Knots as processes

Towards a new kind of invariant

Greg Meredith[1] David Snyder[2] Matthias Radestock[3]
Francis Tang[3]

[1]Biosimilarity LLC

[2]Department of Mathematics
Texas State University San Marcos

[3]LShift Ltd

21-07-2006 / Cats Kets and Cloisters

Outline

Outline

Related work

- Goubault, Van Glabbeek, Pratt and others have extensively investigated connections between algebraic topology and process algebras
- Herlihy has investigated connections between algebraic topology and concurrent algorithms

Our contribution

- The work cited above is primarily oriented around mining the more mature body of maths (algebraic topology) for insights into the younger body (concurrency) – using space to investigate behavior
- The present work is about turning the tables – using behavior to investigate space
  - We exhibit an encoding of knots as processes in which knots are equivalent (ambient isotopic) iff their encodings as processes are equivalent (weakly bisimilar)

Invariants

- The latter half of the 20th century saw the emergence of a new class of algebraic structures.
  - The λ and π-calculi are distinguished by explicit *internal* representations of dynamics
  - C.f. structures like vector spaces where dynamics is expressed by maps *between* structures
- Can these new structures be mined for invariants?
- What sort of information might the internal representation of dynamics be sensitive to?

Proof methods

- Concommitantly, *bisimulation* has emerged as a powerful proof method
  - Intuitive
    - Entities are distinguished iff there is a distinguishing experiment
  - Adaptable
    - Find the proper notion of experiment
  - Sporting all manner of up-to techniques
- Can the scope of bisimulation be extended to more traditional areas of mathematics?

Space as behavior

- These two observations are linked – bisimulation has been an exceptionally effective notion and methodology across these algebraic structures
- Underlying this link is common world-view (very explicit in the λ and π calculis)
  - Ontology arises out of behavior
  - Things *are* because they *do*
- Realizing this program for a notion of space(time) connects with the Einsteinian program – space arises out of behavior

Outline

Trefoil as computing device
Working with projections

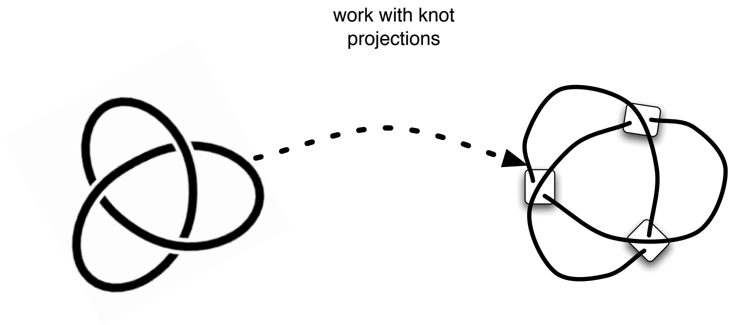


Figure: Trefoil as projection

Trefoil as computing device
Crossings as circuits

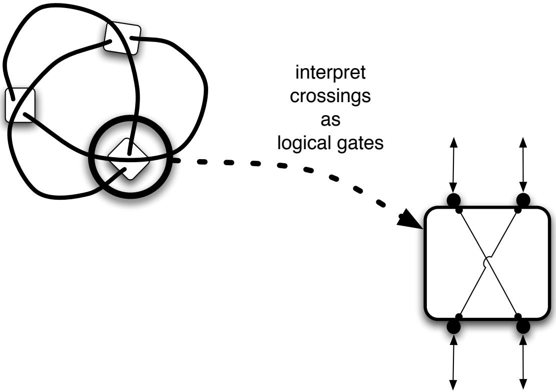


Figure: Crossings as circuits

Trefoil as computing device

Wiring it all together

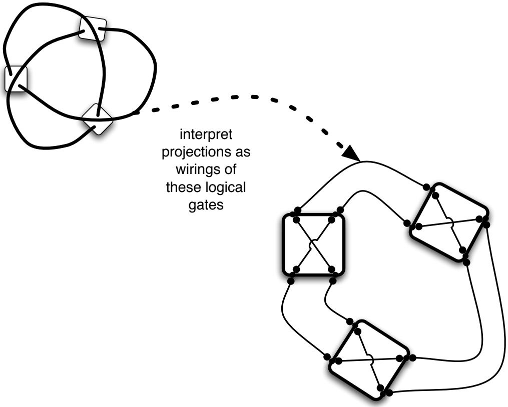
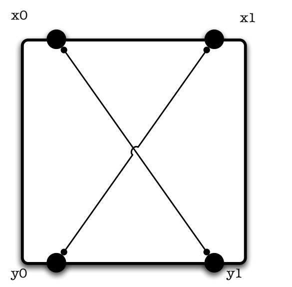


