Applicability of Loop Recombination in Ciliates using the Breakpoint Graph

Robert Brijder, Hendrik Jan Hoogeboom, and Michael Muskulus

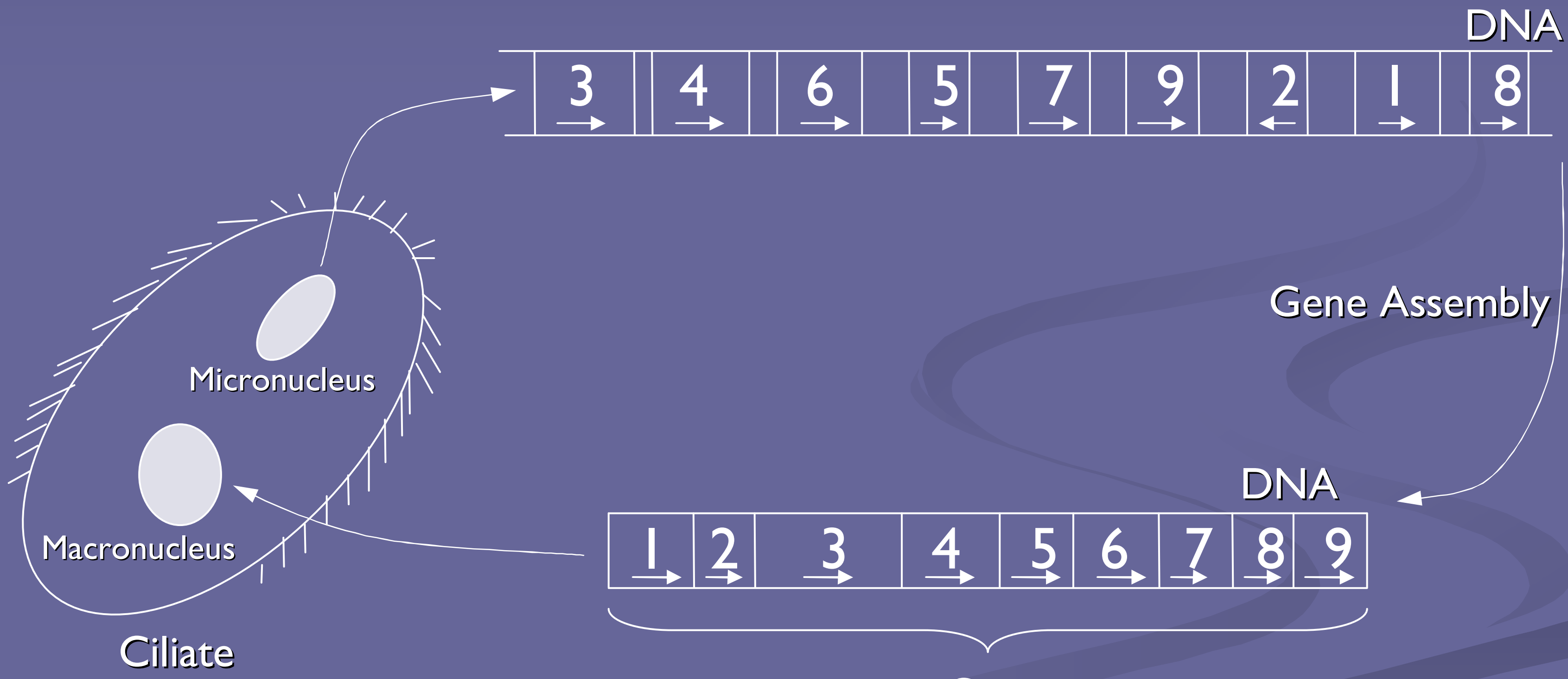
Leiden Institute of Advanced Computer Science, Leiden University

Overview

- Brief overview of gene assembly in ciliates.
- Brief overview of a formal model.
- Recall reduction graph.
- Introduce graph `on top of' reduction graph, and show its uses.

Gene Assembly in Ciliates

A gene of the Sterkiella nova



Gene Assembly

- The gene assembly process is done using molecular operations.
- With the aid of *pointers* these operations `know' how these parts need to be glued together.

