



Geometric Propagation of Melnikov Grammar using NITROS

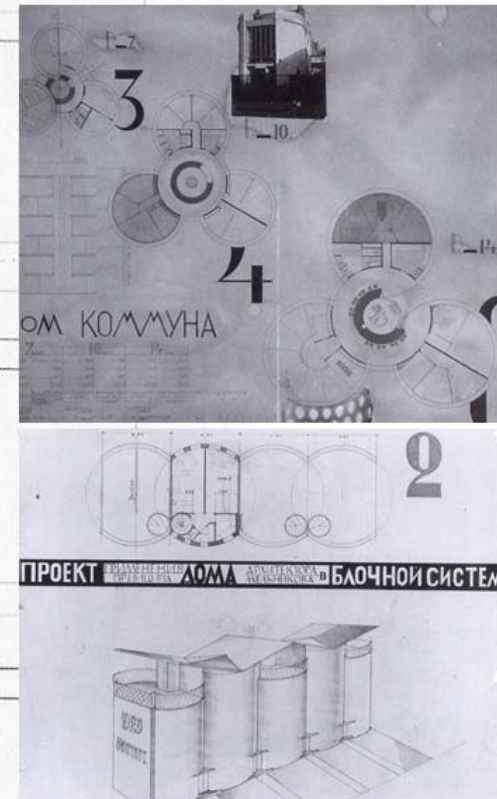
This study extracts the basic design principles of Russian Architect Konstantin Melnikov's architecture and turns them into a set of geometric rules that are implemented in a computer program using AutoLisp and NITROS. By implementing collision detection methods and other constraints the grammar is able to propagate a zone in an "autonomous" way. This document shows both the grammar and the results of different experiments resulting from different constraints in the implementation of the program.

Daniel Cardoso SMArchS - Computation
Department of Architecture

4.564. Formal Design Knowledge and Programmed Constructs

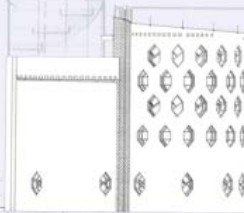
School of Architecture and Planning
Massachusetts Institute of Technology

Schemes of 1929 for clustering cylindrical brick units for communal housing

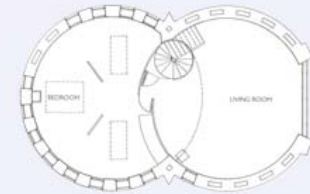


Melnikov's Principles

Melnikov's use of cylinders to articulate space is a salient feature of his architecture. Most of his designs were not built, but the house he built for himself stands as a summary of some of his design principles. The additive character of his architecture is very suitable for grammatical inquiry. A first detailed grammar for the house is developed.














Melnikov House, Elevation



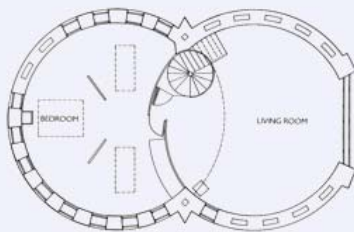
Plan

Component Shapes

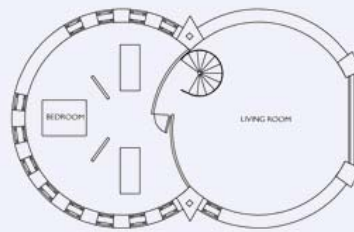
abstract_end Guide: Not a part of the actual physical space  a	circle Parameter: none  b	division_wall Parameter: none  c1	division_wall Parameter: length  c2
door Parameter: none  d	window Parameter: none  e1	window Parameter: length  e2	pilaster Parameter: none  f1
plaster Parameter: none  f2	stair Parameter: none  g	door Parameter: width (These elements end the grammar)  h	

Melnikov's House Lexicon

These are the component shapes of Melnikov's house (above). And the simplification of the architectural elements that led to the definition of the basic Melnikov lexicon (left).

