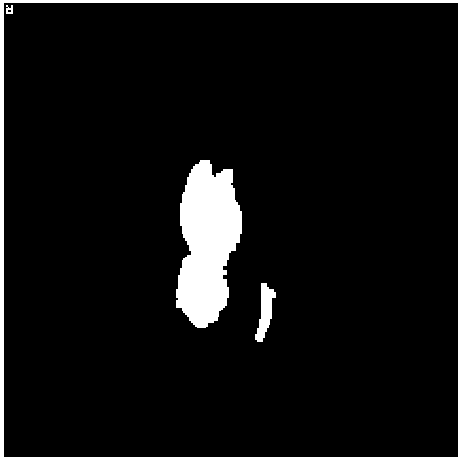
```
let growTum = grow(hyperIntense,veryIntense)

let tumSim = filter(2.0,similarTo(growTum) >. 0.6)

let gtv = grow(growTum,tumSim)
```



intensity



growTum

tumSim

Results

VoxLogicA vs manual segmentation

BRATS17 dataset

Type	Dice 193 cases	Dice 210 cases
Gross Tumour Volume	0.85(0.10)	0.81(0.18)
Clinical Target Volume	0.90(0.09)	0.87(0.15)

Higher is not better!

Human factors, decisions, uncertainty. 0.90 is actually "perfect".

$$\mathrm{Dice}(X, Y) = \frac{2 \cdot |X \cap Y|}{|X| + |Y|} = \frac{2 \cdot \mathrm{TruePos}}{2 \cdot \mathrm{TruePos} + \mathrm{FalsePos} + \mathrm{FalseNeg}}$$

18 selected techniques of BraTS17

(of 50 papers, those submitting > 100 cases)

Dice GTV avg & range: **0.88 (0.64 to 0.96)**

Part 4

Conclusions & Outlook

- GPU computation
- Geometric Model Checking
- Minimization
- Human-Computer Interaction
- Spatial logics as a query language

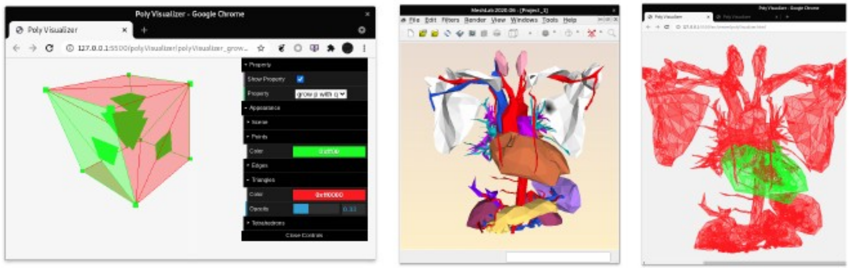
PolyLogicA

joint work with N. Bezhanisvili, D. Gabelaia, G. Grilletti, D. Latella, M. Massink

“Geometric Model Checking of Continuous Space”, https://arxiv.org/abs/2105.06194

- **Prototype**
  - still not optimized
  - custom .json format
  - simple visualizer
- Up to 400k polygons “.obj” mesh (tested so far)
- 20k polygons: 5 seconds

