STR

Magnitude and topological entropy of digraphs

Steve Huntsman
steve.huntsman@str.us

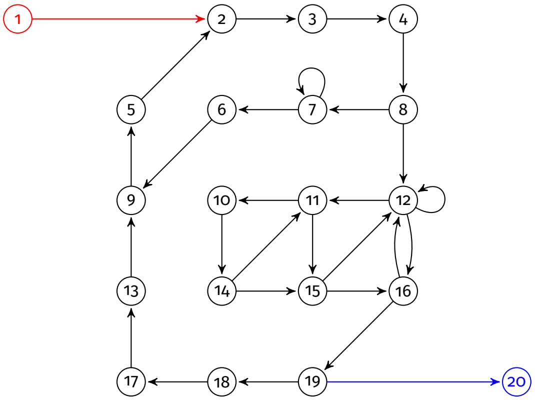
This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

20 July 2022

Part 1: flow graphs

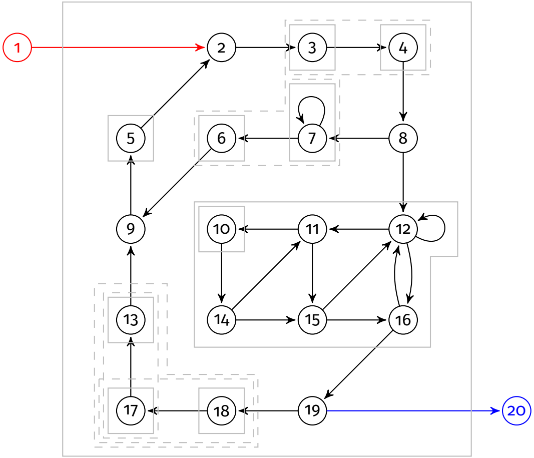
Definition. A *flow graph* is a digraph such that:

- There are unique source (indegree 0) and target (outdegree 0) vertices
- There are unique edges (entry) from the source and (exit) to the target
- Identifying the source and target yields a strongly connected digraph
  - Trivial case: entry = exit



Definition. A *flow graph* is a digraph such that:

- There are unique source (indegree 0) and target (outdegree 0) vertices
- There are unique edges (entry) from the source and (exit) to the target
- Identifying the source and target yields a strongly connected digraph
  - Trivial case: entry = exit



Single-entry/single-exit regions are sub-flow graphs

Definition

Let D be a digraph and $j, k \in V(D)$: j dom k iff every path from a source s in D to k passes through j

- Relation extends to edges; dual relation denoted pdom

Definition

A *single entry/single exit region (SESER)* in a digraph D is an ordered pair of edges (e_1, e_2) s.t.

- e_1 dom e_2
- e_2 pdom e_1
- a cycle in D contains e_1 iff it contains e_2

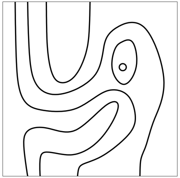
Notes

- (e_1, e_1) is degenerate
- Nondegenerate (e_1, e_2) determined by $(t(e_1), s(e_2))$, where $s(\cdot)$ and $t(\cdot)$ respectively denote the source and target of an edge
- Very easy to find SESERs in DAGs, not so easy in general

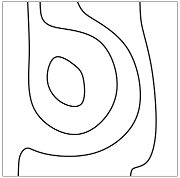
Analogy with tangles (adapted from Baez and Stay, arXiv:0903.0340)

Tan$_k$, the category of *tangles* in a $(k+1)$-dimensional box, has series and parallel monoidal structures

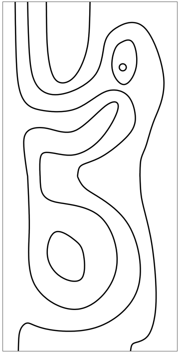
f_1



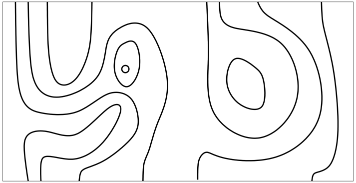
f_2



$f_1 \boxtimes f_2$



$f_1 \otimes f_2$



The category of flow graphs requires finesse with loops

- There ought to be one that behaves like $\mathbf{Tan}_k$ or the category of n-cobordisms [1]
- Unfortunately, categories of digraphs are complicated
  - Problem: how to deal with loops [Brown *et al.* 2008]
  - Identifying vertices "should" induce a graph morphism, but edges must also be preserved, so any edges between identified vertices induce a loop
  - Insofar as loops in a "coarse" flow graph ought to correspond to actual loops in a program, this behavior is bad for applications to program analysis
- Solution: treat loops and non-loop edges differently
- Resulting category **Dgph** is awkward to define but works
- **Flow** is the full subcategory whose objects are flow graphs
  - It has an obvious operad structure with convenient algorithmic framework ("program structure tree")
  - Parallel tensor is slightly nontrivial (tricky when entry/exit edges are the same or adjacent)
  - Series tensor is trivial but still interesting …

[1] Morphisms given by manifolds with $(n-1)$-dimensional manifolds as boundaries (for $n=2$, think of "pajamas for all numbers of heads/arms and legs")

We can tensor flow graphs in series

Proposition

$(\mathbf{Flow}, \boxtimes, e)$ is a monoidal category, where the unit object e is the trivial flow graph and:

- $D \boxtimes D'$: identify exit edge of D with entry edge of D'
- For $f \in \mathbf{Flow}(D, D_f)$ and $f' \in \mathbf{Flow}(D', D'_{f'})$, we obtain $f \boxtimes f' \in \mathbf{Flow}(D \boxtimes D', D_f \boxtimes D'_{f'})$ by identifying the output of f on the exit edge of D with that of f' on the entry edge of D'

Proposition

For a flow graph D, we can form a category $\mathbf{SubFlow}_D$ enriched over **Flow** as follows:

- $\mathrm{Ob}(\mathbf{SubFlow}_D) := E(D)$ (this excludes loops: reflexivity);
- For $e_s, e_t \in \mathrm{Ob}(\mathbf{SubFlow}_D)$, $\mathbf{SubFlow}_D(e_s, e_t) \in \mathrm{Ob}(\mathbf{Flow})$ is the (possibly empty) flow graph with entry e_s and exit e_t;
- The composition morphism is induced by $\boxtimes$;
- The identity element is determined by the trivial flow graph

Unlike $\mathbf{Free}(D)$, $\mathbf{SubFlow}_D$ is always finite and we can build it

Part 2: magnitude

The general theory of magnitude suggests looking for examples

- Let $\mathbf{M} = (\mathbf{M}, \otimes, 1)$ be a monoidal category and $\mathbf{C}$ a (small) $\mathbf{M}$-category, a/k/a a (small) category enriched over $\mathbf{M}$. This means $\mathbf{C}$ is specified by:
  - A set $\mathrm{Ob}(\mathbf{C})$;
  - Hom-objects $\mathbf{C}(j, k) \in \mathbf{M}$ for all $j, k \in \mathrm{Ob}(\mathbf{C})$;
  - Identity morphisms $1 \to \mathbf{C}(j, j)$ for all $j \in \mathrm{Ob}(\mathbf{C})$;
  - And composition morphisms $\mathbf{C}(j, k) \otimes \mathbf{C}(k, \ell) \to \mathbf{C}(j, \ell)$ for all $j, k, \ell \in \mathrm{Ob}(\mathbf{C})$
  - Hom-objects and morphisms are required to satisfy associativity and unitality
- The theory of magnitude introduced by Leinster incorporates a $\mathbf{M}$-category and a semiring S via a "size" map $\sigma : \mathrm{Ob}(\mathbf{M}) \to S$ that is constant on isomorphism classes and that satisfies
  - $\sigma(1) = 1$
  - $\sigma(X \otimes Y) = \sigma(X) \cdot \sigma(Y)$

Magnitude is readily defined in finite settings

Definition

If $n := |\mathrm{Ob}(\mathbf{C})| < \infty$ then its *similarity matrix* $Z \in M(n, S)$ has entries $Z_{jk} := \sigma(\mathbf{C}(j, k))$

Definition

A *weighting* is a column vector w satisfying $Zw = 1$, where the semiring matrix multiplication and column vector of ones are indicated. A *coweighting* is the transpose of a weighting for Z^T

Definition

If Z has a weighting and a coweighting, its *magnitude* is the sum of either (a line of algebra shows these necessarily coincide)

Magnitude is a general concept but examples are lacking

- Magnitude has been the subject of increasing attention over the past 15 years, but almost entirely in the setting of Lawvere metric spaces
  - Over the last two years applications have begun to emerge based on properties of (co)weightings in Euclidean space, which is the only case that has been explored in detail
  - Only one non metric example we know of (involves a **Vect**-category) besides the one presented here
- The Lawvere metric space setting emerges from the choice $\mathbf{M} = (([0, \infty], \geq), +, 0)$
  - Assuming continuity at just a single point, this requires $\sigma(x) = \exp(-tx)$ for some constant t; varying this constant leads to the notion of a *magnitude function*
  - The corresponding enriched categories are precisely the *Lawvere metric spaces*, also known as *extended quasipseudometric spaces* since they generalize metric spaces by allowing distances that are infinite (extended), asymmetric (quasi-), or zero (pseudo-)
  - It turns out that seemingly "adjacent" monoidal structures on $([0, \infty], \geq)$ in fact lead to the same construction, so to move away from the generalized metric space setting at all, it is necessary to move quite far indeed …

Part 3: max-plus magnitude for flow graphs

Topological entropy is a good notion of size for flow graphs

str

Definition

A digraph D determines a *(sub)shift of finite type*, and the corresponding *topological entropy* $h(D) := \lim_{N\uparrow\infty} N^{-1} \log W(D, N)$ measures the growth of the number $W(D, N)$ of paths in D of length N

- Happens that $h(D) = \log \rho(A(D))$ where $A(D)$ = adjacency matrix and ρ = spectral radius

Proposition

$h(\boxtimes_j D_j) = \max_j h(D_j)$

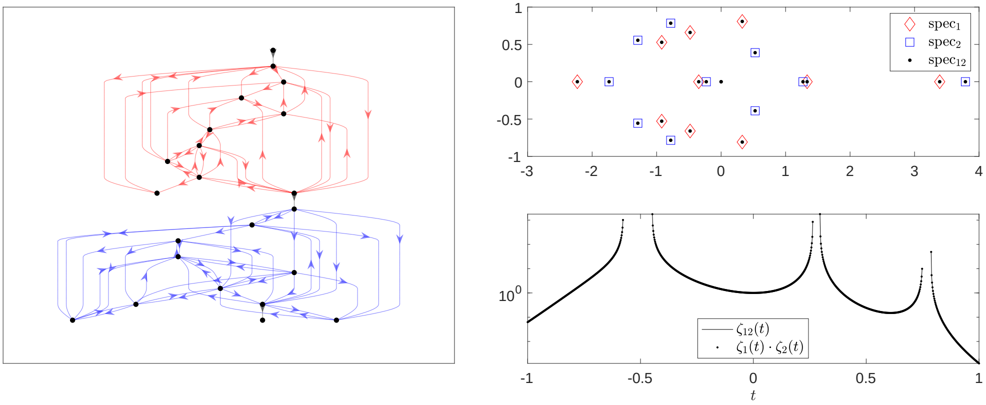
In fact more is true:

spec $A(\boxtimes_j D_j) = \{0\} \cup \bigcup_j \text{spec } A(D_j)$; if we define the zeta function [2] $\zeta_D(t) := 1/\det(I - tA(D))$ then furthermore $\zeta_{\boxtimes_j D_j} = \prod_j \zeta_{D_j}$

---

[2] It turns out (Mizuno, 2001) that $\zeta_D(t) = \prod_{[\gamma]}(1 - t^{|\gamma|})^{-1}$ where γ denotes a prime reduced cycle in D (i.e., a cycle that is not a power ≥ 2 of another cycle and with a no-backtracking restriction) and $[\cdot]$ denotes the equivalence class obtained by quotienting cycles by shifts. This "Euler product" justifies the zeta function terminology.

