PRISMA 7 (2024): 899-905
PRISMA, Prosiding Seminar Nasional Matematika
https://proceeding.unnes.ac.id/prisma
ISSN 2613-9189

Verifikasi Untai Dyck dengan *Pushdown Automata*

Ahmad Sabri[a*]

[a] *Universitas Gunadarma, Jl. Margonda Raya 100, Depok 16424, Indonesia*

[*] *Alamat surel: sabri@staff.gunadarma.ac.id*

Abstrak

Penelitian ini menerapkan *pushdown automata* untuk memverifikasi untai Dyck q-er, yang merupakan untai kombinatorial atas alfabet $\{0,1\}$ yang diawali oleh 1, selanjutnya pada sebarang proper prefiksnya berlaku banyak 0 tidak melebihi $(q-1)$ kali banyaknya 1, dan pada untai utuhnya banyak 0 adalah $(q-1)$ kali banyaknya 1. Metode *pushdown automata* ini dipilih karena memberikan kompleksitas waktu linier. Untuk memverifikasi untai Dyck biner $(q=2)$ secara lebih efisien dalam hal penggunaan memori, maka dilakukan redefinisi terhadap himpunan simbol pushdown/stack Γ, dan fungsi transisi δ. Hasil ini kemudian digeneralisasi dengan pendefinisian *pushdown automata* untuk untai Dyck q-er. Hal ini menghasilkan *pushdown automata* yang memiliki kompleksitas $O(n)$ dan penggunaan memori yang lebih sedikit dibandingkan pusdown automata yang telah ada sebelumnya.

Kata kunci: palindrom, palstars, *pushdown automata*, untai Dyck

1. Pendahuluan

Salah satu implementasi *pushdown automata* (PDA) adalah untuk memverifikasi untai dari bahasa bebas konteks (*context free grammar*). Knuth, Prat dan Morris (Knuth et al., 1977) menggagas penggunaan PDA untuk memverifikasi palindrom dan palstars (*palindrome stars*). Sejauh penelusuran, metode klasik ini memiliki kompleksitas $O(m+n)$, paling efisien dibandingkan metode lain dalam memverifikasi pola panjang m pada sebuah untai panjang n. Terinspirasi oleh penelitian Knuth dan rekannya tersebut, penelitian ini menerapkan PDA untuk memverifikasi untai Dyck q-er. Pemilihan kelas kombinatorial untai Dyck disebabkan kelas ini memiliki sifat yang mirip dengan palindrom, dalam hal komposisi simbol yang setara, dan penempatan simbol yang mengikuti aturan tertentu.

Pushdown automata untuk *balanced parentheses language* merupakan kajian yang banyak dibahas dalam literatur dan buku teks, antara lain oleh (Rich, 2007). Di sisi lain, untai Dyck biner berkorespondensi bijektif dengan *balanced parentheses language*. Oleh karena itu verifikasi untai Dyck dengan *pushdown automata* merupakan hal yang dapat dilakukan.

Pada literatur tersedia studi yang melimpah tentang untai Dyck (dan variannya) dalam berbagai tinjauan. Di antaranya *generating algorithm* dan kode Gray untuk untai Dyck dan Grand-Dyck (Vajnovszki, 2012), permutasi pada untai Dyck (Barnabei et al., 2016), analisa kemampuan recurent neural networks untuk mengenali gramar bebas konteks (Yu et al., 2019), dan varian untai Dyck simetris berganda (Cori et al., 2021), dan tentang pattern avoidance dan untai Dyck sebagai faktor dari untai Thue-Morse (Mol et al., 2023). Sejauh penelusuran, belum terdapat studi tentang verifikasi untai Dyck q-er menggunakan *pushdown automata*.

Penelitian ini memberikan redefinisi terhadap *existing* PDA untai Dyck biner (Rich, 2007), dengan dua tujuan:

1. Untuk memperkecil ukuran stack sehingga dapat menghemat memori.
2. Untuk memungkinkan dilakukannya generalisasi, sehingga dapat didefinisikan PDA untai Dyck q-er.

2. Definisi dan Landasan Teori

To cite this article:
Sabri, A. (2024). Verifikasi Untai Dyck dengan Pushdown Automata. *PRISMA, Prosiding Seminar Nasional Matematika 7*, 899-905.

