Optimizing Sanskrit Antyakshari with Directed Graphs: A Competitive and Strategic Tool for Enhanced Play

Shankararama Sharma
Vyoma Linguistic Labs Foundation
Bengaluru - 560040
rama.vyoma@gmail.com

Abstract

Antyākṣarī, a popular word-linking game in Bhārat, has traditionally been played using songs, but its application to Sanskrit verses offers a structured and competitive dimension. This paper presents a novel approach to modelling the game using directed graphs, enabling a strategic analysis of gameplay. We introduce two rule sets—Rule Set A, which follows conventional linking constraints, and Rule Set B, which introduces vowel-based continuity for enhanced playability.

We construct a Directed Multigraph (DMG) to represent verse transitions and analyse key properties such as source nodes, sink nodes, cycles, isolated nodes, and longest paths. Using data from two Sanskrit scriptures—Bhagavad-Gītā and Śrīman-Nārāyaṇīyam—we compute optimal verse sequences for extended gameplay and explore strategies for competitive advantage.

Our findings demonstrate that Rule Set B significantly increases playable sequences, while high-degree nodes can be leveraged for maximizing difficulty. This graph-based formulation also enables the creation of an AI-assisted Antyākṣarī interface, allowing users to play, practice, and strategize with dynamic rule customizations. This work lays the foundation for future computational Sanskrit applications, modernizing the game while preserving its rich poetic traditions.

1 Introduction

Antyākṣarī is a well-known game across Bhārat. It is an extremely engaging activity that provides both intellectual stimulation and an educational experience. The very name *Antyākṣarī* (also referred to as *Antākṣarī* in some regional languages) encapsulates the core concept of the game through its etymology: *Antya* means "last", and *akṣarī* means "having the letter". The game is fundamentally based on the principle of continuing with the last letter of a given verse.

The most popular contemporary form of Antyākṣarī is played with songs, particularly those from films. However, a less common but equally—if not more—exhilarating variation involves Sanskrit verses. The rules of this version are simple and precise:

- The last *Gunitākṣara* (a letter with an associated vowel) of a verse must be used to begin the next verse.
- No verse may be repeated.

We shall refer to this as "Rule Set A".
For example, consider the well-known verse:

गुरुर्ब्रह्मा गुरुर्विष्णुः गुरुर्देवो महेश्वरः।
गुरुः साक्षात् परं ब्रह्म तस्मै श्रीगुरवे नमः॥

Here, the last *Gunitākṣara* is 'मः'. Hence, the next verse must begin with 'म्', which can be any *Gunitākṣara* such as म, मा, मि, मौ, मं, मः,or even a *Samyuktākṣara* (conjunct consonant) like म्र, म्लु,etc.
If the verse ends with a consonant in its halanta (without an inherent vowel) form, such as:

कराग्रे वसते लक्ष्मीः करमध्ये सरस्वती ।
करमूले स्थिता गौरी प्रभाते करदर्शनम् ॥

then we consider the last pronounced *Gunitākṣara*, which is "न" in this case. The next verse should begin with any *Gunitākṣara* or *Samyuktākṣara* of "न".

Although this is the fundamental rule, there can be modifications, variations, and exceptions depending on the playing context. One such modification arises from the rarity of verses beginning with certain letters, particularly those from the *ṭa-varga* (ट, ठ, ड, ढ, ण), as well as थ, ख, and ष. In the two scriptures under consideration, there are no known verses that begin with these akṣaras. Therefore, we introduce an extension to "Rule Set A":

- If no extant verse begins with the last akṣara, then the game continues by using the svara (vowel) associated with that akṣara.

This modified rule set shall be referred to as "Rule Set B".
This addition has two significant advantages:

1. It allows the game to continue when a verse ends with an akṣara that has no known continuation (a scenario that occurs frequently).

2. It brings into play verses that begin with svaras, which would otherwise not be usable in the game.

The tradition of memorizing verses is deeply ingrained in the Bharatiya Parampara (Indian tradition). Sanskrit, in particular, boasts an extensive corpus of texts across diverse genres, many of which are composed in poetic form. For those accustomed to this memorization practice, Antyākṣarī provides a source of enjoyment while also serving as a tool for reinforcing and indexing memorized verses.

