CIBB 2019

Modification of Valiant's Parsing Algorithm for String-Searching Problem

Semyon Grigorev, **Yuliya Susanina**, Anna Yaveyn

JetBrains Research, Programming Languages and Tools Lab
Saint Petersburg University

September 6, 2019

Formal grammars and languages

- $G = (\Sigma, N, R, S)$ — context-free grammar (CFG) in normal Chomsky form
  - $A \to BC$, where $A, B, C \in N$
  - $A \to a$, where $A \in N, a \in \Sigma$
  - $S \to \varepsilon$, where ε is an empty string
- $L_G(A) = \{\omega \mid A \Rightarrow^* \omega\}$, where $A \in N$, $\omega \in \Sigma^*$
- Parsing — does ω belong to $L_G(S)$?

RNA analysis

- RNA sequences are treated as strings over $\{A, G, C, U\}$
- Formal grammars describe RNA secondary structure features
- Parsing as method to find all strings or substrings with these features
- Applications: RNA secondary structure prediction, classification and recognition problems
  - *Eddy S. R., Durbin R.* "RNA Sequence Analysis Using Covariance Models" 1994
  - *Knudsen B., Hein J.* “Rna secondary structure prediction using stochastic context-free grammars and evolutionary history” 1999
  - *Grigorev S., Lunina P.* “The composition of dense neural networks and formal grammars for secondary structure analysis” 2019

Tabular parsing algorithms

- Input:
  - Grammar $G = (\Sigma, N, R, S)$ in Chomsky normal form
  - String $\omega = a_1 a_2 \dots a_n$, $a_i \in \Sigma$
- Parsing table T:
  - $T_{i,j} = \{A | A \in N, a_{i+1} \dots a_j \in L_G(A)\} \quad \forall i < j$
  - $\omega \in L_G(S) \iff S \in T_{0,n}$
- Process of filling:
  - $T_{i-1,i} = \{A | A \to a_i \in R\}$
  - $T_{i,j} = f(P_{i,j})$, where $P_{i,j} = \bigcup\limits_{k=i+1}^{j-1} T_{i,k} \times T_{k,j}$

$$f(P_{i,j}) = \{A | \exists A \to BC \in R : (B, C) \in P_{i,j}\}$$

Computational complexity

- CYK: $\mathcal{O}(|G|n^3)$
  Younger, D. H. "Context-free language processing in time n^3" 1966
- GFPQ: $\mathcal{O}(|G|n^2 BMM(n))$
  Azimov, R. and Grigorev, S. "Context-free path querying by matrix multiplication" 2018

Computational complexity

- CYK: $\mathcal{O}(|G|n^3)$
  Younger, D. H. "Context-free language processing in time n^3" 1966
- GFPQ: $\mathcal{O}(|G|n^2 BMM(n))$
  Azimov, R. and Grigorev, S. "Context-free path querying by matrix multiplication" 2018
- Valiant: $\boxed{\mathcal{O}(|G|BMM(n)log(n))}$
  Valiant, L. G. "General context-free recognition in less than cubic time" 1975

Valiant's parsing algorithm

- Reduction to matrix multiplication

  $X, Y \in T$

  $X \times Y = Z$, where $Z_{i,j} = \bigcup_{k=1}^{l} X_{i,k} \times Y_{k,j}$

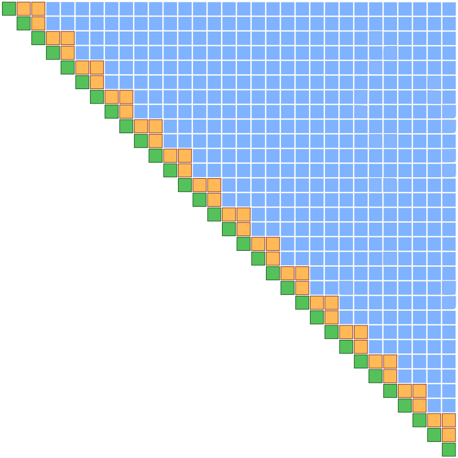
- Reduction to Boolean matrix multiplication