  400k polygons: 2 minutes



(a) Visualizer (b) Simple 3D medical image (c) Heart

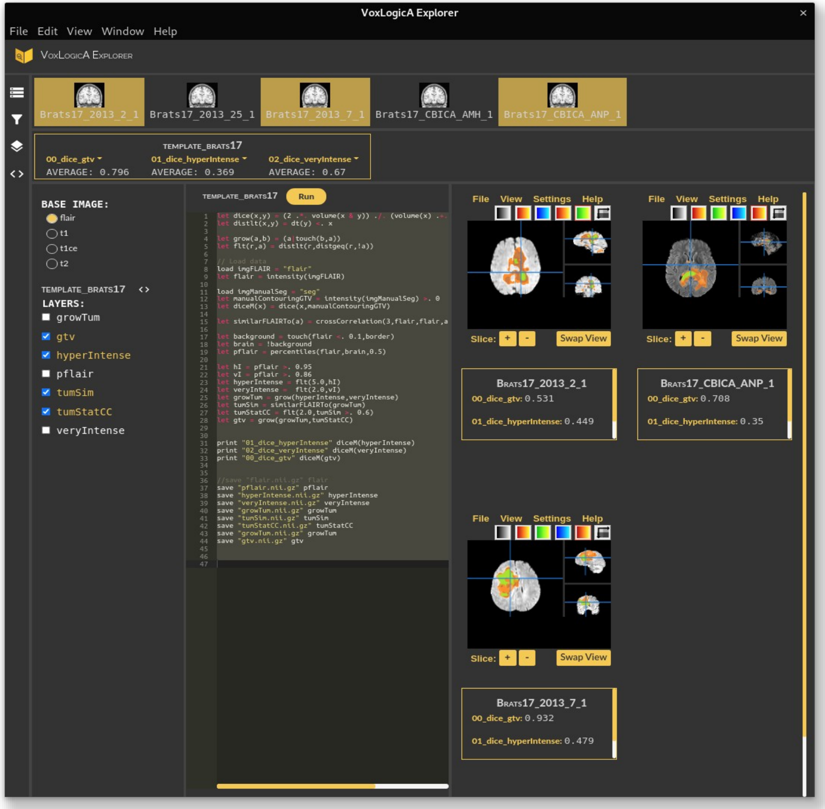
(d) Liver (e) Some veins (f) Selected vein

HCI: Dataset-oriented User interface (work in progress)

joint work (in progress) with G. Broccia, D. Latella, M. Massink

- **HCI methods**
  to improve understanding
  of logic formulas by end users.

- Reduce the **cognitive load**
  of imaging tasks!



MiniLogicA

joint work (in progress) with D. Latella, M. Massink, E. de Vink

https://arxiv.org/abs/2005.05578 - https://arxiv.org/abs/2105.06690

- A **spatial minimizer**, "strong" equivalence.

- Coming soon:

  "**Stuttering**" / "branching" bisimilarity.

  Jan Friso Groote, David N. Jansen, Jeroen J. A. Keiren, Anton Wijs: An O(mlogn) Algorithm for Computing Stuttering Equivalence and Branching Bisimulation. ACM Trans. Comput. Log (2017)

MiniLogicA

joint work (in progress) with D. Latella, M. Massink, E. de Vink

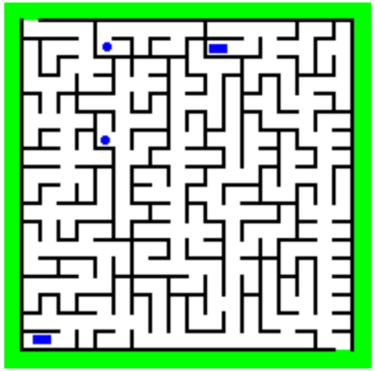
https://arxiv.org/abs/2005.05578 - https://arxiv.org/abs/2105.06690

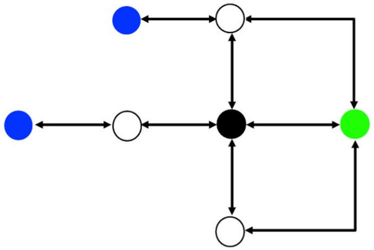
- A **spatial minimizer**, "strong" equivalence.

- Coming soon:

  "**Stuttering**" / "branching" bisimilarity.

