Operations for arrays in Cogent

December 3, 2020

Currently, only one hole (= taken index) is allowed in arrays, rather than multiple ones. This avoids the need for checking that two general symbolic indices differ (to prevent taking twice the same index). Note that checking equality of symbolic indices is still required for Put, but the trivial syntactic equality test is, I guess, already strong enough for common usage.

We say that that an index is **present** if the hole is (provably) not at its position.

Remark 1. *The led driver explicitely involves writable arrays of non-linear (discardable and sharable) elements. Through the bang operation, it also involves read-only arrays of non-linear elements.*

Remark 2. *Constant sized arrays can be (non-efficiently) copmiled to records. Accessing some symbolic index would be translated by explicitely pattern-matching on the value of the index.*

Here are the available operations for arrays:

index (similar to Member) for discardable arrays: return the n^{th} element of an array, where n is a present symbolic index which is provably smaller than the array size (this condition is to be implemented later with refinement types).

take (similar to Take for records, not used in the led driver), executes a continuation provided with

- the n^{th} element of an array (if the array has no holes[1]), for n a symbolic index provably smaller than the array size,
- the array where the hole is set at n, unless the type is sharable, in which case the hole may be left unset.

put updates a non-readonly array anywhere, as long as the type is discardable.

put' (not used in the led driver) updates a non-readonly array at a taken symbolic index (equality between the taken index and the put index can be left syntactic, or refined later using refinement types), and returns the updated array

[1] Take could be allowed for arrays with holes if the type is sharable. In this case, it could be enforced (or not?) that the hole remains unchanged in the continuation.

fold (not used in the led driver) for discardable arrays. There could also be a version returning the array, or executing a continuation, for non-discardable sharable arrays[2].

map2 takes two non-readonly arrays without holes, a function taking two arguments and returning their updated values, and returns the pair of updated arrays.

[2] For non-discardable arrays, sharability is required, otherwise all the indices should be marked as taken, but we only allow arrays with one hole.