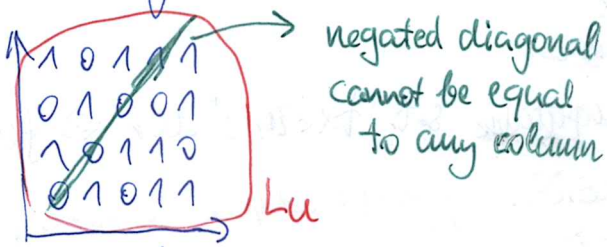
tape S: current state of M_x stored as $1^n 0^{|\gamma|}$

Init: Split $\langle \alpha, \beta \rangle$, copy α to tape K, encode β on tape T, initialize tape S ↳ & set tape M

Step: Read current symbol x from T, find entry for state s and symbol x on K, write new symbol & state, move head on T.

Lemma: $\overline{L_u} \notin RE$ ← We define $\overline{L} := \{0,1\}^* \setminus L$ for $L \subseteq \{0,1\}^*$

Proof: Use diagonalization



negated diagonal cannot be equal to any column

→ **diagonal language**

$L_d := \{\alpha \in \{0,1\}^* \mid \alpha \notin L_\alpha\}$

$L_d \notin RE$ ... assume $L_d \in RE$

Then $\exists \alpha: L_d = L_\alpha$

but: $\alpha \in L_d \Leftrightarrow \alpha \notin L_\alpha \Leftrightarrow \alpha \notin L_d$ ↯

If $\overline{L_u}$ were partially decidable, we could modify the machine accepting $\overline{L_u}$ to a machine accepting L_d ↯

Corollaries:

- $L_u \notin R$ (R is closed under complement, so $L_u \in R$ would imply $\overline{L_u} \in R \subseteq RE$)
- $R \subsetneq RE \subsetneq 2^{\{0,1\}^*}$
  - $R \subsetneq RE$: witnessed by L_u
  - $RE \subsetneq 2^{\{0,1\}^*}$: witnessed by $\overline{L_u}$
- RE is not closed under complement

Exercise: Are R and RE closed under ∩ or ∪? And under composition (assuming $f(\uparrow) = \uparrow$)?

Thm (Post's): $L \in R \Leftrightarrow L \in RE \;\&\; \overline{L} \in RE$. ← equivalently: $R = RE \cap \text{co-RE}$, where co-RE $= \{\overline{L} \mid L \in RE\}$

Pf: $\Rightarrow$ trivial, because $R \subseteq RE$ & R closed under complement.

$\Leftarrow$ "run machines accepting L and $\overline{L}$ in parallel" (one step of each at a time) One of them certainly stops.

Operations on machine codes

- swap q^+ with q^-: given M_x deciding ~~accepting~~ L, find M_β deciding $\overline{L}$
- compose two machines: find M_γ, which runs first M_x and then M_β on its output
- substitute M_x for an oracle in M_β

} all these are computable functions

More decision problems regarding machine codes

$L_{halt} := \{ \langle \alpha, \beta \rangle \mid M_\alpha \text{ halts on input } \beta \}$

$L_{empty} := \{ \alpha \mid L_\alpha = \emptyset \}$ $\quad L_{total} := \{ \alpha \mid L_\alpha = \{0,1\}^* \}$

$L_{eq} := \{ \langle \alpha, \beta \rangle \mid L_\alpha = L_\beta \}$ $\quad L_{fin} := \{ \alpha \mid L_\alpha \text{ is finite} \}$

Exercise: Which of these (& their complements) are in R and/or RE?

- Return to proof of $L_u \notin RE$ via $L_d \notin RE$: "if we can find a machine accepting L_u, we can use it to accept L_d ↯"
  ↳ let's generalize this.

Df: Many-to-one reduction between languages:

$K \leq_m L \equiv \exists f: \{0,1\}^* \to \{0,1\}^*$ computable s.t. $\forall \alpha \in \{0,1\}^* : \alpha \in K \Leftrightarrow f(\alpha) \in L$

↑ sometimes denoted $K \to L$

$\leq_m$ is a partial quasi-order on languages "by hardness"

Lemma: If $K \leq_m L$ and $L \in RE$, then $K \in RE$.
If $K \leq_m L$ and $L \in R$, then $K \in R$.

proof: compose machines for f and L

Corollary: If $K \leq_m L$ and $K \notin RE$, then $L \notin RE$. (Similarly $K \notin R \Rightarrow L \notin R$.)

