Efficient enumeration for annotated grammars

Antoine Amarilli
Louis Jachiet
Télécom, Paris

Martín Muñoz
Cristian Riveros
PUC, Chile & IMFD

for PODS, 2022

Document Spanners

Fagin, Kimelfeld, Reiss, Vansummeren 2013

Document Spanners

Fagin, Kimelfeld, Reiss, Vansummeren 2013

Information Extraction

$d =$ "Title, Director, Role \n Taxi Driver, Martin Scorsese, 1976 \n Freaky Friday, Gary Nelson, 1976 \n the Accused, Jonathan Kaplan, 1988 \n The Silence of The Lambs, Jonathan Demme, 1991 \n Little Man..."

Title	Director	Year
Taxi Driver	Martin Scorsese	1976
Freaky Friday	Gary Nelson	1976
The Accused	Jonathan Kaplan	1988
The Silence of the Lambs	Jonathan Demme	1991
Little Man Tate	Jodie Foster	1991

Document Spanners

$$d = \text{"}\underbrace{\$\text{lorem}\$\text{ipsum}\$\$\text{dolor}\$\text{sit}\$\text{amet}\$\$}_{\text{size } n}\text{"}$$

Spans: $[i,j\rangle \quad , \quad i,j \in [1, n+1] \quad , \quad i \leq j$

Document Spanners

$d =$ "$lorem$ipsum$$dolorsitamet$$"

$[2,7\rangle$ $[15,20\rangle$

Variables: $\mathcal{V} = \{x_1, x_2, \dots, x_k\}$

Mappings	x_1	x_2	$\dots$	x_k
$\mu_1 =$	$[1,5\rangle$	$[6,9\rangle$		$[10,15\rangle$
$\mu_2 =$	$[2,20\rangle$	$[3,4\rangle$	$\dots$	$[5,5\rangle$
$\mu_3 =$	$[3,3\rangle$	$[4,22\rangle$		$[6,9\rangle$

Variables: $\mathcal{V} = \{x_1, x_2, \dots, x_k\}$

Mappings or "d-tuples"

	x_1	x_2	$\dots$	x_k
$\mu_1 =$	$[1,5\rangle$	$[6,9\rangle$		$[10,15\rangle$
$\mu_2 =$	$[2,20\rangle$	$[3,4\rangle$	$\dots$	$[5,5\rangle$
$\mu_3 =$	$[3,3\rangle$	$[4,22\rangle$		$[6,9\rangle$

Variables: $\mathcal{V} = \{x_1, x_2, \ldots, x_k\}$

Mappings

	x_1	x_2	$\ldots$	x_k
$\mu_1 =$	$[1, 5\rangle$	$[6, 1\rangle$		$[10, 15\rangle$
$\mu_2 =$	$[2, 20\rangle$	$[3, 4\rangle$	$\ldots$	$[5, 5\rangle$
$\mu_3 =$	$[3, 3\rangle$	$[4, 22\rangle$		$[6, 9\rangle$

Spanner S:

$$
\begin{aligned}
S(d) = \{ & \{ x_1 : [1,5\rangle,\ x_2 : [6,9\rangle, \ldots, x_k : [10,15\rangle \}, \\
& \{ x_1 : [2,20\rangle,\ x_2 : [3,4\rangle, \ldots, x_k : [5,5\rangle \}, \\
& \{ x_1 : [3,3\rangle,\ x_2 : [4,22\rangle, \ldots, x_k : [6,9\rangle \} \}
\end{aligned}
$$

Title	Director	Year
Taxi Driver	Martin Scorsese	1976
Freaky Friday	Gary Nelson	1976
The Accused	Jonathan Kaplan	1988
The Silence of the Lambs	Jonathan Demme	1991
Little Man Tate	Jodie Foster	1991

variables

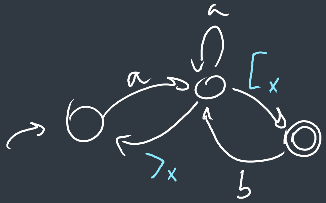
Title	Director	Year
Taxi Driver	Martin Scorsese	1976
Freaky Friday	Gary Nelson	1976
The Accused	Jonathan Kaplan	1988
The Silence of the Lambs	Jonathan Demme	1991
Little Man Tate	Jodie Foster	1991

