Ch. 5 Computation

1 Memory

1.1 Definition

Define Memory; a set of elements

$$\mathcal{M} := \{b_1, b_2, ..., b_M\}$$

1.2 Silicon Computation

In silicon based computation memory is represented with bits 0/1

$$\mathcal{M} := \{b_1, b_2, ..., b_m\}$$
$$b_i \in \{0, 1\} \;\; \forall i$$

2 Logical Operations

Working list; possibly any set operation/function

+
-

/
exp
$\circlearrowright$
$\leftarrow$
delete
remove
insert
append
if
==
!
$f[X_i] \rightarrow Y_i$

3 Program

3.1 Logical Instructions

Define $\mathcal{L}$; an ordered set of logical operations s_i

$$\mathcal{L} := \{s_1, s_2, ..., s_N\}$$

3.2 State |

Define state; the memory required to perform program P

$$P := \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M\} =$$
$$\{s_1, s_2, ..., s_N, b_1, b_2, ..., b_M\}$$

3.3 Boolean Programs

Define a boolean program; boolean programs can represent functions with inputs x_i and boolean output y_o

$$X = \{x_1, ..., x_n\}$$
$$P = P[X] := \{s_1, s_2, ..., s_N \mid b_1, b_2, ..., b_M, X_i, y_o\} =$$
$$P[X] \to y_o \in \{\mathbb{T}, \mathbb{F}\}$$

3.4 Void Programs

Define a void program; a program with inputs x_i and no output

$$X = \{x_1, ..., x_n\}$$
$$P = P[X] := \{s_1, s_2, ..., s_N \mid b_1, b_2, ..., b_M, X_i\}$$

3.5 Numerical Programs

Define a numerical program; a program with inputs x_i and real, rational output y_o

$$X = \{x_1, ..., x_n\}$$
$$P = P[X] := \{s_1, s_2, ..., s_N \mid b_1, b_2, ..., b_M, X_i, y_o\} =$$
$$P[X] \to y_o \in \mathbb{Q} \;\; y_o \geqslant 0$$

3.6 System Programs

Naming convention to be formalized; a program that outputs one or more elements

$$X = \{x_1, ..., x_n\}$$
$$P = P[X] := \{s_1, s_2, ..., s_N \mid b_1, b_2, ..., b_M, X_i, Y_o\} =$$
$$P[X] \rightarrow Y_o = \{y_1, y_2, ..., y_K\}$$

3.7 Mathematical Programs

Define a mathematical program; a program with inputs x_i and numerical output y_o

$$X = \{x_1, ..., x_n\}$$
$$P = P[X] := \{s_1, s_2, ..., s_N \mid b_1, b_2, ..., b_M, X_i, y_o\} =$$
$$P[X] \rightarrow y_o \in \mathbb{Q}$$

4 No-op ;

4.1 Definition

Define ";" , the no-op (The C representation of a null statement or no-op)
$y_o = y_{o2}$ $\forall$; insertions

4.2 Properties of No-op

1. Can be added to any solution S_i and remain a solution for all i anywhere in the order for all j

5 For Loop $\circlearrowright$

5.1 Definition

6 Nested For Loop $\circlearrowright^n$

6.1 Definition

7 Decision Problems

7.1 Definition

Define decision problem; a function with inputs x_i and boolean output "answer" a_o

$$X_i = \{x_1, ..., x_n\}$$
$$D := f[X_i] \to a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$

8 Solutions

8.1 Definition

Program P is a solution s^+ if P outputs answer a_o for all inputs X_i $\forall i$
s^+ is a function of the number of inputs n

$$X_i = \{x_1, ..., x_n\}$$
$$D := f[X_i] \to a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s^+ = s^+[n] := P[X_i] \to y_o : y_o = a_o \quad \forall X_i$$
$$P[X_i] = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, X_i, y_o\}$$
$$s^+ = P[X_i] = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, X_i, y_o\} \quad \forall X_i$$

