

Lattices visualized as 3D-objects in a browser and in virtual reality

Natanael Alpay and Peter Jipsen

Chapman University



What is a lattice?

A **lattice** is an algebra (A, \wedge, \vee) defined by the following equations for all $x, y, z \in A$

$$\begin{aligned} (x \vee y) \vee z &= x \vee (y \vee z) & (x \wedge y) \wedge z &= x \wedge (y \wedge z) \\ x \vee y &= y \vee x & x \wedge y &= y \wedge x \\ x \vee (x \wedge y) &= x & x \wedge (x \vee y) &= x \end{aligned}$$

Example of lattice

Consider the lattice of subsets of the set $\{a, b, c\}$

$$P(\{a, b, c\}) = (\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}, \cap, \cup)$$

which can be visualized as a cube

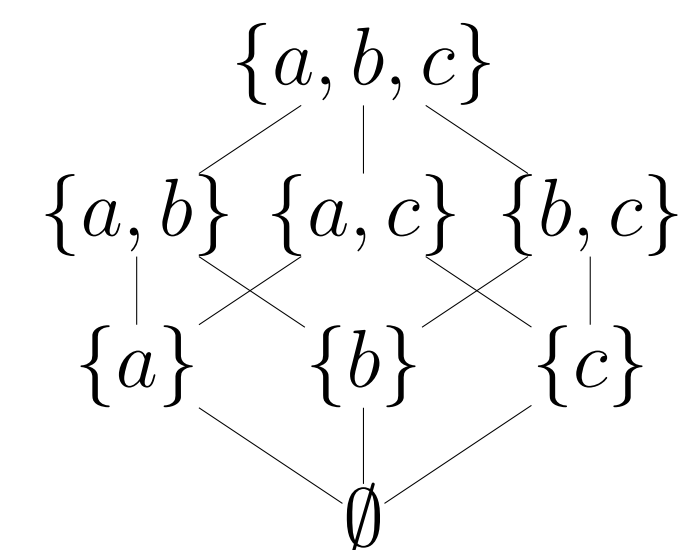


Figure: Lattice representation of the power set of P .

For example $\{a, b\} \cap \{a, c\} = \{a\}$ and $\{a\} \cup \{c\} = \{a, c\}$.

More visualized lattices



Figure: QR code for our webpage containing more lattices visualizations.

More about lattices

A Lattice is **distributive** if $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

A lattice is **bounded** if it has a bottom and a top element.

In a lattice, x is a **complement** of y if $x \wedge y$ is the bottom element and $x \vee y$ is the top element.

A bounded distributive lattice is **Boolean** if every element has a complement.

A lattice is **complete** if $\wedge S$ and $\vee S$ exist for all $S \subseteq A$.

Outline

Abstract algebra consists of the study of algebraic structures such as groups, rings, fields, modules, vector spaces, and algebras. Each of these mathematical structures has a subalgebra lattice and a congruence lattice, which are partially ordered structures that can be represented graphically in two or three dimensions. However, even for small algebraic structures these lattices can be large, and as the number of elements grow, can very easily get hard to be analyzed with pen and paper as a 2D projection on a plane. Using 3D models and virtual reality one can model lattices, and investigate these mathematical objects with better understanding and visualization of their form and properties. This is particularly helpful for students who have just been introduced to the concept of abstract algebraic structures, and gives them a tool to help visualize the abstract concepts.

We have developed JavaScript programs to visualize 3D models of lattices within the Babylon.js framework. This allows the lattices to be viewed as 3D objects in a standard web browser, as well as with a virtual reality headset. In the project, we will present a visualization of the lattice representation of a hypercube, the lattice of all preorders on a 3-element set with 29 elements, and the lattice of all preorders on a 4-element set which contains 355 elements.

Lattices visualised 2D

Before we can visualize the lattices in virtual reality in 3D, we first study them as a 2D projection in order to get the positions and covers for vertices and edges respectively.

