Note: Every Turing machine $M$ (in Turing-UT form) can be converted into the text's form by adding states $q_{acc}, q_{ rej}$ and for all $q \in Q$, if $P$ st: no instruction begins $(q, c) \rightarrow (-):$ if $q \in E$ add $(q, c/c, S, q_{acc})$ if $q \notin E$ add $(q, c/c, S, q_{ rej})$.

We may suppose this has been done (without loss of generality).

Def: $M(x)$ 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 $(\forall x \in \Sigma^*) M(x) \downarrow$ (i.e., halts for all inputs).

Def: A language $L \subseteq \Sigma^*$ is \textit{computably enumerable} (CE) if there is a (deterministic) TM $M$ st: $L(M) = L$. If also $M$ is total, then $L$ is \textit{decidable} (synonym = recursive).

Historical terms: A. Church Lambda calculus, General grammar, Recursively enumerable, and algorithm

Individual Cases

Of Languages $L$:

$L_{\text{int}} = \{ x \in \Sigma^* \mid \text{the } \texttt{int} \text{ TM $M_3$ that we found last Tuesday accepts } x \text{ as a binary number} \}$.

$L_{\text{int}}$ is CE: if $M$ accepts $x$, we know $x$ goes to $1$.

We believe $L_{\text{int}} = \{0111^*0^* \mid \text{which is regular and so decidable.} \}$

But $L_{\text{int}}$ is not CE.

Because a DFA is a total TM - which halts within $n+1$ steps.

We don't know whether $M_3$ is total, indeed we don't even know whether the (platonically real) $L_{\text{int}}$ is decidable!

The class of CE languages is denoted by $\text{REC}$.

The subclass of decidable languages is $\text{REC} \subseteq \text{DEC}$. 

The class of CFL is denoted by $\text{CFL}$. 

The class of RE languages is denoted by $\text{RE} = \text{REJ} \cup \text{DFL} \cup \text{CFL} \cup \text{REC} \cup \text{R}$. 

The class of decidable languages is $\text{REC}$ or $\text{DEC}$. 

$\text{NTM} \uparrow \text{PML} \uparrow \text{PAL} \uparrow \text{CFL} \uparrow \text{RE}$ in preorder.
Fact 1: A DPDA $M$ might not be total, but can always be converted into a DPDA $M'$ that is total and equivalent: $L(M') = L(M)$.

Fact 2: For any $K > 1$ and $K$-tape TM $M$, we can build a 2-tape (0) TM $M'$ s.t. $L(M') = L(M)$ and $M'$ can simulate any $t$ steps of $M$ in $O(t^2)$ steps of its own — so if $M$ is total, then $M'$ is total.

Proof Idea: $M'$ uses alphabet $\Gamma' = \Sigma \cup \{ \cdot \}$, where $\cdot$ is a dotted copy of $M$'s alphabet $\Sigma$.

Initially, after the conversion, $x_1$ and the $\cdot$'s below it are dotted at a later stage.

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

Read-Write-Execute (RWE) Loop: Execute R/W and move dots while sweeping $R$ to $L$ vs. $L$ to $R$ separately. Coded routine for each $C_i$ per the instruction.
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 \( \text{L}(M_3) = \text{L}(R) \).

Lecture demo'd 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 "Physicist 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 substantive challenge to the Physicist CTT at Oxord 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 (multi-tape) Turing machine in time \( T(n) = O(n^k) \). Here Deutsch may get his revenge, because the quantum computing model can factor n-digit numbers in \( O(\sqrt{n} \log n) \) time, but no "classical" algorithm better than time \( 2^{O(n^{1/3} - \epsilon)} \) is known.