Actually

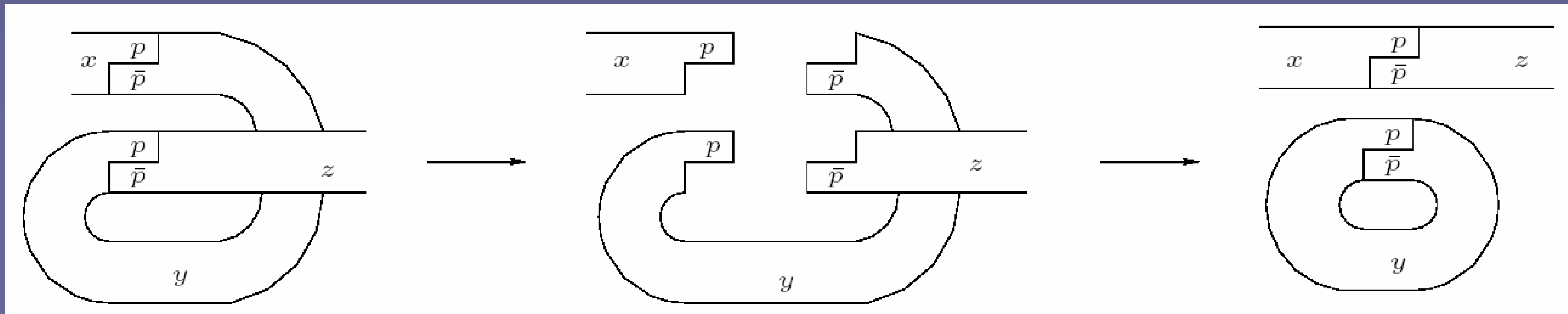
Gene Assembly

The gene assembly process is accomplished using three molecular operations:

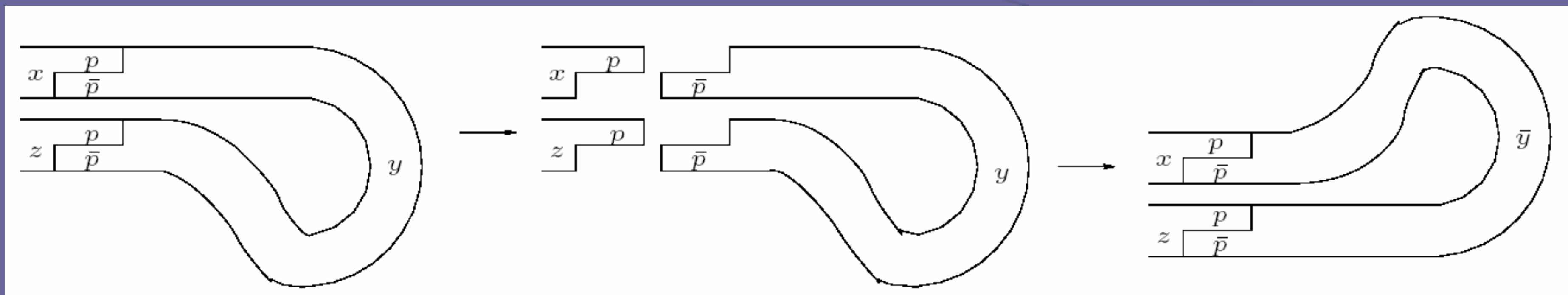
- Loop recombination
- Hairpin recombination
- Double-loop recombination

Gene Assembly

- Loop recombination

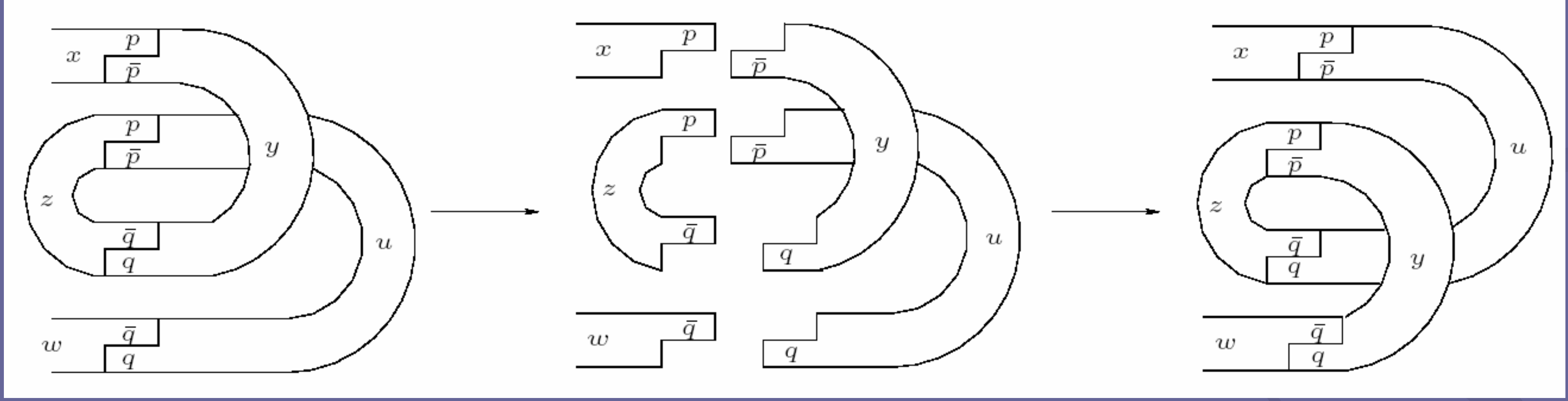


- Hairpin recombination



Gene Assembly

- Double-loop recombination



Gene Assembly

The process is irreversible:

when a molecular operation is applied on a pointer, then this pointer cannot be used again.

Overview

- Brief overview of gene assembly in ciliates.
- ***Brief overview of a formal model.***
- Recall reduction graph.
- Introduce graph `on top of' reduction graph, and show its uses.

