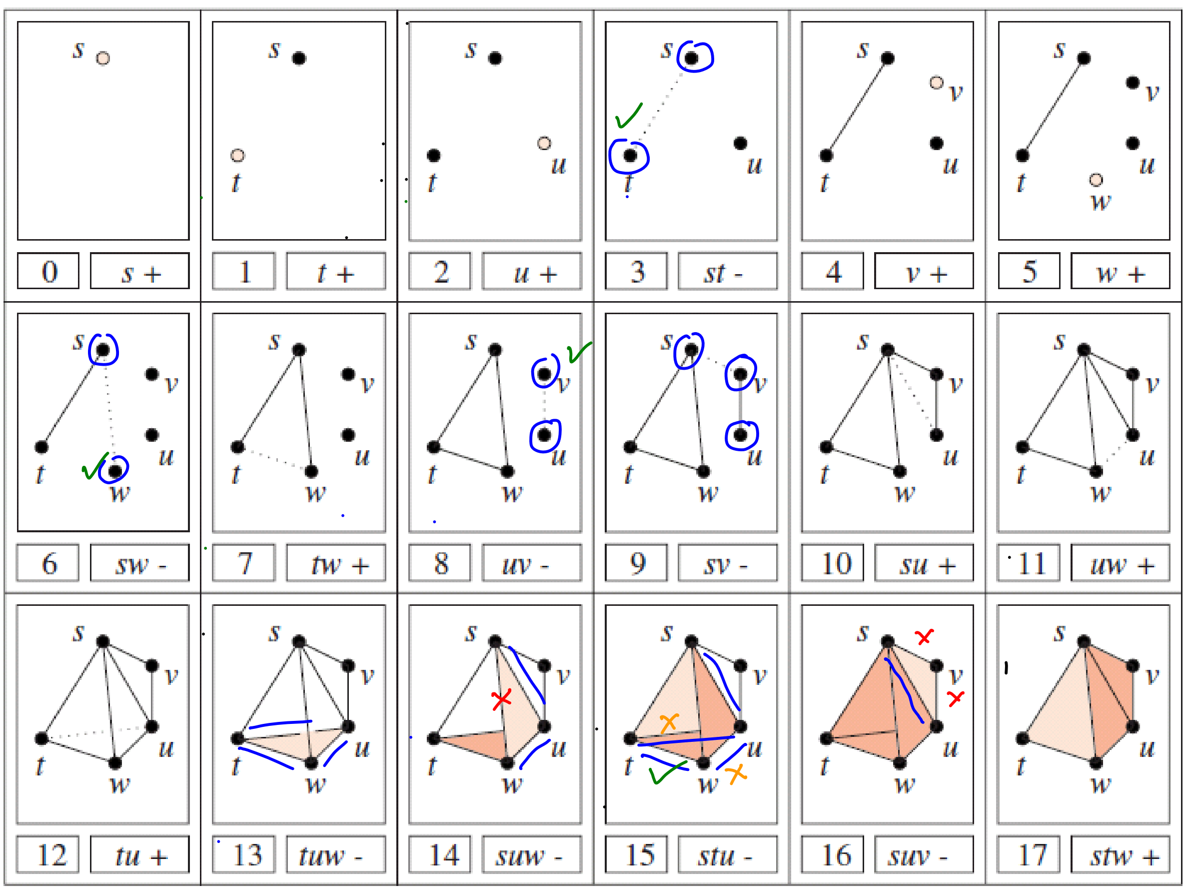
# MATH 529 – Lecture 21 (03/26/2024)

Today: * example of persistence algorithm
* implementation of pairing

## Example

The convention: the latest simplex to come in is shaded lightly (for vertices and triangles) or is shown dashed (edges).