8.2 Definition of S^+

Define S^+; the set of solutions to decision problem D

$$X_i = \{x_1, ..., x_n\}$$
$$D := f[X_i] \to a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s_j^+ := P[X_i] \to y_o : y_o = a_o \quad \forall X_i$$
$$S^+ := \{s_j^+, ...\} \quad \forall j$$

8.3 Definition of Solvable

Define solvable

$$X_i = \{x_1, ..., x_n\}$$
$$D := f[X_i] \to a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$

$$solvable := solvable[D] \to b_o \in \{\mathbb{T}, \mathbb{F}\} =$$
$$\exists s^+ : s^+ = P[X_i] \to y_o : y_o = a_o \quad \forall X_i$$

9 The set of all Decision Problems $\mathbb{D}$

9.1 Definition

Define the set of decision problems $\mathbb{D}$

$$X_i = \{x_1, ..., x_n\}$$
$$D_j := f_j[X_i] \to a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$\mathbb{D} := \{D_j, ...\} \quad \forall j$$

10 Instruction and Memory Notation

Define $\mathcal{L}$ a set of logical operations
Define $\mathcal{M}$ a set of bits "memory"

$$P[X_i] \to y_o = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, X_i, y_o\}$$
$$\mathcal{L} := \{s_1, s_2, ..., s_{O_T[n]}\}$$
$$\mathcal{M} := \{b_1, b_2, ..., b_{O_S[n]}\}$$
$$P[X_i] = \{\mathcal{L}, \mathcal{M}, X_i, y_o\}$$

11 Complexity

11.1 Time Complexity of a Decision Problem $O_T[n]$

Define Time Complexity $O_T[n]$ of Decision Problem D with solution s^+

$$X_i = \{x_1, ..., x_n\}$$
$$D := f[X_i] \to a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s^+ := P[X_i] \to y_o : y_o = a_o \quad \forall X_i =$$
$$\{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, X_i, y_o\} = \{\mathcal{L}, \mathcal{M}, X_i, y_o\}$$
$$O_T[n] := |\mathcal{L}| = N$$

11.2 Space Complexity $O_S[n]$

Define Space Complexity $O_S[n]$ of Decision Problem D with solution s^+

$$X_i = \{x_1, ..., x_n\}$$
$$D := f[X_i] \rightarrow a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s^+ := P[X_i] \rightarrow y_o : y_o = a_o \quad \forall X_i =$$
$$\{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, X_i, y_o\} = \{\mathcal{L}, \mathcal{M}, X_i, y_o\}$$
$$O_S[n] := |\mathcal{M}| = M$$

12 Definition of Complexity

Define Complexity $O[n]$ as a vector of dimension C

$$\mathbf{O}[n] := \; < O_T[n], O_S[n], O_3[n], O_4[n]..., O_C[n] >$$

13 Total Complexity

$$O[n] := O_T[n] + O_S[n] + \textstyle\sum_{i=3}^{C} O_i[n]$$

14 Simple Computational Complexity

The remainder of this chapter assumes simple computational complexity of dimension 2

14.1 Definition

Define simple computational complexity of dimension 2

$$\mathbf{O}[n] := \ < O_T[n], O_S[n] >$$

14.2 Time Complexity

Restate definition of Time Complexity $O_T[n]$

$$s^+ = \{\mathcal{L}, \mathcal{M}, X_i, y_o\}$$

$$O_T[n] := |\mathcal{L}| = N$$

14.3 Space Complexity

Restate definition of Time Complexity $O_S[n]$

$$s^+ = \{\mathcal{L}, \mathcal{M}, X_i, y_o\}$$
$$O_S[n] := |\mathcal{M}| = M$$

14.4 Total Complexity

$$O[n] := O_T[n] + O_S[n]$$
$$= |\mathcal{L}| + |\mathcal{M}| = N + M$$

15 Optimal Complexity

15.1 Definition

Define Optimal Complexity; the minimum total complexity required to solve a decision problem

$$O_{opt}[n] :=$$
$$\nexists \hat{O}[n] : \hat{O}[n] < O_{opt}[n] \;\; \forall n$$