Modeling Gene Assembly

SPRS

$342\bar{3}\bar{2}4$ legal string

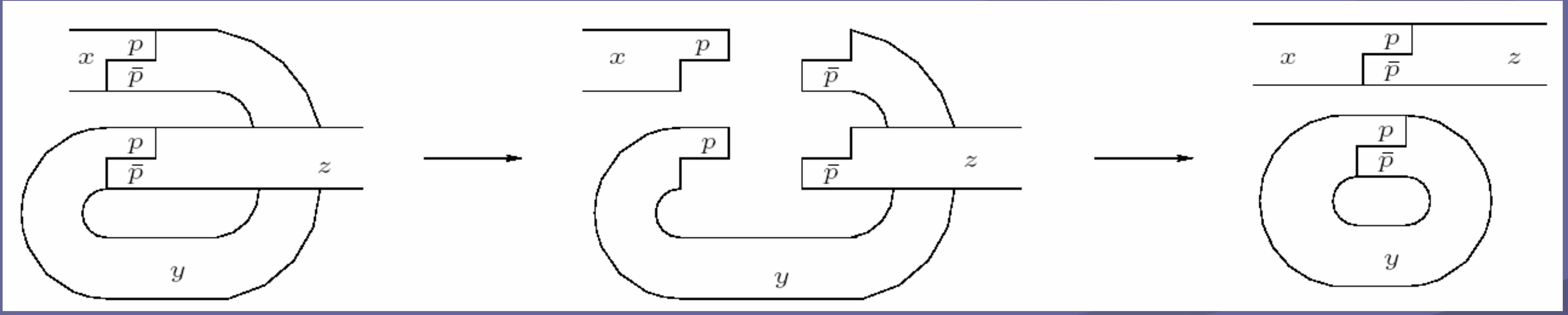
$\Pi = \{2, \bar{2}, 3, \bar{3}, \ldots\}.$

Modeling Gene Assembly

Loop recombination - String negative rule

For $p \in \Pi$ and $x, z \in \Pi^*$:

$$snr_p(xppz) = xz$$



$$\text{e.g.,}\ snr_{\bar{3}}(45\bar{3}\bar{3}45) = 4545.$$

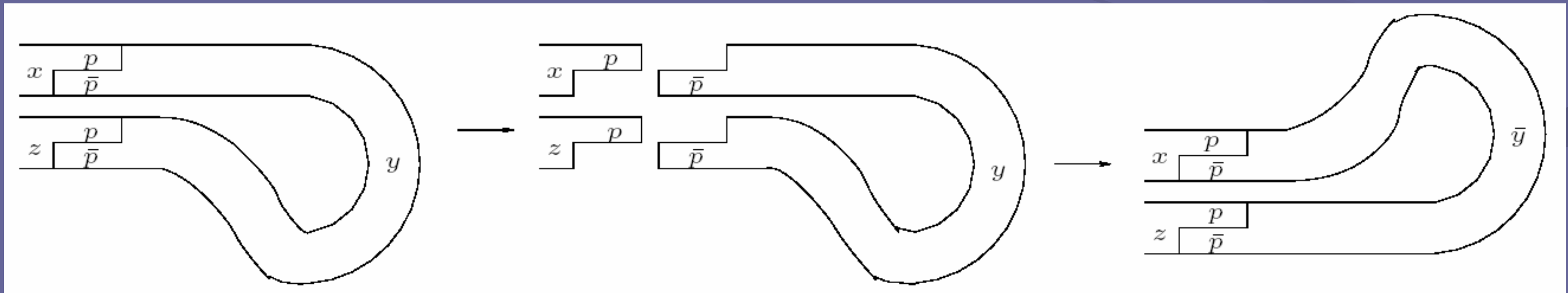
Modeling Gene Assembly

Hairpin recombination - String positive rule

$$spr_p(xpy\overline{p}z) = x\overline{y}z$$

where $\overline{\overline{p}} = p$

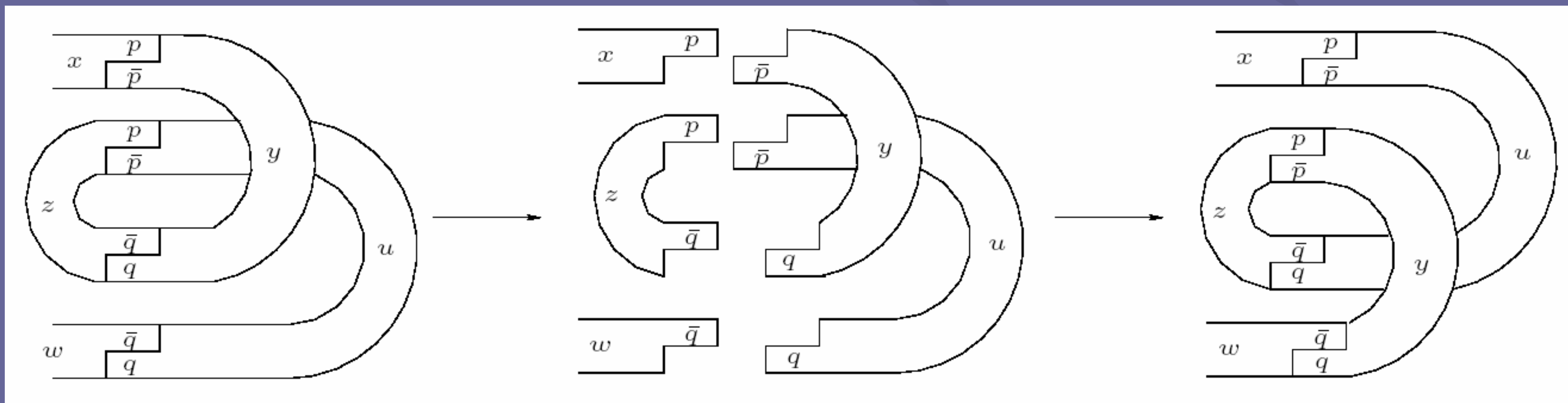
$$u = x_1 x_2 \cdots x_n \Rightarrow \overline{u} = \overline{x}_n \overline{x}_{n-1} \cdots \overline{x}_1$$



