1 Instruction

1.1 Definiton of Instruction

Define Instruction s

$$Instruction\ s :=$$
$$(s \equiv function) \oplus (s \equiv operation)$$

2 Instruction Set

2.1 Definiton of Instruction Set

Define $\mathcal{I}$; an ordered Set of Instructions s_i

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

2.2 Abstraction Notation

Define $\mathcal{I}$; an ordered Set of Instructions s_i

$$\text{Instruction Set } \mathcal{I} :$$
$$(\mathcal{I} \equiv \text{Set}) \wedge (s_i \equiv \text{Instruction} \quad \forall s_i \in \mathcal{I})$$

3 Memory

3.1 Definiton of Memory

$$Memory\ m :=$$
$$m \in \Omega$$

3.2 Definition of Memory Set

Define Memory Set $\mathcal{M}$; an ordered set of objects m_i

$$\mathcal{M} := \{m_1, m_2, ..., m_M\}$$

3.3 Abstraction Notation

Define Memory $\mathcal{M}$; an ordered set of either bools numbers or objects m_i

$$\text{"Memory Set"} := \mathcal{M} :$$
$$(\mathcal{M} \equiv Set) \wedge (m_i \equiv Object\ \ \forall m_i \in \mathcal{M})$$

4 Program

4.1 Definition of a Program

Given Instruction Set $\mathcal{I}$, Memory Set $\mathcal{M}$, and Set X; Define Program $\mathcal{P}$[X]

Given

$$\mathcal{I} = \{s_1, s_2, ..., s_N\}, \;\; \mathcal{M} = \{m_1, m_2, ..., m_M\}, \;\; X = \{x_1, x_2, ..., x_n\}$$

$$\mathcal{P}[X] := \{s_1[\hat{X}_1, \hat{\mathcal{M}}_1], s_2[\hat{X}_2, \hat{\mathcal{M}}_2], ..., s_N[\hat{X}_N, \hat{\mathcal{M}}_N]\} \to Y :$$
$$(\hat{X}_i \subseteq X \quad \forall i \leqslant N) \wedge (\hat{\mathcal{M}}_i \subseteq \mathcal{M} \quad \forall i \leqslant N) \wedge (Y \in \Omega \quad \forall X \in \mathbb{D}_{\mathcal{P}})$$

4.2 Abstraction Notation

Define a method $\mathcal{P}$; a tensor of instructions and memory

$$Program \;\; P = P \to Y :$$
$$(\mathcal{P} \equiv \text{Set}) \wedge (p_i \equiv Instruction \quad \forall p_i \in \mathcal{P})$$
$$\wedge$$
$$(p_i.input \subseteq \mathcal{M} \;\; \forall p_i \in \mathcal{P}) \wedge (Y \in \Omega)$$

5 Workers

5.1 Definition of a Worker

6 Boolean Methods

6.1 Definition of Boolean Methods

Define a boolean method; a method with inputs x_i and boolean output y

$$X_n = \{x_1, ..., x_n\} \ : \ x_i \in \Omega \ \ \forall x_i \in X_n$$

$$P_{boolean}[X_n] := \{s_1, s_2, ..., s_N \mid y, b_2, ..., b_M\} \rightarrow y \ : \ y \ \in \{\mathbb{T}, \mathbb{F}\} \ \ \forall X_n \in \mathbb{D}_P$$

6.2 Abstraction Notation

$$\text{"Boolean Method"} := P_{boolean}[X_n] :$$

$$(P_{boolean}[X_n] \equiv \text{method}) \wedge (\exists \mathbb{D}_P : X_n \in \mathbb{D}_P \Leftrightarrow P_{boolean}[X_n] \rightarrow y \equiv bool)$$

6.3 Boolean Method Output

The output of a boolean method y is read as "the result"

$$P_{boolean}[X_n] \rightarrow y$$

"Boolean Method $P_{boolean}$ with input(s) X_n outputs result y"

7 True or False Questions

7.1 Definition of True or False Question

Define a True or False Question; a bounded boolean expression given input(s) X_n

Given

$$X_n = \{x_1, x_2, ..., x_n\}$$

$$Q^{T/F} := \; Q[X_n] \; : \; Q[X_n] \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_n \in \mathbb{D}_Q$$

A True or False Question is sometimes read as a "decision question"

7.2 Abstraction Notation

Given

$$X_n = \{x_1, x_2, ..., x_n\}$$

$$\text{"True or False Question"} := Q^{T/F}[X_n] :$$

$$\exists \mathbb{D}_Q : (X_n \in \mathbb{D}_Q \Leftrightarrow Q^{T/F}[X_n] \equiv bool)$$

8 Approach to a True or False Question

?? Does ($\mathbb{D}_S == \mathbb{D}_Q$) or merely contained/bounded???

8.1 Definition of an Approach to a True or False Question

Given input(s) X_n and True or False Question $Q^{T/F}$; Define an approach to True or False Question $Q^{T/F}$; a boolean method bounded by the domain of $Q^{T/F}$

Given

$$X_n = \{x_1, x_2, ..., x_n\}$$
$$Q^{T/F}[X_n]$$

$$S_Q[X_n] := P_Q[X_n] = \{s_1, s_2, ..., s_N \mid y, b_2, ..., b_M\} :$$
$$(P_Q[X_n] \to y) \wedge (y \in \{\mathbb{T}, \mathbb{F}\}) \quad \forall X_n \in \mathbb{D}_Q$$

8.2 Abstraction Notation

Given

$$X_n = \{x_1, x_2, ..., x_n\}$$
$$Q^{T/F}[X_n]$$

$$\text{"Approach to True or False Question } Q\text{"} := S_Q[X_n] :$$
$$(S_Q[X_n] \equiv\equiv \text{boolean method}) \wedge (\mathbb{D}_S == \mathbb{D}_Q)$$

Solution to a True or False Question

8.3 Definition of Solution to a True or False Question

Given input(s) X_n and True or False Question $Q^{T/F}$;
Define a solution of True or False Question $Q^{T/F}$; an approach to $Q^{T/F}$ asserting output y is the answer for all input(s) in the domain of Q

Given

$$X_n = \{x_1, x_2, ..., x_n\}$$
$$Q = Q^{T/F}[X_n]$$

$$S_Q^+[X_n] := P_Q[X_n] = \{s_1, s_2, ..., s_N \mid y, b_2, ..., b_M\} :$$
$$(P_Q[X_n] \to y) \wedge (y == a) \quad \forall X_n \in \mathbb{D}_Q$$

8.4 Abstraction Notation

Given

$$X_n = \{x_1, x_2, ..., x_n\}$$
$$Q = Q^{T/F}[X_n]$$

$$\text{"Solution to True or False Question } Q\text{"} := S_Q^+[X_n] :$$
$$(S_Q^+[X_n] \equiv\equiv \text{approach to a True or False Question Q}) \wedge$$
$$(S_Q^+[X_n].output == |Q^{T/F}[X_n]|)$$

Incomplete Solution to a True or False Question

8.5 Definition of Incomplete Solution to a True or False Question

8.6 Abstraction Notation

9 Complexity of a Method

9.1 Definition of a Method's Complexity

Given

$$\mathcal{P} = \{s_1, s_2, ..., s_N \mid y, b_2, ..., b_M\}$$

$$\begin{aligned} \mathbf{O}_{\mathcal{P}} &:=< |\{s_1, s_2, ..., s_N\}|, |\{y, b_2, ..., b_M\}| > \\ &= < |\mathcal{I}|, |\mathcal{M}| > = < N, M > \end{aligned}$$

9.2 Abstraction Notation

$$\text{"Method"} := \mathcal{P} :$$
$$(\mathcal{P} \equiv \text{Tensor}) \wedge (\mathcal{P} \supseteq \mathcal{I}) \wedge (\mathcal{P} \supseteq \mathcal{M})$$

1. By definition

$$\mathcal{P} \equiv \text{tensor}$$

2. By inheritence of Tensor

$$\mathcal{P} \;\; \overset{a}{\supseteq} \;\; \text{size}$$

$\overset{a}{\supseteq}$is read as "has the attribute of"

3. Therefore size of $\mathcal{P}$ is defined

$$\therefore |\mathcal{P}| :=< |S_1|, |S_2|, ... >$$

4. By definition

$$(\mathcal{P} \supseteq \mathcal{I}) \wedge (\mathcal{P} \supseteq \mathcal{M})$$

5. Let

$$|\mathcal{P}| = \mathbf{O}_P = \text{"Complexity of Method } \mathcal{P}\text{"}$$

6. By 3, 4, and 5

$$\text{"Complexity of Method } \mathcal{P}\text{"} = \mathbf{O}_P :=< |\mathcal{I}|, |\mathcal{M}| >$$

