Top Hat #3198

Note Every Turing machine M (in "Turing Kit form") can be converted into the text's form by adding states $\{q_{acc}, q_{rej}\}$ and for all $q \in Q$, $c \in \Gamma$ s.t. no instruction begins $(q, c/\text{—})$: if $q \in F$ add $(q, c/c, S, \underline{q_{acc}})$
if $q \notin F$ add $(q, c/c, S, \underline{q_{rej}})$.
We may suppose this has been done (without loss of generality).

Defn: $M(x)$ halts (sometimes written $M(x)\downarrow$) if the computation of $M(x)$ eventually enters either q_{acc} (it accepts) or q_{rej} (it rejects).
A (deterministic) TM M is total if $(\forall x \in \Sigma^*) M(x)\downarrow$. ("M halts for all inputs").

Defn: A language $L \subseteq \Sigma^*$ is computably enumerable (c.e.) if there is a (det.) TM M s.t. $L(M) = L$. If also M is total, then L is decidable (synonym: recursive). [Synonyms for c.e.: r.e. for recursively enumerable, and Turing acceptable]
Historical terms: A. Church lambda calculus, General grammars

Individual Cases of Languages L: $L_{3n+1} = \{x \in \{0,1\}^* :$ The "3n+1" TM M_3 (demoed last Tuesday) accepts x (as a binary number$)\}$.

- L_{3n+1} is c.e.: if M accepts x, we know x goes to 1.

We believe → • $L_{3n+1} = \{0,1\}^* \setminus 0^*$ which is regular and so decidable.
(default: $n = |x|$) because a DFA is a total TM — which halts within $n+1$ steps

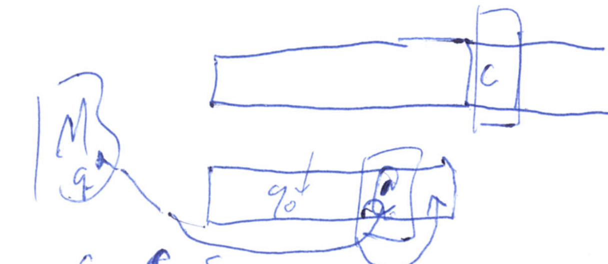
But • we don't know whether M_3 is total, indeed we don't even know whether the (Platonically real) L_{3n+1} is decidable!

The class of c.e. languages is denoted by $\underline{RE}$
The subclass of decidable languages in $\underline{REC}$ or $\underline{DEC}$.

Chomsky Hierarchy
FACT: $REG \subsetneq DCFL \subsetneq CFL \subsetneq REC \subsetneq RE$
MNT PAL CFL∩CFL to be proved

(2)

Fact 1: A DPDA M might not be total, but can always be[(*)] converted into a DPDA M' that is total and <u>equivalent</u>: $L(M') = L(M)$



$(q \overset{c}{e} / \overset{s}{e} \; \overset{s}{s} \; q)$ – instant loop

$(q \overset{c}{\sqcup} / \overset{c}{e} \; \overset{s}{R} \; q)$ – runaway stack

Fact: All such cases $(q, c_1 c_2)$ can be determined ahead of time and the code of M rewritten to M' [* not so easy].

so that all such (q, c) combos go straight to q_{rej}

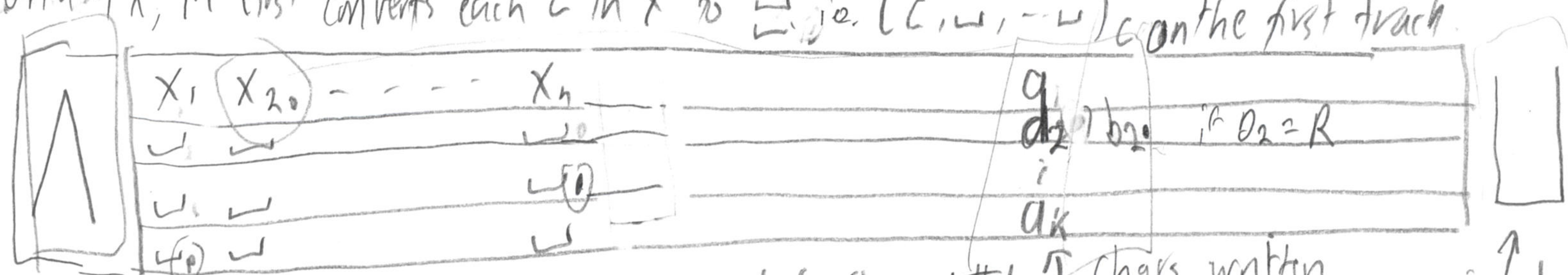
This fact proves DCFL ⊆ REC. [What about CFL]

(D) (works for NTM too)

Fact 2: For any $K > 1$ and K-tape TM M we can build a 1-tape (D) TM M_1 s.t. $L(M_1) = L(M)$ and M_1 can simulate any t steps of M in $O(t^2)$ steps of its own — so if M is total, then M_1 is total.

