Signed Dual Paths

by Sven Nilsen, 2019

In this paper I formalize Signed Dual Paths, which can construct possible unique paths between points, under some assumptions that are similar to deciding road direction "traffic rules" for cities.

A Signed Path is the following property:

$$\text{is\_signed\_path}(f : T \times T \rightarrow U) = \forall\ a, b, c\ \{\ f(a, c) = f(a, b) + f(b, c)\ \}$$

One can think of this as a shape where every selection of 3 points has the same property.

The type `U` must be signed and support addition. For example, a real number or a vector.

From the definition above one can derive the following properties, by setting `a = c`:

$$\frac{\forall\ a, b\ \{\ f(a, a) = f(a, b) + f(b, a)\ \}}{f(a, a) = 0 \qquad \wedge \qquad f(a, b) = -f(b, a)}$$

This means the points on the paths are objects of a groupoid.

Every Signed Path has a dual version with just the sign flipped.

$$\text{sp\_dual}(f : T \times T \rightarrow U) = \text{neg} \cdot f$$

Signed Paths occur naturally in geometry, for example a line:

$$\text{line}(a : \mathbb{R}^n, b : \mathbb{R}^n) = \backslash(t : \mathbb{R}) = a + (b - a) \cdot t$$

A line in itself is not a Signed Path, because it takes only one parameter, but one can construct a line segment function that returns a vector between two points on the line:

$$\text{line\_segment}(a : \mathbb{R}^n, b : \mathbb{R}^n) = \backslash(t_0 : \mathbb{R}, t_1 : \mathbb{R}) = \text{line}(a, b)(t_1) - \text{line}(a, b)(t_0)$$

The Signed Path **of** a line is a such line segment function.
This transformation is done automatically using a function with a shorthand name `sp`:

$$\text{sp}(f : T \rightarrow U) = \backslash(t_0 : T, t_1 : T) = f(t_1) - f(t_0)$$

Using Higher Order Operator Overloading (HOOO), one can say that `line` has the sub-type:

$$\text{line} : [\text{sp}]\ \text{line\_segment}$$

HOOO abstracts over all start and end-points one can use to construct lines.

By flipping start and end points of a line, the Signed Path becomes the dual:

$$\forall\ a, b\ \{\ sp(line(a, b)) = sp\_dual(sp(line(b, a))\ \}$$

From lines one can construct planes, cubes etc. All these shapes have corresponding Signed Paths. All spaces with Signed Paths are complete, since the Singed Path encodes this semantics.

Since a plane has this property for every selection of 3 points, one can define an arbitrary sub-set of a plane and it will have the same property. Having a Signed Path is a uniform set property.

The parameters of a sub-set of a plane are two real coordinates.

The Signed Path of a single parameter equals the Signed Path of the identity function:

$$sp(id_{\mathbb{R}}) = \backslash(t_0 : \mathbb{R}, t_1 : \mathbb{R}) = t_1 - t_0$$

The generalized version for `n` dimensions is called the Trivial Signed Path and written `$\forall_{sp}$`:

$$\forall_{sp} f := \backslash(t_0 : \mathbb{R}^n, t_1 : \mathbb{R}^n) = t_1 - t_0$$

The Existential Signed Path is just the `sp` transform:

$$\exists_{sp} f \iff sp(f)$$

A line that is transformed in any way onto a plane, has a Trivial Signed Path and an Existential Signed Path. This is not very interesting in itself and also generalized for standard geometric shapes.

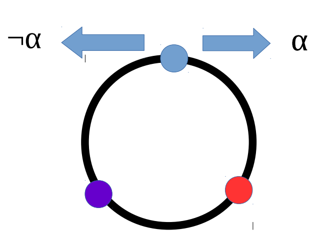
However, instead of lines, consider a dual line where the sign is controlled by a flag:

$$dual\_line(\alpha : bool) = \backslash(a : \mathbb{R}^n, b : \mathbb{R}^n) = \text{if } \alpha\ \{\ line(a, b)\ \}\ \text{else}\ \{\ line(b, a)\ \}$$

A **Signed Dual Path** is a Signed Path of a higher order function that takes a `$\alpha : bool^n$` parameter.