Figure: Trefoil as device

Crossing circuits



$$
\begin{aligned}
C(x_0, x_1, y_0, y_1, u) :=& \\
& x_1?(s).y_0!(s).(C(x_0, x_1, y_0, y_1, u)|u!) \\
& +y_0?(s).x_1!(s).(C(x_0, x_1, y_0, y_1, u)|u!) \\
& +x_0?(s).u?.y_1!(s).(C(x_0, x_1, y_0, y_1, u)) \\
& +y_1?(s).u?.x_0!(s).(C(x_0, x_1, y_0, y_1, u))
\end{aligned}
$$

Wires and buffers

$$W(x,y) \quad := \quad (\nu\ n\ m)(\mathit{Waiting}(x,n,m)|\mathit{Waiting}(y,m,n))$$

$$\begin{aligned}
&\mathit{Waiting}(x,c,n) := \\
&\qquad x?(v).(\nu\ m)(\mathit{Cell}(n,v,m)|\mathit{Waiting}(x,c,m)) \\
&\qquad +c?(w).c?(c).\mathit{Ready}(x,c,n,w) \\
&\mathit{Ready}(x,c,n,w) := \\
&\qquad x?(v).(\nu\ m)(\mathit{Cell}(n,v,m)|\mathit{Ready}(x,c,m,w)) \\
&\qquad +x!(w).\mathit{Waiting}(x,c,n)
\end{aligned}$$

$$\mathit{Cell}(c,v,n) \quad := \quad c!(v).c!(n).0$$

Main theorem

$$\text{Main theorem: } K_1 \sim K_2 \iff [\![K_1]\!] \simeq [\![K_2]\!]$$

Need to unpack

- $[\![-]\!] : \mathit{Knots} \rightarrow \pi$
  - specify target language, π-calculus
  - specify a source language, *Knots*
- notion of equivalence, $\sim$, in *Knots*
- notion of equivalence, $\simeq$, in π-calculus

Outline

Target language: *pi* in 5

Syntax

$$
\begin{array}{llcl}
\text{summation} & N & ::= & \Sigma_{i \in I} x_i . A_i \\
\text{agent} & A & ::= & F \mid C \mid (\nu\, \vec{x}) A \\
\text{abstraction} & F & ::= & (\vec{x}) P \mid (\nu\, \vec{x}) F \\
\text{concretion} & C & ::= & [\vec{x}] P \mid (\nu\, \vec{x}) C \\
\text{process} & P, Q & ::= & N \mid P|Q \mid (\nu\, \vec{x}) P \\
 & & \mid & X \langle \vec{y} \rangle \mid (\mathrm{rec}\, X(\vec{x}).P) \langle \vec{y} \rangle
\end{array}
$$

$$
\begin{array}{rcl}
x?(\vec{y}).P & \triangleq & x.(\vec{y})P \\
x!(\vec{y}).P & \triangleq & x.[\vec{y}]P
\end{array}
$$

Target language: *pi* in 5

Structural equivalence

The *structural congruence*, $\equiv$, between processes is the least congruence closed with respect to alpha-renaming, satisfying AC for $|$ and $+$, 0 following axioms:

1. the scope laws:

$$
\begin{aligned}
(\nu\, x)0 &\equiv 0, \\
(\nu\, x)(\nu\, x)P &\equiv (\nu\, x)P, \\
(\nu\, x)(\nu\, y)P &\equiv (\nu\, y)(\nu\, x)P, \\
P|(\nu\, x)Q &\equiv (\nu\, x)(P|Q), \ \textit{if } x \notin \mathcal{FN}(P)
\end{aligned}
$$

