Operation-Based Notation for Archimedean Graph

Hidetoshi Nonaka
Research Group of Mathematical Information Science
Division of Computer Science, Hokkaido University
N14W9, Sapporo, 060 0814, Japan

ABSTRACT

We introduce three graph operations corresponding to polyhedral operations. By applying these operations, thirteen Archimedean graphs can be generated from Platonic graphs that are used as seed graphs.

Keyword: Archimedean graph, Polyhedral graph, Polyhedron notation, Graph operation.

1. INTRODUCTION

Archimedean graph is a simple planar graph isomorphic to the skeleton or wire-frame of the Archimedean solid. There are thirteen Archimedean solids, which are semi-regular polyhedra and the subset of uniform polyhedra. Seven of them can be formed by truncation of Platonic solids, and all of them can be formed by polyhedral operations defined as Conway polyhedron notation [1-2].

The author has recently developed an interactive modeling system of uniform polyhedra including Platonic solids, Archimedean solids and Kepler-Poinsot solids, based on graph drawing and simulated elasticity, mainly for educational purpose [3-5]. Obviously, it is possible to draw a polyhedral graph only with two graph operations: *vertex addition* and *edge addition*. However there is an individual difference, it might be time-consuming to input a graph with steady steps, especially in the case of Archimedean graph.

In this paper, three graph operations corresponding to polyhedral operations are introduced, and it is shown that every Archimedean graph can be generated using these operations and a Platonic graph as the seed graph.

2. ARCHIMEDEAN SOLIDS

Thirteen Archimedean solids are listed in Table 1, and illustrated in Figures 1-2. The symbols in the table stand for the vertex configurations. For example, $A_{(3\cdot4)^2}$ indicates that two regular triangles and two squares are gathered alternately on each vertex. Archimedean solids are surrounded by several sorts of congruent regular polygons, and their vertex figures are not regular but congruent polygons. The term *vertex figure* was introduced by H. Coxeter as the segment joining the mid-point of the two sides through a vertex [6].

Table 1. The list of Archimedean solids, where p, q, r are the number of vertices, edges, and faces, respectively.

Symbol	Name of polyhedron	p	q	r
$A_{(3\cdot4)^2}$	Cuboctahedron	12	24	14
$A_{4\cdot6\cdot10}$	Great Rhombicosidodecahedron	120	180	62
$A_{4\cdot6\cdot8}$	Great Rhombicuboctahedron	48	72	26
$A_{(3\cdot5)^2}$	Icosidodecahedron	30	60	32
$A_{3\cdot4\cdot5\cdot4}$	Small Rhombicosidodecahedron	60	120	62
$A_{3\cdot4^3}$	Small Rhombicuboctahedron	24	48	26
$A_{3^4\cdot4}$	Snub Cube	24	60	38
$A_{3^4\cdot5}$	Snub Dodecahedron	60	150	92
$A_{3\cdot8^2}$	Truncated Cube	24	36	14
$A_{3\cdot10^2}$	Truncated Dodecahedron	60	90	32
$A_{5\cdot6^2}$	Truncated Icosahedron	60	90	32
$A_{4\cdot6^2}$	Truncated Octahedron	24	36	14
$A_{3\cdot6^2}$	Truncated Tetrahedron	12	18	8

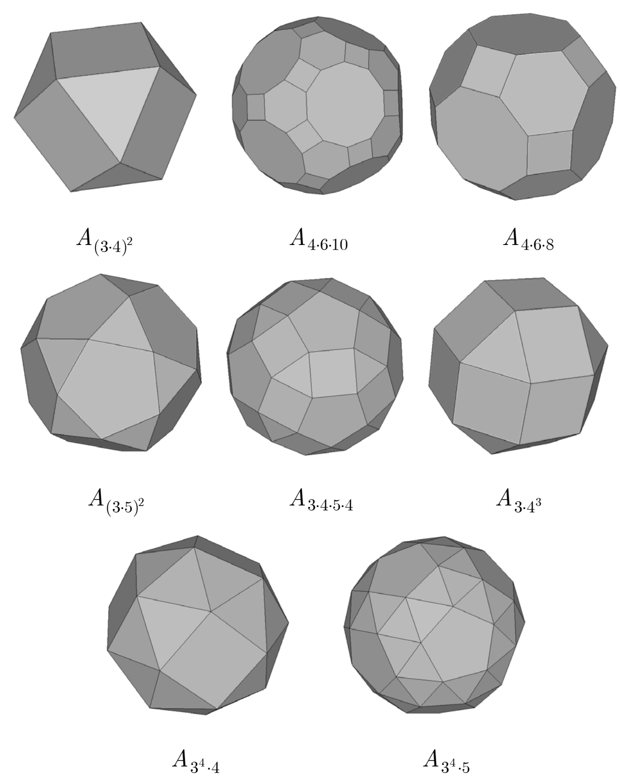


Figure 1. Appearance of thirteen Archimedean solids (1/2).

