Concepts for Digram types

Matthias Dürksen

December 13, 2018

Equivalence classes

Equivalence classes for nodes

If several nodes are replaced to a Nonterminal node, then it must still be possible to identify to which inner node an edge is incident.

Fig. 1 must show which edge is incident to V and which edge is incident to X. The following figure shows which edge is incident to X.
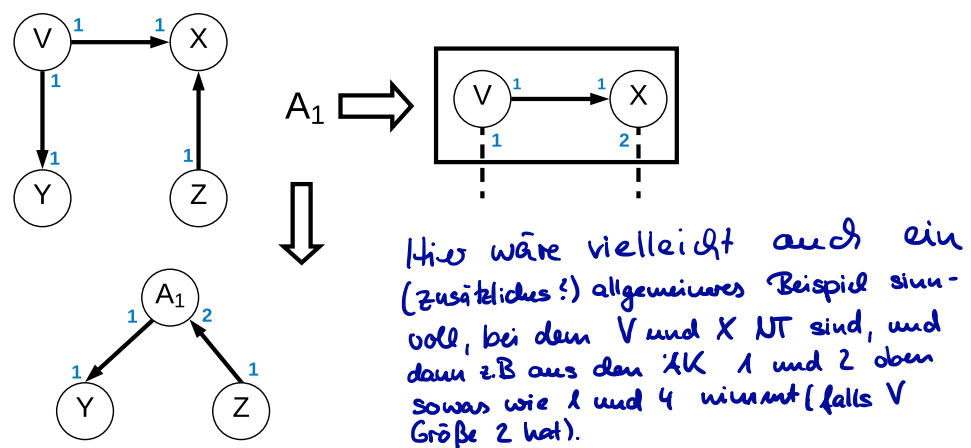

Figure 1: Visualization of equivalence classes of the nodes

Therefore, the inner nodes of a nonterminal rule are numbered consecutively, where we call the number of the inner nodes equivalence classes. Thus, incident edges of a nonterminal symbol are labeled with the equivalence class of the inner node to which the edge was incident before its replacement.

If the node is a terminal, it has equivalence class 1. To simplify matters, the equivalence classes for terminals are often not displayed in the representations.

Equivalence classes for edges

If multiple edges are replaced by one nonterminal hyper-edge, the information about which of the inner edges were incident to which node must still be available.

In the Fig. 2 we see a replacement of several edges to a hyper-edge. If there were no equivalence classes for the edges, one would not know to which node the inner edge with the label d must be incident. No information about the direction of the edge ends is required, so the hyperedge is undirected. Therefore, the edge ends are numbered, where we call the numbers equivalence classes for edges.
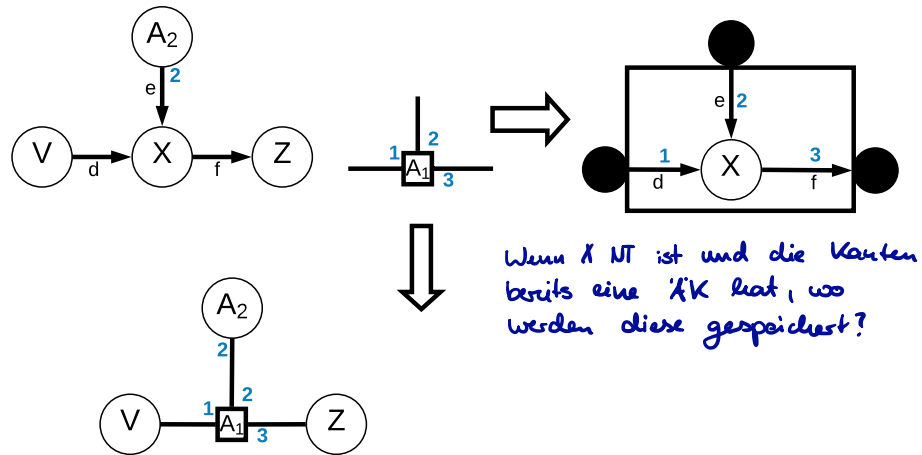

Figure 2: Visualization of equivalence classes of the edges

If two internal edges contain the same information (same direction, label and incident with the same equivalence class at the inner node), the edge ends have the same equivalence class.