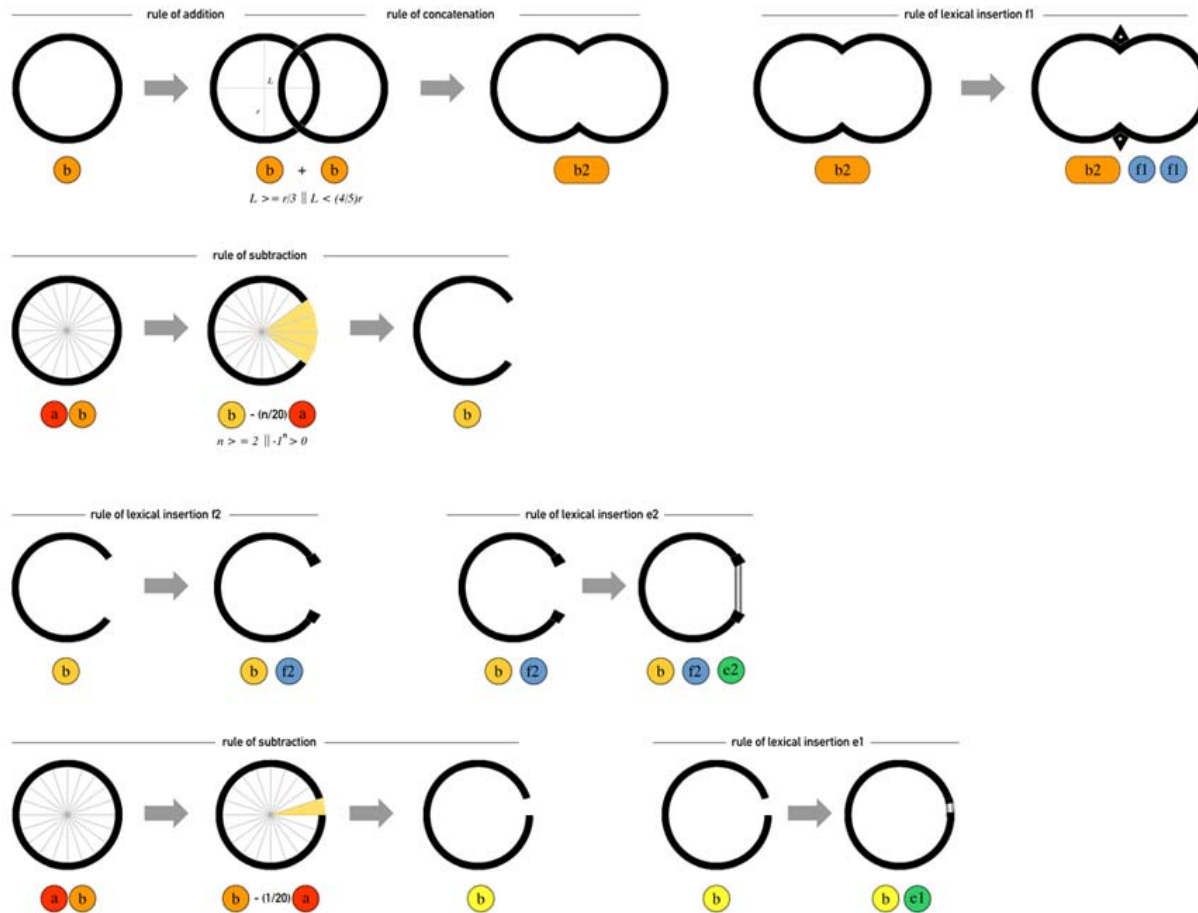


Plan of the House

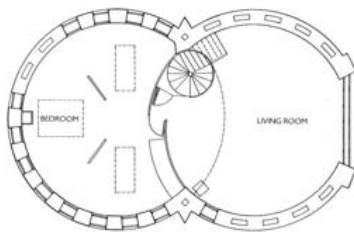


Simplified Plan

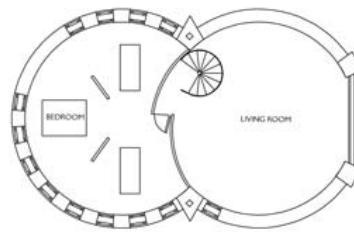
Composition rules 1



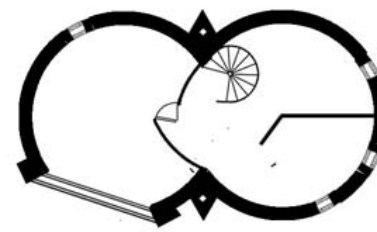
Melnikov's House rules



Plan of the House

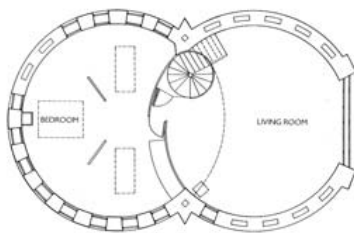
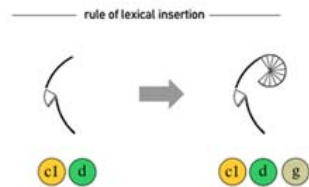
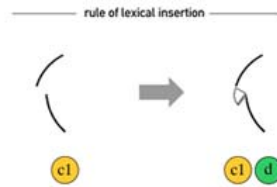
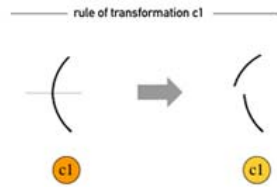
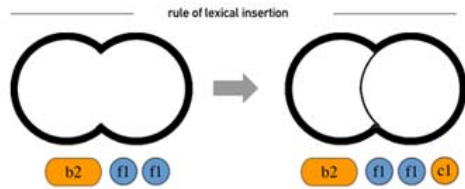


Simplified Plan

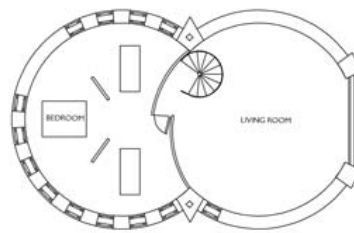


A new generation of the Melnikov House

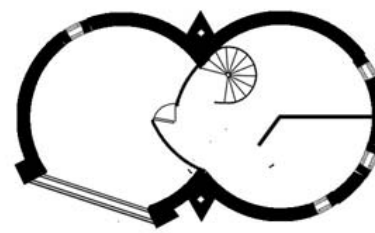
Composition rules 2



Plan of the House

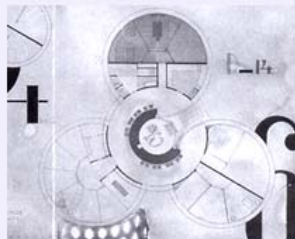
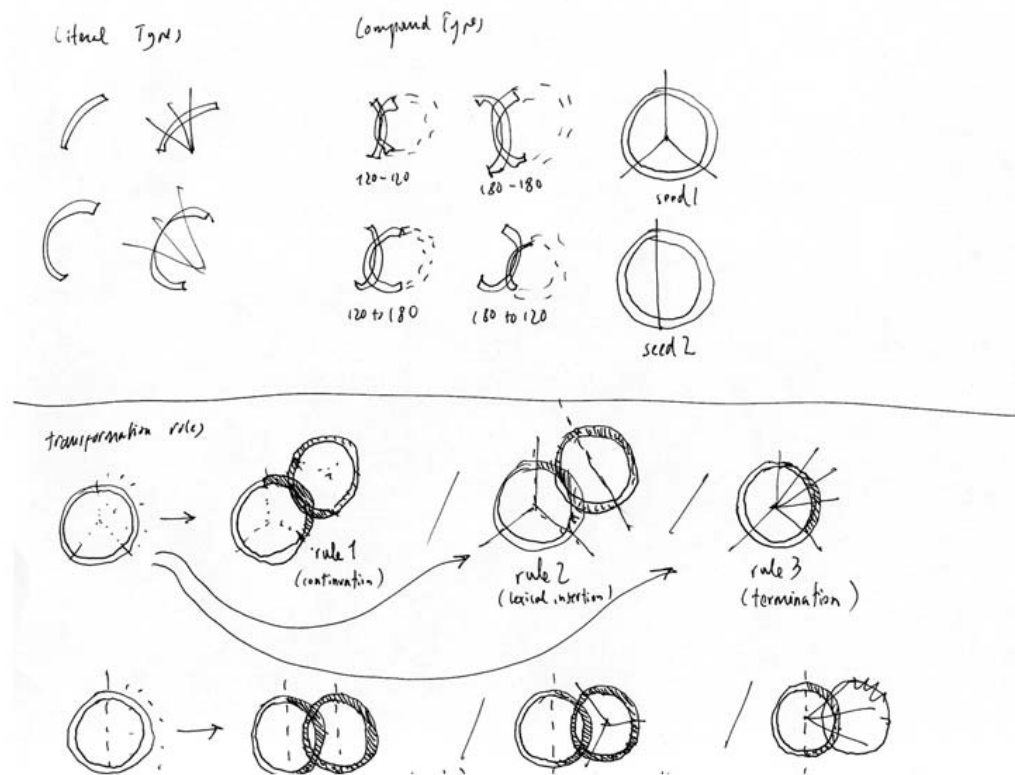


Simplified Plan



A new generation of the Melnikov House

Melnikov's House rules






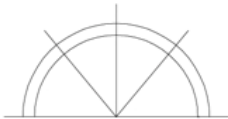




+



The additive process of Melnikov Architecture works can be rationalized. Above, 120 and 180 degrees addition of cylinders

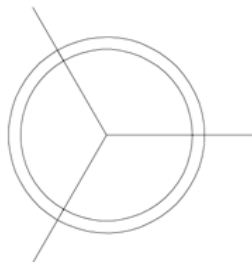
New Approach

In order to implement Melnikov's Grammar in a computer program, I decide to combine the two main geometric principles of Melnikov's Architecture; the connection of cylinders divided in three and two sections respectively. This grammar takes distance from the specific resolution of the Melnikov House, to provide a set of rules and shapes that relate to Melnikov's architecture in a more general way.

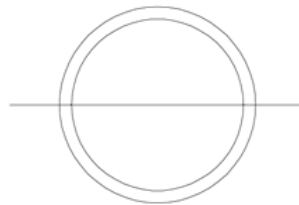
 <p>ar_120_part</p>	 <p>ar_180_part</p>
 <p>ar120_end</p>	 <p>ar180_end</p>
 <p>ar120 to 120 link</p>	 <p>ar180 to 180 link</p>
 <p>ar120 to 180 link</p>	 <p>ar180 to 120 link</p>

Literal Component Types

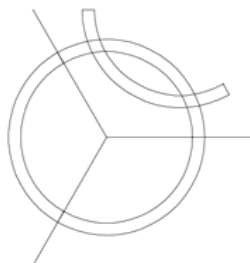
In order to implement Melnikov's Grammar in a computer program, I decide to combine the two main geometric principles of Melnikov's Architecture; the connection of cylinders divided in three and two sections respectively. This grammar takes distance from the specific resolution of the Melnikov House, to provide a set of rules and shapes that relate to Melnikov's architecture in a more general way.