In this filtration, all k-simplices do **not** come in before the (k+1)-simplices. Indeed, edge st comes in before vertices v and w. All we need to insure is that $K^i \subseteq K^j$ for the filtration, by insuring that all proper faces of $\sigma^i$ come in before $\sigma^i$ itself.




pairings

(t, st)

(w, sw)

(v, uv)

(u, sv)

(tu, tuw)

(uw, suw)

(tw, stu)

(su, suv)

$\Gamma(tuw) = \{tu, uw, tw\}$

$\Gamma(suw) = \{uw, su\}$

$\Gamma(stu) = \{tu, su\}$
$\downarrow$
uw
tw

$\Gamma(suv) = \{su\}$

# Details of the pairing

For the negative 1-simplex $sw$, we consider $\Gamma = \{s, w\}$, and we choose the younger 0-simplex between $s$ and $w$, which came in at time (or index) 0 and 5, respectively, pairing $(w, sw)$.

Consider the negative 1-simplex $sv$. We start with $\Gamma = \{s, v\}$, and $v$ is younger. But $v$ is already paired, so we add $u$ to $\Gamma$, as $u$ is homologous to $v$ (since $\overline{uv}$ is present). $u$ is positive, and is younger than $s$. So we pair $(u, sv)$.

This calculation illustrates that an edge could (often) be paired with a simplex that is not its own face.

Pairing for stu: $\Gamma$ has ~~st~~ $\{tu, su\}$. Here, $tu$ is younger but is already paired.
$\hookrightarrow$ st is $-$ve

Due to $\Delta tuw$, $\{tw, uw\}$ is homologous to $tu$. $tw$ and $uw$ are both positive, and hence are candidates. But $uw$ is already paired. Hence we pair $tw$ with $stu$.

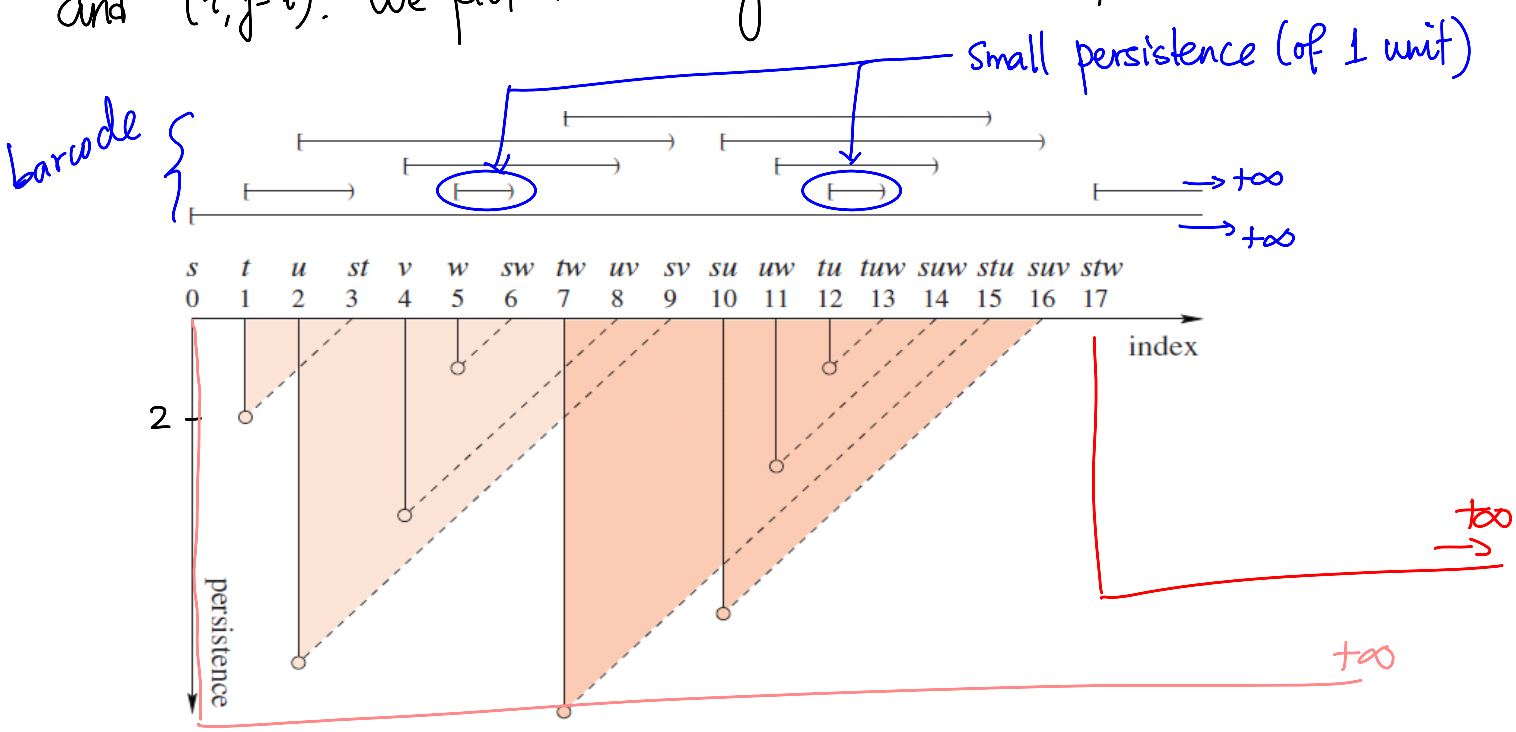
Finally, $\Gamma$ for $suv$ has just $su$, since $sv$ and $uv$ are both negative. Since $su$ is still unpaired, we pair $su$ with $suv$.

