Conjugacy Class Enumeration in Transformation Semigroups (Conjclassts): User Manual

Clinton Oluranran Kayoh

June 14, 2025

Abstract

This document provides comprehensive user documentation for the Python package dedicated to enumerating conjugacy classes in transformation semigroups and their subsemigroups. The package implements algorithms for generating various types of transformation semigroups and analyzing their conjugacy class structure using graph isomorphism techniques.

Contents

1 Introduction

1.1 Mathematical Background

A **transformation semigroup** on a finite set $X = \{1, 2, \ldots, n\}$ is a set of functions from X to X that is closed under composition. The **full transformation semigroup** $\mathcal{T}_n$ consists of all n^n functions from X to X. Two transformations

$f, g \in \mathcal{T}_n$ are **conjugate** if their graphs are isomorphic (i.e they posses the same path structure) or there exists an invertible transformation h such that $g = h^{-1} \circ f \circ h$. Conjugacy classes partition the semigroup and provide important structural information.

1.2 Package Overview

This Python package provides tools for:

- Generating various transformation semigroups
- Identifying conjugacy classes via graph isomorphism
- Analyzing structural properties of semigroups

2 Installation and Dependencies

2.1 Required Packages

```
import itertools
import networkx as nx
import matplotlib.pyplot as plt
```

2.2 Installation (Placeholder)

TODO: Complete installation instructions will be provided when the package is finalized.

3 Core Functions

3.1 Full Transformation Semigroup

3.1.1 FT_semigroup(n)

Description: Generates the full transformation semigroup of degree n by creating all possible n-tuples over the set $\{1, 2, \ldots, n\}$.

Mathematical Basis: Implements $\mathcal{T}_n$, the semigroup of all transformations on n points with n^n elements.

Parameters: `n`: `int` - The degree of the transformation semigroup

Returns: `list` - List of tuples representing all transformations

Complexity: $\mathcal{O}(n^n)$ time and space complexity

```
def FT_semigroup(n):
elements = list(range(1, n+1))
combinations = itertools.product(elements, repeat=n)
combinations_list = list(combinations)
return combinations_list
```

Example 1. *For $n = 3$, the function returns all $3^3 = 27$ transformations from $\{1, 2, 3\}$ to itself.*

3.2 Order-Preserving Transformation Semigroups

3.2.1 check_increasing_or_constant(combination)

Description: Checks if a transformation preserves order (non-decreasing).

Mathematical Basis: A transformation f is order-preserving if $i \leq j$ implies $f(i) \leq f(j)$.

Parameters: `combination`: `tuple` - A transformation represented as a tuple

Returns: `bool` - True if the transformation is order-preserving

```python
def check_increasing_or_constant(combination):
for i in range(1, len(combination)):
if combination[i] < combination[i-1]:
return False
return True
```

3.2.2 Order_Preserving_FT(n)

Description: Generates the order-preserving full transformation semigroup.

Mathematical Basis: The semigroup $\mathcal{OP}_n$ of all order-preserving transformations on n points.

Parameters: `n`: `int` - The degree of the semigroup

Returns: `list` - List of order-preserving transformations

```python
def Order_Preserving_FT(n):
combinations = FT_semigroup(n)
ordered_combinations = [combination for combination in combinations
if check_increasing_or_constant(combination)]
return ordered_combinations
```

3.2.3 check_decreasing_or_constant(combination)

Description: Checks if a transformation is order-reversing (non-increasing).

Parameters: `combination`: `tuple` - A transformation

Returns: `bool` - True if the transformation is order-reversing

3.2.4 Order_Preserving_Or_Reversing_FT(n)

Description: Generates transformations that are either order-preserving or order-reversing.

Parameters: `n`: `int` - The degree

Returns: `list` - List of monotone transformations

3.3 Contraction Transformation Semigroups

3.3.1 check_contraction(combination)

Description: Checks if a transformation is a contraction (Lipschitz constant ≤ 1).

Mathematical Basis: f is a contraction if $|f(i) - f(j)| \leq |i - j|$ for adjacent elements.

Parameters: `combination`: `tuple` - A transformation

Returns: `bool` - True if it's a contraction

```python
def check_contraction(combination):
for i in range(1, len(combination)):
if abs(combination[i] - combination[i-1]) > 1:
return False
return True
```

3.3.2 Contraction_FT(n)

Description: Generates the full contraction transformation semigroup.

Parameters: `n`: `int` - The degree

Returns: `list` - List of contraction transformations

3.3.3 Order_Preserving_Contraction_FT(n)

Description: Generates order-preserving contraction transformations.

Mathematical Basis: Intersection of order-preserving and contraction semigroups.

Parameters: `n`: `int` - The degree

Returns: `list` - List of transformations satisfying both properties

3.4 Conjugacy Class Analysis

3.4.1 conjugacy_class(tuples_list)

Description: Partitions transformations into conjugacy classes using graph isomorphism.