For that, we consider Dr. Jipsen's website [3] in which we enter a list of maximal closed sets. For the case of the Hypercube, the input will consist of the all 3 element subset of a 4 element set (entered as lists). That is

$$[[["a", "b", "c"], ["a", "b", "d"], ["a", "c", "d"], ["b", "c", "d"]],]$$

which could also be entered as

$$[[[1, 2, 3], [1, 2, 4], [1, 3, 4], [2, 3, 4]],]$$

This would result in the following lattice output:

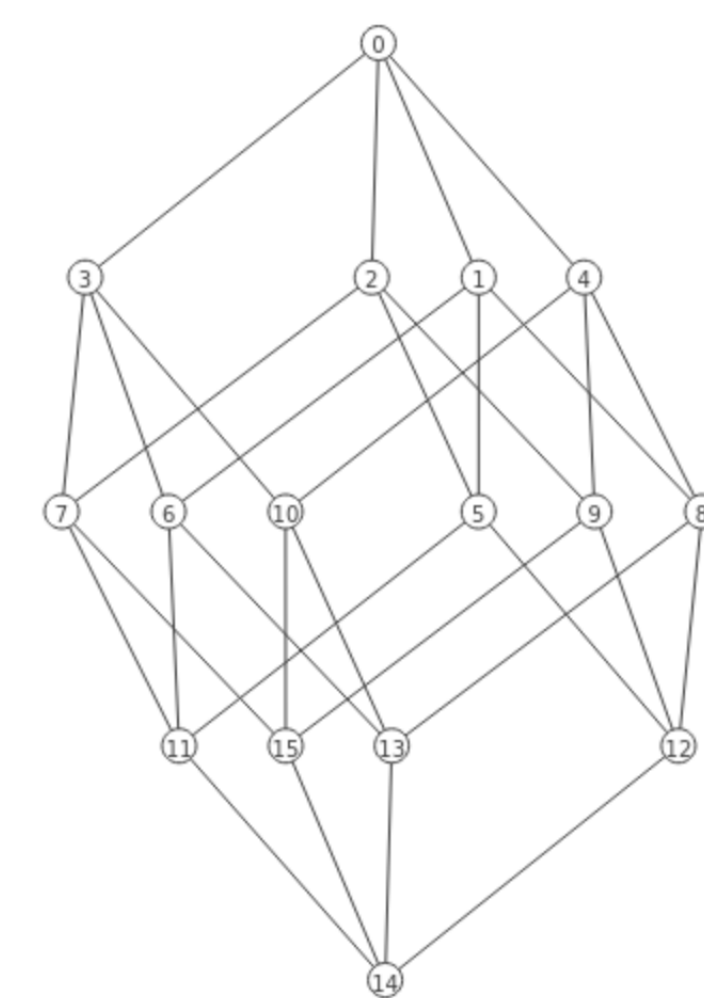
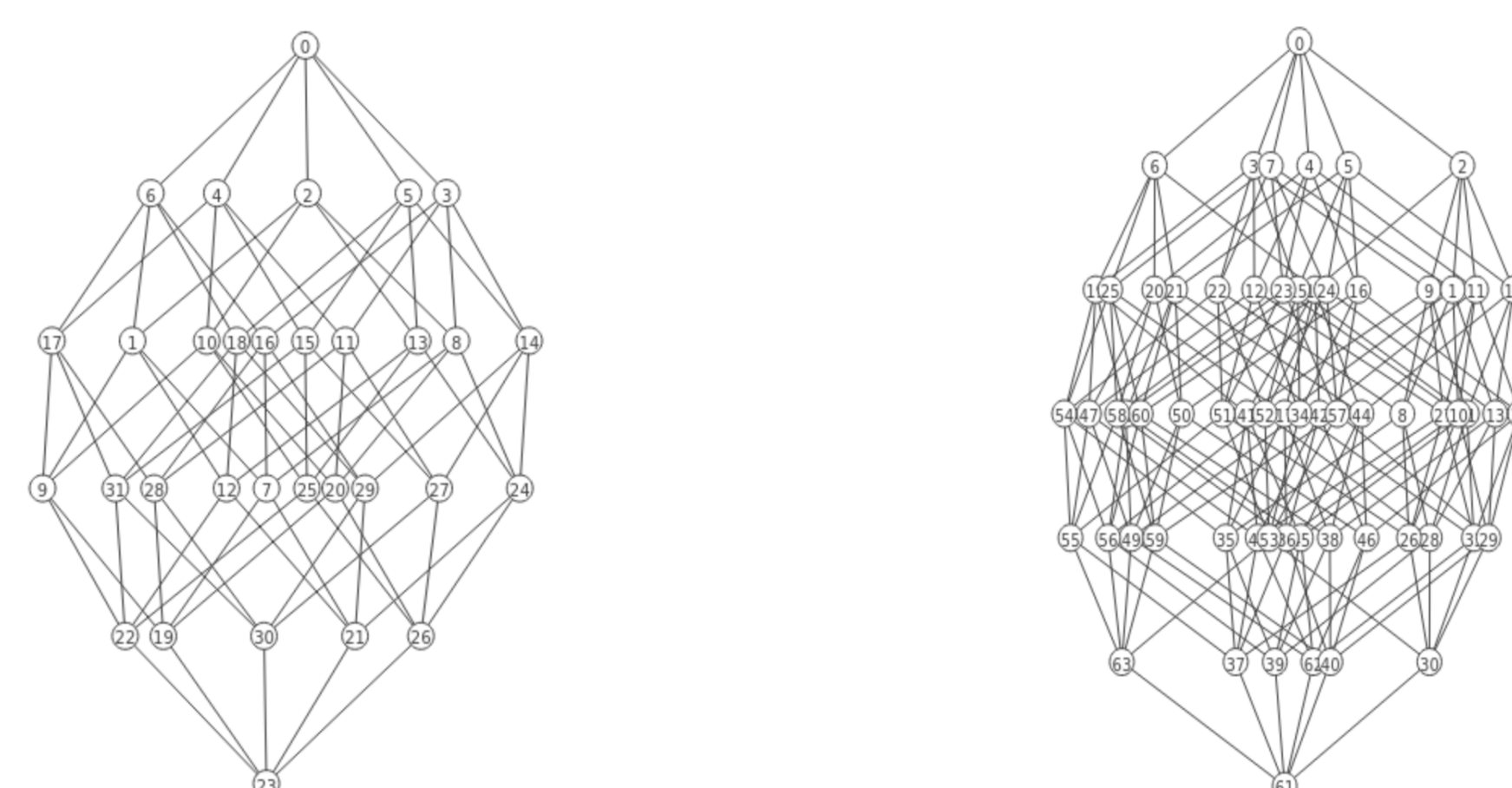