Untai Dyck merupakan untai kombinatorial panjang $2n$ (genap) atas alfabet $\{0,1\}$ yang diawali oleh 1, selanjutnya pada sebarang proper prefiksnya berlaku banyak 0 tidak melebihi banyaknya 1, dan pada untai utuhnya banyak 0 dan 1 adalah sama. Kardinalitas untai Dyck panjang $2n$ diberikan oleh bilangan Catalan ke n (Vajnovszki, 2012). Generalisasi untai Dyck adalah untai Dyck q-er panjang qn atas alfabet $\{0,1\}$ yang diawali oleh 1, selanjutnya pada sebarang proper prefiksnya berlaku banyak 0 tidak melebihi $(q-1)$ kali banyaknya 1, dan pada untai utuhnya banyak 0 adalah $(q-1)$ kali banyaknya 1. Penyebutan untai Dyck tanpa disertai dengan eritasnya (q-er) merujuk pada untai Dyck biner $(q=2)$. Contoh untai Dyck biner dengan panjang 8 beserta balanced parentheses yang berkorespondensi ditampilkan pada Tabel 1. Dalam hal ini 1 mewakili simbol kurung buka '('dan 0 mewakili simbol kurung tutup ')'. Contoh untai Dyck terner $(q=3)$ diberikan pada Tabel 2.

Tabel 1. Untai Dyck biner dan balanced parantheses yang berkorespondensi.

Untai	Balanced parentheses
10101010	( )( )( )( )
10101100	( )( )(( ))
10111000	( )((( )))
10110100	( )(( )( ))
10110010	( )(( ))( )
11100010	((( )))( )
11100100	((( ))( ))
11101000	((( )( )))
11110000	(((( ))))
11010010	(( )( ))( )
11010100	(( )( )( ))
11011000	(( )(( )))
11001100	(( ))(( ))
11001010	(( ))( )( )

Tabel 2. Untai Dyck terner.

Untai
100100100
100101000
100110000
101100000
101010000
101001000
101000100
111000000
110100000
110010000
110001000
110000100

Berikut diberikan definisi *pushdown automata*.

Definisi 1. Sebuah *pushdown automata* M adalah sebuah 6-tupel $(Q, \Sigma, \Gamma, \delta, S, A)$ di mana:

- Q himpunan berhingga dari stata,
- Σ alfabet input,
- Γ alfabet stack,
- $q_0 \in Q$ stata awal,
- $F \subseteq Q$ himpunan stata penerima,
- δ fungsi transisi.

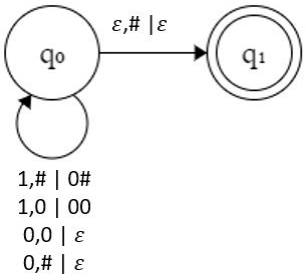
Stack dituliskan secara horizontal secara berurut di mana simbol paling kiri menyatakan stack teratas. Pada stack berlaku dua operasi yaitu pop dan push. Operasi pop menghapus elemen teratas, dan operasi push menambahkan elemen dengan cara mem-pop terlebih dahulu elemen teratas pada stack, kemudian menambahkan elemen yang di-push pada stack. Dead state adalah keadaan di stata non penerima di mana tidak tersedia fungsi transisi yang memetakan input ke stata berikut. Dalam hal ini, untai input tidak dapat diterima sebagai bahasa yang dikenal oleh automata.

Fungsi transisi didefinisikan sebagai pemetaan $\delta: Q \times \Sigma \times \Gamma \to Q \times \Gamma$. Parameter yang dipetakan mencakup stata awal, simbol input dan simbol teratas pada stack (yaitu simbol yang akan di-pop).. Hasil transisi mencakup stata berikut dan simbol yang di-push pada stack.

Definisi 2. PDA untai Dyck biner didefinisikan sebagai berikut (Rich, 2007):

- $Q = \{q_0, q_1\}$,
- $\Sigma = \{0,1\}$,
- $\Gamma = \{0, \#\}$,
- q_0 stata awal,
- q_1 stata penerima,
- $\delta =$ $\{((q_0, 1, \#)|(q_0, 0\#)), ((q_0, 1,0)|(q_0, 00)), ((q_0, 0,0)|(q_0, \varepsilon)), ((q_0, 0, \#)|(q_0, \varepsilon)), ((q_0, \varepsilon, \#)|(q_1, \varepsilon))\}$.

Diagram *pushdown automata* untuk untai Dyck biner diberikan oleh Gambar 1.



Gambar 1. PDA untuk untai Dyck biner.