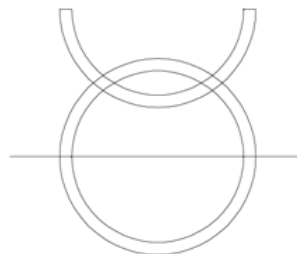
seed 120



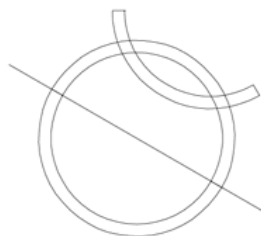
seed 180



linked 120



linked 180



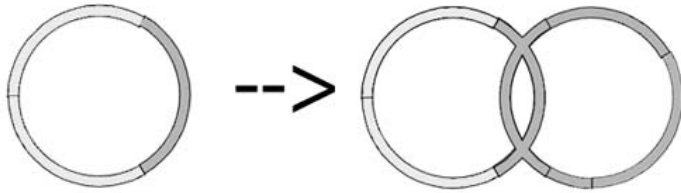
linked 120 to 180



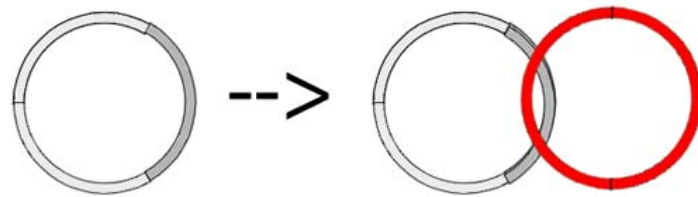
linked 180 to 120

Compound Types

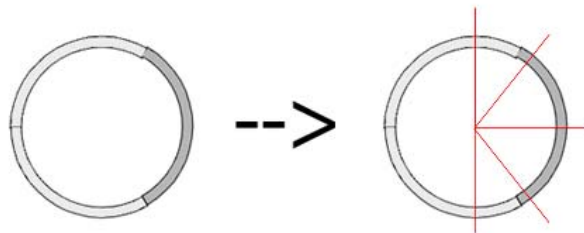
These shapes are generated from the addition of two or more literal types



Rule 1a - Continuation $ar120 \rightarrow ar120_part + ar120_link$
 Required Parameter: $ar120_part$ Center (x,y,z) , rotation angle



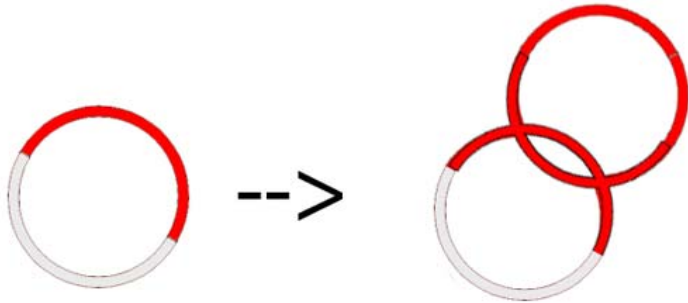
Rule 2a - Lexical Insertion $ar120 \rightarrow ar120 + ar120to180_link$
 Required Parameter: $ar120_part$ Center (x,y,z) , rotation angle



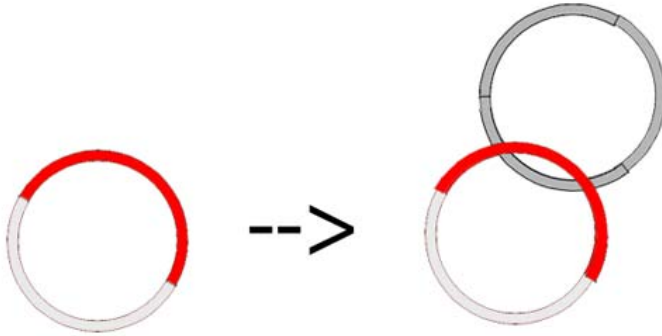
Rule 3a - Termination $ar120 \rightarrow ar120_part + ar120_end$
 Required Parameter: $ar120_part$ Center (x,y,z) , rotation angle

Rules of Transformation

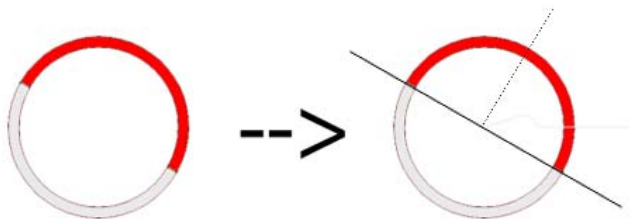
This set of rules transforms the "120" shapes.



Rule 1b - Continuation $\text{ar180} \rightarrow \text{ar180_part} + \text{ar180_link}$
 Required Parameter: ar180_part Center (x,y,z), rotation angle



Rule 2a - Lexical Insertion $\text{ar180} \rightarrow \text{ar180} + \text{ar120to180_link}$
 Required Parameter: ar120_part Center (x,y,z), rotation angle



Rule 3b - Termination $\text{ar180} \rightarrow \text{ar180_part} + \text{ar180_end}$
 Required Parameter: ar180_part Center (x,y,z), rotation angle

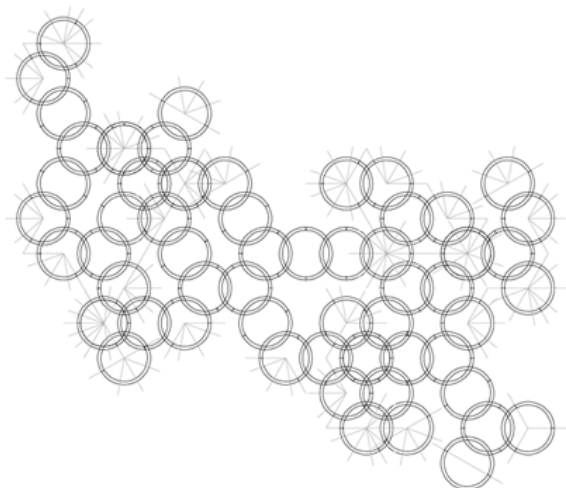
Rules of Transformation

This set of rules transforms the "180" shapes.

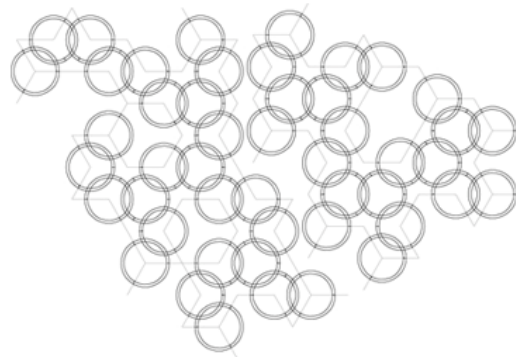
		Rules of Transformation					
	Literal Types	continue120	end120	switchTo180	continue180	end180	switchTo120
	ar120_part						
	ar120_end						
	ar120_link						
	ar180_part						
	ar180_end						
	ar180_link						
	ar120to180_link						
	ar180to120_link						
	Compound Types						
	seed120						
	linked120						
	linked120to180						
	seed180						
	linked180						
	linked180to120						

Rules of Transformation

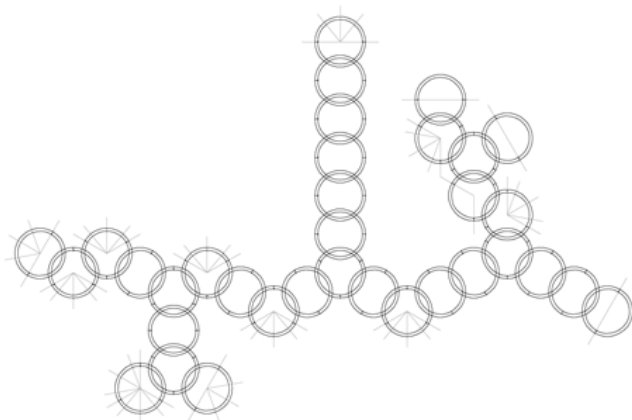
This chart relates shapes to rules. It shows how the connection pieces can't be transformed.