2. the recursion law:

$$
(\mathrm{rec}\, X(\vec{x}).P)\langle\vec{y}\rangle \equiv P\{\vec{y}/\vec{x}\}\{(\mathrm{rec}\, X(\vec{x}).P)/X\}
$$

Target language: *pi* in 5

Operational semantics

$$\frac{|F| = |C|}{x.F \mid x.C \rightarrow F \circ C}\ \textsc{Comm}$$

$$\textsc{Par}\ \frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \qquad \frac{P \rightarrow P'}{(\nu\, x)P \rightarrow (\nu\, x)P'}\ \textsc{New}$$

$$\frac{P \equiv P' \qquad P' \rightarrow Q' \qquad Q' \equiv Q}{P \rightarrow Q}\ \textsc{Equiv}$$

$$(\vec{y})P \circ (\nu\vec{v})[\vec{z}]Q \triangleq (\nu\vec{v})(P\{\vec{z}/\vec{y}\}|Q)$$

As usual, write $\Rightarrow$ for $\rightarrow^*$.

Target language: *pi* in 5

Bisimulation

Definition

An agent, B, occurs *unguarded* in A if it has an occurence in A not guarded by a prefix x. A process P is observable at x, written here $P \downarrow x$, if some agent $x.A$ occurs unguarded in P. We write $P \Downarrow x$ if there is Q such that $P \Rightarrow Q$ and $Q \downarrow x$.

Target language: *pi* in 5

Bisimulation

Definition

A *barbed bisimulation* is a symmetric binary relation $\mathcal{S}$ between agents such that $P \mathrel{\mathcal{S}} Q$ implies:

1. If $P \rightarrow P'$ then $Q \Rightarrow Q'$ and $P' \mathrel{\mathcal{S}} Q'$.
2. If $P \downarrow x$, then $Q \Downarrow x$.

P is barbed bisimilar to Q, written $P \simeq Q$, if $P \mathrel{\mathcal{S}} Q$ for some barbed bisimulation $\mathcal{S}$.

Shape of the encoding

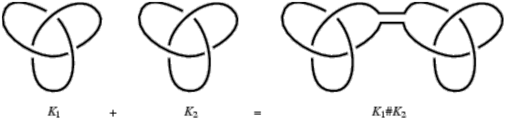
Now we are in a position to unpack the general shape of the encoding. It's just a parallel composition of crossings and wires wired up to respect the graph underlying the knot projection

$$
\begin{aligned}
&[\![K]\!] = \\
&\quad (v_0 \ldots v_{4n-1})(\Pi_{i=0}^{n-1}(\nu\ u)[\![C(i)]\!](v_{4i}, \ldots, v_{4i+3}, u) \\
&\qquad\qquad |\Pi_{i=0}^{n-1} W(v_{\omega(i,0)}, v_{\omega(i,1)})|W(v_{\omega(i,2)}, v_{\omega(i,3)}))
\end{aligned}
$$

Outline

A (very) little knot theory

- A knot is an embedding of the circle into $\mathbb{R}^3$
- Two knots, K_1 and K_2 can be composed, $K_1 \# K_2$ by cutting each and fusing the respective ends together
- A prime knot cannot be represented as the composition of knots
- We can work with knot projections because of a well-known theorem stating that knots are ambient isotopic iff you can convert the projection of one into the projection of the other via a sequence of the Reidemeister moves.



Reidemeister moves

In ‘digitizing’ knots by working with their projections we obtained another notion of equivalence: the Reidemeister moves *operationalize* ambient isotopy.

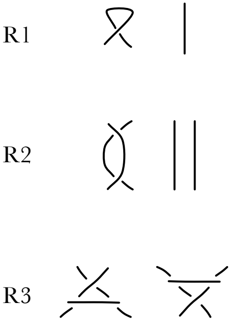


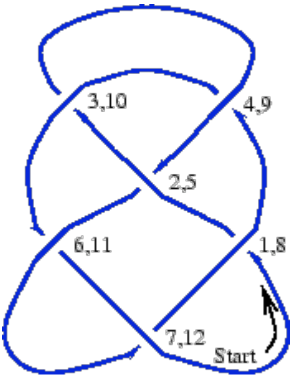
Figure: Reidemeister moves

Source language

Candidates for a language for representing knots as input to the encoding

- Dowker-Thistlethwaite codes
  - *unique* for prime knots
- John Horton Conway's Tangle Calculus a.k.a. Knotation
  - Representation theorem for rational tangles
- Signed planar graphs

