Tese de Church

Diante do fato demonstrado de que a máquina de Turing não tem o seu poder computacional ampliado através de qualquer alteração estrutural (vide resumo anterior). Isto é, tornar a fita duplamente infinita, aumentar o número de fitas, de cabeçotes, criar uma máquina não-determinística, nada aumenta o poder computacional da máquina de Turing conforme foi concebida.

Diante da existência de modelos computacionais distintos, como:

- Máquina de Turing
- Máquina de Post
- Funções recursivas
- Cálculo-λ

Diante da demonstração de que os modelos são equivalentes em seu poder computacional.

Enfim, todos estes fatos nos levam a considerar que todos os modelos de computação são equivalentes (não apenas os estudados até então) e enunciar a **Tese de Church**:
"*As máquinas de Turing são versões formais de algoritmos, e assim sendo, nenhum procedimento computacional será considerado um algoritmo se não puder ser representado como uma máquina de Turing.*"

Quando se utiliza a Tese de Church, e isso é usual em computação, pode-se demonstrar a equivalência entre um modelo computacional qualquer e a máquina de Turing apenas simulando no modelo a máquina de Turing.

Computação por Gramáticas

Pode-se utilizar gramáticas como dispositivos geradores de linguagens, mas também como dispositivos computacionais. Desta maneira, as gramáticas podem ser vistas como computadores de funções, desde que tais funções sejam de e para cadeias. Para isso, pode-se usar a Tese de Church.

Lema: Seja M=(K, Σ, δ, s) uma máquina de Turing qualquer. Então existe uma gramática G tal que para quaisquer configurações (q, u, a, v), (q', u', a', v') de M, $(q, u, a, v) \vdash_M^* (q', u', a', v') \Leftrightarrow [uqav] \Rightarrow_G^* [u'q'a'v']$. Onde, [e] são símbolos não presentes em K ou Σ, e assume-se que K e Σ são disjuntos.

Demonstra-se representando cada configuração como uma cadeia de símbolos, na qual o estado é um símbolo não-terminal de G, e separa o símbolo sob o cabeçote da cadeia à esquerda deste. Os símbolos [e] inseridos servem para indicar o final dos símbolos na fita, isto é, à direita de] só há símbolos "#". Produzindo uma representação em forma de regra de produção para cada movimento da máquina de Turing a ser simulada, obtém-se o efeito desejado. G=(K $\cup$ {[,],h,S}, Σ, P, S).

Funções Computáveis por Gramática

A partir do Lema anterior pode-se definir uma função gramaticamente computável.

Def.: Sejam Σ_0 e Σ_1 alfabetos que não contêm #, e seja f uma função de Σ_0^* para Σ_1^*. Então f será computável por gramática se e somente se houver uma gramática G=(V,T,P,S), onde Σ_0, $\Sigma_1 \subseteq$ T, e houver cadeias x, y, x', y' $\in$ (V $\cup$ T)*, tais que para qualquer u $\in \Sigma_0^*$, v $\in \Sigma_1^*$, f(u)=v $\Leftrightarrow$ xuy $\Rightarrow_G^*$ x'vy'. A definição para funções numéricas é semelhante à da Máquina de Turing. Observa-se que x, y, x', y' são marcadores associados à gramática G, assim a computação só terá efeito dentro dos marcadores.

Teorema: Toda função Turing-computável é gramaticalmente-computável.

Demonstra-se a partir de uma máquina de Turing qualquer M=(K, Σ, δ, s), para a qual f(u)=v $\Leftrightarrow$ $(s,\#u\#) \vdash_M^* (h,\#v\#)$. Aplica-se o Lema anterior para obter G, e faz-se x=x'="[#", y="s#]", y'="h#]".

Funções μ-Recursivas

Definem-se as funções recursivas primitivas e, a partir destas e de minimização ilimitada sobre funções regulares de Kleene, definem-se as funções μ-recursivas.

Faz-se um paralelo com cálculo-λ, fundamento das linguagens funcionais como Scheme.

Máquinas de Turing Universais

Pode-se definir uma codificação padrão para um alfabeto infinito contável e um conjunto de estados infinito contável em termos de cadeias de símbolos. Aceita-se, dessa forma, que qualquer máquina de Turing pode ter seus estados e símbolos codificados dentro do padrão. Assim tem-se os seguintes conjuntos:
$K_\infty = \{q_1, q_2, ...\}$ e $\Sigma_\infty = \{a_1, a_2, ...\}$,

De tal maneira que para toda máquina de Turing o conjunto de estados seja um subconjunto finito de K_∞, e o alfabeto da fita seja um subconjunto finito de Σ_∞. Adota-se a seguinte correspondência entre os símbolos componentes de uma máquina de Turing e cadeias sobre o alfabeto {I}.

σ	$\lambda(\sigma)$
q_i	I^{i+1}
h	I
L	I
R	II
a_i	I^{i+2}

Esta escolha elimina a possibilidade de haver dois elementos do mesmo conjunto representados da mesma forma.

Usa-se um outro símbolo ("c") para configurar a completamente a máquina, assim o alfabeto de representação será {c, I}. Portanto, cada símbolo e cada estado de uma máquina de Turing M=(K, Σ, δ, s) podem ser representados, e cada elemento da função de transição de uma máquina de Turing também pode, da seguinte maneira: seja δ(q, a)=(p, b), onde q e p $\in K_\infty \cup$ {h}, a $\in \Sigma_\infty$, e b $\in \Sigma_\infty \cup$ {L, R}, então cada elemento será representado por k.ℓ cadeias S_{pr}="$cw_1cw_2cw_3cw_4c$", ($1 \le p \le k$, $1 \le r \le \ell$) onde $w_1=\lambda(q)$, $w_2=\lambda(a)$, $w_3=\lambda(p)$ e $w_4=\lambda(b)$. Faça-se $S_0=\lambda(s)$ e defina-se a função de codificação $\rho(M)=cS_0cS_{11}S_{12} \ldots S_{k\ell}c$.

A máquina de Turing Universal opera sobre o alfabeto {c, I, #}, e recebe em sua fita a codificação ρ(M) seguida da codificação ρ(w). A operação da máquina de Turing Universal então é bastante simples, simula a execução de M a partir de s, acompanhando a função de transição e o símbolo encontrado na fita. Observe-se que mesmo que w contenha símbolos #, ρ(w) não possui #.

Assim define-se a máquina de Turing Universal U como:
$U=(K_U, \Sigma_U, \delta_U, s_U)$, tal que para toda máquina de Turing $M=(K, \Sigma, \delta, s)$ e para toda cadeia $w \in \Sigma^*$,

1. Se (h, u$\underline{a}$v) é uma configuração de parada de M tal que (s, #w$\underline{\#}$) $\vdash_M^*$ (h, u$\underline{a}$v), então (s_U, #ρ(M)ρ(w)$\underline{\#}$) $\vdash_U^*$ (h, #ρ(uav)$\underline{\#}$)

2. Se (s_U, #ρ(M)ρ(w)$\underline{\#}$) $\vdash_U^*$ (h, u'$\underline{a}$'v') é uma configuração de parada de U, então a'=#, v'=ε, u'=#ρ(uav), para algum u, a, v tais que (h, u$\underline{a}$v) seja uma configuração de parada de M, e que (s, #w$\underline{\#}$) $\vdash_M^*$ (h, u$\underline{a}$v).

Esta máquina define o padrão de computação usual, qualquer dispositivo computacional pode ser representado por uma máquina de Turing Universal, e cada máquina de Turing construída representa um algoritmo, um programa para a máquina Universal.