- Our original proof used $L_d \leq_m \overline{L_u}$ & $L_d \notin RE$ to show $\overline{L_u} \notin RE$.

Exercise: Find reductions between L_u, L_{halt}, L_{empty}, L_{eq} & their complements.

Example: ③ $\overline{L_{halt}} \xrightarrow{\leq_m} L_{empty}$: given $\langle \alpha, \beta \rangle$, construct TM M_γ which ignores its input & runs M_α on input β
↳ $L_\gamma = \emptyset$ if $M_\alpha(\beta)$ diverges
$L_\gamma = \{0,1\}^*$ otherwise
$\Rightarrow \left(\gamma \in L_{empty} \Leftrightarrow \langle \alpha, \beta \rangle \in \overline{L_{halt}} \right)$

④ $L_{empty} \xrightarrow{\leq_m} \overline{L_{halt}}$: for given α, construct M_β which ignores its input, simulates M_α on all inputs in parallel & stops if $M_\alpha(\beta)$ stops on some β... $\langle \beta, \varepsilon \rangle \in \overline{L_{halt}}$
$\Leftrightarrow L_\alpha = \emptyset$.
$\Leftrightarrow \alpha \in L_{empty}$.

① $L \xrightarrow{\leq_m} M \Leftrightarrow \overline{L} \xrightarrow{\leq_m} \overline{M}$

② Also, $L_{halt} \leq_m L_u \leq_m L_{halt}$

- Semantic properties of machines

Df: Property of languages: $P \subseteq RE$ ... P is **non-trivial** $\equiv P \neq \emptyset \;\&\; P \neq RE$.
(semantic)

$L_P := \{ \alpha \in \{0,1\}^* \mid L_\alpha \in P \}$ ... all machines whose languages have the property P

Thm (Rice's): For every non-trivial property P, the language L_P is undecidable.

Proof idea: Show that $L_{halt} \to L_P$ for every non-trivial P.

Proof: Assume that $L_P \in R$ for some P.
WLOG $\emptyset \notin P$ ... otherwise use $\overline{P}$ ... $L_{\overline{P}} = \overline{L_P}$, so it's also in R.
Find $L_w \in P$ ... exists as P is non-trivial
Reduction: if we want to answer $\langle \alpha, \beta \rangle \in L_{halt}$, i.e. if $M_\alpha(\beta)$ halts

construct M_γ which does on input δ:
- run M_α on β (1)
- run M_w on δ (2)

this is computable

not exactly true: they can differ on invalid machine codes, but that's a decidable property

if $\langle \alpha, \beta \rangle \in L_{halt}$: (1) halts, (2) halts if $\delta \in M_w \Rightarrow L_\gamma = L_w \in P$
$\notin L_{halt}$: (1) diverges, (2) doesn't run $\Rightarrow L_\gamma = \emptyset \notin P$

So this shows $L_{halt} \leq_m L_P$ ... but $L_{halt} \notin R$, so $L_P \notin R$.

What is the "hardest" language in a class?
- Let $\mathcal{C}$ be a set of languages.
- L is $\mathcal{C}$-hard $\equiv \forall K \in \mathcal{C}: K \leq_m L$
- L is $\mathcal{C}$-complete $\equiv L$ is $\mathcal{C}$-hard & $L \in \mathcal{C}$

more precisely, it's $\mathcal{C}$-m-complete (complete wrt. $\leq_m$)

Thm: L_u is RE-complete.

Pf: (1) $L_u \in RE$
(2) for $K \in RE$, we find $\alpha: L_\alpha = K$
Then a reducing K to L_u is $\beta \mapsto \langle \alpha, \beta \rangle$

Also: If K is $\mathcal{C}$-complete and $K \leq_m L$ for $L \in \mathcal{C}$, then L is $\mathcal{C}$-complete, too.
Hence L_{halt} is also RE-complete.

"Natural" undecidable problems (not directly involving machines)
- given a set of axioms and a formula φ, is φ provable?
- given a system of multi-variate polynomial equations over $\mathbb{Z}$, does it have a solution in $\mathbb{R}$? → Matijasević theorem

both in $RE \setminus R$ (in suitable encoding)

& many more (e.g., plane tiling)

Relative computability

- given any language $A \subseteq \{0,1\}^*$, we can define TM with an oracle giving access to A (see section on TM extensions)
- we can define relative language classes $R[A]$ and $RE[A]$
- we also have $M_\alpha[A]$, $L_\alpha[A]$, $L_u[A]$

If $A \in R$, then $R[A] = R$ and $RE[A] = RE$ (in particular for $A = \emptyset$)