In the special case that an edge is not a hyperedge, the edge has two ends. Thus the edge can be represented normally and the equivalence classes 1 and 2 can be represented over one direction of the edge, see Fig. 3. The end, which is incident to the start node, has the equivalence class 1 and the end, which is incident to the end node, has the equivalence class 2.

Figure 3: Simplified representation of an edge with equivalence classes

Parameters

It may make sense to allow a parameter for a digram. If a pattern occurs again and again, but the node label at a node, for example, varies again and again, this can be

combined into a digram and the node label can be stored as a parameter. This can significantly increase the number of occurrences found.

Any internal information of a digram rule can be parameterized. Thus the edge direction, equivalence classes, nodes/edge labels can be parameterized.

In addition, external elements can possibly become internal elements and the varying information can be parameterized. Thus, for example, the external node in the adjacency digram (Section 3) become an internal node and its node label becomes a parameter. It should be noted that the modified form of adjacency digram with double application of basic digrams (Section 1). But this digram will probably not be created only with basic digrams, because the middle node label varies and therefore no frequently occurring basic digram can be found. This characteristic will probably also occur with other variations.

1 Basic digram

The basic digram (Fig. 4) consists of two adjacent nodes and the edge that connects these nodes. Thus, two adjacent nodes and the incident edge can be replaced by one nonterminal node. Each node is assigned with an equivalence class. For each occurrence the node labels (x and y), the edge label (e), and the equivalence classes (i and j) of the inner edge must match as defined in the rule.
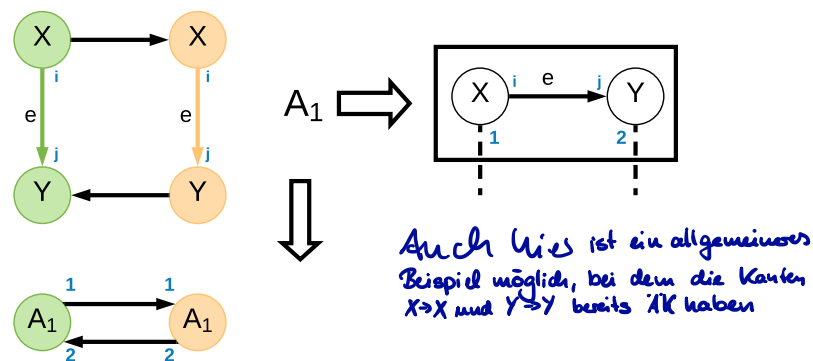

Figure 4: Application of a replacement of a basic digram

2 Circle digram

The circle digram (Fig. 5) contains a node and an edge, which is a self-cycle of the node. The node and the edge are replaced by a nonterminal node. For each occurrence the node label x, the edge label e, and the equivalence classes (i and j) of the edge must match as defined in the rule.
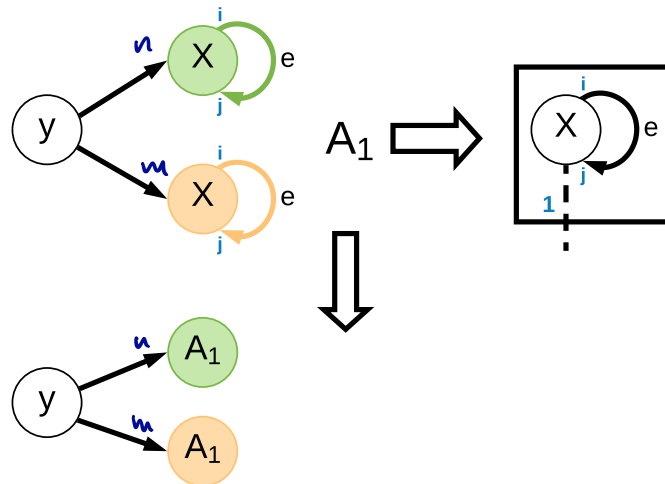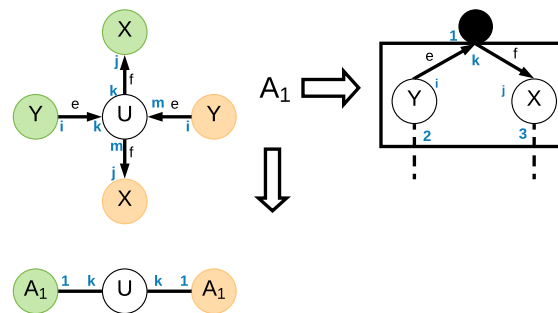
Figure 5: Application of a replacement of a circle digram