DT-codes by example



DT Code: 8,10,2,12,4,6

Figure: DT-code example

DT-codes

Just the facts

- Provides a bijective map, DT, between
  - $\{i : odd(i), 1 \leq i \leq 2n\}$
  - $\{i : even(i), 2 \leq i \leq 2n\}$
- Connects $C(i)$ to
  - $C(i-1)$
  - $C(i+1)$
  - $C(DT^{-1}(DT(i)-1))$
  - $C(DT^{-1}(DT(i)+1))$
- Provides enough information to say whether i-path or $DT(i)$-path is the over-crossing

DT-codes

Wiring algorithm

```
let DTWiring i dt dti knot acc = if (i <= (numCrossings knot))
then let ic = (2*i - 1) in (DTWiring (i+1) dt knot (union acc [
W(x1(C(knot,ic)), (if (over dt ic-1) then y0 else y1));
W(y0(C(knot,ic)), (if (over dt ic+1) then x1 else x0));
W(x0(C(knot,ic)), (if (over dt (dti ((dt i)-1))) then y0 else y1));
W(y1(C(knot,ic)), (if (over dt (dti ((dt i)+1))) then x1 else x0)) ]))
else acc
```

Supporting definitions

Definition

We will say that the encoding of a knot is *alive* as long as it is firing. If it ever ceases to push signal through, then it is *dead*. We can ascertain an upperbound on initial signal that guarantees liveness of the encoding. Surely, $2\#(K)$ will guarantee the liveness of the encoding. More declaratively, we simply demand that $[\![K]\!] | \mathit{initialSignal}$ be live before we are willing to admit it as a representation of the knot.

Outline

Ambient isotopic knots have bisimilar encodings

- Since $K_1 \sim K_2$ we know there is a sequence of Reidemeister moves converting K_1 to K_2
- Each move corresponds to a bisimilarity preserving transformation on the process encoding

R-move interfaces

- For the following two lemmas we have to keep the *interface*, i.e. splice points, of the left and right hand sides of the R-move the same. So, for R1L and R2L we must restrict the ports that are not the splice points.
- One way to address this is to embed the restrictions into the encodings of R1L and R2L. Algebraically,

$$
\begin{aligned}
&[\![R1L]\!](y0, y1) = \\
&\quad (\nu\ x_0\ x_1)((\nu\ u)C(x_0, x_1, y_0, y_1, u)|W(x_0, x_1)) \\
&[\![R2L]\!](x_{00}, x_{01}, x_{10}, x_{11}) = \\
&\quad (\nu\ y_{00}, y_{01}, y_{10}, y_{11},)((\nu\ u_0)C(x_{00}, x_{01}, y_{00}, y_{01}, u0) \\
&\quad |W(y_{00}, y_{11})|W(y_{01}, y_{10})|(\nu\ u_1)C(x_{10}, x_{11}, y_{10}, y_{11}, u_1))
\end{aligned}
$$

- Technically it will be convenient to break out the restrictions

R-move interfaces
Example

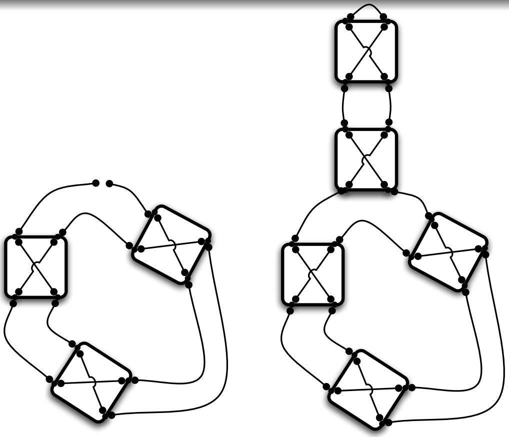


Figure: Reidemeister move and context

R-moves: context lemma

$\forall i \in \{1, 2, 3\}$ if $K_1 \xrightarrow{Ri} K_2$ then there exists a context M and (possibly empty) vector of distinct names, $\vec{w}$ s.t.

$$
\begin{aligned}
(\nu\, \vec{w})[\![K_1]\!]\langle v : w \rangle &= (\nu\, \vec{w}) M[[\![RiL]\!]] \\
[\![K_2]\!] &= M[[\![RiR]\!]]
\end{aligned}
$$

Pf: This follows directly from the definition of the encoding.

