Clink

Matthew Temple

Overview

The concept link, or clink, is a data structure I invented in 2009. It is a messy container, in that objects that were in the clink at one time may not be there later, and in that its contents do not reflect requests from the user to add and remove items from the clink.

Instead, the clink is a reflection of associations, or links, between concepts. The user of the clink informs the clink of these links. The clink is then a reflection of some aspects of this series of concept associations.

The concepts of which the clink may be aware are any type of object. In a specific implementation, they might be references to objects or any other type. In my implementation, the objects of which the clink may be aware are C void *s. The concepts in a clink may correspond to integer indexes, English words, colors, sounds, data structures, instances of any type of variable, or whatever you like.

Reference implementation

http://github.com/clownfysh/cf/tree/master/inferno/clink/

Link function overview

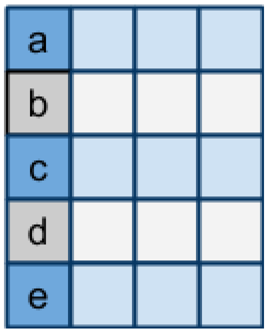
The link function is simply stated: it links two concepts, a and b. The notation for linking b to a is $a \rightarrow b$.

The link function tells the clink that two concepts are linked. The link function does not fail; it is always successful.

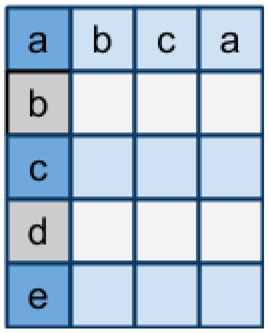
The order of the concepts in the linking is significant. $a \rightarrow b$ means something different than $b \rightarrow a$, and $a \rightarrow b$ does not imply $b \rightarrow a$, nor does $b \rightarrow a$ imply $a \rightarrow b$.

Data structure

The clink is an ordered list of concepts, each containing an ordered list of linked concepts. Here, each horizontal row represents one concept. This clink has five primary concepts, the ordered list (a, b, c, d, e):



Each primary concept in the clink has an ordered list of linked concepts. Here, primary concept a's linked concepts are the ordered list (b, c, a):



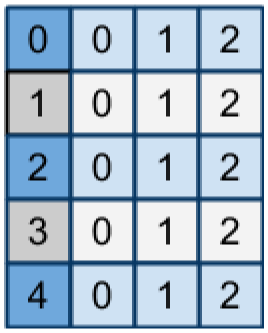
a	b	c	a
b			
c			
d			
e			

That's it. A particular clink has a maximum number of primary concepts and a maximum number of linked concepts per primary concept. These limits are fixed when the clink is created. They're called $max\_concepts$ and max_links. In the clinks pictured above, $max\_concepts = 5$ and $max\_links = 3$.

For convenience in discussing the details of the link function, $[a]$, for a primary concept a, gives the index of the zero-indexed row in which a resides, within the clink data structure. For primary concept a, in the clinks above, $[a] = 0$, $[b] = 1$, $[e] = 4$.

For a linked concept b, within the context of the row in the clink data structure containing information about primary concept a, $[b]$ gives the index of the zero-indexed column in which b resides, within the ordered list of linked concepts linked with primary concept a. For primary concept a, in the clink above, $[b] = 0$, $[c] = 1$, $[a] = 2$.

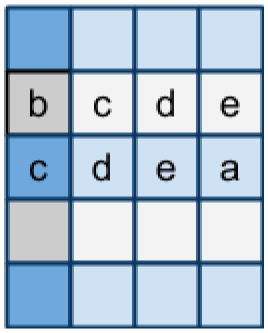
Values for each of these two index functions ($[x]$ for each x that happens to be in that location in the clink) are:



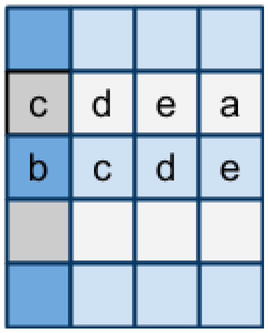
0	0	1	2
1	0	1	2
2	0	1	2
3	0	1	2
4	0	1	2

Link function details

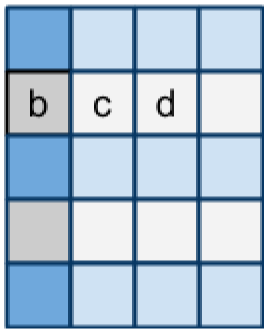
For primary concept indexes x and y, the function $x \leftrightarrow y$ means to swap the primary concepts—the rows—corresponding to those indexes. Given this clink…



..$1 \leftrightarrow 2$ for primary concepts at indexes 1 and 2, yields this clink:

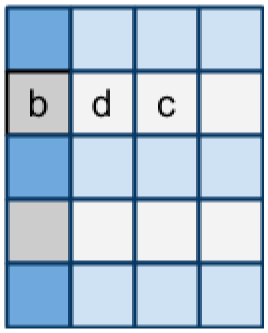


For a particular primary concept, given linked concept indexes x and y, the function $x \leftrightarrow y$ means to swap the two linked concepts—the columns—corresponding to those indexes. This swap takes place only on the row of the particular primary concept in question. Given this clink…



..for primary concept b, for linked concepts at indexes 0 and 1,

$0 \leftrightarrow 1$ yields this clink:



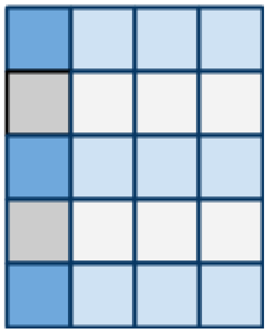
Here's what $a \rightarrow b$ does, algorithmically, with respect to the clink data structure:

1. Note the primary concept.
    a. If a exists as a primary concept within the clink, then if $[a] > 0$, $[a] \leftrightarrow [a] - 1$.
    b. Otherwise, if there's an empty primary concept row in the clink, place the primary concept a into the clink at the empty row with the lowest index.
    c. Otherwise, place the primary concept a into the clink at the last row, the row with the highest index, replacing whatever primary concept is in that row.
2. Note the link between the concepts. For the primary concept a,
    a. If linked concept b exists in the linked concept list, then if $[b] > 0$, $[b] \leftrightarrow [b] - 1$.
    b. Otherwise, if there's an empty column in the linked concept list, place the linked concept b into that list at the empty column with the lowest index.
    c. Otherwise, place the linked concept b into the linked concept list at the last column, the column with the highest index, replacing whatever linked concept is in that column.

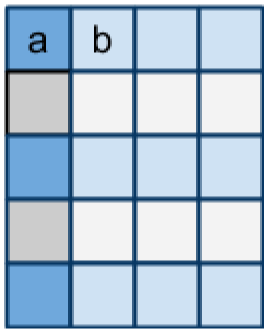
Link function examples

Step-by-step

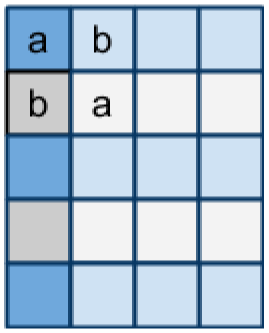
Given the initial, empty clink…



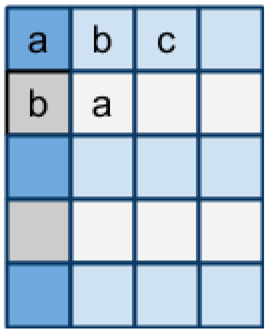
.. $a \to b$ yields:



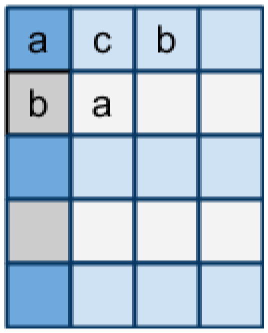
Then, $b \to a$ yields:



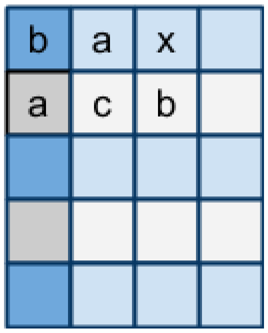
Then, $a \to c$ yields:



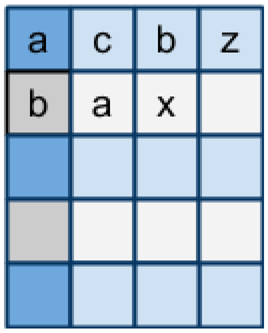
Then, $a \rightarrow c$ yields:



Then, $b \rightarrow x$ yields:

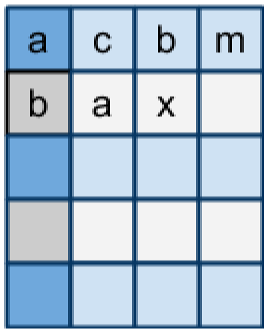


Then, $a \rightarrow z$ yields:



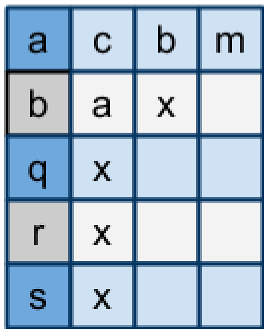
a	c	b	z
b	a	x	

Then, $a \rightarrow m$ yields:



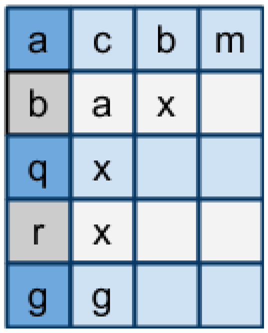
a	c	b	m
b	a	x	

Then, $q \rightarrow x$, $r \rightarrow x$, $s \rightarrow x$ yields:



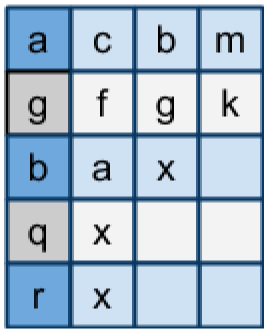
a	c	b	m
b	a	x	
q	x		
r	x		
s	x		

Then, $g \rightarrow g$ yields:



a	c	b	m
b	a	x	
q	x		
r	x		
g	g		

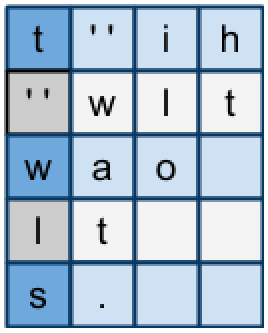
Then, $g \to f$, $g \to k$, $g \to f$ yields:



a	c	b	m
g	f	g	k
b	a	x	
q	x		
r	x		

A sentence, letter-by-letter

Take the string “ It was the best of times, it was the worst of times.”. Link each consecutive pair of letters, starting with a space: $‘\ ’ \to I$, $I \to t, t \to ‘\ ’, ‘\ ’ \to w, w \to a, a \to s, s \to ‘\ ’, ‘\ ’ \to T, T \to h, h \to e$ $, e \to ‘\ ’, ‘\ ’ \to b, b \to e, e \to s, s \to t, t \to ‘\ ’, ‘\ ’ \to o\ o \to f, f \to ‘\ ’,$ $‘\ ’ \to t, t \to i, i \to m, m \to e, e \to s, s \to ,, , \to ‘\ ’, ‘\ ’ \to i, i \to t,$ $t \to ‘\ ’, ‘\ ’ \to w, w \to a, a \to s, s \to ‘\ ’, ‘\ ’ \to t, t \to h, h \to e, e \to ‘\ ’,$ $‘\ ’ \to w, w \to o, o \to r, r \to s, s \to t, t \to ‘\ ’, ‘\ ’ \to o, o \to f, f \to ‘\ ’,$ $‘\ ’ \to t, t \to i, i \to m, m \to e, e \to s, s \to .$. The resulting clink, with $max\_concepts = 5$ and $max\_links = 3$ is:



t	' '	i	h
' '	w	I	t
w	a	o	
I	t		
s	.		

There’s no magic in starting the string with a space. It just allows the clink function to start with the first letter of the sentence, using an arbitrary initial, preceding letter. Alternately, the link function on that sequence of letters could start with $I \rightarrow t$.

Here’s the resulting clink, for that same string, with $max\_concepts = 8$ and $max\_links = 8$:

t	' '	i	h					
' '	I	t	w	o	b	i		
s	t	' '	,	.				
w	a	o						
I	t							
h	e							
a	s							
e	s							

Notice the effect of the clink either running out of space, or not running out of space, during the link function. For instance, notice that while…

t	' '	i	h

..is intact and equally prominent in both the above clinks, there is a difference in the linked concept list for the primary concept ‘ ’. The smaller clink has:

' '	w	I	t

The larger clink has:

' '	I	t	w	..

Also, consider that in an application using a clink to observe a sequence of letters, words, or other concepts, the linking wouldn't necessarily happen in the order shown here. For two letters, I and t, occurring in that order, an application might do any combination of $\{I \to t, t \to I\}$, including doing neither.

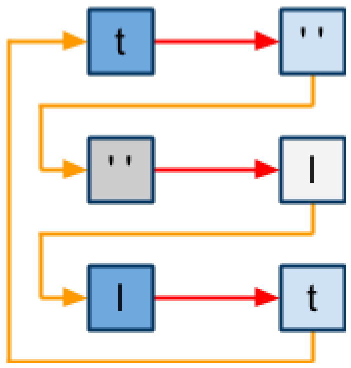
Browse function

A clink is a mind, a memory, a set of associations. It can be browsed while link functions are happening, or once link functions are done.

Train of thought

The train of thought method of browsing a clink works like this:

1. Start with the most prominent primary concept, in this case t .
2. Travel to its most prominent linked concept, in this case ' '.
3. Repeat from step 2 using the last linked concept as the next primary concept.