3 Adjacency digram

The adjacency digram (Fig. 6) is formed by two nodes which both have a common adjacency node U. Thus, the adjacency digram contains the two nodes and the two edges that are incident to the node U. This digram can be replaced by a nonterminal node and an edge from the nonterminal node to the node U. Each inner node and the extern node is assigned an equivalence class. The incident edges must have the same equivalence classes k at the external node, but can differ for each occurrence.

Figure 6: Application of a replacement of an adjacency digram

4 Node association digram

The node association digram (Fig. 7) is a generalization of an adjacent digram. This digram consists of two nodes, which are replaced in a substitution of an occurrence by a node. Each node is assigned with an equivalence class. In this way, frequently occurring

nodes are replaced by a nonterminal node and these do not have to be adjacent. However, this replacement increases the number of incident edges, since the new node unites the incident edges of the old nodes.
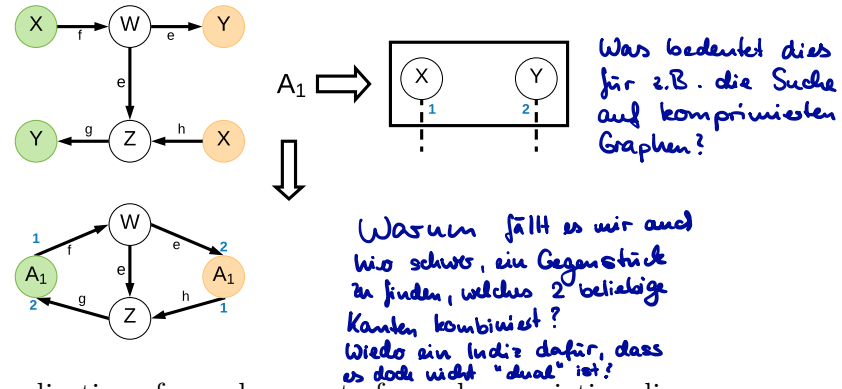

Figure 7: Application of a replacement of a node association digram

5 Clique digram

The clique digram (Fig. 8) is a generalization of an adjacency digram. This clique digram consists of two nodes, which are strongly connected. The occurrences are replaced by a node. Each node is assigned an equivalence class. The digram can also be generated by applying the basic digram 1 and then the circle digram 2. The advantage of this digram rule is that the individual edges do not have to be stored in the digram rule, because the information is implicit present. If one of the two nodes is already a nonterminal symbol, the other node should be strongly connected to each inner node.
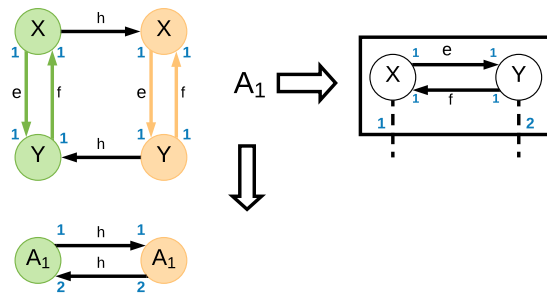

Figure 8: Application of a replacement of a clique digram

Because the digram can be reproduced by other digrams, this type of digram is not taken into account when searching for and replacing the most frequent digram, but possibly only when pruning the existing rules to more efficient rules.