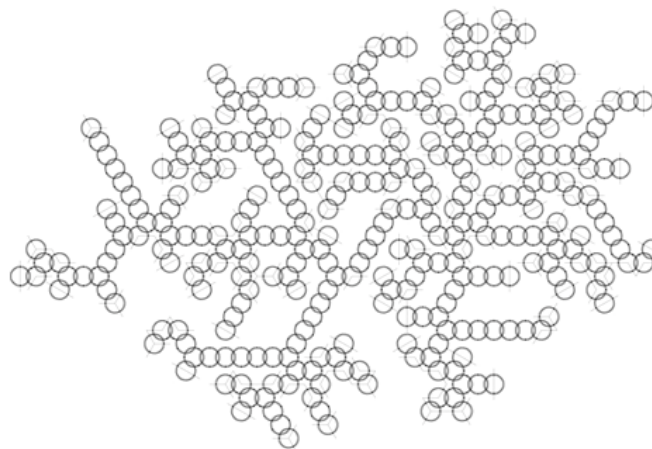
All rules are working in this example. No collisions check



Termination rules deactivated. Open System. Collisions check yes.



All rules are working in this example. Collisions check yes.

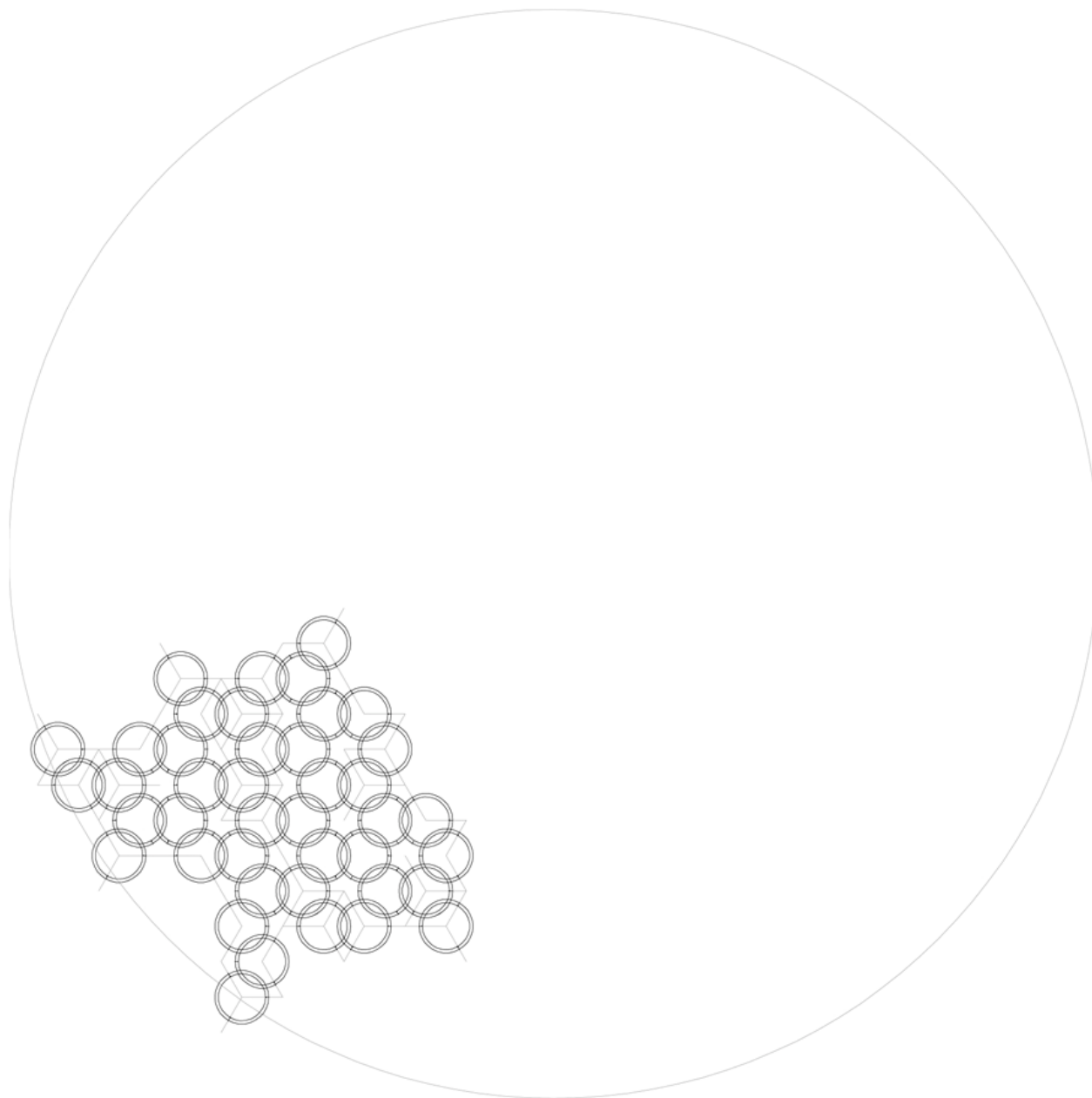


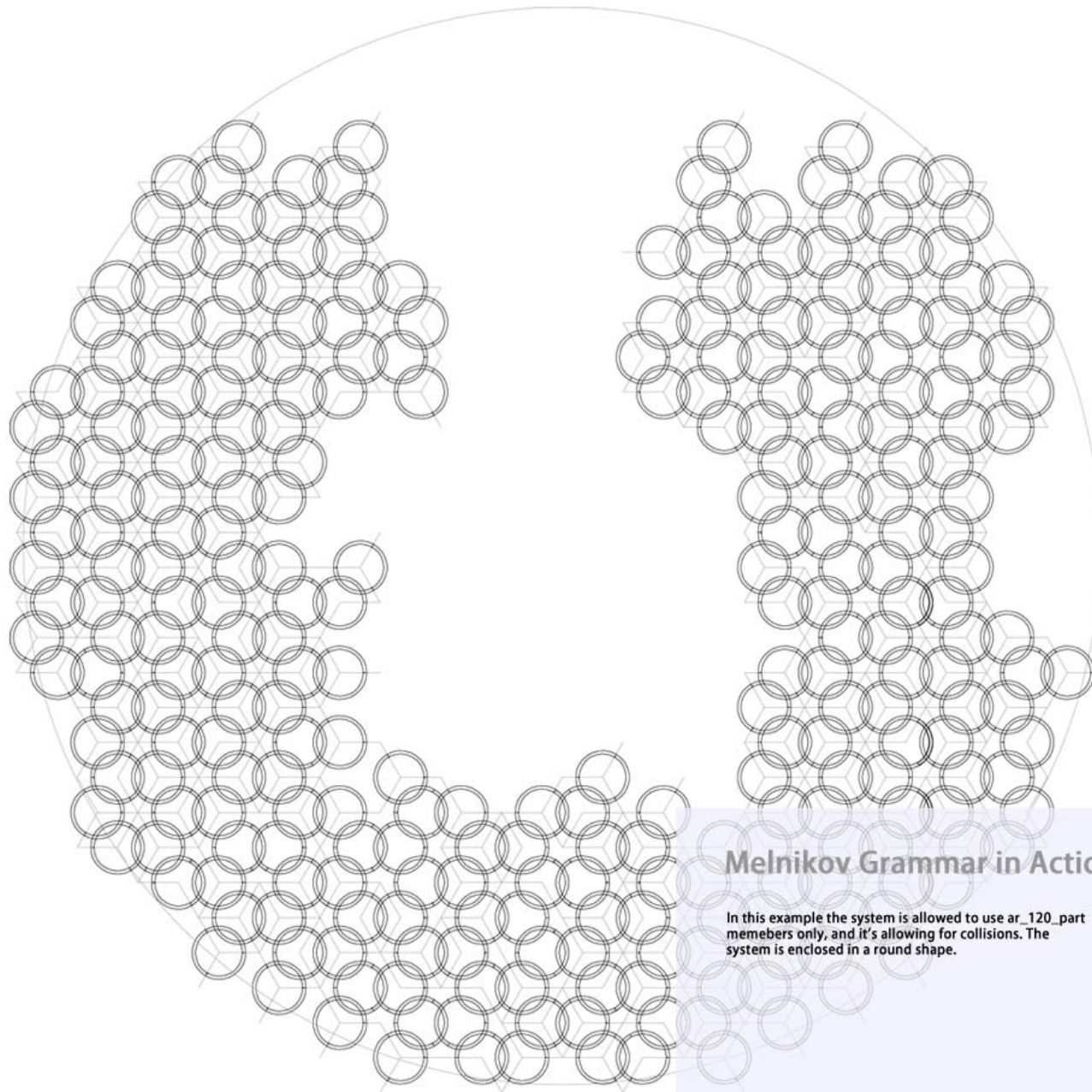