7. By definition of $\mathcal{I}, \mathcal{M}$

$$\mathcal{I} \equiv \text{Set} \wedge \mathcal{M} \equiv \text{Set}$$

8. By inheritance of Sets

$$\mathcal{I} \;\trianglerighteq\; \text{count} \wedge \mathcal{M} \;\trianglerighteq\; \text{count}$$

9. Therefore count of $\mathcal{I}$ and $\mathcal{M}$ is defined; by definition of $\mathcal{I}$ and $\mathcal{M}$

$$\therefore |\mathcal{I}| = |\{s_1, s_2, ..., s_N\}| := N; \;\; |\mathcal{M}| = \{y, b_2, ..., b_M\} := M$$

10. By 6 and 9

$$\text{"Complexity of Method } \mathcal{P}\text{"} = \mathbf{O}_P :=< |\mathcal{I}|, |\mathcal{M}| >= \; < N, M >$$

10 Total Complexity of a Method

11 Simple Computational Complexity

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

11.1 Definition

Define simple computational complexity of dimension 2

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

11.2 Total Complexity

Define Total Complexity of solution s^+

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

$$= |\mathcal{L}| + |\mathcal{M}| + |y_o| = N + M + 1$$

11.3 Time Complexity

Restate definition of Time Complexity $O_T[n]$ of solution s^+

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

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

11.4 Space Complexity

Restate definition of Time Complexity $O_S[n]$ of solution s^+

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

$$O_S[n] := |\mathcal{M}| + |y_o| = M + 1$$

11.5 Total Complexity as a Function of Time and Space Complexity

$$\begin{aligned} O[n] := |s^+[X_n]| &= |\{\mathcal{L}, \mathcal{M}, y_o\}| \\ &= |\mathcal{L}| + |\mathcal{M}| + |y_o| \\ &= O_T[n] + O_S[n] \end{aligned}$$

11.6 $O_S[n] > 0^*$

Assuming Program is not void

11.6.1 Proof

Assume $O_S[n] = 0$

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

$$O_S[n] = 0 \Rightarrow \mathcal{M} = y_o = \varnothing$$

$$y_o = \varnothing;\ \ y_o \in \{\mathbb{T}, \mathbb{F}\} \text{ by definition of } s^+$$

$\therefore O_S[n] = 0$ contradicts the definition of solution s^+ of a decision problem

$$O_S[n] \geqslant 0 \text{ by definition of magnitude}$$

$$\therefore O_S[n] > 0$$

11.7 $O_T[n] > 0^*$

Assuming Program is not void

11.7.1 Proof

Assume $O_T[n] = 0$

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

$$O_T[n] = 0 \Rightarrow y_o \notin \{\mathbb{T}, \mathbb{F}\}$$

$$y_o \notin \{\mathbb{T}, \mathbb{F}\};\ \ y_o \in \{\mathbb{T}, \mathbb{F}\} \text{ by definition of } s^+$$

$\therefore O_T[n] = 0$ contradicts the definition of solution s^+ of a decision problem

$$O_T[n] \geqslant 0 \text{ by definition of magnitude}$$

$$\therefore O_T[n] > 0$$

11.8 $O[n] > 0^*$

Assuming Program is not void

11.8.1 Proof

$$O[n] := O_T[n] + O_S[n]$$

$$O_T[n] > 0; \;\; O_S[n] > 0$$

$$\therefore O[n] > 0$$

11.9 $O[n] > O_T[n]*$

Assuming Program is not void

11.9.1 Proof

$$O[n] := O_T[n] + O_S[n]$$

$$O_S[n] > 0$$

$$\therefore O[n] > O_T[n]$$

11.10 $O[n] > O_S[n]^*$

Assuming Program is not void

11.10.1 Proof

$$O[n] := O_T[n] + O_S[n]$$

$$O_T[n] > 0$$

$$\therefore O[n] > O_S[n]$$

11.11 $O[n+1] \geqslant O[n]$

11.11.1 Proof

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$

$$O[n] = |\ s^+[X_i]|$$
$$O[n+1] = \hat{O}[n] = |\ s^+[\hat{X}_i]|$$

For general solutions s^+

$$
\begin{aligned}
&s^+[\hat{X}_i] \supseteq s^+[X_i] \\
&\Rightarrow |s^+[\hat{X}_i]| \geqslant |s^+[X_i]| \\
&\therefore \hat{O}[n] = O[n+1] \geqslant O[n]
\end{aligned}
$$

12 Definition of Problem

Define Problem (also denoted as Question); a function with input(s) X_n and "answer" Y

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

12.1 Set of Questions

Define $\mathbb{Q}$; the set of questions

$$\mathbb{Q} := \{Q_1, Q_2, ...\} :$$
$$Q_i = f[X_j] = Y_o \subseteq \Omega \ \ \forall X_j, i$$

12.2 Decision Questions / Decision Problems

12.2.1 Definition

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

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

12.3 Numerical Questions / Numerical Problems

12.3.1 Definition

Define numerical problem; a function with inputs x_i and numerical output y_o

$$X_i = \{x_1, ..., x_n\}$$
$$Q := f[X_i] = y_o \in \mathbb{R} \quad \forall X_i$$

12.4 System Questions / System Problems

12.4.1 Definition

Define system problem; a function with inputs x_i and outputs y_j

$$X_i = \{x_1, ..., x_n\}$$
$$Q := f[X_i] = Y_o = \{y_1, ..., y_m\} \quad \forall X_i$$

13 Solutions

13.1 Definition

Program P is a solution s^+ to decision problem D if

1. P outputs answer a_o for all inputs X_i $\forall i$
and
2. $s^+[X_i]$ is a subset of $s^+[\hat{X}_i]$

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_n, x_{n+1}, C\}$$
$$D := f[X_i] \rightarrow a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s^+ = s^+[X_i] := P:$$
$$(P[X_i] \rightarrow y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$

$$P[X_i] = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\}$$

$$s^+ = P[X_i] = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\} \quad \forall X_i$$

13.1.1 Property of No-op ;

No-op ; can be added to any solution S_i without modifying the output y_o or memory b_i

$$; := \varnothing$$
$$s^+ = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$\hat{s}^+ = \{s_1, s_2, ..., \; ; \; , ..., s_{O_T[n]+1}, \hat{b}_1, \hat{b}_2, ..., \hat{b}_{O_S[n]}, \hat{y}_o\}$$
$$\hat{y}_o = y_o \quad \forall k$$

13.2 Definition of S^+

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

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_n, x_{n+1}, C\}$$
$$D := f[X_i] \rightarrow a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$s_j^+ = s_j^+[X_i] := P:$$
$$(P[X_i] \rightarrow y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$
$$S^+ := \{s_j^+, ...\} \quad \forall j$$

13.3 Definition of Solvable

Define solvable

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_n, x_{n+1}, C\}$$
$$D := f[X_i] \to a_o \in \{\mathbb{T}, \mathbb{F}\} \quad \forall X_i$$
$$solvable = solvable[D] = b_o \in \{\mathbb{T}, \mathbb{F}\} :=$$
$$\exists P : (P[X_i] \to y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$

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

14.1 Definition

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

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

15 Complexity

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

Define Time Complexity $O_T[n]$ of solution s^+ to Decision Problem D as the total number of logical operations

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$

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

$$s^+[X_i] := P :$$
$$(P[X_i] \to y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$= \{\mathcal{L}, \mathcal{M}, y_o\}$$

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

15.2 Space Complexity $O_S[n]$

Define Space Complexity $O_S[n]$ of solution s^+ to Decision Problem D as the total number of memory elements

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$

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

$$s^+[X_i] := P :$$
$$(P[X_i] \to y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$= \{\mathcal{L}, \mathcal{M}, y_o\}$$

$$O_S[n] := |\mathcal{M}| + |y_o|^* = M + 1$$

*It is convention to reserve one memory element for output y_o.
Void programs do not require the y_o memory element for output

16 Definition of Complexity

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

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

17 Complexity of Canonical Instructions

$$c := a \leftarrow l[X_n]$$

18 Complexity of Computational Operations

18.1 $+$

Express the bounds of complexity for Computational Operation $+$

19 Inductive Functions

19.1 Inductive Function f_{n+1}

$$O[n] = O_T[n] + O_S[n]$$
$$O[n+1] = O_T[n+1] + O_S[n+1]$$

$$f_{n+1}[n] := O[n+1] - O[n]$$

19.2 Inductive Space and Time Formulas

$$f^T_{n+1}[n] := O_T[n+1] - O_T[n]$$
$$O_T[n+1] = O_T[n] + f^T_{n+1}[n]$$
$$f^S_{n+1}[n] := O_S[n+1] - O_S[n]$$
$$O_S[n+1] = O_S[n] + f^S_{n+1}[n]$$

