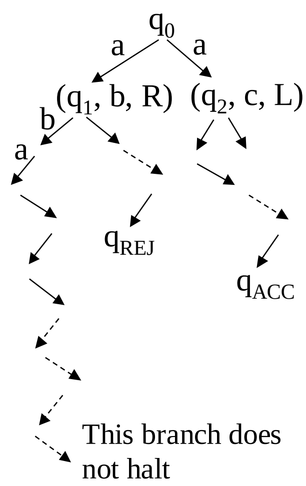
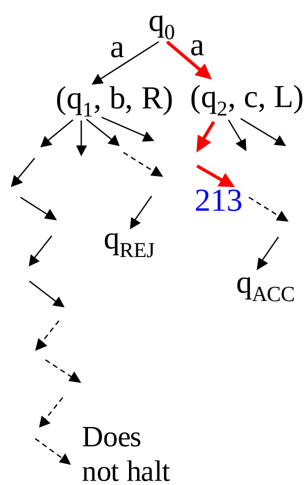
Simulating Nondeterministic TMs

- Nondeterministic TMs (NTMs)
  - $\delta: Q \times \Gamma \rightarrow \mathrm{Pow}(Q \times \Gamma \times \{L,R\})$
  - No ε transitions but can simulate them by reading and writing same symbol and moving head back to same position
- Any nondeterministic TM N can be simulated by a deterministic TM M
- N accepts w iff there is at least 1 path in N's tree for w ending in q_{ACC}
- Proof idea: Use breadth first search to simulate each branch
  - Explore all branches at depth *n* before *n+1*



Simulating Nondeterminism: Details, Details

- Use a 3-tape DTM M for breadth-first traversal of N's tree on w:
  - Tape 1 keeps the input string w
  - Tape 2 stores N's tape during simulation along 1 path (given by tape 3) up to a particular depth, starting with w
  - Tape 3 stores current path number
    E.g. ε = root node q_0
    213 = path made up of 3^{rd} child of 1^{st} child of 2^{nd} child of root
- See text for more details



Closure Properties of Decidable Languages

- Decidable languages are closed under ∪, °, *, ∩, and complement
- Example: Closure under ∪
- Need to show that union of 2 decidable L's is also decidable

  Let M1 be a decider for L1 and M2 a decider for L2

  A decider M for L1 ∪ L2:

  On input w:

  1. Simulate M1 on w. If M1 accepts, then ACCEPT w. Otherwise, go to step 2 (because M1 has halted and rejected w)
  2. Simulate M2 on w. If M2 accepts, ACCEPT w else REJECT w.

  M accepts w iff M1 accepts w OR M2 accepts w

  i.e. L(M) = L1 ∪ L2

Closure Properties

- Consider the proof for closure under ∪

  A decider M for L1 ∪ L2:

  On input w:

  1. Simulate M1 on w. If M1 accepts, then ACCEPT w. Otherwise, go to step 2 (because M1 has halted and rejected w)
  2. Simulate M2 on w. If M2 accepts, ACCEPT w else REJECT w.

  M accepts w iff M1 accepts w OR M2 accepts w

  i.e. L(M) = L1 ∪ L2

Will this proof work for showing <u>Turing-recognizable languages</u> are closed under ∪? Why/Why not?

Closure for Recognizable Languages

- Turing-Recognizable languages are closed under ∪, °, *, and ∩ (but not complement! We will see this later in Chapter 4)
- Example: Closure under ∩

  Let M1 be a TM for L1 and M2 a TM for L2 (both may loop)

  A TM M for L1 ∩ L2:

  On input w:

  1. Simulate M1 on w. If M1 halts and accepts w, go to step 2. If M1 halts and rejects w, then REJECT w. (If M1 loops, then M will also loop and thus reject w)
  2. Simulate M2 on w. If M2 halts and accepts, ACCEPT w. If M2 halts and rejects, then REJECT w. (If M2 loops, then M will also loop and thus reject w)

  M accepts w iff M1 accepts w AND M2 accepts w i.e. L(M) = L1∩L2