2 Methodology

The structure of Antyākṣarī naturally aligns with the mathematical framework of graphs. Despite its widespread popularity and recognition as an effective tool for language learning and teaching,(Saha, 2006)(Sinha, 2017) there has been no formal mathematical formulation or published study analysing its structural properties. This paper aims to bridge that gap by modelling the game using graph theory, enabling a systematic exploration of its mechanics and strategic possibilities.

2.1 Definition of Graphs and Directed Graphs

A **graph** is a mathematical structure used to model pairwise relationships between objects. Formally, a graph G is defined as an ordered pair:

$$G = (V, E)$$

where:

- V is a finite, non-empty set of *vertices* (also called *nodes*).

- $E \subseteq \{\{u, v\} \mid u, v \in V\}$ is a set of *edges*, where each edge is an unordered pair of distinct vertices.

A graph is called a **simple graph** if it has neither multiple edges nor loops (edges that connect a vertex to itself). If multiple edges or loops are allowed, the graph is referred to as a **multigraph**.

A **directed graph** (or **digraph**) is a graph where the edges have a direction associated with them. Formally, a directed graph G is defined as:

$$G = (V, E)$$

where:

- V is a finite, non-empty set of vertices.

- $E \subseteq V \times V$ is a set of *directed edges* (also called *arcs*), where each edge is an ordered pair (u, v), indicating a directed connection from vertex u to vertex v.

A directed graph is called a **simple directed graph** if it has no multiple arcs or loops. If multiple arcs or loops are allowed, it is referred to as a **directed multigraph**.

These definitions provide the fundamental basis for graph representation and analysis in various applications. We will now use directed multigraphs to analyze the Antyākṣarī game.

2.2 Application to Antyākṣarī

In this paper, we restrict our analysis to two scriptures:

1. **Bhagavad-Gītā**, from the Mahābhārata.(Project, 2021)
2. **Śrīman-Nārāyaṇīyam**, authored by Melpathur Nārāyaṇa Bhaṭṭatiri.(Murarka, 2019)

However, the methodology can be extended to any poetic text, provided the requisite data is available. The **nodes** of our *Directed Multigraph (DMG)* contain:

- The first *akṣara* of the verse.
- The last complete *akṣara* of the verse.
- The last *svara* (vowel sound) of the verse.

First akṣara means, the vyañjana or the svara with which the verse begins. vyañjana-s everywhere are represented without the halanta for ease of reading. Last complete akṣara means the last vyañjana in the last Guṇitākśara of the verse.

A representation of this structure is illustrated in Figure 1 and Figure 2 using a small set of verses from the Bhagavad-Gītā. A similar representation of some verses from Śrīman-Nārāyaṇīyam is illustrated in Figure 3

