Temperley-Lieb Algebras

as two-way automata

Peter Hines – York University

EPSRC
Engineering and Physical Sciences Research Council

QIRC

Project GR/S82176/01

Slides available at : *http://www-users.cs.york.ac.uk/∼phines/QNET06.pdf*

Background

This talk is about various related topics :

The **Temperley-Lieb algebra** :

- A von Neumann algebra, developed for statistical mechanics, very important in knot theory.

Polynomial knot invariants :

- Polynomials derived from knot presentations, invariant of the way the knot is drawn.

The **Geometry of Interaction** construction :

- A categorical construction that gives *compact closed* categories from *traced monoidal* categories.
- Originating from Girard's Linear Logic, and 'Geometry of Interaction' program.

Two-way automata :

- Simple finite state machines, also known as 'read-only Turing machines'.

What is already known ?

- **V. Jones** The T-L algebra plays a key role in knot invariants.

- **L. Kauffman** It also has a presentation as ‘planar diagrams’.

- **PMH** Models of 2-way automata are examples of the GoI construction.

- **S. Abramsky** The T-L algebra has a *fully abstract* presentation, as planar diagrams within a GoI category.

For experts : There are 2 very different flavours of GoI, ‘particle-style’ and ‘wave-style’. This talk is all about particle-style GoI.

Why the interest ?

- (2006) **Aharanov, Jones, Landau** The 'Quantum Jones Polynomial' algorithm.
  - This gives an exponential speedup in computing the Jones knot polynomial,
  - but only at certain distinguished values ...

- This algorithm relies on :

1. The Hadamard test — a standard bit of QM algorithm toolkit.
2. *Unitary representations of the T-L algebra.*
3. A clever result on the uniqueness of traces, in various settings.

Some questions ...

1. There is an implicit connection between the T.-L. algebra and two-way automata :

   – can this be made explicit ?

2. (This requires :) what does planarity mean for two-way automata ?

3. What is the complexity class of the resulting machines ?

4. Is there a connection with :
   (a) quantum two-way automata ?
   (b) knot theory ?
   (c) The Jones polynomial algorithm ?

The Temperley-Lieb algebra

A purely algebraic definition :

The **Temperley-Lieb monoid** M_n :

This has generators $\delta, U_1, U_2, \ldots U_n$, and relations

$$U_i U_j U_i = U_i \quad \text{for all} \quad |i-j| = 1$$

$$U_i^2 = \delta U_i = U_i \delta$$

$$U_i U_j = U_j U_i \quad \text{for all} \quad |i-j| > 1$$

The **Temperley-Lieb algebra** TL_n :

- Consider the ring L_X of all 1-variable Laurent polynomials over X ...

- The T.-L. algebra is the monoid algebra of formal linear combinations

$$\sum_i l_i m_i \qquad \text{where } l_i \in L_X \quad \text{and} \quad m_i \in M_n$$

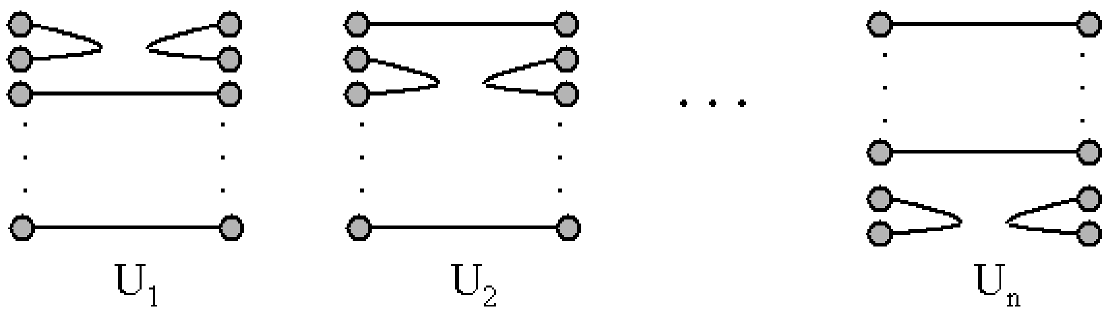
up to some quotient $\delta = \tau.1$, where $\tau \in L_X$..

The Temperley-Lieb monoid as planar diagrams

The T.-L. algebra :

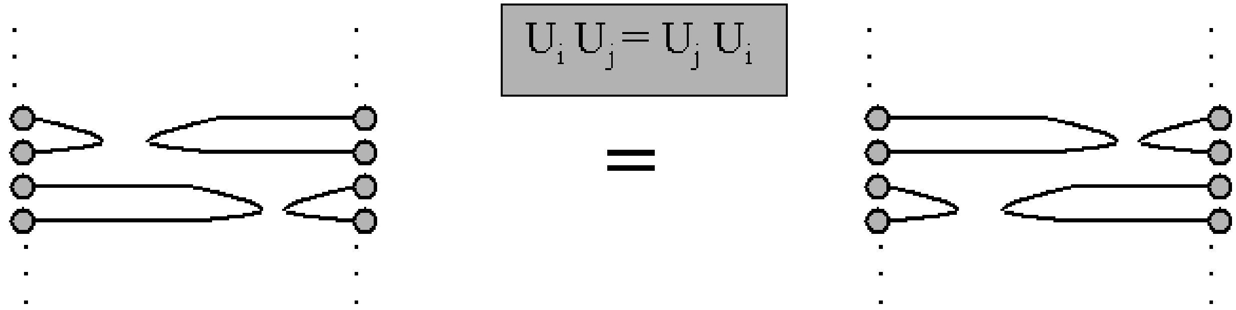
- Independently rediscovered by V. Jones, (in 1985), it plays a starring rôle in his knot polynomial.
- L. Kauffman gave an interpretation (in 1990) as *planar diagrams*.

