QUICKHULL for Sets of Standard Quadruples

Wolfgang W. Bein * Lawrence L. Larmore †

Abstract
Finding an independent set.

1 Definitions
Throughout, we assume that $C > 0$ is a constant.

1.1 Standard and Co-standard Quadruples
A standard quadruple is a 4-tuple of numbers $k = (v^x, v^y, p^x, p^y)$ satisfying the following conditions:
1. $|v^x − v^y| ≤ 1$
2. $p^x ≥ 0$
3. $p^y ≥ 0$
4. $p^x + p^y = 1$

A co-standard quadruple is a 4-tuple of numbers $h = (\tilde{v}^x, \tilde{v}^y, \tilde{p}^x, \tilde{p}^y)$ satisfying the following conditions:
1. $|\tilde{v}^x − \tilde{v}^y| ≤ 1$
2. $\tilde{v}^x ≥ 0$
3. $\tilde{v}^y ≥ 0$
4. $\tilde{v}^x + \tilde{v}^y = C$

If $\{k_i\}$ are standard quadruples and $k = \sum_{i=1}^{m} \lambda_i k_i$, where each $\lambda_i ≥ 0$ and $\sum_{i=1}^{m} \lambda_i = 1$, then $k$ is a standard quadruple, and we say that $k$ is a linear combination of $\{k_i\}$. We similarly define a linear combination of co-standard quadruples.

*Department of Computer Science, University of Nevada, Las Vegas, NV 89154. Email: bein@cs.unlv.edu. Research supported by NSF grant CCR-0312093.
†Department of Computer Science, University of Nevada, Las Vegas, NV 89154. Email: larmore@cs.unlv.edu. Research supported by NSF grant CCR-0312093.
1.2 Dependency

If \( k_1, k_2 \) are standard quadruples, we say that \( k_1 \) depends on \( k_2 \) if, for some constant \( A \geq 0 \):
1. \( |p_{i1}^k - p_{i2}^k| \leq A \)
2. \( C v_{i1}^k \geq C v_{i2}^k + A \)
3. \( C v_{j1}^k \geq C v_{j2}^k + A \)

If \( h_1, h_2 \) are co-standard quadruples, we say that \( h_1 \) depends on \( h_2 \) if, for some constant \( B \geq 0 \):
1. \( |\overline{v}_{i1}^h - \overline{v}_{i2}^h| \leq B \)
2. \( \overline{p}_{i1}^h \geq \overline{p}_{i2}^h + B \)
3. \( \overline{p}_{j1}^h \geq \overline{p}_{j2}^h + B \)

We say that a standard quadruple \( k \) depends on a set \( S \) of standard quadruples if \( k \) depends on some linear combination of members of \( S \). Similarly, we say that a co-standard quadruple \( h \) depends on a set \( T \) of co-standard quadruples if \( h \) depends on some linear combination of members of \( T \).

We say that a set \( S \) of standard quadruples is closed under dependency (or just closed if the context is understood) if every standard quadruples which depends on any linear combination of members of \( S \) is a member of \( S \). Similarly, we define closed under dependency for a set of co-standard quadruples.

Given a set \( S \) of standard quadruples, we define the dependency set of \( S \) to be the set of all standard quadruples which are dependent on some linear combination of members of \( S \). We similarly define the dependency set of a set of co-standard quadruples.

We define an independent set \( I \) of standard quadruples to be any set of standard quadruples such that no member of \( I \) depends on any linear combination of other members of \( I \). We similarly define an independent set of co-standard quadruples.

**Theorem 1** If \( S \) is any finite set of standard quadruples, then \( S \) contains a unique maximal independent subset.

**Theorem 2** If \( T \) is any finite set of co-standard quadruples, then \( T \) contains a unique maximal independent subset.

1.3 Constraints and Co-Constraints

Let \( k \) be a standard quadruples and \( h \) a co-standard quadruple. If \( h \cdot k \geq 0 \), we say that \( h \) is a standard constraint of \( k \), and we also say that \( k \) is a standard co-constraint of \( h \).

If \( S \) is set of standard quadruples, we define a standard constraint of \( S \) to be a co-standard quadruple \( h \) such that \( h \cdot k \geq 0 \) for all \( k \in S \). Similarly, if \( T \) is set of co-standard quadruples, we define a standard co-constraint of \( T \) to be a standard quadruple \( k \) such that \( h \cdot k \geq 0 \) for all \( h \in T \).