Topological entropy is a good notion of size for flow graphs



Left: $D_1 \boxtimes D_2$ for two flow graphs D_1 and D_2 on 10 vertices.
Upper right: spectra $\text{spec}_x \subset \mathbb{C}$ of $A(D_x)$ for $x = 1$, $x = 2$, and $x = 12$ with $D_{12} := D_1 \boxtimes D_2$.
Lower right: zeta functions ζ_{12} and $\zeta_1 \cdot \zeta_2$ with $\zeta_x \equiv \zeta_{D_x}$.

We can define magnitude for flow graphs

str

- Recall that max furnishes a monoidal structure on the poset $([0,\infty],\geq)$ of extended nonnegative real numbers, and that categories enriched over this are Lawvere ultrametric spaces
- Similarly, $([-\infty,\infty),\leq,-\infty,\max)$ is a monoidal poset
- This is sufficient data for us to define the magnitude of $\mathbf{SubFlow}_D$ over the *max-plus or tropical semiring*
- Unpacking details:
  - $(Z_D^{\boxtimes})_{st} \equiv Z_D^{\boxtimes}(e_s,e_t) := h(D\langle e_s,e_t\rangle)$
  - If there exist v, w satisfying the max-plus matrix (co)weighting equations $\max_s[v_s + (Z_D^{\boxtimes})_{st}] = 0 = \max_t[(Z_D^{\boxtimes})_{st} + w_t]$ then the maxima of v and w coincide and also equal the magnitude of $Z_D^{\boxtimes}$
  - Linear equations over the max-plus semiring yield "principal solutions" (which may not be *bona fide* solutions in general) $\hat{v}_s := -\max_t(Z_D^{\boxtimes})_{st}$ and $\hat{w}_t := -\max_s(Z_D^{\boxtimes})_{st}$

We can define magnitude for flow graphs

Lemma

$Z_D^\boxtimes$, and hence $\mathbf{SubFlow}_D$, has well-defined magnitude z over the max-plus semiring iff

$$\max_s[-\max_t(Z_D^\boxtimes)_{st}] = z = \max_t[-\max_s(Z_D^\boxtimes)_{st}]$$

- Such a z must be the negative of the largest value in both its row and column of $Z_D^\boxtimes$
- It is not obvious that such a z always exists . . .
- . . . but any nontrivial $D\langle e_s, e_t\rangle \equiv \mathbf{SubFlow}_D(e_s, e_t) \in \mathbf{Flow}$ must be of the form $\boxtimes_j D\langle e_{j-1}, e_j\rangle$ where the $D\langle e_{j-1}, e_j\rangle$ are minimal

Theorem

$Z_D^\boxtimes$, and hence $\mathbf{SubFlow}_D$, has well-defined magnitude over the max-plus semiring

(Co)weighting identifies regions of high topological entropy

Part 4: magnitudes of balls in the universal cover of a digraph

The universal cover of a digraph is a straightforward construction

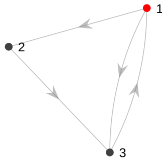
Definition. The *universal cover* $U_D := (V_U, E_U)$ of a weak digraph $D = (V, E)$ is

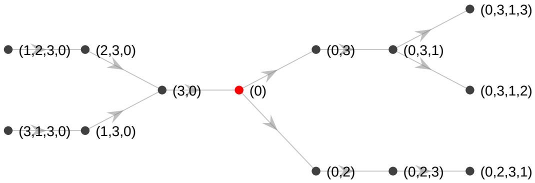
a polytree defined as follows: pick $v_0 \in V$ and set

$$V_U := \{(v_0, v_1, \ldots, v_L) : (v_{j-1}, v_j) \in E; v_{j-1} \neq v_j\} \cup \{(v_L, v_{L-1}, \ldots, v_0) : (v_j, v_{j-1}) \in E; v_j \neq v_{j-1}\}$$

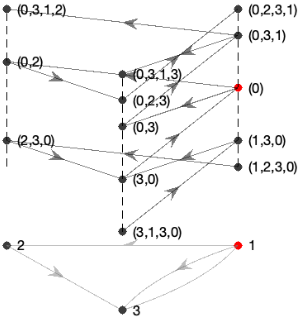
where $v_j \in V$ and $e_j \in E$ identically; and set

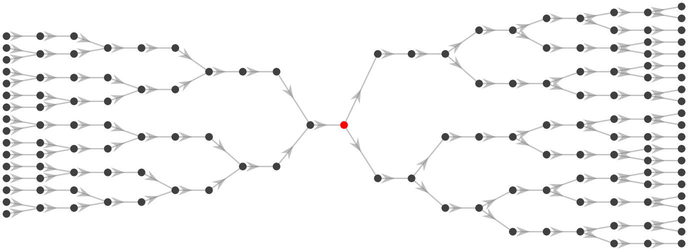
$$E_U := \{((v_0, v_1, \ldots, v_{L-1}), (v_0, v_1, \ldots, v_L)) : (v_{L-1}, v_L) \in E\}$$
$$\cup \{((v_0, v_1, \ldots, v_L), (v_0, v_1, \ldots, v_{L-1})) : (v_L, v_{L-1}) \in E\}.$$





The universal cover of a digraph is a straightforward construction





(L) The portion of U_D with vertices at distance ≤ 3 to or from v_0 with covering of D (at bottom) indicated.
(R) The portion of U_D with vertices at distance ≤ 10 to or from v_0.

