McCreight's Suffix Tree Algorithm

```
Construct tree for x[1..n]
for i = 1 to n do
          if head(i)=\epsilon then
                  head(i+1) = slowscan(\epsilon, s(tail(i)))
                  add i+1 and head(i+1) as node if necessary
                  continue
          u = parent(head(i)); v = label(u,head(i))
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else
                        w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  add a node for w
                  head(i+1) = w
          else if w is a node then
                  head(i+1) = slowscan(w,tail(i))
                  add head(i+1) as node if necessary
          s(\text{head}(i)) = w
          add leaf i+1 and edge between head(i+1) and i+1
```

Special Case

Construct tree for x[1..n]

s(head(i)) = w

```
for i = 1 to n do
```

```
if head(i)=\epsilon then
head(i+1) = slowscan(\epsilon,s(tail(i)))
add i+1 and head(i+1) as node if necessary
continue
```

add leaf i+1 and edge between head(i+1) and i+1

```
u = \operatorname{parent}(\operatorname{head}(i)); v = \operatorname{label}(u,\operatorname{head}) Special Case:

if u \neq \epsilon then w = \operatorname{fastscan}(s(u),v)

else w = \operatorname{fastscan}(\epsilon,v[2..|v|]) When \operatorname{head}(i) is empty

if w is an edge then add a node for w

head(i+1) = w to exploit

else if w is a node then head(i+1) = slowscan(w,\operatorname{tail}(i)) add head(i+1) as node if necessary
```

Short Cut in Search

```
Construct tree for x[1..n]
for i = 1 to n do
         if head(i)=\epsilon then
                head(i+1) = slowscan(\epsilon, s(tail(i)))
                add i+1 and head(i+1) as node if necessary
                continue
         u = \text{parent}(\text{head}(i)); v = \text{label}(u, \text{head}(i))
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
                         w = \text{fastscan}(\epsilon, v[2..|v|])
         else
         if w is an edge then
                                    Use suffix link and fastscan
                add a node for w
                head(i+1) = w
                                    to quickly find w, the
         else if w is a node then
                head(i+1) = slowscaoverlap between
                add head(i+1) as no s(head(i)) and head(i+1)
         s(\text{head}(i)) = w
         add leaf i+1 and edge between head(i+1) and i+1
```

Add inner nodes

```
Construct tree for \mathbf{x}[1..n]
```

for i = 1 to n do

```
if head(i)=\epsilon then
       head(i+1) = slowscan(\epsilon, s(tail(i)))
       add i+1 and head(i+1) as node if necessary continue Add w and head(i+1) as
u = parent(head(i)); v = label(uneeded)
if u \neq \epsilon then w = \text{fastscan}(s(u), v)
            w = fastscan(\epsilon v[2 | v])
else
if w is an edge then
       add a node for w
       head(i+1) = w
else if w is a node then
       head(i+1) = slowscan(w,tail(i))
       add head(i+1) as node if necessary
s(\text{head}(i)) = w
add leaf i+1 and edge between head(i+1) and i+1
```

Update Suffix-Link

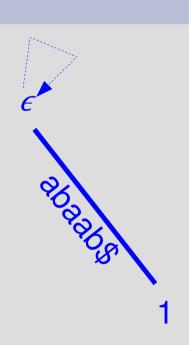
```
Construct tree for x[1..n]
for i = 1 to n do
          if head(i)=\epsilon then
                  head(i+1) = slowscan(\epsilon, s(toil(i)))
                  add i+1 and head(i+1) a make the suffix link of
                  continue
          continue
u = \text{parent}(\text{head}(i)); v = \text{label}(u, \text{meas}(i)) point to w
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
                        w = \text{fastscan}(\epsilon, v[2..|v|])
          else
          if w is an edge then
                  add a node for w
                  head(i+1) = w
          else if w is a node then
                  head(i+1) = slowscan(w,tail(i))
                  add head(i+1) as node if necessary
          s(\text{head}(i)) = w
          and leaf i+1 and edge between head(i+1) and i+1
```

Add leaf for suffix x[i..n]

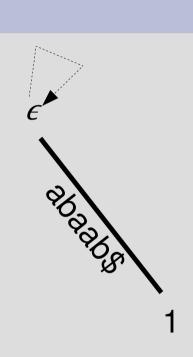
```
Construct tree for x[1..n]
for i = 1 to n do
         if head(i)=\epsilon then
                 head(i+1) = slowscan(\epsilon,s(tail(i)))
                 add i+1 and head(i+1) as node if necessary
                 continue
         u = parent(head(i)); v = label(u,head(i))
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
                      w = \text{fastscan}(\epsilon, v[2..|v|])
         else
         if w is an edge then
                 add a node for w
                 head(i+1) = w
         else if w is a node then
                 head(i+1) = slowscan(w,tail(i))
                                                        Add leaf for x[i..n]
                 add head(i+1) as node if necessary
          s(head(i)) - w
         add leaf i+1 and edge between head(i+1) and i+1
```