6 Basic edge digram

The basic edge digram (Fig. 9) is the opposite of the basic digram. Here, two edges and a node, where both edges are incident to the node, are replaced by one edge. The directions of the edges and the edge labels must be the same in all occurrences. In addition, the equivalence classes at the inner node must match. Furthermore, the inner node must not have any other incident edges than the two internal edges.
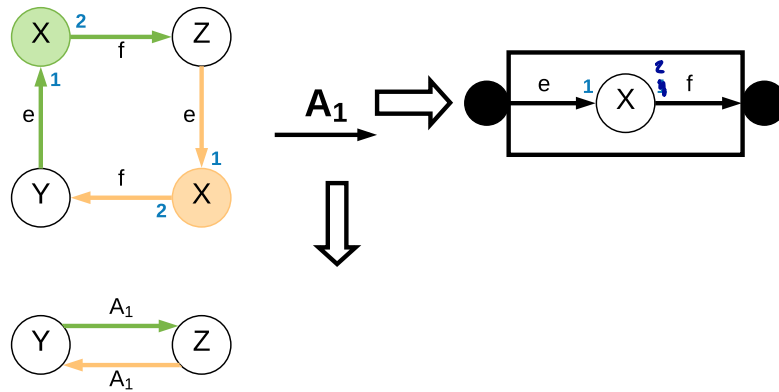
Figure 9: Application of a replacement of a basic edge digram

Incedent edges to an inner node

The basic edge digram described above only works if the inner node has no other incident edges. This is probably rarely the case in reality, so the rule has to be extended so that the node can be of any number of nodes may be incident.
I currently have two approaches, both of which have their downsides.

Case 1: All incident edges are transformed into a hyper edge
The two internal edges (with the labels e and f) have been transformed into a simple edge A_1 so far. In addition, all incident edges of the node x are now added to the new edge A_1, so that the edge is a hyper edge if there are external incident edges. However, the information (edge direction, edge label and equivalence class to the internal node) of the individual edges must not be lost.
The figure 10 illustrates how the edge directions for all external incident edges can be handled. Therefore incoming edges get the equivalence class 3 and outgoing edges get the equivalence class 4. *Zzgl. der bereits existierenden AK?*
Thus, the label and the equivalence class must be parameterized for each external incident edge. You must also make sure that the parameters can be assigned to the correct edge end of the hyper edge. Since the number of external incident edges is arbitrary, the number of parameters can be arbitrarily large.
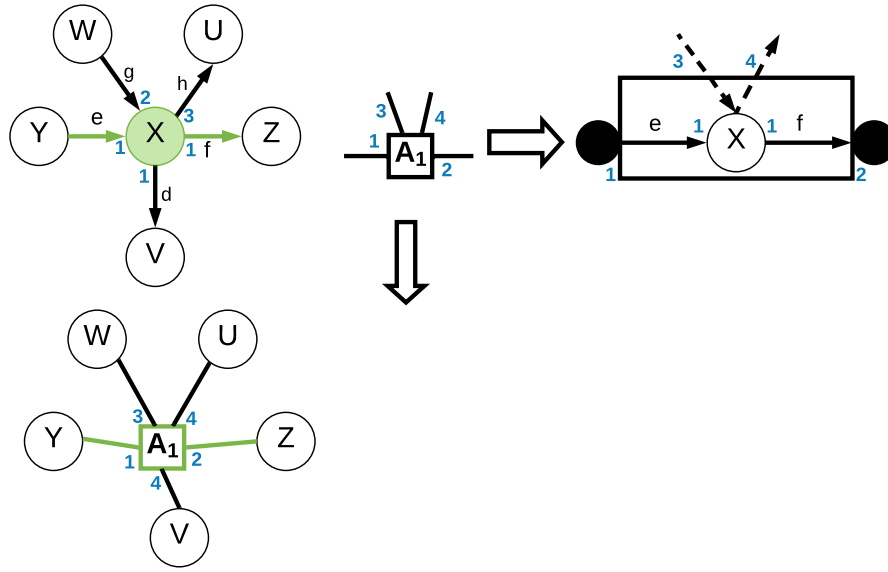In this case, the parameter must therefore be included:

Figure 10: Application of a replacement of a basic edge digram with incident edges to the internal node

- W: Label is g; equivalence class is 2

- U: Label is h; equivalence class is 3

- V: Label is d; equivalence class is 1

- Which end contained which labels

Case 2: External incident edges of the internal node become incident edges of an external node.

All external incident edges to X (see Fig. 11) become incident to the external nodes, which is the start node of the edge (in this case Y). This changes the number of incident edges to Y, where you have to be able to distinguish which of the edges were incident to X before. Therefore, the equivalence classes in Y must be changed so that this is possible.