Balls in the universal cover of a digraph evoke Katz centrality

Proposition

Let $\gamma \in V_U$. Then there is either a unique path in U_D from v_0 to γ or *vice versa*.

Remark (recall loops are cycles of length 1)

$\left|\{\text{paths from } v_0 \text{ of length } L \text{ in } U_D\}\right| = \left|\{\text{loopless paths from } v_0 \text{ of length } L \text{ in } D\}\right|$

Definition: $B_{v_0}(L)$ is the sub-polytree of U_D (defined with basepoint v_0)

induced by its vertices at (the usual notion of digraph) distance $\leq L$ from (versus to) v_0.

Proposition

If D is loopless, then $B_{v_0}(L)$ is an arboresence with $|V(B_{v_0}(L))| = \sum_{\ell=0}^{L} \sum_k (A^\ell)_{jk}$, where A is the adjacency matrix of D and j is the matrix index corresponding to v_0.

Remark

The *Katz centrality* is $\sum_{\ell=1}^{\infty} \alpha^\ell \sum_i (A^\ell)_{ij}$, where α is restricted to ensure convergence. The Katz centrality of the graph with all edges reversed is therefore $\sum_{\ell=1}^{\infty} \alpha^\ell \sum_k (A^\ell)_{jk}$.

Magnitudes of balls in the universal cover are nice

Lemma

Let F be a DAG whose corresponding undirected graph is a forest. [3] Then the magnitude function of F (i.e., the magnitude of the matrix $\exp(-td_{jk})$ where d is the usual Lawvere metric on F) is

$$\mathrm{Mag}(F,t) = |V(F)| - |E(F)|e^{-t}$$

Remark

Since an arborescence (or more generally a polytree) has one more vertex than it has edges, the lemma above yields that for D loopless, the magnitude function of $B_{v_0}(L)$ is

$$\mathrm{Mag}(B_{v_0}(L),t) = |V(B_{v_0}(L))| - (|V(B_{v_0}(L))| - 1)e^{-t}$$

and there is an elementary algorithm for computing $|V(B_{v_0}(L))|$. If D is loopless and strong, we have

$$h(D) = h(U_D) =: \lim_{L\uparrow\infty} L^{-1} \log|V(B_{v_0}(L))|, \quad \forall v_0.$$

[3] Note that if F is a polytree, then $|V(F)| = |E(F)| + 1$.

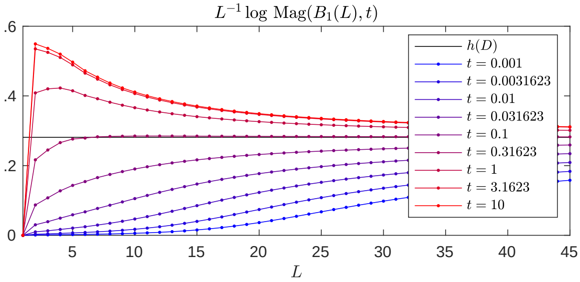
Magnitudes of balls in the universal cover are nice

Proposition

Let D be a strong loopless digraph and $v_0 \in V(D)$. Then

$$\lim_{L \uparrow \infty} L^{-1} \log \mathrm{Mag}(B_{v_0}(L), t) \leq h(D)$$

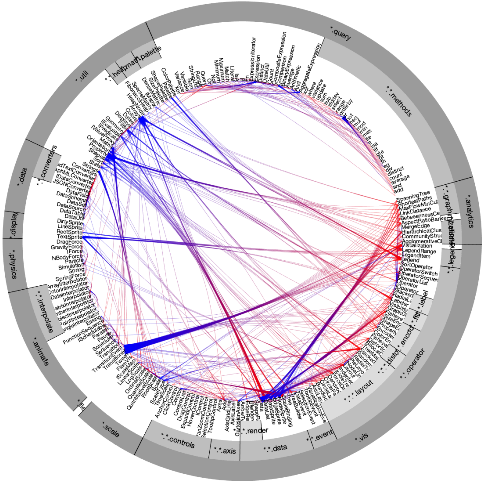
with equality at $t = \infty$, and the left hand side is independent of v_0 for any t.



$L^{-1} \log \mathrm{Mag}(B_{v_0}(L), t) \to h(D)$ for $t > 0$, but depends strongly on t even for fairly large L.

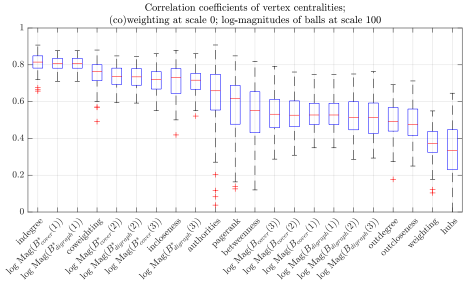
Log-magnitudes of small balls are useful features for graph matching

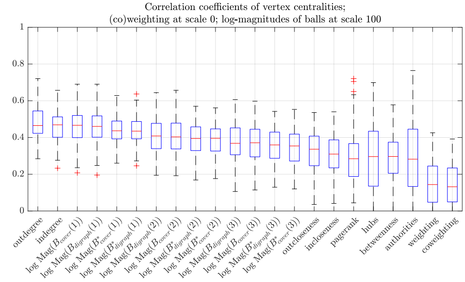
Import graph of `Flare` software hierarchy:
sources (resp., targets) colored red (resp., blue)



- $N = 100$ realizations of two random subgraphs: removed edges with probability $3/4$ and kept the largest weak component: computed
  - (Co)weightings at scale 0
  - Log-magnitudes of balls of radius ≤ 3 at scale $t = 100 \approx \infty$
  - Common vertex centralities
- Computed correlation coefficients for all items above on vertices common to both subgraphs
- Coweighting and log-magnitudes of balls in the universal cover of the digraph with edges reversed are very strongly correlated …
- Similarly considered $N = 100$ realizations of an Erdős-Renyí digraph ($n = 100$ vertices; edge probability $= 4/n$), formed two subgraphs by removing edges with probability $1/2$, then keeping the largest weak component …

Log-magnitudes of small balls are useful features for graph matching





Upper panel: `Flare` import digraph

* indicates a ball in the digraph with all edges reversed.