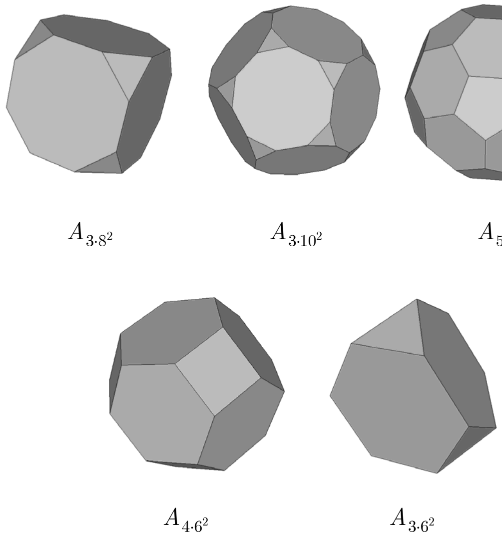


Figure 2. Appearance of thirteen Archimedean solids (2/2).

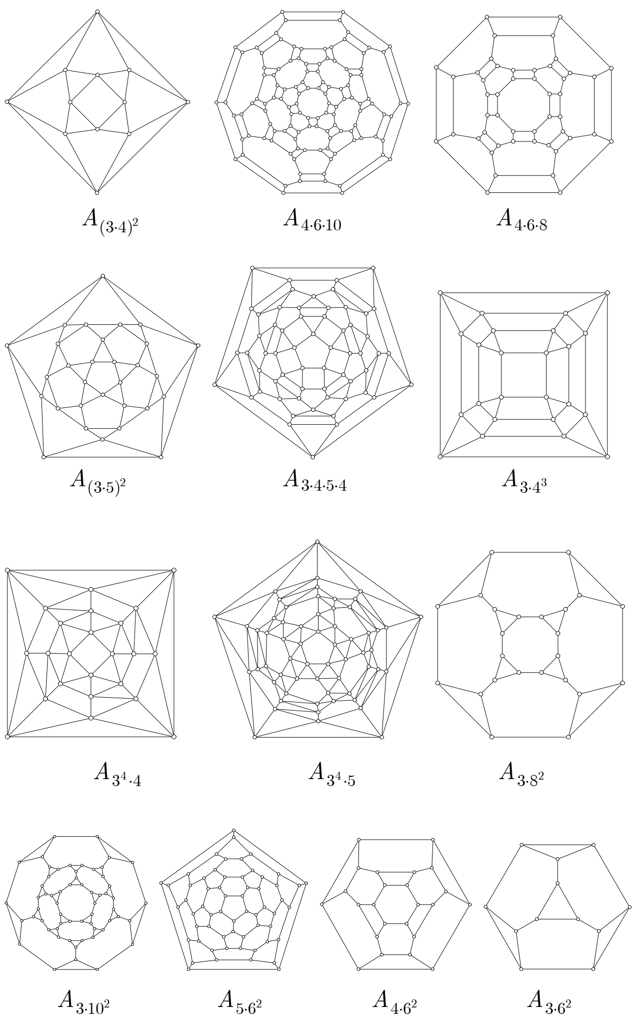


Figure 3. Thirteen Archimedean graphs drawn as plane graphs.

3. GRAPH OPERATIONS

Thirteen Archimedean graphs isomorphic to the skeletons of Archimedean solids are depicted in Figure 3. A graph $G = (V, E)$ is defined by the set of vertices $V = \{v_0, \cdots, v_{p-1}\}$ and edges $E = \{e_0, \cdots, e_{q-1}\}$. All graphs considered in this paper are 3-, 4-, or 5-regular polyhedral graphs, which are simple, planar and 3-connected graphs [7]. Detection of faces with n sides is equivalent to finding n-cycles. After selecting faces $F = \{f_0, \cdots, f_{r-1}\}$, a polyhedral graph is redefined as $\tilde{G} = \{V, E, F\}$ [3]. The set of faces is subdivided as follows,

$$F = F_3 \cup F_4 \cup F_5 \cup F_6 \cup F_8 \cup F_{10},\ \forall i \neq j : F_i \cap F_j = \phi ,$$

where F_n denotes the set of faces with n sides.