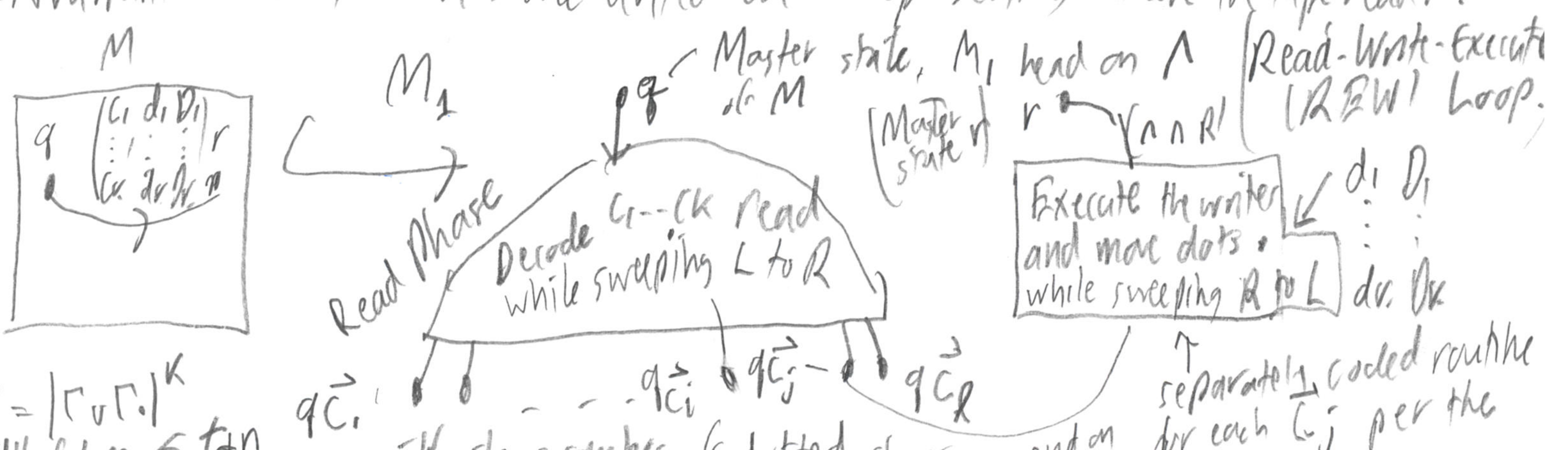
<u>Proof Idea:</u> M_1 uses alphabet $\Gamma_1 = \{\sqcup, \wedge\} \cup \Sigma \cup (\Gamma \cup \dot{\Gamma})^K$ where $\dot{\Gamma}$ is a dotted copy of M's alphabet Γ.

On any x, M first converts each c in x to $\overset{c}{\underset{\sqcup}{\sqcup}}$ i.e. $(c, \sqcup, \ldots \sqcup)$ on the first track.



Initially, after the conversion, x_1 and the $\sqcup$s below it are dotted.

INVariant: each track has one dotted char representing where the tape head of M is.



$\ell = |\Gamma \cup \dot{\Gamma}|^K$ all possible char combos of dotted chars.

width of tape $\leq t + n$

Hence REW time $\leq 2t$

Fact 3: For every "Random Access Machine" R (hence any high-level language program R") we can build a (3-tape) TM M_3 that simulates R, in particular $L(M_3) = L$

[Lecture demoed the "Universal RAM Simulator" handout]

Added:

This fact is the most solid (IMHO) plank in the Church-Turing Thesis.
Philosophical discussions of CTT range into the extent to which human beings will be replaceable by robots/algorithms, specifically for decision making.

The "Physics CTT" asserts: No theoretical model whose decidability criterion includes more languages than Turing Machines can decide will ever have a physical realization — not on Earth or anywhere in the cosmos.

That aliens' computers cannot surpass ours in theoretical makeup makes reasonable the plot device in the movie "Independence Day" where Jeff Goldblum uploads a virus to the spaceship's computer. I witnessed what is still considered the most substantial challenge to the Physics CTT at Oxford in 1985, when David Deutsch — then a fellow grad student — claimed quantum computers could do so. His argument was refuted (only) by a subtle point about calculus measures applied to infinite random sequences.

There is also a "Polynomial Time Church-Turing Thesis" which states that for time measures $T(n)$ on any physically realizable model — adding up total time or effort if the model does parallel processing — there is a k such that the same computation can be realized on a (multitape) Turing machine in time $t(n) = O(T(n))$.
Here Deutsch may get his revenge, because the quantum computing model can factor n-digit numbers in $O(n^{2+\epsilon})$ time, but no "classical" algorithm better than time $2^{O(n^{1/3 - o(1)})}$ is known.