THE GENERATORS OF THE TEMPERLEY-LIEB MONOID

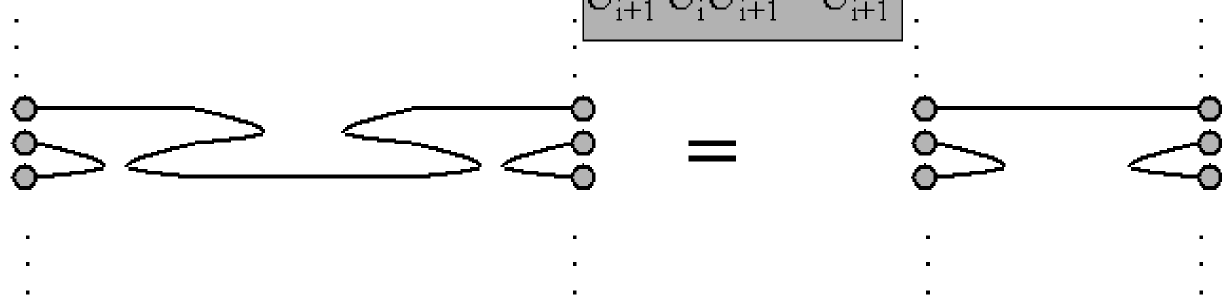


These are considered up to *planar isotopy* — and this provides the relations between generators.

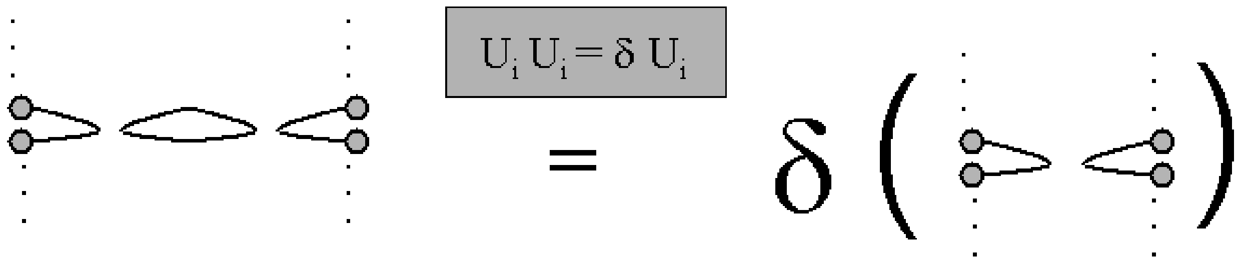
$$U_i U_j = U_j U_i$$



$$U_{i+1} U_i U_{i+1} = U_{i+1}$$



The only ‘non-obvious’ relation is that closed loops become global scaling factors :



Special cases :

- $\delta = 0$.
  - Everything is trivial.

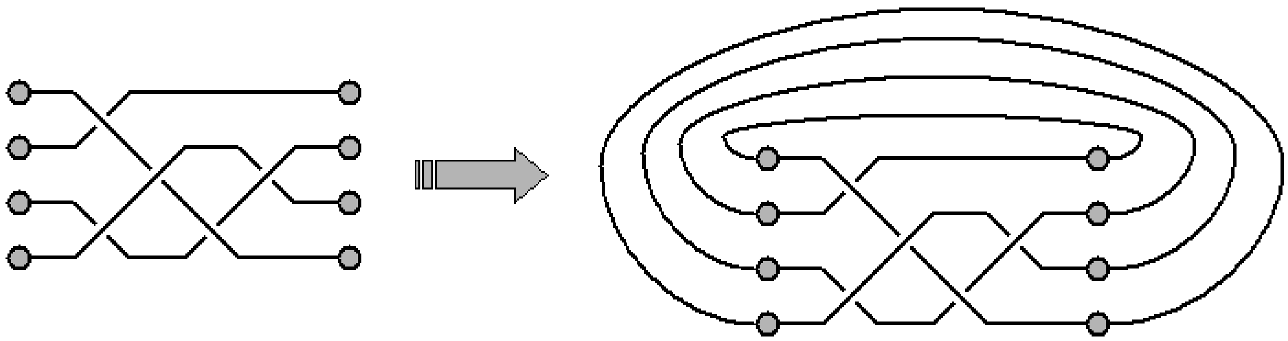
- $\delta = 1$.
  - Loops are ignored completely.

- $\delta = \omega$, an p-th root of the identity, so $\omega^p = 1$.
  - This is the case covered by the quantum algorithm (when p is *prime*).

Knot invariants from the Temperley-Lieb algebra

The key steps are :

Braid closure A braid diagram may be closed by adding in feedback loops.



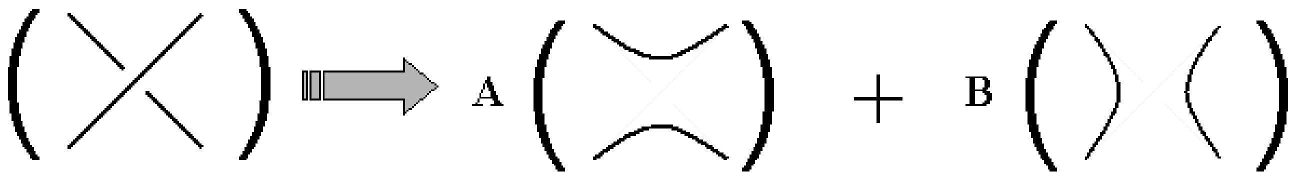
Traditional knot theory – every knot or link is the closure of a braid diagram.

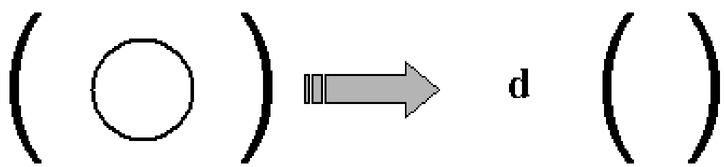
Kauffman computed the **Jones polynomial** using a recursive algorithm to ‘eliminating crossings in a diagram’.

A *diagram with crossings* is mapped to the formal sum of *diagrams without crossings* — in an exponential number of steps.

The **Jones polynomial**, as described by **Kauffman**, is computed by

1. Replacing crossings with weighted formal sums of link diagrams.
2. Replacing unknotted loops with values.





The weights are *Laurent polynomials* over 1 variable, X, and taking

- $A = X$
- $B = X^{-1}$
- $d = -X^2 + X^{-2}$

maps equivalent knot diagrams to the same polynomial.