We define three graph operations for $\tilde{G} = \{V, E, F\}$: *diagonal addition*, *edge contraction*, and *vertex splitting*. Examples are shown in Figures 4-6. A *diagonal addition* to a face in F_4 is an edge addition between a pair of non-adjacent vertices in a quadrangular face. An *edge contraction* is a graph contraction of an edge. A *vertex splitting* is defined conventionally as the reverse of edge contraction, but in this paper, we define a *vertex splitting* of v as the composition of the operations of subdivision of incident edges on v, connecting the new vertices in a proper order, and deleting the vertex v. Conventional *vertex splitting* is equivalent to the present *vertex splitting* followed by several *edge contractions*.

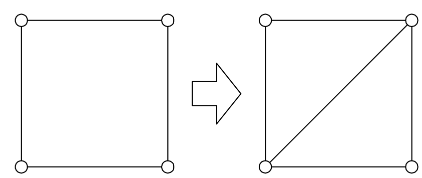


Figure 4. An example of *diagonal addition* operations.

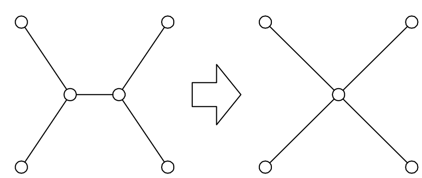


Figure 5. An example of *edge contraction* operations.

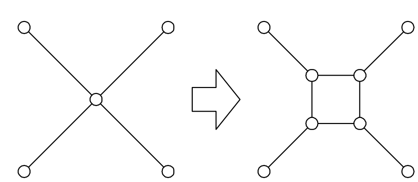


Figure 6. An example of *vertex splitting* operations by the present definition in this paper.

4. OPERATION-BASED NOTATION OF ARCHIMEDEAN GRAPH

The symbol δ stands for applying *diagonal addition* to each quadrangular face of F_4 which is adjacent to two triangular faces of F_3. The operator $\llbracket m,n \rrbracket (= \llbracket n,m \rrbracket)$ stands for applying *edge contraction* to each edge incident on two faces of F_m and F_n. The operator σ stands for applying *vertex splitting* to every vertex.

The operator "truncate" in the Conway polyhedron notation [1-2] corresponds to the graph operation σ. The operator "ambo", "expand" and "snub" are expressed by $\llbracket m,n \rrbracket \sigma$, $\llbracket k,l \rrbracket \sigma \llbracket m,n \rrbracket \sigma$, and $\delta \llbracket k,l \rrbracket \sigma \llbracket m,n \rrbracket \sigma$, respectively. In the case of Archimedean graph, equalities $l = m = n$ hold. We use following notations for Platonic graphs and Platonic solids: cube P_{4^3}, dodecahedron P_{5^3}, icosahedron P_{3^5}, octahedron P_{3^4}, and tetrahedron P_{3^3}.

As a consequence, every Archimedean graph can be expressed using one Platonic graph and graph operations as follows.

