Christian-Albrechts-Universität zu Kiel
Institut für Informatik

Priv.-Doz. Dr. Frank Huch, Marius Rasch

Institut
für Informatik

10. Übung zur Vorlesung „Concurrent and Distributed Programming“

Abgabe am Monday, 24. June 2019 - 18:00

Aufgabe 1 - Transactions in a Key-Value Store

1 Punkt

Implement a Key-Value-Store, which allows transactions. It should be possible to create this transactions by dynamic concatination of the following operations:

- `insert(Key,Value)`: Stores the given Key-Value-Pair. Overrides an existing entry with the same key.
- `lookup(Key)`: Lookup the value for the given key. Is not successful, if the key does not exist.
- `delete(Key)`: Deletes the entry specified by the given key. Is even successful, if the key does not exist.

Transfer the optimistic implementation approach used for `TVar`s in the lecture to a simple Key-Value-Store implementation.

All operations always have to relate to only one Key-Value-Store. To implement this you could use the Key-Value-Store as parameter for `atomically`.

Hint: Make sure, that your implementation of `delete` is correct. Keep in mind that is should be possible to add a value which was deleted earlier. Also consider that a `delete` can cause the need to stop the suspension of other transitions.

Aufgabe 2 - Retry without locking

1 Punkt

The STM implementation from the lecture locks all TVars in the read set before validating them, if a retry happens. It is possible to implement retry without locking the TVars. To do this we let the TVars check if the version changed. If the version is already a newer one, the TVar can wake the suspending process immediately. Otherwise the TVars add the process to the list of suspended processes.

Implement the version of retry explained above.