Wheeler graphs, part 2

Ben Langmead

Please sign guestbook (www.langmead-lab.org/teaching-materials) to tell me briefly how you are using the slides. For original Keynote files, email me (ben.langmead@gmail.com).
Wheeler graphs

Can we show Wheeler Graphs are “special,” giving us analogs to FM index queries?

We start with “path coherence,” the graph version of *consecutivity*
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 $\alpha$, the nodes reached by following edges matching $\alpha$ also form a consecutive range.
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 \([i, j]\).
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 $c'$. Suppose $c' \neq c$.

Recall: $a < a' \iff v < v'$

Since $x \not< i'$, we have $c' \not< c$.

Since $j' \not< x$, we have $c \not< c'$.

We have $c' \geq c$, $c \geq c'$, and $c' \neq c$, giving a contradiction.
Wheeler graphs

Could node $x \notin [i, j]$ be a $c-$predecessor of a node $y, i' < y < j'$?

No. Proof is by contradiction, similar to previous argument and using:

$$(a = a') \land (u < u') \implies v \leq v'$$
Wheeler graphs

For any consecutive range \([i, j]\) of nodes and string \(\alpha\), the nodes reached by following edges matching \(\alpha\) also form a consecutive range

Proof takes previous arguments and extends them to string \(\alpha\) inductively
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

Here, $O = 0101001010101001101010101$
Wheeler graphs

\[ I = 10010101010101010101010101010101 \]

\[
\begin{array}{ccccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\end{array}
\]
Wheeler graphs

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

$$O = 01010010101011010101$$
$$L = t \ t \ cg \ g \ a \ a \ a \ a \ t \ a$$
Wheeler graphs

\[ I = 10010101010101010101010101 \]
\[ O = 0101001010101110101010101 \]
\[ L = \text{ttcggaaataata} \]

How long is \( I \)? \((\# \text{ edges}) + (\# \text{ nodes})\) bits
How long is \( O \)? \((\# \text{ edges}) + (\# \text{ nodes})\) bits
How long is \( L \)? \((\# \text{ edges})\) chars
Wheeler graphs

I = 1001010101010101010101010101010101

O = 01010010101010110110101011

L = ttcgggaataata

How to find in- or outdegree of a node?
Wheeler graphs

$I = 10010101010101010101010101010101$

$G \cdot \text{indegree}(i) = I \cdot \text{select}(i) - I \cdot \text{select}(i - 1) - 1$

(let $I \cdot \text{select}(-1) = 0$)

$G \cdot \text{indegree}(1) = I \cdot \text{select}(1) - I \cdot \text{select}(0) - 1$

$= 3 - 0 - 1 = 2$
Wheeler graphs

\[ O = 0101001010101011010101 \]

\[ G \cdot \text{outdegree}(i) = O \cdot \text{select}_1(i) - O \cdot \text{select}_1(i - 1) - 1 \]

(let \( O \cdot \text{select}(-1) = 0 \))

\[ G \cdot \text{outdegree}(2) = O \cdot \text{select}_1(2) - O \cdot \text{select}_1(1) - 1 \]
\[ = 6 - 3 - 1 = 2 \]
Wheeler graphs

How to access the labels of the edges outgoing from a node?
Wheeler graphs

\[ O = 01010010101011010101 \]
\[ L = \text{ttcggaaata} \]

\[ \text{off} = O \cdot \text{rank}_0(O \cdot \text{select}_1(i - 1)) \]
Wheeler graphs

\[ O = 01010010101011010101 \]
\[ L = ttcggaaataa \]

num = G . outdegree(i)
off = O . rank_0(O . select_1(i - 1))
for j in \{0, 1, \ldots, \text{num}-1\}:
    L . access(off + j)
Wheeler graphs

\[ O = 01010010101011010101 \]
\[ L = \text{ttcggaaata} \]

\[ \text{num} = G \cdot \text{outdegree}(2) = 2 \]
\[ \text{off} = O \cdot \text{rank}_0(O \cdot \text{select}_1(2 - 1)) \]

for \( j \) in \{0, 1, \ldots, \text{num}-1\}:
\[ L \cdot \text{access} (\text{off} + j) \]
Wheeler graphs

\[ O = 010100101010101101010101 \]
\[ L = ttcggaaataa \]

\text{num} = 2

\text{off} = O \cdot \text{rank}_0(O \cdot \text{select}_1(2 - 1) = 3)

\text{for } j \text{ in } \{0, 1, \ldots, \text{num}-1\}:

\text{L} \cdot \text{access} (\text{off} + j)
Wheeler graphs

\[ O = 01010010101011010101 \]
\[ L = \texttt{ttcggaaata} \]

\[ \text{num} = 2 \]
\[ \text{off} = O \cdot \text{rank}_0(3) = 2 \]

for \( j \) in \( \{0, 1, \ldots, \text{num}-1\} \):
\[ L \cdot \text{access}(\text{off} + j) \]
Wheeler graphs

\[ O = 01010010101011010101 \]
\[ L = ttcggaataa \]

```
num = 2
off = 2

for j in {0, 1}:
    L.access(off + j)
```
Wheeler graphs

\[ O = 01010010101011010101 \]
\[ L = \text{ttcggaaata} \]

\[
\begin{align*}
\text{num} &= 2 \\
\text{off} &= 2 \\
\text{for } j \text{ in } \{0, 1\}: \\
L.\text{access}(2 + 0) &= c
\end{align*}
\]
Wheeler graphs

\[ O = 01010010101011010101 \]
\[ L = \texttt{ttcggaaataa} \]
num = 2
off = 2
for \( j \) in \( \{0, 1\} \):

\[ L . \text{access}(2 + 1) = g \]
Wheeler graphs

**FM Index** match query loop:

$C[c], C[c + 1]$

$L . \text{rank}_c(\ldots) \quad \ldots + C[c])$
Wheeler graphs

**Wheeler graph** match query:

\[ \text{Wheeler graphs} \]

\[ C[c], C[c + 1] \]

\[ O \cdot \text{rank}_0(O \cdot \text{select}_1(\ldots)) \quad I \cdot \text{rank}_1(I \cdot \text{select}_0(\ldots + C[c])) \]

\[ L \cdot \text{rank}_c(\ldots) \]

\[ I : 1001010101010101010101 \]

\[ O : 0101001010101101010101 \]

\[ L : \text{ttcggaaata} \]