Notice that the 2-simplex $stw$ is positive, and is left unpaired as there are no (negative) 3-simplices. Similarly, the first 0-simplex $s$, which is positive, is also left unpaired, representing the single connected component that is the final simplicial complex.

We visualize the pairings by converting the intervals to triangles that are open on one side.

## Index-Persistence Diagram

For a pair $(\sigma^i, \sigma^j)$, consider the triangle with vertices $(i,0)$, $(j,0)$ and $(i, j-i)$. We plot the triangles on the index-persistence axes.



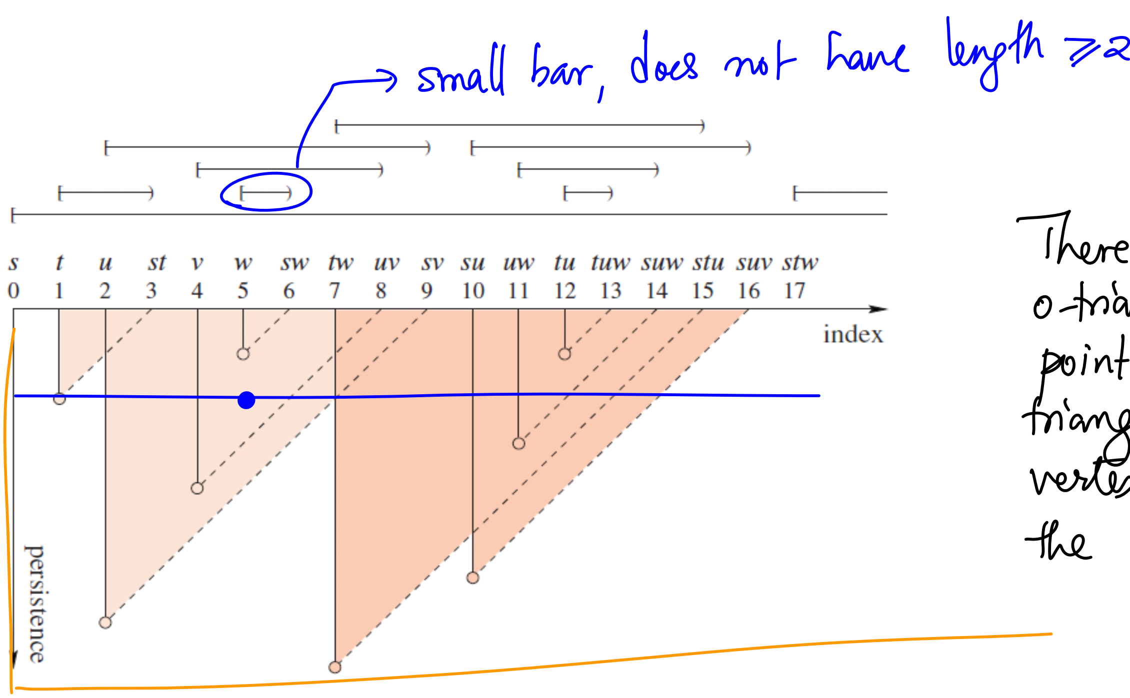

The positive 0-simplex $s$ and positive 2-simplex $stw$ are not paired, and hence have infinite persistence. These two infinite triangles are not shown.