R-moves: substitution lemma

We argue that *RiL* is bisimilar to *RiR* in the context of a live encoding.That is if

- $[\![K]\!]|\mathit{initialSignal}$ is alive, and
- $[\![K]\!]|\mathit{initialSignal} = C[[\![RiL]\!]]$

then we can substitute $[\![RiR]\!]$ in its place without change of behavior, i.e.

$$\forall i \in \{1,2,3\}\ (\nu\ \vec{w})C[[\![RiL]\!]] \simeq C[[\![RiR]\!]]$$

R-moves as bisimilarity preserving xforms

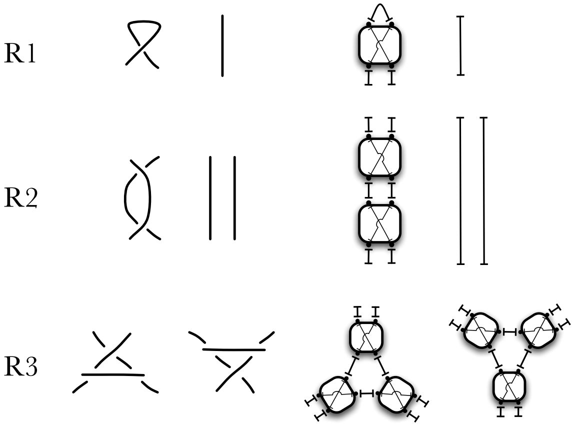


Figure: Reidemeister moves as bisimilar processes

R-moves: technical meaning of forward direction

- $[\![K_1]\!]$ is an abstraction in $4\#(K_1)$
- $[\![K_2]\!]$ is an abstraction in $4\#(K_2)$
- let $\#_{Min}(K) := min\{\#(K') : K' \sim K\}$

We assert that there is an $4\#_{Min}(K_1) \leq n \leq 4 * max\{\#(K_1), \#(K_2)\}$ for any vector of names, $\vec{v}$, s.t.

- $|\vec{v}| = n$,
- $v[i] \neq v[j] \iff i \neq j$
- there exists two vectors of names, $\vec{w_1}, \vec{w_2}$, also all distinct, s.t.

$$(\nu\ \vec{w_1})[\![K_1]\!]\langle\vec{v} : \vec{w_1}\rangle \quad \simeq \quad (\nu\ \vec{w_2})[\![K_2]\!]\langle\vec{v} : \vec{w_2}\rangle$$

with $|\vec{w_i}| = 4\#(K_i) - n$.

R-moves: technical meaning of forward direction
Strengthening

Let

$$L_C([\![K]\!]\langle\vec{u}\rangle,\vec{v}) :=$$
$$\{(\nu\, u)C(\vec{z}) : \exists P\ [\![K]\!]\langle\vec{u}\rangle = (\nu\, u)C(\vec{z})|P\}$$
$$L_W([\![K]\!]\langle\vec{u}\rangle,\vec{v},\vec{w}) :=$$
$$\{W(a,b) : \exists P\ [\![K]\!]\langle\vec{u}\rangle = W(a,b)|P, a,b \in \vec{v}, a,b \notin \vec{w}\}$$
$$L([\![K]\!]\langle\vec{u}\rangle,\vec{v},\vec{w}) :=$$
$$\Pi_{C\in L_c([\![K]\!]\langle\vec{u}\rangle,\vec{v})} C | \Pi_{W\in L_W([\![K]\!]\langle\vec{u}\rangle,\vec{v},\vec{w})} W$$

we also have

$$L([\![K_1]\!]\langle\vec{v} : \vec{w_1}\rangle,\vec{v},\vec{w_1} : \vec{w_2}) = L([\![K_2]\!]\langle\vec{v} : \vec{w_2}\rangle,\vec{v},\vec{w_1} : \vec{w_2})$$

Forward direction: moral content

- When the knots are ambient isotopic the encodings *share* a set of crossings and wires at least as big as a minimal crossing representative of the isotopy class.
- And the other parts are R-move complications of wires that would complete the knot from shared core – hidden under restriction.

