Каждый контейнер держит внутри себя структуру данных. У очереди и стека это однонаправленный список, у множества – двоичное дерево.

Каждый из узлов внутренней структуры содержит в себе таблицу версий в виде ассоциативного массива. Ключом массива служит версия, значением – ссылка на следующий элемент. Узлы дерева имеют две таблицы, для левого и правого элемента соответственно.

При добавлении/удалении элемента в таблицу добавляется запись с номером текущей версии и новым значением.

Слияние двух контейнеров происходит следующим образом:

1. Формируется множество версий сливаемой ветки.
2. Формируется новая версия, которая будет создана в целевой (той, в которую сливают) ветке.
3. Далее происходит обход внутренней структуры данных. Для каждого текущего элемента находится версия с последним изменением (как максимальное значение из пересечения множества версий элемента с множеством версий сливаемой ветки). Ссылка из таблицы версий, соответствующая этой версии, сопоставляется новой версии целевой ветки.

Так как реализация производится на Python, то метод Close становится ненужным. Когда на сегмент не останется ссылок, сборщик мусора сам уберет сегмент. При этом значения, соответствующие его версии, уже недоступны – он либо уже был Collapsed, либо слит. Они просто остаются в таблицах версий элементов. Их можно убрать, но это потребует усложнения архитектуры: поддержания ссылок на измененные элементы.