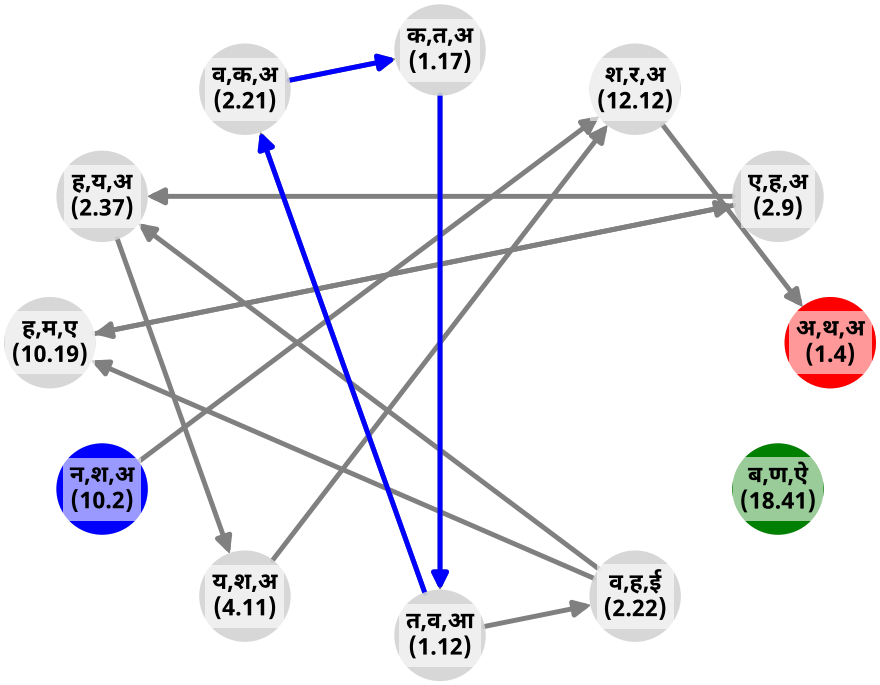

Figure 1: Example of Directed Multigraph (DMG) with source, sink, cycle, isolated nodes for a few verses from Bhagavad-Gītā. The first letter, last letter and associated swara are given in each node.

2.3 Graph Theoretical Properties of Antyākṣarī

Source Node: A **source node**, shown in blue, is a node with only outgoing edges and no incoming edges. Formally, a node v is a source if:

$$\forall u \in V, (u, v) \notin E \quad \text{and} \quad \exists (v, u) \in E.$$

This means that the corresponding verse cannot be reached by any other verse—i.e., no other verse ends with the starting letter of this verse.

Sink Node: A **sink node**, marked in red, is a node with only incoming edges and no outgoing edges. Formally, a node v is a sink if:

$$\forall u \in V, (v, u) \notin E \quad \text{and} \quad \exists (u, v) \in E.$$

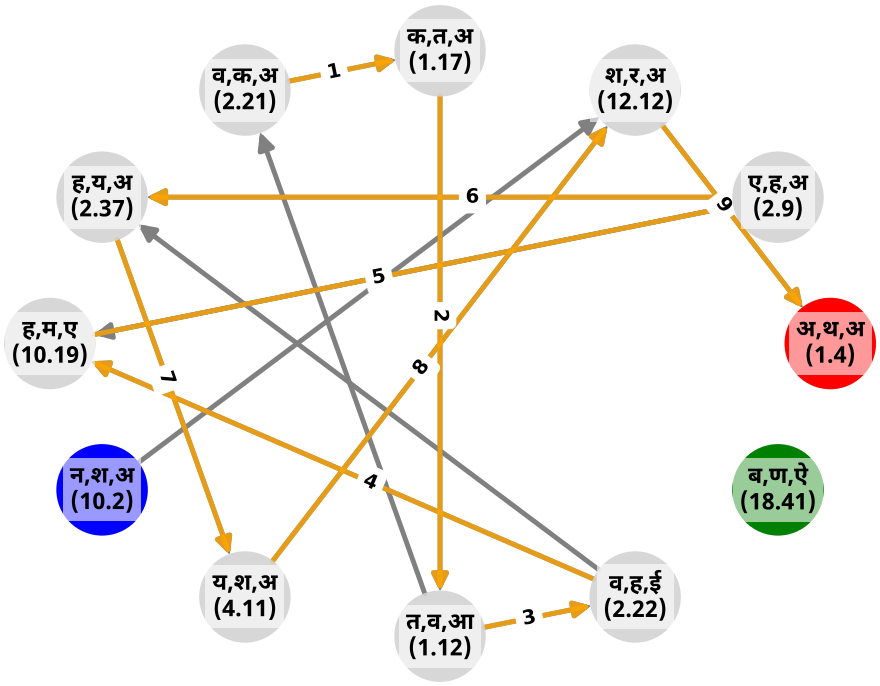

Figure 2: The longest path for a few verses from Bhagavad-Gītā. The numbers in the brackets represent the chapter and verse.

This indicates that the corresponding verse can be reached by others but does not lead to any further verse—i.e., no other verse begins with the last letter of this verse.