2-way automata — a complete change of subject

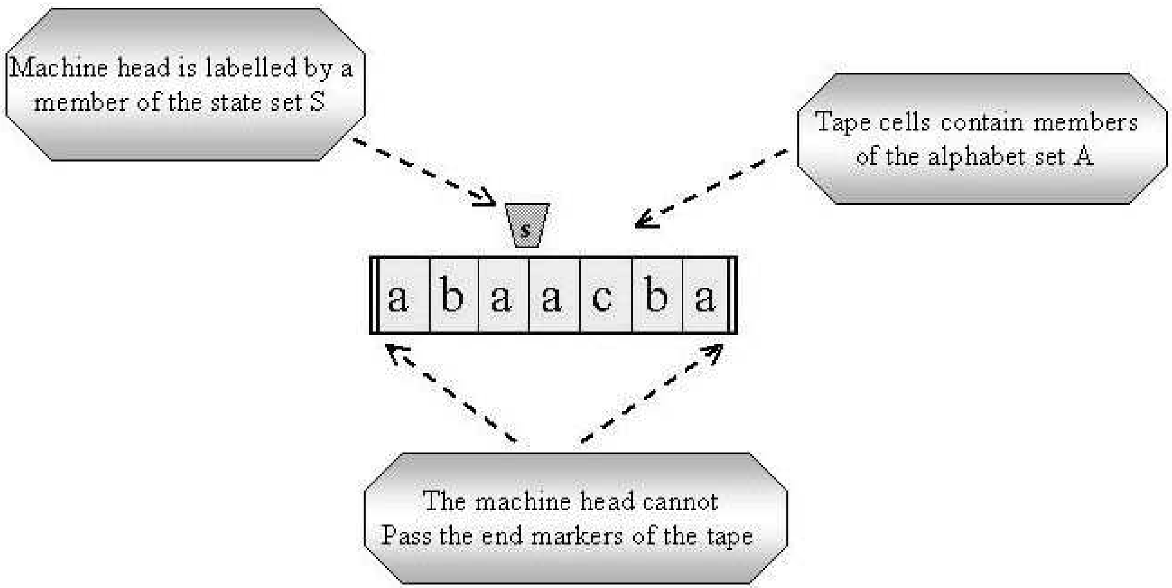
A **two-way automaton** is specified by :

- A set A of *Alphabet Symbols*
- A set S of *States*
  - S is divided into *left-moving states* L, and *right-moving states* R, so $S = L \uplus R$.
- For each $a \in A$, a *next-state relation* $[a] \subseteq S \times S$.

As a state machine, there is :

- A **finite tape**, with alphabet symbols written on it.
- A single **machine head**, labelled by a state.
- **End markers** for the tape.

The anatomy of a 2-way automaton



This is one definition. Others are similar, and provably equivalent.

The dynamics of a 2-way automaton

At each *primitive step* :

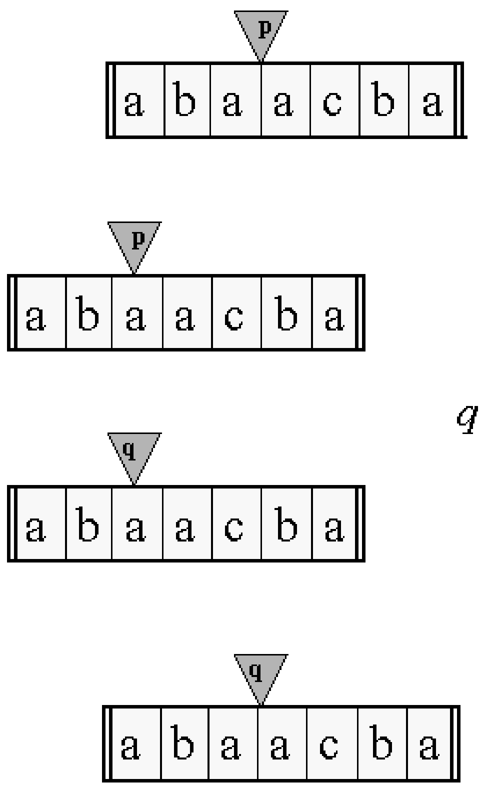
1. If the machine head has a *left-moving* label, it moves onto the cell to the *left*.
   – alternatively, it moves onto the cell to the *right*.

2. The cell contents determine a new label for the machine head.

3. If the new label is *left-moving*, the machine head moves to the *left* of the cell.
   – alternatively, it moves to the *right* of the cell.

(This description is due to PMH. It is simpler than, but equivalent to, Birget's definition).

An example 2-way automaton computation:



p is left-moving

$q\ [a]\ p$

q is right-moving

Boundary configurations

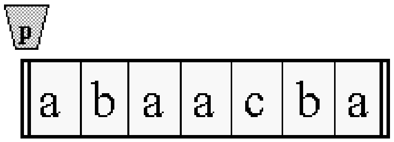
A **configuration** is simply an instantaneous description of a 2-way automaton.

From the definition — a configuration with the machine head over an end-marker has either

1. no *‘previous configuration’* under the machine evolution.

2. no *‘next configuration’* under the machine evolution.

Call such configurations the **boundary configurations**.



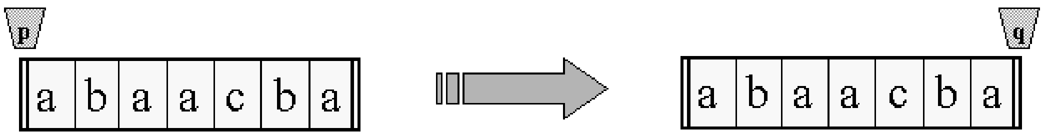
Birget's Relations

Each word $w \in A^*$ determines a relation $[w]$ on the state set.

q is related to p by $[w]$, written $q[w]p$ exactly when :

There exists a boundary-to-boundary computation that

1. Starts with p labelling the machine head.
2. Finishes with q labelling the machine head.



This is called the **global transition relation** of w.

Some basic results :

- The **transition relation** for a singleton $a \in A$ is exactly the **next-state relation** from the definition.
- If a two-way automaton is **deterministic**, every transition relation is a **partial function**.
- If a 2-way automaton is **reversible**, every transition relation is a **partial bijection**.