The train of thought can alternatively start at any primary or linked concept.

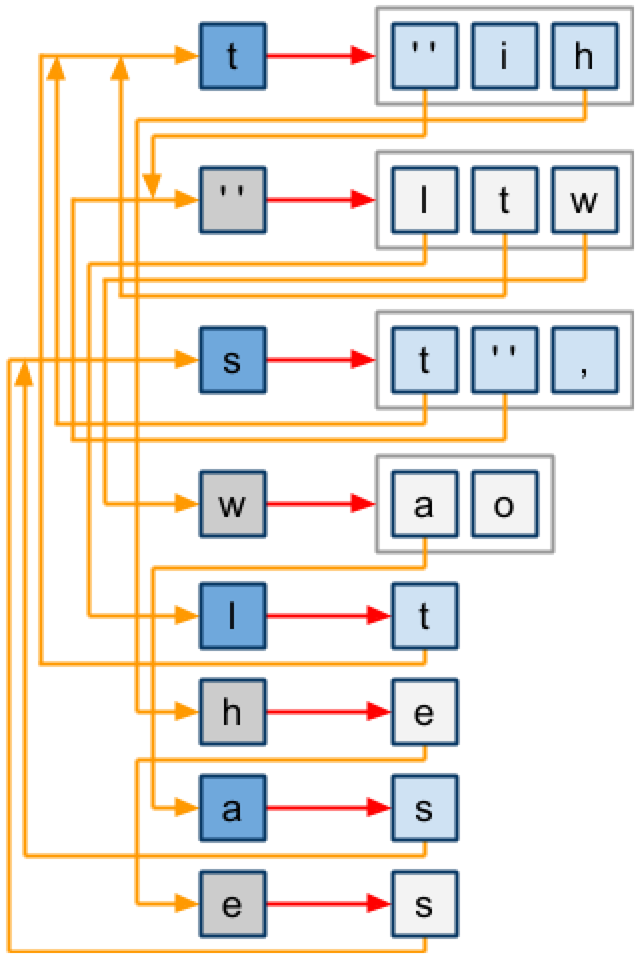
Tree of thought

The tree of thought method of browsing a clink works like this:

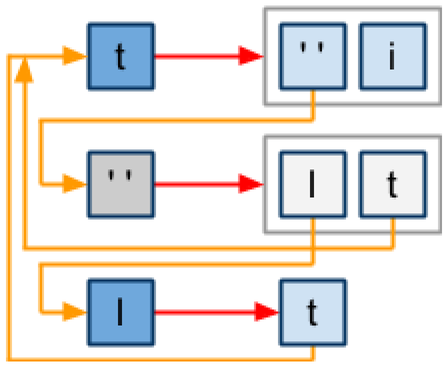
1. Start with the most prominent primary concept, in this case t .
2. Travel to its n most prominent linked concepts. For $n = 3$, in

this case, $(' ', i, h)$.

3. Repeat from step 2 using the last n linked concepts as the next primary concepts.



Here's the tree of thought for the same clink with $n = 2$:



With $n = 1$, the tree of thought is the same as the train of thought.

The tree of thought can alternatively start at any primary or linked concept.

[Look at adding a parameter to the browse functions that imposes a minimum prominence for primary concepts involved in the browse]

[in C, have ways to end cyclic browses..like a max-links-jumped parameter to the recursive functions]

State of mind

A clink, with its link and browse functions, represents something like a state of mind, an active, associative memory that evolves based on its observation of objects in the world, and which possesses cyclic internal states.

The tree of thought browse function may be modified with further state. For instance, instead of a straight $n = x$ way of determining which linked concepts to follow, a subset of the linked concepts may be selected by using other state—state that may or may not be contained within the clink.

For instance, some portion of the recent observation history—some set of recently-observed concepts—might be used to further weight the linked concepts to choose which ones of them to follow during a tree-of-thought-type browse algorithm.

[explore the idea of the clink's concepts having an additional dimension of weightings based on browse function history..concepts that have been most activated by the browse]

[consider making an option for replacing a random primary/linked concept, instead of the least prominent one, when the clink is full..a random or a stochastically at-the-end one]

Applications

EOC System

[show an example of clink used in an EOC system]
http://github.com/clownfysh/cf/tree/master/inferno/clink/eoc/

[expand the clink EOC system to use the browse functions]

[use the visual output and metrics from the EOC system to describe clink system dynamics]

Poetry

I’ve used clink to write poetry based on existing text. The clink reads a string of text. Then, the clink’s tree of thought becomes a poem. Here are clink poems based on one chapter of:

- The Great Gatsby
- Snowbunny
- Things Said in Dreams