Degree of a Node: The *degree* of a node quantifies its connectivity and is classified as follows:

- **In-degree** ($d^-(v)$): The number of edges directed *into* node v. Formally,

$$d^-(v) = |\{u \in V \mid (u, v) \in E\}|$$

- **Out-degree** ($d^+(v)$): The number of edges directed *out of* node v. Formally,

$$d^+(v) = |\{u \in V \mid (v, u) \in E\}|$$

- **Total degree** ($d(v)$): The sum of in-degree and out-degree.

$$d(v) = d^-(v) + d^+(v)$$

Cycle: A **cycle**, represented by blue-colored edges, is a closed sequence of edges where the starting and ending nodes are identical, and each intermediate node appears at most once. Formally, a cycle is a sequence of nodes $v_1, v_2, \ldots, v_k$ where:

$$(v_i, v_{i+1}) \in E \quad \text{for } i = 1 \text{ to } k-1, \quad \text{and} \quad (v_k, v_1) \in E.$$

Cycles result in repetition, which must be resolved to ensure progression in Antyākṣarī, as verses cannot be repeated.

Isolated Node: An **isolated node**, depicted in green, has neither incoming nor outgoing edges. Formally, a node v is isolated if:

$$\forall u \in V, (v, u) \notin E \quad \text{and} \quad (u, v) \notin E.$$

This means that the corresponding verse will never be used in Antyākṣarī since no other verse connects to or from it under the given rules.

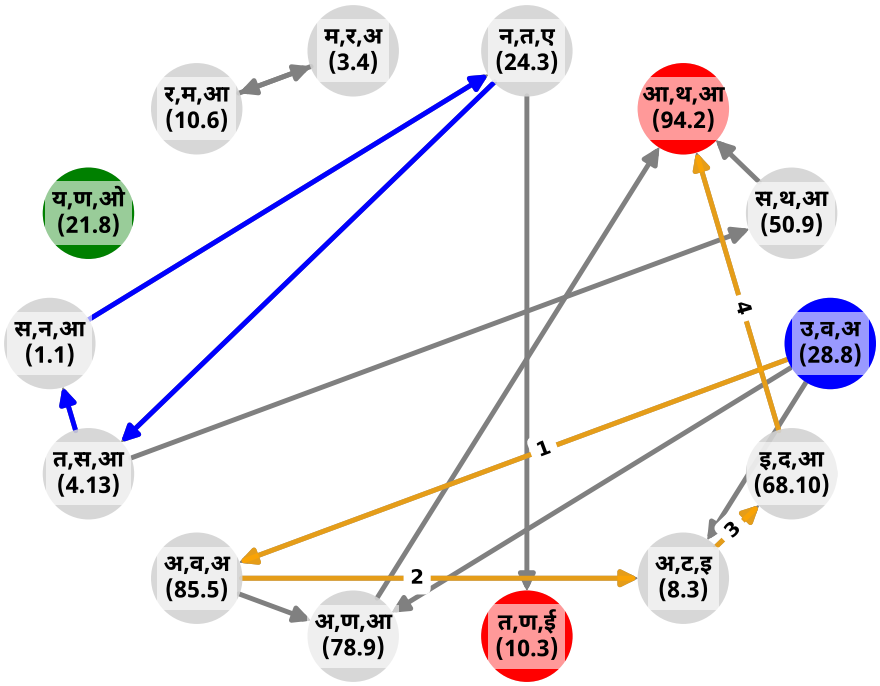

Figure 3: A DMG with all components for a few verses from Śrīman-Nārāyaṇīyam

Longest Path: The **longest path**, highlighted in orange in Figure 2 and Figure 3, represents the longest sequence of edges forming a simple directed path, where no node is visited more than once. Mathematically, it is the maximum-length sequence:

$$(v_1, v_2), (v_2, v_3), \ldots, (v_k, v_{k+1})$$

such that no vertex appears more than once. The edges are numbered to indicate the sequence of traversal, translating to the maximum number of consecutive verses that can be played in Antyākṣarī without repetition.

3 Results and Analysis on the Bhagavad-Gītā and Nārāyaṇīyam

We now apply the definitions and methodology outlined above to analyze the Bhagavad-Gītā and Śrīman-Nārāyaṇīyam both separately and together. Figures 1, 2, and 3 illustrate how the graph representation is constructed when verses are connected. These figures provide a sample representation of the structure.

Figures 1 and 2 depict verses from the Bhagavad-Gītā. The numbers within brackets in each node represent the chapter and verse numbers. Since the cycle and longest path overlap, they have been shown in two separate figures for clarity—cycles are highlighted in blue in Figure 1, while the longest path is shown in orange in Figure 2.