19.3 Inductive Function Expressions

Relate $f_{n+1}[n]$ to equivalence functions

$$O[n] = O_T[n] + O_S[n]$$
$$O[n+1] = O_T[n+1] + O_S[n+1] = O[n] + f_{n+1}[n]$$
$$O_T[n] = O[n] - O_S[n]$$
$$O_S[n] = O[n] - O_T[n]$$

$$f_{n+1}[n] = O[n+1] - O[n]$$
$$f_{n+1}[n] = O_T[n+1] + O_S[n+1] - O[n]$$
$$f_{n+1}[n] = O_T[n+1] - O_T[n] + O_S[n+1] - O_S[n]$$
$$f_{n+1}[n] = O[n+1] - O_T[n] - O_S[n]$$
$$f_{n+1}[n] = f^T_{n+1}[n] + f^S_{n+1}[n]$$

19.4 Zero Order Space Inductive Function

$$Let\ \ O_S[n] \sim n^0$$
$$f_{n+1}[n] = O_T[n+1] - O_T[n] + O_S[n+1] - O_S[n] = O_T[n+1] - O_T[n]$$

20 Polynomial Complexity

20.1 Definition

Decision problem D with solution s^+ has polynomial total complexity $O[n]$ if

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

20.2 Polynomial Problems

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

$$\mathbb{P} := \{D_1, D_2, ...\} :$$

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

20.3 Polynomial Order of Complexity

Solution s^+ with total complexity $O[n]$ is said to be of order n^K

$$O[n] \sim n^K$$
$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

20.4 Property of Polynomial Complexity 1

Solutions with polynomial complexity have convergent complexity

$$lim_{n \to \infty} \frac{O[n+1]}{O[n]} = 1$$

20.4.1 Proof

$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C$$
$$O[n+1] = (\lambda_K (n+1))^K + (\lambda_{K-1} (n+1))^{K-1} ... + \lambda_1 (n+1) + C$$
$$= (\lambda_K n)^K + (\tilde{\lambda}_{K-1} n)^{K-1} ... + \tilde{\lambda_1} n + \tilde{C}$$

$$lim_{n \to \infty} \frac{O[n+1]}{O[n]}$$

$$= lim_{n \to \infty} \frac{(\lambda_K n)^K + (\tilde{\lambda}_{K-1} n)^{K-1} ... + \tilde{\lambda_1} n + \tilde{C}}{(\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C}$$

$$= lim_{n\to\infty}\frac{(\lambda_K n)^K}{(\lambda_K n)^K+(\lambda_{K-1}n)^{K-1}...+\lambda_1 n+C} + \frac{(\tilde{\lambda}_{K-1}n)^{K-1}}{(\lambda_K n)^K+(\lambda_{K-1}n)^{K-1}...+\lambda_1 n+C} + ... + \frac{\tilde{\lambda}_1 n}{(\lambda_K n)^K+(\lambda_{K-1}n)^{K-1}...+\lambda_1 n+C} + \frac{\tilde{C}}{(\lambda_K n)^K+(\lambda_{K-1}n)^{K-1}...+\lambda_1 n+C}$$

$$= 1 = lim_{n\to\infty}\frac{O[n+1]}{O[n]}$$

20.5 Property of Polynomial Complexity 2

$$\exists K, \hat{C}, \hat{\lambda}_1, ..., \hat{\lambda}_{K-1} :$$
$$O[n+1] - O[n] = f_{n+1}[n] = (\hat{\lambda}_{K-1}n)^{K-1}... + \hat{\lambda}_1 n + \hat{C} \quad \forall n$$

20.5.1 Proof

$$O[n] = (\lambda_K n)^K + (\lambda_{K-1}n)^{K-1}... + \lambda_1 n + C$$
$$O[n+1] = (\lambda_K(n+1))^K + (\lambda_{K-1}(n+1))^{K-1}... + \lambda_1(n+1) + C$$
$$= (\lambda_K n)^K + (\tilde{\lambda}_{K-1}n)^{K-1}... + \tilde{\lambda_1}n + \tilde{C}$$
$$O[n+1] - O[n] = ((\tilde{\lambda}_{K-1} - \lambda_{K-1})n)^{K-1}... + (\tilde{\lambda_1} - \lambda_1)n + (\tilde{C} - C)$$
$$O[n+1] - O[n] = (\hat{\lambda}_{K-1}n)^{K-1}... + \hat{\lambda}_1 n + \hat{C}$$

20.6 Property of Polynomial Complexity 3

$$limit_{n\to\infty}\frac{f_{n+1}[n]}{O[n]} = 0$$

20.6.1 Proof

$$limit_{n\to\infty}\frac{O[n+1]}{O[n]} = 1$$
$$limit_{n\to\infty}\frac{O[n]+f_{n+1}[n]}{O[n]} = 1$$
$$limit_{n\to\infty}\frac{O[n]}{O[n]} + \frac{f_{n+1}[n]]}{O[n]} = 1$$
$$limit_{n\to\infty}1 + \frac{f_{n+1}[n]]}{O[n]} = 1$$
$$limit_{n\to\infty}\frac{f_{n+1}[n]]}{O[n]} = 0$$

20.7 Total Polynomial Complexity Implies Time bounded Polynomial Complexity

$$D \in \mathbb{P} \Longrightarrow O_T[n] < (\lambda_K n)^K + (\lambda_{K-1}n)^{K-1}... + \lambda_1 n + C$$

20.7.1 Proof

$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \ \ \forall n$$

$$O[n] := O_T[n] + O_S[n]; \ \ O_S[n] > 0$$

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

20.8 Total Polynomial Complexity Implies Space bounded Polynomial Complexity

$$D \in \mathbb{P} \Longrightarrow O_S[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C$$

20.8.1 Proof

$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \ \ \forall n$$

$$O[n] := O_T[n] + O_S[n]; \ \ O_T[n] > 0$$

$$\therefore O_S[n] < (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \ \ \forall n$$

20.9 Polynomial Complexity in Space and Time Implies Polynomial Total Complexity

$$(O_S[n] == (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + \lambda_0)$$

$$\wedge$$

$$(O_T[n] == (\hat{\lambda}_M n)^M + (\hat{\lambda}_{M-1} n)^{M-1} ... + \hat{\lambda}_1 n + \hat{\lambda}_0)$$

$$\Rightarrow D \in \mathbb{P}$$

20.9.1 Proof

$$O_S[n] = \lambda_K n^K + \lambda_{K-1} n^{K-1} + ... + \lambda_1 n + \lambda_0$$

$$O_T[n] = \hat{\lambda}_M n^M + \hat{\lambda}_{M-1} n^{M-1} + ... + \hat{\lambda}_1 n + \hat{\lambda}_0$$

$$O[n] = O_S[n] + O_T[n]$$

$$^*O[n] = (\hat{\lambda}_0 + \lambda_0) + n(\lambda_1 + \hat{\lambda_1}) + ... + n^K(\lambda_K + \hat{\lambda_K}) + \hat{\lambda}_{K+1} n^{K+1} + ... + \hat{\lambda}_M n^M$$

$$\therefore O[n] \text{ has polynomial total complexity by definition}$$

* Assume K < M, similar proof for K=M, K>M

21 Non-Polynomial Complexity

21.1 Definition

Decision problem $\tilde{D}$ with solution s^+ has non-polynomial total complexity $O[n]$ if

$$\nexists K, \lambda_0, \lambda_1, ..., \lambda_K \ :$$
$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

21.2 Non-Polynomial Problems

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

$$\mathcal{N} := \{\tilde{D}_1, \tilde{D}_2, ...\} :$$
$$\nexists K, C, \lambda_1 ... \lambda_K \ :$$
$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n, \ s^+ \in S_i^+, \ \tilde{D}_i \in \mathcal{N}$$

21.3 $\mathbb{P}$ and $\mathcal{N}$ are disjoint

$$\mathbb{P} \cap \mathcal{N} = \varnothing$$

21.3.1 Proof

Let D $\in \mathcal{N}$

$$\nexists K, C, \lambda_1 ... \lambda_K \ :$$
$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

Assume D $\in \mathbb{P}$

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

Contradicts the definition of $\mathcal{N}$

$$\therefore D \in \mathcal{N} \Rightarrow D \notin \mathbb{P}$$

Let D $\in \mathbb{P}$

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

Assume D $\in \mathcal{N}$

$$\nexists K, C, \lambda_1 ... \lambda_K \ :$$
$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

