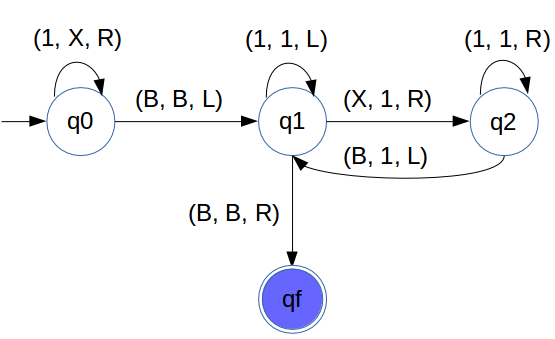
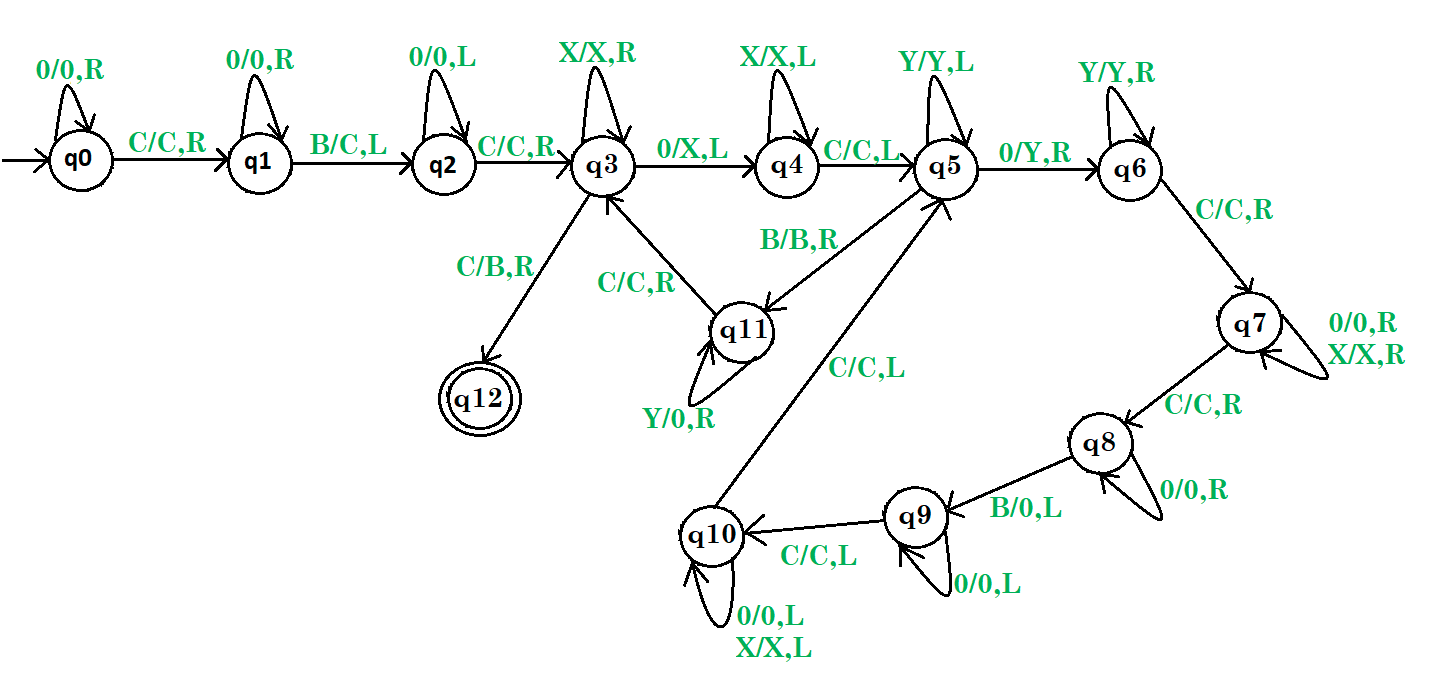
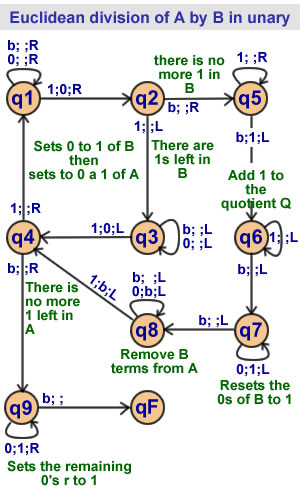
Turing machine as COPIER