A not-so-basic result :

- The relation $[uv]$ can be derived from $[u]$ and $[v]$ separately.
  - Formulæ to do this were given by J.-C. Birget.
  - These are just the composition given by the $\mathbf{GoI}$ construction.

From two-way automata to planar diagrams

A two-way automaton has :

- A set A of **Alphabet Symbols**
- Sets L and R of **left-moving** and **right-moving** States.
- For each $a \in A$, a **transition relation** $[a] \subseteq S \times S$.

We also require :

1. A **partial order** $\leq$ on the state set S, satisfying:
   - The subsets L and R are *chains* – i.e. totally ordered subsets.
   - Left-moving and right-moving states are *incomparable*, so $l \# r$ for all $l \in L$, $r \in R$.
2. An **bijection** $\sigma : S \to S$, satisfying
   - σ is an *involution*, so $\sigma^2 = 1_S$.
   - σ is *anti-monotonic*, so $p \leq q \;\; \Rightarrow \;\; \sigma(q) \leq \sigma(p)$.

Conventions :

For 2n states, write the left-moving states as

$$\overleftarrow{1} \leq \overleftarrow{2} \leq \ldots \leq \overleftarrow{n}$$

and the right-moving states as

$$\overrightarrow{1} \geq \overrightarrow{2} \geq \ldots \geq \overrightarrow{n}$$

The axioms state that :

$$\sigma(\overleftarrow{a}) = \overrightarrow{a} \text{ and } \sigma(\overrightarrow{a}) = \overleftarrow{a}$$

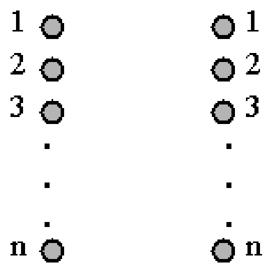
$$\overleftarrow{p} \# \overrightarrow{q} \quad \text{for all } 1 \leq p, q \leq n$$

Transition diagrams

We can now give a diagrammatic presentation of transition relations.

Let $w \in A$ be a word over the input alphabet.

Start with 2 columns of nodes, labelled $1 ... n$

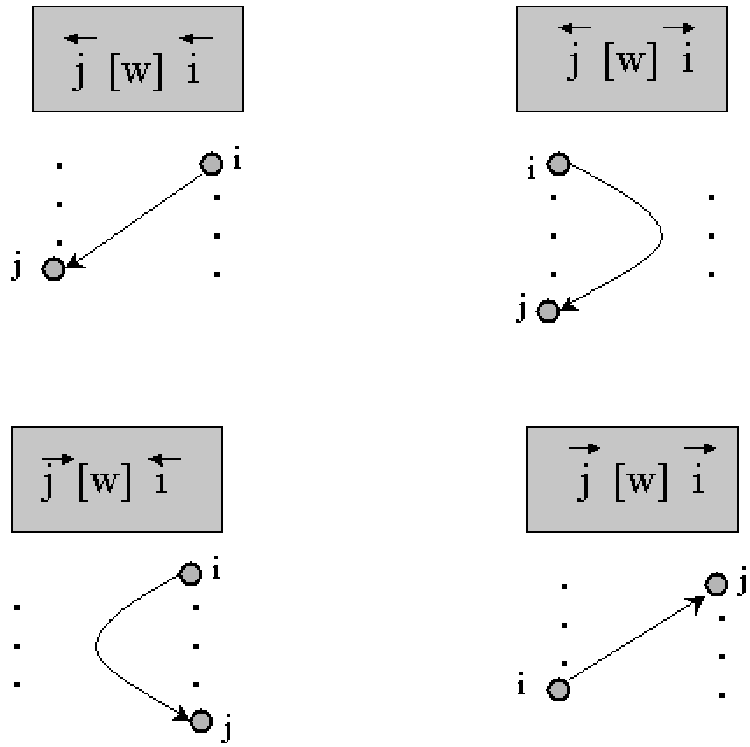


For each pair of states q, p related by the transition relation $[w]$,

$$q[w]p$$

Draw a directed line on this diagram :

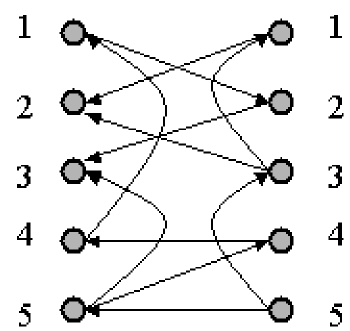
From relations to diagrams :



Each transition relation determines, and is determined by a **transition diagram**.

the T.-L. monoid, and transition diagrams ?

Every transition relation $[w]$ determines, and is determined by, a diagram such as



Questions : When are these diagrams

1. **undirected ?**

   i.e. The direction on the arrows does not matter.

2. **planar ?**

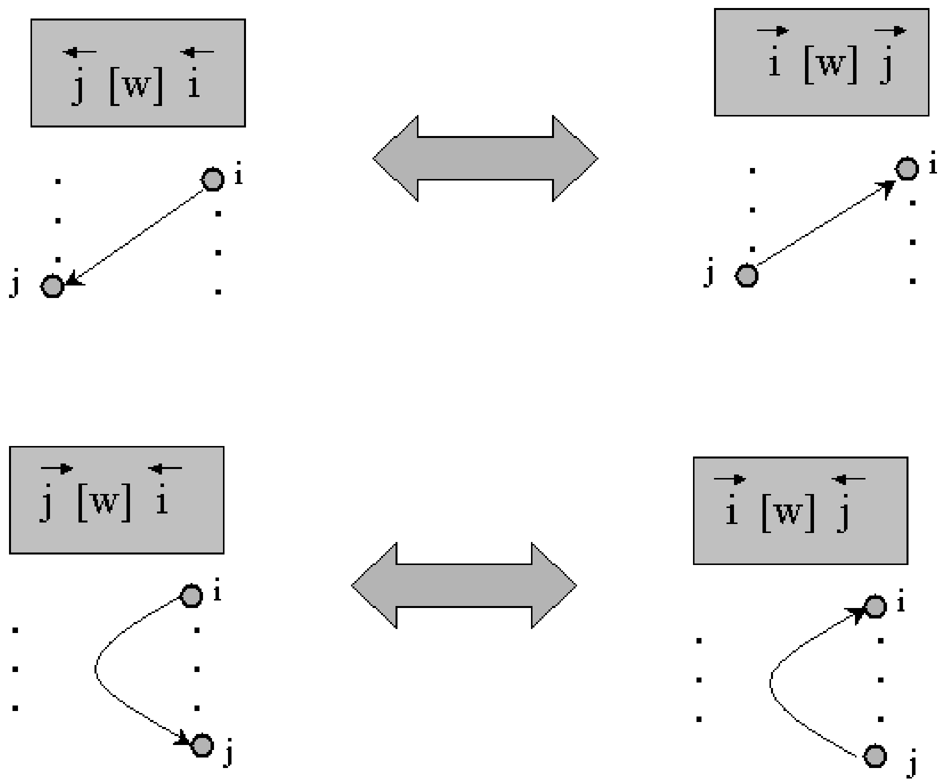
   i.e. Lines in the diagram do not cross.