R-moves: one step lemma

If K_1 is one R-move away from K_2 then

$$[\![K_1]\!]\langle v\rangle \quad \simeq \quad (\nu\, w)[\![K_2]\!]\langle v : w\rangle$$

This follows directly from the context and substitution lemmas.

R-moves: iteration of one-step lemma

- Even if you have a simplifying step followed by a complicating step, you can iterate the one-step lemma, mimicking the Reidemeister theorem.
- The reason is that crossings in a complicating step can never be involved in any other part of the context. They are effectively hidden behind the interface defined by the simplified side of the R-move.

R-moves: R1L -> R1R; R2R -> R2L

R1L -> R1R

- The R1L $\rightarrow$ R1R step means we have a context M such that

$$\begin{aligned}(\nu\, x_0\, x_1)[\![K_1]\!]\langle \vec{v_0} : x_0 : x_1 \rangle &= (\nu\, x_0\, x_1) M[[\![R1L]\!]] \\ &\simeq M[[\![R1R]\!]] \\ &= [\![K_2]\!]\langle \vec{v_0} \rangle\end{aligned}$$

R-moves: R1L -> R1R; R2R -> R2L, cont.

R2R -> R2L

- The R2R $\rightarrow$ R2L step means we have a context M' such that

$$\begin{aligned}(\nu\, y_{00}\, y_{01}\, y_{10}\, y_{11})[\![K_3]\!]\langle \vec{v_1} : y_{00} : y_{01} : y_{10} : y_{11}\rangle \\ = (\nu\, y_{00}\, y_{01}\, y_{10}\, y_{11})M'[[\![R2L]\!]] \\ \simeq M'[[\![R1R]\!]] \\ = [\![K_2]\!]\langle \vec{v_1}\rangle\end{aligned}$$

- We emphasize $\vec{v_0}, \vec{v_1}$ are just lists of distinct names with
  - $|\vec{v_0}| = |[\![K_2]\!]| = |\vec{v_1}|$
- so, pick $\vec{v_0} = \vec{v_1}$, dropping subscript, to conclude

$$
\begin{aligned}
&(\nu\, x_0\, x_1)[\![K_1]\!]\langle \vec{v} : x_0 : x_1 \rangle \simeq \\
&\qquad (\nu\, y_{00}\, y_{01}\, y_{10}\, y_{11})[\![K_3]\!]\langle \vec{v} : y_{00} : y_{01} : y_{10} : y_{11} \rangle
\end{aligned}
$$

- with $[\![K_2]\!]\langle \vec{v} \rangle$ forming the shared core

Outline

Bisimilar encodings come from isotopic knots

Strategy: assume encodings are bisimilar but knots not ambient isotopic and derive contradiction.

- W.l.o.g. demand knots be given in minimal crossing projections
- If crossing numbers are different then free names differ – contradicting bisimilarity
- Therefore crossing numbers must be the same
  - $\Pi_{i=0}^{n-1}[\![C(i)]\!](\ldots)|\Pi_{i=0}^{n-1}W(\ldots)|W(\ldots) \simeq \Pi_{j=0}^{n-1}[\![C(j)]\!](\ldots)|\Pi_{j=0}^{n-1}W'(\ldots)|W'(\ldots)$
  - $\Rightarrow \Pi_{i=0}^{n-1}W(\ldots)|W(\ldots) \simeq \Pi_{j=0}^{n-1}W'(\ldots)|W'(\ldots)$
  - If any of these wires differ, then there is a distinguishing barb
  - But, if none of them differ the knots must be ambient isotopic because their respective sets of crossings are wired identically – contradiction

Conclusions

- Computational calculi constitute a reasonable new source of invariants.
- Bisimulation is a proof method ready for wider exploitation.
- A first step towards another way to think about space as behavior.

Future work

- Some things we haven't said
  - Knot sum has a direct representation in this encoding
  - Kauffman bracket has a direct representation in this encoding
- Applications and future developments
  - Structure of knots now susceptible to inspection via Hennessy-Milner logics
  - Applications to biology – protein folding
  - Approach generalizes to give a direct representation of *spin networks*