  $Z_{i,j}^{(B,C)} = 1 \iff (B, C) \in Z_{i,j}$

  $Z^{(B,C)} = X^B \times Y^C$

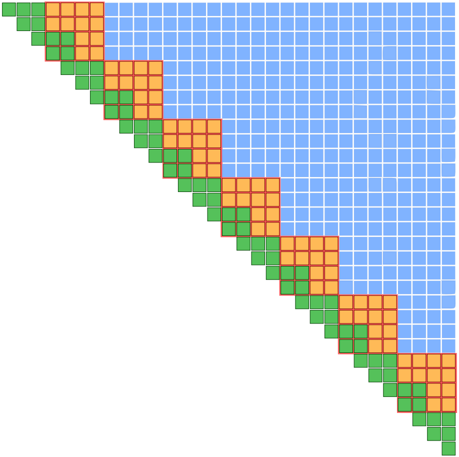
Layered submatrices processing (1)

- Rearranging the order in which submatrices are processed in Valiant's algorithm
- Division the parsing table into layers of disjoint submatrices



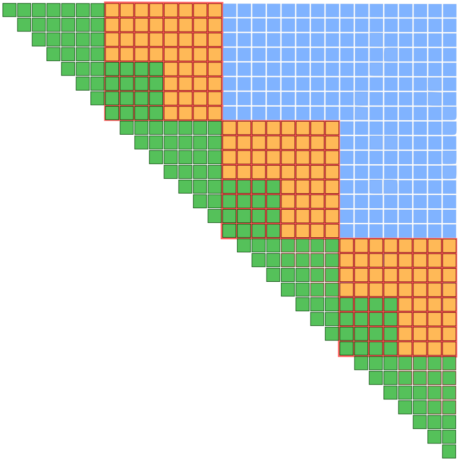
Layered submatrices processing (1)

- Rearranging the order in which submatrices are processed in Valiant's algorithm
- Division the parsing table into layers of disjoint submatrices



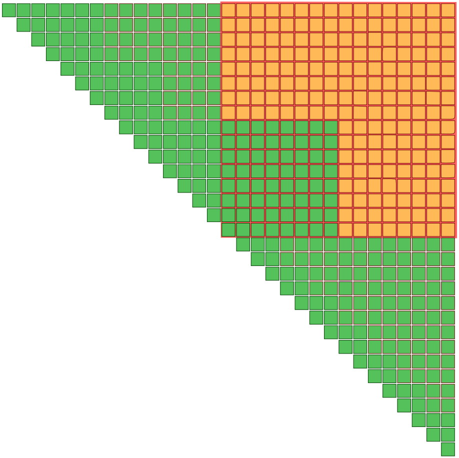
Layered submatrices processing (1)

- Rearranging the order in which submatrices are processed in Valiant's algorithm
- Division the parsing table into layers of disjoint submatrices



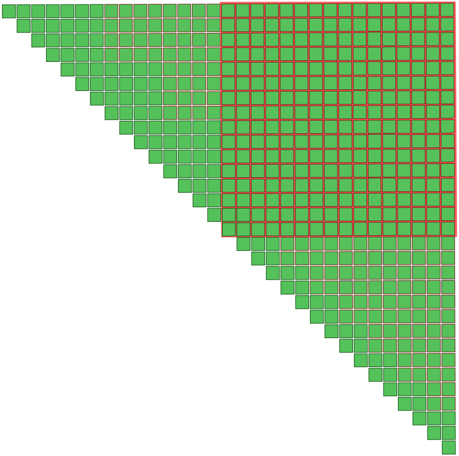
Layered submatrices processing (1)

- Rearranging the order in which submatrices are processed in Valiant's algorithm
- Division the parsing table into layers of disjoint submatrices



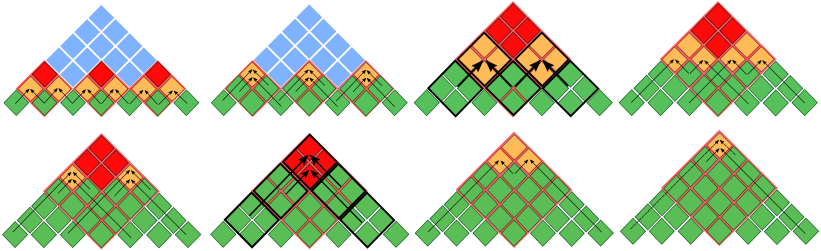
Layered submatrices processing (1)