Thus an existing node is changed and also the new nonterminal edge. Therefore, two persisting elements are changed by the replacement of the digram, whereby this type of digram turns the graph into a non context-free grammar. Furthermore it should be clarified how in node Y the label is not lost but the reference to the rule A_1, which is normally represented by the label of the element, is still present. Additionally, by changing external nodes, the subsequent compression is made more difficult, because parts of the occurrences of Y are modified and are therefore no longer synonymous with unmodified Y occurrences.
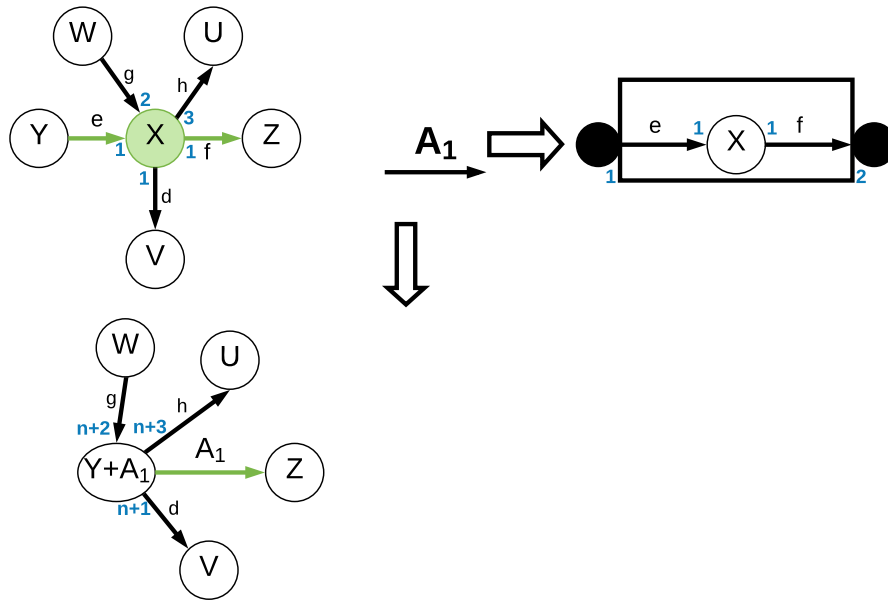
Figure 11: Application of a replacement of a basic edge digram with incident edges to the internal node. 'n' is the number of different equivalence classes before substitution

7 Edge association digram

This edge association digram (Fig. 12) combines two adjacent edges to one hyper edge Here only the two edges are internal elements of the rule. The equivalence classes at the middle external node must be the same. The replacement results in a hyper edge which in turn has equivalence classes of its own. The directions of the edges and the edge labels must be the same in all occurrences.

8 Other digram types

All other digram types from Maneth can be adopted analogously. First, you should definitely determine how incident edges can be treated to internal nodes (Section. 6), since this case also occurs in the other types.