Contradicts the definition of $\mathbb{P}$

$$\therefore D \in \mathbb{P} \Rightarrow D \notin \mathcal{N}$$

$$D \in \mathcal{N} \Rightarrow D \notin \mathbb{P}; D \in \mathbb{P} \Rightarrow D \notin \mathcal{N}$$
$$\therefore \mathbb{P} \cap \mathcal{N} = \varnothing$$

22 Discrete Derivative; Z Transform

22.1 Discrete Derivative

Define derivative for discrete function f[n]

$$\Delta_n^1 f[n] := f[n+1] - f[n]$$

We will use the above definition for the remainder of this document

22.2 Zero Order Derivative

$$\Delta_n^0 f[n] = f[n]$$

22.3 Kth Discrete Derivative

Define the K^{th} derivative of discrete function $f[n]$

$$\Delta_n^K f[n] := \Delta_n^{K-1} f[n+1] - \Delta_n^{K-1} f[n]$$

22.4 Kth Discrete Derivative as an Alternating Sum

$$\Delta_n^K f[n] := \Delta_n^{K-1} f[n+1] - \Delta_n^{K-1} f[n]$$

$$= (\Delta_n^{K-2} f[n+2] - \Delta_n^{K-2} f[n+1]) - (\Delta_n^{K-2} f[n+1] - \Delta_n^{K-2} f[n])$$

$$= (\Delta_n^{K-2} f[n+2] - 2\Delta_n^{K-2} f[n+1] - \Delta_n^{K-2} f[n])$$

$$= \textstyle\sum_{i=0}^{K} (-1)^j \ ({}_K C_j) \ \Delta_n^0 f[n+j]$$

$$= \textstyle\sum_{i=0}^{K} (-1)^j \ ({}_K C_j) \ f[n+j]$$

22.5 Z Transform

Define the Z Transform for discrete function f[n]

$$\mathcal{Z}(f[n]) := \textstyle\sum_{n=0}^{\infty} f[n] z^{-n}$$

22.6 Z Transform of 0 Order Derivative

$$\Delta_n^0 f[n] := f[n]$$

$$\mathcal{Z}(\Delta_n^0 f[n]) = \mathcal{Z}(f[n])$$

22.7 Z Transform of 1st Derivative

$$\Delta_n^1 f[n] := f[n+1] - f[n]$$

$$\mathcal{Z}(\Delta_n^1 f[n]) = \mathcal{Z}(f[n+1] - f[n])$$

$$= \textstyle\sum_{n=0}^{\infty} (f[n+1] - f[n]) z^{-n}$$

$$= \textstyle\sum_{n=0}^{\infty} (f[n+1] z^{-n} - f[n] z^{-n})$$

$$= \textstyle\sum_{n=0}^{\infty} f[n+1] z^{-n} - \sum_{n=0}^{\infty} f[n] z^{-n}$$

$$= \textstyle\sum_{m=0}^{\infty} f[m+1] z^{-m} - \sum_{n=0}^{\infty} f[n] z^{-n}$$

Let

$$\hat{m} = m+1; \;\; m = \hat{m} - 1$$

$$= \textstyle\sum_{m=0}^{\infty} f[\hat{m}] z^{-(\hat{m}-1)} - \mathcal{Z}(f[n])$$

$$= z^1 \textstyle\sum_{\hat{m}=1}^{\infty} f[\hat{m}] z^{-\hat{m}} - \mathcal{Z}(f[n])$$

$$= z^1 \sum_{\hat{m}=1}^{\infty} f[\hat{m}]z^{-\hat{m}} + f[0] - f[0] - \mathcal{Z}(f[n])$$

$$= z^1 \sum_{\hat{m}=0}^{\infty} f[\hat{m}]z^{-\hat{m}} - f[0] - \mathcal{Z}(f[n])$$

$$= z^1 \mathcal{Z}(f[n]) - f[0] - \mathcal{Z}(f[n])$$

$$\mathcal{Z}(\Delta_n^1 f[n]) = \mathcal{Z}(f[n])(z^1 - 1) - f[0]$$

22.8 Z Transform of Kth Derivative

$$\mathcal{Z}(f[n]) := \sum_{n=0}^{\infty} f[n]z^{-n}$$

$$\mathcal{Z}(\Delta_n^K f[n]) = \sum_{n=0}^{\infty} \Delta_n^K f[n]z^{-n}$$

$$= \sum_{n=0}^{\infty} \sum_{i=0}^{K} (-1)^j \ ({}_K C_j) \ f[n+j]z^{-n}$$

$$= \sum_{n=0}^{\infty} (f[n+K] - ({}_K C_1) f[n+K-1] + ({}_K C_2) f[n+K-2] - ... \pm f[n])z^{-n}$$

$$= \sum_{n=0}^{\infty} f[n+K]z^{-n} - ({}_K C_1) f[n+K-1]z^{-n} + ({}_K C_2) f[n+K-2]z^{-n} - ... \pm f[n]z^{-n}$$

$$= z^K \mathcal{Z}(f[n]) + \sum_{i=0}^{K-1} f[i] - ({}_K C_1) z^{K-1} \mathcal{Z}(f[n]) - \sum_{j=0}^{K-2} f[j] + ({}_K C_2) z^{K-2} \mathcal{Z}(f[n]) + \sum_{k=0}^{K-3} f[k] - ... \pm \mathcal{Z}(f[n])$$

When K is odd

$$\mathcal{Z}(\Delta_n^K f[n]) = (z-1)^K \mathcal{Z}(f[n]) + \sum_{i=0}^{\frac{n+1}{2}} f[2i] \quad K > 0$$

When K is even

$$\mathcal{Z}(\Delta_n^K f[n]) = (z-1)^K \mathcal{Z}(f[n]) + \sum_{j=0}^{\frac{n}{2}} f[2j+1] \quad K > 0$$

23 Divergent Complexity

23.1 Definition of Converges to

??Can f[n] = 0??

$$f[n] \ converges \ to \ C = convergent[f[n], C] = a_o; \ a_o \in \{\mathbb{T}, \mathbb{F}\} =$$

$$|C - f[n+1]| < |C - f[n]| \ \ \forall n$$

$$\wedge$$

$$\nexists K : |C - f[\hat{n}]| > K \quad \forall n; K > 0$$

23.1.1 Notation

C is commonly denoted by a limit

$$C = lim_{n\to\infty} f[n]$$

23.2 Definition of General Convergence

$$f[n] \text{ is } convergent = convergent[f[n]] = a_o;\ a_o \in \{\mathbb{T}, \mathbb{F}\} =$$

$$\exists C :$$
$$convergent[f[n], C] == \mathbb{T}$$

Alternatively

$$f[n] \text{ is } convergent = convergent[f[n]] = a_o \in \{\mathbb{T}, \mathbb{F}\} =$$
$$\exists C :$$
$$f[n]\ converges\ to\ C$$

23.3 Definition of Divergence

$$diverges[f[n]] = \neg converges[f[n]] = d_o;\ d_o \in \{\mathbb{T}, \mathbb{F}\}$$

$$:= \nexists C : convergent[f[n], C] == \mathbb{T}$$

23.4 Alternate Definition of Divergence

$$diverges[f[n]] = \neg converges[f[n]] = d_o;\ d_o \in \{\mathbb{T}, \mathbb{F}\}$$

$$= convergent[f[n], C] == \mathbb{F} \quad \forall C$$

23.4.1 Proof of Equivalence; Alternate Definition of Divergence

23.5 ?Necessary or Sufficient? Criteria 1 For Divergence

? The derivative as a function of K ? Function f[n] diverges if the K^{th} derivative of f[n] is strictly increasing

$$diverges[f[n]] := \nexists C : convergent[f[n], C] == \mathbb{T}$$

$$\iff$$

$$\Delta_n^{K+1} f[n] > \Delta_n^K f[n] \quad \forall K$$

Alternatively

$$\Delta_n^{K+1} f[n] - \Delta_n^K f[n] > 0 \quad \forall K$$

$$\Delta_n^{K+2} > 0 \ \forall K$$

23.5.1 Criteria 1; Proof of Necessity and Sufficiency

$$diverges[f[n]] = d_o; \ d_o \in \{\mathbb{T}, \mathbb{F}\}$$

$$= \nexists C : convergent[f[n], C] == \mathbb{T}$$

Let

$$f[n] :$$
$$\Delta_n^{K+2} > 0 \quad \forall K$$

23.6 ?Necessary or Sufficient? Criteria 2 For Divergence

Function f[n] diverges if the Derivative as a function of K does not Converge

$$f[n] \ is \ Divergent = Divergent[f[n]] = a_o \in \{\mathbb{T}, \mathbb{F}\} :=$$

$$\nexists c : lim_{n \to \infty} \Delta_n^K f[n] = c$$

23.6.1 Criteria 2; Proof of Necessity and Sufficiency

$$diverges[f[n]] = d_o; \ d_o \in \{\mathbb{T}, \mathbb{F}\}$$

$$= \nexists C : convergent[f[n], C] == \mathbb{T}$$

23.7 Verbal Expressions

$$f[n] \ diverges = f[n] \ is \ divergent =$$
$$f[n] \ is \ not \ convergent = f[n] \ does \ not \ converge$$

