0.1 Checking safety property using CSP

The history can be captured as a trace of a CSP system. In addition to performing the function body, each function call sends a Call event before the function body and a Return event after the function body. Figure **??** is the definition of Call and Return channel in CSP. The definition of the channel usually needs to include all the identity of the calling process, the function called, and its parameter.

```
--identity of the calling process
--function called by the process
channel Call : TypeThreadID . TypeOps
--identity of the calling process
--function called by the process
--return value of the function call
channel Return: TypeThreadID . TypeOps . TypeThreadID
```

Figure 1: Definition of Call and Return channel

To check the safety property, we check that a testing system built from some processes using the concurrent datatype refines a specification process built from the object definition in the CSP trace model.

A generic and scalable system is used for the testing system to generate possible histories of processes using concrete datatype. Each process in the testing system can call any function from the concurrent object with any arguments allowed. Each process must be allowed to terminate. Otherwise, the testing system only models a system that runs forever, given that there is no deadlock. We shall see how this affects bug finding in a concurrent datatype in later objects.

The specification generates all valid histories. The process uses the same number of linearizer processes synchronizing on events from the Sync channel. Event from the Sync channel should include information from all participating processes, and Figure **??** is the Sync channel definition for ... Each linearizer process repeatedly calls a function, synchronize with zero or some processes, and returns according to the calling argument and extra information from the synchronization point. To match the definition of the generic and scalable testing system, each linearizer process can also choose to terminate.

```
--Identity of thread calling ManSync
--Return of ManSync
--Identity of thread calling WomanSync
--Return of WomanSync
channel Sync: TypeThreadID . TypeThreadID . TypeThreadID . TypeThreadID
```

Figure 2: Definition of Sync channel

We shall see a concrete testing system and specification process in the Men-Women section.

0.2 Checking liveness property using CSP

For liveness property, we check the same generic and scalable testing system refines the same specification process, but in the failure model. Suppose all process calls manSync. Since a linearizer process calling manSync sends Return event only after synchronizing Sync event with another linearizer process calling womanSync, the linearizer will refuse to return any function call, which is a expected behavior. One can use a datatype-specific specification process that does not explicitly use any synchronization points. However, reusing the linearizer process is easier.