Mappings

$$S(d) =$$

$$\{ \{ x_T : [21, 32\rangle, \quad x_D : [33, 48\rangle, \quad x_Y : [49, 53\rangle \},$$
$$\{ x_T : [54, 67\rangle, \quad x_D : [68, 79\rangle, \quad x_Y : [80, 84\rangle \},$$
$$\{ x_T : [85, 97\rangle, \quad x_D : [98, 113\rangle, \quad x_Y : [114, 118\rangle \},$$
$$\{ x_T : [119, 143\rangle, \quad x_D : [144, 158\rangle, \quad x_Y : [159, 163\rangle \},$$
$$\{ x_T : [164, 179\rangle, \quad x_D : [180, 192\rangle, \quad x_Y : [193, 197\rangle \}, \ldots \}$$

Types of Spanners:



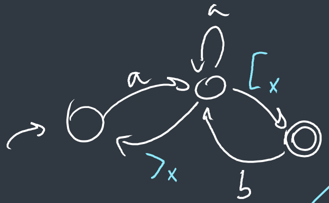
$$.^* \, [_x \, a^* b^* \backslash n \, \rangle_x \, .^*$$

$S \to AS \mid A$

$A \to BXC \qquad X \to bXa \mid \varepsilon$

$B \to [_x \qquad C \to \rangle_x$

Types of Spanners:



$$.^* \, [_x \, a^* b^* \backslash n \, \rangle_x \, .^*$$

$$S \to AS \mid A$$

$$A \to BXC \qquad X \to bXa \mid \varepsilon$$

$$B \to [_x \qquad C \to \rangle_x$$

Extraction Grammars

Peterfreund, 2020

Extraction Grammars

Peterfreund, 2020

$$S \to S'\$A\$B\$S' \qquad S' \to aS' \quad \forall a \in \Sigma$$

$$A \to [_x E \rangle_x \qquad B \to [_y E \rangle_y$$

$$E \to aE \mid a \quad \forall a \in \Sigma \setminus \{\$\}$$

Extraction Grammars

Peterfreund, 2020

$S \to S'\$A\$B\$S' \qquad S' \to aS' \quad \forall a \in \Sigma$

terminals from Σ

$A \to [_x E \rangle_x \qquad B \to [_y E \rangle_y$

$E \to aE \mid a \quad \forall a \in \Sigma \setminus \{\$\}$

Extraction Grammars

Peterfreund, 2020

$$S \to S'\$A\$B\$S' \qquad S' \to aS' \quad \forall a \in \Sigma$$

$$A \to [_x E \rangle_x \qquad B \to [_y E \rangle_y$$

terminals built from $\mathcal{V}$

$$E \to aE \mid a \quad \forall a \in \Sigma \setminus \{\$\}$$

produces words like

$$abc\$[_x ab\rangle_x\$[_y abcd\rangle_y\$a\$a$$

produces words like

$$\underbrace{abc\$[_x ab\rangle_x\$[_y abcd\rangle_y\$a\$a}_{\text{refword}}$$

input document

d = "$lorem $ipsum $$ dolor $ sit $ amet $$"

input document

$$d = \text{"}\$\ \text{lorem}\ \$\ \text{ipsum}\ \$\$\ \text{dolor}\ \$\ \text{sit}\ \$\ \text{amet}\ \$\$\text{"}$$

relevant refwords

$$r_1 = \$[_x\,\text{lorem}\rangle_x \$[_y\,\text{ipsum}\rangle_y \$\$\ \text{dolor}\ \$\ \text{sit}\ \$\ \text{amet}\ \$\$$$

$$r_2 = \$\ \text{lorem}\ \$\ \text{ipsum}\ \$\$[_x\,\text{dolor}\rangle_x \$[_y\,\text{sit}\rangle_y \$\ \text{amet}\ \$\$$$

$$r_3 = \$\ \text{lorem}\ \$\ \text{ipsum}\ \$\$\ \text{dolor}\ \$[_x\,\text{sit}\rangle_x \$[_y\,\text{amet}\rangle_y \$\$$$

$$r_1 = \$ [_x \text{lorem} \rangle_x \$ [_y \text{ipsum} \rangle_y \$ \$ \text{ dolor } \$ \text{ sit } \$ \text{ amet } \$ \$$$

$$r_2 = \$ \text{ lorem } \$ \text{ ipsum } \$ \$ [_x \text{dolor} \rangle_x \$ [_y \text{sit} \rangle_y \$ \text{ amet } \$ \$$$