Pada keadaan awal, stack hanya berisi simbol #. Ketika input awal (simbol 1) diterima, maka dilakukan pop terhadap #, setelahnya dilakukan push 0#, dan stata tetap pada q_0. Ini menandakan bahwa pada subuntai tersisa diharapkan terdapat sebuah simbol 0. Begitu seterusnya ketika input 1 diterima, maka dilakukan pop pada stack dan push 00 (setiap input 1 diterima, maka stack akan bertambah dengan sebuah simbol 0). Sebaliknya, jika input 0 diterima, maka dilakukan pop pada stack tanpa adanya push. Seterusnya ini akan mengurangi banyaknya 0 pada stack sampai akhirnya tersisa simbol #. Dalam keadaan ini, banyaknya input 1 dan 0 adalah sama. Jika tidak tersedia simbol yang diinput berikutnya, maka dilakukan input ε yang akan mem-pop simbol # dari stack, sehingga stack sepenuhnya kosong, dan stata beralih ke q_1 yang merupakan stata penerima.

Sebagai contoh, input untai 110100 pada PDA untai Dyck biner di atas memberikan alur sebagaimana ditunujkkan pada Tabel 3.

Tabel 3. Alur pada PDA untai Dyck biner untuk input 110100.

Stata Awal	Input	Stack	Stata Berikut
q_0		#	q_0
q_0	1	0#	q_0
q_0	1	00#	q_0
q_0	0	0#	q_0
q_0	1	00#	q_0
q_0	0	0#	q_0
q_0	0	#	q_0
q_0	ε	ε	q_1

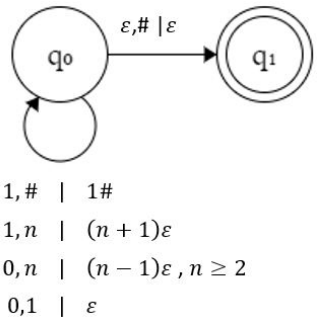
3. Pembahasan

Berikut diberikan redefinisi *pushdown automata* untai Dyck biner.

Definisi 3. *Pushdown automata* untai Dyck biner diberikan sebagai berikut

- $Q = \{q_0, q_1\}$,
- $\Sigma = \{0,1\}$,
- $\Gamma = \{\#,1,2,3, \ldots\}$,
- q_0 stata awal,
- q_1 stata penerima,
- $\delta =$ $\{((q_0, 1, \#)|(q_0, 1\#)), ((q_0, 1, n)|(q_0, (n+1)\varepsilon)), ((q_0, 0, n)|(q_0, (n-1)\varepsilon)), ((q_0, 0,1)|(q_0, \varepsilon)), ((q_0, \varepsilon, \#)|(q_1, \varepsilon))\}$.

Pada redefinisi ini, perubahan dilakukan terhadap Γ, yang berimplikasi terhadap perubahan δ. Pada definisi semula, Γ beranggotakan 0 dan #. Setiap simbol 0 di-push pada stack, maka stack akan ditambahkan dengan simbol 0 tersebut. Jika simbol 0 di-push berkali-kali, maka memori untuk menyimpan stack akan semakin besar. Untuk menghindari hal tersebut, maka simbol pada stack diredefinisikan sebagai "banyaknya 0 pada stack", sehingga Γ diredefinisikan beranggotakan integer $0,1,2, \ldots$. Dengan redefinisi ini, stack yang semula berisi $00\cdots0\#$ (dengan n simbol 0), dinyatakan sebagai $n\#$. Dengan cara ini maka penggunaan memori dapat dikurangi. Diagram automata PDA untai Dyck biner diberikan pada Gambar 2.



Gambar 2. Redefinisi PDA untuk Dyck biner.

Pada keadaan awal, stack berisi simbol #. Ketika input awal (yaitu 1) diterima, maka stack diperbaharui menjadi 1#. Ini bermakna bahwa PDA telah menerima satu kali input 1, dan diharapkan di subuntai tersisa terdapat simbol 0 sebanyak 1. Oleh karena banyak 1 dan 0 belum sama, maka stata berikut tetap pada q_0 untuk menerima input berikutnya.

Aturan selanjutnya adalah jika top stack adalah n, maka setiap input simbol 1 akan mem-pop n dari stack dan mem-push $n+1$ pada stack. Hal ini bermakna bahwa PDA telah menerima input 1 sebanyak $n+1$ kali, dan diharapkan di subuntai tersisa terdapat simbol 0 sebanyak $n+1$. Oleh karena banyak 1 dan 0 belum sama, stata berikut tetap pada q_0 untuk menerima input berikutnya.