  Jan Friso Groote, David N. Jansen, Jeroen J. A. Keiren, Anton Wijs: An O(mlogn) Algorithm for Computing Stuttering Equivalence and Branching Bisimulation. ACM Trans. Comput. Log (2017)





Some References

- **Our contribution in the SPIN proceedings:**
  “A Hands-on Introduction to Spatial Model Checking using VoxLogicA”

- **Handbook of spatial logics:**
  https://www.springer.com/la/book/9781402055867

- **Model Checking Spatial Logics for Closure Spaces:**
  https://lmcs.episciences.org/2067 (definition of SLCS, LMCS 2016)

- **VoxLogicA: a Spatial Model Checker for Declarative Image Analysis:**
  http://www.voxlogica.org Official web site.
  https://github.com/vincenzoml/VoxLogicA Free and Open Source, Apache2-Licensed.
  https://link.springer.com/chapter/10.1007/978-3-030-17462-0_16 Brain Tumour Segmentation, TACAS 2019.
  https://ieeexplore.ieee.org/document/9460947 Contouring of Nevi, j.w. with G. Broccia, FormaliSE@ICSE 2021.
  https://link.springer.com/chapter/10.1007%2F978-3-030-78089-0_12 GPU, L. Bussi, V. Ciancia, F. Gadducci, FORTE 2021.
  https://arxiv.org/abs/2105.06194 Polyhedral Models, N. Bezhanishvili, V. Ciancia, D. Gabelaia, G. Grilletti, M. Massink, D. Latella

Some Related Work

- Haghighi, I., Jones, A., Kong, Z., Bartocci, E., Grosu, R., Belta, C.: Spatel: A novel spatial-temporal logic and its applications to networked systems. HSCC, ACM 2015.
  A spatio-temporal logical language designed to be easily “learned” (in the sense of Machine Learning)
- Nenzi, L., Bortolussi, L., Ciancia, V., Loreti, M., Massink, M.: Qualitative and quantitative monitoring of spatio-temporal properties. RV, Springer 2015
  A logical language that merges the operators of the Signal Temporal Logic with SLCS
- Bartocci, E., Bortolussi, L., Loreti, M., Nenzi, L.: Monitoring mobile and spatially distributed cyber-physical systems. MEMOCODE, ACM 2017.
  Spatio-temporal Logic STREL, extending the Signal Temporal Logic with Spatial Operators inspired by SLCS
- Tsigkanos, C., Kehrer, T., Ghezzi, C.: Modeling and verification of evolving cyber-physical spaces. ESEC/FSE, ACM 2017.
  A spatio-temporal language with operators from LTL and from SLCS, devoted to the analysis of bigraphical structures
- Linker, S., Papacchini, F., Sevegnani, M.: Analysing spatial properties on neighbourhood spaces. MFCS, LIPIcs 2020.
  Bisimilairty in the SLCS setting
- Davide Castelnovo, Marino Miculan: Closure hyperdoctrines, with paths. CoRR abs/2007.04213 (2020)
  A categorical rendition of SLCS with reachability operators, generalising closure spaces to many different settings
- Loreti, M., Quadrini, M.: A spatial logic for a simplicial complex model (2021), https://arxiv.org/abs/2105.08708
  A Spatial Logical Setting interpreting SLCS on Simplicial Complexes, with applications to data analysis
- Audrito, G., Casadei, R., Damiani, F., Stolz, V., Viroli, M.: Adaptive distributed monitors of spatial properties for cyber-physical systems. Journal of Systems and Software, Elsevier 2021.
  Distributed interpretation of SLCS that provides runtime verification of formulas

Thanks* for

Listening!

*Further acknowledgements to former and current coauthors in the area of spatial logics & model checking:
F. **Banci Buonamici**, D. **Basile**, G. **Belmonte**, N. **Bezhanisvili**, G. **Broccia**, L. **Bortolussi**, L. **Bussi**, E. **de Vink**, D. **Gabelaia**, F. **Gadducci**, S. **Gilmore** (RIP), S. **Gnesi**, G.**Grilletti**, D. **Latella**, M. **Loreti**, M. **Massink**, L. **Nenzi**, R. **Paskauskas**, G. **Spagnolo**, M. **ter Beek**, A. **Vandin**.