Modeling Gene Assembly

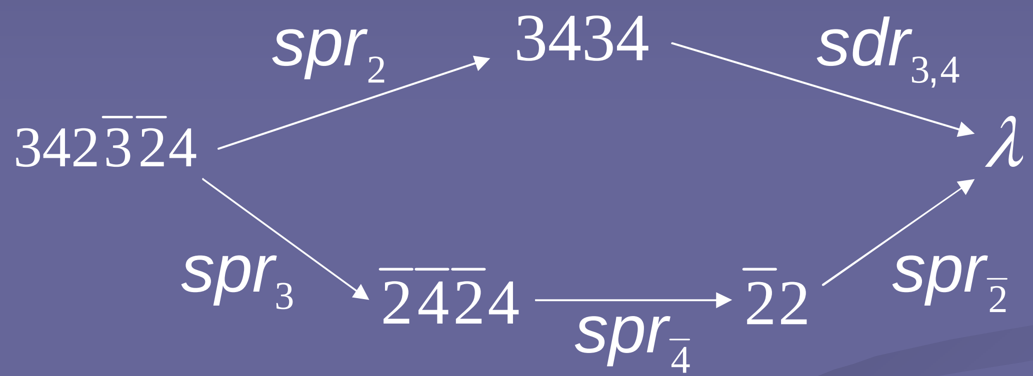
Double-loop recombination - String double rule

$$sdr_{p,q}(xpyqzpuqw) = xuzyw$$

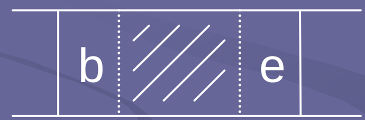


Modeling Gene Assembly

- Successful reductions:



- λ corresponds to a successful assembled gene.
- Non-deterministic process.