15.2 Proof of Existence

Prove the existence of at least one $O_{min}[n]$ by induction/contradiction

16 Optimal solution

Define an optimal solution s^+_{opt}

16.1 Definition

$$X_i = \{x_1, ..., x_n\}$$
$$D_j := f[X_i] \rightarrow a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s^+ := P[X_i] \rightarrow y_o : y_o = a_o \quad \forall X_i$$
$$s^+_{opt} := s^+ :$$
$$\nexists \hat{O}[n] < O_{opt}[n] \quad \forall n, \; s^+ \in S^+_j$$

16.2 Optimal Time Complexity Solution

$$X_i = \{x_1, ..., x_n\}$$
$$D_j := f[X_i] \rightarrow a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s^+ := P[X_i] \rightarrow y_o : y_o = a_o \quad \forall X_i =$$
$$\{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, X_i, y_o\} = \{\mathcal{L}, \mathcal{M}, X_i, y_o\}$$
$$O_T[n] := |\mathcal{L}| = N$$
$$s^+_T := s^+ :$$
$$\nexists \hat{O_T}[n] < O_T[n] \quad \forall n, \; s^+ \in S^+_j$$

16.3 Optimal Space Complexity Solution

$$X_i = \{x_1, ..., x_n\}$$

$$D_j := f[X_i] \rightarrow a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$

$$s^+ := P[X_i] \rightarrow y_o : y_o = a_o \quad \forall X_i =$$

$$\{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, X_i, y_o\} = \{\mathcal{L}, \mathcal{M}, X_i, y_o\}$$

$$O_S[n] := |\mathcal{M}| = M$$

$$s_S^+ := s^+ :$$

$$\nexists \hat{O}_S[n] < O_S[n] \quad \forall n, \ s^+ \in S_j^+$$

17 Polynomial Complexity

17.1 Definition

Decision problem D with solution s^+ has (optimal) total complexity $O[n]$ bounded by polynomial complexity if

$$\exists K, C, \lambda_1 ... \lambda_K \quad :$$
$$O_{opt}[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

17.2 Polynomial Problems

Define $\mathbb{P}$, the set of Decision Problems that can be solved with Polynomial Complexity

$$\mathbb{P} := \{D_1, D_2, ...\} :$$
$$\exists K, C, \lambda_1 ... \lambda_K :$$
$$O_{opt}[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n, D_j \in \mathbb{P}$$

17.3 Polynomial Order of Complexity

Total complexity $O[n]$ is said to be of order K_{opt}

$$O[n] \sim K_{opt}$$
$$O_{opt}[n] := O[n] :$$
$$\nexists \hat{O}[n] < O_{opt}[n] \;\; \forall n$$
$$O_{opt}[n] < (\lambda_{K_{opt}} n)^{K_{opt}} + (\lambda_{K_{opt}-1} n)^{K_{opt}-1} ... + \lambda_1 n + C \quad \forall n$$
$$K_{opt} := K :$$
$$\nexists \hat{K} : O_T[n] < (\lambda_{\hat{K}} n)^{\hat{K}} + (\lambda_{\hat{K}-1} n)^{\hat{K}-1} ... + \lambda_1 n + C \quad \forall n, \; \hat{K} < K$$

17.4 Corrolary of Optimal Complexity

$$\nexists s^+ \in S^+ :$$
$$O_T[n] < (\lambda_{\hat{K}} n)^{\hat{K}} + (\lambda_{\hat{K}-1} n)^{\hat{K}-1} ... + \lambda_1 n + C \quad \forall n, \; \hat{K} < K_{opt}$$

17.4.1 Proof

Proof by contradiction; definition of optimal complexity

17.5 Property of Polynomial Complexity 1

lim n->inf O[n+1] / O[n] approaches 1

17.6 Property of Polynomial Complexity 2

lim n->inf O[n+1] - O[n] approaches

18 Polynomial Time Complexity

18.1 Definition

Decision problem D with (optimal) Time Complexity $O_T[n]$ is bounded by polynomial time complexity if

$$\exists K, C, \lambda_1 ... \lambda_K \quad :$$
$$O_T[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