$$
\begin{aligned}
&A_{3\cdot6^2} = \sigma P_{3^3},\\
&A_{4\cdot6^2} = \sigma P_{3^4} = \llbracket 4,8 \rrbracket \sigma \llbracket 8,8 \rrbracket \sigma P_{4^3},\\
&A_{3\cdot8^2} = \sigma P_{4^3} = \llbracket 4,6 \rrbracket \sigma \llbracket 6,6 \rrbracket \sigma P_{3^4},\\
&A_{(3\cdot4)^2} = \llbracket 6,6 \rrbracket \sigma P_{3^4} = \llbracket 8,8 \rrbracket \sigma P_{4^3},\\
&A_{4\cdot6\cdot8} = \sigma A_{(3\cdot4)^2} = \sigma \llbracket 6,6 \rrbracket \sigma P_{3^4} = \sigma \llbracket 8,8 \rrbracket \sigma P_{4^3},\\
&A_{3\cdot4^3} = \llbracket 6,8 \rrbracket A_{4\cdot6\cdot8} = \llbracket 8,6 \rrbracket \sigma \llbracket 6,6 \rrbracket \sigma P_{3^4} = \llbracket 6,8 \rrbracket \sigma \llbracket 8,8 \rrbracket \sigma P_{4^3},\\
&A_{3^4\cdot4} = \delta A_{3\cdot4^3} = \delta \llbracket 8,6 \rrbracket \sigma \llbracket 6,6 \rrbracket \sigma P_{3^4} = \delta \llbracket 6,8 \rrbracket \sigma \llbracket 8,8 \rrbracket \sigma P_{4^3},\\
&A_{5\cdot6^2} = \sigma P_{3^5} = \llbracket 4,10 \rrbracket \sigma \llbracket 10,10 \rrbracket \sigma P_{5^3},\\
&A_{3\cdot8^2} = \sigma P_{5^3} = \llbracket 4,6 \rrbracket \sigma \llbracket 6,6 \rrbracket \sigma P_{3^5},\\
&A_{(3\cdot5)^2} = \llbracket 6,6 \rrbracket \sigma P_{3^5} = \llbracket 10,10 \rrbracket \sigma P_{5^3},\\
&A_{4\cdot6\cdot10} = \sigma A_{(3\cdot5)^2} = \sigma \llbracket 6,6 \rrbracket \sigma P_{3^5} = \sigma \llbracket 10,10 \rrbracket \sigma P_{5^3},\\
&A_{3\cdot4\cdot5\cdot4} = \llbracket 6,10 \rrbracket A_{4\cdot6\cdot10} = \llbracket 10,6 \rrbracket \sigma \llbracket 6,6 \rrbracket \sigma P_{3^5}\\
&\qquad = \llbracket 6,10 \rrbracket \sigma \llbracket 10,10 \rrbracket \sigma P_{5^3},\\
&A_{3^4\cdot5} = \delta A_{3\cdot4\cdot5\cdot4} = \delta \llbracket 10,6 \rrbracket \sigma \llbracket 6,6 \rrbracket \sigma P_{3^5} = \delta \llbracket 6,10 \rrbracket \sigma \llbracket 10,10 \rrbracket \sigma P_{5^3},
\end{aligned}
$$

$$
\begin{aligned}
&\llbracket 3,6 \rrbracket A_{3\cdot6^2} = P_{3^3}, \quad \llbracket 6,6 \rrbracket A_{3\cdot6^2} = P_{3^4},\\
&\llbracket 3,8 \rrbracket A_{3\cdot8^2} = P_{4^3}, \quad \llbracket 3,10 \rrbracket A_{3\cdot10^2} = P_{5^3},\\
&\llbracket 3,4 \rrbracket A_{3\cdot4^3} = P_{4^3}, \quad \llbracket 4,4 \rrbracket A_{3\cdot4^3} = \llbracket 3,4 \rrbracket A_{3^4\cdot4} = P_{3^4},\\
&\llbracket 5,6 \rrbracket A_{5\cdot6^2} = P_{3^5}, \quad \llbracket 4,6 \rrbracket A_{3\cdot8^2} = P_{5^3},\\
&\llbracket 3,4 \rrbracket A_{3\cdot4\cdot5\cdot4} = P_{5^3}, \quad \llbracket 4,5 \rrbracket A_{3\cdot4\cdot5\cdot4} = \llbracket 3,5 \rrbracket A_{3^4\cdot5} = P_{3^5},\\
&\llbracket 3,4 \rrbracket A_{3^4\cdot4} = P_{3^4}, \quad \llbracket 3,5 \rrbracket A_{3^4\cdot5} = P_{3^5}, \quad \delta A_{(3\cdot4)^2} = P_{3^5}
\end{aligned}
$$

Figure 7 shows the relations of 5 Platonic graphs and 13 Archimedean graphs. There are five more possibilities of *edge contraction* operations to Archimedean graphs: $\llbracket 3,4 \rrbracket A_{(3\cdot4)^2}$, $\llbracket 3,5 \rrbracket A_{(3\cdot5)^2}$, $\llbracket 4,4 \rrbracket A_{3\cdot4^3}$, $\llbracket 3,3 \rrbracket A_{4\cdot3^4}$, and $\llbracket 3,3 \rrbracket A_{5\cdot3^4}$, which lead to trivial graph with one isolated vertex.