Figure 3 presents a subset of verses from the Nārāyaṇīyam, where the numbers in brackets within the nodes denote Dashakam and verse numbers. The Nārāyaṇīyam consists of 100 Dashakams. Since the cycle and longest path do not overlap for this subset, they are visualized within a single figure.

3.1 Computational Complexity and Algorithmic Analysis

Finding the Longest Simple Path (a path without repeated vertices) in a general directed graph with cycles is known to be NP-hard. In the context of Antyākṣarī, the objective is to find the maximum number of unique verses that can be played without repetition. Because our graph contains cycles, the DFS-based approach with memoization serves as a heuristic to find a high-quality sequence. ‘

- **Complexity Analysis:** The implementation exhibits a time complexity of $O(V^2 + E)$, where V is the number of nodes (verses) and E is the number of directed edges (transitions). The graph traversal component is $O(V + E)$, while the memoization step involves path-copying that, in the worst-case scenario of a linear chain, results in an $O(V^2)$ cost.

- **Heuristic Nature:** Given the presence of cycles in the Sanskrit Antyākṣarī graph, this algorithm is a heuristic that identifies the longest path found during the search space exploration. It should be noted that the reported results (234 for Bhagavad-Gītā and 455 for Nārāyaṇīyam) represent lower bounds of the possible gameplay length rather than proven global optima.
- **Empirical Performance:** For the combined dataset of 1,737 verses, the algorithm converges rapidly(in 0.015s for the whole dataset of 1734 verses). This efficiency is largely due to the structural sparsity of the graph, where the average out-degree is low and the search is frequently pruned by sink nodes (96 in the combined dataset under Rule Set A) and isolated nodes (25 in the combined dataset).

3.2 Longest Path Using Rule Set A

Analyzing the longest path allows us to determine the maximum number of consecutive verses that can be played in Antyākṣarī without repetition. Knowledge of this path enables players to extend the game duration optimally.

3.2.1 Longest Path in the Bhagavad-Gītā

For the Bhagavad-Gītā, using Rule Set A, the longest path comprises 234 verses, forming the following sequence:

$$10.5 \to 1.1 \to 1.22 \to 2.14 \to 1.12 \to 2.21 \to 1.17 \to \cdots \to 7.8$$

It begins with Chapter 10, Verse 5:

अहिंसा समता तुष्टिस्तपो दानं यशोऽयशः।
भवन्ति भावा भूतानां मत्त एव पृथग्विधाः ॥

and ends with Chapter 7, Verse 8:

रसोऽहमप्सु कौन्तेय प्रभास्मि शशिसूर्ययोः।
प्रणवः सर्ववेदेषु शब्दः खे पौरुषं नृषु ॥

3.2.2 Longest Path in the Nārāyaṇīyam

For the Nārāyaṇīyam, using Rule Set A, the longest path consists of 455 verses, forming the following sequence:

$$1.10 \to 1.1 \to 1.4 \to 3.4 \to 10.6 \to 3.9 \to \cdots \to 57.1$$

It begins with Dashakam 1, Verse 10:

ऐश्वर्यं शङ्करादीश्वरविनियमनं विश्वतेजोहराणां
तेजस्संहारि वीर्यं विमलमपि यशो निस्पृहैश्चोपगीतम् ।
अङ्गासङ्गा सदा श्रीरखिलविदसि न क्वापि ते सङ्गवार्ता
तद्वातागारवासिन् मुरहर भगवच्छब्दमुख्याश्रयोऽसि ॥

and ends with Dashakam 57, Verse 1:

रामसखः क्वापि दिने कामद भगवन् गतो भवान् विपिनम् ।
सूनुभिरपि गोपानां धेनुभिरभिसंवृतो लसद्वेषः ॥

3.2.3 Longest Path in the Combined Scriptures

When combining both scriptures and applying Rule Set A, the longest path consists of 783 verses out of the total 1737 verses available in both texts. Due to space constraints, we do not list all verse numbers here.

3.3 Longest Path Using Rule Set B

As expected, Rule Set B results in longer paths than Rule Set A, as the relaxed rules allow for more connections, thereby extending the game duration.