Previous arguments about plain TM can be trivially relativized, so in particular: $R[A] \subsetneq RE[A] \subsetneq 2^{\{0,1\}^*}$

$R[A] = RE[A] \cap co\text{-}RE[A]$ ← $co\text{-}T = \{L \mid \overline{L} \in T\}$

and also $L_u[A]$ is $RE[A]$-complete

Def: Arithmetical Hierarchy: classes $\Sigma_n, \Pi_n, \Delta_n$ for $n \in \mathbb{N}$

- $\Sigma_0 = \Pi_0 = \Delta_0 = R$
- $\Sigma_{n+1} = RE[\Sigma_n]$
- $\Pi_{n+1} = co\text{-}RE[\Pi_n]$
- $\Delta_{n+1} = R[\Sigma_n]$

this means: $RE[\mathcal{C}] = \bigcup_{L \in \mathcal{C}} RE[L]$

→ **We have:** $\Sigma_1 = RE$, $\Pi_1 = co\text{-}RE$, $\Delta_1 = R[R] = R$ } oracles from R do not add power to TM

$\Sigma_n \subseteq \Sigma_{n+1}$

this inclusion is strict ✱

~~does not hold if Σ_n is RE~~

$RE[L_u^n] \notin \Sigma_n$, ~~otherwise we would have~~ $\Sigma_n \subseteq R[\Sigma_n] = R[L_u^n] \subsetneq RE[L_u^n] = \Sigma_{n+1}$

- $L_u^1 = L_u$, $L_u^{n+1} = L_u[L_u^n]$ → L_u^n is Σ_n-complete (by induction: $RE[\Sigma_n] = RE[L_u^n]$, so $L_u[L_u^n]$ is Σ_{n+1}-complete)
- $\Pi_{n+1} = co\text{-}RE[\Sigma_n]$
- $\Pi_{n+1} = co\text{-}\Sigma_{n+1}$ → $\overline{L_u^n}$ is Π_n-complete

Also: $\Sigma_n \subseteq \Delta_{n+1}$ ... and this is strict as Σ_n is not closed under complement, while Δ_{n+1} is

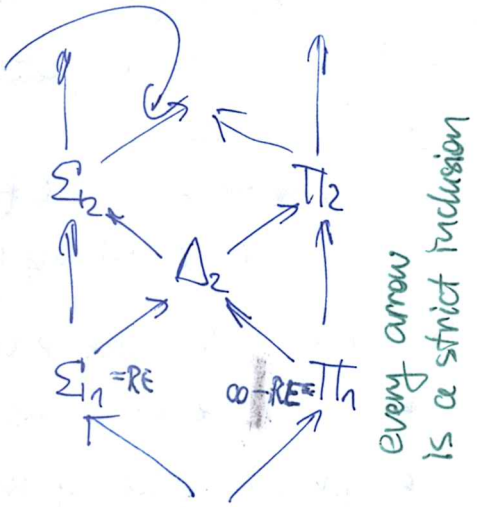
$\Pi_n \subseteq \Delta_{n+1}$ ... we can negate oracle's answer

$\Delta_{n+1} = \Sigma_{n+1} \cap \Pi_{n+1}$ ... relative Post's thm.

$\Delta_{n+1} \subseteq \Sigma_{n+1}$ ... this is $R[L_u^n] \subsetneq RE[L_u^n]$

$\Delta_{n+1} \subseteq \Pi_{n+1}$ ... analogous for co-RE

AH: $\bigcup_i \Sigma_i$ is still countable ⇒ "most" languages are outside AH



Quantified formulas

- every language in R can be interpreted as a predicate $\varphi(\alpha)$ with string parameter φ – decidable predicates
- $\psi(\beta) \equiv \exists \alpha\, \varphi(\alpha, \beta)$ lies in RE ... and every $L \in RE$ can be written in this way [α = # steps after which a machine stops]
- $\psi(\beta) \equiv \forall \alpha\, \varphi(\alpha, \beta)$ ... this is co-RE ($\neg \forall \alpha\, \varphi(\alpha,\beta) \Leftrightarrow \exists \alpha\, \underbrace{\neg \varphi(\alpha,\beta)}_{\text{decidable}}$)
- ... $\exists \alpha_1 \exists \alpha_2\, \varphi(\alpha_1, \alpha_2, \beta)$ is again RE ... we can say $\exists \alpha$ s.t. $\alpha = \langle \alpha_1, \alpha_2 \rangle$ & decode α inside φ
- $\exists \alpha_1 \forall \alpha_2\, \varphi(\alpha_1, \alpha_2, \beta)$ is Σ_2
- in general: $\exists \alpha_1 \forall \alpha_2 \exists \alpha_3 \ldots Q_n \alpha_n\, \varphi(\alpha_1 - \alpha_n, \beta)$ is Σ_n
  $\forall \alpha_1 \exists \alpha_2 \forall \alpha_3 \ldots Q_n \alpha_n\, \varphi(\alpha_1 - \alpha_n, \beta)$ is Π_n