As L increases, boundary effects cause the log-magnitudes of balls in the universal cover to become (slightly) more correlated to each other than the log-magnitudes of balls in the digraph itself. Note that the three best-performing centralities are computing almost exactly the same thing.

Lower panel: Erdős-Renyí digraph with $n = 100$ vertices and edge probability $q = 4/n$

Thanks

Backup

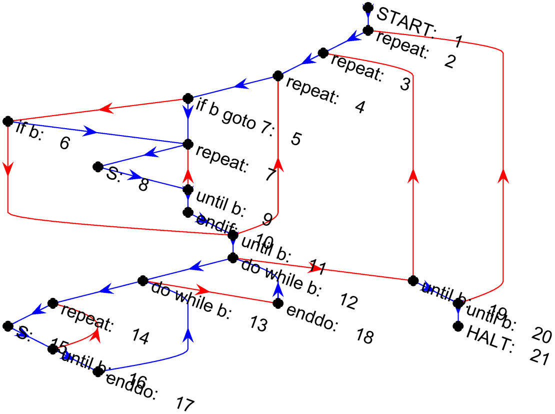
The category Dgph of reflexive digraphs

- Object is $D = (U, \alpha, \omega)$
  - U is a set
  - $\alpha, \omega : U \to U$ are *head* and *tail* functions that satisfy $\alpha \circ \omega = \omega$ and $\omega \circ \alpha = \alpha$
- For $D' = (U', \alpha', \omega')$, a morphism $f \in \mathbf{Dgph}(D, D')$ is a function $f : U \to U'$ such that $f \circ \alpha = \alpha' \circ f$ and $f \circ \omega = \omega' \circ f$
- The *vertices* of $D = (U, \alpha, \omega)$ are the (mutual) image $V \equiv V(D)$ of α and ω
- The *loops* are the set $L \equiv L(D) := \{u \in U : \alpha(u) = \omega(u)\}$ (so that $V \subseteq L$),
- The *edges* are the set $E \equiv E(D) := U \backslash L$
- We recover the usual notion of a digraph by considering $\alpha \times \omega$ and its appropriate restrictions on U^2, L^2, and E^2:
  - E.g., we can abusively write $E = (\alpha \times \omega)(E^2)$, where the LHS and RHS respectively refer to usual and reflexive notions of digraph edges
- Thus a morphism $f : U \to U'$ restricts to $f|_V : V \to V'$, $f|_L : L \to L'$, and $f|_E : E \to U'$
- Since morphisms are only partially specified by their actions on vertices, defining **Flow** as a full subcategory of **Dgph** is essentially a convention about vertex identification

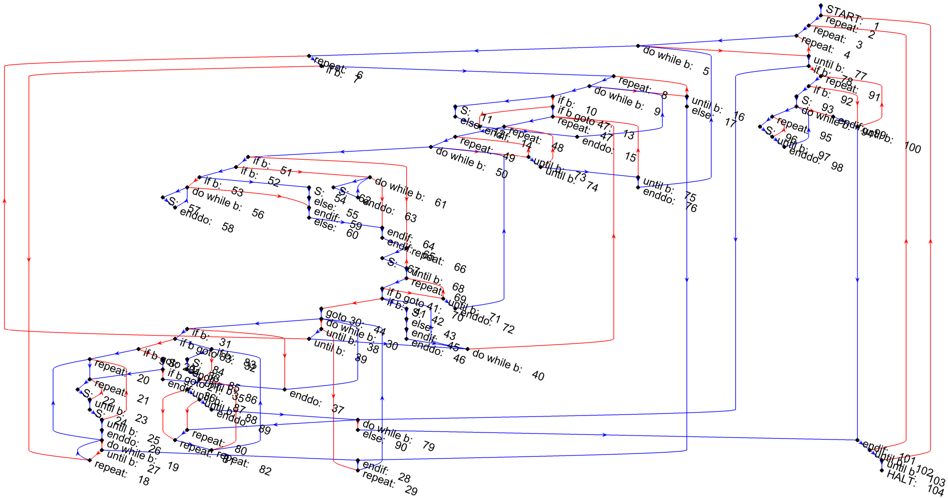
Control flow graphs (CFGs) model computational paths

Each S is its own statement or subroutine;
each b is its own Boolean predicate;
branches are colored according to associated b evaluating to ⊤ or ⊥

```
1    START
2    repeat
3        repeat
4            repeat
5                if b goto 7
6                if b
7                    repeat
8                        S
9                    until b
10               endif
11           until b
12           do while b
13               do while b
14                   repeat
15                       S
16                   until b
17               enddo
18           enddo
19       until b
20    until b
21    HALT
```



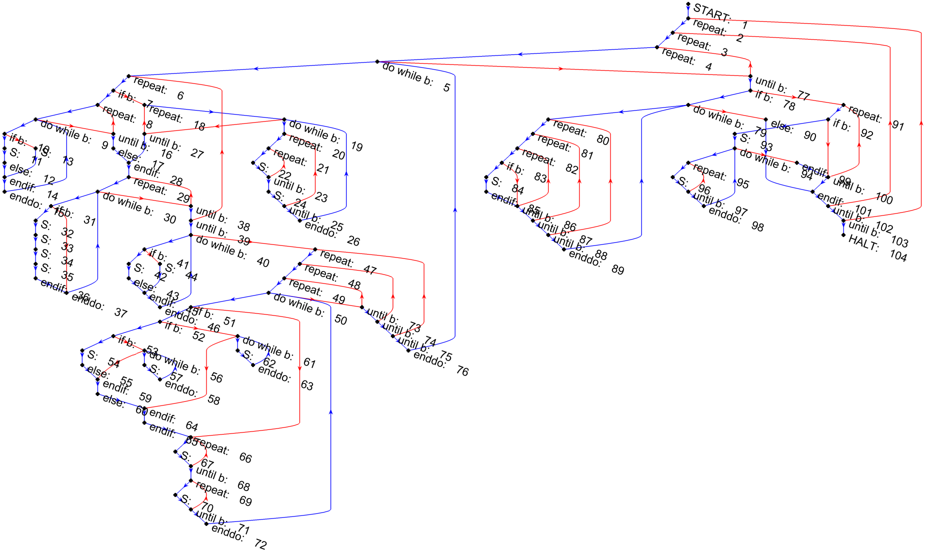
In practice CFGs are much bigger than this