18.2 Polynomial Time Problems

Define $\mathbb{P}_{time}$, the set of Decision Problems that can be solved with polynomial time complexity

$$\mathbb{P}_{time} := \{D_1, D_2, ...\} :$$
$$\exists K, C, \lambda_1 ... \lambda_K :$$
$$O_T[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n, D_j \in \mathbb{P}$$

18.3 Total Polynomial Complexity Implies Time bounded Polynomial Complexity

Proof Trivial

$$D \in \mathbb{P} \Longrightarrow D \in \mathbb{P}_{time}$$

18.4 Time bounded Polynomial Complexity implies Total Polynomial Complexiy

Proof harder use O[n] = |L| + |M|

$$D \in \mathbb{P}_{time} \Longrightarrow D \in \mathbb{P}$$

18.5 Polynomial Time Complexity iff Polynomial Complexity

Follows from previous 2 sections

18.6 Property of Polynomial Time Complexity 1

lim n->inf O[n+1] / O[n] approaches 1

18.7 Property of Polynomial Time Complexity 2

lim n->inf O[n+1] - O[n] approaches

18.8 Order of Complexity

Time complexity $O_T[n]$ is said to be of order

$$O_T[n]$$

19 Polynomial Space Complexity

19.1 Defintion

19.2 Polynomial Space Problems

19.3 Total Polynomial Complexity Implies Space bounded Polynomial Complexity

19.4 Space Bounded Polynomial Complexity Implies Total Polynomial Complexity

19.5 Polynomial Space Complexity iff Polynomial Complexity

19.6 Property of Polynomial Space Complexity 1

19.7 Property of Polynomial Space Complexity 2

19.8 Order of Complexity

20 Proof of the existence of $\mathbb{P}$

Trivial; N sum problem

21 Polynomial Duality

21.1 D $\in \mathbb{P}$ iff polynomial in Time

21.2 D $\in \mathbb{P}$ iff polynomial in Space

21.3 Polynomial in time iff Polynomial in space

21.4 Theorem Either OT or OS is on the order of Oopt

Proof by contradiction

21.5 Duality Functions

21.6 Theorem of Polynomial Duality

For all Problems in P there exists a duality function
Formally define dynamic programming, Optimal polynomial complexity minimizes the difference between time and space complexity order

$$D \in \mathbb{P}$$
$$O[n] := O_T[n] + O_S[n]$$
$$O_{opt}[n] := O[n] :$$
$$\nexists \hat{O}[n] < O[n] \quad \forall n$$

$$O_T^+[n] := |\mathcal{L}| = N :$$
$$\nexists \hat{O_T}[n] < O_T^+[n] \quad \forall n$$
$$O_S^+[n] := |\mathcal{M}| = M$$
$$\nexists \hat{O_S}[n] < O_S^+[n] \quad \forall n$$

21.7 Proof

Prove that Order can be subtracted from Os or Ot and added to the other; double check cauchy schwartz inequal

$$O[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n, D_j \in \mathbb{P}$$
$$O[n] := O_T[n] + O_S[n]$$
$$O_T[n] + O_S[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n, D_j \in \mathbb{P}$$

21.8 There exists an optimal OT and OS on the order of Oopt

21.9 Even ordered decision problems

M-N = N-M = 0

21.10 Odd ordered decision problems

N = M + 1 or M = N + 1

22 Definition of Non-Polynomial Problems

Define $\mathcal{N}$, the set of Decision Problems that cannot be solved with Polynomial Time Complexity

$$\mathcal{N} := \{D_1, D_2, ...\}$$
$$\nexists K, C, \lambda_1 ... \lambda_K :$$
$$O_T[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n, D_j \in \mathcal{N}$$