8
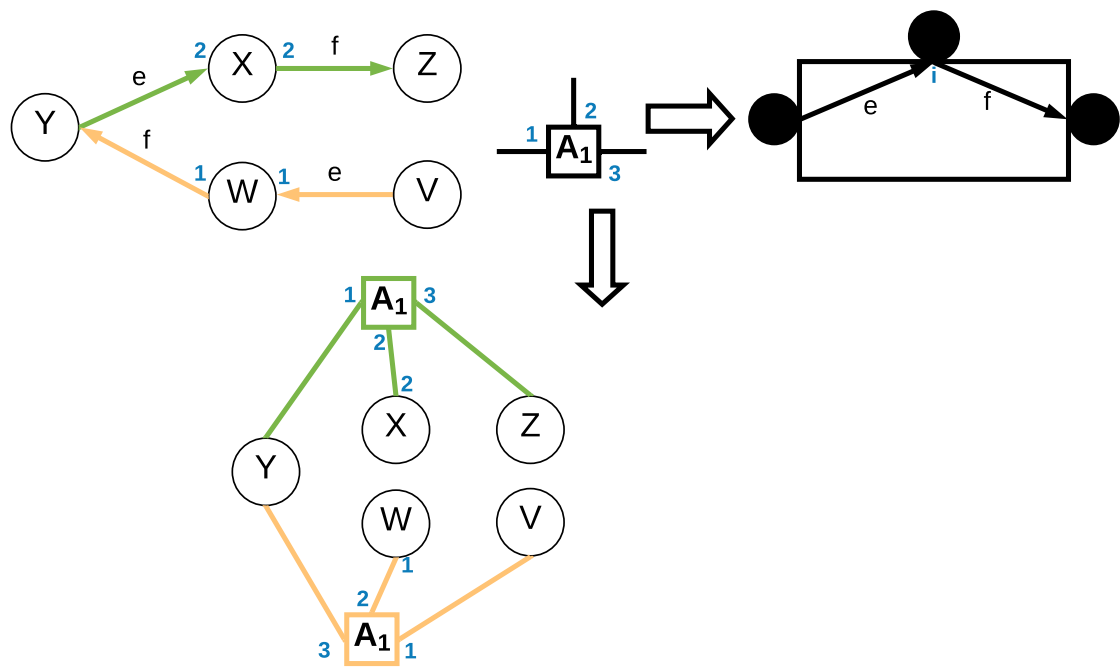
Figure 12: Application of a replacement of an edge association digram

All different types of digram types from Maneth

Matthias Dürksen

December 13, 2018

Purpose of the document

This document provides an overview of the different types of digrams described in the Maneth paper. The edges have been labelled in contrast to Maneth and equivalence classes for nodes have been added. The Digram types are therefore not necessarily directly applicable to the approach of our graph compression.

1 Maneth's digrams

Incident edges to internal nodes are also a problem here. This is solved by Maneth's approach by using a new digram rule where the incident edges are contained in the digram. So instead of simple edges hyper edges are created. This was illustrated in Figure 2.
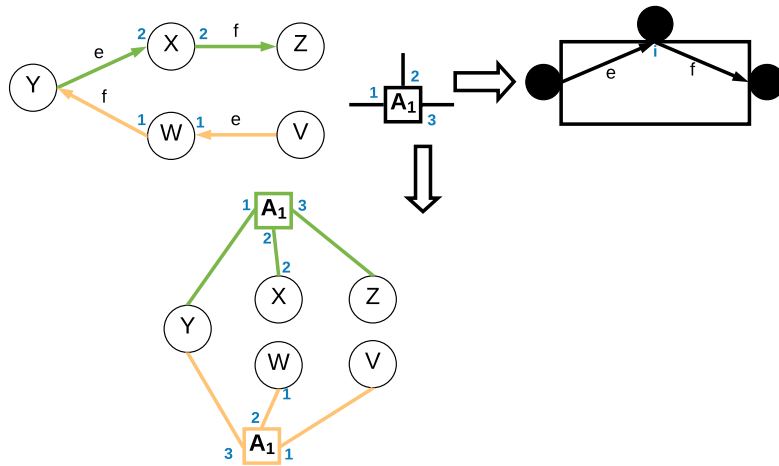
Figure 1: Application of a replacement of one digram type
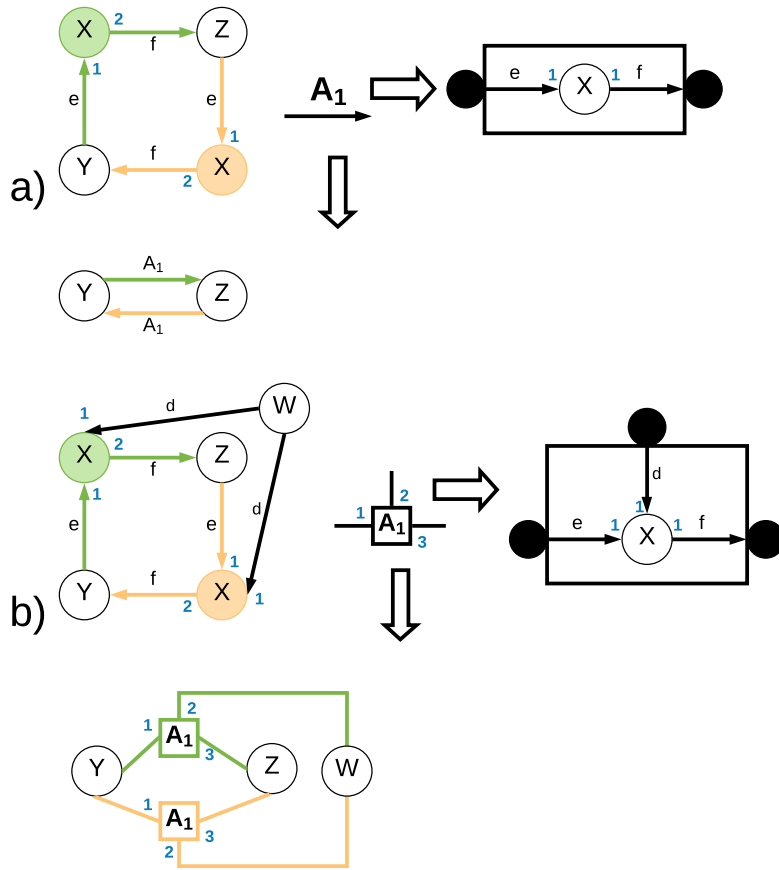
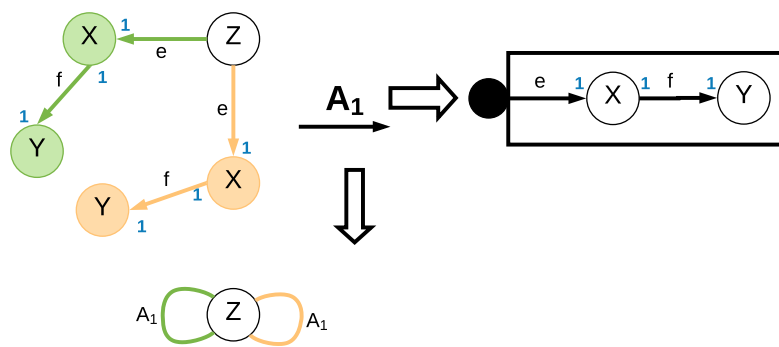Figure 2: Application of a replacement of one digram type