Jika input 0 diterima, maka dilakukan pop terhadap n dan push $n-1$ (berlaku untuk $n \geq 2$). Hal ini bermakna bahwa input 0 yang diharapkan dari subuntai tersisa adalah sebanyak $n-1$. Oleh karena banyak 1 dan 0 belum sama, stata berikut tetap pada q_0 untuk menerima input berikutnya.

Jika pop-value pada stack adalah 1, maka hanya dilakukan pop dan tidak dilakukan push, karena dalam hal ini PDA telah menerima input 1 dan 0 yang sama banyaknya. Dalam keadaan ini, stata berikut tetap pada q_0 untuk menerima input berikutnya, dan stack hanya berisi #. Oleh karena stack kembali pada keadaan semula, maka input berikutnya yang diizinkan adalah 1 atau ε. Jika input 0 diterima, maka untai ditolak sebagai untai Dyck.

Tabel 4 menampilkan alur pada PDA untuk untai 110100 yang berakhir pada final state q_1, sedangkan Tabel 5 menampilkan alur pada PDA untuk untai 100110, yang berakhir pada dead state sehingga tidak diterima. Walaupun untai 100110 memuat 1 dan 0 yang sama banyaknya, namun prefiks 100 melanggar syarat banyak 0 tidak melebihi banyaknya 1.

Tabel 4. Alur pada PDA untai Dyck biner untuk input 110100.

Stata Awal	Input	Stack	Stata Berikut
q_0		#	q_0
q_0	1	1#	q_0
q_0	1	2#	q_0
q_0	0	1#	q_0
q_0	1	2#	q_0
q_0	0	1#	q_0
q_0	0	#	q_0
q_0	ε	ε	q_1

Tabel 5. Alur pada PDA untai Dyck biner untuk input 100110.

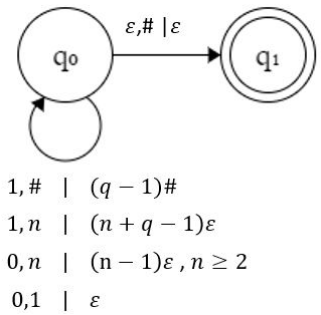
Stata Awal	Input	Stack	Stata Berikut
q_0		#	q_0
q_0	1	1#	q_0
q_0	0	#	q_0
q_0	0		(dead state)

Pembahasan berikut adalah PDA untai Dyck q-er panjang qn, yang merupakan generalisasi dari PDA untai Dyck biner.

Definisi 5. *Pushdown automata* untuk untai Dyck q-er diberikan sebagai berikut:

- $Q = \{q_0, q_1\}$,
- $\Sigma = \{0,1\}$,
- $\Gamma = \{\#,1,2,3, \ldots\}$,
- q_0 stata awal,
- q_1 stata penerima,
- $\delta =$ $\{((q_0, 1, \#)|(q_0, 1\#)), ((q_0, 1, n)|(q_0, (n+1)\varepsilon)), ((q_0, 0, n)|(q_0, (n-1)\varepsilon)), ((q_0, 0,1)|(q_0, \varepsilon)), ((q_0, \varepsilon, \#)|(q_1, \varepsilon))\}$.

Diagram *pushdown automata* untai Dyck q-er panjang qn diberikan oleh Gambar 3.



Gambar 3. PDA untuk Dyck q-er.

Generalisasi dilakukan pada fungsi transisi. Input awal 1 mem-push $q-1$ ke stack. Hal ini karena perbandingan banyak simbol 1 terhadap banyak simbol 0 adalah $1:q-1$. Oleh karena itu diharapkan dalam subuntai tersisa terdapat simbol 0 sebanyak $q-1$. Oleh karena banyak 1 dan 0 belum memenuhi rasio $1:q-1$, maka stata berikut tetap pada q_0 untuk menerima input berikutnya.

Selanjutnya, setiap simbol 1 diterima, maka stack akan mem-pop n, dan mem-push $n+q-1$. Hal ini berarti banyaknya 0 yang diharapkan sebelumnya, yaitu n, bertambah sebanyak $q-1$ akibat diterimanya input 1 yang baru. Oleh karena banyak 1 dan 0 belum memenuhi rasio $1:q-1$, maka stata berikut tetap pada q_0 untuk menerima input berikutnya.