$\subseteq \Sigma_2$: $\exists \alpha_1 \forall \alpha_2\, \varphi(\ldots) \Leftrightarrow \exists \alpha_1 \neg(\exists \alpha_2\, \varphi(\ldots))$ ← so this is in $RE[\Sigma_1] = \Sigma_2$

can be answered by oracle $L_u \in \Sigma_1$

$\supseteq \Sigma_2$ consider $L \in \Sigma_2 = RE[\Sigma_1]$:

$\exists y$ computation of TM $\in \Sigma_1[L_u]$ $\exists \delta$ queries for L_u (y, δ consistent & answers for δ true)

see next page

(continued) We want to show that $\forall L \in \Sigma_2$ there is an equivalent formula $\exists \forall \ldots$

- let M be a TM $[\Sigma_1]$ ... that is TM $[L_u]$ accepting L
- formula: $\exists \delta$ (check that δ is consistent with input α, with oracle L_u, and with itself)

$\exists \delta$: computation of M including all oracle queries & answers

consistent with input α: decidable

with oracle L_u: for positive answers: $\exists \gamma_1 - \gamma_k$ $\psi(\gamma_1, \text{question } 1)$ ⋮ $\psi(\gamma_k, \text{question } k)$ — Σ_1-formula for L_u

itself: decidable

together: $\exists \delta \left(\exists \gamma \; \forall \varepsilon \; \varphi(\alpha, \delta, \gamma, \varepsilon) \right)$

- γ: code of $\gamma_1 - \gamma_k$
- ε: code of $\varepsilon_1 - \varepsilon_\ell$

for negative answers: $\forall \varepsilon_1 - \varepsilon_\ell \; \neg\psi(\varepsilon_1, \ldots) \; \& \; \ldots$

... but this is $\exists \langle \delta, \gamma \rangle \forall \varepsilon \; \varphi(\ldots)$ as we need.

COMPLEXITY

Df:

- For a multi-tape machine M, define run time $t_M(\alpha)$ for input α as # steps before computation $M(\alpha)$ halts.
  - $t_M : \{0,1\}^* \to \mathbb{N}^*$ (generally Σ^*), $\mathbb{N}^* = \mathbb{N} \cup \{\infty\}$ (sorry for confusing notation: this $*$ is not iteration, but: for divergent computations)

- Time complexity of machine M: $T_M : \mathbb{N} \to \mathbb{N}^*$ s.t. $T_M(n) = \max_{\alpha \in \Sigma^n} t_M(\alpha)$.
  - M always halts $\Leftrightarrow$ $T_M(n)$ finite for all n.

Let's use $\alpha \in \Sigma^*, |\alpha| \le n$ instead: this is slightly non-standard, but much more convenient. In particular, $T_M(n)$ will be non-decreasing.

Df: Asymptotic notation: for functions $f, g : \mathbb{N} \to \mathbb{R}$ define:

1. $f \in O(g) \equiv \exists c > 0 \; \forall^* n \; f(n) \le c \cdot g(n)$ ← asymptotic upper bound
   - $\forall^* n$: for all but finitely many exceptions
2. $f \in \Omega(g) \equiv \exists c > 0 \; \forall^* n \; f(n) \ge c \cdot g(n)$ ← asymp. lower bound
3. $f \in \Theta(g) \equiv f \in O(g) \; \& \; f \in \Omega(g)$ ← both at once
   ... that is, $\Theta(g) = O(g) \cap \Omega(g)$
4. $f \in o(g) \equiv \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$ ← strict upper bound
5. $f \in \omega(g) \equiv g \in o(f)$ ← strict lower bound

Examples: $f : n \mapsto 5n^3 - 7n^2 + 18$

$\in O(n^3), O(n^4), O(2^n)$ $\quad \in o(n^4)$

$\in \Omega(n^3), \Omega(n^2), \Omega(1)$ $\quad \in \omega(n^2)$

$\in \Theta(n^3)$ "drop lower-order terms & multiplicative constant"

$O(1)$ = "bounded by constant"