23 Divergent Problems

$$\hat{\mathcal{D}} := \{D_1, D_2, ...\}$$

$$\hat{O}_T[n] \geqslant n^n \quad \forall n, D_j \in \mathcal{N}$$

23.1 Property of Divergent Problems 1

23.2 Property of Divergent Problems 2

24 Fundamental Theorem of Computation

24.1 Prove the universal bound of Polynomial Problems is n^n

24.2 Non-Polynomial implies Divergent

25 Proof of the existence of $\hat{\mathcal{D}}$

Non-trivial; Formalize the traveling salesman problem as a decision problem (any optimization problem)

25.1 The Traveling Salesman Problem

English description

25.2 Formal Definition

$$X_i = \{c_1, c_2, ..., c_n\} :$$
$$dim[c_i] = C > 1$$

$$\bar{X}_i = \{c_1, c_2, ..., c_n, \bar{P}, \bar{f}[c_i, c_j]\}$$
$$\bar{P} := \{c_k, ...\} :$$
$$\exists c_k \in \bar{P} \;\; \forall c_k \in X_i$$

25.3 Prove Traveling Salesman Problem is in $\hat{\mathcal{D}}$

Either show not in $\mathbb{P}$ or in $\hat{\mathcal{D}}$

25.3.1 Determine a duality function for The Traveling Salesman Problem

25.3.2 Show that the limit of O[n+1]/O[n] is a function of n

25.3.3 Show that The Traveling Salesman Problem must be in $\hat{\mathcal{D}}$

26 Proof of "P $\neq$ NP"

27 Theorem of Prime Numbers "Riemann Hypothesis"

Riemann Zeta Function

$$\zeta(s) \equiv \sum_{n=1}^{\infty} \frac{1}{n^s} \quad [2]$$

"The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2." [2]

Prove the problem is divergent
There fore it can only be proven to a certain degree
The limit as n approaches infinity implies a real part of one half
Connection with the real and imaginary part of O $[n]$

27.1 Prove O_{opt} is performing O_{opt} recursively for the ints less than square root of n

Testing the primes less than sqrt(n)? double check
1. Optimal solution for n=1,2,3, everything else is a recursive optimal proof by induction
Time Complexity seems to be on the order of n log n... implies divergence or lack of bound? Add in the complexity of division.. probably approaches n^n

27.2 Since divergent, no s^+ exists.. only rules

Express as a limit

27.3 Show that the limit as n $\rightarrow \infty$ implies the real part is 1/2

$1/2 \pm 14.134725$ i $1/2 \pm 21.022040$ i $1/2 \pm 25.010858$ i $1/2 \pm 30.424876$ i $1/2 \pm 32.935062$ i $1/2 \pm 37.586178$ i

Z = $\zeta(1/2 + it)$

27.4 Notation, real imaginary parts of the problem

Even numbers and numbers ending in 5 are automatically convergent
Testing numbers ending in 1,3,7,9 results in divergent expression
we can continue to add rules to a certain degree

28 Divergent Problems

Define $\hat{\mathbb{D}}$ the set of decision problems with no convergent?/finite? solution $\hat{D}_j$

$$\hat{\mathbb{D}} := \{\hat{D}_j, ...\}$$
$$\hat{D}_j \in \mathbb{D} \quad \forall j$$
$$\nexists \hat{s}^+ \in S^+ : \hat{s}^+ \text{ solves } \hat{D}_j \quad \forall \hat{D}_j \in \hat{\mathbb{D}} \Longleftrightarrow$$
$$\nexists s^+ \in S^+ : O_j[n] < n^n \quad \forall n, j$$

There exists no such solution such that O[n] $< n^n$, but there is a right and wrong answer
Either here or in the next chapter we'll prove you can only solve to a certain degree
!!! There exists no such solution such that O[n] $< n^n$ $\quad \forall$ n

28.1 Definition

$$\hat{O}[n] := n^n$$
$$limit_{n\to\infty} \frac{O_{opt}[n]}{\hat{O}[n]} \ \ diverges$$
$$diverges[\ O_{opt}[n], \hat{O}[n]\ ] \to \mathbb{T}$$

28.2 Theorem of Divergent Programs

Prove that Divergent implies not in polynomial (trivial)
Prove that Divergent implies Non-polynomial (trial after proving above)
Show that there exists at least one member of Divergent