– the intention is to reproduce Kauffman's diagrammatic presentation of the Temperley-Lieb monoid.

Undirected transition diagrams, graphically

A diagram is <u>undirected</u> when :

- whenever there is a line from node x to node y, there is also a line from node y to node x.



Undirected transition diagrams, algebraically

"whenever there is a line from node x to node y, there is also a line from node y to node x" states that :

$$y[w]x \;\Leftrightarrow\; \sigma(x)[w]\sigma(y)$$

or, using the relational converse,

$$y[w]x \;\Leftrightarrow\; \sigma(y)[w]^c\sigma(x)$$

writing σ in relational form, and noting that this is quantified over x, y :

$$[w] \;=\; \sigma[w]^c\sigma \;=\; \sigma^{-1}[w]^c\sigma$$

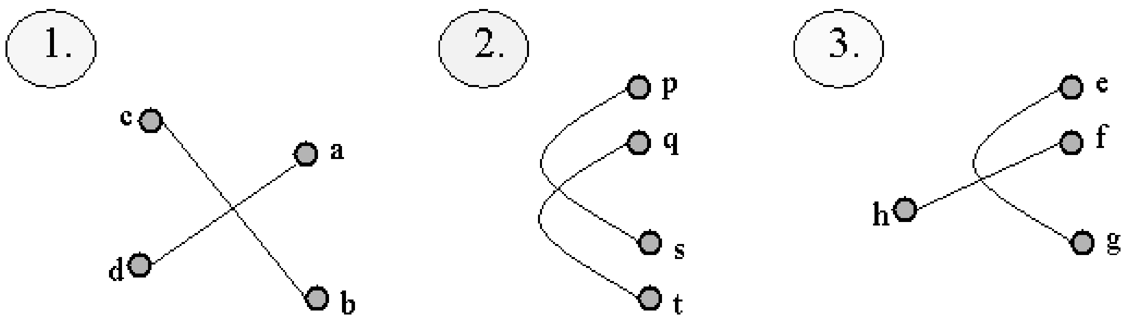
– giving a characterisation of undirected transition diagrams.

Enforcing Planarity – diagramatically

Let $w \in A^*$ be an input word, with an undirected transition diagram.

Question when is this planar ??

To enforce planarity we need to rule out 3[a] possibilities :



Each undirected diagram corresponds to 4 statements a transition relation $[w]$

— planarity for directed diagrams requires 4 times as many axioms !

[a] (Up to left-right symmetry ...)

Enforcing Planarity – algebraically

Claim : 2 conditions force a transition diagram for w to be planar.

- **Weak monotonicity :**

  Given

$$q[w]p \quad \text{and} \quad q'[w]p'$$

  then

$$p \leq p' \;\Rightarrow\; q \leq q' \;\text{ or }\; q \# q'$$

- **The interval condition :**

  Given

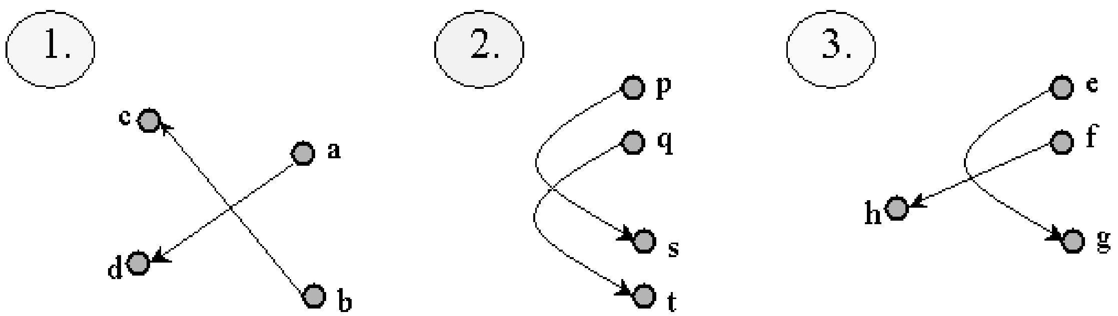
$$y[w]x \quad \text{and} \quad b[w]a$$

  then

$$a \leq x \leq \sigma(b) \;\Rightarrow\; \sigma(a) \leq y \leq b$$

How do these conditions work ?

Consider 3 distinct crossing types, drawn with an orientation :



From **1.** : $\overleftarrow{d}\,[w]\,\overleftarrow{a}$ and $\overleftarrow{c}\,[w]\,\overleftarrow{b}$. However, $\overleftarrow{a} \leq \overleftarrow{b}$ but $\overleftarrow{d} \geq \overleftarrow{c}$, contradicting **weak montonicity**.

From **2.** : $\overrightarrow{s}\,[w]\,\overleftarrow{p}$ and $\overrightarrow{t}\,[w]\,\overleftarrow{q}$. However, $\overleftarrow{p} \leq \overleftarrow{q}$ but $\overrightarrow{s} \geq \overrightarrow{t}$, contradicting **weak montonicity**.