5. ADDITION OF THE GRAPH OPERATIONS TO AN INTERACTIVE MODELING SYSTEM

An interactive modeling system has been developed by the author [3-5]. The coordinate of vertices are computed without the knowledge of metric information, but only with the structure of the isomorphic polyhedral graph. The system consists of three subsystems: graph input subsystem, wire-frame subsystem, and polygon subsystem. The overviews of the subsystems are described in the continuing subsections.

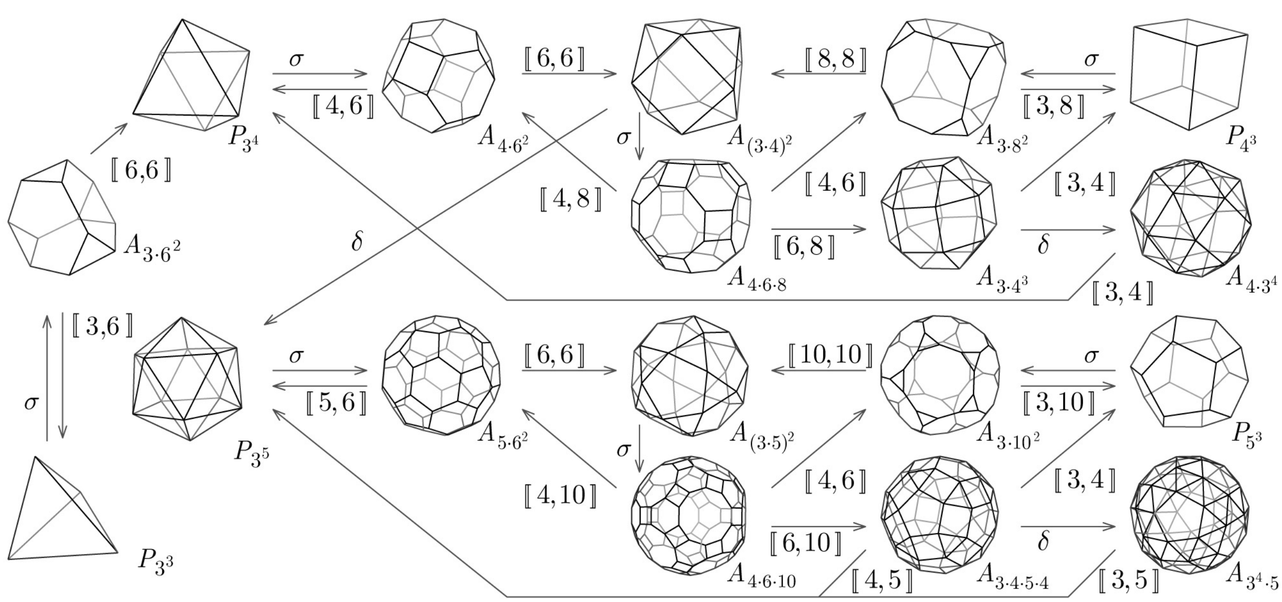


Figure 7. Relations of 5 Platonic graphs and 13 Archimedean graphs induced by *diagonal addition*, *edge contraction* and *vertex splitting*.

5.1 Graph Input Subsystem

Figure 8(a) shows a screen shot of GUI of graph input subsystem. Drawing a planar graph isomorphic to polyhedron is the first step of polyhedron modeling. In the subsystem, *vertex addition*, *vertex deletion*, *edge addition*, and *edge deletion* are implemented as fundamental operations. Vertices can be moved attended with the incident edges. In addition, *edge contraction* and *vertex splitting* are introduced.

5.2 Wire-Frame Subsystem

Figure 8(b) shows a screen shot of GUI of wire-frame subsystem. After constructing a polyhedral graph, the next step is arranging vertices in 3-dimesional space. We define three binary relations between two vertices: *adjacent*, *neighbor*, and *diameter*. The relation *adjacent* corresponds to the length of an edge in a 3-dimensional space. The relation *neighbor* means that the length of path between two vertices is 2, and two vertices are neighborhood of another vertex. It corresponds to the shape of vertex figure in a 3-dimentional space. The relation *diameter* means that the length of path between two vertices is the diameter of the graph. It corresponds to the circum-sphere of polyhedron. Virtual elastic forces are assumed between vertices according to these three relations and Hooke's law. Wire-frame polyhedron can be formed semi-automatically by controlling the natural length of virtual springs corresponding to the three types of binary relations.