29 Properties of Solvable and Divergent problems

29.1 Solvable and Divergent are disjoint

Prove by contradiction

30 "Theorem of Divergent Programs"

30.1 Divergence Test

1. Let $d_j \in D$
2. $d_j = (d_j \in \hat{\mathcal{D}}) \cup (d_j \in$ set of solvable problems) by disjoint condition of solvable and divergent
3. Let $O_{opt}[n]$, the optimal complexity of d_j
4. $\rightarrow s_i^+$ that solve d_j have larger complexity $\forall i$
5. 2 implies $O_{opt}[n]$ is either bounded by n^n or not
6. $\hat{O}[n] \equiv n^n$
7. Easy Suppose $d_j \in$solvable$limit_{n\rightarrow\infty}\frac{O_{opt}[n]}{\hat{O}[n]} = 0$
8.Suppose $d_j \in \hat{\mathcal{D}} limit_{n\rightarrow\infty}\frac{O_{opt}[n]}{\hat{O}[n]} \neq 0$ (by disjoint condition)

$$limit_{n\rightarrow\infty}\frac{O_{opt}[n]}{\hat{O}[n]} = 1$$

31 Connection to verification in polynomial time

32 Fundamental Theorem of Computation

n^n or $\lambda n^n + C$ the universal bound to solvable computational complexity $(\lambda n)^n$ + C ?

32.1 Time Complexity Argument

Suppose decision problem d with optimal time complexity $O_{T_{min}}[n]$ and solution s^+, an arbitrary decision problem in P with polynomial complexity
Assumptions
1. $d \in P$, $s^+ \in S^+$
Assertions
2. $\exists K, C, \lambda_1 ... \lambda_K \; : \; O_{T_{min}}[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C, \quad \forall n$
3. Define $f[K, C, \lambda_1, ... \lambda_K] \equiv (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C$
4. $\exists K, C, \lambda_1 ... \lambda_K : O_{T_{min}}[n] < f[K, C, \lambda_1, ... \lambda_K] \quad \forall n$
5. Let $\hat{s}^+ \equiv \circlearrowright^n [s^+]$
6. $O_T[n] <= \hat{O}_T[n]$ (by defition of nested loop)

7. $\hat{O}_{T_{min}}[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C$
8. $\hat{O}_{T_{min}}[n] < limit_{n \to \infty} \circlearrowright^n [s^+]$ (by definition of limit + definition of nested loop, expand to show full derivation, valid because this is a series, probably need to show limit applies)
9. $\therefore O_{T_{min}}[n] < \hat{O}_{T_{min}}[n] < n^n = limit_{n \to \infty} \circlearrowright^n [s^+]$
I want to say forall n but seems refutable for n= 1,2.. but as n approach infinity it's a contradiction to say a solvable problem in P $\hat{O}_{T_{min}} = n^n \quad \forall n$
10. For "sufficiently large n"

$$\nexists \hat{s}^+ \in S^+ : |\hat{s}^+| \equiv O_{T_{min}}[n] < n^n, \quad \forall n$$
$$\hat{O}[n] \equiv n^n$$

32.2 Space Argument

Similar but additional notation required?

33 Divergence Criterion

Necessary condition for divergent program, iff
or you can show there exists no lambda, C such that O $[n]$ is n^n is bounded by $\lambda n^n + C$ for all n

$limit_{n \to \infty}$ div / solvable > 1

Assumptions

1. Define the "Null Space of $\mathcal{D}$" or "Null Set" $O_\perp$

$$O_\perp = \{\hat{d}_1, \hat{d}_2, ..., \hat{d}_j\}, \quad j > 0$$
$$\hat{O}_j[n] \equiv (O[n])^n, \forall j$$

2. $O_P \cup O_\mathcal{N} = \mathcal{D}$ (by definition)

