Int. J. Contemp. Math. Sci., Vol. 1, 2006, no. 7, 317 - 325

us-Crop Based Compact Plasma Memory

Fevzi Ünlü

Yasar University, Alsancak
Izmir, Turkey
fevzi.unlu@yasar.edu.tr

Abstract

[1] introduces the notion of *I@I*. [2] develops, studies and introduces convoluted Prüfer's type *I@I*. [3, 4, 5] develop, study and introduce the notion of KBO. [6] develops, studies and introduces us-crops in us-culture. [4] develops, studies and introduces integration and derivation of KBO. [7] develops, studies and introduces linear circular convoluted us-crop I@I KBO cluster generating us-crop. [8] develops, studies and introduces tuze-channeled KBO of DCP. This paper will develops, study and introduce us-crop based plasma memory. For this: 1) An abstract formal language *plemal L* to generate any us-crop based plasma memory has been designed on an alphabet called plema $\ddot{A}$ and it is introduced. *2) The found result is planed to be used for generating an us-culture based pointing action that generates a remote FTD programmable plasma memory cluster type.*

Mathematics Subject Classification: 00A05, 00A08, 00A35, 00A71, 03F50, 03H35, 44A55, 47A65, 68P05, 68P20, 94A05, 94A15

Keywords: *KBO*, *I@I*, pointer, pointing action, convolution, us-crop, us-culture, us-crop *I@I KBO* cluster, us-crop *I@I KBO* cluster memory, us-crop *I@I KBO based compact plasma memory type*

1 Introduction

[1] introduces the notion of *I@I*. [2] develops, studies and introduces convoluted Prüfer's type *I@I* clusters. [3, 4, 5] develop, study and introduce the notion of knowledge based object -*KBO*. [6] develops, studies and introduces

us-crops in us-culture. [4] develops, studies and introduces integration and derivation of *KBO*. [7] develops, studies and introduces linear-circular convoluted us-crop *I@I KBO* cluster generating us-crops. [8] develops, studies and introduces tuze-channeled KBO of DCP. This paper is developing, studying and introducing us-crop based compact plasma memory.

2 Preliminary Notes

In this section, the preliminary notions which are necessary in us-crop based compact plasma memory will be developed.

2.1 u-Crop and s-Crop Based Plasma Memory Types

Definition 1 (a) Let *FTD* be formal technology dependent. (i) A mod p enforced and m-ary negative u-crop I@I KBO cluster memory is called as an m-ary negative FTD plasma memory type. It is represented by $X_- = p\circlearrowleft m \iff$ [-m; -0]X@X [$+\phi$; $+\phi$] = array[-m ... -0] of (X_- mod p, X_+ mod ϕ). It appears as in Fig 1 under zooming operator action.

(ii) A mod q enforced and n-ary positive s-crop I@I KBO cluster memory is called an n-ary positive plasma memory named by $X_+ = n\circlearrowleft q \iff$ [-ϕ; -ϕ]X@X [+0; $+n$] mod q = array[+0 ... +n] of (X_- mod ϕ, X_+ mod q). It appears as in Fig 2 under zooming operation.

2.2 us-Crop Based Plasma Memory Type

Here an (m,n)-ary (positive, negative) FTD plasma memory type will be developed by a corollary.

Corollary 1 There is a deterministic (m, n)-ary (negative, positive) plasma memory type construction A = [-m; -0]A@A[+0; +n](A_- mod p, A_+ mod q) generating FTD technology.

Proof Let 1) (a) Let *FTD* be formal technology dependent. 2) Let A_- be a mod p enforced circularly m-ary negative plasma memory type. It can briefly be represented by $A_- = p\circlearrowleft m$. 3) Let A_+ be a mod q enforced circularly n-ary positive plasma memory. It can briefly be represented by $A_+ = n\circlearrowleft q$. 4) Let @ be a root origin dependent concatenation operator. One can obtain $A = A_- @ A_+ = (p\circlearrowleft m)@(n\circlearrowleft q)$ = [-m; -0]A@A[+0; +n] (A_- mod p, A_+ mod

q). It appears as in Fig 3 under zooming operation. Hence there is a FTD technology for generating A.

3 Developments

In this section, some data structures on the u-crop, s-crop and s-crop for the purpose of developing us-culture by a formal language plemal L defined on a special alphabet plea Ä, are studied and introduced.