All Rules are working, termination rules deactivated. Open System. Collisions check yes.



Melnikov Grammar in Action

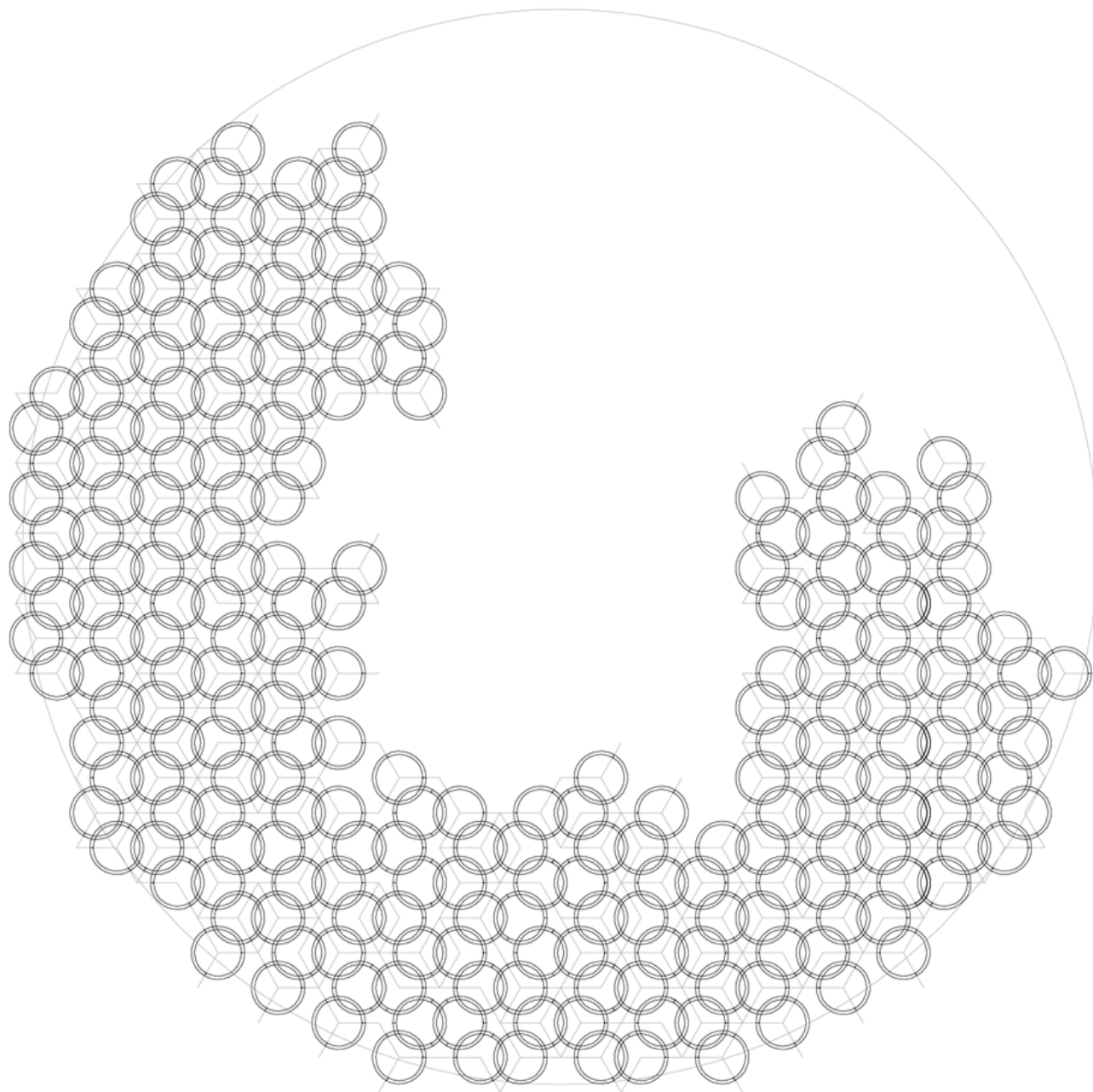
In this example the system is allowed to use ar_120_part members only, and it's allowing for collisions. The system is enclosed in a round shape.