Assertions

3. $O_P \cap O_\perp = \varnothing$
4. Let $O_\mathcal{N} \cap O_\perp = \hat{O} = \{\hat{O}_i, ...\}$, $i > 0$
5. Consider $D_j \in O_\mathcal{N}$
6. D_j has finite complexity by definition

$$O_j[n] = C$$

7. D_j has at least one optimal solution by the necessity of optimal solution (theorem Z)

$$O_j[n] = C$$

34 Proof of "P $\neq$ NP"

34.1 Proof N implies D

Is trivial by implication of Theorem x and Theorem y

$$\mathcal{N} \equiv \{d_j, ..\} \quad \forall j, \mathcal{N} \in D$$
$$\nexists K, C, \lambda_1 ... \lambda_K :$$
$$O_T[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C, \quad \forall n, \ \forall d_j \in \mathcal{N}$$

1. $\rightarrow \mathcal{P} \cap \mathcal{N} = \varnothing$ by definition of P,N
2. $d_i \in \hat{D} \vee d_i \in$solvable
3. $1 \rightarrow d_i \notin$ solvable
4. $\therefore d_i \in \hat{D} \quad \forall i$ (theorem y) Show that Definition of Non-Polynomial Problems automatically implies Divergent

1. We've proven Solvable Union are disjoint and complete set P 2. N not in P by definition 3. therefore N in divergence by set theory

Currently we have only defined solvable problems and divergent problems
Additionally polynomial problem which the existance of is trivial
Plus we defined non-polynomial complexity
Prove the existence of $\mathcal{N}$ the set of non polynomial problems

34.2 Proof that D implies N

34.3 D iff N

Show $O[n]$ in the $\varnothing$ the set of problems with $n^n > O[n] > n^k + c$
Proving there's Polynomial and Divergent, in the set of all decision problems

A neat follow up, tie in the definition of $\mathcal{N}$ implies membership to divergent problems

35 Prove the existance of D = N,The Traveling Salesman Problem

Define the traveling salesman problem, prove it is divergent and has the same solution as current approaches
Consider proving with both definition and necessary condition

35.1 Compute every sub path or recursive subpaths in memory

Trade off between time and space, $O_{salesman}[n]$ diverges with a polynomial $O_P[n]$

36 Prove Polynomial and Divergent problems are Complements

Implied by the previous sections

37 Solvable Union Divergent = all decision problems

Trivial as a result of the previous section by definition of Ω

$$\mathbb{P} \cup \hat{\mathbb{D}} = \mathbb{D}$$

38 Theorem of Prime Numbers "Riemann Hypothesis"

Riemann Zeta Function

$$\zeta(s) \equiv \sum_{n=1}^{\infty} \frac{1}{n^s} \quad [2]$$

"The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2." [2]

Prove the problem is divergent
There fore it can only be proven to a certain degree
The limit as n approaches infinity implies a real part of one half
Connection with the real and imaginary part of O $[n]$

38.1 Prove O_{opt} is performing O_{opt} recursively for the ints less than square root of n

Testing the primes less than sqrt(n)? double check
1. Optimal solution for n=1,2,3, everything else is a recursive optimal proof by induction
Time Complexity seems to be on the order of n log n... implies divergence or lack of bound? Add in the complexity of division.. probably approaches n^n

38.2 Since divergent, no s^+ exists.. only rules

Express as a limit

38.3 Show that the limit as n $\rightarrow \infty$ implies the real part is 1/2

1/2 $\pm$ 14.134725 i 1/2 $\pm$ 21.022040 i 1/2 $\pm$ 25.010858 i 1/2 $\pm$ 30.424876 i 1/2 $\pm$ 32.935062 i 1/2 $\pm$ 37.586178 i

Z $= \zeta(1/2 + it)$

38.4 Notation, real imaginary parts of the problem

Even numbers and numbers ending in 5 are automatically convergent
Testing numbers ending in 1,3,7,9 results in divergent expression
we can continue to add rules to a certain degree

Citations

[1] $https://www.claymath.org/millennium-problems$
[2] $https://www.claymath.org/sites/default/files/official\_problem\_description.pdf$
[3] $http://www.math.uchicago.edu/ may/VIGRE/VIGRE2011/REUPapers/Riffer-Reinert.pdf$