$$r_3 = \$ \text{ lorem } \$ \text{ ipsum } \$ \$ \text{ dolor } \$ [_x \text{sit} \rangle_x \$ [_y \text{amet} \rangle_y \$ \$$$

$$\mathrm{out}(r_1) = \{ x : [2,7\rangle , y : [8,13\rangle \}$$

$$\mathrm{out}(r_2) = \{ x : [15,20\rangle , y : [21,24\rangle \}$$

$$\mathrm{out}(r_3) = \{ x : [21,24\rangle , y : [25,29\rangle \}$$

$$r_1 = \$[_x \text{lorem}\rangle_x \$[_y \text{ipsum}\rangle_y \$\$ \text{ dolor } \$ \text{ sit } \$ \text{ amet } \$\$$$

$$r_2 = \$ \text{ lorem } \$ \text{ ipsum } \$\$ [_x \text{dolor}\rangle_x \$[_y \text{sit}\rangle_y \$ \text{ amet } \$\$$$

$$r_3 = \$ \text{ lorem } \$ \text{ ipsum } \$\$ \text{ dolor } \$[_x \text{sit}\rangle_x \$[_y \text{amet}\rangle_y \$\$$$

$$S(d) = \Big\{ \{ x : [2,7\rangle,\ y : [8,13\rangle \},\ \{ x : [15,20\rangle,\ y : [21,24\rangle \},\ \{ x : [21,24\rangle,\ y : [25,29\rangle \} \Big\}$$

How long does it take to produce these mappings?

How long does it take
to produce these mappings?

well, a lot :/

Worst-case scenario

there is some fixed grammar G_k that produces $\Omega(n^{2k})$ outputs

How long does it take to produce these mappings?

How quickly can we produce these mappings?

Enumeration Problem

1. Preprocessing

1. Preprocessing

$\mathcal{I} = \Big($  ,

$$S \to AS \mid A$$

$$A \to BXC \qquad X \to bXa \mid \varepsilon$$

$$B \to [_x \qquad C \to \rangle_x$$

$\Big) \Rightarrow$ 

1. Preprocessing

$I = ($  ,

$S \to AS \mid A$

$A \to BXC \quad X \to bXa \mid \varepsilon$

$B \to [_x \quad C \to]_x$

$)$ $\Rightarrow$ 

2. Enumeration

1. Preprocessing

$I = ($  ,

$S \to AS \mid A$

$A \to BXC \quad X \to bXa \mid \varepsilon$

$B \to [_x \quad C \to \rangle_x$

$)$ $\Rightarrow$ 

2. Enumeration

$: \#out_1 \#out_2 \#out_3 \#out_4 \#out_5 \#out_6 \#out_7$

time, space $= O(f|\mathcal{I}|)$

1. Preprocessing

$\mathcal{I} = ($ 

, $S \to AS \mid A$
$A \to BXC \quad X \to bXa \mid \varepsilon$
$B \to \langle_x \quad C \to \rangle_x$ $)$ $\Rightarrow$ 

2. Enumeration

$: \#out_1\#out_2\#out_3\#out_4\#out_5\#out_6\#out_7$

1. Preprocessing

time, space $= O(f|\mathcal{I}|)$

$|\mathcal{D}| = O(g|\mathcal{I}|)$

$\mathcal{I} = ($

$,$

$S \to AS \mid A$

$A \to BXC \quad X \to bXa \mid \varepsilon$

$B \to [_x \quad C \to \rangle_x$

$)$

2. Enumeration

$: \#out_1 \#out_2 \#out_3 \#out_4 \#out_5 \#out_6 \#out_7$

1. Preprocessing

$I = ($  $,$

$S \to AS \mid A$

$A \to BXC \quad X \to bXa \mid \varepsilon$

$B \to [_x \quad C \to \rangle_x$

$) \Rightarrow$ 

2. Enumeration

 $: \#out_1 \#out_2 \#out_3 \#out_4 \#out_5 \#out_6 \#out_7$

1. Preprocessing

$I = ($ 

$S \to AS \mid A$

$A \to BXC \quad X \to bXa \mid \varepsilon$

$B \to [_x \quad C \to]_x$ $)$ $\Rightarrow$

2. Enumeration

<u>delay</u>

: $\#out_1 \#out_2 \#out_3 \#out_4 \#out_5 \#out_6 \#out_7$

1. Preprocessing

$I = \Bigg($  ,

$S \to AS \mid A$

$A \to BXC \quad X \to bXa \mid \varepsilon$

$B \to [_x \quad C \to]_x$

$\Bigg) \Rightarrow$ 

2. Enumeration

delay $= \Delta t$

 : $\#out_1\#out_2\#out_3\#out_4\#out_5\#out_6\#out_7$

1. Preprocessing

$$\mathcal{I} = \Big($$  $$, \quad \begin{array}{l} S \to AS \mid A \\ A \to BXC \quad X \to bXa \mid \varepsilon \\ B \to [_x \quad C \to]_x \end{array} \Big) \Rightarrow$$ 