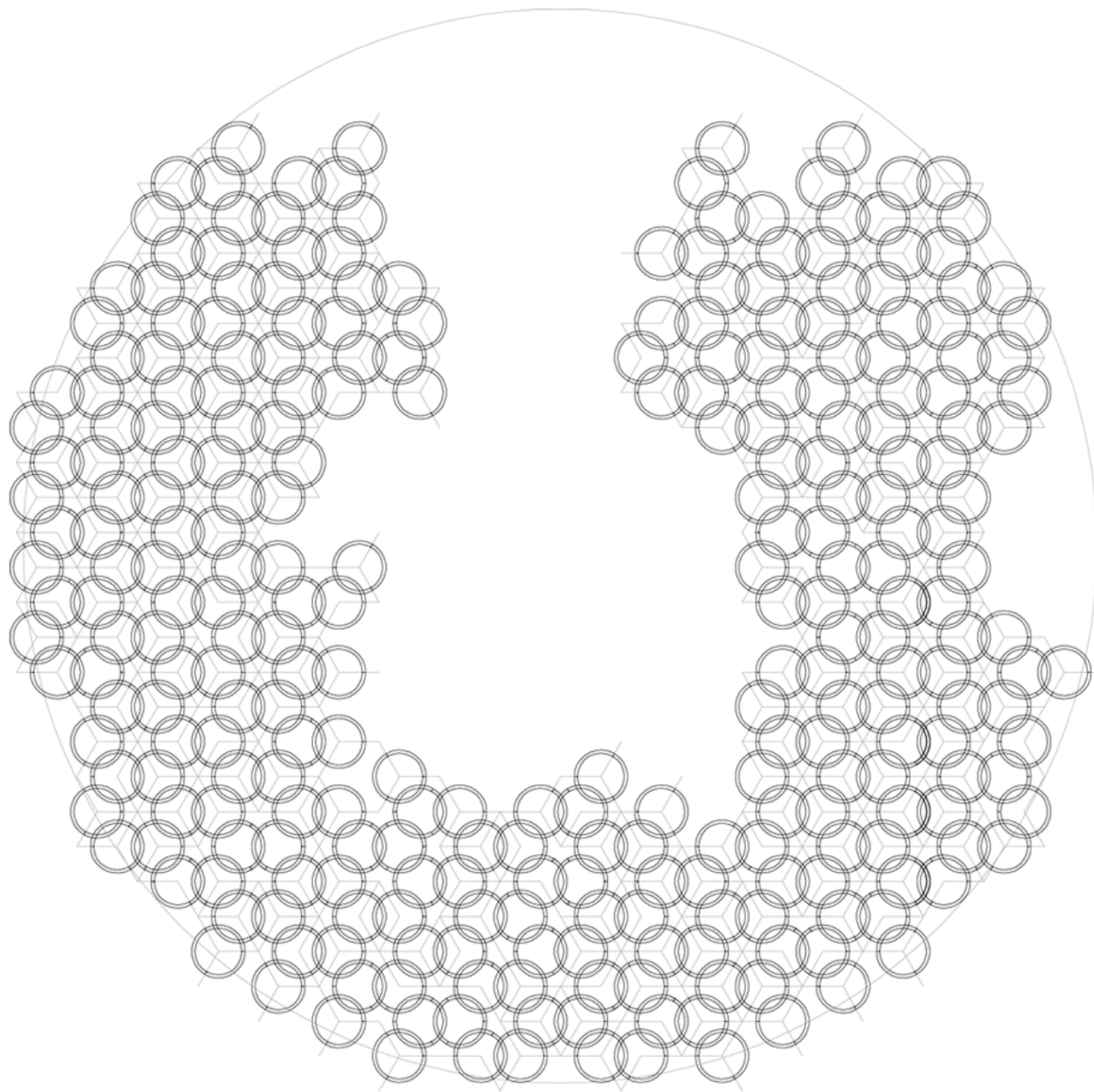


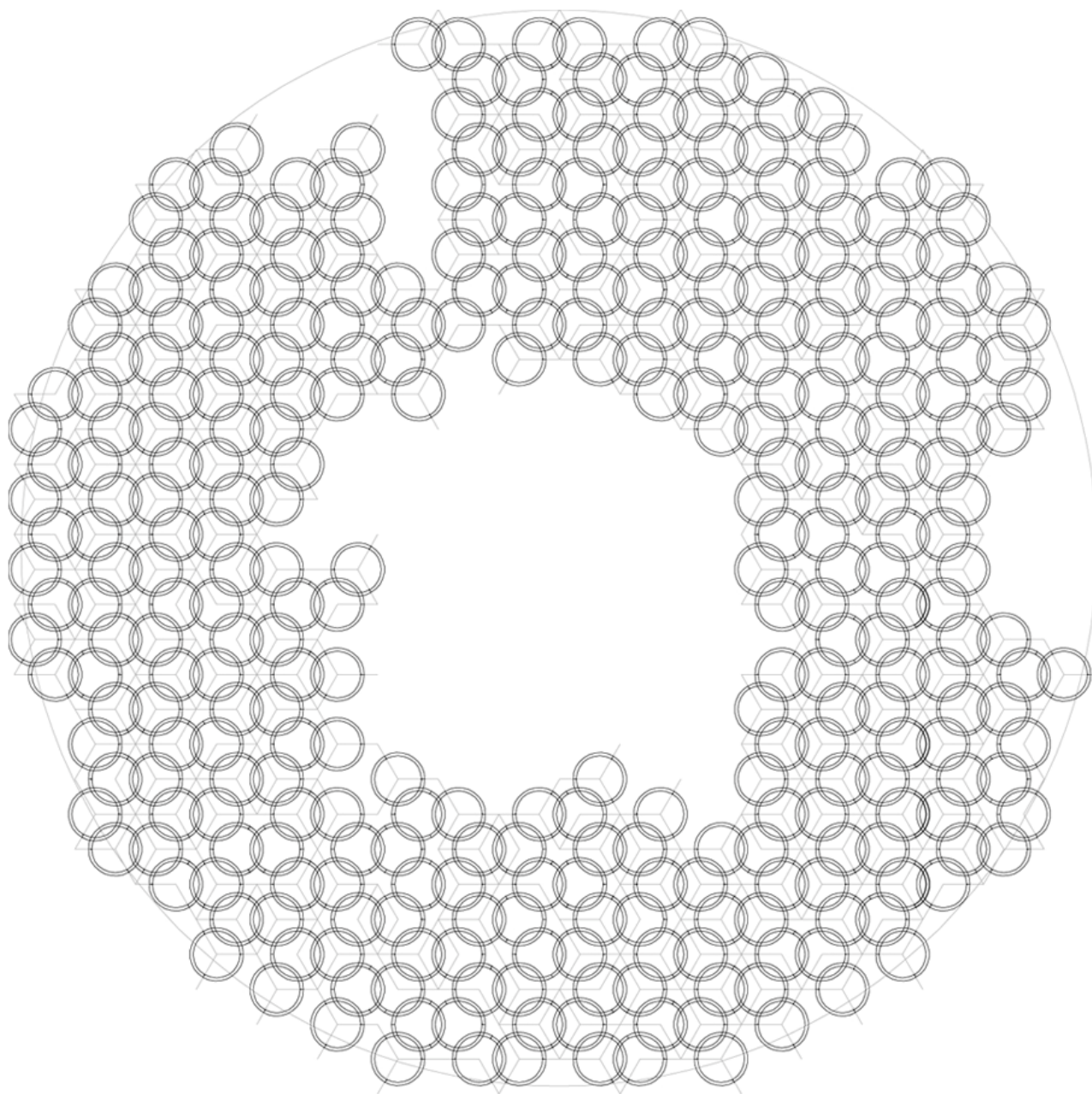


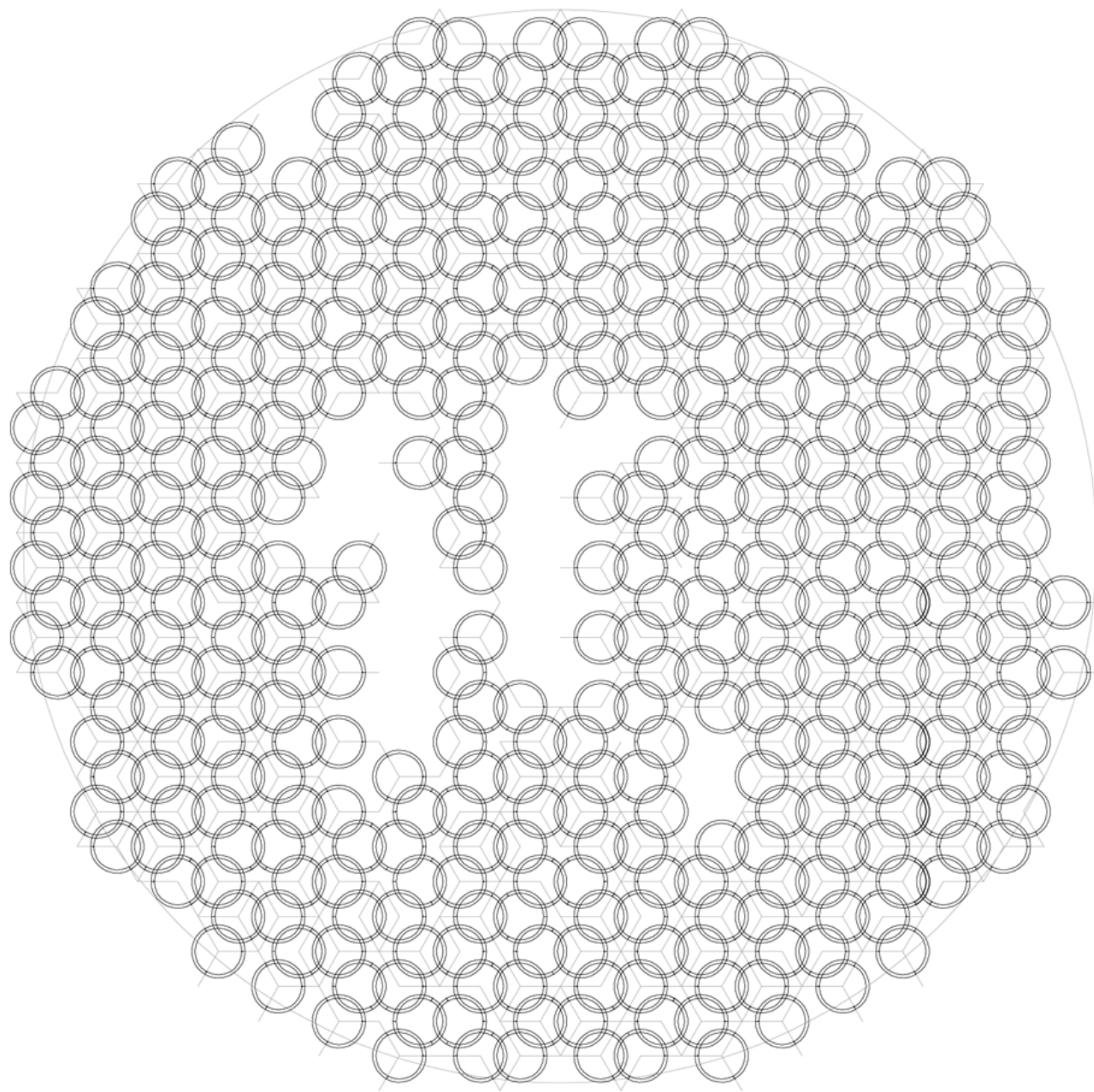
Melnikov Grammar in Action