5.3 Polygon Subsystem

Figure 8(c) shows a screen shot of GUI of polygon subsystem. After arranging vertices in 3-dimensional space, the last step is detecting faces, selecting faces, and rendering the solid [3]. In the case of Platonic solids, Archimedean solids, prisms, and anti-prisms, common routine is used. The faces of Kepler-Poinsot solids are detected by separate routine.

5.4 Addition of the graph operations

The graph operations of *vertex splitting* and *edge contraction* have been already implemented in the previous version of the graph input subsystem, but they are applied to individual vertex and edge respectively. In the present version, simultaneous *vertex splitting* of all the vertices is realized, and simultaneous *edge contraction* of all the edges between specified faces are also implemented. Such operations require 3-dimensional coordinate of each vertex and configuration of each face, therefore, they are available in the wire-frame subsystem and polygon subsystem. In these subsystems, *vertex splitting* and *edge contraction* are displayed as animation.

6. REFERENCES

[1] G. W. Hart, "Conway Notation for Polyhedra", URL: http:/ /www.georgehart.co/virtual-polyhedra/ conway_notation.html.

[2] E.W.Weisstein,"Conway Polyhedron Notation", MathWorld.

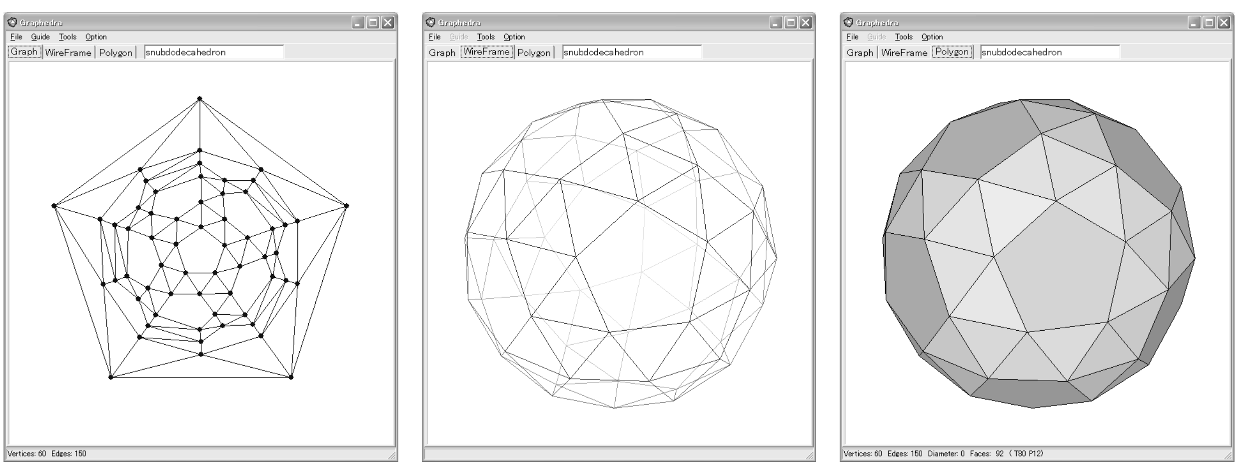
[3] H. Nonaka, "Detection of Faces in Wire-Frame Polyhedra", Proceedings of Int. Conf. on Computer Graphics Theory and Applications (GRAPP 2008), 134-137, Madeira, Jan. 2008.

[4] H. Nonaka, "Modeling of Kepler-Poinsot Solid Using Isomorphic Polyhedral Graph", International Journal of Computer Science, **3**, 1, 64-67, 2008. URL: http:/www.waset.org/ijcs/former.html

[5] H. Nonaka, "Interactive Modeling of Uniform Polyhedra by Means of Graph Drawing", Proceedings of Int. Conf. on Information Technology and Applications (ICITA 2008), 822-827, Cairns, June 2008.

[6] H. S. M. Coxeter, "Regular Polytopes", Dover Pub., 1973.

[7] Branko Grünbaum, "Graphs of polyhedra; polyhedra as graphs", Discrete Mathematics, **307**, 445-463, 2007.



(a) Graph input subsistem (b) Wire-frame subsystem (c) Polygon subsystem

Figure 8. Screen shot of GUI of the interactive modeling system.