Here, the lighter shade triangles correspond to (vertex, edge) pairs and the darker shade ones correspond to (edge, triangle) pairs.

We can compute the persistent Betti numbers by simply counting the numbers of triangles, as described below.

**Theorem** $\beta_k^{\ell,p}$ = the number of $k$-triangles containing the point $(\ell,p)$ in the index-persistence plane.

For example, consider $(\ell,p)=(5,2)$, the point marked on the index-persistence diagram. There are 3 0-triangles containing the point $(5,2)$ and no 1-triangles. Thus, there are 3 connected components that have persistence at least 2 in $K^5$. Notice that the subcomplex $K^5$ has 4 connected components (refer to the last box in Row 1 of the filtration figure). But one of these 4 components — represented by vertex $w$, merges with the bigger component represented by vertex $s$ in the next step, and hence is not 2-persistent.




There are indeed 3 0-triangles containing the point $(5,2)$. The infinite triangle corresponding to vertex $s$ (at index 0) is the non-obvious one!

# Implementation of Pairing

How do we find the youngest $k$-simplex in $\Gamma(\bar{d})$?

Store the index of pairings in a linear array $T[0, \ldots, m-1]$. The pair $(\sigma^i, \sigma^j)$ is stored by setting $T[i] = j$. We also store $\Lambda^i$ = list of positive simplices representing the cycle created by $\sigma^i$ and destroyed by $\sigma^j$. These simplices in $\Lambda^i$ are not necessarily only the ones in $\bar{d} = \partial_{k+1}(\sigma^j)$, but could include simplices in chains homologous to these simplices (in $\bar{d}$), and contains the youngest positive simplex.

For example, consider the negative edge $sv$ coming in at step 9. $\partial_1(\overline{sv}) = \{s, v\}$. Here, $v$ is younger than $s$, but it has already been paired (with edge $uv$ in the previous step). But $u \sim v$, because of edge $uv$ being already present. Hence $\Gamma(\overline{sv}) = \{s, u\}$, and $u$ is indeed unpaired (and younger than $s$). Hence we pair $u$ with $sv$.

We now describe the function to find the youngest positive $k$-simplex for pairing with a negative $(k+1)$-simplex.

integer YOUNGEST $(\sigma_j)$

$\Lambda = \{\sigma \in \partial_{k+1}(\sigma_j) \mid \sigma \text{ is positive}\};$

while (true)
{
  $i = \max(\Lambda);$ largest index
  if $T[i]$ is empty
  {
    $T[i] = j;$ "store" $\Lambda$ also in $T[i]$ → associate this $\Lambda$ with $T[i]$
    break;
  }
  else
    $\Lambda = \Lambda +_2 \Lambda[i];$ → the list associated with $T[i]$
}
return $i$;