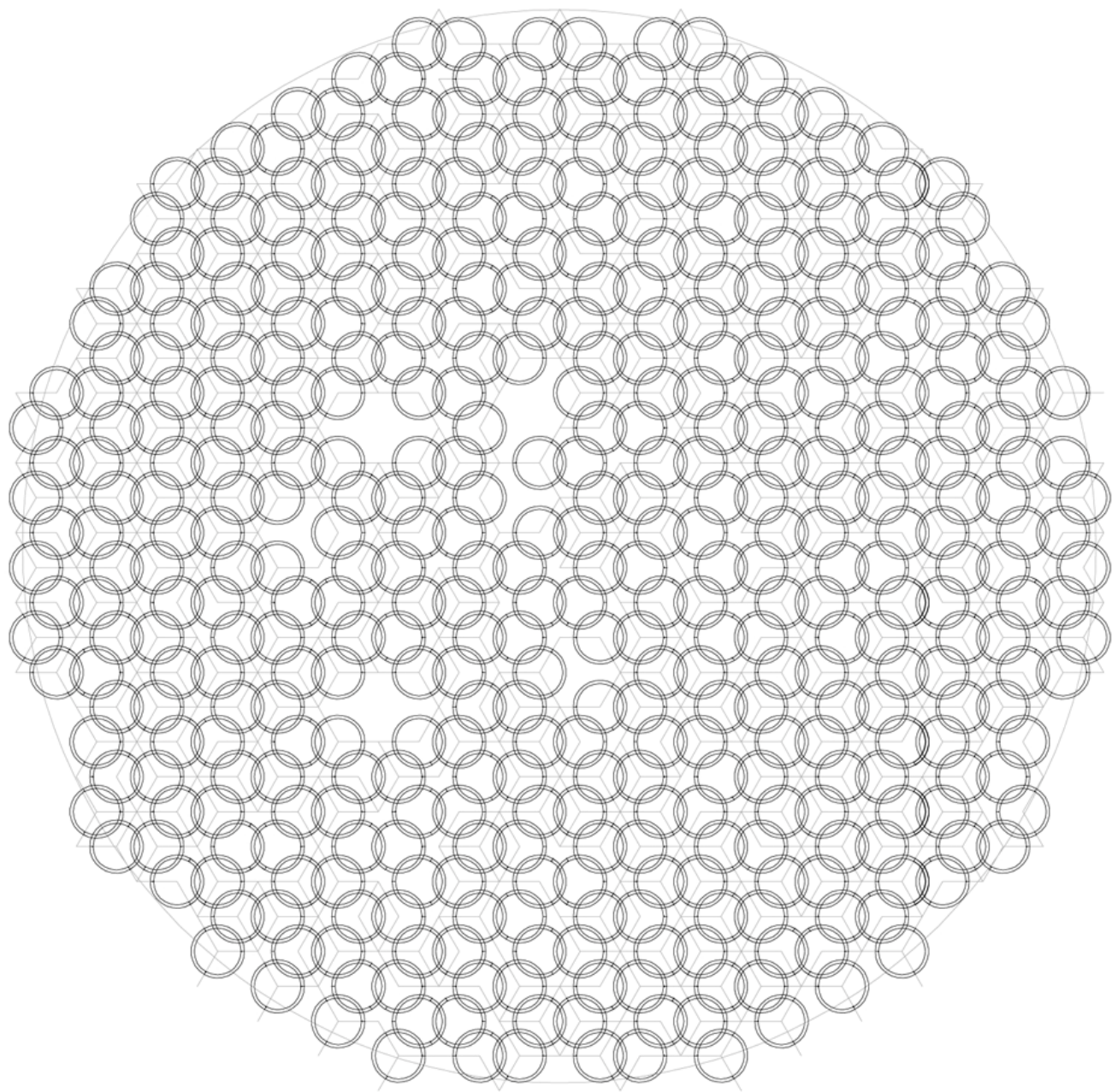
In this example the system is allowed to use ar_120_part members only, and it's allowing for collisions. The system is enclosed in a round shape.