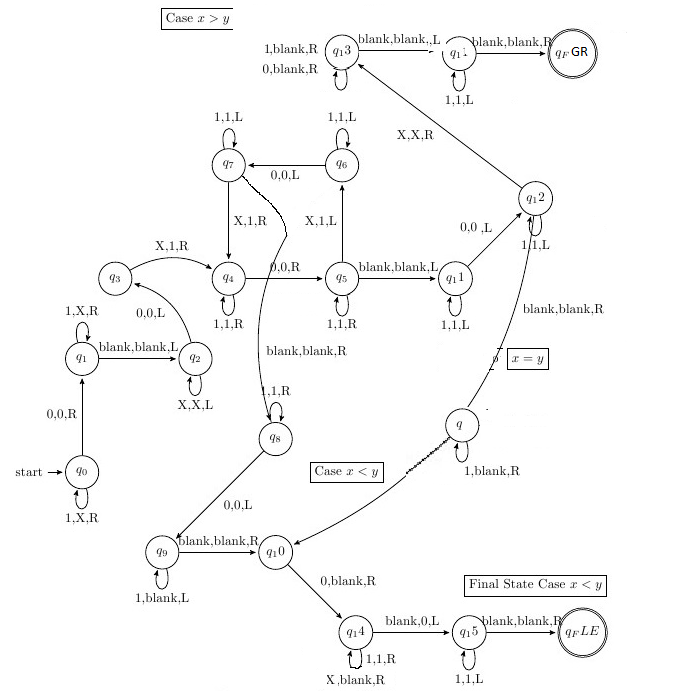
TURING MACHINE FOR MULTIPLICATION OF TWO NUMBERS



TURING MACHINE FOR DIVISION



TURING Machine for F(x,y) = |x – y|



* **Definition:** Let M = (Q, Σ, Г, δ, q0, B, F) be a TM, and let w be a string in Σ\*. Then w is *accepted* by M iff

q0w |—\* α1pα2

where p is in F and α1 and α2 are in Г\*

* **Definition:** Let M = (Q, Σ, Г, δ, q0, B, F) be a TM. The *language accepted by M*, denoted L(M), is the set

{w | w is in Σ\* and w is accepted by M}

* **Notes:**
  + In contrast to FA and PDAs, if a TM simply *passes through* a final state then the string is accepted.
  + Given the above definition, no final state of a TM need to have any transitions. *Henceforth, this is our assumption*.
  + **If x is NOT in L(M) then M may enter an infinite loop, or halt in a non-final state**.
  + Some TMs halt on ALL inputs, while others may not. In either case the language defined by TM is still well defined.
* **Definition:** Let *L* be a language. Then *L* is *recursively enumerable* if there exists a TM *M* such that L = L(M).
  + If *L* is r.e. then L = L(M) for some TM *M*, and
    - If *x* is in *L* then *M* halts in a final (accepting) state.
    - If *x* is not in *L* then *M* may halt in a non-final (non-accepting) state or no transition is available, or loop forever.

In this case, we say M accepts L.

* **Definition:** Let *L* be a language. Then *L* is *recursive* if there exists a TM *M* such that L = L(M) and M halts on all inputs.
  + If *L* is recursive then L = L(M) for some TM *M*, and
    - If *x* is in *L* then *M* halts in a final (accepting) state.
    - If *x* is not in *L* then *M* halts in a non-final (non-accepting) state or no transition is available (does not go to infinite loop).

In this case, we say M decides L.

The set of all recursive languages is a subset of the set of all recursively enumerable languages.

L is Recursively enumerable:

*TM exist: M0, M1, …*

*They accept string in L, and do not accept any string outside L*

L is Recursive:

*at least one TM halts on L and on ∑\*-L, others may or may not*

L is Recursively enumerable but not Recursive:

*TM exist: M0, M1, …*

*but none halts on all x in ∑\*-L*

*M0 goes on infinite loop on a string p in ∑\*-L,*

*while M1 on q in ∑\*-L*

*However, each correct TM accepts each string in L, and none in ∑\*-L*

L is not R.E*:*

*no TM exists*