## Illustration on the Example filtration

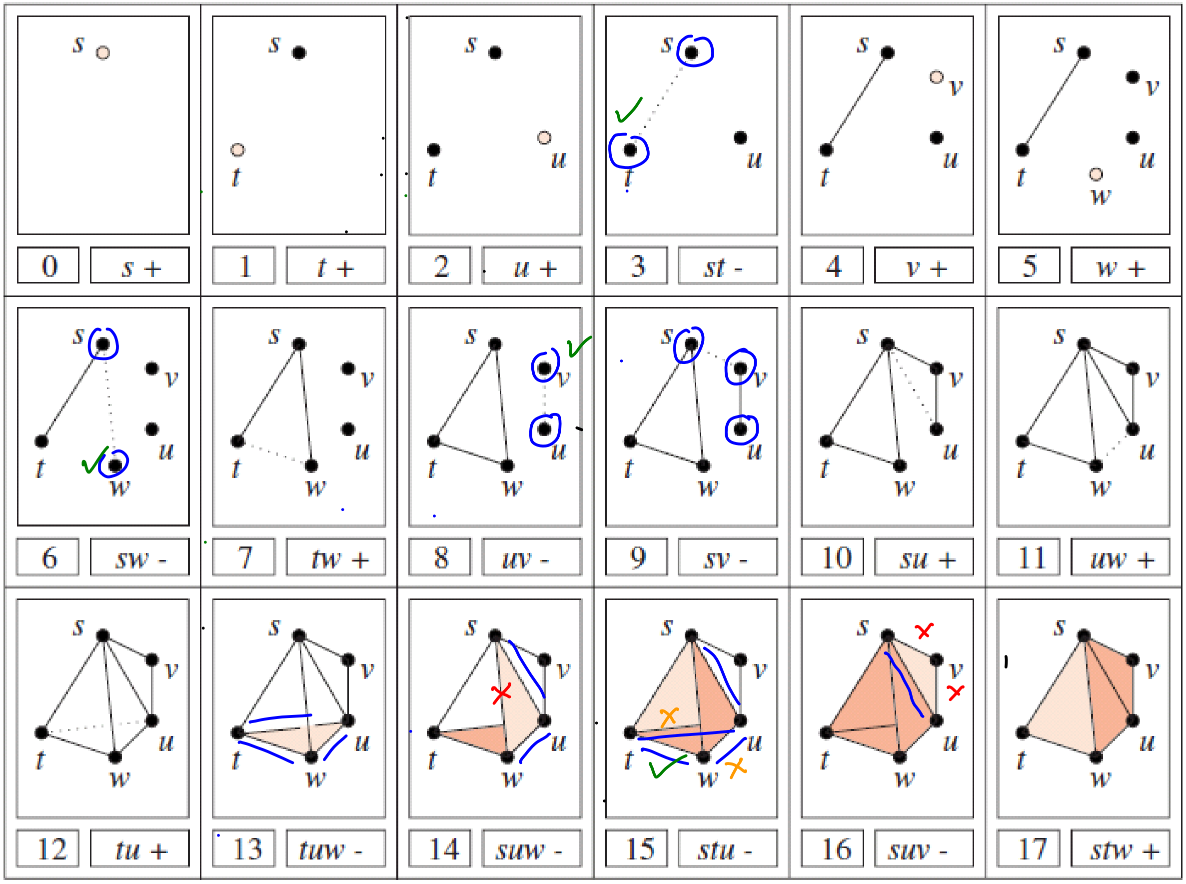
Negative simplices are indicated in blue

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| simplex | s | t | u | $\overline{st}$ | v | w | $\overline{sw}$ | tw | $\overline{uv}$ | $\overline{sv}$ | su | uw | tu | $\overline{tuw}$ | $\overline{suw}$ | $\overline{stu}$ | $\overline{suv}$ | stw |
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| T | | 3 | | | 8 | 6 | | | | | | | | | | | | |
| list | | t, s | | | v, u | w, s | | | | | | | | | | | | |

latest addition shown in blue (to T[4]).

For st, sw, and uv, the function does not go into the "else" part of if-statement (indices 3, 6, 8, respectively).

Here are the filtration and pairings again:




pairings

(t, st)

(w, sw)

(v, uv)

(u, sv)