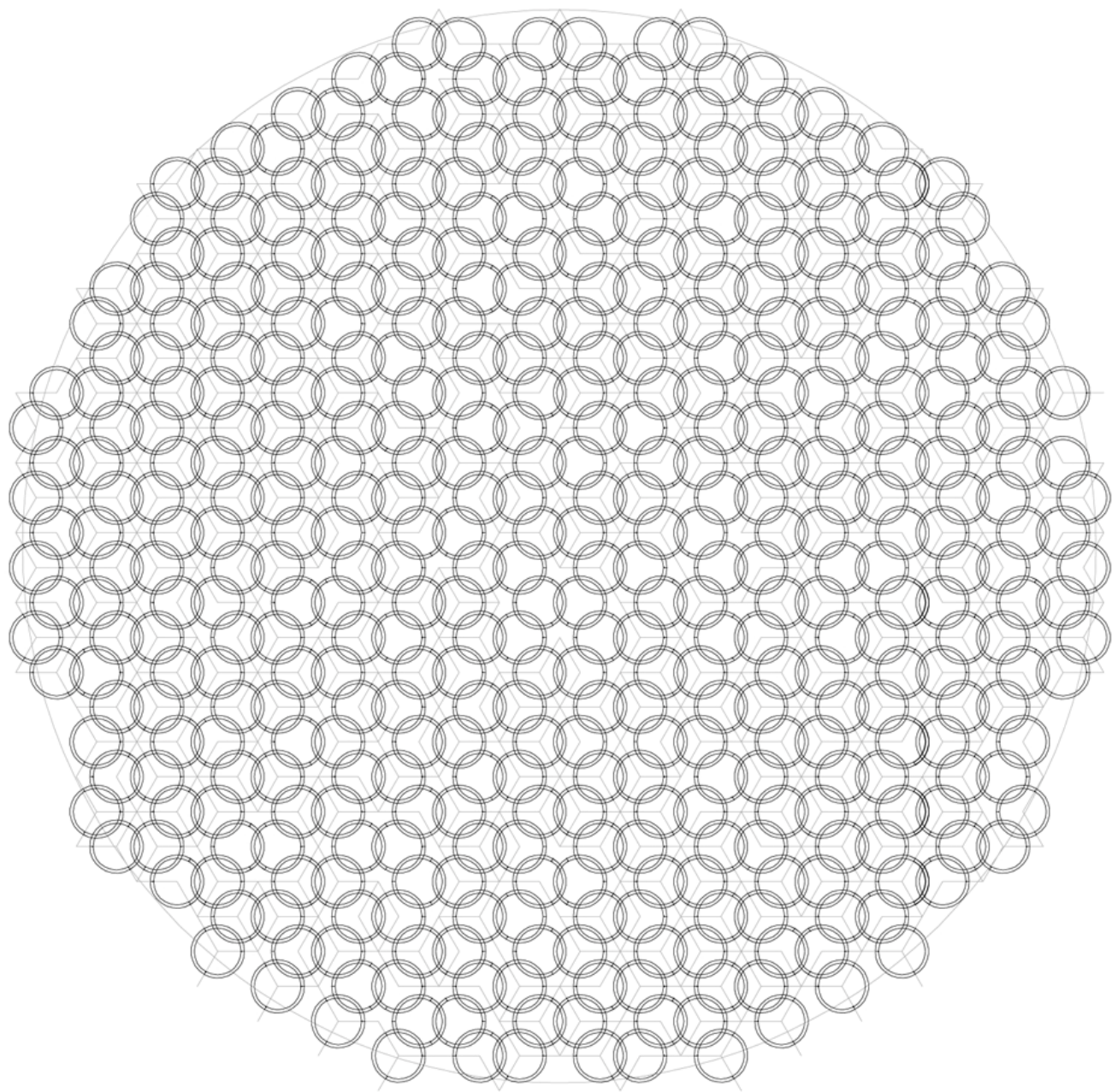


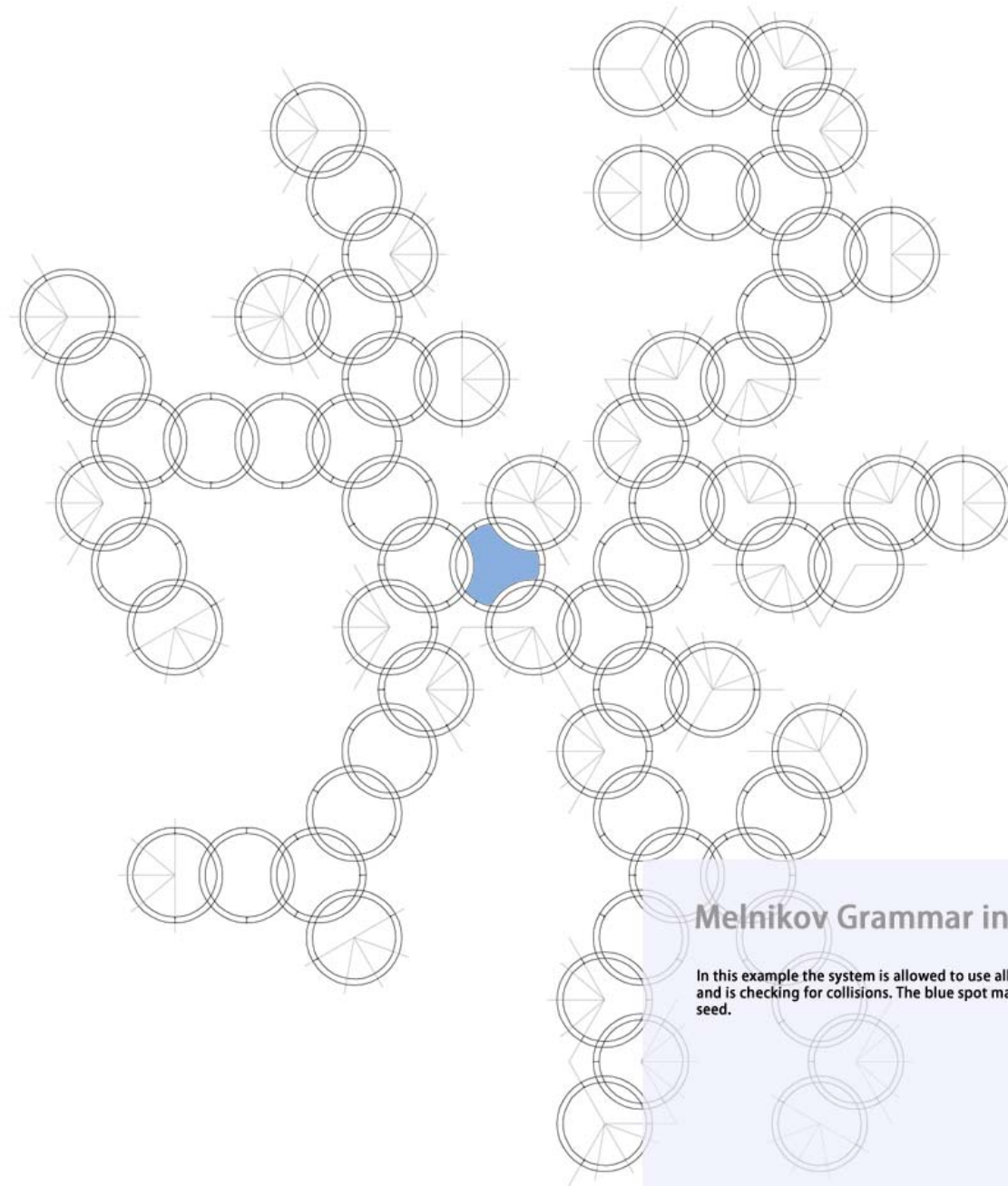












Melnikov Grammar in Action

In this example the system is allowed to use all the rules, and is checking for collisions. The blue spot marks the seed.