From this idea and further on, it becomes interesting, since Higher Order Operator Overloading over all start and end points gives you geometric combinatorics. In particular, what it means to move directly from one point on a such shape to another. A such path is unique: It has no overlapping line segments.

For example, the trivial gradient of a dual circle increases in both directions until it hits the same point:



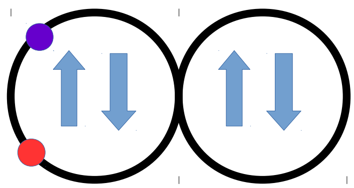
From purple to red, one can move in two ways, through the glue-point, or through the shortest path

Because `$(\exists_{sp} f)(a, a) = 0$`, every set of identical points must have value `0` computed by the Existential Signed Path. If one is allowed to “jump” from one value to another in the Trivial Signed Path for all identical points, then the start and end-point of the dual circle becomes a glue-point. The same law that allows one to move through a glue-point on a dual circle, generalizes to any shape.

By changing `α`, one can control a particle that moves in positive trivial direction. It is possible to construct new dual shapes from existing dual shapes, such that the available choices of some particle is determined by an outer product of `α`-parameters:

$$\alpha^a \times \alpha^b = \alpha^{a+b}$$

For example, by intersecting one dual circle with another dual circle at a single point, any way of navigating from one point on to another can be described by a configuration of `α` that controls the spin "up" or "down" of each circle:



This is similar to deciding "traffic rules" of cities where you are only allowed to drive in one direction.

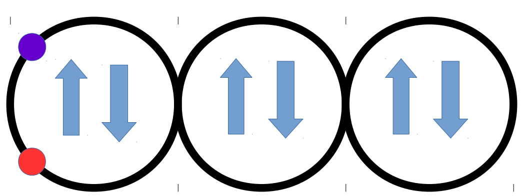
If one picks two points on the same circle, then there are two configurations of the first circle spinning counter-clockwise that permits moving along the shortest path: When the other circle spins up, plus when the other circle spins down.

One can create a table that shows the various paths, identified by a path code:

	0	1	stay/jump (columns), counter-clockwise/clockwise spin (rows)	
00	S0.	S0.	S = South	`S#` means South on dual circle `#`
01	S0.	S0.	N = North	`.` means finished
10	N0.	N0S1.	W = West	`-` means goal is impossible to reach
11	N0.	N0N1.	E = East	`\` means going one stack level up before jumping

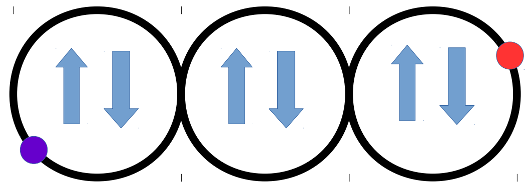
Here, `S0.` is obviously the shortest path. One can infer that `N0.` ignores the second circle since it spins in both directions. A such table shows every possible path that follows positive trivial direction.

Another example with 8 possible paths:



	00	01	10	11
000	S0.	S0.	S0.	S0.
001	S0.	S0.	S0.	S0.
010	S0.	S0.	S0.	S0.
011	S0.	S0.	S0.	S0.
100	N0.	N0.	N0S1.	N0S1S2.
101	N0.	N0.	N0S1.	N0S1N2.
110	N0.	N0.	N0N1.	N0N1S2.
111	N0.	N0.	N0N1.	N0N1N2.