Example: x = abaab

```
Construct tree for x[1..n]
for i = 1 to n do
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```



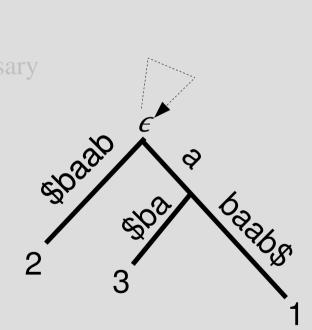
```
Construct tree for x[1..n]
for i = 1 to n do
                                              head(1)=\epsilon
          if head(i)=\epsilon then
                 head(i+1) = slowscan(\epsilon,s(tail(i)))
                  add i+1 and head(i+1) as node if necessary
                  continue
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```



```
Construct tree for x[1..n]
                                            head(2)=\epsilon
for i = 1 to n do
                                            tail(2)=baab$
                                                                  20230
          if head(i)=\epsilon then
                  head(i+1) = slowscan(\epsilon, s(tail(i)))
                  add i+1 and head(i+1) as node if necessary
                  continue
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```

```
Construct tree for x[1..n]
                               head(2)=\epsilon
                                                    head(3)=a
for i = 1 to n do
                                                    tail(3)=ab$
                               tail(2)=baab$
          if head(i)=\epsilon then
                 head(i+1) = slowscan(\epsilon, s(tail(i)))
                  add i+1 and head(i+1) as node if necessary
                  continue
                                                                e bado
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                 head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```

```
Construct tree for x[1..n]
                                                     head(3)=a
for i = 1 to n do
                                                     tail(3)=ab$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```



```
Construct tree for x[1..n]
                                                    head(3)=a
for i = 1 to n do
                                                    tail(3)=ab$
         if head(i)=\epsilon then
                 add i+1 and head(i+1) as node if necessary
                 continue
         u = parent(head(i)); v = label(u,head(i))
                                                               e Vaalo
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
         else w = \text{fastscan}(\epsilon, v[2..|v|])
         if w is an edge then
                 head(i+1) = w
         else if w is a node then
                 add head(i+1) as node if necessary
```

head(i)

```
Construct tree for x[1..n]
                                                      head(3)=a
for i = 1 to n do
                                                      tail(3)=ab$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
                                                                  $508.8D
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
                        w = \text{fastscan}(\epsilon, v[2..|v|])
          else
          if w is an edge then
                                       v[2..1] = \epsilon
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```

```
Construct tree for x[1..n]
                                                    head(3)=a
for i = 1 to n do
                                                    tail(3)=ab$
         if head(i)=\epsilon then
                 add i+1 and head(i+1) as node if necessary
                 continue
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
         else w = \text{fastscan}(\epsilon, v[2..|v|])
         if w is an edge then
                 head(i+1) = w
         else if w is a node then
                 head(i+1) = slowscan(w,tail(i))
                 add head(i+1) as node if necessary
```

head(i+1)

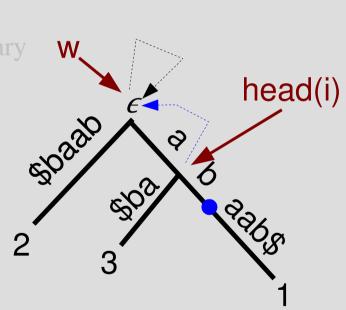
solar or head(i+1)

2

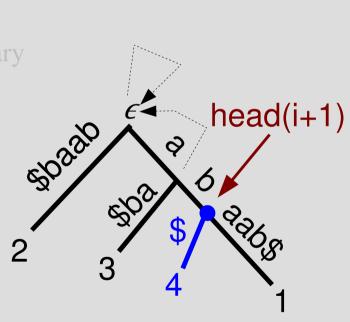
3

```
Construct tree for x[1..n]
                                                       head(3)=a
for i = 1 to n do
                                                       tail(3)=ab$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
                                                                                     head(i+1)
                                                                   <sup>2</sup>/<sub>2</sub>O32/0
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```

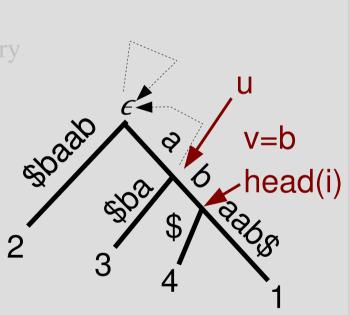
```
Construct tree for x[1..n]
                                                      head(3)=a
for i = 1 to n do
                                                     tail(3)=ab$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
          s(\text{head}(i)) = w
```