Multiresolution analysis lets us cope with CFG scale

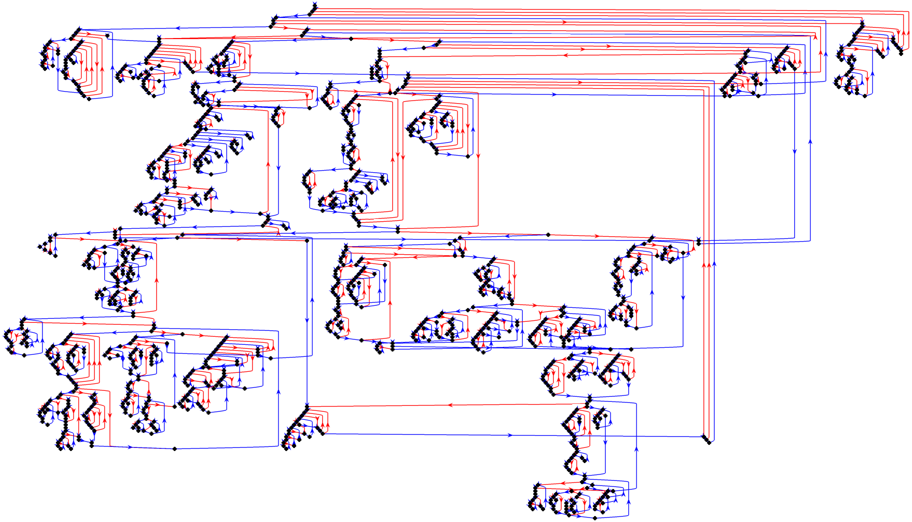
- Code restructuring can eliminate `goto`s [Zhang and D'Hollander, 2004]
  - Effective version of Böhm-Jacopini structured program theorem
  - Dovetails with the constructions we discuss here
- Subroutines are programs in their own right
  - Recursively (de)compose programs: multiresolution analysis
  - Much more interesting when trying to parallelize source or reverse engineer binary code than when merely parsing Python
- Similar considerations inform myriad other domains where flow graphs are good process models

A CFG with no `goto`s is nicer but still complicated



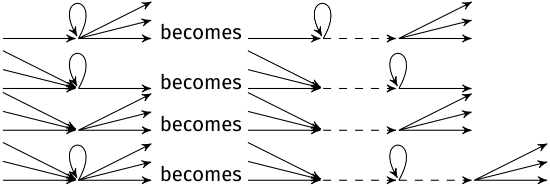
A CFG with no gotos is nicer but still complicated

str



Stretching flow graphs helps coax SESERs into existence

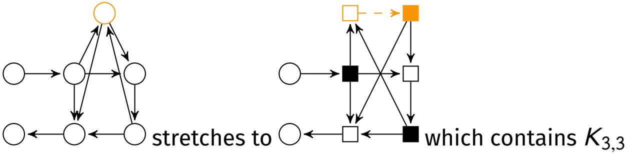
- Insert edges into a flow graph as follows:



Lemma

The resulting *stretching* is well defined

- There is a planar flow graph whose stretching is nonplanar:



The program structure tree organizes SESERs

Definition

The *interior* of a SESER (e_s, e_t) is the set of vertices on paths from $t(e_s)$ that do not encounter $t(e_t)$

- Differs from flawed def. 6 of [Johnson, Pearson, Pingali, 1994]
- §5 of [Boissinot *et al.*, 2012] illustrates this and why it matters

Definition

A nondegenerate SESER (e_1, e_2) is *canonical* if

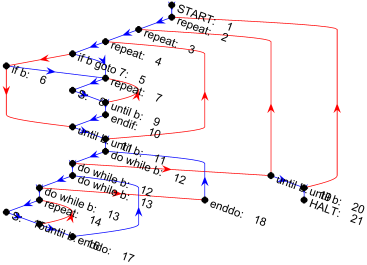
- For any SESER (e_1, e_2') we have e_2 dom e_2'
- For any SESER (e_1', e_2) we have e_1 pdom e_1'

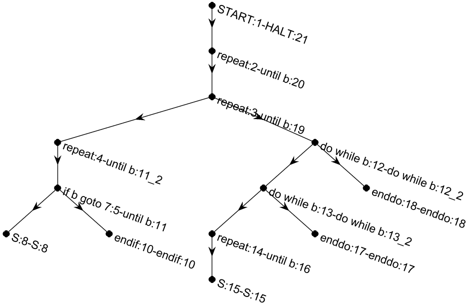
Theorem (easily salvaged from JPP'94)

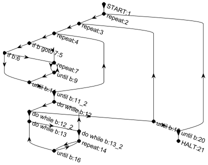
Interiors of distinct canonical SESERs are either disjoint or nested

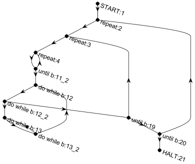
- *"Canonical = minimal"*
- *Inclusion relation induces the program structure tree (PST)*

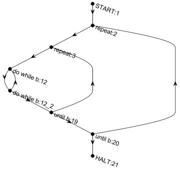
Stretching, PST, and “coarsening” 1, 2, 3, & 6x