Suppose \( S \) is any set of standard quadruples. We define a complete set of standard constraints of \( S \) to be a set \( H \) of standard constraints of \( S \) such that, given any standard quadruple \( k \), \( k \) is in the dependency closure of \( S \) if and only if every member of \( H \) is a standard constraint of \( k \). Similarly, Suppose \( T \) is any set of co-standard quadruples. We define a complete set of standard co-constraints of \( T \) to be a set \( K \) of standard co-constraints of \( T \) such that, given any co-standard quadruple \( h \), \( h \) is in the dependency closure of \( T \) if and only if every member of \( K \) is a standard co-constraint of \( h \).
Theorem 3: Given any non-empty finite set \( S \) of standard quadruples, then \( S \) has a unique minimal complete set of standard constraints, and this set is independent.

Theorem 4: Given any non-empty finite set \( T \) of standard quadruples, then \( T \) has a unique minimal complete set of co-standard constraints, and this set is independent.

1.4 Minimum Independent Set Problem

Primal Independent Set Problem. Given a non-empty finite set \( S \) of standard quadruples, find the unique maximal independent subset of \( S \) and the unique minimal complete set of standard constraints of \( S \).

Dual Independent Set Problem. Given a non-empty finite set \( T \) of standard quadruples, find the unique maximal independent subset of \( T \) and the unique minimal complete set of standard co-constraints of \( T \).

Theorem 5: If \( K \) is a finite independent set of standard quadruples and \( H \) is the minimal complete set of standard constraints of \( K \), then \( K \) is the minimal complete set of standard co-constraints of \( H \).

Theorem 6: The dual problem reduces to the primal problem, replacing \( C \) by \( \frac{1}{C} \).

The reduction maps the standard quadruple \((v^x, v^y, p^x, p^y)\) to the co-standard quadruple \((\frac{1}{C}p^x, \frac{1}{C}p^y, v^x, v^y)\), and the co-standard quadruple \((\bar{v}^x, \bar{v}^y, \bar{p}^x, \bar{p}^y)\) to the standard quadruple \((\bar{p}^x, \bar{p}^y, \frac{1}{C}\bar{v}^x, \frac{1}{C}\bar{v}^y)\).

We remark that the minimum independent set problem is is very similar to the problem of finding the vertices and faces of the convex hull of a finite set in \( \mathbb{R}^3 \). The well-known algorithm QUICKHULL for the convex hull problem will be adapted here to solve the minimum independent set problem.

2 Proofs of Theorems

3 Description of QUICKHULL

We shall assume that we are given a non-empty finite set \( S \) of standard quadruples.

3.1 Virtually Standard Quadruples

We define a set of four quadruples that play an important role in the algorithm. We call these virtually standard quadruples.

- \((\frac{1}{C}, \frac{1}{C}, -1, 1) = k_{up}\) is the upper virtually standard quadruple.
- \((\frac{1}{C}, \frac{1}{C}, 1, -1) = k_{down}\) is the lower virtually standard quadruple.
- \((0, 1, 0, 0) = k_{left}\) is the left virtually standard quadruple.
- \((1, 0, 0, 0) = k_{right}\) is the right virtually standard quadruple.
3.2 Top Level Description of QUICKHULL

Our algorithm proceeds as follows: (The definitions of the terms used in this description will follow.)

1. Initialize. The data structure will have five vertices, the four virtually standard quadruples, which we call infinite vertices, and one additional vertex $k_0$. It will also have four faces and four edges.

2. While there is any excessive quadruple, execute BREAKFACE. The $i$th iteration of BREAKFACE adds one vertex, $k_i$, to the data structure, and also deletes and inserts faces and edges.

3. Identify which faces are external.

4. The minimal independent set will be all vertices of the data structure except for the infinite vertices, namely $\{k_0, \ldots, k_m\}$, where $m$ is the number of iterations of BREAKFACE.

5. the maximal independent set of constraints consists of all faces of the data structure, except for those which are external.

6. After any given step, the data structure may be mapped onto the extended plane, where each infinite vertex maps to a “point at infinity,” each other vertex maps to a point within the rectangle whose corners are $\{\pm 1, \pm 1\}$, and the entire plane is the union of convex polygons which are the images of the faces.

3.3 The Abstract Data Structure

Our abstract data structure consists of the following parts.