2. Enumeration

constant delay: $\Delta t \leq c$

: $\#out_1 \#out_2 \#out_3 \#out_4 \#out_5 \#out_6 \#out_7$

Theorem (Peterfreund, 2020)

Given a spanner S from an extraction grammar G, we can enumerate the set $S(d)$ with constant delay, after a preprocessing that takes $O(|d|^5 \times |G|^2 \times q^{2k})$ time.

Our solution

Annotated Grammars

Amarilli, Jachiet
Muñoz, Riveros , 2022

Annotated Grammars

Amarilli, Jachiet
Muñoz, Riveros, 2022

$$G = (N, \Sigma, O, P, S)$$

Annotated Grammars

Amarilli, Jachiet
Muñoz, Riveros, 2022

$$G = (N, \Sigma, O, P, S)$$

output alphabet

Annotated Grammars

Amarilli, Jachiet, Muñoz, Riveros, 2022

e.g. $\mathcal{O} = \{\triangle, \circ\}$

$$S \to aSb \mid SS \mid \varepsilon \mid (a, \triangle)\, S\, (b, \circ)$$

Annotated Grammars

Amarilli, Jachiet, Muñoz, Riveros, 2022

e.g. $\Omega = \{\Delta, \circ\}$

$$S \rightarrow a S b \mid SS \mid \varepsilon \mid (a, \Delta) S (b, \circ)$$

terminals from Σ

Annotated Grammars

Amarilli, Jachiet, Muñoz, Riveros, 2022

e.g. $\Omega = \{\triangle, \circ\}$

$$S \to aSb \mid SS \mid \varepsilon \mid (a, \triangle)\, S\, (b, \circ)$$

terminals from $\Sigma \times \Omega$

produces words like

$$\hat{d} = aba\overset{\Delta}{a}b\overset{O}{b}aabb$$

annotated words

produces words like

$$\hat{d} = ab(a, s)ab(b, o)aabb$$

annotated words

produces words like

$$\hat{d} = ab\overset{\triangleright}{a}ab\overset{\circ}{b}aabb$$

annotated words

input

$d = $ "abaabb"

input

$$d = \text{"abaabb"}$$

relevant annotated words

$$\hat{d}_1 = abaabb$$

$$\hat{d}_2 = abaabb$$

$$\hat{d}_3 = abaabb$$

input

$$d = \text{"ababbb"}$$

relevant annotated words

$$\hat{d}_1 = ababbb \qquad \hat{d}_4 = ababbb$$

$$\hat{d}_2 = ababbb \qquad \hat{d}_5 = ababbb$$

$$\hat{d}_3 = ababbb \qquad \hat{d}_6 = ababbb$$

$$d = \text{"abaabb"}$$

$$[\![G]\!](d) = \{$$

$$\ldots,$$

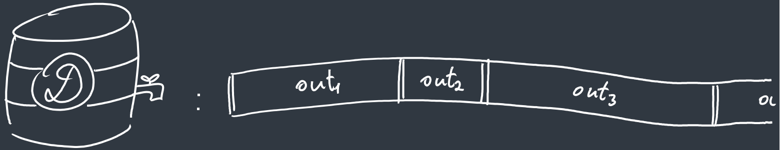
$$\mathrm{ann}(a\,b\,\overset{\triangle}{a}\,\overset{\triangle}{a}\,\overset{\circ}{b}\,\overset{\circ}{b}) = (\triangle, 3)(\triangle, 4)(\circ, 5)(\circ, 6),$$

$$\mathrm{ann}(\overset{\triangle}{a}\,\overset{\circ}{b}\,a\,\overset{\triangle}{a}\,\overset{\circ}{b}\,b) = (\triangle, 1)(\circ, 2)(\triangle, 4)(\circ, 5),$$