3.1 Data Structure on us-Crops

Syntax 1 (a) If X, Y and Z are us-crops, then the set of ordered pairs Z = X × Y = { (x, y): x ∈ (X_- ∪ X_+) ⟹ x ∈ (X_- @ X_+) while y ∈ (Y_- ∪ Y_+) ⟹ y ∈ (Y_- @Y_+)} is called the Cartesian product of X and Y. (b) If X and Y are the same us-crop, say X = Y = W, then X x Y may be written as Z = W^2. (c) Similarly ordered n-tuples(or n-ary) as members of a set W may be constructed. (d) There is natural way combining r-tuples and s-tuples to form (r+s)–tuples : ((x_1, ..., x_r) (y_1, ..., y_s)) = (x_1, ..., x_r, y_1, ..., y_s).

Semantics 1 (a) Variable symbols such as x and y stands for r-tuples as well as individual quantities. (b) If x is r-tuples and y is s-tuples the convention of (x, y) ∈ W^{r+s} means that x ∈ W^r and y ∈ W^s. (c) There is a single o-tuple ( ) constitute the set W^0, so that ( ) (x_1 ,..., x_r) = (x_1 ,..., x_r) ( ) = (x_1, .., x_r) and W^0 x W^r = W^r x W^0 = W^r. (d) Z = {z = (x, y) : x ∈ (X_- ∪ X_+) ⟹ x ∈ (X_- @ X_+) while y ∈ (Y_- ∪ Y_+) ⟹ y ∈ (Y_- @Y_+)} means Z = {z = (x, y) : (x, y) ∈ ((X_- × Y_-) ⟹ y ∈ (X_- @Y_-) ∪ (X_- × Y_+) ⟹ y ∈ (X_- @Y_+) ∪ (X_+ × Y_-) ⟹ y ∈ (X_+ @Y_-) ∪ (X_+ × Y_+) ⟹ y ∈ (X_+ @Y_+)) }. (e) Examples (i) W^3 = W x W x W is the set of 3-ary pointers in the three dimensional space, (ii) W^2= W x W is the set of 2-ary pointers in the two dimensional space, and W^0 is the 0-ary pointers in the root origin space which is represented in this study by $[-\phi; -\phi]W^0@W^0[+\phi; +\phi]$ ↬ @ or $[-\phi; -\phi]W^0@W^0[+\phi; +\phi]$ ↞ @ . Where ↬ is called root origin concatenation operator generating operator and ↞ is called root origin us-crop KBO plasma memory generating operator.

Syntax 2 (a) A relation f : X ⟶ Y from X to Y is any collection of ordered pairs (x, y) ∈ X x Y. That is a subset of X x Y. (b) The domain of f, written dom f, is { x : (x, y) ∈ f, for some y }, that is the set of inputs which produce some output. (c) The range of f, written ran f, is { y : (x, y) ∈ f,

for some x }, that is, the set of outputs. (d) The inverse $f^{-1} : Y \longrightarrow X$ of a relation $f : X \longrightarrow Y$ is the set of pairs $\{(y, x) : (x, y) \in f \}$.

Semantics 2 (a) A relation f: $X \longrightarrow Y$ is a function if f(x) contains at most one element for each $x \in X$, that is, each input produces a unique output or not output at all. b) A relation (or function) f is 1-1 if f(x) = y and $f(x^{\dagger}) =$ y implies $x = x^{\dagger}$ that is, two different inputs never produce the same output.

3.2 us-Culture Generating Plemal L on Plema Ä

Definition 2 (a) Let p, m, n and q be finite positive integers. 1) $\bar{A} = \{$ p◡m, n◡q $\}$ is called an **us-alphabet of formal FTD I@I KBO plasma memory type**. 2) $£ \subseteq \bar{A}^*$ is called an **us-language of formal FTD I@I KBO plasma memory types**. Where $\bar{A}^*$ is the Kleen's Closure of $\bar{A}$. (b) A quadruple G = (T, N, P, Σ) is called an **us-grammar of formal FTD I@I KBO plasma memory types. Where** 1) T is a **finite set of terminal formal FTD token I@I KBO plasma memory type**, 2) N is a **finite set of non-terminal formal FTD token I@I KBO plasma memory type,** 3) P is *a* **finite set of production rules** on terminal formal FTD token I@I KBO plasma memory type and non-terminal formal FTD token I@I KBO plasma memory type, Σ is special **linear accumulator register** for registering some pointers for pointing out the elements of an us-crop cluster by its integrated pointing action.

Theorem 1 1) Let p, m, n and q be finite positive integers. 2) Let $\bar{A}$ be an us-alphabet of formal FTD I@I KBO plasma memory type. 3) Let £ be an us-language **of formal FTD** I@I KBO plasma memory type. There is at least one us-grammar G of formal FTD I@I KBO plasma memory types for constructing an us-language $\pounds$ of formal FTD I@I KBO plasma memory type on us-alphabet $\bar{A}$ of formal FTD I@I KBO plasma memory type.

Proof (a) 1) Let p, m, n and q be finite positive integers. 2) Let $\bar{A}$ be an us-alphabet of formal FTD I@I KBO plasma memory types. 3) Let £ be an us-language of formal FTD I@I KBO plasma memory type. (b) Let 1) $\leftharpoonup$ be **assignment operator** for assigning a KBO at right to a KBO at the left . 2)Let $\mapsto$ be **transformation operator** for deriving a KBO at the right from a KBO at the left. (c) 1) Let a and b two terminal symbol registers representing $X_{-} = p◡m$ and $X_{+} = n◠q$ under assignments of a $\leftharpoonup p◡m$ and b $\leftharpoonup n◠q$. 2) Let S be a non-terminal token symbol. 3) Σ be a special accumulator register variable for registering an us-crop string during a production process. Under these considerations one has : 1) A terminal us-alphabet T = {a, b} = { a $\leftharpoonup$ $X_{-} = p◡m$, b $\leftharpoonup$ $X_{+} = n◠q$ }. 2) A non-terminal N = { S }. 3) Assume

a set of FTD production rules as P = { $R_1 : \Sigma \longmapsto S$, R_2: $S \longmapsto$ aSb, R_3 = S $\longmapsto \lambda$} is available. Hence there is an us-grammar G = (T, N, P, Σ). This G generates a language $\pounds = \{a^k \circ b^k = a^k\ b^k$: $0 \leq$ k $\leq$ maxint(m, n) } by the following Algorithm-A and Algorithm-B(k) :

Algorithm A *S0: Begin; S1:* $\Sigma \longmapsto$ *S; S2:* $k \leftarrowtail 0$; ℓ = *maxint(m,n); S3: Do Algorithm-B(k) while* $k \leq \ell$; *S4 : Stop; S5: End.*

Algorithm-B(k) *S0: Begin; S1:* $r \leftarrowtail 0$; *do* $S \longmapsto$ *aSb,* $r \leftarrowtail r + 1$, *while* $r \leq k$; *S2:* $S \longmapsto \lambda$ *; S3:* $\Sigma \leftarrowtail$ *S; S4: write*$[\Sigma]$ *; S5: return; S6: End. Where Si stands for Step(i).*

Definition 3 Let N be the set of natural numbers. Consider m, p, q, n $\in$ N. Let ${}^u r_1$, ${}^u r_2$, ${}^s r_1$ and ${}^s r_2$ be four registers designed for registering m, p, q and n in their contents. 1) ${}^u r = {}^u r_1 \circlearrowleft {}^u r_2 \Longrightarrow \square \circlearrowleft \square \Longrightarrow$ m$\circlearrowleft$p is called an u-crop plasma memory cluster generating register that controlled by m and p. 2) ${}^s r = {}^s r_1 \circlearrowleft {}^s r_2 \Longrightarrow \square \circlearrowleft \square \Longrightarrow$ q$\circlearrowleft$n is called s-crop plasma memory cluster generating register controlled by q and n. 3) (a) m or n is called dimension controlling values while they are in ${}^u r = {}^u r_1 \circlearrowleft {}^u r_2$ or ${}^s r = {}^s r_1 \circlearrowleft {}^s r_2$; (b) p or q is called mod controlling values while they are in ${}^u r = {}^u r_1 \circlearrowleft {}^u r_2$ or ${}^s r = {}^s r_1 \circlearrowleft {}^s r_2$.

Result 1 (a) Assume [r] is the content operator of the register r. One has: $[{}^u r_1]$ = m, $[{}^u r_2]$ = p, $[{}^s r_1]$ = q, and $[{}^s r_2]$ = n, $[{}^u r]$ = m$\circlearrowleft$ p and $[{}^s r]$ = q$\circlearrowleft$n (b) m$\circlearrowleft$p is an u-crop plasma memory cluster generating code. (c) q$\circlearrowleft$n is an s-crop plasma memory cluster generating code.

Definition 4 Let m, p, q, n $\in$ N. Ä = $\{{}^u r, {}^s r\ \} = \{{}^u r_1 \circlearrowleft {}^u r_2, {}^s r_1 \circlearrowleft {}^s r_2\}$ $\Longrightarrow\{(\square \circlearrowleft \square)\ , (\square \circlearrowleft \square)\}\Longrightarrow$ {(m$\circlearrowleft$p), (q$\circlearrowleft$n)}$\Longrightarrow$ {m$\circlearrowleft$p, q$\circlearrowleft$n} is called a **p**rogrammable **l**inearly-**e**xtending-contracting-plasma **m**emory **a**lphabet. It is briefly called in this paper as *plema* Ä .

Theorem 2 There exist 16 distinguishable configuration states while one is programming on a *plema*.

Proof Let Ä be a plema. Let $\square$ be a generalized linearly extending-contracting register token KBO for representing ${}^u r_1$, ${}^u r_2$, ${}^s r_1$,and ${}^s r_2$ of ${}^u r = {}^u r_1 \circlearrowleft {}^u r_2$ or ${}^s r = {}^s r_1 \circlearrowleft {}^s r$ in the *plema* Ä. Each $\square$ can be at least in two states, namely empty and non empty. Hence three exist 4 variables that each one is taking 2 values. This leads that three exist 16 distinguishable configuration states while one is programming on *plema* Ä.

Definition 3 Let * be Kleen's closure operator. L $\subseteq$ Ä* is called a *plemal*. Hence *plemal* is a formal language on *plema* Ä .

Theorem 2 Let L be a *plemal* on *plema* Ä. There exist a semantic algorithm to interpret the meaning of each vocabulary x $\in$ L such a way that x

becomes a code to generate an us-culture.

Proof Let L be a *plemal* on *plema* Ä. If x ∈ L. The following Algorithm-A converts x to an us-culture x.

Algorithm-C *S0: Begin; S1: Mark m◡p appearing in x by counting from left to right in a suitable modulo-δ for finding polarization index of each m◡p. S2: Mark q◡n in x by counting from left to right in modulo-ρ for finding polarization index of each q◡n. S3: Find μ = maximum(δ, ρ). S4: Generate a flying root concatenation object @ that it flies on x and points to construct polarized us-crops of an us-culture by matching the found polarization indexes in the counting process on u-crops with the found polarization indexes in the counting process on s-crops of x. S5: If there is no matching found for the pointed polarization index of an u-crop or s-crop, then match it by an empty s-crop or an empty u-crop. S6: Construct an μ-ary polarized us-culture named by x for the given vocabulary x ∈ L. S7: Stop; S8: End.*

Example 1 1) Let x = (3◡5)(5◡2)(4◡5)(3◡4)(3◡2)(3◡2)(5◡3) in a *plemal L*. 2) The counting process gives the following data structure:

x = (3◡5)(5◡2)(4◡5)(3◡4)(3◡2)(3◡2)(5◡3).
0 1 0 1 2 2 3

3)This data structure is a vocabulary in L. Further it is a code that produces the following polarized us-culture named by x under Algorithm C.

x = { [-3; -0] ^{0}x@ ^{0}x[+0;+5] (mod 5, mod4), [-5; -0] ^{1}x@ ^{1}x[+0; +4](mod 2, mod 3), [-3; -0] ^{2}x@ ^{2}x[+0;+2](mod 2, mod 3), $[-\phi; -\phi]^{3}x@\ ^{3}x[+0; +3]$ (mod ϕ, mod 5) }

Theorem 3 (a) A non-empty vocabulary in *plemal L* with a set of pointing action, generates a remote programmable I@I KBO token cluster. (b) An us-crop that its spike values are vocabularies in *plemal L*, each has a set of enforced pointing actions, generates a remote programmable us-culture I@I KBO cluster of remote programmable us-culture I@I KBO clusters.

Proof 1) Let w be a non-empty vocabulary in *plemal* L. Its enforced pointing action to produce an us-culture like in *Example 1.* 2) Let x be us-crop. It has a type of [-m; -0]x@x[+0; +n] (A_{-} mod w, A_{+} mod w) and its enforced pointing actions generates a remote programmable us-culture I@I KBO cluster of remote programmable us-culture I@I KBO clusters. Fig 4 represents a remote enforced point action generating tool *f*or generating a remote I@I KBO cluster of remote I@I KBO clusters.

4 Main Results

In this paper: 1) An us-crop based compact plasma memory type has been studied, constructed and introduced. 2) An abstract formal language *plemal* L to generate any us-crop based plasma memory has been designed on an alphabet called *plema* $\ddot{A}$ and it is introduced. *The found result is planed to be used for generating an us-culture based pointing action that generates a remote FTD programmable plasma memory cluster of remote FTD programmable plasma memory clusters, in an other paper.*

References

[1] F. Ünlü and S. Sönmez, *I@I : Tıkız Internet Tabanlı Tıkız Internet*, II. Proceeding of Information Technology Congress, pp. 246-248, 1-May 2003, Pamukkale University-Denizli.