- A set $V^+$ of vertices. Each vertex is either a member of $S$ or a virtually standard quadruple. Every virtually standard quadruple is a member of $V^+$. Write $V = V^+ \cap S$.

- A set $F$ of faces. Each face is a co-standard quadruple. For all $h \in F$ and all $k \in V^+$, $h \cdot k \geq 0$. If $h \in F$ and $k \in V$, we say that $h$ is adjacent to $k$ if $h \cdot k = 0$.

- We say that any $k \in S$ is dead if $k \not\in V$ and $h \cdot v \geq 0$ for all $h \in F$.

- We say that any $k \in S$ is excessive if $h \cdot k < 0$ for some $h \in F$. We define the excess of $k$ to be $\max_{h \in F} \{-h \cdot k\}$.

- A regular edge is a set of two vertices and two faces, such that each of the two faces is adjacent to each of the two vertices. At most one of those vertices may be a virtually standard quadruple. We say those faces and those vertices are adjacent to that edge. We also say that those two faces are adjacent to each other and that the two vertices are adjacent to each other.

- An outer edge is a set of two virtually standard quadruples and one face. The face is adjacent to each of the two vertices. We say that the two vertices are adjacent to each other, and that both are adjacent to that face, and that all three are adjacent to that edge.

- There are four outer edges, one adjacent to $k_{up}$ and $k_{left}$, one adjacent to $k_{down}$ and $k_{left}$, one adjacent to $k_{up}$ and $k_{right}$, and one adjacent to $k_{down}$ and $k_{right}$.
• Each face has a boundary which is a cycle of alternating edges and vertices. We assume that the cycle is counter-clockwise. The boundary contains the same number of edges as vertices, and that number is at least 3.

3.4 Initialization

Initially, $F$ has four members, and $V$ has five members, there are four outer edges, and four regular edges.

• Let $k_0$ be any member of $S$ that is independent of $S - \{k_0\}$.

• Let $V = \{k_0\}$

• Let $F$ be the set consisting of the unique co-standard quadruple which is adjacent to $\{\text{up}, k_\text{left}, k_0\}$, the unique co-standard quadruple which is adjacent to $\{\text{up}, k_\text{right}, k_0\}$, the unique co-standard quadruple which is adjacent to $\{\text{down}, k_\text{left}, k_0\}$, and the unique co-standard quadruple which is adjacent to $\{\text{down}, k_\text{right}, k_0\}$.

• For each $k \in S$ other than $k_0$, if the excess of $k$ is less than or equal to zero, $k$ is dead. Otherwise, $k$ is excessive.

3.5 The Procedure BREAKFACE

The $i^{th}$ iteration of BREAKFACE adds one new vertex, $k_i$, to $V$. What follows is a high-level description of that procedure.

Note that an edge is adjacent to a face, or two faces are adjacent, if they are both adjacent to the same two vertices. One vertex in common is not enough.

• Choose $k_i$ to be the most excessive item. If there are multiple items with maximum excess, it is necessary to ensure that $k_i$ is not a linear combination of any of the others.

• Partition $F$ as follows.
  
  – A face $h$ is exposed if $h \cdot k_i < 0$.
  – A face $h$ is critical if $h \cdot k_i = 0$.
  – A face $h$ is hidden if $h \cdot k_i > 0$.

• A vertex $k \in V^+$ will be called visible if it is adjacent to a exposed face. Other vertices will be called passive.

• Partition the set of edges as follows.
  
  – An edge is exposed if it is adjacent to two exposed faces.
  – An edge is critical if it is adjacent to one exposed face and one critical face.
  – An edge is active if it is adjacent to one exposed face and one hidden face, or if it is an outer edge adjacent to one exposed face.
  – All other edges are passive.

• For each visible vertex $k$, a new edge will be created which is adjacent to $k$ and to $k_i$. 

5
• If \( k, k' \in V^+ \) are vertices adjacent to the same active edge, a new face will be created whose adjacent vertices are \( k_i, k, \) and \( k' \). Its three edges will be that active edge and the two new edges which are adjacent to \( k \) and \( k' \), respectively.

• Each critical face \( h \in F \) has one critical edge, and its other edges are passive. That critical edge will be deleted, and \( h \) will acquire two new edges, the edges which contains the two vertices adjacent to that critical edge, and one new vertex, \( k_i \).

• All exposed faces and exposed edges will be deleted.