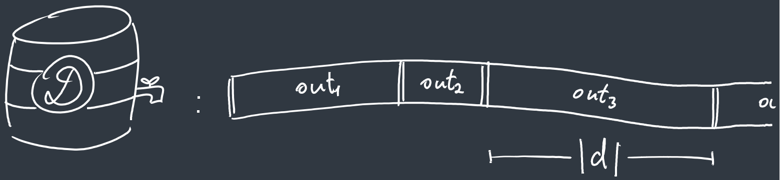
$$\mathrm{ann}(\overset{\triangle}{a}\,\overset{\circ}{b}\,\overset{\triangle}{a}\,\overset{\triangle}{a}\,\overset{\circ}{b}\,\overset{\circ}{b}) = \ldots$$

$$\}$$

Output-linear delay



Output-linear delay



Output-linear delay

: 1

Output-linear delay

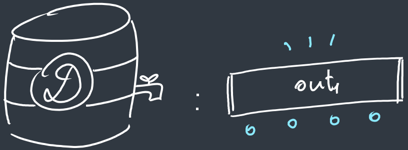
: $\|$

6

Output-linear delay

Output-linear delay

Output-linear delay



Output-linear delay

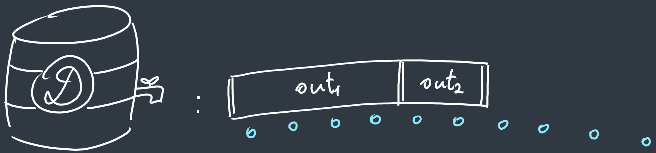
Output-linear delay

Output-linear delay

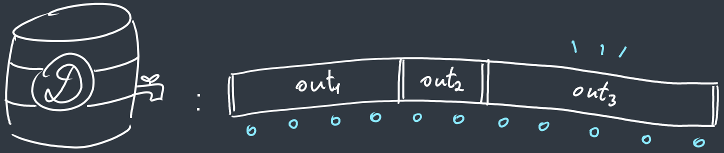
Output-linear delay

Output-linear delay

Output-linear delay



Output-linear delay



Theorem (Amarilli, Jachiet, Muñoz, Riveros, 2022)

Given an annotated grammar G, we can enumerate the set $[\![G]\!](d)$ with output-linear delay, after a preprocessing that takes $O(|d|^3 \times |G|)$ time.

Can we use this for spanners?

Can we use this for spanners?

Extraction grammars

Annotated grammars

Extraction grammars

Annotated grammars

1st, remove invalid refwords

(Peterfreund, 2020)

Extraction grammars $\rightarrowtail$ Annotated grammars

$$(G, d) \to (G', d\$)$$

Extraction grammars $\rightarrowtail$ Annotated grammars

$$(G, d) \rightarrow (G', d\$)$$

(because spans go up to n+1)

Extraction grammars

Annotated grammars

variable set $\mathcal{V}$ $\longrightarrow$ output alphabet

$$\mathcal{O} = 2^{\{[x, \rangle_x \mid x \in \mathcal{V}\}}$$

Extraction grammars

Annotated grammars

$$A \to [_x [_y \, a \quad \longrightarrow \quad A \to (a, \{[_x, [_y\})$$

Extraction grammars $\rightleftarrows$ Annotated grammars

what if

$A \to a[_x$?

$\to$ ...

Extraction grammars

Annotated grammars

what if

$A \to a\,[_x$

?

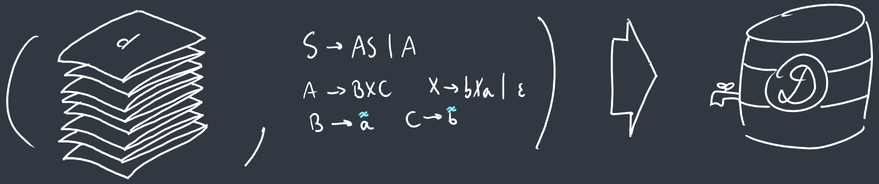
$\longrightarrow$

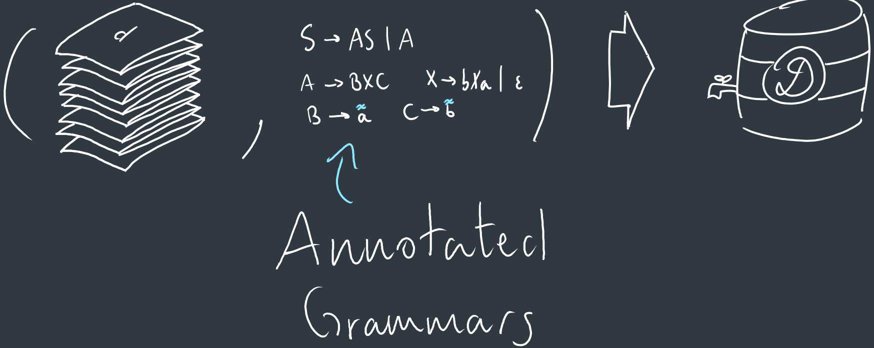
...