23.8 Definition of Divergent Function

23.8.1 Definition 1

Define Divergent Function f[n] having strictly increasing K^{th} derivative

$$f[n] \; is \; Divergent = Divergent[f[n]] = a_o \in \{\mathbb{T}, \mathbb{F}\} :=$$

$$\Delta_n^{K+1} f[n] > \Delta_n^K f[n] \quad \forall K$$

Alternatively

$$\Delta_n^{K+1} f[n] - \Delta_n^K f[n] > 0 \quad \forall K$$

$$\Delta_n^{K+2} > 0 \; \forall K$$

23.8.2 Definition 2

$$f[n] \; is \; Divergent = Divergent[f[n]] = a_o \in \{\mathbb{T}, \mathbb{F}\} :=$$

$$\nexists c : lim_{n \to \infty} \Delta_n^K f[n] = c$$

23.8.3 Proof of Equivalence Definition 1 $\Leftrightarrow$ Definition 2

23.8.4 Sufficient Proof

$$f[n] \; is \; Divergent \Longleftarrow$$

$$\nexists c : lim_{n \to \infty} \Delta_n^K f[n] = c$$

23.8.5 Necessary Proof

$$f[n] \; is \; Divergent \Longrightarrow$$

$$\nexists c : lim_{n \to \infty} \Delta_n^K f[n] = c$$

Proof by contradiction of definition of limit Using the definition of increasing convergence for a discrete function*

23.9 Defintion

Decision problem $\hat{D}$ with solution s^+ has divergent total complexity $O[n]$ if

$$lim_{n \to \infty} \frac{O[n+1]}{O[n]} \;\; diverges$$

23.10 Divergent Problems

$$\hat{\mathcal{D}} := \{\hat{D}_1, \hat{D}_2, ...\} :$$

$$lim_{n\to\infty}\frac{O[n+1]}{O[n]} \ \ diverges \ \ \forall s^+ \in S_i^+, \ \hat{D}_i \in \hat{\mathcal{D}}$$

23.11 The Set of Polynomial Solutions and the Set of Divergent Solutions are disjoint

$$\mathbb{P} \cap \hat{D} = \varnothing$$

23.12 Proof

Let D $\in \hat{\mathcal{D}}$

$$lim_{n\to\infty}\frac{O[n+1]}{O[n]} \ \ diverges \text{ by definition}$$

Assume D $\in \mathbb{P}$

$$lim_{n\to\infty}\frac{O[n+1]}{O[n]} = 1$$

$$lim_{n\to\infty}\frac{O[n+1]}{O[n]} = 1 \text{ contradicts the definition of Divergent Problems}$$

$$\therefore D \in \hat{\mathcal{D}} \Rightarrow D \notin \mathbb{P}$$

Let D $\in \mathbb{P}$

$$lim_{n\to\infty}\frac{O[n+1]}{O[n]} = 1 \text{ by property of Polynomial complexity}$$

Assume D $\in \hat{D}$

$$lim_{n\to\infty}\frac{O[n+1]}{O[n]} \text{ diverges}$$

$$lim_{n\to\infty}\frac{O[n+1]}{O[n]} \text{ diverges contradicts a property of Polynomial complexity}$$

$$\therefore D \in \mathbb{P} \Rightarrow D \notin \hat{\mathcal{D}}$$

$$\therefore \mathbb{P} \cap \hat{\mathcal{D}} = \varnothing$$

24 Subprograms

24.1 Definition of Subprogram

Define a Subprogram of Program P; a subset of Program P

$$P := \{s_1, s_2, ..., s_N, b_1, b_2, ..., b_M, y_o\} = \{\mathcal{L}, \mathcal{M}, y_o\}$$
$$P_{sub} := \tilde{P} \mid$$
$$\tilde{P} \subseteq P$$

24.2 Identity Subprogram

24.2.1 Definition

24.2.2 Prove the Identity Subprogram is a Subprogram of P

24.3 Restate the subprogram condition of general solutions

Recall the definition of general solution s^+

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$
$$s^+ = s^+[X_i] := P :$$
$$(P[X_i] \rightarrow y_o == a_o \;\; \forall X_i) \;\; \cap \;\; (P[\hat{X}_i] \supseteq P[X_i] \;\; \forall X_i, \hat{X}_i)$$

The subprogram condition is one of two conditions for a general solution

$$P[\hat{X}_i] \supseteq P[X_i] \;\; \forall X_i, \hat{X}_i$$

The term subprogram is used interchangeably with the term subfunction

24.4 Prove O[n] is a non-decreasing function

Consider solution s^+ with complexity $O[n]$

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$
$$s^+ = s^+[X_i] := P :$$
$$(P[X_i] \rightarrow y_o == a_o \;\; \forall X_i) \;\; \cap \;\; (P[\hat{X}_i] \supseteq P[X_i] \;\; \forall X_i, \hat{X}_i)$$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$= \{\mathcal{L}, \mathcal{M}, y_o\}$$

$$O[n] := O_T[n] + O_S[n]$$
$$O_T[n] := |\mathcal{L}| = N$$
$$O_S[n] := |\mathcal{M}| + |y_o| = M + 1$$

O[n+1] denotes the total complexity for solution $s^+[\hat{X}_i]$

$$s^+[\hat{X}_i] = \hat{s}^+$$

Let

$$O[n+1] < O[n]$$
$$\Rightarrow \hat{N} + \hat{M} < N + M$$
$$\hat{s}^+ = \{s_1, s_2, ..., s_{\hat{N}} | b_1, b_2, ..., b_{\hat{M}}, y_o\}$$

$$\Rightarrow \hat{s}^+ \nsupseteq s^+$$
$$P[\hat{X}_i] \nsupseteq P[X_i] \quad \forall X_i, \hat{X}_i$$

$$\therefore O[n+1] < O[n] \text{ contradicts the definition of solution } s^+$$
$$O[n+1] \geqslant O[n]$$

24.5 Definition of Subfunction

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$
$$s^+ = s^+[X_i] := P :$$
$$(P[X_i] \rightarrow y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$= \{\mathcal{L}, \mathcal{M}, y_o\}$$

$$Sub[X_i] := S = \{s_j, ... | b_k, ..., y_o\} :$$
$$s_j, b_k \in s^+ \quad \forall s_j, b_k \in S$$

24.5.1 $s^+[X_i]$ is a subfunction of $s^+[\hat{X}_i]$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$\hat{s}^+ = \{s_1, s_2, ..., s_N, ..., s_{\hat{N}} | b_1, b_2, ..., b_M, ..., b_{\hat{M}}, y_o\}; \;\; \hat{N} + \hat{M} \geqslant N + M$$

By definition of solution

$$\hat{s}^+ = P[\hat{X}_i] \supseteq P[X_i] = s^+ \quad \forall X_i, \hat{X}_i$$
$$\Rightarrow s_j, b_k \in \hat{s}^+ \;\; \forall s_j, b_k \in s^+$$

24.6 Subfunction Decomposition of Solutions

FIX Double check conditions!!! Solutions s^+ can be written as the union of subfunctions $Sub_k[X_i]$

$$X_i = \{x_1, ..., x_n, C\}; \;\; \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$
$$s^+ = s^+[X_i] := P :$$
$$(P[X_i] \rightarrow y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$= \{\mathcal{L}, \mathcal{M}, y_o\}$$

$$s^+ = Sub_1[X_i] \cup Sub_2[X_i] \cup ... \cup Sub_z[X_i]$$
$$= \{\mathcal{L}_1 | \mathcal{M}_1, y_o\} \cup \{\mathcal{L}_2 | \mathcal{M}_2, y_o\} \cup ... \cup \{\mathcal{L}_z | \mathcal{M}_z, y_o\} :$$
$$\mathcal{L}_j \cap \mathcal{L}_k = \varnothing \;\; \forall j, k \neq j$$

$$s^+ = \{s_1^1, ..., s_{N_1}^1 | b_1^1, ..., y_o\} \cup \{s_1^2, ..., s_{N_2}^2 | b_1^2, ..., y_o\} \cup ... \cup \{s_1^z, ..., s_{N_z}^z | b_1^z, ..., y_o\} :$$
$$\textstyle\sum_{l=1}^{z} N_l = N = O_T[n]$$