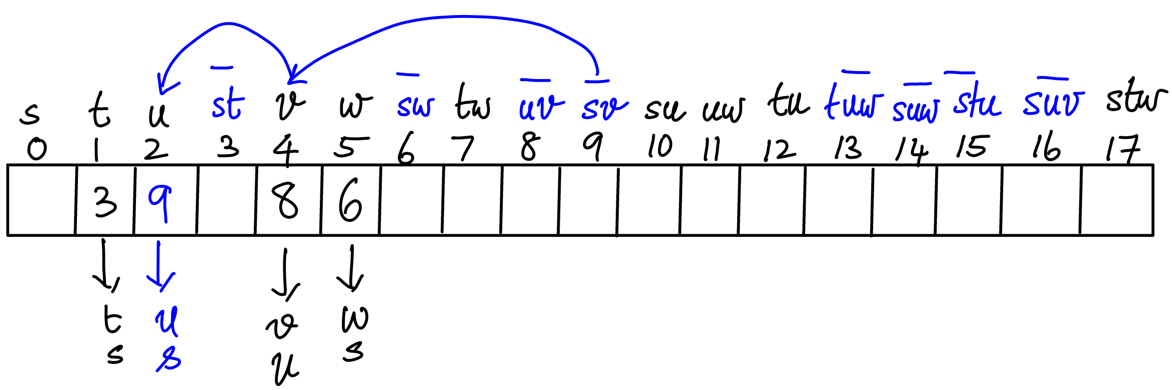
(tu, tuw)

(uw, suw)

(tw, stu)

(su, suv)

For sv at index 9: $\Lambda = \{s, v\}$ v is younger, at 4. But T[4] is not empty, and has $\Lambda[4] = \{v, u\}$. So, we set $\Lambda = \Lambda +_2 \Lambda[4] = \{s, \cancel{v}\} +_2 \{\cancel{v}, u\} = \{s, u\}$.




A step where we update $\Lambda$ by taking sum modulo 2 as above is termed a **collision**.

We keep proceeding with the function till step/index 15, when $\Delta stu$ comes in. The array T has the following form before Step 15.

| | $s$ | $t$ | $u$ | $\overline{st}$ | $v$ | $w$ | $\overline{sw}$ | $tw$ | $\overline{uv}$ | $\overline{sv}$ | $su$ | $uw$ | $tu$ | $\overline{tuw}$ | $\overline{suw}$ | $\overline{stu}$ | $\overline{suv}$ | $stw$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| T | | 3 | 9 | | 8 | 6 | | 15 | | | | 14 | 13 | | | | | |
| | | $t$, $s$ | $u$, $s$ | | $v$, $u$ | $w$, $s$ | | $tw$ | | | | $uw$, $su$ | $tu$, $uw$, $tw$ | | | | | |

$stu$ (at Step 15): $\Lambda = \{su, tu\}$.

$tu$ is positive, and is youngest, at 12, but $T[12]$ is not empty. $tu$ is not included in $\Lambda$ here.

Notice that $st \in \partial_2(stu)$ is negative, and hence not included in $\Lambda$ here.

$$\Lambda + \Lambda[12] = \{su, \cancel{tu}\} +_2 \{\cancel{tu}, uw, tw\} = \{su, uw, tw\}$$

Now, $uw$ is youngest at 11. But, $T[11]$ is occupied, so we take another sum mod 2 (i.e., there is a second collision here).

$$\Lambda + \Lambda[11] = \{\cancel{su}, \cancel{uw}, tw\} +_2 \{\cancel{uw}, \cancel{su}\} = \{tw\}.$$

| | $s$ | $t$ | $u$ | $\overline{st}$ | $v$ | $w$ | $\overline{sw}$ | $tw$ | $\overline{uv}$ | $\overline{sv}$ | $su$ | $uw$ | $tu$ | $\overline{tuw}$ | $\overline{suw}$ | $\overline{stu}$ | $\overline{suv}$ | $stw$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| T | $+\infty$ | 3 | 9 | | 8 | 6 | | 15 | | | 16 | 14 | 13 | | | | | $+\infty$ |
| | | $t$, $s$ | $u$, $s$ | | $v$, $u$ | $w$, $s$ | | $tw$ | | | $su$ | $uw$, $su$ | $tu$, $uw$, $tw$ | | | | | |

We then pair $suv$ (step 16) with edge $su$ – there are no collisions in this case.

Finally, slots corresponding to the two remaining positive simplices $s$ and $stw$ are filled with $+\infty$ (some large number in practice).

The final filled in array $T$ is given below: