Wheeler graphs: Succinct pattern matching on sequence graphs

DS202: Algorithmic Foundations of Big Data Biology

Presented by:
Jyotshna Rajput
M.Tech. (Research)
ATCG Lab, CDS

Slides reference:
https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Content

- Motivation
- Wheeler graph: Definition
- How to store a Wheeler graph?
- Matching pattern on a Wheeler graph

Motivation

- We know that FM-index enables operations like locating & matching a pattern in a given string.
- Now, our aim is to generalize this concept for more than one strings using graphs.
- For example:

***T*: g a t t a c a t $**

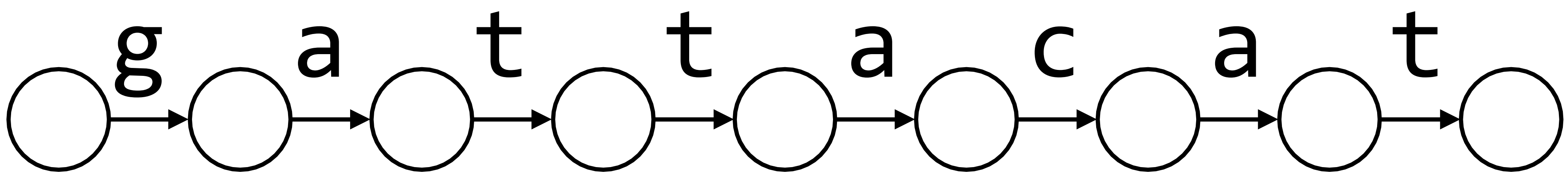
***P*: g a t**

We use BWT matching

$ g a t t a c a **t**
a c a t $ g a t **t**
a t $ g a t t a **c**
a t t a c a t $ **g**
c a t $ g a t t **a**
g a t t a c a t **$**
t $ g a t t a c **a**
t a c a t $ g a **t**
t t a c a t $ g **a**

BWT: matching

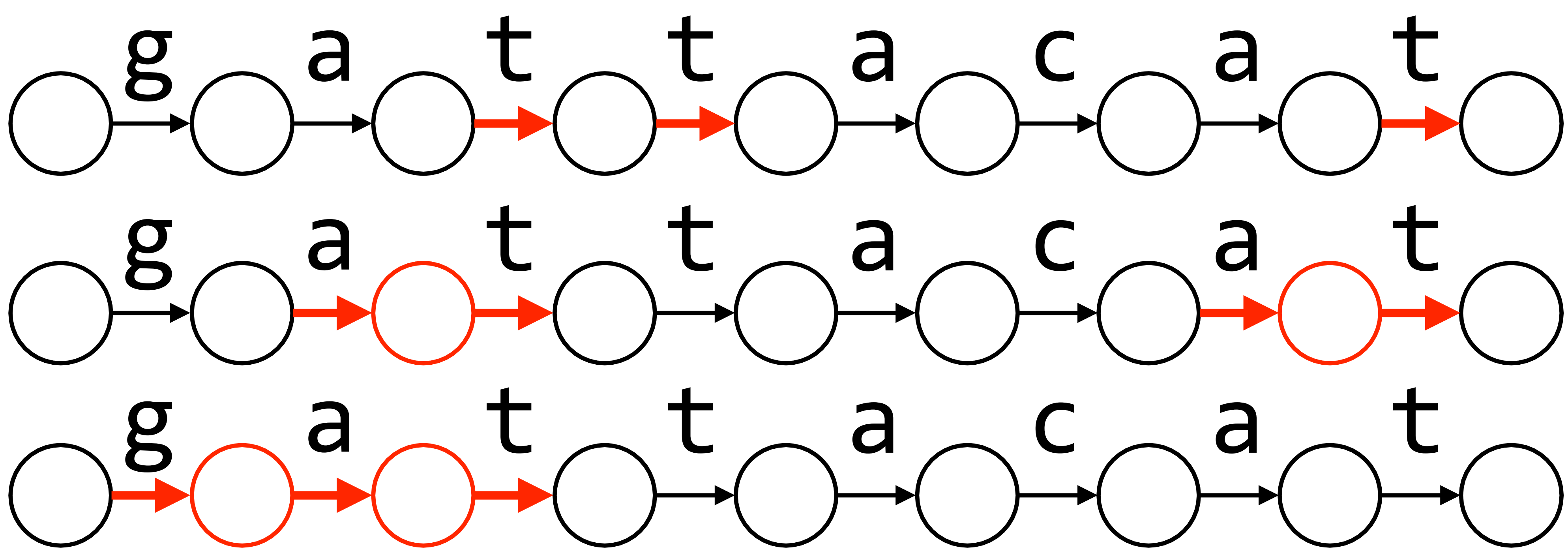
T :



P : g a t

```
$ g a t t a c a t
a c a t $ g a t t
a t $ g a t t a c
a t t a c a t $ g
c a t $ g a t t a
g a t t a c a t $
t $ g a t t a c a
t a c a t $ g a t
t t a c a t $ g a
```

BWT:matching



g a t t a c a t

g a t t a c a t

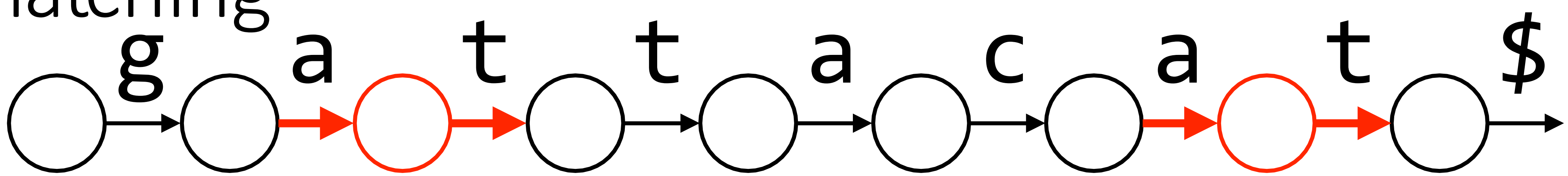
g a t t a c a t

Two interpretations:

- we're finding matching *substrings* in a *string*,
- or we're finding matching *paths* in a *graph*.

BWT:matching

T :

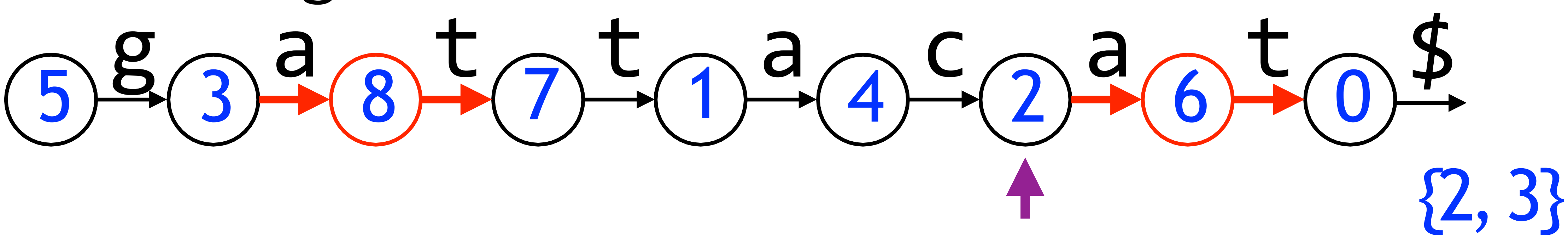


0	$	g	a	t	t	a	c	a	t
1	a	c	a	t	$	g	a	t	t
2	a	t	$	g	a	t	t	a	c
3	a	t	t	a	c	a	t	$	g
4	c	a	t	$	g	a	t	t	a
5	g	a	t	t	a	c	a	t	$
6	t	$	g	a	t	t	a	c	a
7	t	a	c	a	t	$	g	a	t
8	t	t	a	c	a	t	$	g	a

Consecutivity. In BW order, rows with same prefix are consecutive.

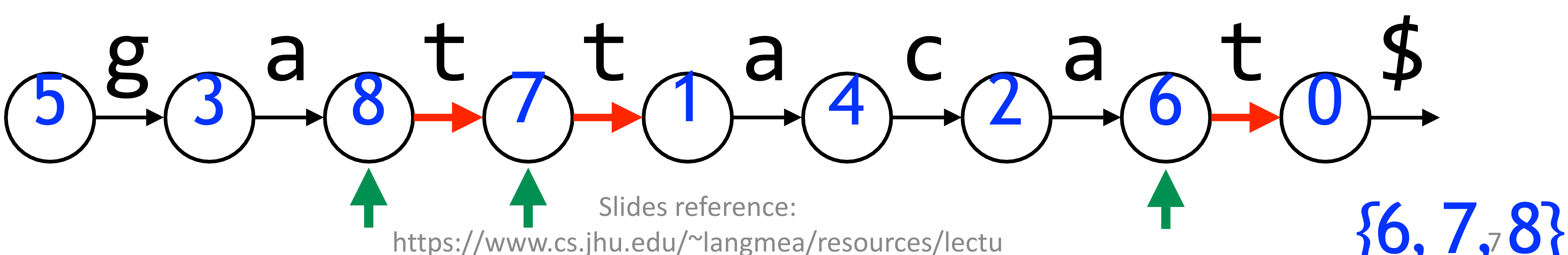
Is this visible in the graph? Let's label nodes with BW order…

BWT:matching



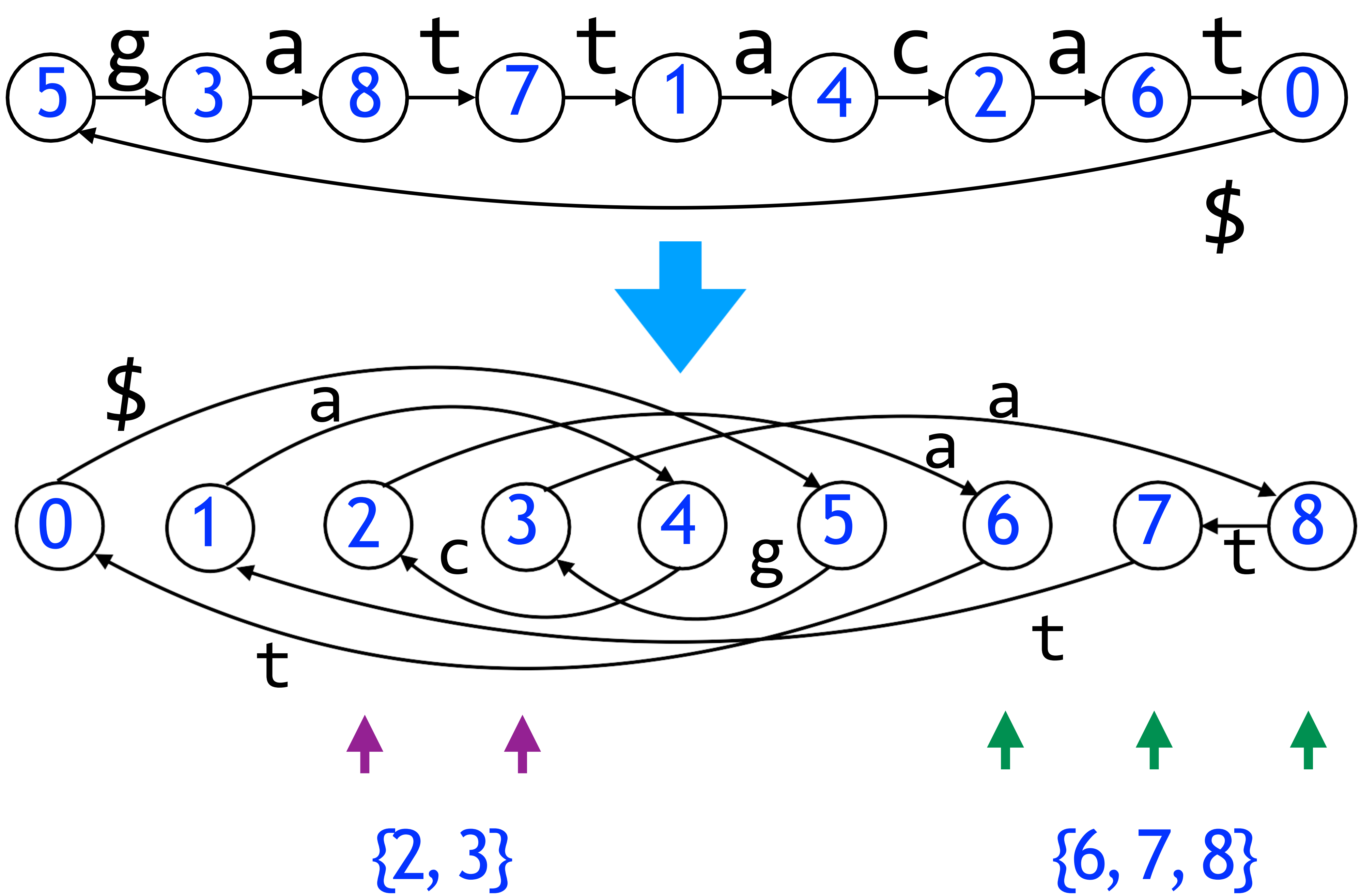
0	$ g a t t a c a t
1	a c a t $ g a t t
2	a t $ g a t t a c
3	a t t a c a t $ g
4	c a t $ g a t t a
5	g a t t a c a t $
6	t $ g a t t a c a
7	t a c a t $ g a t
8	t t a c a t $ g a

- Consecutivity holds for labels of nodes in the BW range;
- would be clearer if we redrew the graph in BWT(T) order rather than T order



Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

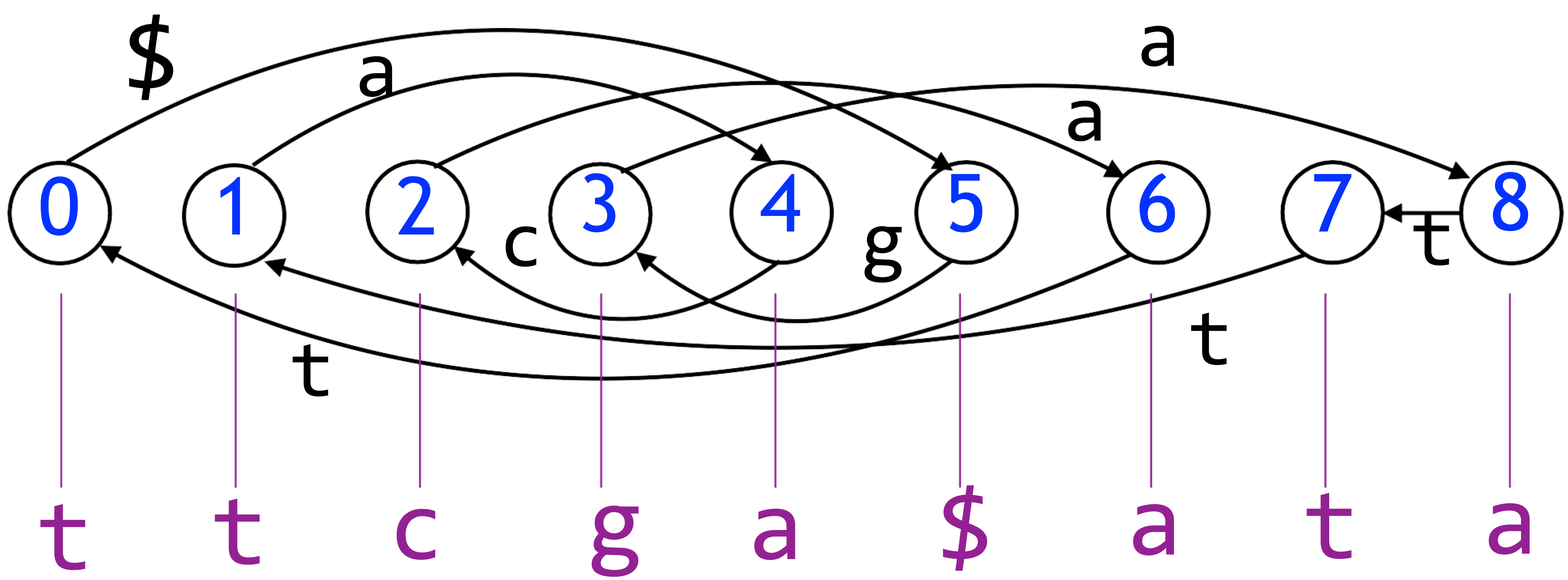
BWT:matching



Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

BWT:matching

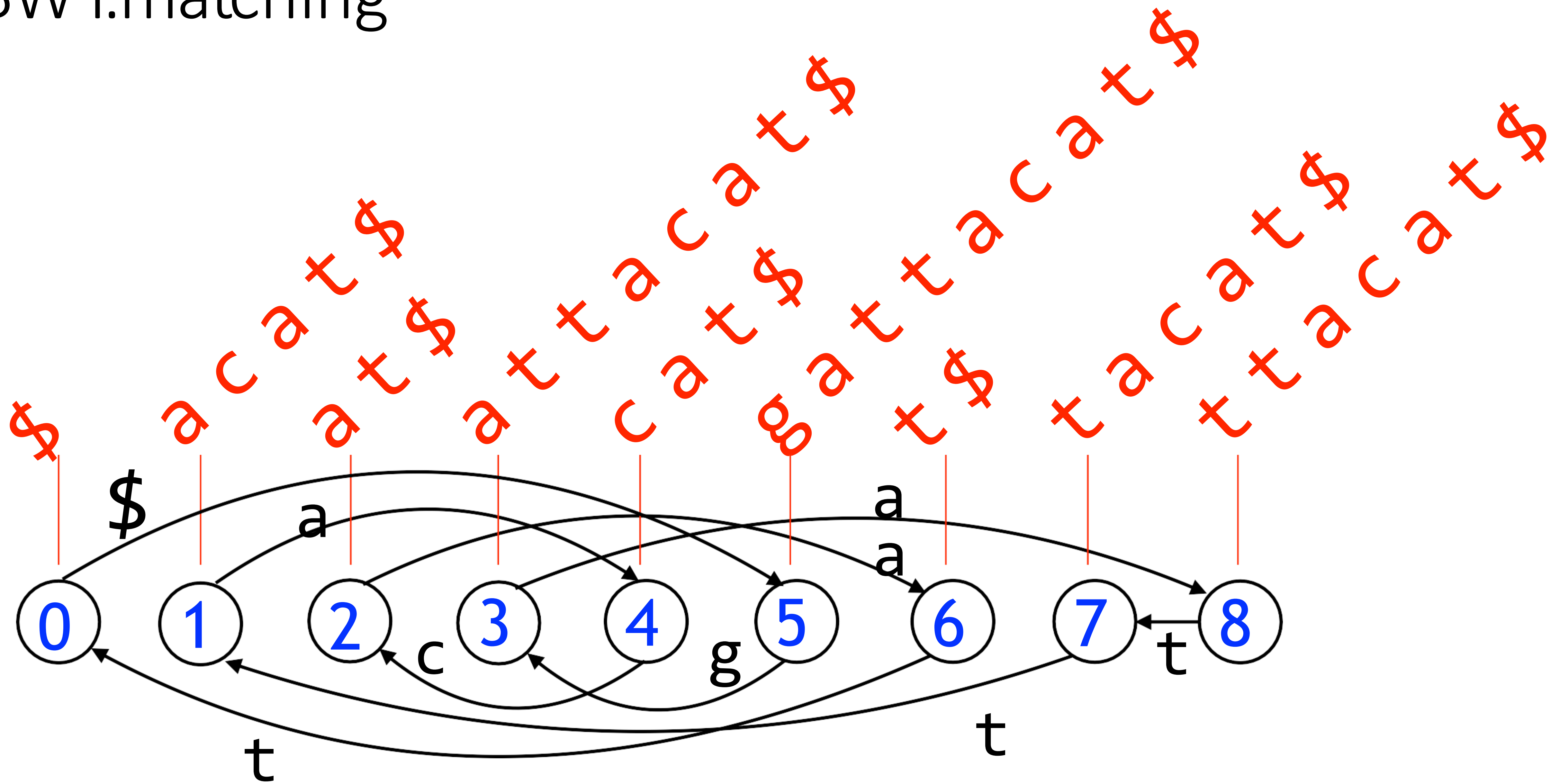
Nodes can be thought of according to what comes after (outgoing edges) and or just before (incoming)



Incoming edges spell out BWT

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

BWT:matching



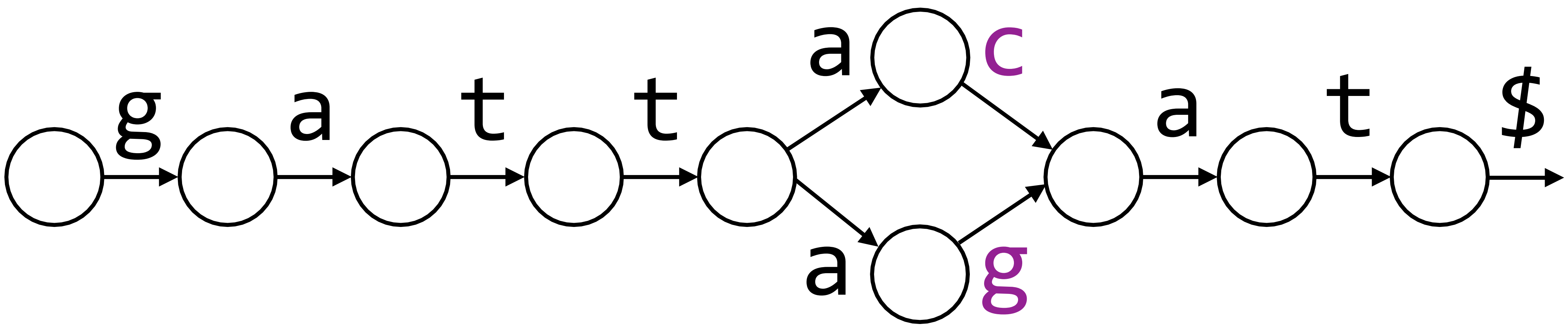
Outgoing paths spell out suffixes/rotations

BWT: matching

Can we go beyond straight-line graphs?

BWT:matching

What does this mean?



g a t t a c a t $

or

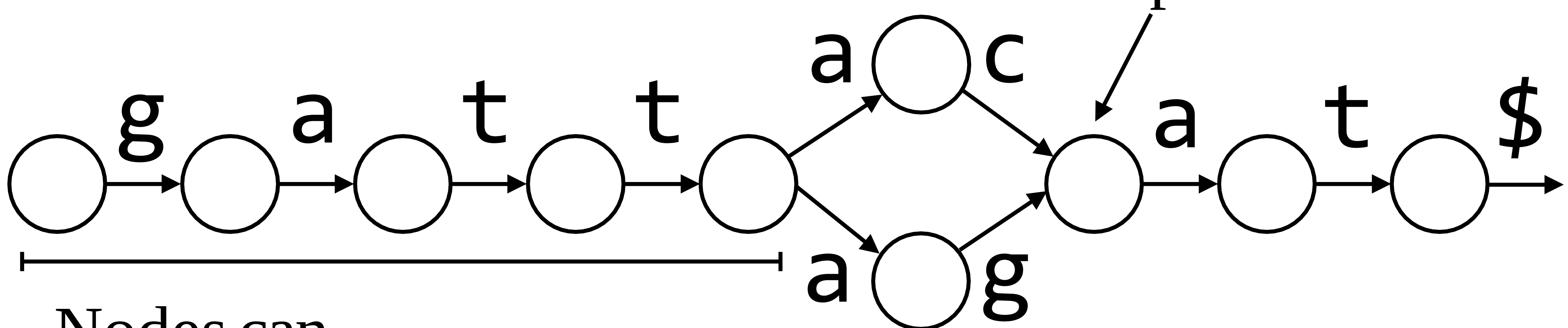
g a t t a g a t $

BWT:matching

Does our way of thinking about nodes still hold?

No:

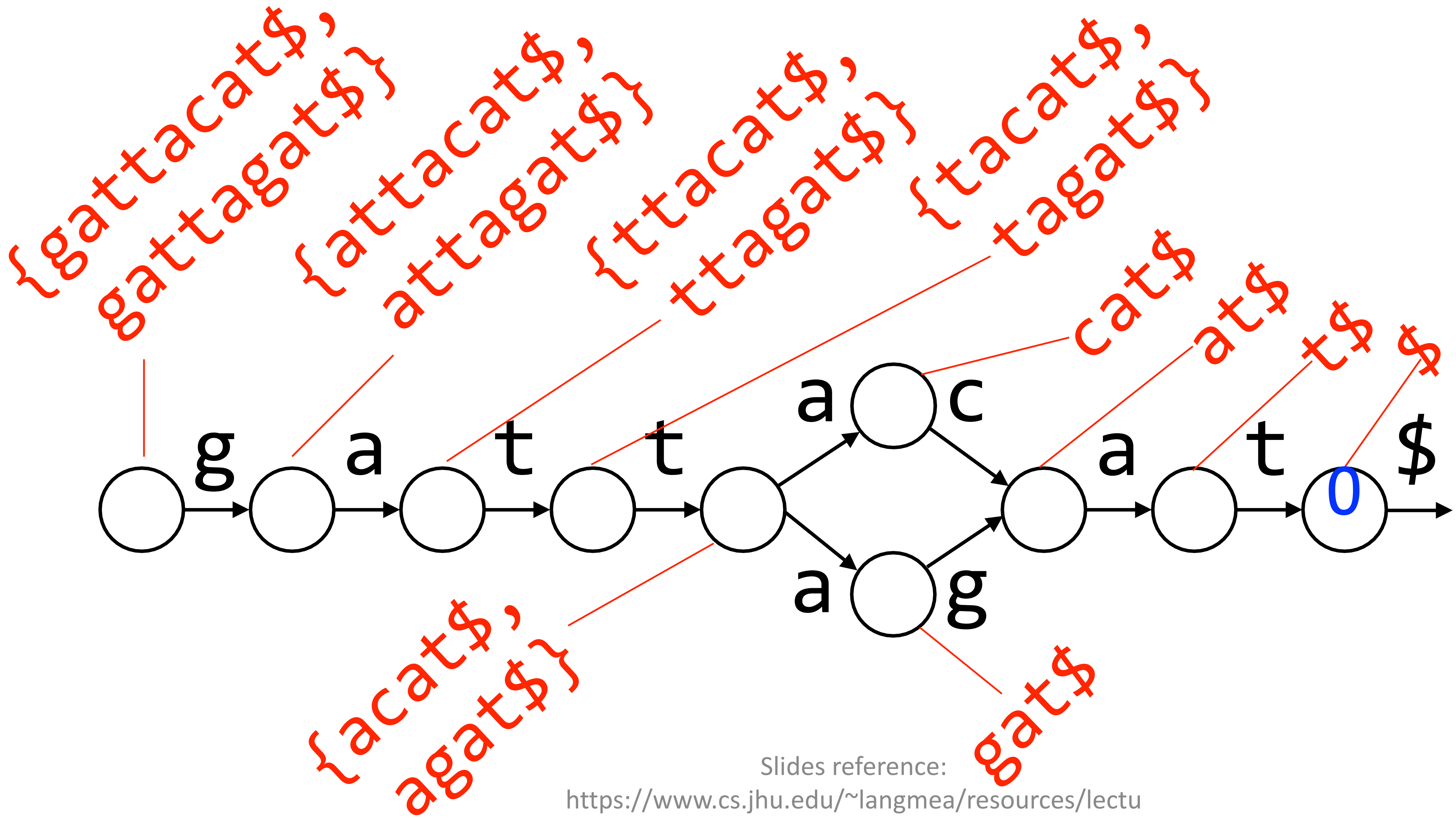
Nodes can have multiple predecessors



Nodes can have multiple suffixes leading out from them

BWT:matching

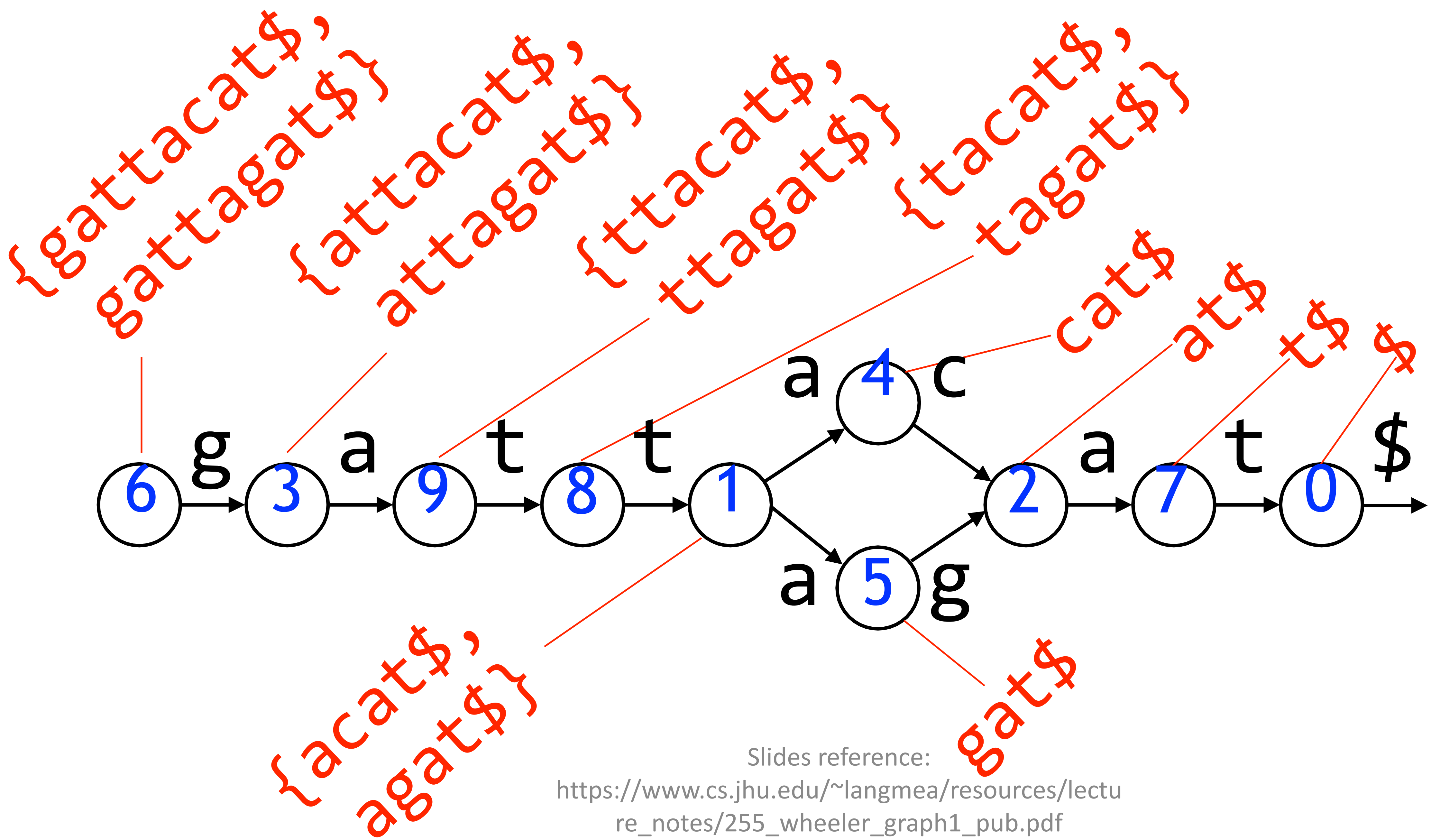
Can we preserve a total order over outgoing suffixes, even when there's >1 per node?



Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

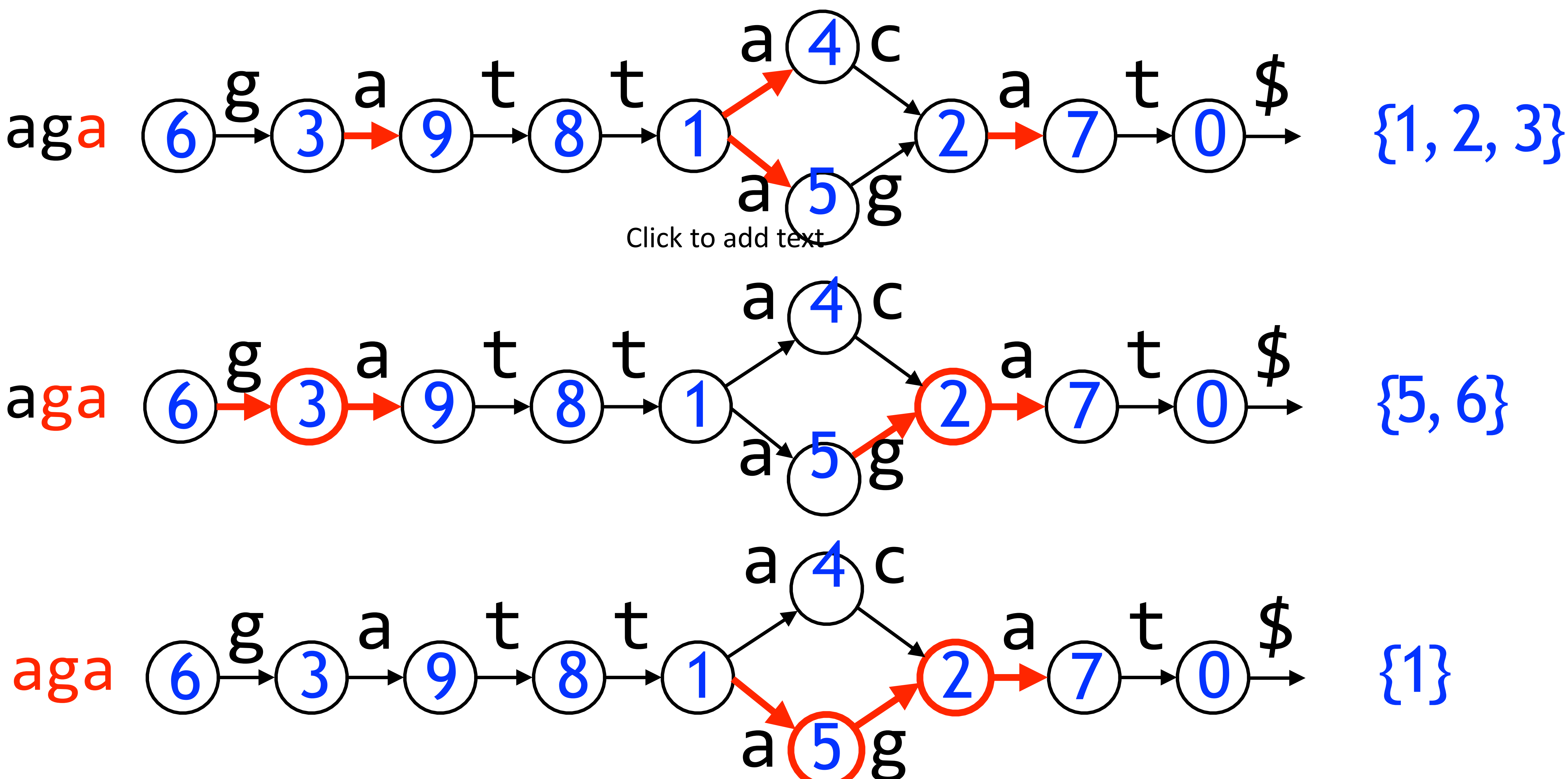
BWT: matching

Can we preserve a total order over outgoing suffixes, even when there's >1 per node?



Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

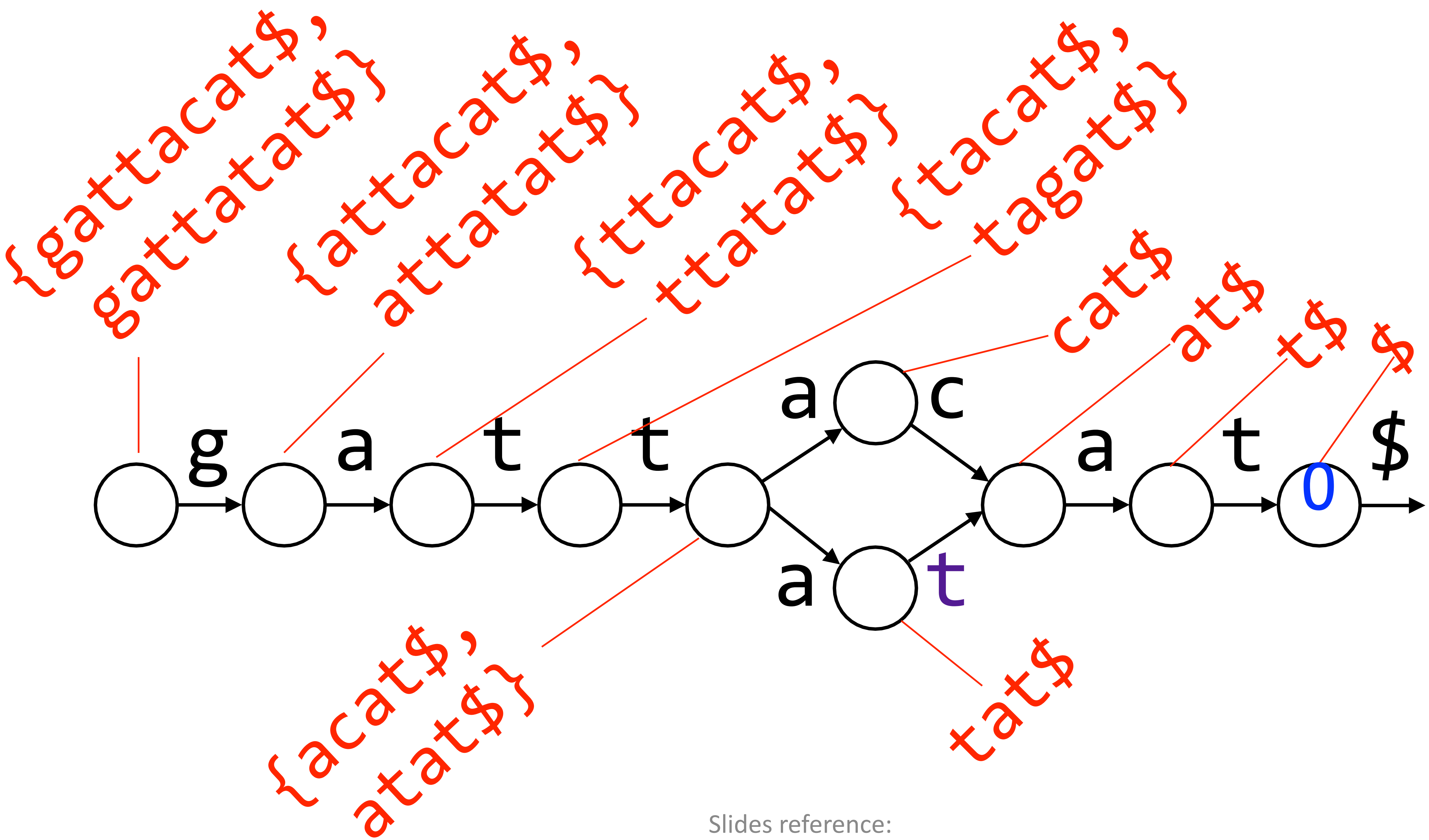
Graph has something like a BW order! Matching aga, we still have consecutivity.



Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

BWT:matching

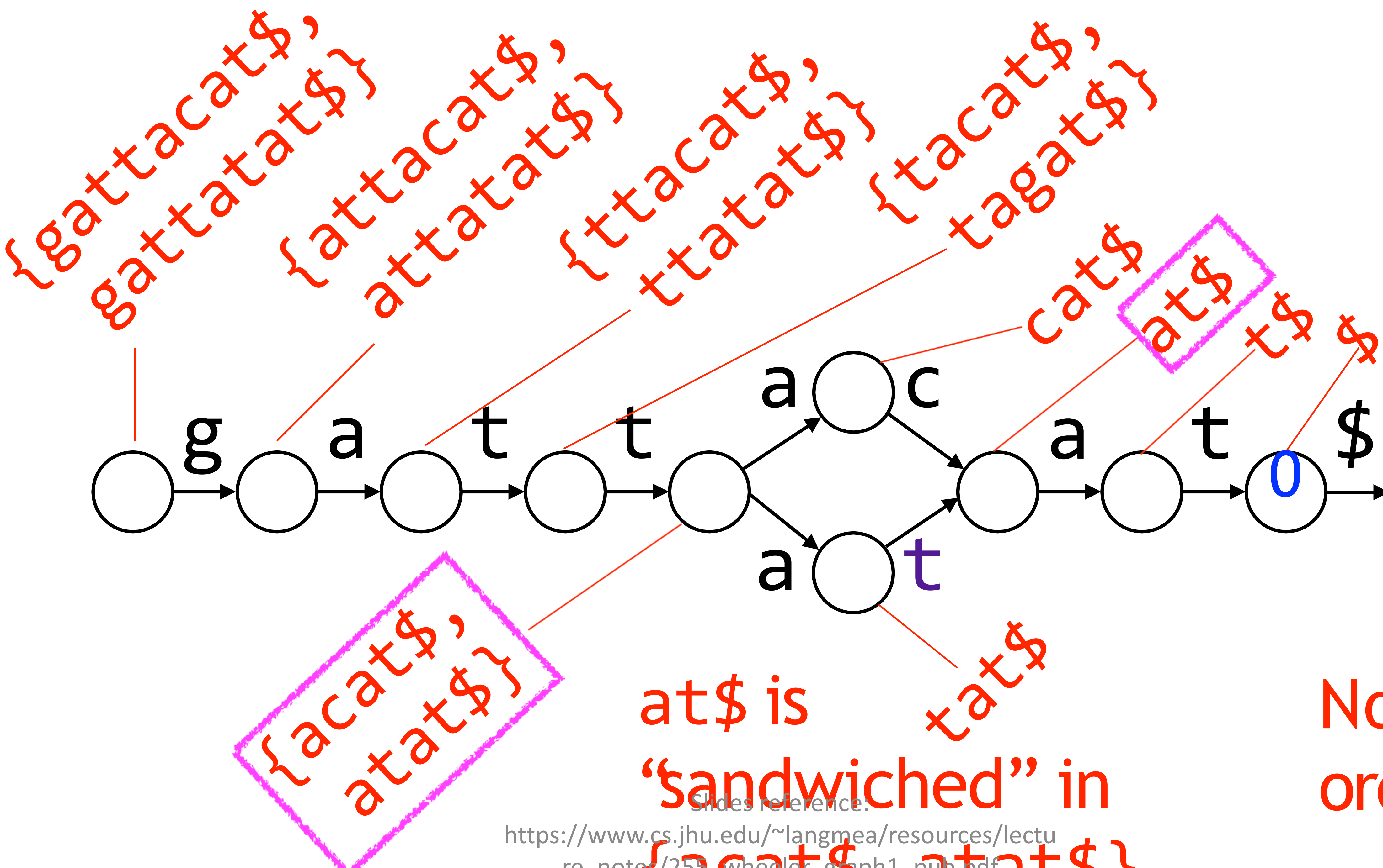
Does it work for every graph?



Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

BWT:matching

Does it work for every graph?

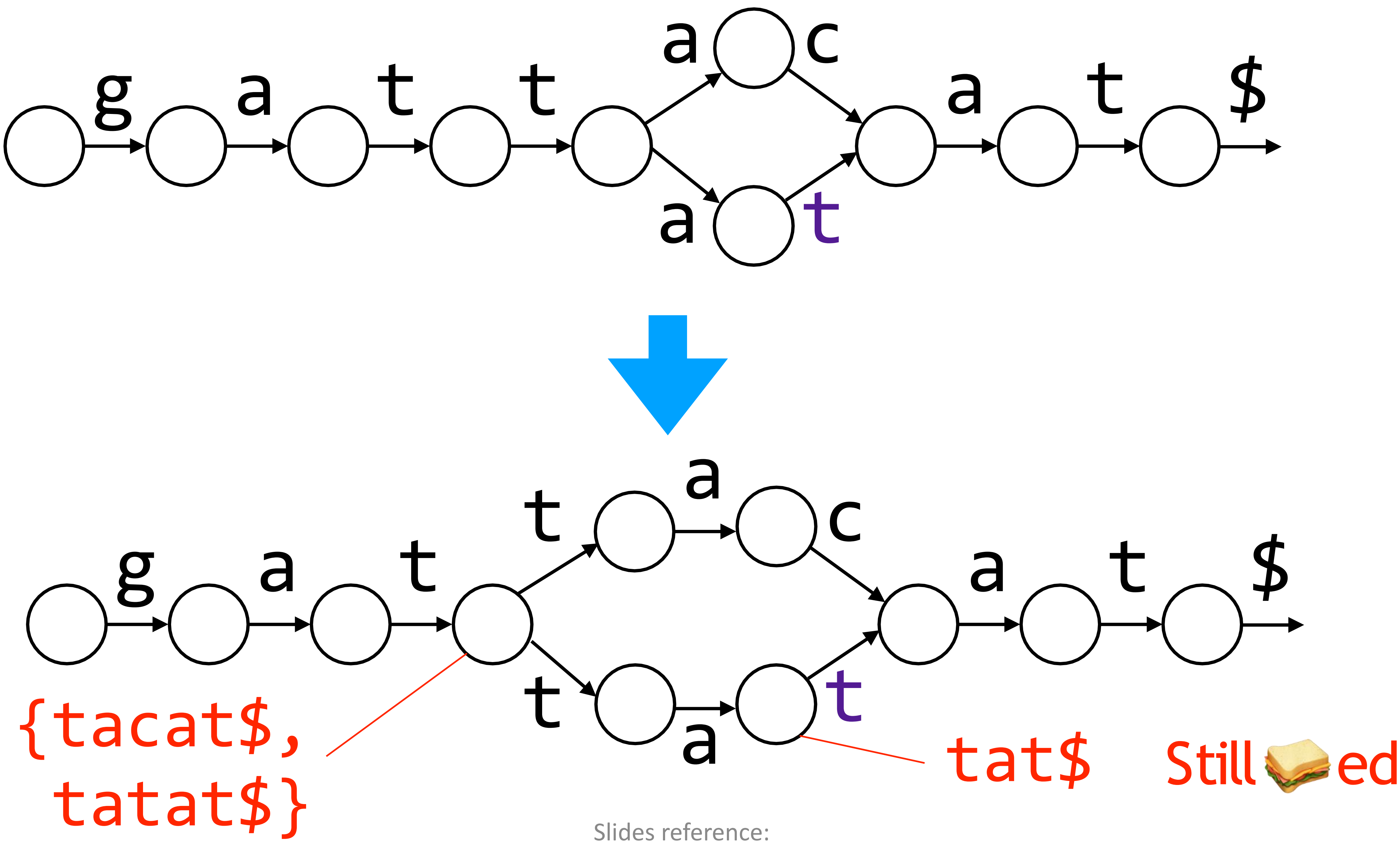


at$ is “sandwiched” in {acat$, atat$}

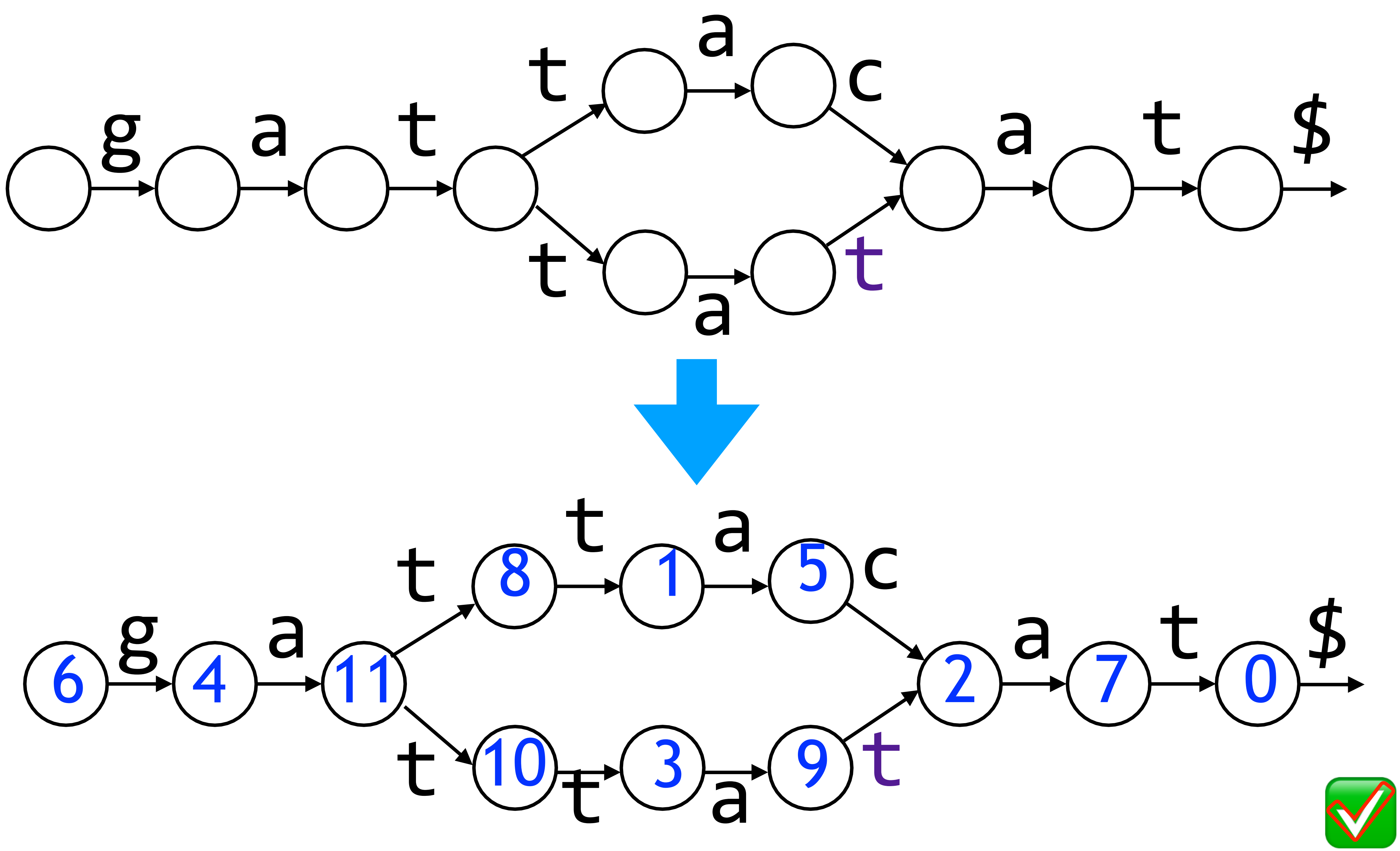
No total order!

BWT:matching

Can I fix it?



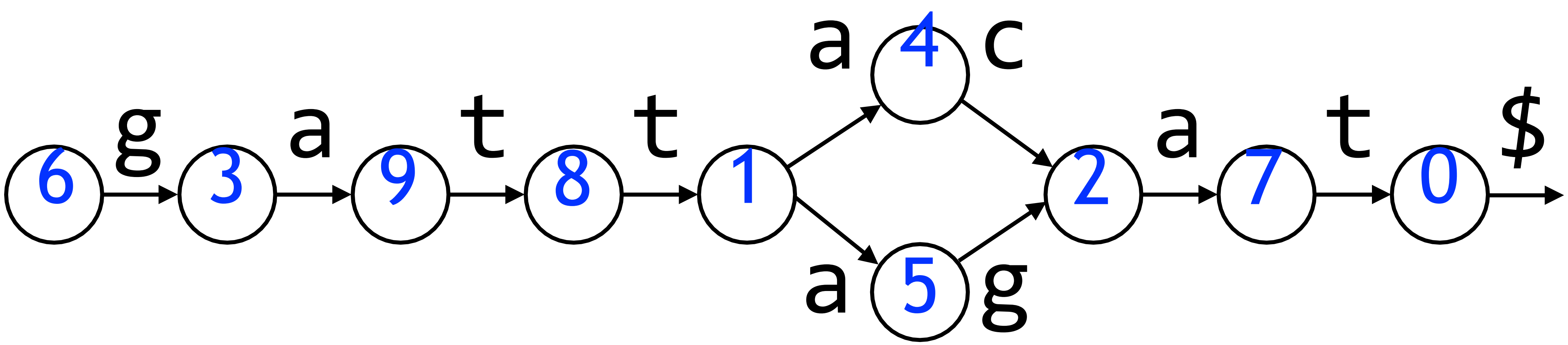
Slides reference:
https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf



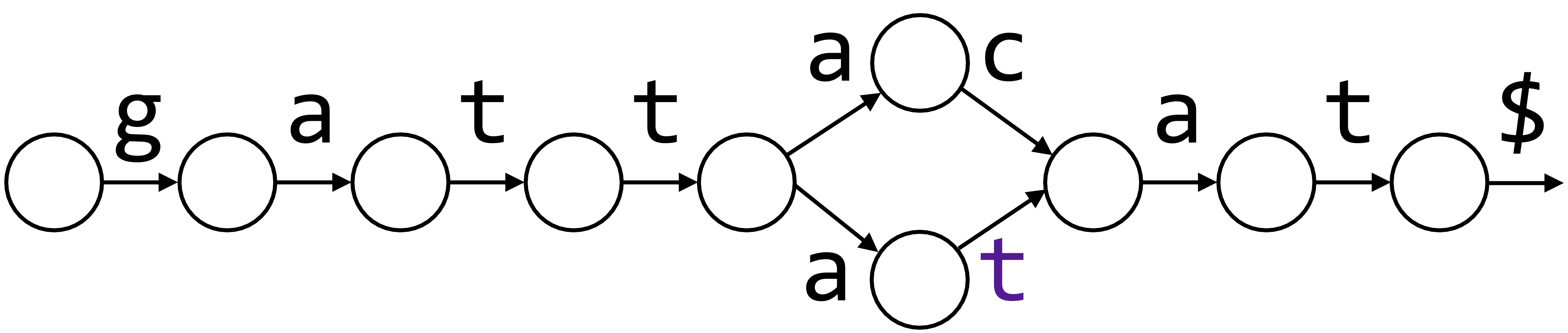
g
a
t
t
a
c
t
a
t
a
t
$
6
4
11
8
1
5
10
3
9
2
7
0

BWT:matching

For some graphs,total order exists



For others,not (but we can "fix" them sometimes)



BWT:matching

Questions:

Which graphs does it work for?

Do these graphs provably have the desired consecutivity property, so we can do matching?

How do we represent and query the graph?

Wheeler graphs

An edge-labeled directed multigraph is a ***Wheeler Graph*** if nodes can be ordered such that:

1. 0 in-degree nodes come before others
2. For all pairs of edges $e = (u, v), e' = (u', v')$ labeled $, a$ respectively, we have:

$$a \prec a' \implies v < v',$$

$$(a = a') \wedge (u < u') \implies v \le v'.$$

$\prec$ alphabetical, $<$ total order over node labels

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Wheeler graphs

For each pair of edges:

If edges have different labels, the destination of the edge with the smaller label must come before the destination of the edge with the larger label

Consequence: node cannot have 2 incoming edges with different labels

$$a \prec a' \implies v < v',$$

$$(a = a') \wedge (u < u') \implies v \le v'.$$

If edges have the same label but different sources, the destination of the edge from the low source must not come after the destination of the edge from the high source

Wheeler graphs

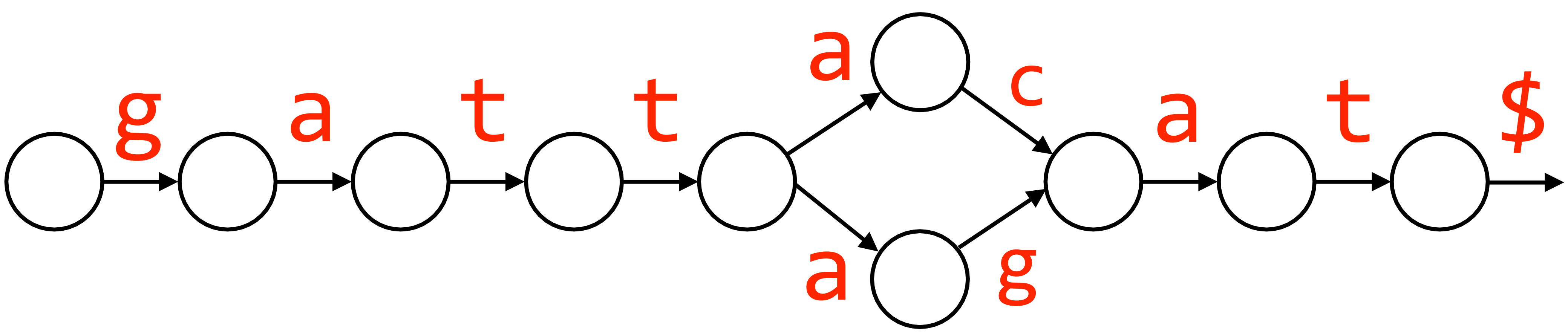
0 in-degree nodes come before others (1)

For all pairs of edges

$$a \prec a' \implies v < v' \quad (2)$$

$$(a = a') \wedge (u < u') \implies v \le v' \quad (3)$$

- Is this a Wheeler Graph?



Wheeler graphs

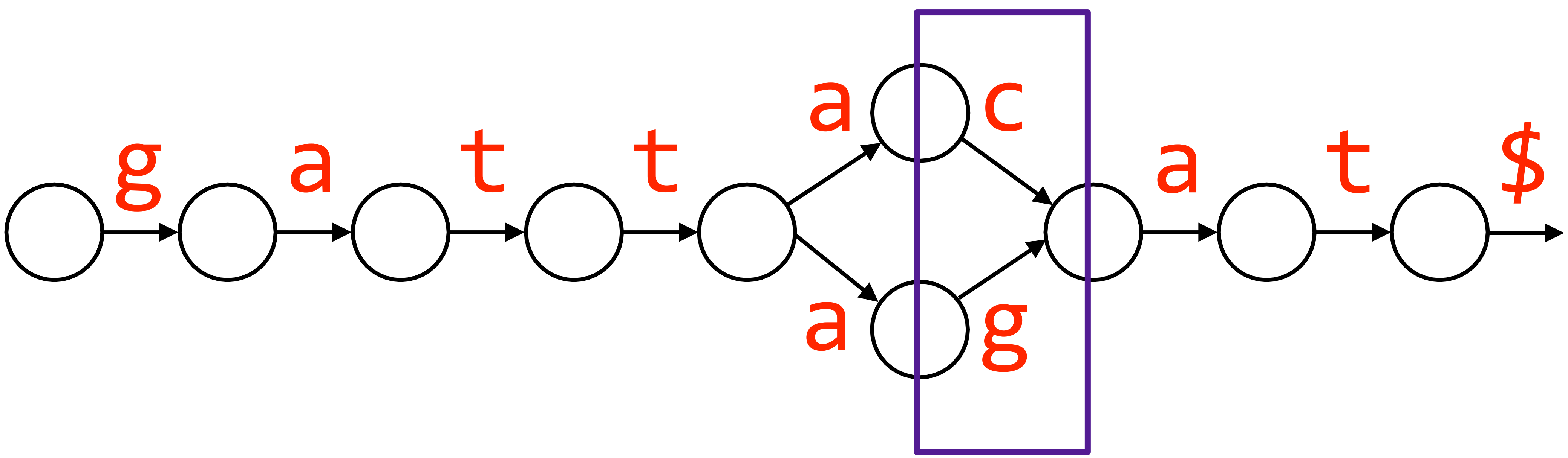
0 in-degree nodes come before others (1)

For all pairs of edges

$$a \prec a' \implies v < v' \quad (2)$$

$$(a = a') \wedge (u < u') \implies v \le v' \quad (3)$$

- Is this a Wheeler Graph? **No**



$a \prec a'$ but $v = v'$ 2) cannot hold

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Wheeler graphs

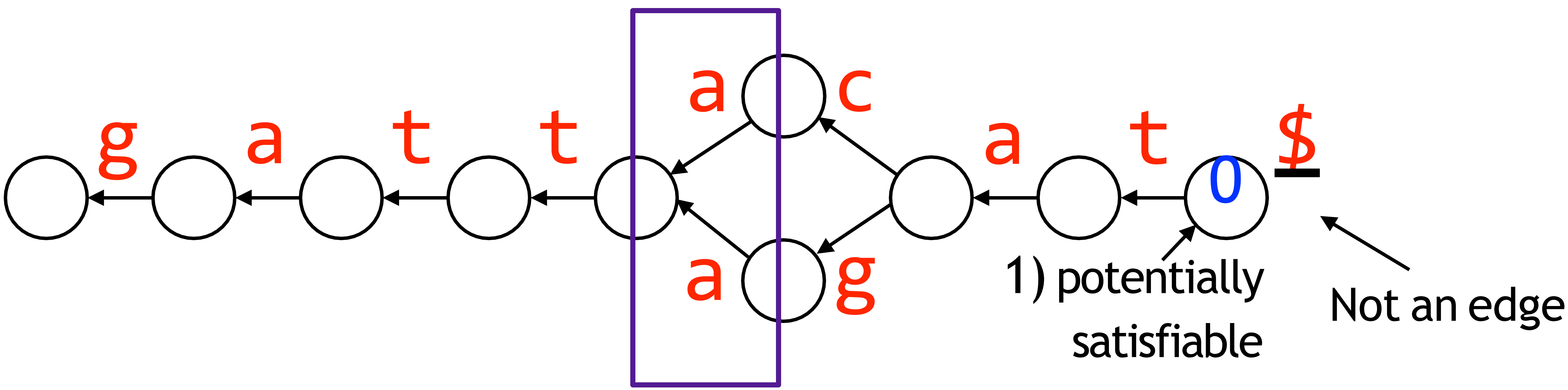
0 in-degree nodes come before others (1)

For all pairs of edges

$$a \prec a' \Longrightarrow v < v' \quad (2)$$

$$(a = a') \wedge (u < u') \Longrightarrow v \leq v' \quad (3)$$

- What if we flip edges to follow the direction of matching?



$a = a$ and $v = v$, so 3) is satisfied whether or not $u < u'$

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Wheeler graphs

0 in-degree nodes come before others (1)

For all pairs of edges

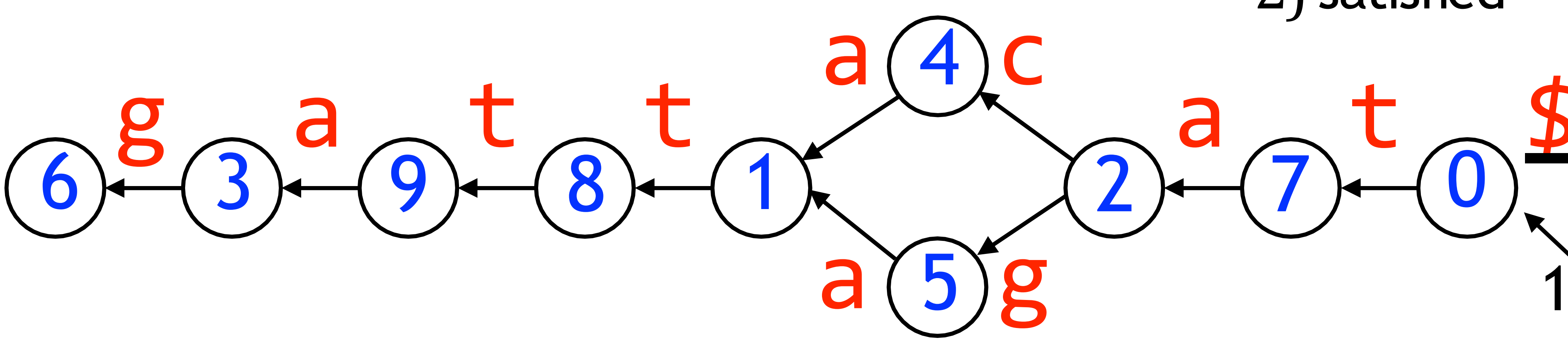
$$a \prec a' \Longrightarrow v < v' \quad (2)$$

$$(a = a') \wedge (u < u') \Longrightarrow v \leq v' \quad (3)$$

Successors of edges labeled: a {1, 2, 3 g {5, 6

c {4 t {7, 8, 9

2) satisfied



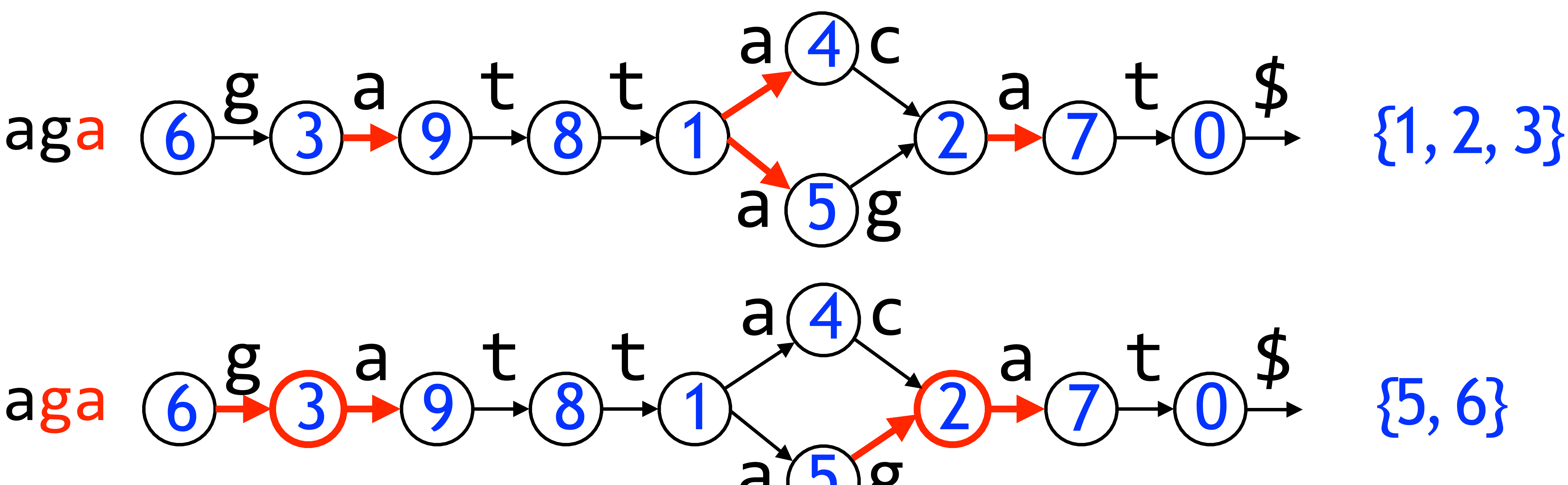
1) satisfied

Wheeler

Wheeler graphs

A graph is ***path-coherent*** if there is a total order of the nodes such that:

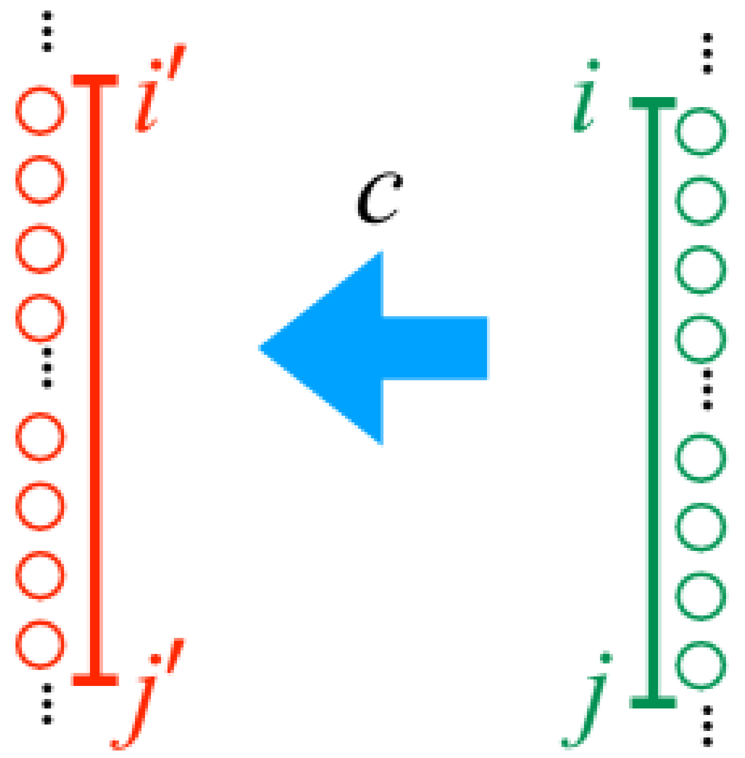
For any consecutive range [i, j] of nodes and string α, the nodes reached by following edges matching α also form a consecutive range.



Slides reference:
https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Wheeler graphs

Consider a single step where our initial set of nodes are in consecutive range $[i, j]$ and, after advancing on a single character $c \in \Sigma$, $[i', j']$ is the smallest range containing our next set of nodes



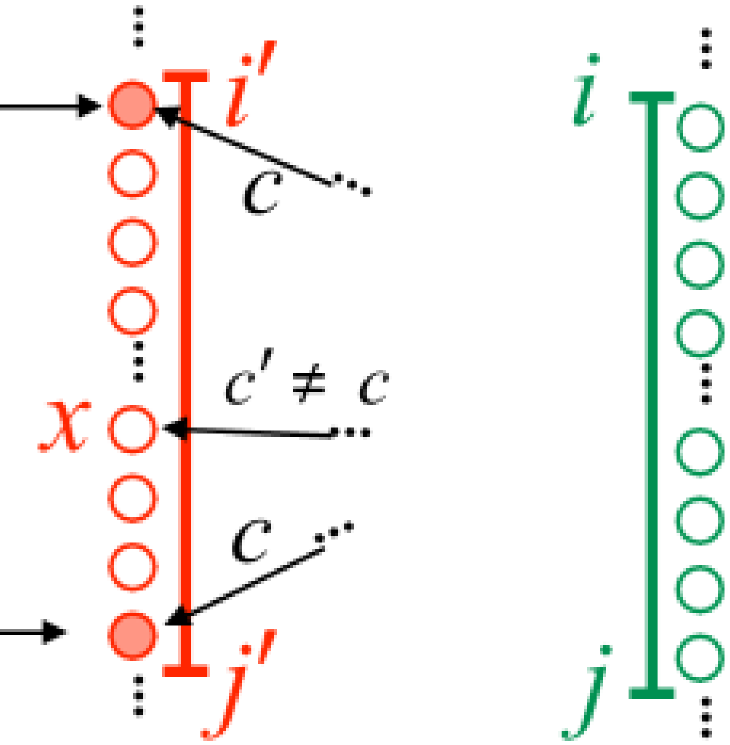
We want to show that the nodes in $[i', j']$ consist *only* of the c-successors of nodes $[i, j]$

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Wheeler graphs

As defined, *i'* is reachable via an edge labeled *c* from a node in $[i, j]$

Same for j'



Consider node x, where $i' < x < j$ with incoming edge labeled . Suppose $c' \neq c$.

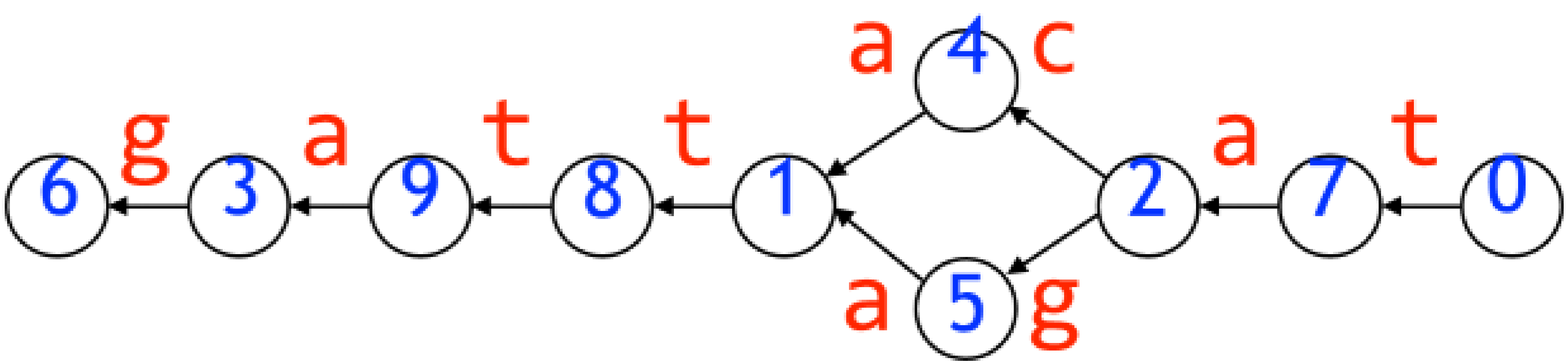
Recall: $a \prec a' \Longrightarrow v < v'$

Since $x \nless i$, we have $c' \nprec c$

Since $j' \nless x$, we have $c \nprec c$

We have $c' \succcurlyeq c$, $c \succcurlyeq c$, and $c' \neq c$, giving a contradiction

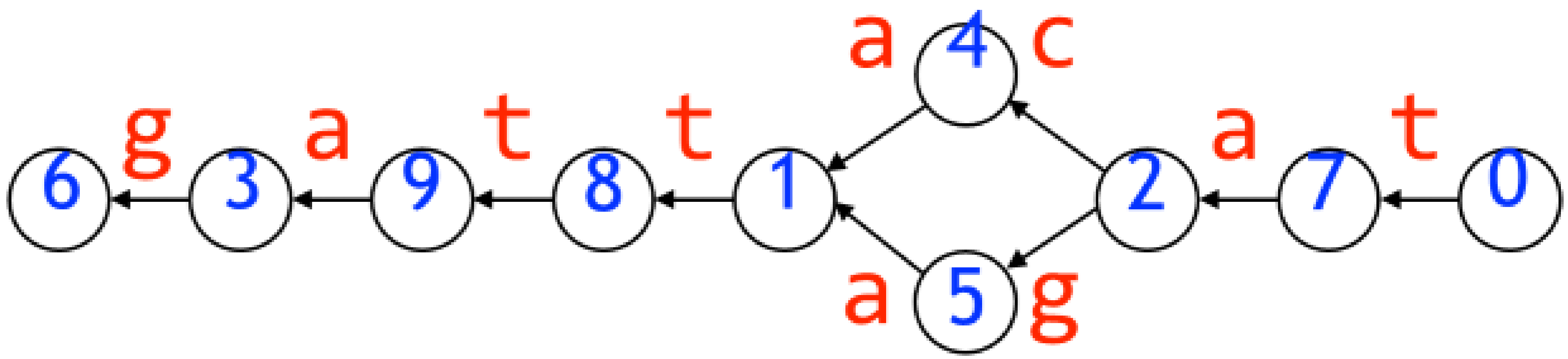
Wheeler graphs



How would we represent a Wheeler graph with bitvectors?

> Need to represent *structure* as well as *node and edge labels*

Wheeler graphs



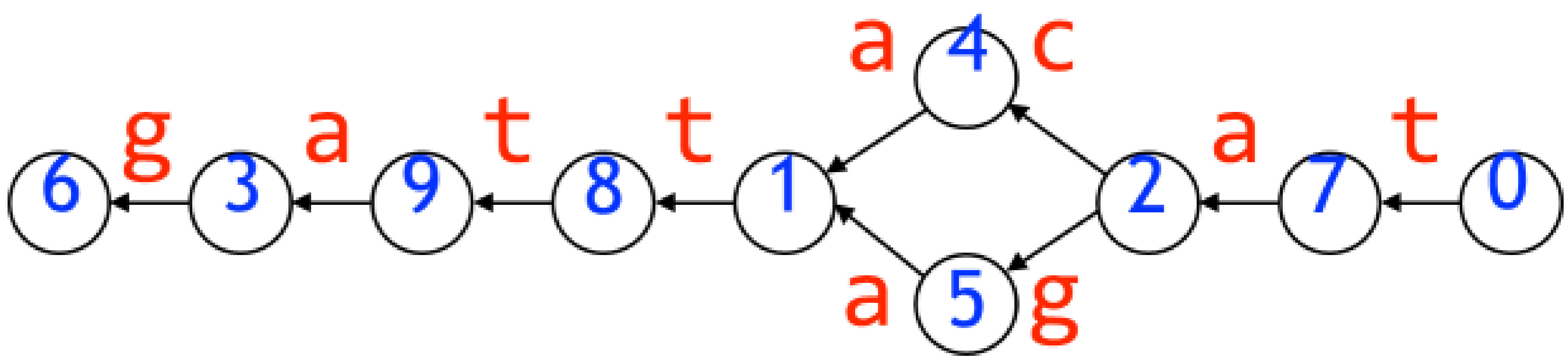
Idea 1: Encode in- and outdegree of each node in unary

Idea 2: Concatenate in order by node

Idea 3: Encode edge labels corresponding to 0s in O

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Wheeler graphs



$$I = 100101010101010101010101$$

$$O = 010100101010101101010101$$

$$L = \text{ttcggaaata}$$

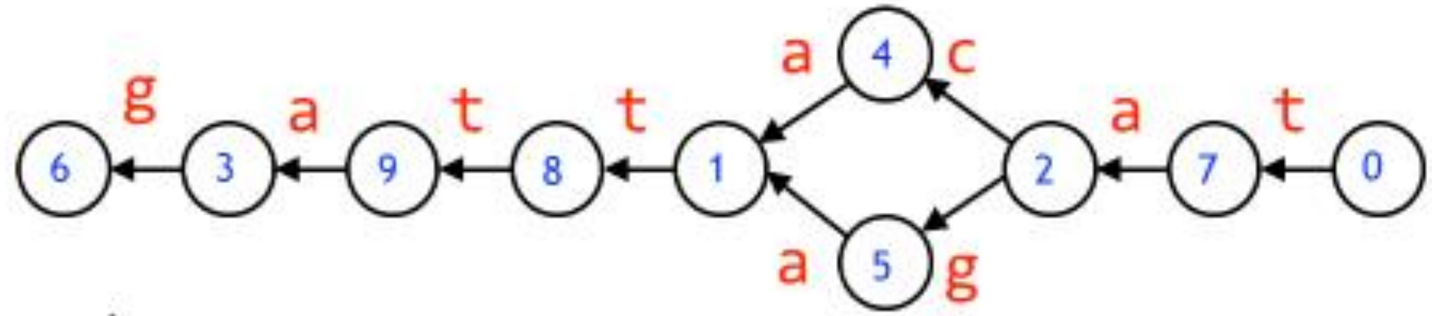
How long is ? (#edges) + (#nodes) bits

How long is ? (#edges) + (#nodes) bits

How long is L ? (#edges) chars

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

Wheeler graphs



Wheeler graph match query loop:

I :10010101010101010101

O :01010010101011010101

L :ttcggaaata

Find range of characters in L:

L.rank$_c$(. . .)

Skip: ***C[c]***

Find outgoing edges in O:

O.rank$_0$(O . select$_1$(. . .))

Follow incoming edges in I:

$\boldsymbol{I}$ **. rank$_1$($\boldsymbol{I}$. select$_0$(. . .))**

Slides reference: https://www.cs.jhu.edu/~langmea/resources/lecture_notes/255_wheeler_graph1_pub.pdf

THANK YOU!