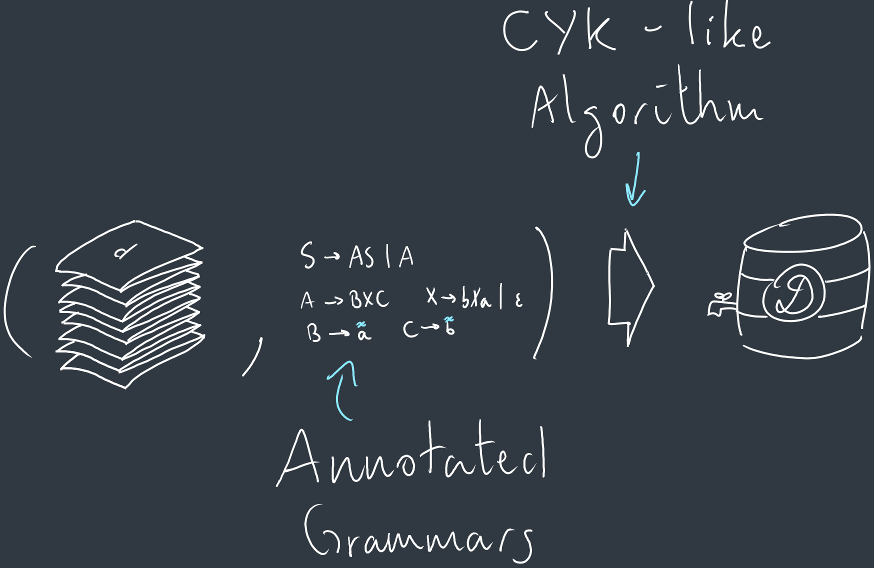
it gets ugly

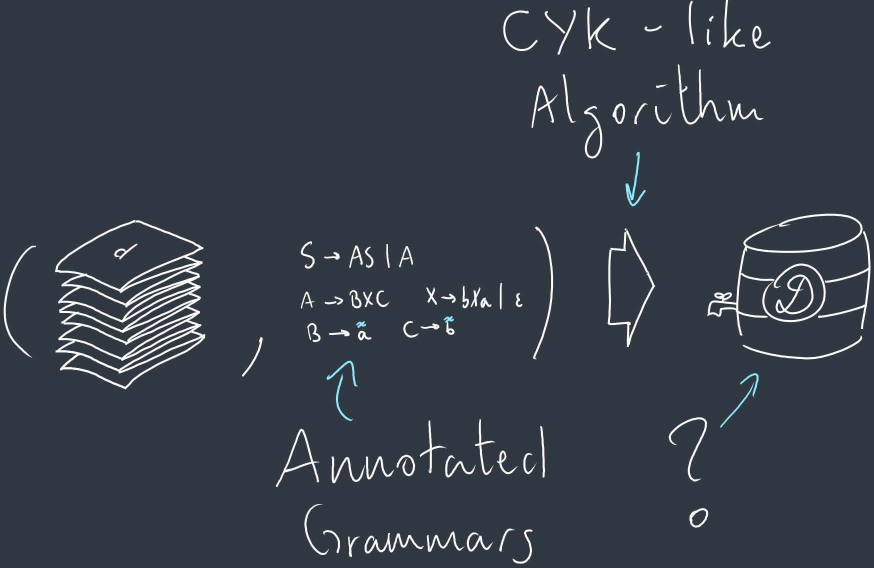
Theorem (Amarilli, Jachiet, Muñoz, Riveros, 2022)

Given a spanner S from an extraction grammar G, we can enumerate the set $S(d)$ with constant delay, after a preprocessing that takes $O(|d|^3 \times |G|^2 \times q^{3k})$ time.


d
S → AS | A
A → BXC
X → bXa | ε
B → a
C → b
D


S → AS | A
A → BXC
X → bXa | ε
B → a
C → b
Annotated Grammars


CYK - like
Algorithm
d
S → AS | A
A → BXC X → bXa | ε
B → ã C → b̃
D
Annotated
Grammars


CYK-like Algorithm
S → AS | A
A → BXC
X → bXa | ε
B → a
C → b
Annotated Grammars
?