From **3.** : $\overleftarrow{e} \leq \overleftarrow{f} \leq \sigma(\overrightarrow{g})$. However, $\sigma(\overleftarrow{e}) = \overrightarrow{e} \leq \overrightarrow{g}$ but $\overrightarrow{e}\,\#\,\overleftarrow{h}$ and $\overleftarrow{h}\,\#\,\overrightarrow{g}$, contradicting the **interval condition**.

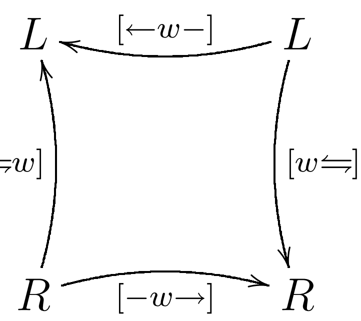
Composing transition relations — the GoI composition

Each transition relation $[w] \subseteq S \times S$ may be *decomposed* into 4 components.

1. $[\leftarrow w-] \;\subseteq\; L \times L$
2. $[\leftrightharpoons w] \;\subseteq\; L \times R$
3. $[w \leftrightharpoons] \subseteq R \times L$
4. $[-w \rightarrow] \subseteq R \times R$.

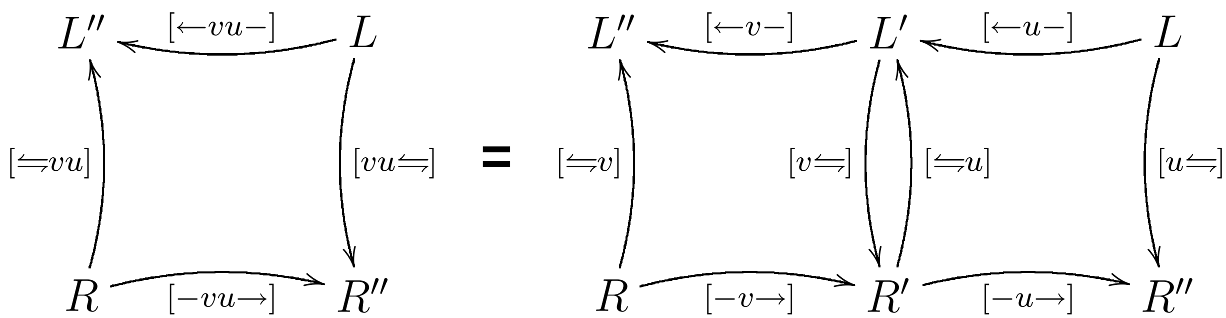
This gives the **matrix** or **directed graph** of the transition relation

$$[w] = \begin{pmatrix} [\leftarrow w-] & [\rightleftharpoons w] \\ [w \rightleftharpoons] & [-w \rightarrow] \end{pmatrix}$$ drawn as :



The GoI composition (cont.)

Given such graphs for $[v]$ and $[u]$, we draw the composite as



This concatenation denotes ‘taking the union over all paths”, giving

$$[\leftarrow vu-] = [\leftarrow v-] \bigcup_{n=0}^{\infty} ([\leftrightharpoons u][v \leftrightharpoons])^n [\leftarrow u-]$$

$$[\leftrightharpoons vu] = [\leftrightharpoons v] \ \cup \ [\leftarrow v-] \bigcup_{n=0}^{\infty} ([\leftrightharpoons u][v \leftrightharpoons])^n [-v \rightarrow]$$

and similarly (dually) for $[-vu \rightarrow]$ and $[vu \leftrightharpoons]$.

Composition and planarity

Using either

(i) Algebraic Manipulations, or

(ii) Categorical Structure (via the identification with Geometry of Interaction),

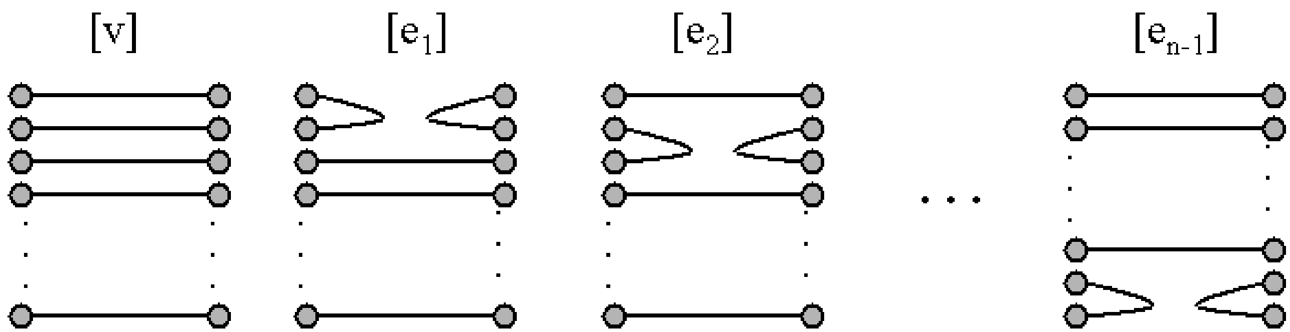
we may show :

1. This composition is **associative**
2. It also preserves **partial injectivity**
3. The composite of undirected transition relations is also **undirected**.
4. The composite of planar transition relations is also **planar**.

An interesting example

Define the 2-way automaton $\mathcal{T}LA_n$ by :

- State set is $S = L \uplus R$, where $L = \{\overleftarrow{1} \leq \overleftarrow{2} \leq \ldots \leq \overleftarrow{n}\}$ and $R = \{\overrightarrow{1} \geq \overrightarrow{2} \geq \ldots \geq \overrightarrow{n}\}$
- Input alphabet is $A = \{v, e_1, e_2, \ldots, e_{n-1}\}$.
- Transition functions given by undirected diagrams :



Properties of $\mathcal{T}LA_n$

It is easy to check that all transition functions :

1. are **undirected** (this is by construction!)
2. are **weakly monotonic**.
3. satisfy the **interval condition.**

Using the GoI composition,

$$[e_i][e_j][e_i] = [e_i] \quad \text{when} \quad |i-j| = 1$$
$$[e_i][e_j] = [e_j][e_i] \quad \text{when} \quad |i-j| > 1$$
$$[e_i][e_i] = [e_i]$$

This gives a representation of the Temperley-Lieb monoid, with a loop value of 1.

