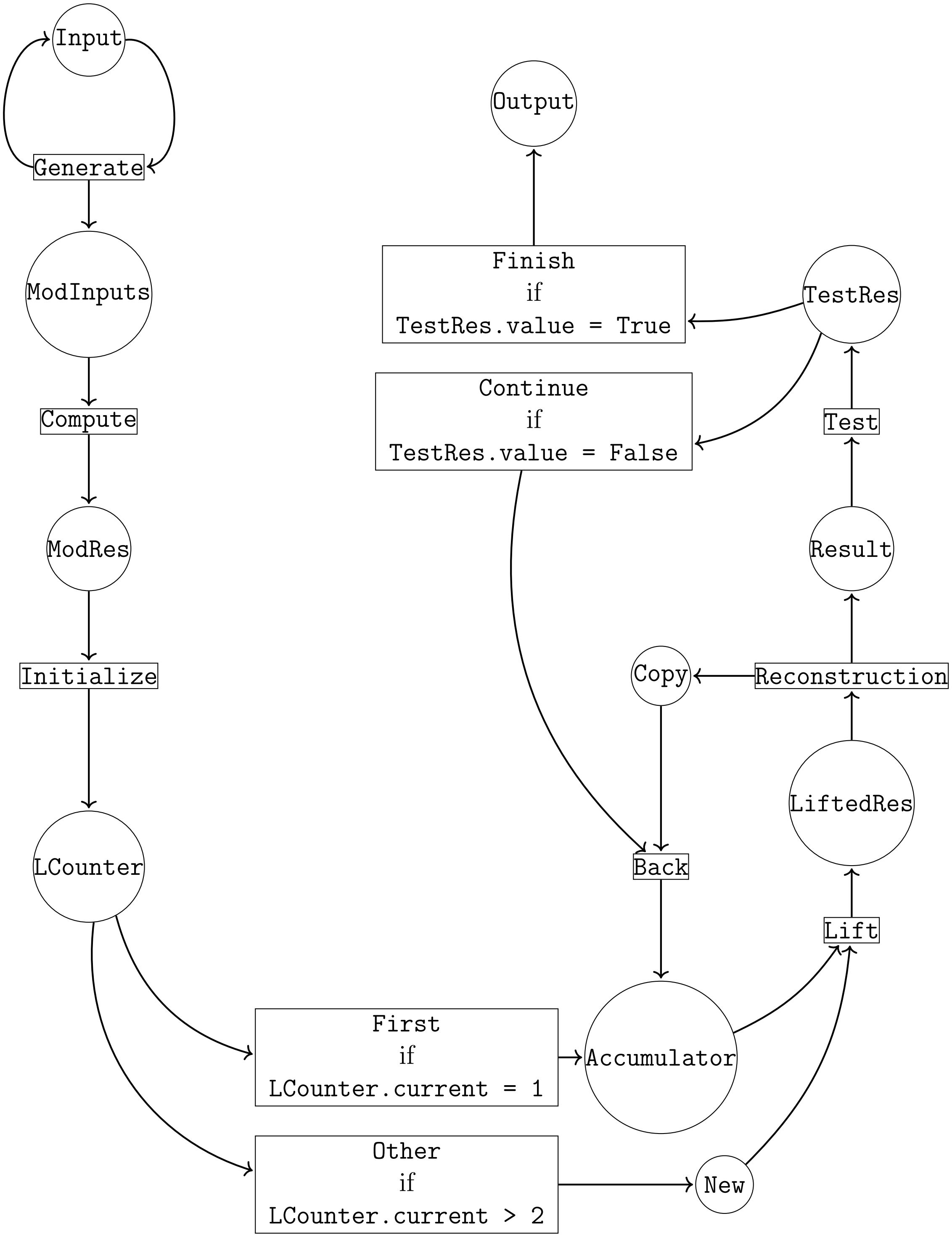
Petri net for modular algorithms in commutative algebra


Input
Output
Generate
ModInputs
Finish
if
TestRes.value = True
TestRes
Compute
Continue
if
TestRes.value = False
Test
ModRes
Result
Initialize
Copy
Reconstruction
LiftedRes
LCounter
Back
Lift
First
if
LCounter.current = 1
Accumulator
Other
if
LCounter.current > 2
New

SINGULAR configures token to initialize the place Input. This token contains information about the primes used so far. The transition Generate generates on the place ModInputs prime suitable with the input and updates information about the prime on Input. The transition Compute performs the computations related to each token on ModInput and the results are placed on ModRes. The transition Initialize increments the value of the token on LCounter by one. The first modular result is transferred to the place Accumulator while the others on New. Once these places contain token, the transition Lift combines the results in one on LiftedRes. Then, the transition Reconstruction saves a copy of the data consumed by it on Copy. The reconstruction over the rational will be on the place Result. The transition Test now verifies the correctness of the result according to the specific test implemented. A boolean value about the test is on the place TestRes. If the test fails, the transition Continue will fire and put back the lifting result on Accumulator via the transition Back. If the test succeeds, the transition Finish will fire and place a boolean token to signal the end of the computations. The final result can be extracted from the place Output.