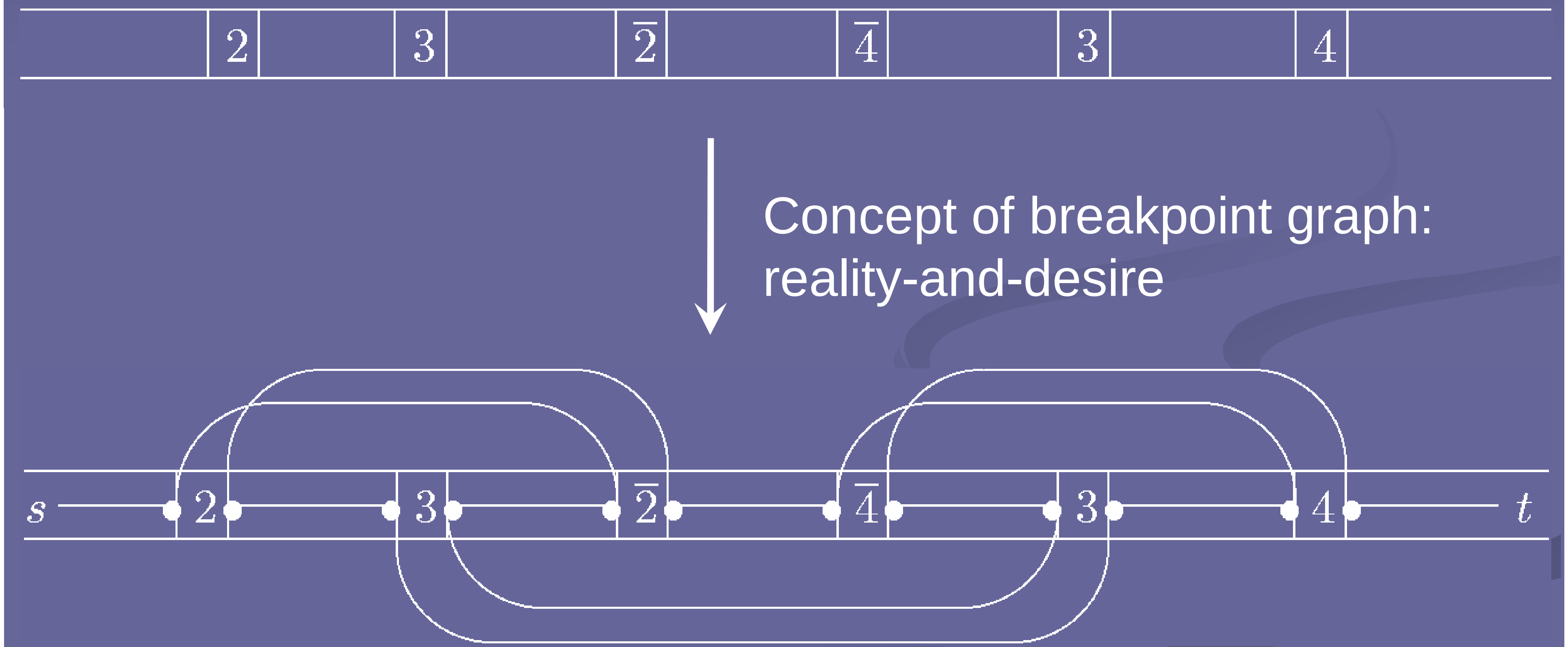


- No deadlocks.
- Finite process. Pointers are removed in every step.

Overview

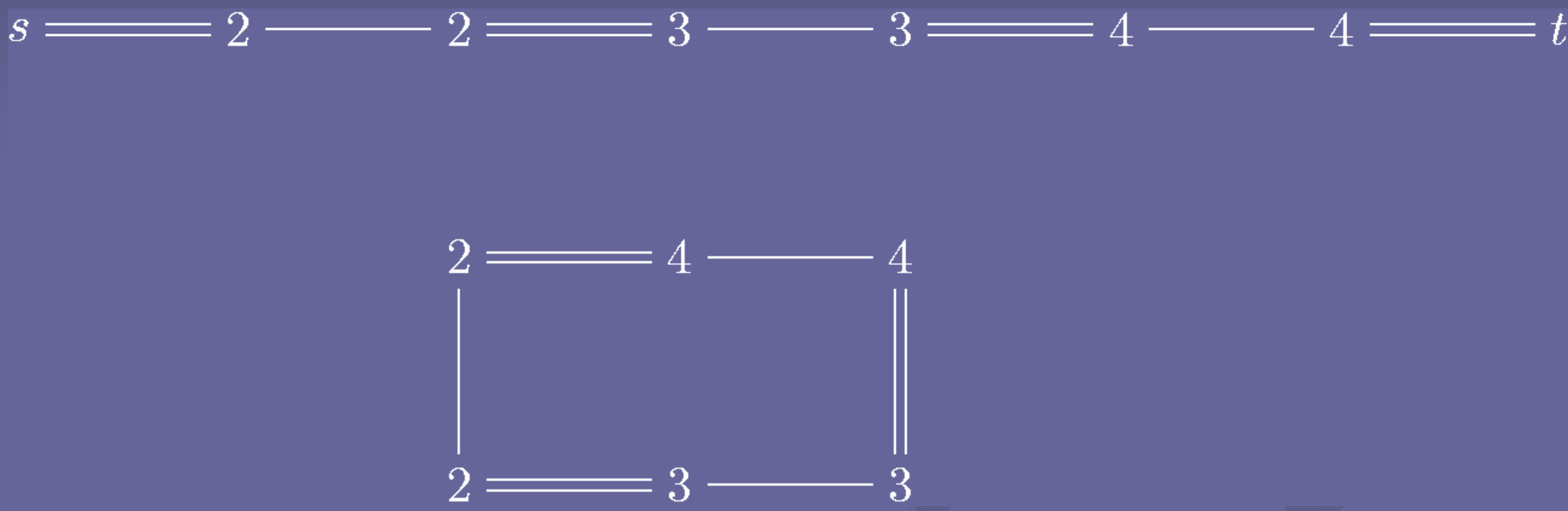
- Brief overview of gene assembly in ciliates.
- Brief overview of a formal model.
- ***Recall reduction graph.***
- Introduce graph `on top of' reduction graph, and show its uses.

Reduction Graph



Concept of breakpoint graph:
reality-and-desire

Reduction Graph Example



- Double edges are the reality edges, single edges the desire edges
- There is a cyclic component.
- This is the less general version of reduction graph.

A known result

Let *u* be a legal string, and let *N* be the number of cyclic components in *R(u)*. Then every successful reduction of *u* has exactly *N* string negative rules.

Complexity: $O(|u|)$

Example

Let $u = 23\overline{2}\overline{4}34$ be a legal string. Then *R(u)* has one cyclic component. Thus, every successful reduction of *u* has exactly one string negative rule.

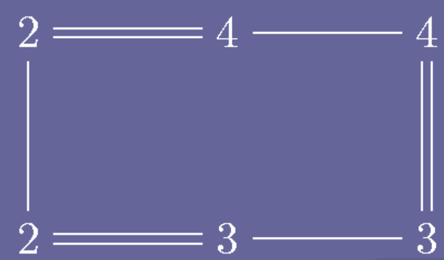
Successful reductions :

$snr_2\ spr_3\ spr_4$,

$snr_3\ spr_2\ spr_4$,

$snr_3\ spr_4\ spr_2$,

$snr_4\ spr_3\ spr_2$.