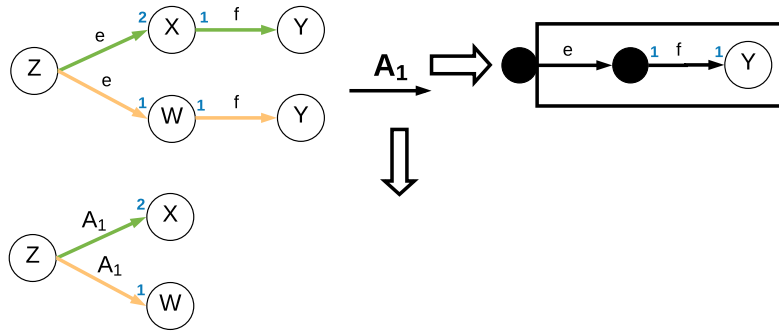Figure 3: Application of a replacement of one digram type

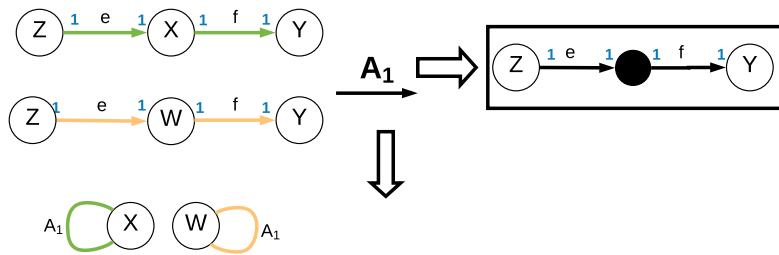Figure 4: Application of a replacement of one digram type