- Bhagavad-Gītā: The longest path under Rule Set B contains 316 verses, starting from 1.1 and ending at 11.23.
- Nārāyaṇīyam: The longest path under Rule Set B contains 563 verses, starting from 79.1 and ending at 57.1.
- Combined Scriptures: The longest path under Rule Set B spans 976 verses.

3.4 Sink, Source, and Isolated Nodes Using Rule Set A and Rule Set B

In this section, we analyze the number of sink, source, and isolated nodes in the Bhagavad-Gītā, Nārāyaṇīyam, and their combined dataset under Rule Set A and Rule Set B.

A sink node is a node that has incoming edges but no outgoing edges. These nodes represent verses that can be reached but do not lead to any further verses in the game.

A source node is a node that has outgoing edges but no incoming edges. These nodes represent verses that start new sequences in the game.

An isolated node is a node with neither incoming nor outgoing edges, meaning that the corresponding verse is entirely disconnected from the game.

3.4.1 Sink Nodes

Table 1 presents the number of sink nodes for the Bhagavad-Gītā, Nārāyaṇīyam, and the combined dataset under both Rule Set A and Rule Set B. Under Rule Set A, a notable number of sink nodes exist due to the rarity of certain akṣaras appearing at the beginning of verses. However, Rule Set B significantly reduces the number of sink nodes by allowing connections via vowels.

Table 1: Number of Sink Nodes Under Rule Set A and B

Dataset	Rule Set A	Rule Set B
Bhagavad-Gītā	48 (ख, घ, ढ, ण, थ, ष)	3 (ण)
Nārāyaṇīyam	59 (ख, ञ, ट, ठ, ड, ढ, ण, थ)	2 (ण)
Combined	96 (ख, ड, ट, ढ, थ, ञ, ण, ठ)	0

3.4.2 Source Nodes

Table 2 shows the number of source nodes under both rule sets. Under Rule Set A, there are more source nodes, indicating the presence of verses that initiate new sequences. However, under Rule Set B, the number of source nodes decreases, as more connections become possible.

Table 2: Number of Source Nodes Under Rule Set A and B

Dataset	Rule Set A	Rule Set B
Bhagavad-Gītā	172 (अ, आ, इ, ई, उ, ऊ, ऋ, ए, ओ, ब)	39 (ए, ऊ, ब, ऋ)
Nārāyaṇīyam	192 (अ, आ, इ, ई, उ, ऊ, ए, ऐ, औ, झ, फ, ब)	23 (ब, औ, फ, ऊ, झ)
Combined	367 (ए, ब, ऐ, फ, औ, उ, आ, ऋ, ऊ, इ, अ, ओ, झ, ई)	42 (ब, फ, ऊ, ऋ, झ)

3.4.3 Isolated Nodes

Table 3 presents the number of isolated nodes. These nodes are completely disconnected, meaning their corresponding verses cannot be played in the game.

Table 3: Number of Isolated Nodes Under Rule Set A and B

Dataset	Rule Set A	Rule Set B
Bhagavad-Gītā	14	1 (ब, ण)
Nārāyaṇīyam	14	0
Combined	25	0

The singular verse that is isolated under Rule Set B is 18.41:

ब्राह्मणक्षत्रियविशां शूद्राणां च परंतप ।
कर्माणि प्रविभक्तानि स्वभावप्रभवैर्गुणैः ॥

3.4.4 Observations

From the data above, we observe the following:

- Rule Set A has a significantly higher number of sink nodes, as it does not permit continuation through vowel-based transitions.
- Rule Set B reduces sink nodes to nearly zero, demonstrating its flexibility in allowing more connections.
- The number of source nodes is higher under Rule Set A, as fewer verses can link back to them.
- Isolated nodes are significantly reduced under Rule Set B, as additional linking possibilities make complete disconnection rare.

This analysis provides insight into how the game can be prolonged by using different rule sets and highlights how certain akṣaras tend to restrict connectivity in the Antyākṣarī graph.

3.5 Shortest Path to a Sink Node Given a Starting Akṣara

In the context of Antyākṣarī, identifying the shortest path to a sink node is useful when trying to make it difficult for opponents to continue the game. If a player strategically selects a verse leading to a sink node, they can potentially end the opponent's turn quickly.