Overview

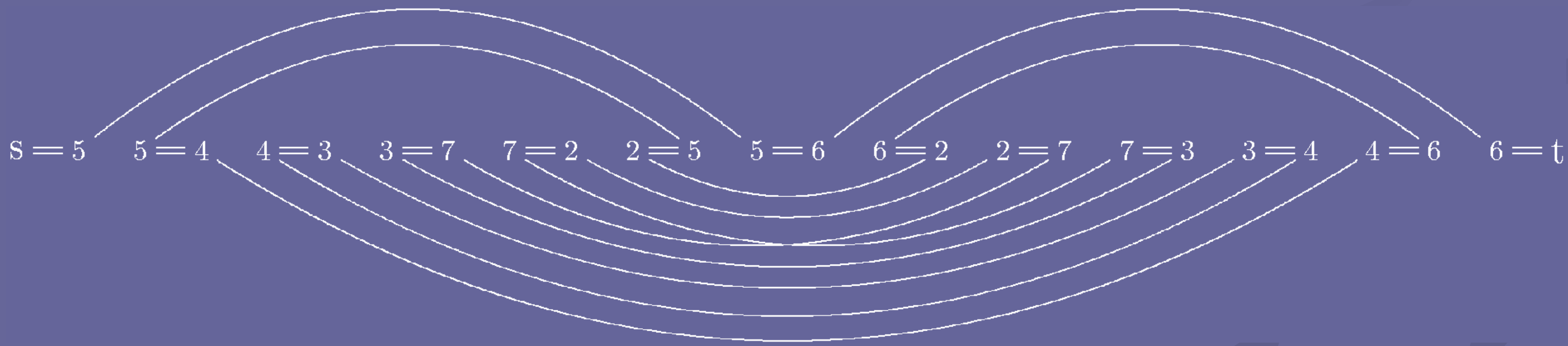
- Brief overview of gene assembly in ciliates.
- Brief overview of a formal model.
- Recall reduction graph.
- ***Introduce graph `on top of' reduction graph, and show its uses.***

Motivation

- Now we know *how many* Snr rules are needed, can we also determine *on which pointers* these Snr rules are applied in successful reductions?
- Previous example: Snr domains are {2}, {3}, {4}.
- If so, is this `information' retained in the reduction graph?