• \( k_i \) will be added to \( V \). All new edges and all new faces will be added to the data structure.

• New excesses will be calculated. Other than \( k_i \), formerly excessive items whose new excesses are not positive will be dead.

3.6 External Faces

Define four sets \( Q_{up}, Q_{down}, Q_{left}, Q_{right} \subseteq \mathbb{R}^4 \), which we call external sets, as follows:

• \( Q_{up} = \{(v^x, v^y, p^x, p^y)|p^x \leq 0\} \). Note that \( k_{up}, k_{left}, k_{right} \in Q_{up} \).

• \( Q_{down} = \{(v^x, v^y, p^x, p^y)|p^y \leq 0\} \). Note that \( k_{down}, k_{left}, k_{right} \in Q_{down} \).

• \( Q_{left} = \{(v^x, v^y, p^x, p^y)|v^y - v^x \geq p^x + p^y\} \). Note that \( k_{left}, k_{up}, k_{down} \in Q_{left} \).

• \( Q_{right} = \{(v^x, v^y, p^x, p^y)|v^y - v^x \geq p^x + p^y\} \). Note that \( k_{right}, k_{up}, k_{down} \in Q_{right} \).

We say that a face \( h \in F \) is external if there is one external set \( Q \) such that all of the vertices in \( V \) adjacent to \( h \) lie in \( Q \).

4 Mapping onto the Extended Plane

We define the extended plane to be \( \mathbb{R}^2 \) together with the “circle at infinity” which consists of one point for each bundle of parallel directed lines.\(^1\)

We map the standard quadruples and the virtual standard quadruples onto the extended plane as follows:

• The standard quadruple \((v^x, v^y, p^x, p^y)\) is mapped to \((v^x - v^y, p^y - p^x)\).

• The virtually standard quadruple \(k_{up}\) is mapped to the infinite point \((0, \infty)\).

• The virtually standard quadruple \(k_{down}\) is mapped to the infinite point \((0, -\infty)\).

• The virtually standard quadruple \(k_{left}\) is mapped to the infinite point \((-\infty, 0)\).

• The virtually standard quadruple \(k_{right}\) is mapped to the infinite point \((\infty, 0)\).

\(^1\)Not to be confused with the projective plane, which contains one infinite point for each bundle of parallel lines. The extended plane has two infinite points for each bundle of parallel lines.
Our abstract data structure gives a partition of the extended plane into polygons. Each vertex maps to a point in the extended plane, and each edge maps to the straight line segment connecting the images of its two vertices. If one vertex of an edge is a virtually standard quadruple, that edge maps to a semi-infinite vertical or horizontal line. The four external edges each map to one quarter of the circle at infinity. Each face maps to the polygon bounded by the images of the edges adjacent to it, and each such polygon is convex.

Let $R \subseteq \mathbb{R}^2$ be the rectangle whose corners are $(\pm 1, \pm 1)$. Then

- All images of standard quadruples lie in $R$.
- A face is external if and only if the image of its interior is entirely outside $R$.

![Diagram](image)

Figure 1: An application example of mapping the data structure to the plane

Figure 1 shows an example that arises out of an actual application. In this case $C = \frac{\sqrt{13}}{12}$. The Independent set of standard quadruples is given in the following table:

<table>
<thead>
<tr>
<th>name</th>
<th>$v^x$</th>
<th>$v^y$</th>
<th>$p^x$</th>
<th>$p^y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$kF$</td>
<td>$-\frac{1}{13}C$</td>
<td>$-\frac{4}{13}C$</td>
<td>$1$</td>
<td>$0$</td>
</tr>
<tr>
<td>$kG$</td>
<td>$-\frac{1}{13}C$</td>
<td>$-\frac{2}{13}C$</td>
<td>$\frac{2}{3} - C$</td>
<td>$-\frac{2}{3} + C$</td>
</tr>
<tr>
<td>$kH$</td>
<td>$1$</td>
<td>$0$</td>
<td>$0$</td>
<td>$1$</td>
</tr>
<tr>
<td>$kI$</td>
<td>$-\frac{1}{13}C$</td>
<td>$-\frac{2}{13}C$</td>
<td>$\frac{2}{3} + \frac{3}{23}C$</td>
<td>$-\frac{2}{3} - \frac{3}{23}C$</td>
</tr>
<tr>
<td>$kJ$</td>
<td>$-\frac{1}{13}C$</td>
<td>$-\frac{2}{13}C$</td>
<td>$\frac{2}{3} - \frac{2}{23}C$</td>
<td>$-\frac{2}{3} + \frac{2}{23}C$</td>
</tr>
<tr>
<td>$kK$</td>
<td>$-\frac{1}{13}C$</td>
<td>$-\frac{2}{13}C$</td>
<td>$\frac{2}{3} + \frac{1}{23}C$</td>
<td>$-\frac{2}{3} - \frac{1}{23}C$</td>
</tr>
<tr>
<td>$kL$</td>
<td>$-\frac{1}{13}C$</td>
<td>$-\frac{2}{13}C$</td>
<td>$\frac{2}{3} - \frac{1}{23}C$</td>
<td>$-\frac{2}{3} + \frac{1}{23}C$</td>
</tr>
<tr>
<td>$kM$</td>
<td>$-\frac{1}{13}C$</td>
<td>$-\frac{2}{13}C$</td>
<td>$0$</td>
<td>$1$</td>
</tr>
</tbody>
</table>
The minimal complete set of constraints for the example illustrated in Figure 1 is given by the following table:

<table>
<thead>
<tr>
<th>name</th>
<th>$\bar{a}^x$</th>
<th>$\bar{a}^y$</th>
<th>$\bar{p}^x$</th>
<th>$\bar{p}^y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$hF$</td>
<td>$\frac{1}{4}C$</td>
<td>$\frac{3}{4}C$</td>
<td>$-3 + C$</td>
<td>$-2 + C$</td>
</tr>
<tr>
<td>$hG$</td>
<td>0</td>
<td>$C$</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>$hH$</td>
<td>$\frac{4}{3}C$</td>
<td>$\frac{1}{3}C$</td>
<td>$-2 + \frac{2}{3}C$</td>
<td>$-1 + \frac{1}{3}C$</td>
</tr>
<tr>
<td>$hI$</td>
<td>$C$</td>
<td>0</td>
<td>$-2 + C$</td>
<td>$-3 + C$</td>
</tr>
<tr>
<td>$hJ$</td>
<td>$-\frac{13}{8} + \frac{5}{8}C$</td>
<td>$\frac{13}{8} - \frac{7}{8}C$</td>
<td>$\frac{13}{8} - \frac{3}{8}C$</td>
<td>$\frac{13}{8} - \frac{5}{8}C$</td>
</tr>
<tr>
<td>$hK$</td>
<td>$\frac{4}{3}C$</td>
<td>$\frac{2}{3}C$</td>
<td>$-\frac{5}{6} + \frac{13}{6}C$</td>
<td>$-\frac{5}{6} + \frac{13}{6}C$</td>
</tr>
<tr>
<td>$hL$</td>
<td>$\frac{1}{4}C$</td>
<td>$\frac{3}{4}C$</td>
<td>$-\frac{5}{6} + \frac{13}{6}C$</td>
<td>$-\frac{5}{6} + \frac{13}{6}C$</td>
</tr>
</tbody>
</table>

Figure 2 shows the mapping of the dual version of the same application onto the extended plane.

![Figure 2: The mapping of the dual for the same application](image)

The dual solution mapped to the extended plane. Note that the roles of vertices and faces have been exchanged.

5 Mid-level Implementation of QUICKHULL

It is helpful to think of the details of this implementation by referring to the mapping to the extended plane.

At any given time, let $E \subset S$ be the set of excessive quadruples, and let $D \subset S$ be the set of dead quadruples. Then $S$ is the exact union of $E$, $V$, and $D$, where $V$ is the set of all vertices that are not infinite vertices. The data structure must include operations which allow access to elements adjacent to any given element in the structure. We say that a set of faces is connected if there is a chain of faces connecting any two of them so that consecutive faces in the chain are adjacent.

For each $h \in F$, let $E_h \subseteq E$ be the set of vertices $k$ such that face$[k] = h$. The data structure should contain a list of the members of $E_h$ for each $h$.

After each step, initialization or BREAKFACE, each member of $k \in E$ is assigned to one
member of \( F \), namely \( \text{face}[k] \), which will be an \( h \in F \) for which \( -h \cdot k \), called the excess of \( k \), will be maximized. The excess will be positive. In case of ties, one choice of \( h \) will be picked arbitrarily. If \( k \in D + V \), then \( h \cdot k \geq 0 \) for all \( k \in F \). If \( k \in V \), then \( h \cdot k = 0 \) for any \( h \in F \) which is adjacent to \( k \). Each \( k \in V \) must be adjacent to at least three members of \( F \). \( D \) will not be stored in the data structure, as these elements will never be needed again.