Figure: 16 elements Hypercube.

Using a similar input but with a largest set, we can build many more Hypercube lattices

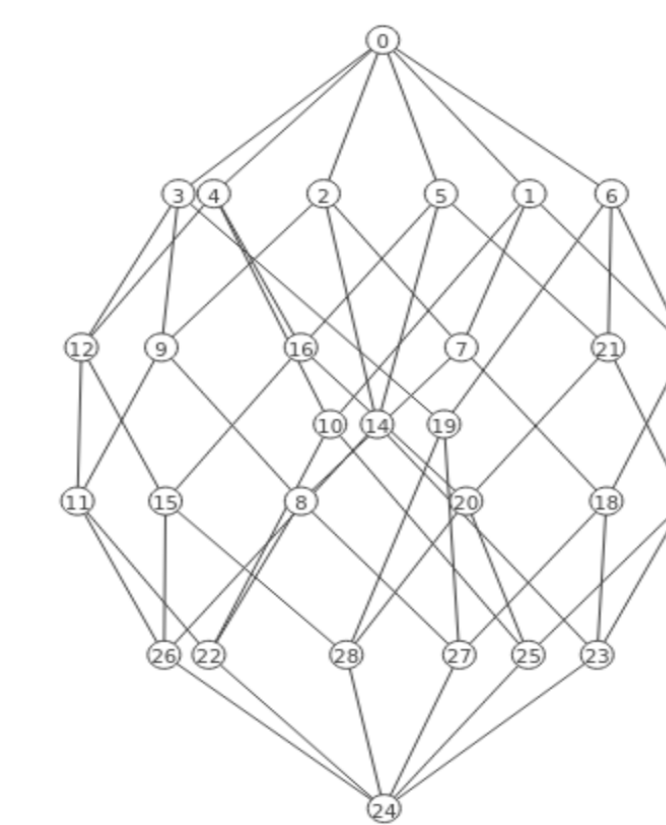


(a) Hypercube 5: 32 Element lattice

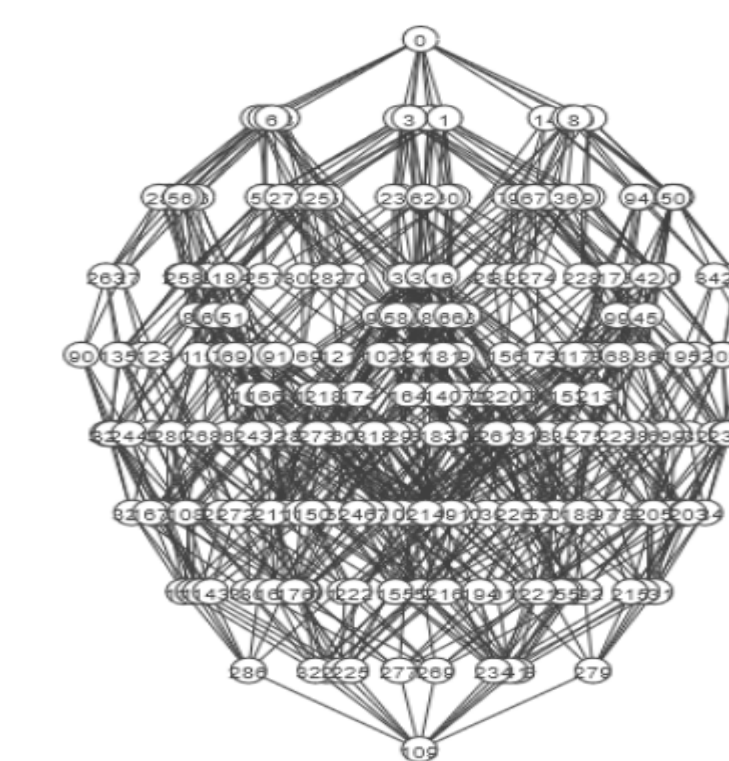
(b) Hypercube 6: 64 Element lattice

Lattices with many elements

From the previous figures it is not always easy to understand what's going on, and even if we can rotate the image, it would not be helpful in the case we have more elements, such as the following case with 355 elements



(a) Pre-order lattice on a 3 element set with 29 elements.



(b) Pre-order lattice on a 4 element set with 355 elements.