We compute the shortest path to a sink node for each akṣara under both Rule Set A and Rule Set B. However, in some cases, no shortest path exists due to:

1. The absence of any verse beginning with the given akṣara.
2. The lack of a path leading to a sink node from the given akṣara.

These cases have been excluded from our analysis.

3.5.1 Rule Set A

For the Bhagavad-Gītā, there are 48 sink nodes (as identified earlier). The shortest path to a sink node is always either 0 or 1 hop away.

For instance, if we start with the letter 'द', the shortest path is:

1.18 (Sink)

द्रुपदो द्रौपदेयाश्च सर्वशः पृथिवीपते।
सौभद्रश्च महाबाहुः शङ्खान्दध्मुः पृथक्पृथक् ॥

For the Nārāyaṇīyam, there are 59 sink nodes. The shortest path to a sink node is also 0 or 1 hop away, except for one case where it takes 2 hops:

$32.7 \rightarrow 4.6 \rightarrow 1.6$ (Sink)

3.5.2 Rule Set B

Under Rule Set B, the shortest path is at most 2 hops away for any starting letter.

For example, in the Bhagavad-Gītā:

- Starting with 'च': $4.13 \rightarrow 2.42 \rightarrow 3.5$ (Sink)
- Starting with 'आ': $2.70 \rightarrow 18.58 \rightarrow 5.27$ (Sink)

न हि कश्चित्क्षणमपि जातु तिष्ठत्यकर्मकृत् ।
कार्यते ह्यवशः कर्म सर्वः प्रकृतिजैर्गुणैः ॥

स्पर्शान्कृत्वा बहिर्बाह्यांश्चक्षुश्चैवान्तरे भ्रुवोः ।
प्राणापानौ समौ कृत्वा नासाभ्यन्तरचारिणौ ॥

For the Nārāyaṇīyam, the shortest path is always at most 2 hops away:

- Starting with 'ष': $23.3 \rightarrow 33.5 \rightarrow 21.8$ (Sink)
- Starting with 'फ': $55.5 \rightarrow 27.7 \rightarrow 8.13$ (Sink)

अस्मिन् परात्मन् ननु पाद्मकल्पेत्वमित्थमुत्थापितपद्मयोननिः ।
अनन्तभूमा मम रोगराशिंनिरुन्धि वातालयवास विष्णो ॥

याम्यां दिशं भजति किंपुरुषाख्यवर्षेसंसेवितो हनुमता दृढभक्तिभाजा ।
सीताभिरामपरमाद्भुतरूपशालीरामात्मकः परिलसन्परिपाहि विष्णो ॥

Under Rule Set B, the number of sink nodes reduces drastically:

- Bhagavad-Gītā: 3 sink nodes.
- Nārāyaṇīyam: 2 sink nodes.
- Combined dataset: No sink nodes, hence no shortest path exists.

3.6 Degrees of Nodes and Most Frequent Akṣaras

The degree of a node provides insights into how frequently an akṣara is used as the first or last letter of a verse.

In-degree corresponds to the most frequent last akṣara. Out-degree corresponds to the most frequent first akṣara. Total degree represents the sum of in-degree and out-degree.

Table 4 presents these values for both Rule Set A and Rule Set B.

Table 4: Node Degree Analysis Under Rule Set A and B

Dataset	**Most Frequent First Letter**	**Most Frequent Last Letter**	**Highest Total Degree**
Rule Set A			
Bhagavad-Gītā	(य, 103)	(त, 232)	335 (all verses beginning with य and ending with त)
Nārāyaṇīyam	(त, 164)	(त, 155)	317 (all verses beginning and ending with त)
Combined	(त, 217)	(त, 387)	602 (all verses beginning and ending with त)
Rule Set B			
Bhagavad-Gītā	(य, 103)	(त, 232)	335 (same as Rule Set A)
Nārāyaṇīyam	(त, 164)	(त, 155)	317 (same as Rule Set A)
Combined	(त, 217)	(त, 387)	602 (same as Rule Set A)

3.6.1 Observations and Game Strategy

The data shows that Rule Set B does not change the frequency distribution of first and last akṣaras, meaning the in-degree, out-degree, and total degree remain the same as in Rule Set A.