25 Subfunction Complexity

25.1 Disjoint Subfunction Operations

$$\mathcal{L}_i \cap \mathcal{L}_j = \varnothing \;\; \forall i, j \neq i$$

25.2 Shared Subfunction Memory

$$|\mathcal{M}_i \cap \mathcal{M}_j| \geqslant 0 \;\; \forall i, j \neq i$$

25.2.1 Time Complexity of Subfunctions

Subfunction time complexity is additive

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

$$Sub_i[X] := S_i = \{s_j, ...|b_k, ..., y_o\} :$$
$$s_j, b_k \in s^+ \quad \forall s_j, b_k \in S_i$$

$$s^+ = \{\mathcal{L}_1|\mathcal{M}_1, y_o\} \cup \{\mathcal{L}_2|\mathcal{M}_2, y_o\} \cup ... \cup \{\mathcal{L}_z|\mathcal{M}_z, y_o\} :$$
$$\mathcal{L}_i \cap \mathcal{L}_j = \varnothing \;\; \forall i, j \neq i$$

$$\mathcal{L} = \cup_{i=1}^{z} \mathcal{L}_i$$
$$\mathcal{L}_i \cap \mathcal{L}_j = \varnothing \;\; \forall i, j \neq i$$

$$O_T[n] = |\mathcal{L}| = N$$
$$O_T[n] = | \cup_{i=1}^{z} \mathcal{L}_i| = \sum_{i=1}^{z} |\mathcal{L}_i|^* = |\mathcal{L}_1| + |\mathcal{L}_2| + ... + |\mathcal{L}_z|$$
$$= O_{T_1}[n] + O_{T_2}[n] + ... + O_{T_z}[n] = N_1 + N_2 + ... + N_z$$

*Due to the disjoint condition of subfunction operations $\mathcal{L}_i \cap \mathcal{L}_j = \varnothing \;\; \forall i, j \neq i$

25.2.2 Space Complexity of Subfunctions

Subfunctions can access the full memory $\mathcal{M}$ with no added space complexity

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

$$Sub_i[X] := S_i = \{s_j, ...|b_k, ..., y_o\} :$$
$$s_j, b_k \in s^+ \quad \forall s_j, b_k \in S_i$$

$$s^+ = \{\mathcal{L}_1|\mathcal{M}_1, y_o\} \cup \{\mathcal{L}_2|\mathcal{M}_2, y_o\} \cup ... \cup \{\mathcal{L}_z|\mathcal{M}_z, y_o\} :$$
$$\mathcal{L}_i \cap \mathcal{L}_j = \varnothing \;\; \forall i, j \neq i$$

$$s^+ = \{\mathcal{L}_1|\mathcal{M}, y_o\} \cup \{\mathcal{L}_2|\mathcal{M}, y_o\} \cup ... \cup \{\mathcal{L}_z|\mathcal{M}, y_o\} :$$
$$\mathcal{L}_i \cap \mathcal{L}_j = \varnothing \;\; \forall i, j \neq i$$

$$\mathcal{M} = \cup_{i=1}^{z} \mathcal{M}_i = \cup_{i=1}^{z} \mathcal{M}$$

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

$$O_S[n] = | \cup_{i=1}^{z} \mathcal{M}_i| = M$$

25.2.3 Shared State Notation

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

$$Sub_i[X] := S_i = \{s_j, ...|b_k, ..., y_o\} :$$
$$s_j, b_k \in s^+ \quad \forall s_j, b_k \in S_i$$

$$s^+ = \{\mathcal{L}_1|\mathcal{M}, y_o\} \cup \{\mathcal{L}_2|\mathcal{M}, y_o\} \cup ... \cup \{\mathcal{L}_z|\mathcal{M}, y_o\} :$$
$$\mathcal{L}_i \cap \mathcal{L}_j = \varnothing \ \ \forall i, j \neq i$$

26 Polynomial Solution Subfunction Properties

26.1 Restate Definition of Subfunction

$$X_n = \{x_1, ..., x_n, C\};\ \ \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$

$$s^+ = s^+[X_n] := P :$$
$$(P[X_n] \rightarrow y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_n] \supseteq P[X_n] \quad \forall X_n, \hat{X}_n)$$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$= \{\mathcal{L}, \mathcal{M}, y_o\}$$

$$Sub[X_n] := S = \{s_j, ... | b_k, ..., y_o\} :$$
$$s_j, b_k \in s^+ \quad \forall s_j, b_k \in S$$

26.2 Property of Polynomial Solution Subfunctions

Let

$$D \in \mathbb{P}$$

$$X_n = \{x_1, ..., x_n, C\};\ \ \hat{X}_n = \{x_1, ..., x_{n+1}, C\}$$

$$s^+ = s^+[X_n] := P :$$
$$(P[X_i] \rightarrow y_o == a_o \quad \forall X_n) \quad \cap \quad (P[\hat{X}_n] \supseteq P[X_n] \quad \forall X_n, \hat{X}_n)$$

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

$$s^+ = Sub_1[X_n] \cup Sub_2[X_n] \cup ... \cup Sub_z[X_n]$$

$$limit_{n\rightarrow\infty} \frac{O[n+1]}{O[n]} = 1$$
$$= limit_{n\rightarrow\infty} \frac{O_T^1[n+1] + O_T^2[n+1] + ... + O_T^z[n+1] + O_S[n+1]}{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n]}$$
$$=$$
$$limit_{n\rightarrow\infty} \frac{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n] + f_{T_{n+1}}^1[n] + f_{T_{n+1}}^2[n+1] + ... + f_{T_{n+1}}^z[n] + f_{S_{n+1}}[n]}{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n]}$$

$$= limit_{n\to\infty} 1 + \frac{f^1_{T_{n+1}}[n]+f^2_{T_{n+1}}[n+1]+...+f^z_{T_{n+1}}[n]+f_{S_{n+1}}[n]}{O^1_T[n]+O^2_T[n]+...+O^z_T[n]+O_S[n]} = 1$$

$$\Rightarrow limit_{n\to\infty} \frac{f^1_{T_{n+1}}[n]+f^2_{T_{n+1}}[n+1]+...+f^z_{T_{n+1}}[n]+f_{S_{n+1}}[n]}{O^1_T[n]+O^2_T[n]+...+O^z_T[n]+O_S[n]} = 0^*$$

$$\Rightarrow limit_{n\to\infty} \frac{f^i_{T_{n+1}}[n]+f_{S_{n+1}}[n]}{O^1_T[n]+O^2_T[n]+...+O^z_T[n]+O_S[n]} = 0 \ \ \forall i$$

$$limit_{n\to\infty} \frac{f^i_{n+1}[n]}{O[n]} = 0 \ \ \forall i$$

* O[n] is a positive, non-decreasing function

26.3 Theorem of Polynomial Subfunctions

The Theorem of Polynomial Subfunctions states a solution has polynomial complexity if and only if all of its subfunctions have polynomial complexity

$$|s^+[X_n]| = O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

$$s^+ = Sub_1[X_n] \cup Sub_2[X_n] \cup ... \cup Sub_z[X_n]$$

$$O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

$$\Longleftrightarrow$$

$$|Sub_i[X_n]| = O_i[n] = (\hat{\lambda}_M n)^M + (\hat{\lambda}_{M-1} n)^{M-1} + ... + \hat{\lambda}_1 n + C \quad \forall i, n$$

26.3.1 Sufficient Proof

Solution s^+ having polynomial complexity implies all of its subfunctions Sub_i have polynomial complexity

Let

$$|s^+[X_n]| = O[n] = (\lambda_K n)^K + (\lambda_{K-1} n)^{K-1} ... + \lambda_1 n + C \quad \forall n$$

$$O[n] = \sum_{i=1}^{z} O_i[n] = O_1[n] + O_2[n] + ... + O_z[n]$$

Since $O[n], O_i[n]$ is positive, non-decreasing

$$O_i[n] = (\hat{\lambda}_{M_i} n)^{M_i} + (\hat{\lambda}_{M_i - 1} n)^{M_i - 1} ... + \hat{\lambda}_1 n + C \quad M_i \leqslant K \ \ \forall i, n$$

$\Rightarrow Sub_i$ has polynomial complexity by definition of polynomial complexity

26.3.2 Necessary Proof

Every subfunction Sub_i having polynomial complexity implies solution s^+ has polynomial complexity