```
Construct tree for x[1..n]
                                                    head(3)=a
for i = 1 to n do
                                                    tail(3)=ab$
         if head(i)=\epsilon then
                 add i+1 and head(i+1) as node if necessary
                 continue
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
         else w = \text{fastscan}(\epsilon, v[2..|v|])
         if w is an edge then
                 head(i+1) = w
         else if w is a node then
                 add head(i+1) as node if necessary
         add leaf i+1 and edge between head(i+1) and i+1
```



```
Construct tree for x[1..n]
                                                     head(4)=ab
for i = 1 to n do
                                                     tail(4)=$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
          u = parent(head(i)) ; v = label(u,head(i))
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```



```
Construct tree for x[1..n]
                                                     head(4)=ab
for i = 1 to n do
                                                     tail(4)=$
         if head(i)=\epsilon then
                 add i+1 and head(i+1) as node if necessars (u)
                  continue
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                 head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```

v=b 2 3 4 1

```
Construct tree for x[1..n]
                                                         head(4)=ab
for i = 1 to n do
                                                         tail(4)=$
          if head(i)=\epsilon then
                   add i+1 and head(i+1) as node if necessary
                   continue
          u = \text{parent}(\text{head}(i)); v = \text{label}(u, \text{head}(i)) + \text{head}(i+1)
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                   add a node for w
                   head(i+1) = w
          else if w is a node then
                   add head(i+1) as node if necessary
```

```
Construct tree for x[1..n]
                                                       head(4)=ab
for i = 1 to n do
                                                       tail(4)=$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
          u = \text{parent}(\text{head}(i)); v = \text{label}(u, \text{head}(i))
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
          s(\text{head}(i)) = w
```

sary
(i+1)

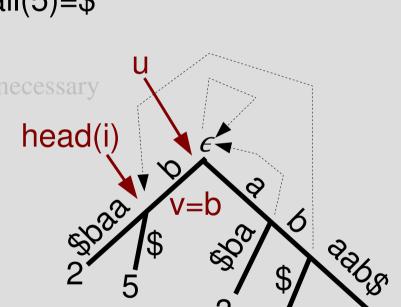
sary
head(i)

sary
2

3

```
Construct tree for x[1..n]
                                                      head(4)=ab
for i = 1 to n do
                                                      tail(4)=$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
         u = \text{parent}(\text{head}(i)); v = \text{label}(u, \text{head}(i))
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
          add leaf i+1 and edge between head(i+1) and i+1
```

```
Construct tree for x[1..n]
                                                     head(5)=b
for i = 1 to n do
                                                     tail(5)=$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
          u = parent(head(i)); v = label(u,head(i))
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                 head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```



```
Construct tree for x[1..n]
                                                       head(5)=b
for i = 1 to n do
                                                      tail(5)=$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
                        w = \text{fastscan}(\epsilon, v[2..|v|])
          else
          if w is an edge then v[2..1] = \epsilon
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
```

necessary

head(i)

v=b

v=b

solves

head(s)

```
Construct tree for x[1..n]
                                                    head(5)=b
for i = 1 to n do
                                                    tail(5)=$
         if head(i)=\epsilon then
                 add i+1 and head(i+1) as node if necessary
                 continue
                                                   head(i+1)
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
         else w = \text{fastscan}(\epsilon, v[2..|v|])
         if w is an edge then
                 head(i+1) = w
         else if w is a node then
                 head(i+1) = slowscan(w,tail(i))
                 add head(i+1) as node if necessary
```

```
Construct tree for x[1..n]
                                                    head(5)=b
for i = 1 to n do
                                                    tail(5)=$
         if head(i)=\epsilon then
                 add i+1 and head(i+1) as node if necessary
                 continue
                                                    head(i+1)
                                                   head(i)-
         if u \neq \epsilon then w = \text{fastscan}(s(u), v)
         else
         if w is an edge then
                 head(i+1) = w
         else if w is a node then
                 add head(i+1) as node if necessary
         s(\text{head}(i)) = w
```

```
Construct tree for x[1..n]
                                                      head(5)=b
for i = 1 to n do
                                                      tail(5)=$
          if head(i)=\epsilon then
                  add i+1 and head(i+1) as node if necessary
                  continue
         u = \text{parent}(\text{head}(i)); v = \text{label}(u, \text{head}(i))
          if u \neq \epsilon then w = \text{fastscan}(s(u), v)
          else w = \text{fastscan}(\epsilon, v[2..|v|])
          if w is an edge then
                  head(i+1) = w
          else if w is a node then
                  add head(i+1) as node if necessary
          add leaf i+1 and edge between head(i+1) and i+1
```

Done