Enumerable Set

Enumerable Set

- Stores sets of strings in some alphabet

Enumerable Set

- Stores sets of strings in some alphabet
- They can be enumerated quickly

Enumerable Set

$\mathcal{O} \times \{1, \dots, n\}$

- Stores sets of strings in some alphabet
- They can be enumerated quickly

Enumerable Set

- Amarilli, Bourhis, Jachiet, Mengel (ICALP 2017)
- Muñoz, Riveros (ICDT 2022)

- Stores sets of strings in some alphabet
- They can be enumerated quickly

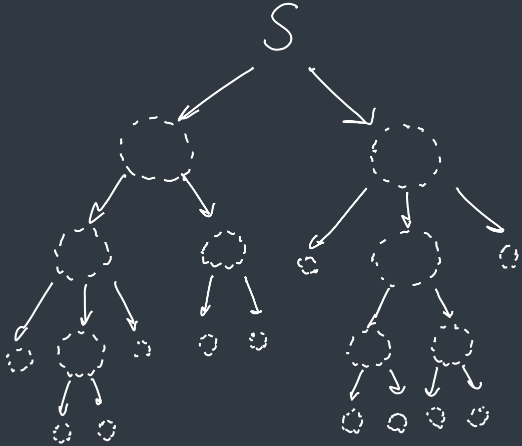
What else can we do with annotated grammars?

Rigid annotated grammars

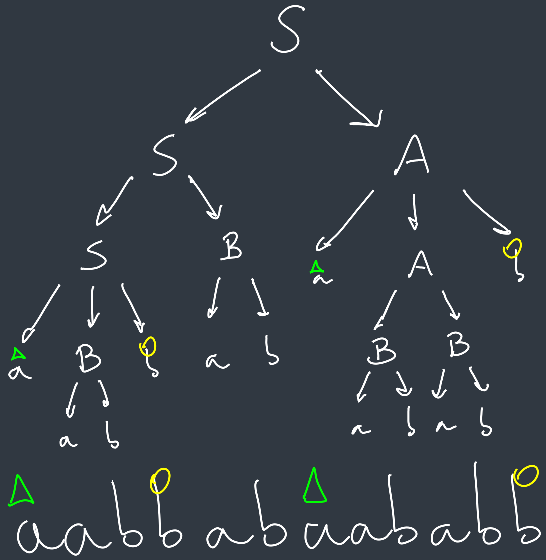
Rigid annotated grammars

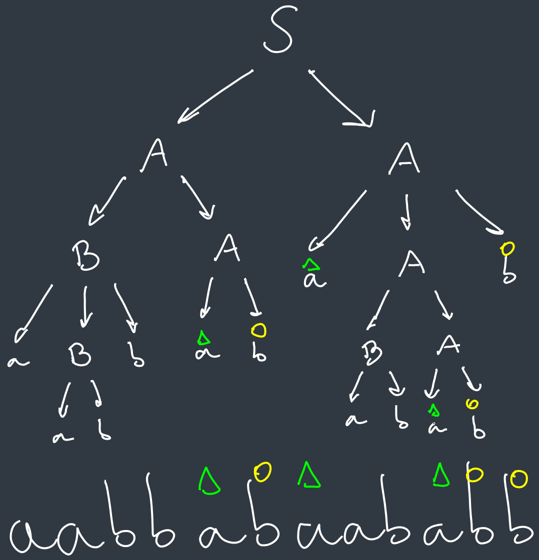
- Given an input d,
  all relevant $\hat{d}$'s are produced
  with the same tree "shape"