Let

$$O_i[n] = (\hat{\lambda}_{M_i} n)^{M_i} + (\hat{\lambda}_{M_i-1} n)^{M_i-1} + ... + \hat{\lambda}_{1_i} n + \hat{\lambda}_{0_i} \quad \forall i, n$$

$$O_{max}[n]^* := \tilde{O}[n] \in \{O_1[n], O_2[n], ...O_z[n]\} :$$

$$lim_{n\to\infty} \frac{\tilde{O}[n]}{\sum_{i=1}^{z} O_i[n]} = c \neq 0$$

$$O_{max}[n] = (\hat{\lambda}_{M_{max}} n)^{M_{max}} + (\hat{\lambda}_{M_{max}1} n)^{M_{max}-1} + ... + \hat{\lambda}_{1_{max}} n + \hat{\lambda}_{0_{max}} \quad \forall i, n$$

$$O[n] = \sum_{i=1}^{z} O_i[n] = O_1[n] + O_2[n] + ... + O_z[n]$$

$$= (\tilde{\lambda}_L n)^L + (\tilde{\lambda}_{L-1} n)^{L-1} + ... + \tilde{\lambda}_1 n + C \quad L = M_{max} \;\; \forall n$$

$$\Rightarrow s^+ \text{ has polynomial complexity by definition of polynomial complexity}$$

* O_{max} is not necessarily unique, but necessarily exists. See appendix for proof

27 Divergent Solution Subfunction Properties

27.1 Restate Definition of Subfunction

$$X_i = \{x_1, ..., x_n, C\};\ \ \hat{X}_i = \{x_1, ..., x_{n+1}, C\}$$
$$s^+ = s^+[X_i] := P :$$
$$(P[X_i] \rightarrow y_o == a_o \quad \forall X_i) \quad \cap \quad (P[\hat{X}_i] \supseteq P[X_i] \quad \forall X_i, \hat{X}_i)$$

$$s^+ = \{s_1, s_2, ..., s_N | b_1, b_2, ..., b_M, y_o\} = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\}$$
$$= \{\mathcal{L}, \mathcal{M}, y_o\}$$

$$Sub[X_i] := S = \{s_j, ... | b_k, ..., y_o\} :$$
$$s_j, b_k \in s^+ \quad \forall s_j, b_k \in S$$

27.2 Property of Divergent Subfunctions

Let

$$D \in \hat{\mathcal{D}}$$

$$X_n = \{x_1, ..., x_n\};\ \ \hat{X}_n = \{x_1, ..., x_{n+1}\}$$

$$s^+ = s^+[X_n] := P :$$
$$(P[X_i] \rightarrow y_o == a_o \quad \forall X_n) \quad \cap \quad (P[\hat{X}_n] \supseteq P[X_n] \quad \forall \hat{X}_n : \hat{X}_n \supset X_n)$$

By Definition of Divergent Problem

$$\nexists c : limit_{n \to \infty} \frac{O[n+1]}{O[n]} = c$$
$$= limit_{n \to \infty} \frac{O_T^1[n+1] + O_T^2[n+1] + ... + O_T^z[n+1] + O_S[n+1]}{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n]}$$
$$=$$
$$limit_{n \to \infty} \frac{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n] + f_{T_{n+1}}^1[n] + f_{T_{n+1}}^2[n+1] + ... + f_{T_{n+1}}^z[n] + f_{S_{n+1}}[n]}{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n]}$$

$$= limit_{n \to \infty} 1 + \frac{f_{T_{n+1}}^1[n] + f_{T_{n+1}}^2[n+1] + ... + f_{T_{n+1}}^z[n] + f_{S_{n+1}}[n]}{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n]} \neq c$$

$$\Rightarrow limit_{n \to \infty} \frac{f_{T_{n+1}}^1[n] + f_{T_{n+1}}^2[n+1] + ... + f_{T_{n+1}}^z[n] + f_{S_{n+1}}[n]}{O_T^1[n] + O_T^2[n] + ... + O_T^z[n] + O_S[n]} \neq c^*$$

Prove

$$\exists i : limit_{n \to \infty} \frac{f^i_{T_{n+1}}[n] + f_{S_{n+1}}[n]}{O^1_T[n] + O^2_T[n] + ... + O^z_T[n] + O_S[n]} \;\; diverges$$

* O[n] is a positive, non-decreasing function

27.3 Theorem of Divergent Subfunctions

The Theorem of Divergent Subfunctions states a divergent subfunction implies divergent total complexity

$$lim_{n \to \infty} \frac{O[n+1]}{O[n]} \;\; diverges$$
$$\Longleftrightarrow$$
$$\exists i : lim_{n \to \infty} \frac{O_i[n+1]}{O[n]} \;\; diverges$$

27.3.1 Sufficient Direction

See 18.2

27.3.2 Necessary Direction

28 Computational Basis

28.1 Definition of a Computational Basis of Program P

Define a Computational Basis B of Program P

$$X_n = \{x_1, x_2, ..., x_n\}$$
$$P[X_n] \to Y_o := \{s_1, s_2, ..., s_N, b_1, b_2, ..., Y_o\} \to Y_o$$

$$B :=$$

For the remainder of this document, "computational basis" is denoted as "basis"

28.2 Definition of the Identity Basis of Program P

28.3 Prove the Identity Basis of Program P is a basis of Program P

28.4 Definition of Canonical Program

28.5 Definition of a Canonical Basis of Program P

28.6 Prove Canonical Basis $\mathbb{B}$ of Program P is a basis of Program P

28.7 Subprogram and Canonical Basis

Prove a subprogram is a canonical basis if and only if it's basis decomposition is the identity subprogram

28.8 Basis of Boolean Program P

29 Fundamental Theorem of Computation

The Fundamental Theorem of Computation states every program P has a canonical basis $\mathbb{B}$

29.1 Proof

30 Input Spaces

30.1 Definition of Input Space

Define the Input Space $\mathbb{I}$ of Program P

30.2 Define the Cardinality Function $C[n]$ of Input Space $\mathbb{I}$

30.3 Existence, Uniqueness, etc.

30.4 Worst Case

30.5 Prove $|\mathbb{B}| = C[n]$

31 Theorem of Solution Complexity

The Theorem of Solution Complexity relates the complexity of solution s^+ to a basis B of solution s^+

$$X_n = \{x_1, x_2, ..., x_n\}$$

$$\mathcal{Q} := f[X_n] \to A_o \subseteq \Omega \quad \forall X_n \in D_{\mathcal{Q}}$$

$$s^+ = s^+[X_n] := P[X_n] \to Y_o :$$

$$(Y_o = A_o \quad \forall X_n \in D_{\mathcal{Q}}) \quad \cap \quad (P[X_{n+1}] \supseteq P[X_n] \quad \forall X_n \in D_{\mathcal{Q}} \quad \forall X_{n+1} \in D_{\mathcal{Q}})$$

32 Parallel Programs

33 Theorem of Divergent Complexity

34 Sum to N Problem with 2 integers

34.1 State formal definition of Sum to N : $x_i + x_j == N$

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

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

$$s^+[X_n] = P[X_n] :$$
$$(P[X_i] = y_o == a_o \quad \forall X_i) \quad \cap \quad (P[X_{n+1}] \supseteq P[X_n] \quad \forall X_{n+1})$$

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

$$D = f[X_i] = \quad \exists x_j, x_k \in X_n \quad j \neq k :$$
$$x_j + x_k == N$$

34.2 Express a formal solution : $O_S[n] \sim n^0$

$$s^+ = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\} = \{\mathcal{L}, \mathcal{M}, y_o\}$$
$$s_1 = y_o \leftarrow \mathbb{F};$$
$$\forall i < n \quad , \quad n \geqslant j > i$$
$$s_2, s_3, s_8, s_9, ..., s_{3ij-4}, s_{3ij-3} ..., s_{3n(n-1)-4}, s_{3n(n-1)-3} = b_1 \leftarrow x_i + x_j$$
$$s_4, s_5, s_{10}, s_{11}, ..., s_{3ij-2}, s_{3ij-1} ..., s_{3n(n-1)-2}, s_{3n(n-1)-1} = b_1 \leftarrow b_1 == N$$
$$s_6, s_7, s_{12}, s_{13} ..., s_{3ij}, s_{3ij+1} ..., s_{3n(n-1)}, s_{3n(n-1)+1} = y_o \leftarrow y_o \vee b_1$$
$$s^+ = \{y_o \leftarrow \mathbb{F}, y_o \leftarrow y_o \vee (x_i + x_j == N) \quad \forall i, j > i \mid b_1, y_o\}$$

34.3 Prove s^+ satisfies the subfunction condition of solutions: $P[X_{n+1}] \supseteq P[X_n] \quad \forall X_{n+1}$

$$X_n = \{x_1, x_2, ..., x_n\}; \quad X_{n+1} = \{x_1, x_2, ..., x_n, x_{n+1}\}$$
$$s^+ = \{s_1, s_2, ..., s_{O_T[n]}, b_1, b_2, ..., b_{O_S[n]}, y_o\} = \{\mathcal{L}, \mathcal{M}, y_o\}$$
$$s^+_{n+1} = s^+ \cup \hat{s}^+$$
$$s_1 = y_o \leftarrow \mathbb{F};$$
$$\forall i < n \quad , \quad n \geqslant j > i$$

$$s_2, s_3, s_8, s_9, ..., s_{3ij-4}, s_{3ij-3}..., s_{3n(n-1)-4}, s_{3n(n-1)-3} = b_1 \leftarrow x_i + x_j$$
$$s_4, s_5, s_{10}, s_{11}, ..., s_{3ij-2}, s_{3ij-1}..., s_{3n(n-1)-2}, s_{3n(n-1)-1} = b_1 \leftarrow b_1 == N$$
$$s_6, s_7, s_{12}, s_{13}..., s_{3ij}, s_{3ij+1}..., s_{3n(n-1)}, s_{3n(n-1)+1} = y_o \leftarrow y_o \vee b_1$$

$$\forall k < n+1$$
$$s... = b_1 \leftarrow x_k + x_{n+1}$$
$$s... = b_1 \leftarrow b_1 == N$$
$$s... = y_o \leftarrow y_o \vee b_1$$

$$s^+ = \{y_o \leftarrow \mathbb{F}, y_o \leftarrow y_o \vee (x_i + x_j == N) \quad \forall i, j > i \mid b_1, y_o\}$$

$$\hat{s}^+ = \{y_o \leftarrow y_o \vee (x_k + x_{n+1} == N) \;\; \forall k < n+1 | \; b_1, y_o\}$$

$$s^+_{n+1} = \{y_o \leftarrow \mathbb{F}, y_o \leftarrow y_o \vee (x_i + x_j == N) \quad \forall i, j > i \mid b_1, y_o\} \;\; \cup$$
$$\{y_o \leftarrow y_o \vee (x_k + x_{n+1} == N) \;\; \forall k < n+1 | \; b_1, y_o\}$$

$$s^+_{n+1} = s^+ \cup \hat{s}^+ = P[X_{n+1}] \supseteq P[X_n] = s^+$$

34.4 Determine $O[n], O_S[n], O_T[n], f_{n+1}[n], f^T_{n+1}[n], f^S_{n+1}[n]$ for the above solution

$$O_S[n] = |y_o| + |b_1| = 2$$
$$O_T[n] = 3n(n-1) + 1 = 3n(n-1) - 1 + O_S[n]$$
$$O[n] = 3n(n-1) + 3 = 3n^2 - 3n + 3$$

$$f^S_{n+1}[n] = 0$$
$$f^T_{n+1}[n] = 6n$$
$$f_{n+1}[n] = f^S_{n+1}[n] + f^T_{n+1}[n]$$

34.5 Verify $O[n+1] = O[n] + f_{n+1}[n]$

$$O[n+1] = O[n] + \hat{O}[n]$$
$$3(n+1)^2 - 3(n+1) + 3 = 3n^2 - 3n + 3 + 6n$$
$$3n^2 + 6n + 3 - 3n - 3 + 3 = 3n^2 + 3n + 3$$

$$3n^2 + 3n + 3 = 3n^2 + 3n + 3$$

34.6 Show s^+ has Polynomial Complexity by the definition of Total Polynomial Complexity

$$O[n] = 3n^2 - 3n + 3$$

34.7 Show the limit ${}_{n\to\infty}\frac{O[n+1]}{O[n]}$ does not Diverge

$$limit_{n\to\infty}\frac{O[n+1]}{O[n]} =$$

$$limit_{n\to\infty}\frac{3n^2+3n+3}{3n^2-3n+3} =$$

$$limit_{n\to\infty}(\frac{3n^2-3n+3}{3n^2-3n+3} + \frac{6n}{3n^2-3n+3}) =$$

$$limit_{n\to\infty}(1 + \frac{6n}{3n^2-3n+3}) = 1$$

35 The Knapsack Problem

35.1 The Knapsack Problem

The Knapsack Problem is a famous problem in computer science which asks if objects can be stored in a knapsack. Typically the problem is designed with two constraints, weight and value. Given objects x_i, each with a respective weight w_i and value v_i, does there exist a combination of objects lighter than input weight W and more valuable than input value V?

35.2 Formal Definition

$$X_n = \{x_1, x_2, ..., x_n\} = \{\{w_1, v_1\}, \{w_2, v_2\}, ..., \{w_n, v_n\}\}$$

$$I = \{i_1, i_2, ..., i_n\} : i_l \in \{0, 1\} \;\; \forall i_l \in I$$

$$D := f[X_n, W, V] = a_o \in \{\mathbb{T}, \mathbb{F}\} = \exists I :$$
$$(\textstyle\sum_{j=1}^{n} i_j w_j < W) \wedge (\textstyle\sum_{j=1}^{n} i_j v_j \geqslant V)$$

35.3 Solve for C[n]

35.3.1 Expressing I as a binary number

$$I = \{i_1, i_2, ..., i_n\} : i_l \in \{0, 1\} \;\; \forall i_l \in I$$

Valid combinations of I

$$I_{valid} =$$
$$\{\{0,0,0,...,0,0,1\}, \{0,0,0,...,0,1,0\}, \{0,0,0,...,0,1,1\}, ..., \{1,1,1,...,1,1,1\}\}$$

$$C[n] = |I_{valid}[n]| = 2^n - 1$$

35.3.2 Using a sum of combinations of inputs x_i

$$X_n = \{x_1, x_2, ..., x_n\} = \{\{w_1, v_1\}, \{w_2, v_2\}, ..., \{w_n, v_n\}\}$$

Valid combinations of x_i

$$X_{valid}[n] =$$
$$\{x_1\}\cup\{x_2\}\cup...\cup\{x_n\}\cup\{x_1,x_2\}\cup\{x_1,x_3\}\cup...\cup\{x_{n-1},x_n\}\cup...\cup\{x_1,x_2,...,x_n\}$$
$$= \ {}_{X_n}C_1 \cup \ {}_{X_n}C_2 \cup ... \cup \ {}_{X_n}C_n$$

$$C[n] = |X_{valid}[n]| = \textstyle\sum_{j=1}^{n} \ {}_nC_j$$

35.3.3 Verify consistency

$$C[n] = |X_{valid}[n]| = |I_{valid}[n]|$$
$$= 2^n - 1 = \textstyle\sum_{j=1}^{n} \ {}_nC_j = {}_nC_1 + {}_nC_2 + ... + {}_nC_n$$
$$= 2^n - 1 = 2^n - 1$$

35.4 Express a solution s^+ of the Knapsack Problem

35.5 Prove s^+ satisfies the subfunction condition of solutions

35.6 Determine $O[n], O_T[n], O_S[n], f_{n+1}[n]$

35.7 Show $s^+ \notin \mathbb{P}$

35.8 Express the Solution Space $\mathbb{S}$ for The Knapsack Problem

35.9 Prove a lower bound for all solutions $s^+ \in S^+ := O_{lower}[n]$

35.10 Prove D has Divergent Complexity

Appendix

Types of Methods

35.11 Void Methods

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

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

35.12 Numerical Methods

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

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

35.13 System Methods

Define a system program; a program with inputs x_i, input set C, and real, output set Y_o

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

35.14 Mathematical Methods

Define a mathematical program; a program with inputs x_i, input set C and numerical output y_o

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

35.15 While Loop

36 Criticism of Overloaded Equivalence

In computer science, it is convention to overload equivalence =

$$a_i = a_i \quad \forall a_i \in \Omega$$

Consider standard C++ syntax
int x = 3;
int y = 4;
int z = x + y;

Int x is not inherently equal to 3. Rather, we are creating an open space "x" for a value and setting the value to 3. Similarly, z is not inherently equal to the value of x + y. Rather, we are creating an open space "z" for a value and setting the value to the sum of x and y which have already been set.

$$x \leftarrow 3$$

$$y \leftarrow 4$$

$$z \leftarrow x + y$$

37 Existence of $O_{max}[n]$

37.1 Proof

37.1.1 Alternate Definition; Left Hand Derivative

Some sources define

$$\Delta_n^1 f[n] = f[n] - f[n-1]$$

Citations

[1] *https://kapeli.com/cheat_sheets/LaTeX_Math_Symbols.docset/Contents/Resources/Docume*
[2] *chat.openai.com*
[3] *google.com*
[4] *wikipedia.org*
[3] *wolframalpha.com*
[4] *youtube.com*
[5] *https://stackoverflow.com/questions/*3518973*/floating-point-exponentiation-without-power-function*
[6] *https://stackoverflow.com/questions/*27086195*/linear-index-upper-triangular-matrix*