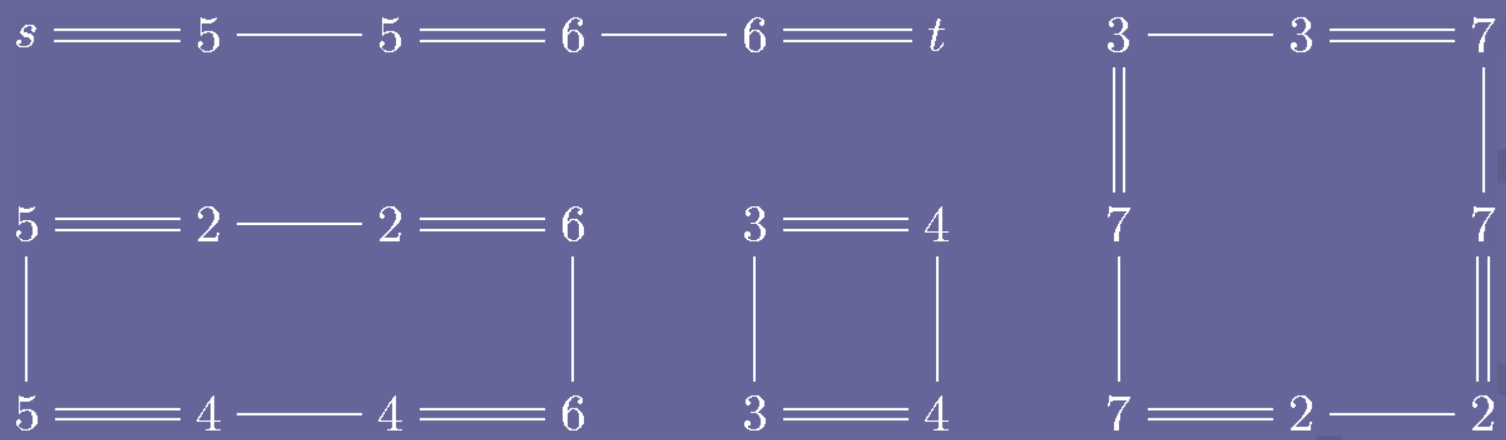
More complicated Example

$u = 54372562\bar{7}346$

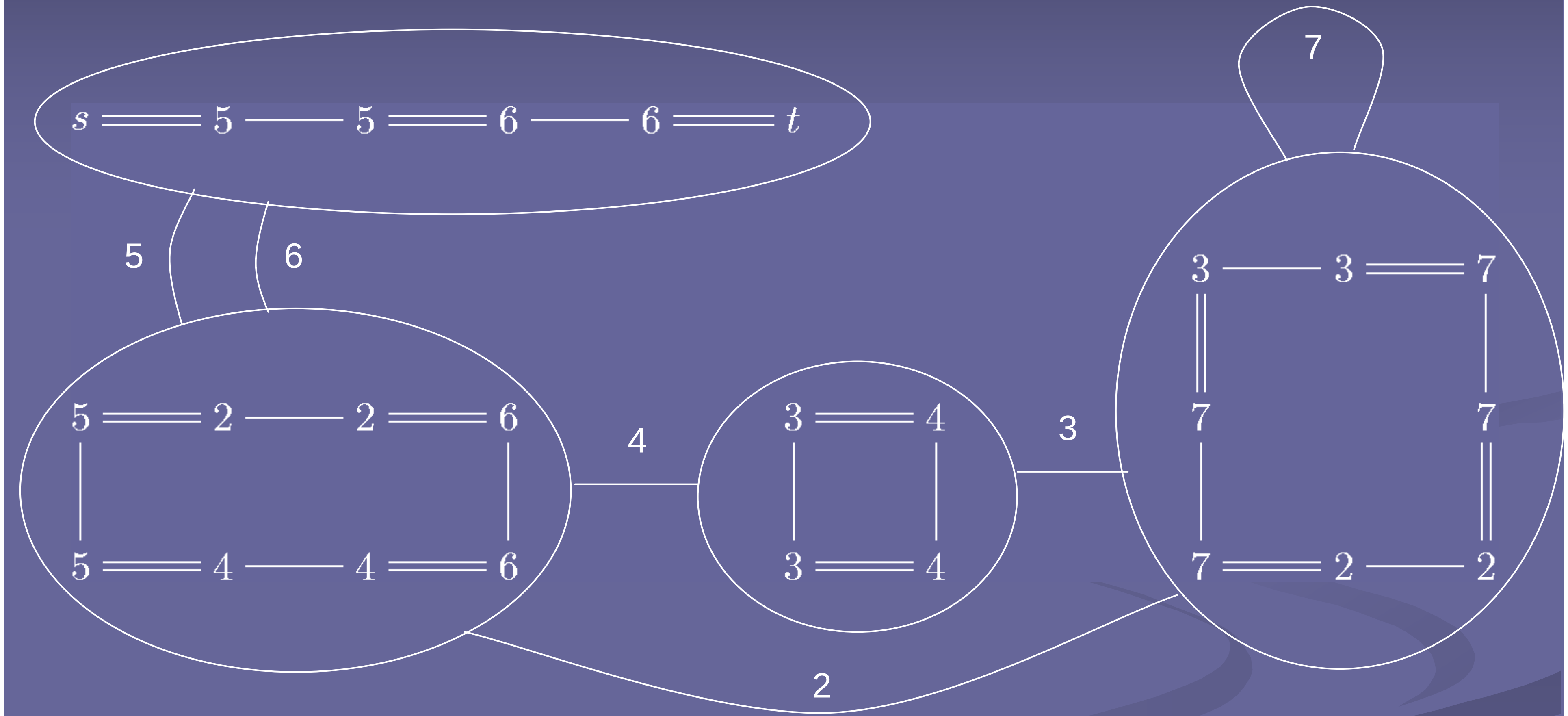


More complicated Example

- Which is:



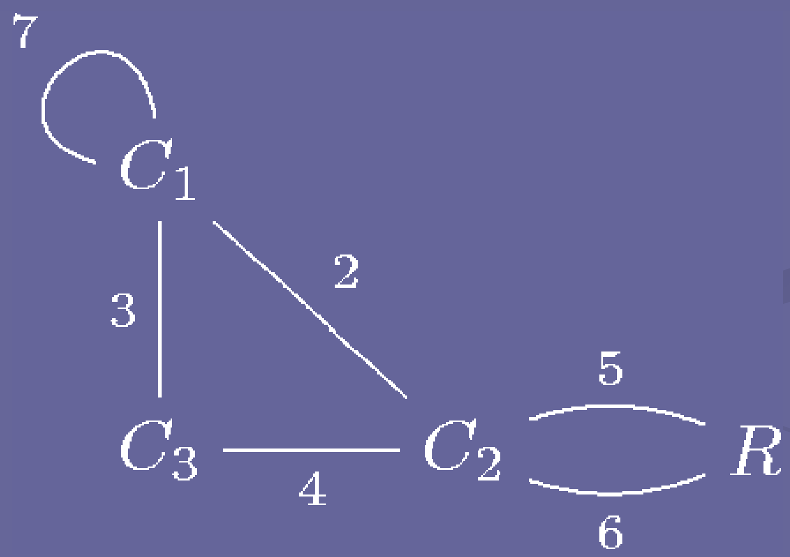
Pointer Component Graph



- A multigraph. Notice that loops and parallel edges are allowed.

Pointer Component Graph

- Which is:



- *R* is the linear connected component, and the others are cyclic components.

Result

For each legal string *u*,

there is a successful reduction of *u* which applies snr on the set of pointers *D*

iff

D induces a spanning tree in the PC graph of *u*.