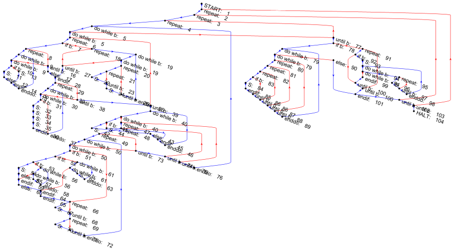


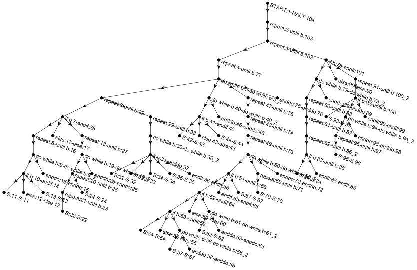


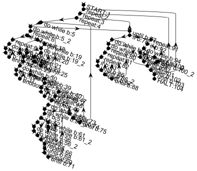


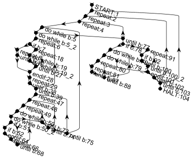


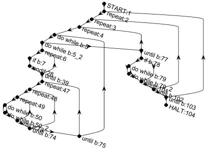
Stretching, PST, and “coarsening” 1, 3, 5, & 13x











Using the PST to "coarsen" CFGs is like a pullback

Definition

For $j, k \in V$, the *absorption* of k into j is the morphism induced by identifying j and k and (if $k \neq j$) annihilating any loop at j (by mapping it to the vertex j)

- Definition chosen to dovetail with ideas of program abstraction
- Absorbing k, m into j is equivalent to absorbing m, k into j
- For $D, D' \in \text{Ob}(\textbf{Flow})$ with $D' \subset D$, define the absorption of D' to be the image of absorbing the interior of D' into its source (considered as a vertex in D)
  - Amounts to replacing D' w/ single edge from source to target

Definition

The *coarsening* $\circledcirc D$ is obtained by absorbing the sub-flow graphs corresponding to leaves of $\text{PST}(D)$

Observation: the pullback of $a \xrightarrow{g \circ f} c \xleftarrow{g} b$ is $a \xleftarrow{id} a \xrightarrow{f} b$

- In particular, f is the pullback of $g \circ f$ by g
- We may therefore think of $\circledcirc D$ literally as a kind of pullback of D by the leaves of $\text{PST}(D)$

Flow graphs give rise to an obvious operad

Let $P(n) := \{\text{flow graphs with } n \text{ ordered edges}\}$ and define

$$\circ : P(n) \times P(k_1) \times \cdots \times P(k_n) \to P(k_1 + \cdots + k_n)$$
$$(D, D_1, \ldots, D_n) \mapsto D \circ (D_1, \ldots, D_n)$$

by replacing the jth edge in D with D_j in the obvious way

- Edge ordering on $D \circ (D_1, \ldots, D_n)$ inherited from constituents

Theorem

The triple $\{e, \{P(n)\}_{n=1}^{\infty}, \circ\}$, where e denotes the trivial flow graph, forms a operad (in **Set***)*

Lemma

If $D \in P(n)$ and $\circledcirc D_j = e \neq D_j$ for $j \in [n]$, then $\circledcirc(D \circ (D_1, \ldots D_n)) = D$

- $\circ$ and $\circledcirc$ are complementary

We can tensor flow graphs in parallel

str

Theorem

$(\mathbf{Flow}, \otimes, e)$ is a monoidal category, where $\otimes$ is defined below and the unit object e is the trivial flow graph

- *$D \otimes D' := D \sqcup D' / \sim$, where $\sim$ is mostly obvious but has messy technical details to account for the cases where entry and exit edges are either identical or adjacent*
  - *$\sim$ always identifies entry edges*
  - *$\sim$ identifies exit edges if interiors are unaffected...*
  - *...otherwise $\sim$ collapses a "small" factor to make things work*
  - *Constraints on how to fill in these technical details are perhaps the main benefit of invoking category theory ab initio*
- *Let $[\cdot]$ denote an equivalence class under $\sim$ and set*

$$(f \otimes f')(k) := \begin{cases} [(f(j), 0)] & \text{if } k = [(j, 0)] \\ [(f'(j'), 1)] & \text{if } k = [(j', 1)] \end{cases}$$

along with an implied extension to edges

Summary

- $\boxtimes$ corresponds to sequential execution
- $\otimes$ corresponds to an `if` (or parallel execution)
-  $\circ\ (e, \bullet, e, e)$ corresponds to a `do while` or `repeat`
- By the structured program theorem and an effective version thereof, we have a category-theoretical framework for (de)composing structured programs up to statement/predicate vertex labels and $\top/\bot$ edge labels
  - Exercise: eliminate the “up to” disclaimer

Takeaway

Requiring that flow graphs exhibit various category-theoretical desiderata places strong but satisfiable restrictions on them that can usefully inform the architecture of program analysis platforms, program synthesizers, compilers, etc.

Generalized metric spaces are hard to get away from ...

str

- As mentioned, the usual addition operation on $([0,\infty],\geq)$ gives a monoidal structure that essentially mandates $\sigma(x)=\exp(-tx)$ for some constant t

Proposition

Let f be a strictly increasing bijection from $[0,\infty]$ to a subset of $[-\infty,\infty]$ containing 0. Then $x\otimes y := f^{-1}(f(x)+f(y))$ gives rise to a strict symmetric monoidal structure on $([0,\infty],\geq)$ with monoidal (additive) unit $f^{-1}(0)$

- A category $\mathbf{C}$ enriched over the strict symmetric monoidal category above has, for every $j,k\in \mathrm{Ob}(\mathbf{C})$, some $\eta_{jk} := \mathbf{C}(j,k)\in[0,\infty]$ such that $\eta_{jj}=f^{-1}(0)$ and $\eta_{jk}\otimes\eta_{k\ell}\geq\eta_{j\ell}$
- That is, we have the triangle inequality $f(\eta_{jk})+f(\eta_{k\ell})\geq f(\eta_{j\ell})$
- Turns out that if $f(\eta)=d$, our similarity matrix Z takes values in the semiring $\mathbb{R}$ with the usual structure (as opposed to some more exotic choice), and we require (any) continuity of σ, then $Z=\sigma(\eta)=\sigma(f^{-1}(d))=\exp(-\tau d)$, i.e., this attempted generalization actually has no material effect

Generalized metric spaces are hard to get away from ... really hard ...