Babylon.js and WebXR

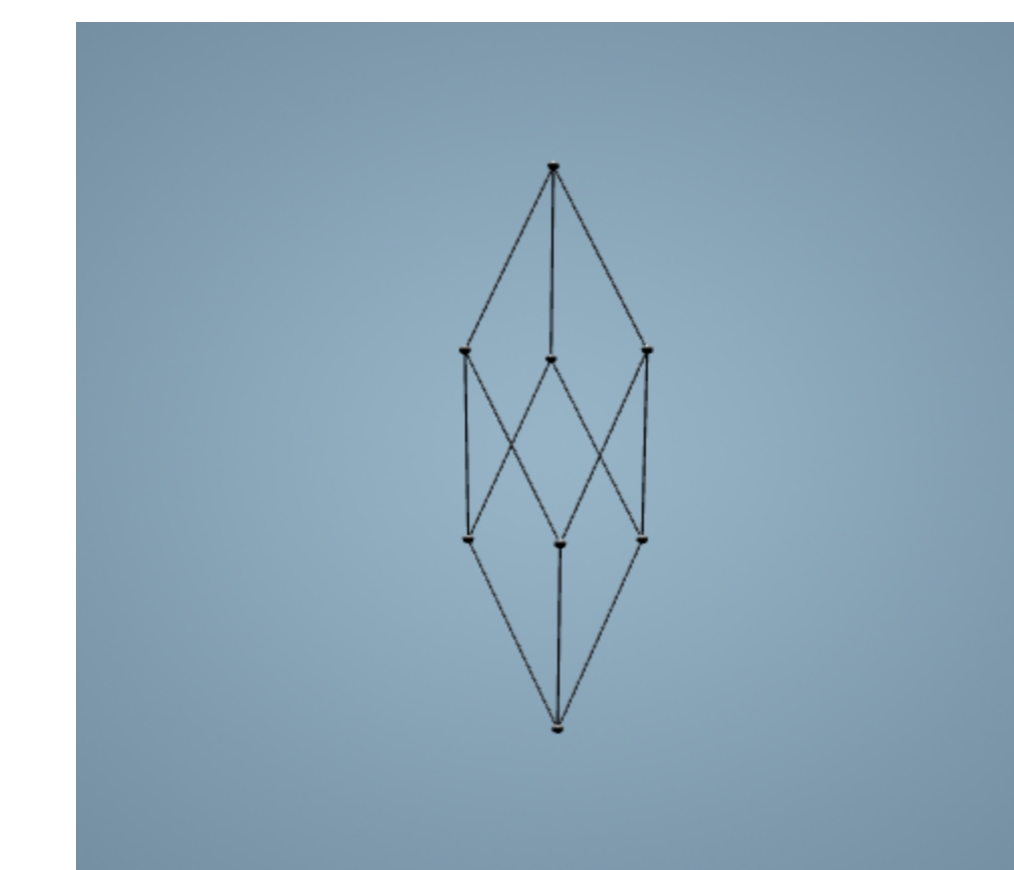
Babylon.js is an open-source javascript web framework which is used to develop 3D applications [4]. It contains most of the necessary tools to create and manage 3D objects, such as in our case lattices.

This framework combined with a function that we build, allows us to translate the output from the previous page [3] to construct 3D models of the lattices. Which leaves us with our final step to visualize these models in virtual reality.

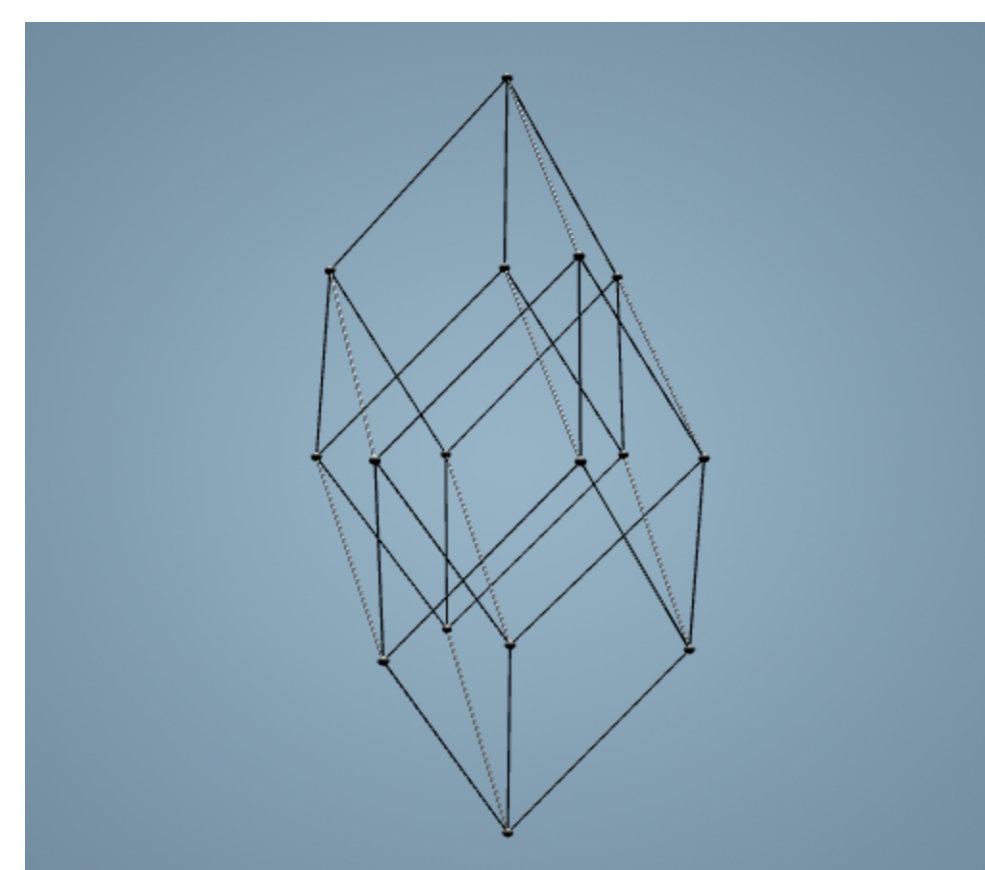
After building the 3D models of the lattices, our goal is to visualize them in virtual reality. For that purpose, we use **WebXR**.