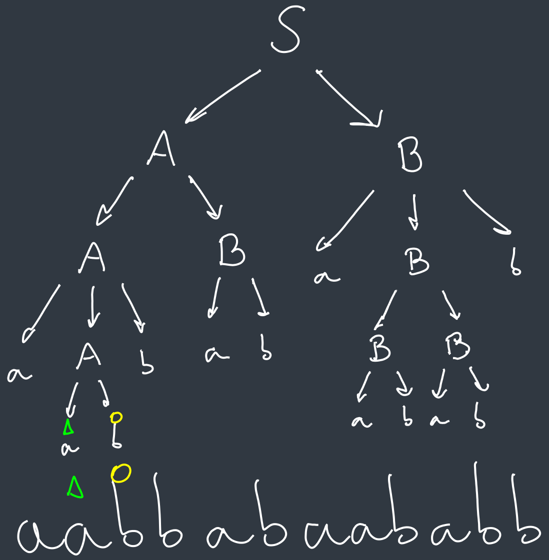
$d = aabbabaababb$

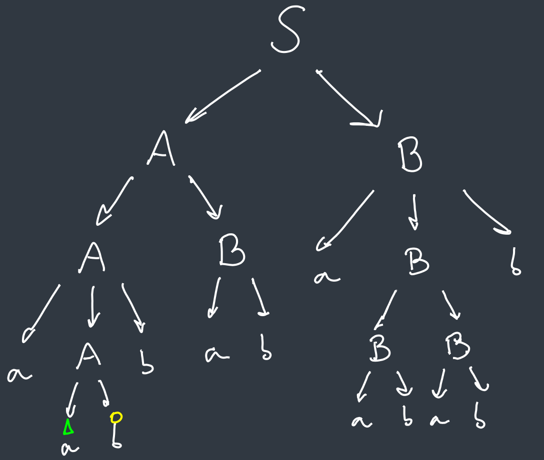


aabbabaababb

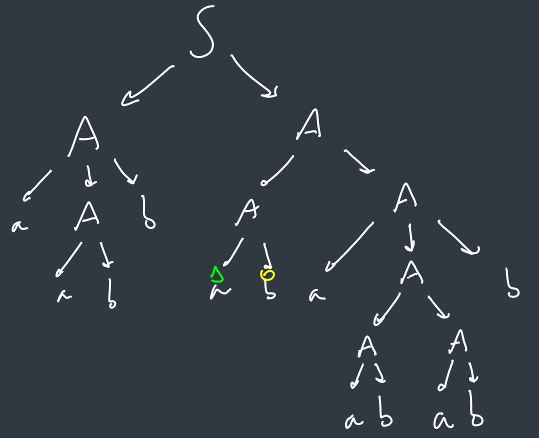

S
S
A
S
B
a
A
b
a
B
b
a
b
B
B
a
b
a
b
a
b
aabbabaababb


aabbabaababb

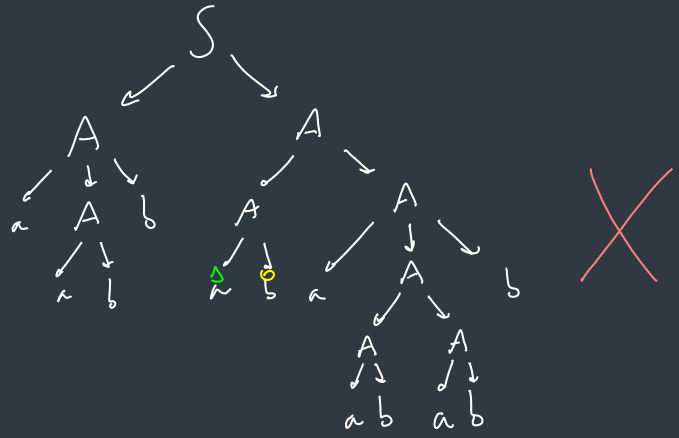

S
A B
A B a B b
a A b a b B B
a b a b a b
aabbabaababb



aabbabaababb



aabbabaababb



aabbabaababb

Theorem (Amarilli, Jachiet, Muñoz, Riveros, 2022)

Given a rigid annotated grammar G, we can enumerate the set $[\![G]\!](d)$ with output-linear delay, after a preprocessing that takes $O(|d|^2 \times |G|)$ time.

Pushdown annotators

Pushdown annotators

- An annotated grammar but it's a PDA.

Profiled-deterministic pushdown annotator

Profiled-deterministic pushdown annotator

- It is the PDA equivalent of rigid

Profiled-deterministic pushdown annotator

- It is the PDA equivalent of rigid
- Also we can find the "shape" deterministically

Theorem (Amarilli, Jachiet, Muñoz, Riveros, 2022)

Given a profiled-deterministic PDAnn $\mathcal{P}$, we can enumerate the set $[\![\mathcal{P}]\!](d)$ with output-linear delay, after a preprocessing that takes $O(|d| \times f(|\mathcal{P}|))$ time.

Conclusions

Conclusions

- Annotated grammars are nice.

Conclusions

- Annotated grammars are nice.
- Working with annotations instead of spans makes enumeration easier.

Conclusions

- Annotated grammars are nice.
- Working with annotations instead of spans can make enumeration easier.

Thank you!