- What about a more exotic semiring structure on $\mathbb{R}$?

Proposition

Let g be a strictly increasing function from $[-\infty, \infty]$ to itself, and taking on the value 0 (and also 1 for the final part of the statement). Then $x \oplus y := g^{-1}(g(x) + g(y))$ gives rise to a strict symmetric monoidal structure on $([-\infty, \infty], \geq)$ with monoidal (additive) unit $g^{-1}(0)$. Moreover, additionally taking $x \odot y := g^{-1}(g(x) \cdot g(y))$ gives a semiring with multiplicative unit $g^{-1}(1)$

- If $g(x) := \text{sgn}(x) \cdot |x|^p$ for $p > 0$, we get the semiring $([-\infty, \infty], \oplus, 0, \cdot, 1)$
- If $g(x) := \exp(-\tau x)$ for $\tau < 0$, then we get the semiring $([-\infty, \infty], \oplus, -\infty, +, 0)$

Generalized metric spaces are hard to get away from …really hard!

str

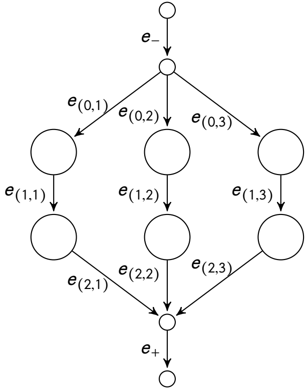
- Trying this more exotic semiring structure $x \oplus y := g^{-1}(g(x) + g(y))$ and $x \odot y := g^{-1}(g(x) \cdot g(y)) \ldots$
- Weighting equation $Zw = 1$ unpacks first to $\bigoplus_k (Z_{jk} \odot w_k) = g^{-1}(1)$ in semiring arithmetic and then to the matrix equation $g[Z]g[w] = 1$ in ordinary arithmetic
- Since $Z = \sigma[\eta]$ and $f[\eta] = d$, we have $Z = \sigma[f^{-1}[d]]$
- Meanwhile, we have the generalized Cauchy equation $\sigma(x \otimes y) = \sigma(x) \odot \sigma(y)$, which unpacks to $\sigma(f^{-1}(f(x) + f(y))) = g^{-1}(g(\sigma(x)) \cdot g(\sigma(y)))$
- Defining $h := g \circ \sigma \circ f^{-1}$, this becomes $h(f(x) + f(y)) = h(f(x)) \cdot h(f(y))$
  - I.e., h satisfies the usual Cauchy equation; assuming any continuity, we have $h(d) = \exp(-\tau d)$
- Since $g[Z] = h[d]$, the weighting equation is $h[d]g[w] = 1$, which apart from the transformation of w is the same as in ordinary arithmetic

Magnitude for flow graphs: example

If $D := \otimes_{k=1}^{K} \boxtimes_{j=1}^{J_k} D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ with PSTs of $D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ all trivial (i.e., there are no nontrivial sub-flow graphs) then $(Z_D^{\boxtimes})_{(j_0,k),(j_1,k)} = \max_{j_0<j\le j_1} h(D\langle e_{(j-1,k)}, e_{(j,k)}\rangle)$, $(Z_D^{\boxtimes})_{-\infty,\infty} = h(D)$, where $\mp\infty$ indicate the entry and exit edges of D, and all other entries of $Z_D^{\boxtimes}$ are trivial

The nontrivial (co)weighting components are

$$\begin{aligned} w_{(j,k)} &= -\max_{j_0<j} h(D\langle e_{(j_0,k)}, e_{(j,k)}\rangle) \\ &= -\max_{j_0<j} h(D\langle e_{(j_0,k)}, e_{(j_0+1,k)}\rangle); \\ v_{(j,k)} &= -\max_{j_1>j} h(D\langle e_{(j,k)}, e_{(j_1,k)}\rangle) \\ &= -\max_{j_1>j} h(D\langle e_{(j_1-1,k)}, e_{(j_1,k)}\rangle) \end{aligned}$$

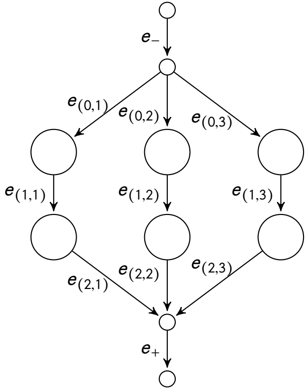


Flow graph of the form $D := \otimes_{k=1}^{K} \boxtimes_{j=1}^{J_k} D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ for $J_k \equiv 2$ and $K = 3$. The large nodes indicate nontrivial interiors of sub-flow graphs

Magnitude for flow graphs: example

That is …

…the weighting w and coweighting v respectively encode the cumulative forward and reverse maxima of the topological entropy along the K "backbones" $\boxtimes_{j=1}^{J_k} D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ of D. In particular, $v_{(j_*-1,k)} = w_{(j_*,k)}$ when $j_* = \arg\max_j h(D\langle e_{(j-1,k)}, e_{(j,k)}\rangle)$



Flow graph of the form $D := \otimes_{k=1}^{K} \boxtimes_{j=1}^{J_k} D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ for $J_k \equiv 2$ and $K = 3$. The large nodes indicate nontrivial interiors of sub-flow graphs