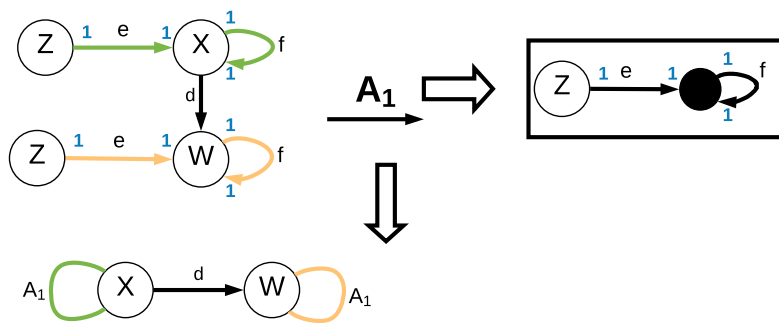Figure 5: Application of a replacement of one digram type

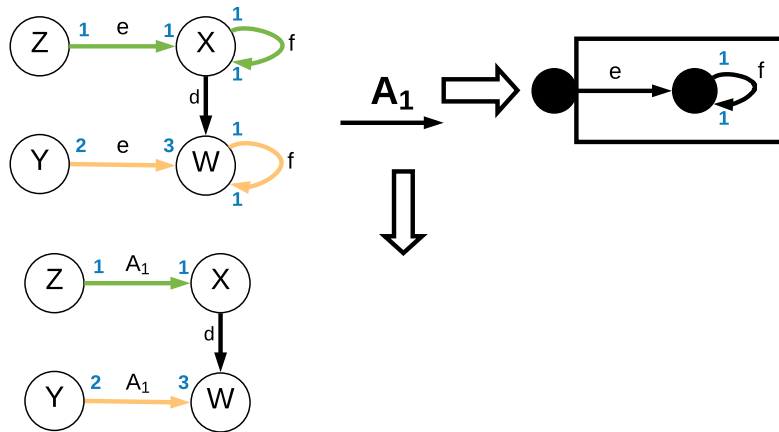Figure 6: Application of a replacement of one digram type
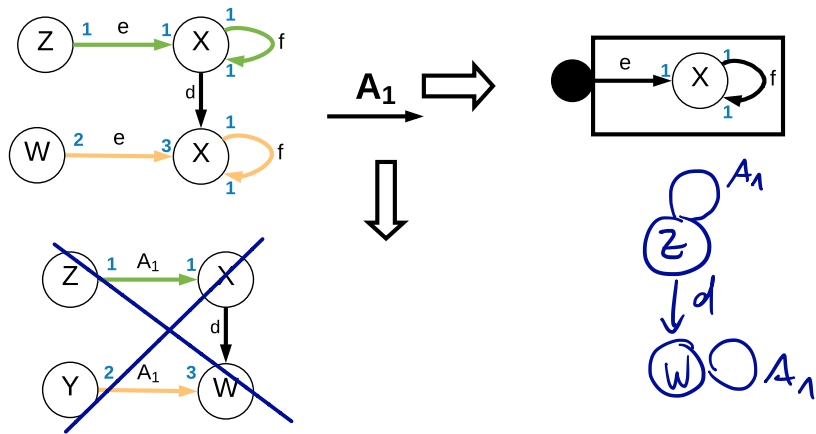
3

Figure 7: Application of a replacement of one digram type
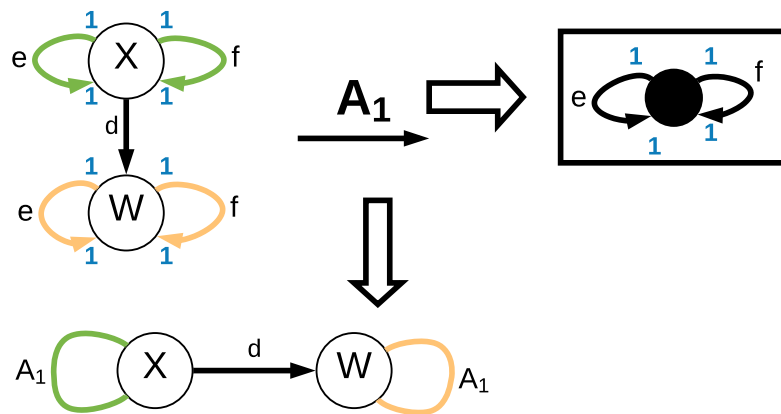

Figure 8: Application of a replacement of one digram type

Figure 9: Application of a replacement of one digram type

4