The problem with loop-values

We have a representation of the T-L monoid, *in the special case where the loop value is* $\delta = 1$.

— the composition of generators 'forgets about closed loops'.

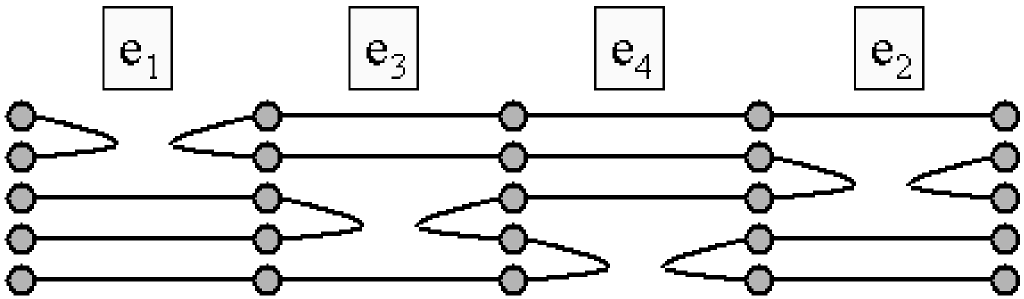
- For the full T-L algebra, we need arbitrary loop values.

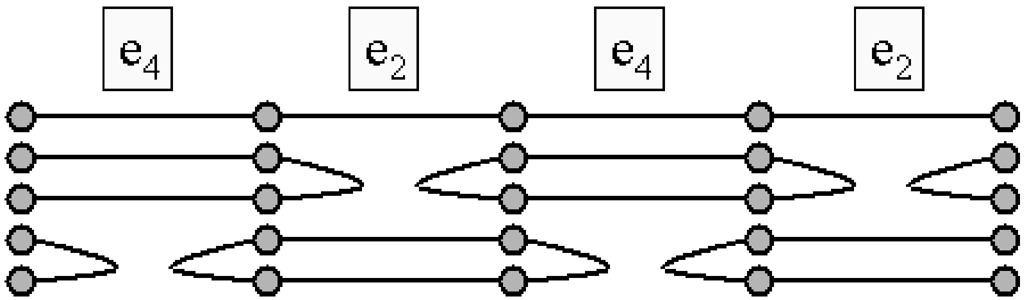
- For the quantum Jones polynomial algorithm, we require $\delta = e^{\frac{2\pi i}{p}}$, for prime p.

Provided we can count closed loops, we can add in a loop value ...

Complexity and 'time-to-termination' of $\mathcal{TLA}_n$

Consider 2 distinct diagrams for $\mathcal{TLA}_5$, with 4 cells on the tape :



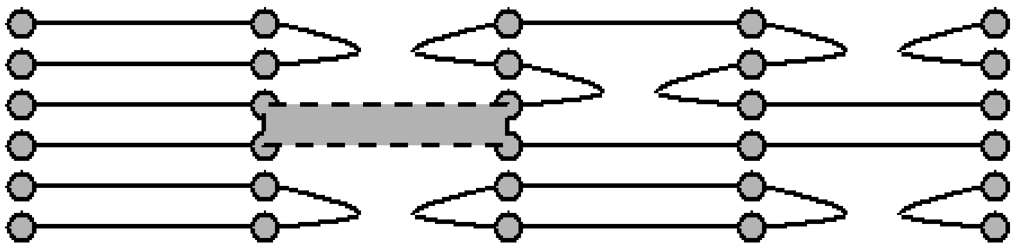


In both cases, the total length of all paths is 20$(= 5 \times 4)$ steps.

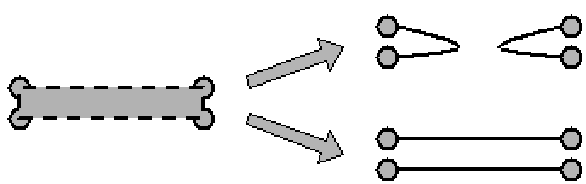
the general case :

A general result Given $\mathcal{T}LA_n$, with k cells on the tape,
the sum of all path lengths *including closed loops* is $n \times k$.

Consider an arbitrary diagram such as



Check that the 2 options

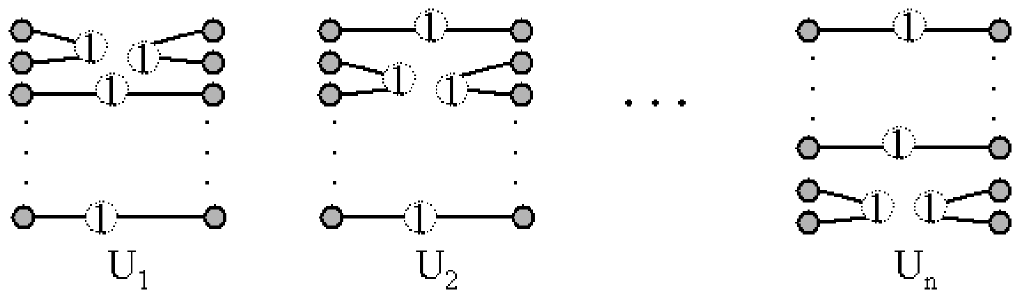


have the same total path length.

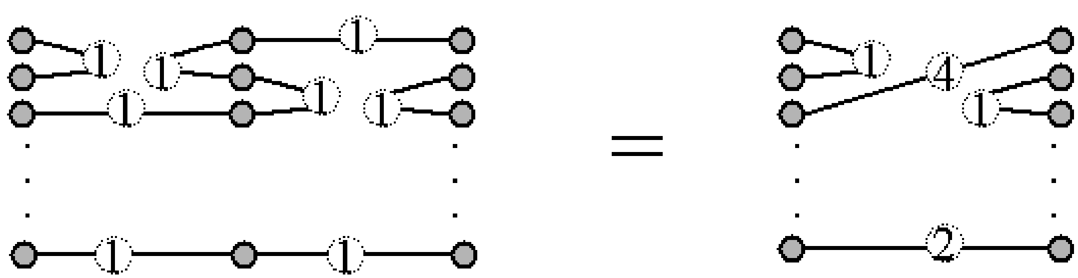
Complexity - diagrammatically

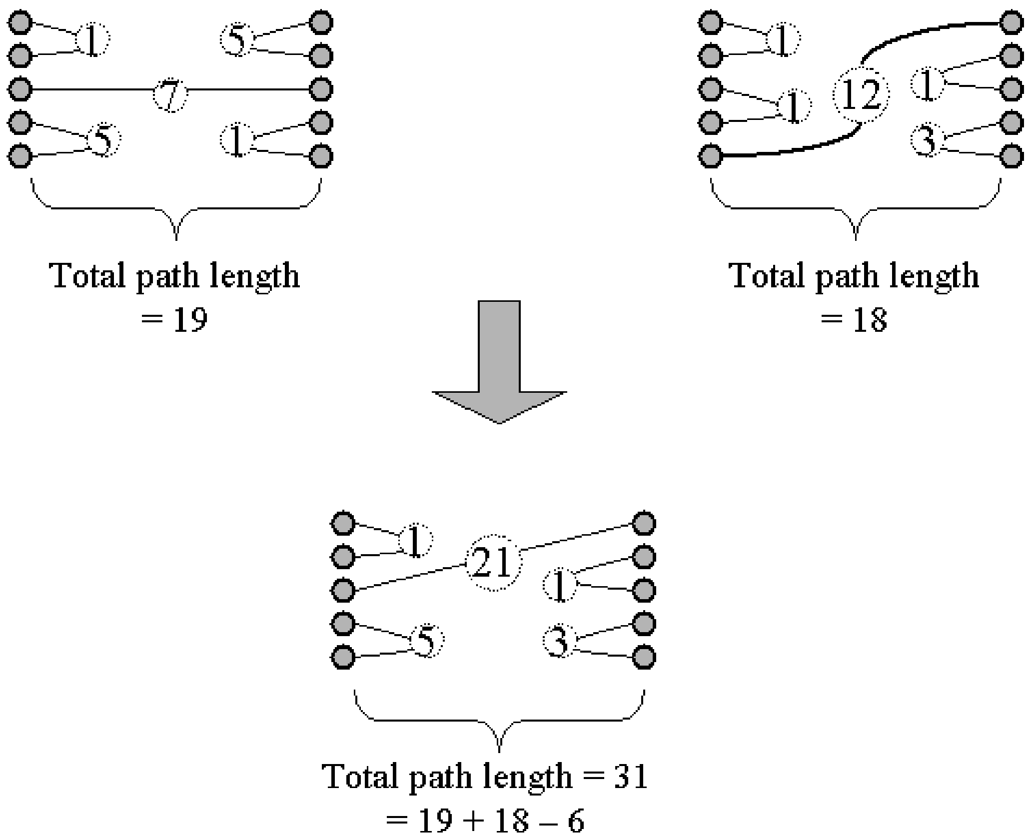
We keep track of 'time-to-termination' by labelling lines in a transition diagram.

For the generators, every path has length 1 :



Relations compose in the usual way — and path labels are added :

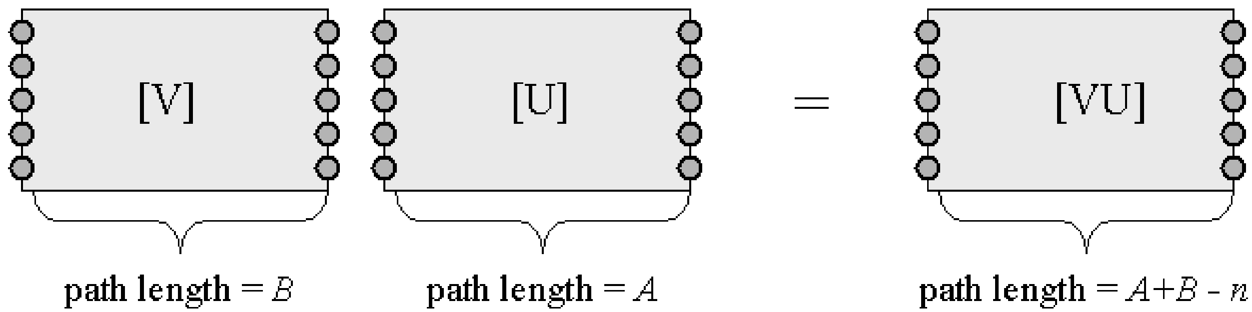




Closed loops show up as ‘missing time’ — the above composite created a closed loop of length 6.

Distinguishing 6 from $4 + 2$ ...

Question Giving a composite such as :



how many closed loops have been added ?

Possible Answers

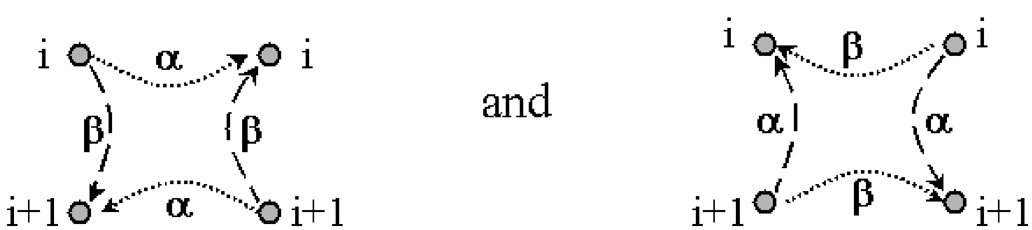
- 1, of length n
- $n/2$, each of length 2.
- somewhere in between ...

How can we count loops ? When either v or u is a generator, at most 1 new closed loop is created.

Future directions :

- The formal setting for 'counting steps' :
  1. Allows us to count closed loops
  2. Lets us represent TLA_n by *unitary maps*.

- In the unitary setting, we can also
  1. label transitions by complex amplitudes, such as



  2. Interpret this as "a coherent superposition of left-moving and right-moving states".

Question : How much of the Jones polynomial algorithm is just a 2-way automaton computation ?