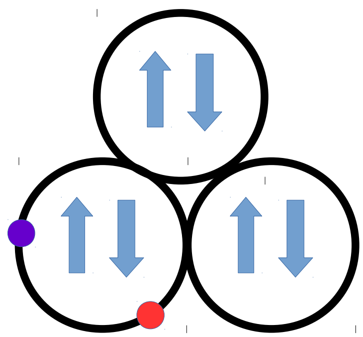
If one chooses two points on two different dual circles, they get “entangled” with those in-between:



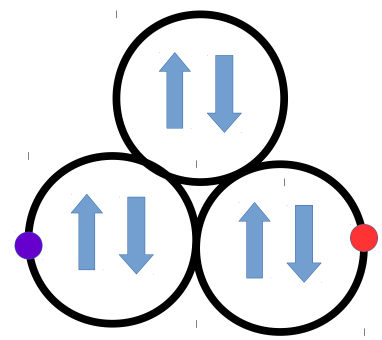
	00	01	10	11
000	S0-	S0-	S0S1-	S0S1S2.
001	S0-	S0-	S0S1-	S0S1N2.
010	S0-	S0-	S0N1-	S0N1S2.
011	S0-	S0-	S0N1-	S0N1N2.
100	N0-	N0-	N0S1-	N0S1S2.
101	N0-	N0-	N0S1-	N0S1S2.
110	N0-	N0-	N0N1-	N0N1S2.
111	N0-	N0-	N0N1-	N0N1S2.

For each configuration above, there is only one possible path.
A more complex example yield different answers depending on where you select the two points:

A



B



Indices of dual circles and intersection points are ordered from left to right.

A	000	001	010	011	100	101	110	111
000	S0.	S0.	S0.	S0.	S0.	S0.	S0.	S0.
001	S0.	S0.	S0.	S0.	S0.	S0.	S0.	S0.
010	S0.	S0.	S0.	S0.	S0.	S0.	S0.	S0.
011	S0.	S0.	S0.	S0.	S0.	S0.	S0.	S0.
100	N0.	N0.	N0.	N0.	N0E1.	N0E1S2.	N0E1\S2.	N0E1S2S0.
101	N0.	N0.	N0.	N0.	N0E1.	N0E1E2.	N0E1\N2.	N0E1E2S0.
110	N0.	N0.	N0.	N0.	N0N1.	N0N1S2.	N0N1\S2.	N0E1S2S0.
111	N0.	N0.	N0.	N0.	N0N1.	N0N1E2.	N0N2\N2.	N0N1E2S0.

S0.(1) N0.(2) N0E1.(3) N0E1S2.(4) N0E1\S2.(5) N0E1S2S0.(6)
N0E1E2.(7) N0E1\N2.(8) N0E1E2S0.(9) N0N1.(10) N0N1S2.(11)
N0N1\S2.(12) N0E1S2S0.(13) N0N1E2.(14) N0N2\N2.(15) N0N1E2S0.(16)

B	000	001	010	011	100	101	110	111
000	S0-	S0-	S0S2.	S0S2.	S0E1-	S0E1S2.	S0S2.	S0S2.
001	S0-	S0-	S0N2.	S0N2W1.	S0E1-	S0E1E2.	S0N2.	S0N2N1W0-
010	S0-	S0-	S0S2.	S0S2.	S0N1-	S0N1S2.	S0S2.	S0S2.
011	S0-	S0-	S0N2.	S0N2N1.	S0N1-	S0N1E2.	S0N2.	S0N2W1W0-
100	N0-	N0-	N0S2.	N0S2.	N0E1-	N0E1S2.	N0E1\S2.	N0E1S2S0-
101	N0-	N0-	N0N2.	N0N2N1.	N0E1-	N0E1E2.	N0E1\N2.	N0E1E2.
110	N0-	N0-	N0S2.	N0S2.	N0N1-	N0N1S2.	N0N1\S2.	N0N1S2S0-
111	N0-	N0-	N0N2.	N0N2W1.	N0N1-	N0N1E2.	N0N1\N2.	N0N1E2.

S0S2.(1) S0E1S2.(2) S0N2.(3) S0N2W1.(4) S0E1E2.(5) S0N1S2.(6)
S0N2N1.(7) S0N1E2.(8) N0S2.(9) N0E1S2.(10) N0E1\S2.(11)
N0N2.(12) N0N2N1.(13) N0E1E2.(14) N0E1\N2.(15) N0N1S2.(16) N0N1\S2.(17)
N0N2W1.(18) N0N1E2.(19) N0N1\N2.(20)

Since `A` has 16 paths, while `B` has `20`, it means possible paths depends on selection of points.