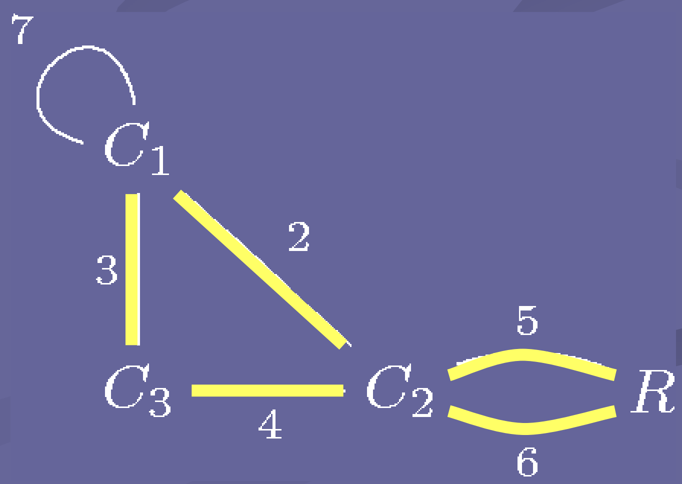
In the example : e.g.

{2,3,5}, {2,4,6},

but not

{2,5,6} or {2,3}.

Also, 7 will never occur in such a set.



Example

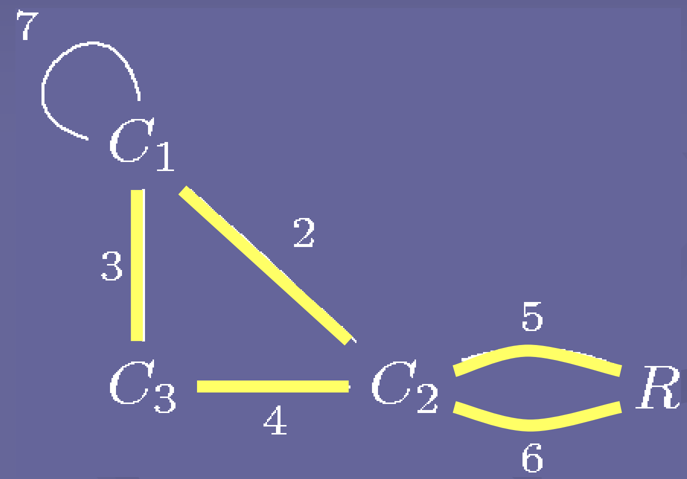
$u = 54372562\bar{7}346.$

$D = \{2,3,5\}$, then, e.g,

$\varphi = snr_5 snr_2 snr_{\bar{3}} spr_{\bar{7}} sdr_{4,6}$ is a successful reduction of u.

$D = \{2,4,6\}$, then, e.g.,

$\varphi = snr_6 snr_2 spr_{\bar{7}} snr_4 sdr_{5,3}$ is a successful reduction of u.



However, if we take $D = \{2,3,4\}$, then we always get stuck.
E.g., $(spr_5 spr_7)(u) = 62\bar{3}\bar{4}\bar{2}346.$ No operation can be applied on pointer 6 and no snr rule is applicable.

Next question

We know:

- *How many* Snr rules are needed in successful reductions.
- *On which pointers* Snr rules are applied in successful reductions.

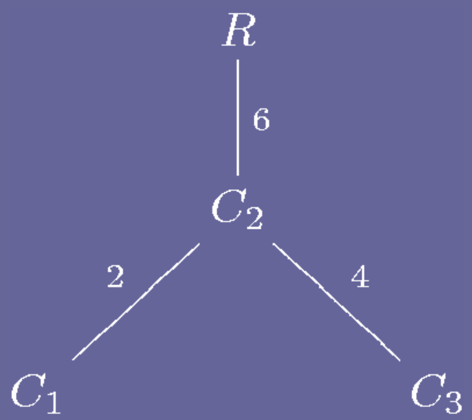
Next:

- What is *the order* of the Snr rules in successful reductions?

Extended result

If D induces a spanning tree T in the PC - graph, then the order in which snr rules are applied is determined by T with root R (the linear component).

Spanning tree in example :



Linear orderings $\longrightarrow$

$(2,4,6) \rightarrow$ e.g. $\varphi = snr_6 snr_4 snr_2 spr_3 spr_5 spr_7$

$(4,2,6) \rightarrow$ e.g. $\varphi = snr_6 snr_2 spr_7 snr_4 sdr_{5,3}$

Reveals (in)dependence of string negative rules

Conclusion

Concept of pointer-component graph proves useful within the theory of gene assembly.

In biological terms: it allows for a characterization of applicability of loop recombination operations for a given micronuclear gene.

These characterizations correspond to efficient algorithms (making the graph and finding spanning trees in a graph are both computationally easy).

The End

Definitions

- Domain : $dom(34\bar{3}\bar{2}) = \{2,3,4\}$.
- A composition of reduction rules φ is called a *reduction of u*, if φ is applicable to u.

  Example : $\varphi = (spr_{\bar{4}}\, spr_3)$ is a reduction of $u = 342\bar{3}\bar{2}4$, since $\varphi(u) = \bar{2}2$.

- A reduction φ of u is called *successful*, if $\varphi(u) = \lambda$.

  Example : $\varphi = (spr_{\bar{2}}\, spr_4\, spr_3)$ is a successful reduction of $u = 342\bar{3}\bar{2}4$.