CHAPTER VII
DATA STRUCTURES

In the study of theoretical computer science data structures form a fundamental area of research. These structures are explained through abstract algebraic structures like categories and commutative algebra in section 7.1. Fractal geometric techniques and genetic algorithms have been used in the optimization procedure. This is our concern in section 7.2. We close this chapter with the introduction to the recent advances in the pattern recognition. Here the application of Walsh transform technique and fuzzy grammar forms a strong basis.

SECTION 7.1. M-CONSTRUCTION

Algebraic theories over ranked sets

An $\omega$-ranked set is a pair $(A, r)$ consisting of a set $A$ and a "ranking" function $r : A \to \omega$. Let $\mathcal{R}$ be the category whose objects are all finite $\omega$-ranked sets.

A morphism $\phi : (A, r) \to (B, s)$ in $\mathcal{R}$ is a function $\phi : A \to B$ such that commutes.

\[ \phi \]
\[ \begin{array}{ccc}
A & \xrightarrow{r} & \omega \\
\downarrow \phi & & \downarrow s \\
B & \xrightarrow{\omega} & \omega
\end{array} \]

Definition 7.1.1

An algebraic theory over $\mathcal{R}$ is a category with the same objects as $\mathcal{R}$ which has, for each object $(A, r)$, an $A$-indexed family of distinguished morphisms,

\[ \langle a : ([1], ar) \to (A, r) \mid a \in A \rangle, \] .... (1)

with the following property.
For any A-indexed family of morphisms, \( \{ f_a : ([1], ar) \to (B, s) | a \in A \} \), there is a unique morphism, \( f : (A, r) \to (B, s) \), such that the diagram commutes for each \( a \in A \).

\[
\begin{array}{ccc}
([1], ar) & \xrightarrow{fa} & (B, s) \\
\downarrow{f} & & \downarrow{f} \\
(A, r) & \xrightarrow{a} & (B, s)
\end{array}
\]

An algebraic theory \( T \) over \( \mathcal{R} \) is non-degenerate if whenever \( A \) have at least two elements, the distinguished morphism \( (1) \) are distinct, so that one may consider \( \mathcal{R} \) to be a subcategory of \( T \).

We will say that an algebraic theory \( T \) over \( \mathcal{R} \) is \( \omega \)-continuous if each hom-set is a strict \( \omega \)-complete poset and composition is left strict and \( \omega \)-continuous.

**Definition 7.1.2**

Let \( T \) and \( T' \) be algebraic theories over \( \mathcal{R} \). An (\( \mathcal{R} \)-theory) morphism \( h : T \to T' \) is a functor which is the identity on the common objects of \( T \) and \( T' \) and which preserves distinguished morphisms; i.e., if \( \Sigma = (A, r) \) is an object of \( T \) then

\[
ah = a
\]

for each \( a \in A \).
Continuous theories, Preliminaries

Definition 7.1.3

An algebraic theory is \(\omega\)-continuous if each hom-set is an \(\omega\)-complete poset, with least element \(\bot\); composition is left-strict and \(\omega\)-continuous by components

\[
\bot \cdot f = f,
\]

\[
(\sqcup \alpha_n) \cdot \beta = \sqcup (\alpha_n \cdot \beta),
\]

\[
\alpha \cdot (\sqcup \beta_n) = \sqcup (\alpha \cdot \beta_n).
\]

Furthermore, composition is order preserving:

\[\alpha \sqsubseteq \beta \text{ implies } \alpha \cdot \gamma \sqsubseteq \beta \cdot \gamma \text{ and } \gamma \cdot \alpha \sqsubseteq \gamma \cdot \beta\]

whenever these composites are defined.

Definition 7.1.4

If \(T_1\) and \(T_2\) are \(\omega\)-continuous algebraic theories, a theory morphism \(h : T_1 \rightarrow T_2\) is \(\omega\)-continuous if \(h\) is strict (i.e., if \(\bot\) is the least element \(u \rightarrow v\) in \(T_1\), then \(\bot h\) is the least element \(u \rightarrow v\) in \(T_2\)), order preserving (if \(\alpha \sqsubseteq \beta\) in \(T_1\), then \(\alpha h \sqsubseteq \beta h\) in \(T_2\)), and \(\omega\)-continuous (if \(\alpha = \sqcup \alpha_n\) in \(T_1\), then \(\alpha h = \sqcup (\alpha_n h)\) in \(T_2\)).

Result 7.1.1

The \(\omega\)-continuous tree theory \(CT_\Sigma\) is freely generated, in the category of \(\omega\)-continuous theories, by the ranked set \(\Sigma\); i.e., there is a rank preserving function \(\eta : \Sigma \rightarrow CT_\Sigma\) such that for any \(\omega\)-continuous theory \(T\) and any rank preserving function \(f : \Sigma \rightarrow T\) there is a unique \(\omega\)-continuous theory morphism \(f^\# : CT_\Sigma \rightarrow T\) such that

\[f \eta = f^\#\]
Furthermore, if $f_1, f_2 : \sum \to T$ are rank-preserving functions with $\sigma f_1 \subseteq \sigma f_2$, for all $\sigma \in \sum$, then $tf^\sigma_1 \subseteq tf^\sigma_2$, for all $t$ in $CT_\Sigma$.

**Result 7.1.2**

Composition of $\omega$-continuous theory morphisms is itself left-strict and $\omega$-continuous by components:

1. If $\perp : T_1 \to T_2$ is the $\omega$-continuous theory morphism whose value on each hom-set in $T_1$ is the least element in the corresponding hom-set in $T_2$, then $\perp \cdot \alpha = \perp$, for all theory morphisms $\alpha : T_2 \to T_3$.

2. If $\{ \alpha_n : T_1 \to T_2 | n \in \omega \}$ is an $\omega$-chain of $\omega$-continuous theory morphisms, and $\alpha : T_0 \to T_1, \beta : T_2 \to T_3$, then $\alpha . (\bigsqcup \alpha_n) = \bigsqcup (\alpha . \alpha_n), (\bigsqcup \alpha_n) . \beta = \bigsqcup (\alpha_n . \beta)$.

**Theorem 7.1.1**

If $T_1$ and $T_2$ are $\omega$-continuous theories, there is a $\omega$-continuous theory $T_1 + T_2$ (called the coproduct of $T_1$ and $T_2$) and $\omega$-continuous theory morphisms (the coproduct injections)

\[ k : T_1 \to T_1 + T_2 , \]
\[ \lambda : T_2 \to T_1 + T_2 , \]

with the following coproduct property: if $\alpha_i : T_i \to T$, $i = 1, 2$, are $\omega$-continuous theory morphisms, then there is a unique $\omega$-continuous theory morphism $(\alpha_1, \alpha_2) : T_1 + T_2 \to T$ such that

\[ k.(\alpha_1, \alpha_2) = \alpha_1 , \quad \text{.... (2)} \]
\[ \lambda.(\alpha_1, \alpha_2) = \alpha_2 . \quad \text{.... (3)} \]
Proof

Let $\mathcal{F}$ be the category whose objects are (one-sorted) $\omega$-continuous theories; a morphism in $\mathcal{F}$ is an $\omega$-continuous theory morphism. Let $U : \mathcal{F} \to \mathcal{R}_\infty$ be the functor which takes a theory $T$ and yields the $\omega$-ranked set $TU = (A, r)$, where $A = U(T(1, n))$ and $x r = n$ iff $x : 1 \to n$ in $T$. The value of $U$ on theory morphisms is clear.

Remark 7.1.1

We may rephrase Result 7.1.1. by saying that $U$ has a left adjoint $F$; i.e., for any ranked set $\Sigma = (A, r)$, there is an $\omega$-continuous theory $\Sigma F = CT_\Sigma$ with the usual universal property. Let $\varepsilon : UF \to \text{Id}$ be the associated natural transformation between the endofunctors $UF$ and the identity factor on $\mathcal{F}$.

We note the following fact about $\varepsilon :$ for each theory $T$ in $\mathcal{F}$

$$\varepsilon_T : TUF \to T$$

is a "surjection" (i.e., $\varepsilon U$ is surjective).

Now given theories $T_1$ and $T_2$ in $\mathcal{F}$ form the ranked sets $T_1 U$ and $T_2 U$ in $\mathcal{R}_\infty$. Since it is clear that $\mathcal{R}_\infty$ has coproducts, let $\zeta_i : T_i U \to T_1 U + T_2 U$, $i = 1, 2$, be the coproduct injections $\mathcal{R}_\infty$. Then,

$$\zeta_i F : T_i UF \to (T_1 U + T_2 U) F = T_1 UF + T_2 UF$$

are coproduct injections since $F$, being a left adjoint, must preserve colimits. Let $T = T_1 UF + T_2 UF$. 
Now let $\sim$ be the least ordered theory congruence on $T$ such that for $i = 1, 2,$ and for all $x, y$ in $T_i \cup F$,

$$\text{if } x \vDash T_i = y \vDash T_i \text{ then } x \vDash F \sim y \vDash F.$$  

Let $\sim T$ be the theory $T/\sim$ and let $\psi : T \to \sim T$ be the canonical ordered theory morphism taking an element to its equivalent class. Note $\sim T$ need not be an object in $\mathcal{S}$, in particular $\sim T$ is not necessarily complete and thus not an $\omega$-continuous theory.

Result 7.1.3

Let $T$ be a subtheory of the $\omega$-continuous theory $\sim T$. There is an $\omega$-continuous subtheory $T_1$ of $\sim T$ (i.e., a subtheory which is $\omega$-continuous and in which least upper bounds agree with least upper bounds in $\sim T$) containing $T$ such that

$$\text{card}(T_1) \leq 2^{\text{card}(T)}.$$  

(here, Card$(T)$ is the cardinality of $U_{n,p} T(n,p)$, and similarly for $T_1$).

Lemma 7.1.1

Let $\psi : T \to \sim T$ be a left-strict order preserving theory morphism in the category of ordered theories. Then there is an $\omega$-continuous theory $\sim T$ and a theory morphism $u : \sim T \to \sim T$ with the following properties:

1.1. The composite $\psi u : T \to \sim T$ is $\omega$-continuous (i.e., preserves all least upper bounds of $\omega$-chains which exist).
1.2. If $g: \bar{T} \to T'$ is any theory morphism such that $T'$ is $\omega$-continuous and $\psi$. $g$ is $\omega$-continuous, then there is a unique $\omega$-continuous theory morphism $g^\#: \bar{T} \to T'$ such that $u.g^\# = g$.

Proof

By Result 7.1.3., we may choose a representative small set $I$ of all those theory morphism $h_i : T_i \to T_j$ such that $T_i$ is $\omega$-continuous and $\psi$. $h_i$ is $\omega$-continuous. Then form, as usual, the target tupling,

$$[h_i] : \bar{T} \to \prod T_i.$$

The product $\prod T_i$ is also $\omega$-continuous, since least upper bounds are computed component-wise.

Let $\bar{T}$ be the least $\omega$-continuous subtheory of $\prod T_i$ containing the image of $[h_i]$. Then $[h_i]$ factors through $\bar{T}$ as $h = u. \tau$, where $u : T \to \bar{T}$ and where $\tau : \bar{T} \to \prod T_i$ is the inclusion. It is clear that $u$ has the required properties (1.1) and (1.2). This completes the proof of Lemma.

The situation that existed prior to the statement of Lemma 7.1.1 may be summarized by the following diagram:
We show the existence of the theory morphisms \( j_1 : T_1 \rightarrow \tilde{T} \) making (4) commute. Recall the fact that \( \varepsilon_i \) is surjective. Further, if \( x \varepsilon_i = y \varepsilon_i \) then \( x \xi F = y \xi F \), \( \psi \), by the construction of \( \psi \). Thus we may define \( j_i \) to make (4) commute. Clearly the \( j_i \) are theory morphisms.

Now we apply Lemma 7.1.1, to \( \psi : T \rightarrow \tilde{T} \) obtaining \( u : \tilde{T} \rightarrow \tilde{T} \).

**Result 7.1.4**

The composites \( k = j_1 . u \) and \( j_2 . u \) are \( \omega \)-continuous.

Indeed,

\[
\varepsilon_1 \cdot j_1 \cdot u = \varepsilon_1 F \cdot (\psi . u),
\]

and the right-hand side of this equation is a composite of \( \omega \)-continuous morphisms. Thus the composite of a surjective \( \omega \)-continuous morphism \( (\varepsilon_1) \) with \( j_1 . u \) is \( \omega \)-continuous. As is easy to check it follows that \( j_1 . u \) is \( \omega \)-continuous. The same goes for \( j_2 . u \).

We now show that \( \langle k : T_1 \rightarrow \tilde{T}, \lambda : T_2 \rightarrow \tilde{T} \rangle \) is a pair of coproduct injections in the category \( \mathcal{E} \).

Let \( \alpha_i : T_i \rightarrow Q \) be an arbitrary pair of \( \omega \)-continuous theory morphisms whose target \( Q \) is an \( \omega \)-continuous theory. Then there exists a unique \( \omega \)-continuous \( \beta : T \rightarrow Q \) such that

\[
\varepsilon_i F \beta = \varepsilon_i \cdot \alpha_i, \ i = 1, 2.
\]
But $\beta$ must be factor through $\psi$ since the theory congruence ~ on $T$ induced by $\beta$ will contain the congruence determined by $\psi$. Hence there is a (unique) $\beta^\# : \tilde{T} \rightarrow Q$ such that $\psi \cdot \beta^\# = \beta$. Therefore, by the universal property of $u$, there is a unique $\omega$-continuous theory morphism $\overline{\beta} : \overline{T} \rightarrow Q$ such that the following diagram commutes:

\[
\begin{array}{c}
\psi \\
\downarrow \sim \\
T \\
\downarrow \mathbf{j}_1 \\
\tilde{T} \\
\alpha_1 \\
\downarrow k \\
T \\
\uparrow \psi \\
\downarrow j_2 \\
Q \\
\end{array}
\]

But then

\[
k \cdot \overline{\beta} = \mathbf{j}_1 \cdot u \cdot \overline{\beta} = \alpha_1
\]

since

\[
\epsilon_1 \cdot \mathbf{j}_1 \cdot u \cdot \overline{\beta} = \mathbf{e} \cdot \mathbf{F} \cdot u \cdot \overline{\beta} = \epsilon_1 \cdot \mathbf{F} \cdot \mathbf{\beta} = \epsilon_1 \cdot \alpha_1
\]

and $\epsilon_1$ is surjective. Similarly $\lambda \cdot \overline{\beta} = \mathbf{j}_2 \cdot u \cdot \overline{\beta} = \alpha_2$, and the proof of Theorem 7.1.1 is complete.
If \( \alpha_i : T_i \rightarrow T'_i \), \( i = 1, 2 \), are \( \omega \)-continuous theory morphisms between the \( \omega \)-continuous theories, then we define \( \alpha_1 + \alpha_2 : T_1 + T_2 \rightarrow T'_1 + T'_2 \) by

\[
\alpha_1 + \alpha_2 = (\alpha_1 \cdot k, \alpha_2 \cdot \lambda)
\]  

..... (5)

**Corollary 7.1.1**

The following equations hold whenever they are meaningful, i.e., when the sources and targets of the theory morphisms match.

\[
(\alpha_1, \alpha_2) \cdot \beta = (\alpha_1 \cdot \beta, \alpha_2 \cdot \beta),
\]  

..... (6)

\[
(\alpha_1 + \alpha_2) \cdot (\beta_1, \beta_2) = (\alpha_1 \cdot \beta_1, \alpha_2 \cdot \beta_2),
\]  

..... (7)

\[
(\alpha_1 + \alpha_2) \cdot (\beta_1 + \beta_2) = (\alpha_1 \cdot \beta_1) + (\alpha_2 \cdot \beta_2)
\]  

..... (8)

\[
\alpha \cdot (k, \lambda) = \alpha,
\]  

..... (9)

\[
(k, \lambda) \cdot \alpha = \alpha.
\]  

..... (10)

**Formulation of M(T)**

**Definition 7.1.5**

Let \( T \) be a fixed \( \omega \)-continuous (one-sorted) theory. Then \( M(T) \) is a category whose objects are the objects of \( R \), i.e., the \( \omega \)-ranked sets. If \( \Sigma = (A, r) \) and \( \Pi = (B, s) \) are objects of \( M(T) \), a morphism \( \alpha : \Sigma \rightarrow \Pi \) in \( M(T) \) is a theory morphism.

\[
\alpha : CT_{\Sigma} \rightarrow T + CT_{\Pi},
\]  

where the "+" on the right-hand side is the coproduct of Theorem 7.1.1.

**Definition 7.1.6**

Composition in \( M(T) \). Let \( \alpha : \Sigma \rightarrow \Pi, \beta : \Pi \rightarrow \Delta, \) in \( M(T) \). The composite \( \alpha \circ \beta : \Sigma \rightarrow \Delta \) is the theory morphism

\[
\alpha \circ \beta = \alpha \cdot (k, \beta),
\]  

..... (11)

where \( k : T \rightarrow T + CT_{\Delta} \) is the indicated coproduct injection.
Result 7.1.5

Composition in \( M(T) \) is associative and the theory morphisms \( \lambda \) are the identity morphisms in \( M(T) \).

Proof

Suppose \( \alpha, \beta \) and \( \gamma \) are composable morphisms in \( M(T) \). Then

\[
\alpha \circ (\beta \circ \gamma) = \alpha \cdot (k, \beta \cdot (k, \gamma)) \quad \text{by (11)}
\]

\[
= \alpha \cdot (k, \beta \cdot (k, \gamma)) \quad \text{by (6), (2)}
\]

\[
= (\alpha \circ \beta) \cdot (k, \gamma) \quad \text{by (11)}
\]

\[
= (\alpha \circ \beta) \circ \gamma \quad \text{by (11)}
\]

Thus composition is associative. For the identities, if \( \lambda : \Sigma \to \Sigma \) is the coproduct injection \( \lambda : CT_{\Sigma} \to T + CT_{\Sigma} \). Then

\[
\alpha \circ \lambda = \alpha \cdot (k, \lambda) = \alpha \quad \text{by (9)}
\]

\[
\lambda \circ \beta = \lambda \cdot (k, \beta) = \alpha \quad \text{by (3)}
\]

Corollary 7.1.2

\( M(T) \) is a category, where \( \circ \) is the operation of composition and the coproduct injections \( \lambda \) are the identity morphisms.

Result 7.1.6

\( M(T) \) is an algebraic theory over \( \mathcal{T} \), where, for \( \Sigma = (A, r) \), \( a \in A \), the distinguished morphism

\[
a : ([1], ar) \to \Sigma
\]

is the theory morphism.
\[ a^\#, \lambda : CT_{([1], ar)} \rightarrow T + CT_\Sigma \]

Here \( a^\# : CT_{([1], ar)} \rightarrow CT_\Sigma \) is determined, according to Result 7.1.1 by the function \( ([1], ar) \rightarrow CT_\Sigma \) whose value is \( a_\eta \).

**Result 7.1.7**

\( M(T) \) is a \( \omega \)-continuous algebraic theory over \( \mathcal{R} \).

**Proof**

The ordering on the collection of morphisms \( \Sigma \rightarrow \Pi \) in \( M(T) \) is defined componentwise, i.e., \( \alpha \sqsubseteq \beta \) if, for each \( t \in CT_\Sigma \), \( t \alpha \sqsubseteq t \beta \) in \( T + CT_\Pi \) or, equivalently, by Result 7.1.1. \( \sigma \alpha \sqsubseteq \sigma \beta \) in \( T + CT_\Pi \), for each \( \sigma \in \Sigma \).

If \( \alpha_n : CT_\Sigma \rightarrow T + CT_\Pi \), \( n \in \omega \), is an \( \omega \)-chain of theory morphisms. \( \sqcup \alpha_n \) is the theory morphism determined by

\[ \sigma(\sqcup \alpha_n) = \sqcup (\sigma \alpha_n) \]

for each \( \sigma \in \Sigma \). The sup on the right of (13) exists in \( T + CT_\Pi \) since this theory is itself \( \omega \)-continuous. The least element \( \Sigma \rightarrow \Pi \) in \( M(T) \) is the "constant" theory morphism \( CT_\Sigma \rightarrow T + CT_\Pi \) whose value on each \( \sigma \in \Sigma \) is the appropriate \( \perp \) in \( T + CT_\Pi \).

(2.1) Composition in \( M(T) \) is left-strict. Indeed, if \( \perp : \Sigma \rightarrow \Pi \) in \( M(T) \) and \( \alpha : \Pi \rightarrow \Delta \), then \( \perp \circ \alpha = \perp \cdot (k, \alpha) = \perp \), since composition of \( \omega \)-continuous theory morphisms is left-strict.
Composition in $M(T)$ is $\omega$-continuous by components. Indeed, this follows as in 2.1 since composition of theory morphisms is $\omega$-continuous by components.

The proof of Result 7.1.7 is complete

**Definition 7.1.7**

Let $\phi = T_1 \rightarrow T_2$ be a $\omega$-continuous theory morphism between the $\omega$-continuous theories $T_i$. Then $M(\phi) : M(T_1) \rightarrow M(T_2)$ is the following $\mathcal{R}$-theory morphism: if $\alpha : \Sigma \rightarrow \Pi$ is a morphism in $M(T_1)$, then $\alpha M(\phi) : \Sigma \rightarrow \Pi$ in $M(T_2)$ is defined as the composite,

$$\alpha M(\phi) = \alpha \circ (\phi + 1),$$

as in the following diagram:

$$\begin{array}{ccc}
CT_{\Sigma} & \xrightarrow{\alpha} & T_1 + CT_{\Pi} \\
\downarrow{\alpha M(\phi)} & & \downarrow{\phi + 1} \\
T_2 + CT_{\Pi} & & \\
\end{array}$$

Here $1$ is the identity morphism $CT_{\Pi} \rightarrow CT_{\Pi}$. Being a composition of $\omega$-continuous theory morphisms, $\alpha M(\phi)$ is also a $\omega$-continuous theory morphism, and thus a morphism in $M(T_2)$.

**Result 7.1.8**

For $\alpha : \Sigma \rightarrow \Pi$, $\beta : \Pi \rightarrow \Delta$ in $M(T_1),$

$$(\alpha \circ \beta) M(\phi) = \alpha M(\phi) \circ \beta M(\phi).$$
Proof

\[(\alpha \circ \beta) \circ \phi = \alpha \cdot (k, \beta \cdot (\phi + 1)) = \alpha \cdot (k, (\phi + 1), \beta \cdot (\phi + 1)) = \alpha \cdot (\phi + 1), (k, \beta \cdot (\phi + 1)) = \alpha \cdot M(\phi) \circ \beta \cdot (\phi + 1) = \alpha \cdot M(\phi) \cdot \beta M(\phi) = \alpha \cdot M((\phi)) \cdot \beta M((\phi)) = \alpha \cdot M((\phi)) \cdot \beta M((\phi))
\]

Result 7.1.9.

If \( \Sigma = (A, r) \) is an object of \( M(T_1) \) and \( a \in A \), then \( a M((\phi)) = a \).

Proof

First \( a = a \cdot \lambda \), by (12)

\[a M(\phi) = a \cdot \lambda \cdot (\phi + 1) = a \cdot \lambda \cdot (\phi k, \lambda) = a \cdot \lambda = a \]

Result 7.1.10.

If \( 1 : \Sigma \rightarrow \Sigma \) is the identity morphism in \( M(T_1) \), \( 1 M(\phi) = 1 : \Sigma \rightarrow \Sigma \) in \( M(T_2) \).

Proof

\[1 : \Sigma \rightarrow \Sigma \text{ in } M(T_1) \text{ is } \lambda : \text{CT}_\Sigma \rightarrow T_1 + \text{CT}_\Sigma, \text{ so}
\]

\[1 M(\phi) = \lambda \cdot (\phi + 1) = \lambda \cdot (\phi k, \lambda) \]
The combination of Result 7.1.8 – 7.1.10 shows that $M(\phi)$ is an $\mathcal{B}$-theory morphism $M(T_1) \rightarrow M(T_2)$. We need some further information.

**Result 7.1.11**

$M(\phi)$ is a $\omega$-continuous $\mathcal{B}$-theory morphism.

**Proof**

This follows by Result 7.1.2 composition of $\omega$-continuous theory morphisms is left-strict and $\omega$-continuous components.

**Result 7.1.12**

Suppose $\phi : T_1 \rightarrow T_2$, $\psi : T_2 \rightarrow T_3$ are $\omega$-continuous theory morphisms. Then

$$M(\phi \cdot \psi) = M(\phi) \cdot M(\psi).$$

**Proof**

Suppose $\alpha : \Sigma \rightarrow \prod$ in $M(T_1)$. Then,

$$\begin{align*}
\alpha M(\phi) \cdot M(\psi) &= \alpha \cdot (\phi + 1) \cdot (\psi + 1) \quad \text{by (7.1.7)} \\
&= \alpha \cdot (\phi \psi + 1) \quad \text{by (8)} \\
&= \alpha M(\phi \psi) \quad \text{by (7.1.7)}
\end{align*}$$

Since it is clear that $M(\phi) = 1 : M(T) \rightarrow M(T)$ if $\phi = 1 : T \rightarrow T$, the above facts have proved the statement 'M is a functor from the category of (one – sorted) $\omega$-continuous theories and $\omega$-continuous theory morphisms to the category of $\omega$-continuous theories over $\mathcal{B}$ and $\omega$-continuous $\mathcal{B}$-theory morphisms.'
Solution of recursion equations

Suppose that $\alpha : \Sigma \to \Sigma + \Pi$ in $M(T_1)$ (where $\Sigma$ and $\Pi$ are disjoint ranked sets). A solution of $\alpha$ in $T_1$ with parameters in $\Pi$ is a morphism $\xi : \Sigma \to \Pi$ in $M(T_1)$ such that

$$\xi = \alpha . (x, 1),$$

where $1 : \Pi \to \Pi$ is the identity in $M(T_1)$. The least such solution, denoted $\alpha^+$, may be obtained just as for one-sorted theories, as the sup of the $\omega$-chain $\alpha_n : \Sigma \to \Pi$, where

$$\alpha_{n+1} = \alpha . (\alpha, \lambda)^n . (\perp, 1),$$

$$\alpha^+ = \bigcup \alpha_n.$$

Here, $\perp : \Sigma \to \emptyset$ is the least theory morphism $CT_\Sigma \to T_1$.

Illustration 7.1.1

For simplicity consider the equation

$$\begin{array}{c}
+ \\
x y \\
\hline
\end{array} = 
\begin{array}{c}
\text{IF} \\
x y \\
\hline
\text{SUCC} \\
\hline
+ \\
PRED \\
x \\
\hline
\end{array}$$

.... (14)

If $\Sigma$ consists of the function symbols, IF, SUCC and PRED and if $\Phi$ consists of just the symbol, $+$, and $\Sigma$ is uninterpreted, then $(14)$ is a morphism

$$\alpha : \Phi \to \Sigma + \Phi$$

in $M(CT_\Sigma)$. The solution $\alpha^+ : \Phi \to \Sigma$ of $\alpha$ in $M(CT_\Sigma)$ is the infinite tree.
Now let \( \phi : \mathbb{CT} \rightarrow T_\alpha \) be the “intended” interpretation of the symbols in \( \Sigma \) as functions on the natural numbers, e.g.,

\[
\phi(\text{IF}) : \mathbb{N}_1^3 \rightarrow \mathbb{N}_1
\]

was defined above, where \( \mathbb{N}_1 \) is the flat poset of the natural numbers with a least element \( \bot \) adjoined. Then the solution \( \alpha^+ \) of \( \alpha \) in \( M(T_\alpha) \) is just the usual addition \( \Theta : \mathbb{N}_1^2 \rightarrow \mathbb{N}_1 \). The infinite tree, (15) may be thought of as an algorithm for computing \( \Theta \); the algorithm is interpreted by \( M(\phi) \).

Last we amplify the remark at the end of the introduction on specifications with parameters. Suppose \( \alpha : \Phi \rightarrow \Phi \oplus \Pi \) is a morphism in \( M(T) \). Let \( \beta : \Pi \rightarrow \varnothing \) in \( M(T) \) be an “interpretation” of the parameters (so \( \beta : \mathbb{CT}_\Pi \rightarrow T \)). Then the resulting system, denoted \( \alpha_\beta \) is the composite in \( M(T) \),
\[
\alpha_\beta = \alpha (1 + \beta),
\]
where \((1 + \beta) : \Phi + \Pi \rightarrow \Phi + \emptyset \dashv \Phi\). We claim that
\[
(\alpha_\beta)^+ = \alpha^+ \cdot \beta.
\]
Indeed, this is one of the basic identities (called the substitution identity) that hold for all rational and iterative theories.

The intuitive meaning of (16) is that one may solve the system \(\alpha_\beta\), in which the parameters in \(\Pi\) have been replaced by their interpretations in \(T\), by first solving the uninterpreted system \(\alpha\) and then interpreting the parameters.

**SECTION 7.2. BASIC CONCEPT IN GENETIC ALGORITHMS**

Genetic algorithms are computerized search and optimization algorithms based on the mechanics of natural genetics and natural selection.

Let us consider the following maximization problem:

Maximize \(f(x), x_i^{(L)} \leq x_i^{(U)}\), \(i = 1, 2, ..., N\).

The working of GAs is completed by performing the following tasks.

In order to use GAs to solve the above problem, variables \(x_i\)'s are first coded in some string structures. Binary-coded strings having 1's and 0's are mostly used. The length of the string is usually determined according to the desired solution accuracy. For example, if four bits are used to code each variable in a two-variable function optimization problem, the strings \((0000 \quad 0000)\) and \((1111 \quad 1111)\) would represent the points

\[
(x_1^{(L)}, x_2^{(L)})^T \quad (x_1^{(U)}, x_2^{(U)})^T,
\]
respectively, because the substrings (0000) and (1111) have the minimum and the maximum decoded values. Any other eight-bit string can be found to represent a point in the search space according to a fixed mapping rule. Usually, the following linear mapping rule is used.

\[ x_i = x_i^{(L)} + \frac{x_i^{(U)} - x_i^{(L)}}{2^i - 1} \text{ decoded value } (s_i). \quad \cdots \text{(17)} \]

In the above equation, the variable \( x_i \) is coded in a substring \( s_i \) of length \( \ell \). The decoded value of a binary substring \( s_i \) is calculated as \( \sum_{r=0}^{\ell-1} 2^r s_i \), where \( s_i \in (0, 1) \) and the string \( s \) is represented as \( (s_\ell s_{\ell-1} \cdots s_2 s_1 s_0) \). It is worthwhile to mention here that with four bits to code each variable, there are only \( 2^4 \) or 16 distinct substrings possible, because each bit-position can take a value either 0 or 1. The accuracy that can be obtained with a four-bit coding is only approximately \( 1/16^{th} \) of the search space. But as the string length is increased by one, the obtainable accuracy increases exponentially to \( 1/32^{th} \) of the search space. It is not necessary to code all variables in equal substring length. The length of a substring representing a variable depends on the desired accuracy in that variable. Generalizing this concept, we may say that with \( \ell \) - bit coding for a variable, the obtainable accuracy is that variable is approximately \( (x_i^{(U)} - x_i^{(L)}) / 2^\ell \). Once the coding of the variables has been done, the corresponding point \( x = (x_1, x_2, \ldots, x_N)^T \) can be found using equation (17). Thereafter, the function value at the point \( x \) can also be calculated by substituting \( x \) in the given objective function \( f(x) \).
**Fitness function**

GAs mimic the survival-of-the-fittest principle of nature to make a search process. GAs are naturally suitable for solving maximization problems. Minimization problems are usually transformed into maximization problems by some suitable transformation. In general, a fitness function \( f(x) \) is first derived from the objective function and used in successive genetic operations. For maximization problems, the fitness function can be considered to be the same as the objective function or \( F(x) = f(x) \). For minimization problems, the following fitness function is often used:

\[
F(x) = \frac{1}{1+f(x)}.
\]

This transformation does not alter the location of the minimum, but converts a minimization problem to an equivalent maximization problem. The fitness function value of a string is known as the string's fitness.

The operation of GAs begins with a population of random strings representing design or decision variables. Thereafter, each string is evaluated to find the fitness value. The population is then operated by three main operators – reproduction, crossover, and mutation—to create new population of points. The new population is further evaluated and tested for termination. If the termination criterion is not met, the population is iteratively operated by the above three operators and evaluated. This procedure is continued until the termination criterion is met. One cycle of these operations and the subsequent evaluation procedure is known as a generation in GA's terminology.
GA operators

Reproduction is usually the first operator applied on a population. Reproduction selects good strings in a population and forms a mating pool. That is why the reproduction operator is sometimes known as the selection operator. There exist a number of reproduction operators in GA literature, but the essential idea in all of them is that the above average strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner. The commonly used reproduction operator is the proportionate reproduction operator where a string is selected for the mating pool with a probability proportional to its fitness. Thus, the $i^{th}$ string in the population is selected with a probability proportional to $F_i$. Since the population size is usually kept fixed in a simple GA, the sum of the probability of each string being selected for the mating pool must be one. Therefore, the probability for selecting the $i^{th}$ string is

$$p_i = \frac{F_i}{\sum_{j=1}^{n} F_j},$$

where $n$ is the population size. One way to implement this selection scheme is to imagine a roulette-wheel with it’s circumference marked for each string proportionate to the string’s fitness. The roulette-wheel is spun $n$ times, each time selecting an instance of the string chosen by the roulette-wheel pointer. Since the circumference of the wheel is marked according to a string’s fitness, this roulette-wheel mechanism is expected to make $F_i/\bar{F}$ copies of the $i^{th}$ string in the mating pool. The average fitness of the population is calculated as
Figure 7.2.1 shows a roulette-wheel for five individuals having different fitness values. Since the third individual has a higher fitness value than any other, it is expected that the roulette-wheel selection will choose the third individual more than any other individual. This roulette-wheel selection scheme can be simulated easily.

Using the fitness value $F_i$ of all strings, the probability of selecting a string $p_i$ can be calculated. Thereafter, the cumulative probability ($P_i$) of each string being copied can be calculated by adding the individual probabilities from the top of the list. Thus, the bottom-most string in the population should have a cumulative probability ($P_n$) equal to 1. The roulette-wheel concept can be simulated by realizing that the $i^{th}$ string in the population represents the cumulative probability values from $P_{i-1}$ to $P_i$. The first string represents the cumulative values from zero to $P_1$. Thus, the cumulative probability of any string lies between 0 to 1. In order to choose $n$ strings, $n$ random numbers between zero to one are created at random. Thus, a string that represents the chosen random number in the cumulative probability range for the string is copied to
the mating pool. This way, the string with a higher fitness value will represent a large range in the cumulative probability values and therefore has a higher probability of being copied into the mating pool. On the other hand, a string with a smaller fitness value represents a smaller range in cumulative probability values and has a smaller probability of being copied into the mating pool.

In reproduction, good strings in a population are probabilistically assigned a large number of copies and a mating pool is formed. It is important to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are created by exchanging information among strings of the mating pool. Many crossover operators exist in the GA literature. In most crossover operators, two strings are picked from the mating pool at random and some portions of the strings are exchanged between the strings. A single-point crossover operator is performed by randomly choosing a crossing site along the string and by exchanging all bits on the right side of the crossing site as shown:

\[
\begin{array}{c|cccc|c|c|c}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0
\end{array}
\]

The two strings participating in the crossover operation are known as parent strings and the resulting strings are known as children strings. It is intuitive from this construction that good substrings from parent strings can be combined to form a better child string, if an appropriate site is chosen. Since the knowledge of an appropriate site is usually not known beforehand, a random site is often chosen. With a random site, the children strings produced may or may not have a combination of good substrings from parent strings, depending on whether or not the crossing site falls in the appropriate place. But we do not worry about this too much, because if
good strings are created by crossover, there will be more copies of them in the next mating pool generated by the reproduction operator. But if good strings are not created by crossover, they will not survive too long, because reproduction will select against those strings in subsequent generations.

It is clear from this discussion that the effect of crossover may be detrimental or beneficial. Thus, in order to preserve some of the good strings that are already present in the mating pool, not all strings in the mating pool are used in crossover. When a crossover probability of $P_c$ is used, only 100 $P_c$ percent strings in the population are used in the crossover operation and 100 $(1-P_c)$ percent of the population remains as they are in the current population.

A crossover operator is mainly responsible for the search of new strings, even though a mutation operator is also used for this purpose sparingly. The mutation operator changes 1 to 0 and vice versa with a small mutation probability, $p_m$. The bit-wise mutation is performed bit by bit by flipping a coin with a probability $p_m$. If at any bit the outcome is true, then the bit is altered; otherwise the bit is kept unchanged. The need for mutation is to create a point in the neighbourhood of the current point, thereby achieving a local search around the current solution. The mutation is also used to maintain diversity in the population.

These three operators are simple and straightforward. The reproduction operator selects good strings and the crossover operator recombines good substrings from good strings together to hopefully create a better substring. The mutation operator alters a string locally to hopefully create a better string. Even though none
of these claims are guaranteed and/or tested while creating a string, it is expected that if bad strings are created they will be eliminated by the reproduction operator in the next generation and if good strings are created, they will be increasingly emphasized.

**Fundamental concepts in fractal geometry**

The greater part of the applied scientific research of the past consisted of the analysis of human-made machines and the physical laws that govern their operation. The success of science relies on the predictability of the underlying experiments. Fractal geometry, as conceived by Mandelbrot, provides a mathematical model for many of the seemingly complex forms found in nature. One of the Mandelbrot's key observations was that these forms possess a remarkable statistical invariance under magnification. This may be quantified by a fractal dimension, a number that agrees with intuitive understanding of dimension but need not be an integer.

Fractal geometric structures exhibit a self-similarity when the distance at which they are viewed is changed. This self-similarity may be either exact or statistical. An exact self-similar fractal is the snowflake curve devised by the Swedish mathematician.

When a self-similar object is given as $N$ copies of itself, each one scaled down by a factor of $r$, the self-similarity dimension of the object is defined as

$$ D = \frac{\log N}{\log 1/r} $$
This definition assigns a dimension 1 to straight lines and 2 to squares, as expected. Fractals typically have a non-integer dimension. The snowflake curve has a dimension \( D = \log 4 / \log 3 = 1.262 \).

The mathematical model for a statistically self-similar object is given by fractional Brownian motion (FBM). In one dimension, FBM is a random process \( X(t) \) with Gaussian increments \( X(t_2) - X(t_1) \). The variance of these increments is proportional to \( |t_2 - t_1|^{2H} \), where \( 0 < H < 1 \). The increments of \( X \) are statistically self-similar with parameter \( H \). This means that, after setting \( t_0 = 0 \) and \( X(t_0) = 0 \), the two random functions, \( X(t) \) and \( r^H X(rt) \) are statistically indistinguishable. For a given number \( X_0 \), the point \( t \) that satisfies \( X(t) = X_0 \) will constitute a fractal point set, which is statistically self-similar. Its dimension is \( D_f = 1 - H \). The graph of \( X(t) \) is not self-similar, since one must scale in the \( t \)- and \( X \)-direction by different factors \( r \) and \( 1/r^H \) to obtain statistically equivalent graphs. This form of similarity has been termed self-affinity. The graph of \( X(t) \) has a fractal dimension of \( 2 - H \). Spectral analysis of FBM yields the spectral density \( S(f) \) of the process \( X(t) \). The density \( S(f) \) is proportional to \( 1/f^\beta \), where the spectral exponent \( \beta \) equals \( 2H+1 \). Thus, \( \beta \) is in the range from 1 to 3.

The generalization of FBM to higher dimensions is a multidimensional process \( X \) and \( (t_1, t_2, \ldots, t_n) \) with the properties analogous to the above. The random field \( X \) has stationary increments and is isotropic (i.e., all points \( (t_1, t_2, \ldots, t_n) \) and all directions are statistically equivalent). The random fields can also be characterized by their spectral density function or, equivalently, by their autocorrelation function.
Algorithms for random fractals

1. Midpoint displacement method

Assume that values \( X(0) = 0 \) and \( X(1) \) are given. \( X(1) \) may be obtained as a sample of a Gaussian random variable of variance \( \sigma^2 \). The interval \([0, 1]\) is partitioned into two sub-intervals \([0, 1/2]\), \([1/2, 1]\) and \( X(1/2) \) is defined as the average of \( X(0) \) and \( X(1) \) plus a displacement \( D_1 \), i.e.,

\[
X(1/2) = \frac{1}{2} [X(0) + X(1)] + D_1
\]

The displacement \( D_1 \) is computed as a sample of a Gaussian random variable with variance \( \Delta_1^2 \) proportional to \( \sigma^2/2^H \). The process is repeated with two intervals, i.e.,

\[
X(1/4) = \frac{1}{2} [X(0) + X(1/2)] + D_2
\]

\[
X(3/4) = \frac{1}{2} [X(1/2) + X(1/2)] + D_2
\]

where \( D_2 \) is Gaussian with variance \( \Delta_2^2 \) proportional to \( \sigma^2(2^2)^2 \). Note that two samples of \( D_2 \) in the above formula may be different. The process is continued with displacements \( D_n \) having variances \( \Delta_n^2 \) proportion to \( \sigma^2/(2^n)^2 \) in the \( n^{th} \) stage. This method is fast, but lacks mathematical accuracy, since the process \( X \) does not have stationary increments for \( H \neq 1/2 \).

2. Successive random additions method

This method improves on the stationarity of the increments of \( x \). Assume that \( X(t) \) is already approximated on an interval at equidistant points with grid size \( \Delta t \), and let \( r > 1 \) be a fixed number denoting a reduction factor. Then the grid size is reduced to \( \Delta t/r \) and values at the new equidistant points are defined by an interpolation procedure. Additionally, all values are offset by a sample of Gaussian random
variable with a proper choice of variance. This procedure is repeated until the desired resolution is achieved. The variance $\Delta_2^n$ of the displacement in the $n^{th}$ such stage of the algorithm must be proportional to $1/r^{2nH}$. The parameter $r > 1$ controls the lacunarity of the fractal. With a large value of $r$, only a very few stages are necessary and the lacunarity is especially drastic.

3. Rescale-and-add-method

An alternative method is to sum

$$X(t) = \sum_{k=k_0}^{k_1} \frac{S(r^k t)}{r^{kH}}$$

where $r > 1$, $0 < H < 1$ and $S$ is an auxiliary function similar to the sine and cosine functions e.g., $S$ may be defined as a smooth interpolant of random data at integer points $t = 0, \pm 1, \pm 2, \ldots$. For $k_0 = -\infty$, $k_1 = \infty$, one obtains a random fractal whose graph has a fractal dimension $2 - H$, and $r > 1$ determines lacunarity. In practice, the numbers $k_0$, $k_1$ are chosen to reflect the upper and lower crossover scales of the fractal i.e., basically $r^k 2^H$ and $r^k S^H$ will define the largest and the smallest structures seen in $X(t)$. This method is a summation of band limited functions. In this method, one-dimensional formulation is almost the same as the Mandelbrot function. The parameters $r$ and $H$ determine lacunarity and fractal dimension ($D = 2 - H$) of the graph of $X(t)$. They need not be fixed globally, but may change depending, for example, on $t$ or even $X(t)$.

There are numerous other applications of fractal geometry in image processing and pattern recognition. Two of them are:
1. Automatic segmentation of images based on fractal dimension, lacunarity, etc., which is useful in differentiating objects in an image such as man-made structures, forest, and sky.

2. Optimization of camouflage methods based on fractal analysis of the surroundings.

A method to generate fractal shapes that grow in space is based on Lindenmayer systems. Objects are represented as strings of symbols that are generated from an axiom and a set of production rules that are applied recursively to the symbols of the axiom and the resulting strings. The geometric representation of these strings is obtained through turtle graphics. Classic fractal curves, such as the snowflake curve, Hilbert’s space filling curve, etc., are easily and compactly formulated as L-systems. The main application is the modeling of growth and form of trees, bushes, and plants. These results stem from an inter-disciplinary effort in computer science and biology.

Illustration 7.2.1
An application of fractal geometry to Brownian motion

In 1827 the botanist R. Brown observed that minute particles suspended in a liquid were in constant motion and described highly irregular paths. This was later explained as resulting from molecular bombardment of the particle. A similar phenomenon was noticed in smoke particles in air.

A rigorous probabilistic model of Brownian motion was proposed by Wiener [72]. He constructed the ‘Wiener process’ which exhibits random behaviour
very similar to that of Brownian motion. Here we outline the details of this model and we shall prove that Brownian paths have Hausdorff dimension 2 with probability one.

If \( n \geq 1 \), let \( \Omega \) denote the class of all continuous paths \( \omega : [0, \infty) \to \mathbb{S}^n \) which have \( \omega(0) \) as the origin. We think of \( \omega(t) \) as the position at time \( t \) of a particle describing the path \( \omega \). It may be shown that there exists a probability measure \( p \) (with \( p(\Omega) = 1 \)) defined on a large system of subsets of \( \Omega \) such that:

(a) the paths have independent increments, that is, \( \omega(t_2) - \omega(t_1) \) and \( \omega(t_4) - \omega(t_3) \) are independent if \( t_1 \leq t_2 \leq t_3 \leq t_4 \).

(b) \( \omega(t+h) - \omega(t) \) has Gaussian distribution with zero mean and variance \( h \) for all \( t \).

In particular, the distribution of the path increments is stationary (\( \omega(t+h) - \omega(t) \) does not depend on \( t \)), is isotropic (independent of direction) and is such that

\[
p \{ \omega : |\omega(t+h) - \omega(t)| \leq \rho \} = ch^{-n/2} \int_0^\rho r^{n-1} \exp(-r^2/2h) dr, \quad \ldots \text{(18)}
\]

where \( c \) is a normalization constant chosen to ensure that \( p(\Omega) = 1 \).

The importance of the Brownian probability distribution is that it is the essentially unique distribution for which the paths have stationary, independent, isotropic increments of finite variance. These are conditions which are likely to apply in any physical situation.
Brownian motion may be thought of as the limiting case of a random walk as the step length tends to zero. The process is statistically self-similar in the sense that the paths \( \omega(t) \) and \( \omega(\gamma^2 t)/\gamma \) have the same probability distribution for any \( \gamma > 0 \).

**Result 7.2.1. (Falconer [16])**

If \( 0 < \lambda < 1/2 \), then for almost all \( \omega \in \Omega \) there exists \( h_0 > 0 \) such that

\[
| \omega(t + h) - \omega(t) | \leq |h|^\lambda
\]

for \( 0 \leq t \leq 1 \) and \( |h| < h_0 \).

**Theorem 7.2.1**

Brownian paths in \( \mathbb{R}^n (n \geq 2) \) have Hausdorff dimension 2 with probability one.

**Proof**

(a) We use a potential-theoretic method to show that \( \dim \omega \geq 2 \) for almost all paths \( \omega \). Fix \( 1 < s < 2 \). Using (18) we get

\[
\int_{\omega \in \Omega} \frac{dp(\omega)}{|\omega(t + h) - \omega(t)|^s} = c h^{-s/2} \int_0^1 r^{s+n-1} \exp(-r^2/2h)dr = c_1 h^{-s/2}.
\]

on substituting for \( r^2/h \) in the integrand, where \( c_1 \) is independent of \( h \) and \( t \).

Thus

\[
\int_{t=0}^1 \int_{u=0}^1 \int_{\omega \in \Omega} \frac{dp(\omega)dtdu}{|\omega(t) - \omega(u)|^s} = \int_{t=0}^1 \int_{u=0}^1 \frac{c_1dtdu}{|t-u|^{s/2}} < \infty.
\]
Then measurability properties of Brownian motion ensure that \(|\omega(t) - \omega(u)|^s\) is measurable with respect to the product measure on \([0, 1] \times [0, 1] \times \Omega\) so we conclude, using Fubini's theorem, that

\[
\int_0^1 \int_0^1 \frac{dt du}{|\omega(t) - \omega(u)|^s} < \infty.
\]

for almost all \(\omega \in \Omega\).

There is a natural mass distribution \(\mu_\omega\) on the path \(\omega\), given by \(\mu_\omega(E) = \mathcal{H} \{t: 0 \leq t \leq 1 \text{ and } \omega(t) \in E\}\). Thus for almost all \(\omega \in \Omega\) the s-energy \(I_s(\mu_\omega)\) is finite, so \(s \leq \dim (\omega[0, 1]) \leq \dim \omega\). This is true for all \(s < 2\), so \(\dim \omega \geq 2\) for almost all \(\omega \in \Omega\).

Take \(s > 2\). The above result 7.2.1 implies that for almost all \(\omega \in \Omega\) we may find \(h_0 > 0\) such that \(|\omega(t + h) - \omega(t)| \leq |h|^{1/s}\) if \(|h| \leq h_0\) and \(0 \leq t < t + h \leq 1\). For such a path \(\omega\) the subarc \(\omega[t, t + h]\) may be enclosed in a ball of radius \(h^{1/s}\) if \(h \leq h_0\).

Consequently, if \(m\) is an integer with \(1/m \leq h_0\), we may enclose

\[
\omega[0, 1] = \bigcup_{j=1}^m \omega[(j-1)/m, j/m] \text{ in } m \text{ balls, each of radius } m^{1/s}.
\]

Thus \(\mathcal{H}^s(\omega[0, 1]) \leq m2^s (m^{1/s})^s = 2^s\) if \(\delta \geq 2m^{-1/s}\). Letting \(m \to \infty\) we conclude that \(\mathcal{H}^s(\omega[0, 1]) \leq 2^s\) if \(s > 2\), so \(\mathcal{H}^s(\omega[0, 1]) = 0\) and thus \(\mathcal{H}^s(\omega) = 0\).

**SECTION 7.3. WALSH TRANSFORM**

When \(N = 2^n\), the discrete Walsh transform of a function \(f(x)\), denoted \(W(u)\), is obtained by substituting the kernel
\[ g(x, u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-1-i}(u)} \]  

into the 1-D Fourier transform

\[ T(u) = \sum_{n=0}^{N-1} f(s) g(x, u) \]  

In other words,

\[ W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-1-i}(u)} \]  

where \( b_k(z) \) is the \( k^{th} \) bit in the binary representation of \( z \).

The values of \( g(x, u) \), excluding the \( 1/N \) constant term, are listed in Table 7.3.1. for \( N = 8 \).

<table>
<thead>
<tr>
<th>( u )</th>
<th>( x )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>7</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
The array formed by the Walsh transformation kernel is a symmetric matrix having orthogonal rows and columns. These properties lead to an inverse kernel identical to the forward kernel except for a constant multiplicative factor of \( 1/N \); that is,

\[
h(x, u) = \prod_{i=0}^{n-1} (-1)^{b_i(x)h_{n-1,i}(u)} \quad \ldots (22)
\]

Thus the inverse Walsh transform is

\[
f(x) = \sum_{u=0}^{N-1} W(u) \prod_{i=0}^{n-1} (-1)^{b_i(x)h_{n-1,i}(u)} \quad \ldots (23)
\]

The validity of Eq.(23) is easily established by substituting Eq.(21) for \( W(u) \) and making use of the orthogonality condition mentioned previously. Since the forward and inverse Walsh transforms differ only by the \( 1/N \) term. Thus any algorithm for computing the forward transform may be used directly to obtain the inverse transform simply by multiplying the result of the algorithm by \( N \).

The 2-D forward and inverse Walsh kernels are given by the relations

\[
g(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(x)h_{n-1,i}(u) + b_i(y)h_{n-1,i}(v)]} \quad \ldots (24)
\]

and

\[
h(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(x)h_{n-1,i}(u) + b_i(y)h_{n-1,i}(v)]} \quad \ldots (25)
\]

Although grouping both \( 1/N \) terms in front of \( g(s, y, u, v) \) or \( h(x, y, u, v) \) is valid, the forms of Eqs.(24 and 25) are preferable in image processing applications, where
there is equal interest in taking the forward and inverse transforms. The forward and
inverse Walsh transforms that are equal in form are given by

\[
W(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v)}
\]

and

\[
f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} W(u, v) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v)}
\]

Thus any algorithm used to compute the 2-D forward Walsh transform may also be
used without modification to compute the inverse transform.

The Walsh transform kernels are separable and symmetric, because

\[
g(x, y, u, v) = g_1(x, u) g_1(y, v)
\]

\[
h(x, u) h_1(y, v)
\]

\[
= \frac{1}{\sqrt{N}} \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}
\]

\[
= \frac{1}{\sqrt{N}} \prod_{i=0}^{n-1} (-1)^{b_i(y) b_{n-1-i}(v)}.
\]

Hence \(W(u, v)\) and its inverse may be computed by successive applications of the 1-D
Walsh transform in Eq.(21).

The Walsh transform may be computed by a fast algorithm, led by the basic
relations

\[
W(u) = \frac{1}{2} \left[ W_{\text{even}}(u) + W_{\text{odd}}(u) \right] \quad \ldots (26)
\]

and
\[
W(u+M) = \frac{1}{2} [W_{\text{even}}(u) - W_{\text{odd}}(u)] \quad \text{...(27)}
\]

where \( M = N/2 \), \( u = 0, 1, \ldots, M - 1 \), and \( W(u) \) denotes the 1-D Walsh transform.

**Illustration 7.3.1**

If \( N = 4 \), use of Eq.21 results in the following sequence of steps:

\[
W(0) = \frac{1}{4} \sum_{x=0}^{3} [f(x) \prod_{i=0}^{1} (-1)^{b_{i}(x)b_{1,i}(0)} ]
\]

\[
= \frac{1}{4} [f(0) + f(1) + f(2) + f(3)]
\]

\[
W(1) = \frac{1}{4} \sum_{x=0}^{3} [f(x) \prod_{i=0}^{1} (-1)^{b_{i}(x)b_{1,i}(1)} ]
\]

\[
= \frac{1}{4} [f(0) + f(1) - f(2) - f(3)]
\]

\[
W(2) = \frac{1}{4} \sum_{x=0}^{3} [f(x) \prod_{i=0}^{1} (-1)^{b_{i}(x)b_{1,i}(2)} ]
\]

\[
= \frac{1}{4} [f(0) - f(1) + f(2) - f(3)]
\]
\[
W(3) = \frac{1}{4} \sum_{x=0}^{3} \left[ \sum_{\nu=0}^{1} (-1)^{\nu} b_{\nu,b_{1,1},(3)} \right] f(x)
\]

\[
= \frac{1}{4} [f(0) - f(1) - f(2) + f(3)].
\]

Subdividing these results into two groups shows the validity of Eqs. (26) and (27):

\[
W_{\text{even}}(0) = \frac{1}{2} [f(0) + f(2)] \quad \text{and} \quad W_{\text{odd}}(0) = \frac{1}{2} [f(1) + f(3)]
\]

\[
W_{\text{even}}(1) = \frac{1}{2} [f(0) - f(2)] \quad \text{and} \quad W_{\text{odd}}(1) = \frac{1}{2} [f(1) - f(3)].
\]

From Eq. (26),

\[
W(0) = \frac{1}{2} [W_{\text{even}}(0) + W_{\text{odd}}(0)]
\]

\[
= \frac{1}{2} [f(0) + f(1) + f(2) + f(3)]
\]

and

\[
W(1) = \frac{1}{2} [W_{\text{even}}(0) + W_{\text{odd}}(1)]
\]

\[
= \frac{1}{4} [f(0) + f(1) - f(2) - f(3)].
\]
Computing the next two terms from these results, using Eq.(27), gives

\[ W(2) = \frac{1}{2} \left[ W_{\text{even}}(0) - W_{\text{odd}}(0) \right] \]

\[ = \frac{1}{2} \left[ f(0) - f(1) + f(2) - f(3) \right] \]

and

\[ W(3) = \frac{1}{2} \left[ W_{\text{even}}(1) - W_{\text{odd}}(1) \right] \]

\[ = \frac{1}{4} \left[ f(0) - f(1) - f(2) + f(3) \right] \]

Thus computation of \( W(u) \) by Eq.21 or by Eqs.(26) and (27), yields identical results.

A transform and its inverse may be expressed in terms of a series expansion involving the appropriate kernels. The kernels depend only on the indexes \( u, v, x \) and \( y \), so that the kernels serve as a set of basis functions whose nature is completely fixed once the dimensions of the image have been fixed.

**Fuzzy grammar and syntactic recognition**

The concept of a formal grammar is often found to be too rigid to handle real patterns, which are usually distorted or noisy yet still retain underlying structure. When the indeterminacy of the patterns is due to inherent vagueness rather than randomness, fuzzy language can be a better tool for describing the ill-defined structural information. In this case, the generative power of a grammar is increased by introducing fuzziness either in the definition of primitives (labels of the fuzzy sets)
or in the physical relations among primitives (fuzzified production rules), or in both of these. A fuzzy grammar produces a language that is a fuzzy set of strings with the membership value of each string denoting the degree of belonging of the string in that language. The grade of membership of an unknown pattern in a class described by the grammar is obtained using a max-min composition rule.

Let $V_T^*$ denote the set of finite strings of alphabet $V_T$, including the null string, $\lambda$. Then, a fuzzy language (FL) on $V_T$ is defined as a fuzzy subset of $V_T^*$. Thus, FL is the fuzzy set,

$$FL = \sum_{x \in V_T^*} \mu_{FL}(x)$$

where $\mu_{FL}(x)$ is the grade of membership of the string $x$ in FL. It is further assumed that all other strings in $V_T^*$ have 0 membership in FL.

A fuzzy grammar may be viewed as a set of rules for generating a fuzzy subset of $V_T^*$. A fuzzy grammar FG is a 6-tuple given by

$$FG = (V_N, V_T, P, S, I, \mu)$$

where in addition to the following production rules

- $\langle$ sentence $\rangle \rightarrow \langle$ noun phrase $\rangle \langle$ verb phrase $\rangle$
- $\langle$ noun phrase $\rangle \rightarrow \langle$ article $\rangle \langle$ noun $\rangle$
- $\langle$ verb phrase $\rangle \rightarrow \langle$ verb $\rangle \langle$ adverb $\rangle$
- $\langle$ article $\rangle \rightarrow \text{The}$
- $\langle$ noun $\rangle \rightarrow \text{athlete}$
- $\langle$ verb $\rangle \rightarrow \text{jumped}$
- $\langle$ adverb $\rangle \rightarrow \text{high}$
We have $J : \{r_i \mid i = 1, 2, \ldots, n\}$, i.e., the number of production rules, and $\mu$ is a mapping $\mu : J \to [0, 1]$, such that $\mu(r_i)$ denotes the membership of $P$ of the rule labeled $r_i$.

**Illustration 7.3.2**

Suppose a fuzzy grammar is given by $FG_2 = (\{S, A, B\}, (a, b, c), P, S, J, \mu)$ where $J$, $P$, and $m$ are as follows:

- $r_1 : S \to aA$ with $\mu(r_1) = \mu_H(a)$
- $r_2 : A \to bB$ with $\mu(r_2) = \mu_V(a)$
- $r_3 : B \to c$ with $\mu(r_3) = \mu_{ab}(a)$

with the primitives $a$, $b$, and $c$ being horizontal, vertical, and oblique directed line segments, respectively, as seen in Fig. 7.3.1. The membership functions for these line segments are given here with reference to Fig. 7.3.2.
Fig. 7.3.2. Membership functions for horizontal, vertical and oblique lines

\[
\mu_H(\theta) = 1 - |\tan \theta| \quad \text{and} \quad \mu_v(\theta) = 1 - \left| \frac{1}{\tan \theta} \right|
\]

\[
\text{and } \mu_{ob}(\theta) = 1 - \left| \frac{\theta - 45^\circ}{45^\circ} \right|
\]

From the three rules, the only string generated is \( x = abc \). This string is, of course, a right triangle, as seen in Fig. 7.3.2, and is formed from the specified sequence in the string \( abc \). In this syntactic recognition, the primitives (line segments) will be concatenated in the "head-to-tail" style.

**Illustration 7.3.3**

For the syntactic pattern recognition of electrical symbols in an electrical design diagram such as a resistor, an inductor, and a capacitor, etc., several fuzzy grammars can be used for each different symbol. An inductor and a resistor can be assigned to the same fuzzy language. If the fuzzy grammar for this language is called \( FG_1 \), then \( FG_1 \) is defined as

\[
FG_1 = (\{A, S\}, \{a_i\}, P, S, \{1, 2, 3\}, \mu)
\]
where J.P, and \( \mu \) are as follows:

1. \( S \rightarrow A \) with \( \mu(1) = 0.15 \)
2. \( S \rightarrow a_i S \) with \( \mu(2) = 0.85 \)
3. \( A \rightarrow a_i \) with \( \mu(3) = 1 \)

and \( a_i \) is a primitive. Suppose \( a_1 \) (i=1) represents the primitive (inductor symbol) \( \cdot \) and \( a_2 \) (i=2) represents the primitive (resistor symbol) \( \cap \). The preceding fuzzy grammar generates the fuzzy language as

\[
L(FG_i) = \{ x | x = a_i^n, \ n = 1, 2, 3, \ldots \}
\]

Further, if the concatenation of a head-tail type is used for the generation of string \( x \), then \( x = a_i^n \) infers the pattern of an inductor when \( i = 1 \) or a resistor when \( i = 2 \).

These ideas are shown in Fig. 7.3.3.

![Fig. 7.3.3. Directed line segments for example 7.3.3. (a) primitives and (b) patterns produced by the primitives](image-url)
The membership values for these two patterns can be expressed as

1. \( i = 1 \), which infers a pattern of an inductor.

\[
\mu_{L(FG_1)}(a_1^n) = \begin{cases} 
0.15 & n = 1 \\
0.85 & n \geq 2
\end{cases}
\]

2. \( i = 2 \), which infers a pattern of a resistor

\[
\mu_{L(FG_1)}(a_2^n) = \begin{cases} 
0.15 & n = 1 \\
0.85 & n \geq 2
\end{cases}
\]

and these patterns and associated membership values are summarized in Table 7.3.2.

Another fuzzy grammar \( FG_2 \) can be developed to recognize the pattern of a capacitor. If \( FG_2 \) is expressed in a general form,

\[
FG_2 = (V_N, V_T, P, S, J, \mu)
\]

then

\[
V_N = \{S, R\}
\]

\[
V_T = \{a_3, a_4\} \text{ or } V_T = \{a_4, a_5\}
\]

\[
P : S \rightarrow L(a_3, a_4) \text{ or } S \rightarrow L(a_4, a_5)
\]

where \( a_i \) (\( i = 3, 4, 5 \)) is a primitive in which \( a_3 \) represents the symbol “)”, \( a_4 \) represents the symbol “|”, and \( a_5 \) represents the symbol “(”. \( L(x,y) \) means “\( x \) is to the right of \( y \)”. Therefore, the fuzzy language decided by \( FG_2 \) represents a capacitor in reality. A pattern \( S \) that meets \( FG_2 \) can be considered a capacitor. Besides, \( FG_2 \) belongs to a context-free grammar.

If an unknown pattern \( S \) has the primitives of \( a_3 \) and \( a_4 \), and the membership values for \( a_3 \) and \( a_4 \) are given by
\[ \mu_{L(FG2)}(a_3) = 0.8 \quad \mu_{L(FG2)}(a_4) = 1 \]

then the membership of \( S \) representing a capacitor is

\[ \mu_c(S) = \min (\mu_{L(FG2)}(a_3), \mu_{L(FG2)}(a_4)) = \min(0.8, 1) = 0.8 \]

Similarly, if an unknown pattern \( S \) has the primitives of \( a_4 \) and \( a_5 \), and the membership values for \( a_4 \) and \( a_5 \) are

\[ \mu_{L(FG2)}(a_4) = 0.9 \quad \mu_{L(FG2)}(a_5) = 0.8 \]

then the membership for \( S \) as a capacitor is

\[ \mu_c(S) = \min (\mu_{L(FG2)}(a_4), \mu_{L(FG2)}(a_5)) = \min(0.9, 0.8) = 0.8 \]

By modification of the fuzzy grammar used for capacitors, \( FG_2 \) can be used to develop a fuzzy language for the electrical source (AC, DC) patterns \( \sim \) and \( \cdot 1 \). In those cases, \( V_T \) and \( P \) are changed to meet requirements of different patterns.

---

**Table 7.3.2. The recognition of electrical elements of inductors and resistors**

<table>
<thead>
<tr>
<th>Number</th>
<th>Membership of primitives</th>
<th>Inference of the element</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = 1 )</td>
<td>( 0.15 )</td>
<td>0</td>
</tr>
<tr>
<td>( n \geq 2 )</td>
<td>( 0.85 )</td>
<td>0</td>
</tr>
<tr>
<td>( n = 1 )</td>
<td>0</td>
<td>0.15</td>
</tr>
<tr>
<td>( n \geq 2 )</td>
<td>0</td>
<td>0.85</td>
</tr>
</tbody>
</table>