Lattices in VR

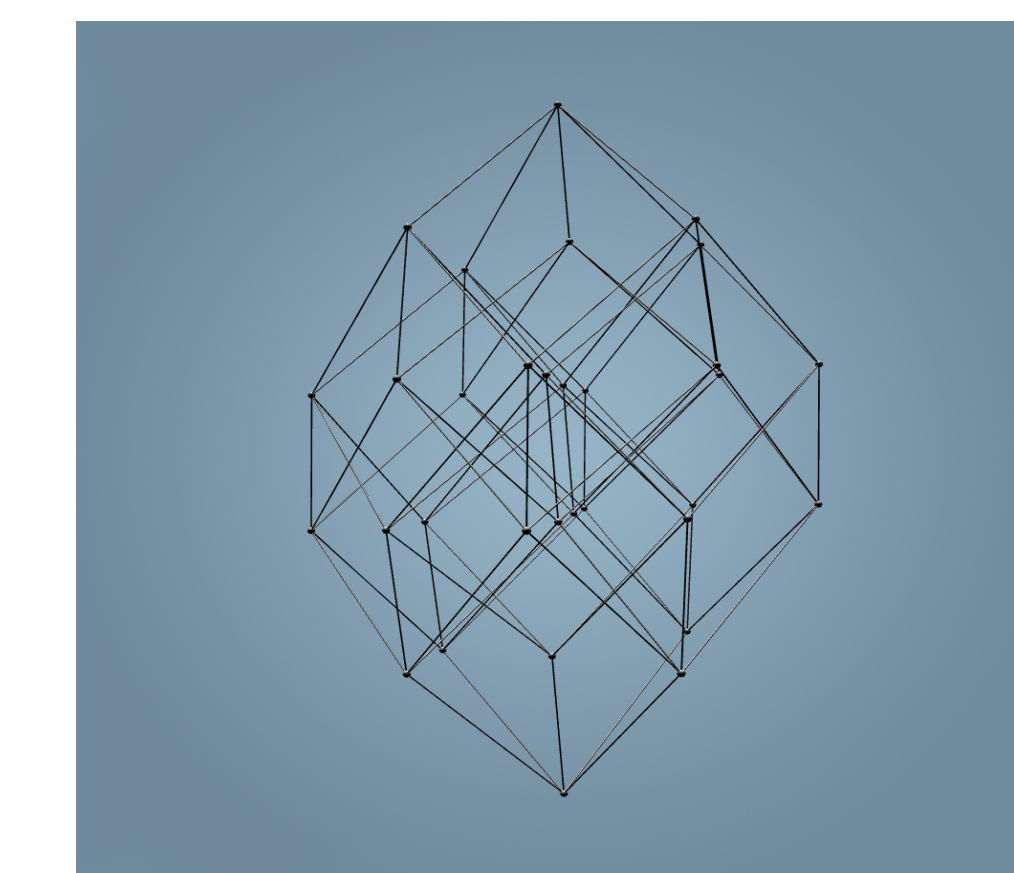
Combining this all allows us to visualize lattices in virtual reality. Since the main goal of the project was to help students and new researchers in the area of Abstract Algebra. We created a website [2] easily accessible in which the 3D models of the lattices can either be interacted directly within the browser, or via an external VR headset.