Sebaliknya jika input 0 diterima, maka stack akan mem-pop $n \geq 2$, yaitu banyaknya 0 yang diharapkan sebelumnya, dan mem-push $n-1$ sebagai akibat diterimanya input 0 tadi. Oleh karena banyak 1 dan 0 belum memenuhi rasio $1:q-1$, maka stata berikut tetap pada q_0 untuk menerima input berikutnya.

Jika $n=1$, maka dilakukan pop namun tidak ada nilai yang di-push ke stack. Dalam hal ini, keadaan di mana banyaknya 0 adalah $q-1$ kali banyaknya 1 telah terpenuhi. Dalam keadaan ini, stata berikut tetap pada q_0 untuk menerima input berikutnya, dan stack hanya berisi #. Oleh karena stack kembali pada keadaan semula, maka input berikutnya yang diizinkan adalah 1 atau ε. Jika input 0 diterima, maka untai ditolak sebagai untai Dyck.

Tabel 6 menampilkan alur pada PDA untuk untai Dyck 3-er (terner) 101000 yang berakhir pada final state q_1, sedangkan Tabel 7 menampilkan alur pada PDA untuk untai 100010, yang berakhir pada dead state sehingga untai tidak diterima sebagai untai Dyck terner. Walaupun untai 100110 memuat 1 dan 0 yang sama banyaknya, namun prefiks 100 melanggar syarat banyak 0 tidak melebihi banyaknya 1.

Tabel 6. Alur pada PDA untai Dyck terner untuk input 101000.

Stata Awal	Input	Stack	Stata Berikut
q_0		#	q_0
q_0	1	2#	q_0
q_0	0	1#	q_0
q_0	1	3#	q_0
q_0	0	2#	q_0
q_0	0	1#	q_0
q_0	0	#	q_0
q_0	ε	ε	q_1

Tabel 7. Alur pada PDA untai Dyck terner untuk input 101000.

Stata Awal	Input	Stack	Stata Berikut
q_0		#	q_0
q_0	1	2#	q_0
q_0	0	1#	q_0
q_0	0	#	q_0
q_0	0	Dead state	

4. Simpulan

Penelitian ini memberikan redefinisi terhadap *pushdown automata* untai Dyck biner panjang $2n$, dan generalisasinya untuk untai Dyck q-er. Metode ini memberikan kompleksitas waktu linier $O(qn)$. Novelty dari *pushdown automata* yang diusulkan dalam penelitian ini adalah penggunaaan memori yang

lebih sedikit untuk menyimpan stack dibandingkan dengan existing *pushdown automata*. Hal ini mengakibatkan verifikasi untai yang panjang dilakukan secara lebih efisien.

Untuk penelitian lanjutan, model ini dapat dikembangkan untuk memverifikasi untai Grand Dyck q-er, yang didefinisikan sebagaimana untai Dyck tanpa menyertakan batasan pada proper prefiksnya. Selain itu dapat dibangun sebuah algoritma untuk mengimplementasikan *pushdown automata* yang didefinisikan pada penelitian ini.

Daftar Pustaka

Barnabei, M., Bonetti, F., Castronuovo, N., & Cori, R. (2016). Some permutations on Dyck words. *Theoretical Computer Science*, *635*, 51–63. https://doi.org/10.1016/j.tcs.2016.05.007

Cori, R., Frosini, A., Palma, G., Pergola, E., & Rinaldi, S. (2021). On doubly symmetric Dyck words. *Theoretical Computer Science*, *896*, 79–97. https://doi.org/10.1016/j.tcs.2021.10.006

Knuth, D. E., Morris, Jr. , J. H., & Pratt, V. R. (1977). Fast Pattern Matching in Strings. *SIAM Journal on Computing*, *6*(2), 323–350. https://doi.org/10.1137/0206024

Mol, L., Rampersad, N., & Shallit, J. (2023). *Dyck Words, Pattern Avoidance, and Automatic Sequences* (pp. 220–232). https://doi.org/10.1007/978-3-031-33180-0_17

Rich, E. (2007). *Automata, Computability and Complexity: Theory and Applications*. Prentice Hall.

Vajnovszki, V. (2012). ECO-based Gray codes generation for particular classes of words. *GASCom 2012*.

Yu, X., Vu, N. T., & Kuhn, J. (2019). Learning the Dyck Language with Attention-based Seq2Seq Models. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 138–146. https://doi.org/10.18653/v1/W19-4815