Mathematical Basis: Two transformations are conjugate iff their functional graphs are isomorphic as directed graphs.

Parameters: `tuples_list`: `list` - List of transformations

Returns: `dict` - Dictionary mapping class numbers to transformations and their graph representations

Visualization: Automatically displays graph diagrams for each conjugacy class

```
def conjugacy_class(tuples_list):
classes = {}
for i, tpl in enumerate(tuples_list, start=1):
G = nx.DiGraph()
for j, el in enumerate(tpl, start=1):
G.add_edge(j, el)
found = False
for k, v in classes.items():
if nx.is_isomorphic(G, v[0]):
v.append(tpl)
found = True
break
if not found:
classes[len(classes)+1] = [G, tpl]
return classes
```

4 Usage Examples

4.1 Basic Semigroup Generation

```
# Generate full transformation semigroup for n=3
T3 = FT_semigroup(3)
print(f"Full transformation semigroup size: {len(T3)}")

# Generate order-preserving transformations
OP3 = Order_Preserving_FT(3)
print(f"Order-preserving transformations: {len(OP3)}")
```

4.2 Conjugacy Class Analysis

```
# Analyze conjugacy classes in T3
classes = conjugacy_class(FT_semigroup(3))
for class_id, elements in classes.items():
print(f"Class {class_id}: {len(elements)-1} elements")
```

5 Mathematical Properties

Theorem 1. *The conjugacy class of a transformation f is determined by its cycle-chain structure, which corresponds to the isomorphism type of its functional graph.*

Theorem 2. *For the full transformation semigroup $\mathcal{T}_n$, the number of conjugacy classes equals the number of isomorphism types of functional graphs on n vertices.*

6 Limitations and Future Development

6.1 Current Limitations

- Only handles small n due to exponential growth ($n \leq 5$ recommended)
- Basic graph isomorphism without optimization
- No algebraic operations (composition, inverses) implemented yet

6.2 Planned Features

TODO:

- Efficient data structures for larger semigroups
- Algebraic operations and Green's relations
- Optimization using cycle-chain decomposition
- Integration with semigroup theory databases

7 Conclusion

This package provides foundational tools for studying transformation semigroups and their conjugacy classes. The current implementation focuses on generation and basic classification, with future development planned for more advanced algebraic operations and optimizations.

References

[1] Howie, J. M. (1995). *Fundamentals of Semigroup Theory*. Oxford University Press.

[2] Cameron, P. J. (1999). *Permutation Groups*. Cambridge University Press.

[3] Abusarris, H., & Ayık, G. (2023). On the rank of generalized order-preserving transformation semigroups. *Turkish Journal of Mathematics*. https://doi.org/10.55730/1300-0098.3420.

[4] Adan-Bante, E. (2005). On nilpotent groups and conjugacy classes. *arXiv: Group Theory*, 345-356.

[5] Ahmad, A., Magidin, A., & Morse, R. (2012). Two generator p-groups of nilpotency class 2 and their conjugacy classes. *Publicationes Mathematicae Debrecen*, 81, 145-166. https://doi.org/10.5486/PMD.2012.5114.

[6] Akinwunmi, S.A. & Makanjuola, S.O. (2019). Enumeration Partial Contraction Transformation Semi-groups. *Journal of the Nigerian Association of Mathematical Physics*, 29(B), 70-77.

[7] Akinwunmi, S.A., Mogbonju, M.M., Adeniji, A.O., Oyewola, D.O., Yakubu, G., Ibrahim, G.R., & Fatai, M.O. (2021). Nildempotency Structure of Partial One-One Contraction CIn Transformation Semigroups. *International Journal of Research and Scientific Innovation (IJRSI)*, 8(1), 230-236.

[8] Mogbonju, M.M., Gwary, T.M., & Ojeniyi, A.B. (2014). Presentation of Conjugacy Classes in the Partial Order-Preserving Transformation Semigroup with the aid of graph. *Mathematical Theory and Modeling*, 4, 110-142.

[9] Mohammadian, A., & Erfanian, A. (2018). On the nilpotent conjugacy class graph of groups. *Journal of Algebra and Its Applications*, 37, 77-90. https://doi.org/10.1142/S0219498818500407.

[10] Ugbene, I.J., Eze, E.O., & Makanjuola, S.O. (2013). On the Number of Conjugacy Classes in the Injective Order-Decreasing Transformation Semigroup. *Pacific Journal of Science and Technology*, 14(1), 182-186.

[11] Ugbene, I.J. & Makanjuola, S.O. (2012). On the Number of Conjugacy Classes in the Injective Order-preserving Transformation Semigroup. *Icastor Journal of Mathematical Sciences*, 6(1), 1-8.

[12] Ugbene, I.J., Suraju, O., & Ugochukwu, N. (2021). Digraph of the full transformation semigroup. *Journal of Discrete Mathematical Sciences and Cryptography*, 25(8), 2457-2465. https://doi.org/10.1080/09720529.2020.1862955.