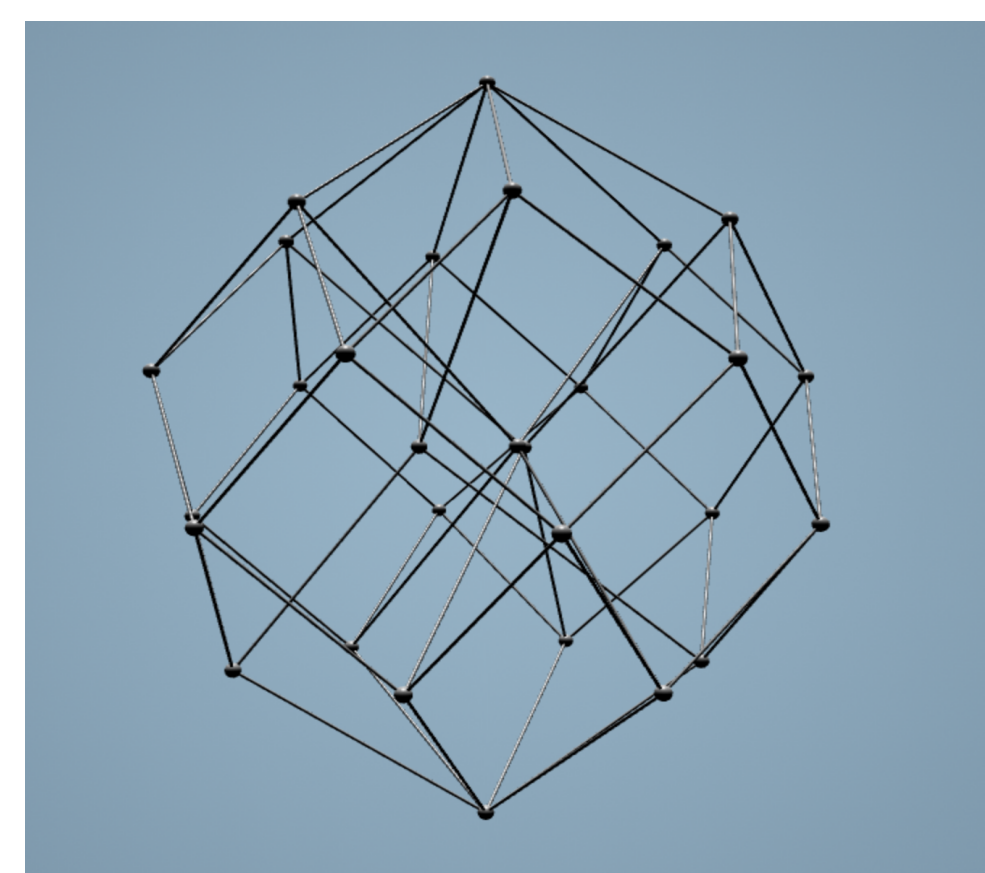
(a) Cube: 8 element.



(b) Hypercube 4: 16 element.



(c) Hypercube 5: 32 Element.



(d) 3-Element preorder lattice: 29 Element.

Figure: Some lattices viable in virtual reality at [2].

Conclusion

From the project, we found a way to visualize and help understand abstract mathematical objects by displaying them in a 3D environment that can be viewed from a standard web browser and with a VR headset. We are also exploring the possibility of creating 3D printed models of some of the lattices that have been constructed.

References

- [1] N. Alpay and P. Jipsen, Commutative doubly-idempotent semirings determined by chains and by preorder forests, in proceedings of the 18th International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS), Lecture Notes in Computer Science Vol 12062, Springer (2020), 1-14
- [2] N. Alpay and P. Jipsen. <https://alpaynatanael.github.io/VRLattices/>
- [3] P. Jipsen. <https://www1.chapman.edu/~jipsen/FCA/hasse.html>
- [4] www.babylonjs.com.
- [5] <https://immersive-web.github.io/webxr-samples/>