[2] F. Ünlü and S. Sönmez, *Convoluted Prüfer's Type I@I,* International Mathematical Journal, Vol. 4, no. 6, pp. 539 - 547, 2003.

[3] F. Ünlü, *An Intuitive Differential Equation Model for Knowledge Based Objects(KBO) Representation of Science*, ISCISXIV, Proceeding of The Fourteenth International Symposium on Computer and Information Science, pp1066-1068, October, 18-20, Kuşadası, Aydın, Turkey.

[4] F. Ünlü, *FTD Grammar Graphs*, (a) International Conference on Mathematical Modelling and Scientific Computing, Middle East Technical University and Selçuk University, Ankara and Konya , Turkey, April 2-6, 2001. (b) International Journal of Computer Mathematics Vol. 80, no. 1, pp1-9, January 2003.

[5] F. Ünlü, Chance Constrained Threshold KBO System Design, International Mathematical Journal, Vol. 5. no. 4, pp 321-328, 2004.

[6] F. Ünlü, A Generalized us-Culture Job Scheduling for Forecasting Problems, International Mathematical Journal, Vol. 4, no. 4, pp 313-320, 2003.

[7] F. Ünlü, S. Sönmez, and Z. I. Ünlü, *Lineer Circular Convoluted I@I KBO Cluster Generating us-Crop,* International Mathematical Journal, Vol. 5. no. 4, pp 329-338, 2004.

[8] F. Ünlü, tuze-Channeled KBO of DPC, International Mathematical Journal, Vol. 5. no. 4, pp 339-346, 2004.

Received: December 7, 2005

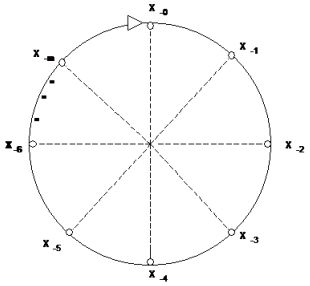

Figure 1: An m-ary negative FTD plasma memory X-type where each spike has a value in mod p.

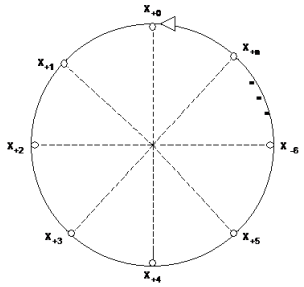

Figure 2: An n-ary positive FTD plasma memory X_+ type where each spike has a value in mod q.

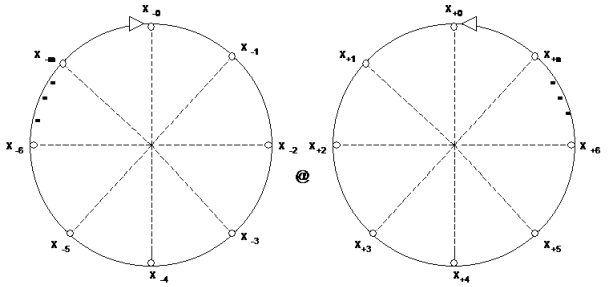

Figure 3: An (m, n)-ary (negative, positive) FTD plasma memory X_{-} @ X $_{+}$ type where each spike in X_{-} has a value in mod p and each spike in X_{+} has a value in mod q.

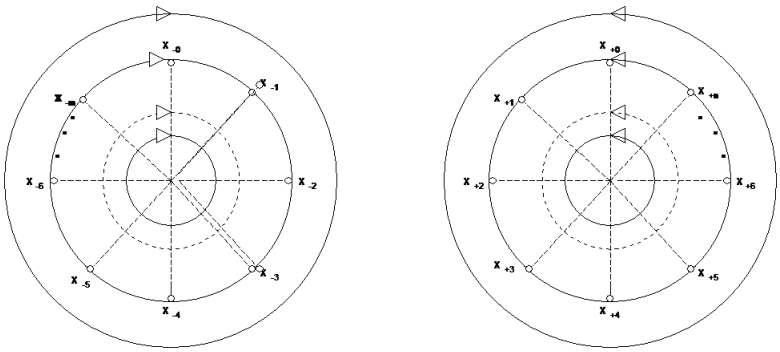

Figure 4: An us-cultural remote enforced point action generating tool for generating a remote I@I KBO cluster of remote I@I KBO clusters.