Game Strategy: Choosing high-degree nodes first makes the game progressively difficult for opponents since fewer connections remain available. Since 'त' appears frequently as both the first and last letter in the combined dataset, players may leverage this to control game flow.

4 Discussion

The graph-based formulation of Antyākṣarī presented in this study can serve as the foundation for an interactive interface that allows users to play, practice, and develop strategies for the game. This

paper provides a proof-of-concept (PoC) demonstrating how the ubiquitous Antyākṣarī game can be systematically analyzed using graph theory.

Given the vast corpus of poetry available in Sanskrit literature, this approach can facilitate the development of an intelligent Antyākṣarī-playing system. The interface would allow players to engage with the game dynamically, where each verse recited by the player and the software would result in the corresponding graph nodes being deleted in real-time. This mechanism ensures that no verse is repeated, maintaining adherence to game rules.

Furthermore, an adaptive gameplay feature could be implemented where users can upload their own database of verses, enabling them to play against the computer or challenge others in a structured manner.

Advanced Antyākṣarī rules, commonly used in competitions, introduce additional constraints such as:

- Playing only with specific metres (chandas).
- Excluding certain metres or verse structures.
- Avoiding popular or well-known verses to increase difficulty.

These rules are introduced in competitive settings to make rounds progressively more challenging. The proposed interface could be designed to support such customizable rule sets, thereby enhancing the user experience and game complexity.

This formulation bears resemblance to chess engines, which, although significantly more powerful than human players, serve as valuable tools for strategizing and improving gameplay in tournaments. While Antyākṣarī does not involve the same computational complexity as chess, an AI-driven Antyākṣarī system could help players refine their strategies, learn new verses, and optimize gameplay choices.

Overall, this approach not only preserves the cultural and linguistic heritage embedded in Sanskrit poetry but also modernizes the game by integrating technology for learning and entertainment.

5 Conclusion

This study has demonstrated how the game of Antyākṣarī can be mathematically modeled using directed graphs, transforming it into a strategic and analytical framework. By leveraging graph theoretical properties, we analyzed how Sanskrit verses connect in gameplay and identified optimal longest paths, shortest paths, and node properties.

Our key findings show that:

- Rule Set A results in more restrictive gameplay, often leading to dead ends due to the lack of available verses beginning with certain akṣaras.
- Rule Set B significantly extends gameplay by allowing vowel-based transitions, reducing the number of sink nodes and increasing connectivity.
- High-degree nodes (verses with frequently occurring first and last akṣaras) can be strategically used to prolong or hinder gameplay.
- The shortest path to a sink node reveals effective ways to challenge opponents, particularly in competitive settings.

Beyond theoretical analysis, this formulation has practical applications for developing interactive Antyākṣarī platforms. A graph-based AI-driven system could enhance gameplay by:

- Allowing users to upload their own verse database.
- Enabling custom rule configurations (e.g., specific meters, verse exclusions).
- Assisting in strategy-building for competitions.

In the broader context, this work contributes to the field of Computational Sanskrit and Digital Humanities, demonstrating how classical linguistic traditions can be integrated with modern graph-based techniques. Future research could explore real-time verse recognition, automated scoring, and integration with natural language processing (NLP) models to further enrich the Sanskrit gaming and learning experience.

Expanding this framework to larger poetic datasets, incorporating intonation-based scoring, and building collaborative online gameplay environments are promising directions for further exploration.

Code Availability

The code and data used for this study are available at the following repository: `https://github.com/Vyoma-Linguistic-Labs/Antyakshari-Krida/tree/main`.

References

[Murarka2019] Vani Murarka. 2019. Nārāyaṇīyam first step. `https://github.com/vanimurarka/Narayaneeyam-First-Step/tree/master`.

[Project2021] Gita Project. 2021. Bhagavad gītā. `https://github.com/gita/gita/tree/main`.

[Saha2006] S. K. Saha. 2006. Design for effective teaching and learning in technical education. In *National Conference on Design for Product Life Cycle*, February.

[Sinha2017] A. K. Sinha. 2017. Word games. *Language and Language Teaching*, 6(1):7–13.