In the previous chapter, we describe flow patterns in spatio-temporal databases to capture the evolution of events in neighboring regions over time. While flow patterns can clearly capture the flow of events to some degree, they rely heavily on the assumption that these events will repeat themselves in exactly the same locations. However, in some applications, the absolute locations in which an event \( e \) has occurred are not important. Rather, it is the relative locations of events with respect to event \( e \) that are interesting. For example, an increase of rainfall at location \( x \) is followed by an increase of dengue fever cases in the Northeastern neighbourhood of \( x \).

In this chapter, we investigate an efficient method to discover this class of relative-location sensitive flow patterns. These generalized flow patterns aim to summarize the sequential relationships between events that are prevalent in sharing the same topological structures. We adopt the pattern growth approach and develop an algorithm called GenSTMiner to discover these patterns. In order to increase the efficiency of the mining process, we also present two optimization techniques. The first is the use of conditional pro-
jected databases to prune infeasible events and sequences, and the second is pseudo projection to reduce memory requirement.

This chapter is organized as follows. We will extend the notations in the previous chapter for generalized flow patterns. We will discuss the concept of generalized flow patterns and present the algorithm GenSTMiner. The performance study indicates that GenSTMiner is highly efficient and outperforms PrefixSpan.

**Notations and Terminologies**

We denote a location as \( l = (x, y) \), and a location-based event as \( e(x, y, t) \). Figure 9.1 shows an example of a spatio-temporal database which records the various locations where cyclones and storms occur over time. The space (shown in Figure 9.1(b)) is partitioned into 25 disjoint locations, and the time is divided into three disjoint time windows. Figure 9.1(a) shows the events \( \{a, b, c, d, etc\} \) that are observed at various locations over time.

Some sequences in Figure 9.1(a) that satisfy the flow pattern definition are as follows:

\[ <a (0, 0), f (0, 1)> \rightarrow d (1, 1) \]
\[ <a (1, 2), f (1, 3)> \rightarrow d (2, 3) \]
\[ <a (1, 1), f (1, 2)> \rightarrow d (2, 2) \]
\[ <a (0, 2), f (0, 3)> \rightarrow d (1, 3) \]

Each of these flow patterns occurs only once and will be discarded by most mining algorithms. However, a closer examination reveals that these patterns actually convey some interesting behavior of the cyclones, that is, “Event \( a \) in an area that has been hit by the storm always leads to event \( f \) in its Northern neighbors and event \( d \) in its Northeastern neighbors.” In other words, the absolute locations in which event \( a \) have occurred are not important. Rather, it is the relative locations of event \( d \) or \( f \) with respect to the event \( a \) that are interesting.

We observe that relative addresses play an important role in capturing the invariant topological relationships of a pattern. In order to incorporate the