- Rearranging the order in which submatrices are processed in Valiant's algorithm
- Division the parsing table into layers of disjoint submatrices



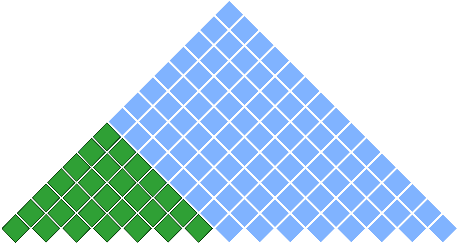
Layered submatrices processing (2)

- Each matrix in the layer can be handled independently
- Increasing the lever of parallelism:
  - Matrix multiplication
  - Each matrix in layer
  - Each pair of nonterminals



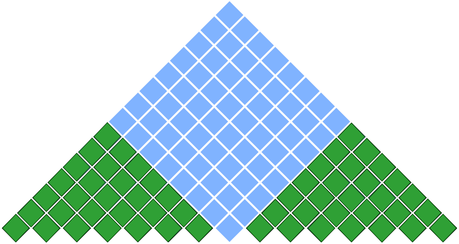
String-searching problem

- **Problem:** for input string of length $n = 2^p - 1$ find all substrings of length s which belong to $L_G(S)$
- **Valiant's algorithm:** it is necessary to calculate at least 2 triangle submatrices of size $\frac{n}{2}$
$\mathcal{O}(|G|BMM(2^{p-1})(p-2))$



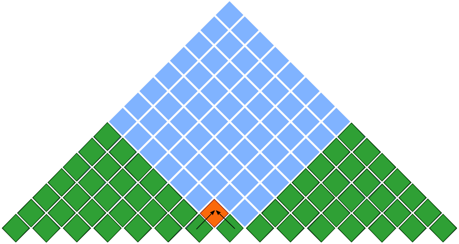
String-searching problem

- **Problem:** for input string of length $n = 2^p - 1$ find all substrings of length s which belong to $L_G(S)$
- **Valiant's algorithm:** it is necessary to calculate at least 2 triangle submatrices of size $\frac{n}{2}$
$\mathcal{O}(|G|BMM(2^{p-1})(p-2))$



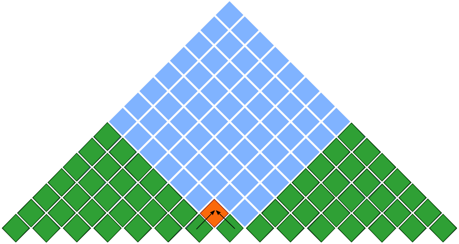
String-searching problem

- **Problem:** for input string of length $n = 2^p - 1$ find all substrings of length s which belong to $L_G(S)$
- **Valiant's algorithm:** it is necessary to calculate at least 2 triangle submatrices of size $\frac{n}{2}$
$\mathcal{O}(|G|BMM(2^{p-1})(p-2))$



String-searching problem

- **Problem:** for input string of length $n = 2^p - 1$ find all substrings of length s which belong to $L_G(S)$
- **Valiant's algorithm:** it is necessary to calculate at least 2 triangle submatrices of size $\frac{n}{2}$
  $\mathcal{O}(|G|BMM(2^{p-1})(p-2))$



- **Modification:** it is necessary to compute layers with submatrices of size not greater than 2^r, где $2^{r-2} < s \leq 2^{r-1}$
  $\mathcal{O}(|G|2^{2(p-r)-1}BMM(2^r)(r-1))$

Conclusion

- We present a modification of Valiant's algorithm
  - Layered submatrices processing
  - Effective utilization of parallel techniques and GPGPU
  - Applicability to the string-searching problem
- Future research
  - High-performance implementation (GPGPU, parallel techniques)
  - Evaluation on real-world data
  - Extension for more expressive classes of formal languages (conjunctive, boolean)

Contact Information

- Yuliya Susanina: jsusanina@gmail.com
- Anna Yaveyn: anya.ayveyn@yandex.ru
- Semyon Grigorev: semen.grigorev@jetbrains.com

Thanks!