A data structure for \( E \) will be maintained which permits access to the item of maximum excess, using \( v^e \) as a tie-breaker, and which allows updating of excesses and deletion of dead items.

Initialization can be accomplished by selecting \( k_0 \in S \) so as to maximize \( v^e + v^p \), using \( v^e \) as a tie-breaker.

After selecting \( k_0 \), there will be four regular edges, each of which has the vertices \( v_0 \) and one infinite vertex, four faces, each of which has the vertices \( k_0 \) and two of the infinite vertices, and the four outer edges. For each \( k \in S - \{ k_0 \} \), if \( h \cdot k \geq 0 \) for each \( h \in F \), then \( k \) will be placed in \( D \). Otherwise, \( k \) will be placed in \( E \), and \( \text{face}[k] \) will be assigned to be that \( h \in F \) such that \( -h \cdot k \) is maximized. (In case of a tie, pick \( h \) arbitrarily.) This value will be called the excess of \( k \). The data structure for \( E \) will be initialized.

For each face \( h \in F \), a counter-clockwise list of adjacent edges and vertices will be maintained.

Implementation may be aided by the following lemma:

**Lemma 1** If \( k \in E \) and \( h = \text{face}[k] \), then the image of \( k \) in the extended plane lies in the polygon which corresponds to \( h \). (Not necessarily in the interior of that polygon, however.)

BREAKFACE is executed if and only if \( E \) is not empty. It can be implemented as follows.

1. Select and delete from \( E \) the item \( k_i \) of maximum excess.
2. The set of exposed faces is connected, the set of exposed and critical faces is connected, and \( \text{face}[k_i] \) is exposed. Thus, exposed and critical edges faces, as well as visible vertices, can be efficiently identified.
3. Alter the adjacencies for each critical face. Each critical face will lose one edge, gain two new edges, and gain one vertex.
4. Let \( E' \) be the set of all \( k \in E \) for which \( \text{face}[k] \) is exposed. Sort \( E' \) by the angle that each \( k \) makes with \( k_i \) in the mapping to the extended plane, in the counter-clockwise direction. (\( E' \) is a temporary set which is not saved between steps.)
5. If \( k \in E - E' \), then \( \text{face}[k] \) and the excess of \( k \) will be unchanged, and \( k \) will remain in \( E \).
6. Sort the set of new faces and critical faces, which will be the new set of faces adjacent to \( k_i \), in the counter-clockwise direction.
7. For each \( k \in E' \), either move \( k \) to \( D \) or assign \( \text{face}[k] \) to be a new face or a critical face, as appropriate, and update its excess. Update the data structure for \( E \) as required.
8. Delete all critical and exposed edges and all exposed faces.

### 6 The Figures Explained

We say that a finite vertex is a boundary vertex if its image in the extended plane is on the boundary of the rectangle \( R \). Otherwise, it is an interior vertex. If a face is not external, we say
that it is a boundary face if it is adjacent to an infinite vertex. Otherwise, we say it is an interior face. If one vertex of an edge is an infinite vertex and the other is a boundary vertex, we say that that edge is external. If both vertices of an edge are boundary vertices, the edge is a finite boundary edge. If one vertex is infinite and the other is an interior vertex, we say it is an infinite boundary edge. All other edges are interior edges.

For example, in Figure 1, $kL$ is an interior vertex, and $kM$ and $kH$ are boundary vertices. There is a finite boundary edge from $kM$ to $kH$, an infinite boundary edge from $kL$ to $k_{up}$, an interior edge from $kL$ to $kM$, and an external edge from $kM$ to $k_{up}$. The face $hJ$ is internal, and $hL$ is a boundary face. The unnamed face whose vertices are $kM$, $kH$, and $k_{up}$ is external.

In the dual figure, interior vertices correspond to interior faces. Boundary vertices correspond to boundary faces. Interior faces correspond to interior vertices and boundary faces correspond to boundary vertices. Interior edges correspond to interior edges, infinite boundary edges correspond to finite boundary edges, and finite boundary edges correspond to infinite boundary edges.

The duality does not extend to infinite vertices, outer edges, external faces, and external edges.