# Church-Turing Thesis

| | | |
|---|---|---|
| "algorithm" | = | "computation" |
| = "process" | | |
| $\parallel$ | | $\parallel$ |
| "software" | = | "hardware" |
| $\parallel$ | | $\parallel$ |
| $\lambda$-calculus | = | Turing-Machine |
| $\parallel$ | | $\parallel$ |
| Java | = | X86 |
| = C | | ARM = |
| = Racket | | MIPS = |

If $X = Y$
and $\forall x \in X.\ \neg P(x)$
then $\forall y \in Y.\ \neg P(y)$

"algorithm" = TM
TM cannot A
then "algorithms" cannot A

What we want from algorithms $\Longleftrightarrow$ Proofs about TMs
(closure operators)
(extensions)

"KNOW" $\Longrightarrow$ VERIFY

discoveries about algorithms $\Longleftarrow$ Conclusions about TMs (neat discoveries)

22-2/ Hilbert's 10th Problem

In 1900, David Hilbert

" A process involving finitely many steps exists to test whether a polynomial has an integral root."

Polynomil = ~~Variable~~ bunch of variables, you have an exponent and a coefficient, you add togeth

$\underbrace{6x^3yz^2}_{(3,1,2)=6} + \underbrace{3xy^2}_{(1,2,0)=3} \underbrace{-x^3}_{(3,0,0)=-1} \underbrace{-10}_{(0,0,0)=-10}$

variables = $x, y, z$

integral means "an integer"

$ax^2+bx+c$

$k=3$ $c_{max} = \max(a,b,c)$ $c_1 = a$

X-poly is a polynomial 1-variable

Matijasevic discovered that the root is bounded by

$$\pm k \cdot \frac{c_{max}}{c_1}$$

$k$ = # of $\neq 0$ coefficients

$c_{max}$ = largest coefficient

$c_1$ = coefficient of biggest exponent

$[-18, 18]$

No bound could exist for multivariable polynomials

we will show that some problems have no TM, therefore no algorith (via C-T-thesis)

---

$\Sigma_0$ : $\cup, \cap, \circ, *,$ $^c$

$\Sigma_1$ : $\cup, \cap, \circ, *$ (NOT under: $^c$)