

TASIM GÜNLÜĞÜ A : TANITIM

Fevzi ÜNLÜ*

ÖZET

TASIM, yazarın İngilizce anlatımlarında “Tidy Automatic Sequential Information-processing Mechanism” söz dizisinin kısaltılması ile oluşturulmuştur. Türkçeye TASIM olarak girmiş bir kelimedir. O kendisi ile oluşturulmuş sonlu sayıda TASIM makinesinin--bilgisayarının “sonlu sayıda ki belleğinde var olan ham bilgilerden sonlu sayıda alt algoritmalarından oluşturulmuş algoritmalar aracılığında, yeni oluşturulan nasıl kullanılacağı belli, amaca veya amaçlara uygun biçimsel bilgiyi veya bilgileri tutumlu biçimde üretir veya türetir. Ürettiği veya türettiği yeni bilgi ve bilgileri sonlu sayıda belleklerden oluşmuş belleğe taşıma tekniklerini bulur ve yönetir. Yazarın bilimsel yaşamını yenidenlikli olarak etkilemiş bir çalışma alanıdır. Temelde, Bilgisayar Bilimlerini bir tümleşik BTBN--“Bilgi Tabanlı Bilgi Nesnesi” olarak görür. Daha genel olarak, bir BTBN bellek kümesinin içeriğinde mevcut biçimsel bilgi ve bilgilerden; 1. Doğruluk--Correctness, 2. Sağlamlık--Robustness, 3. Genişletilebilirlik--Extendibility, 4. Sıkıştırılabilirlik--Contractibility, 5. Yenidenlikli-kullanılabilirlik--Recursive reusability, 6. Erişilebilirlik--Acceptability, 7. Yeterlilik--Efficiency, 8. Taşınabilirlik--Portability, 9. Kanıtlanabilirlik--Verifiability, 10. Tümleştirilebilirlik--Integrity, 11. Kolay-kullanılabilirlik--Easy usability, 12. İçerik-işlenebilirlik--Interoperate ability, 13. Sadelik--Simplicity, 14. Çözülebilirlik--Decomposability, 15. Birleştirilebilirlik--Compos ability, 16. Anlaşılabilirlik--Understandability, 17. Korunabilirlik--Protect ability, 18. Süreklilik--Continuity, 19. Uzaktan-değiştirilebilirlik--Remote changeability, 20. Uzaktan-programlanabilirlik--Remote programmability, 21. Uzaktan-kontrol edilebilirlik--Remote controlability, ..., vs gibi işleç kavramlarını kullanarak gerçekleştirilen bir entegre işlem zinciri altında oluşturulan yeni BTBN’lerini bir başka BTBN kümesi içine çok iyi örgütlenmiş bildirişim tekniğini kullanarak taşır. Dahası, olup biteni çok basitleştirip anlaşılır biçime getirmek için bu çok yoğun ve derin BTBN taşıma işleminin adını taşırcaklama veya taşırcıklama--programlama

* Yaşar Üniversitesi, Fen-Edebiyat Fakültesi, Matematik Bölümü, Bornova, İzmir

olarak adlandırır. Bu yazıda, TASIMın anavatanında tutunabilmesi, çiçek açıp meyve verebilmesi için onun algılanmasına ışık tutulması istenmektedir. Yazar “TASIM Günlüğü A : Tanıtım” adlı bu yazı ile TASIMı çalışma gündemine getirmiştir. Tartışmaya açmıştır. Bunu izleyen başka yazılar sırada beklemektedir. Yetişmekte olan kuşağa yararlı olmasının dileğimizdir.

1.GİRİŞ

Geçen 60 yıl içinde gelişmiş toplumlar, bilimsel alanda, insan yaşamını çok değişik yönden etkileyen, yoğun ve derin kuramsal¹ bilim ve teknik geliştirme ve onları uygulama süreçlerine tanık olmuşlardır. Gelişmeler Bilgisayar Bilimlerinde yepyeni teknik ve uygulama yöntemleri oluşturmuştur. Ortaya konan sorunların çözümünde, bilgisayarın gerçek olarak katkısı büyüktür. Çözümlemesi uzun zaman isteyen karmaşık problemler, bilgisayar desteği sağlanarak birkaç dakikada sonuçlandırılabilir. Bu gün “Bilgisayar Bilimleri” adlı bir bilim dalı mevcuttur. İyi örgütlenmiş üniversitelerin Fen-Edebiyat Fakültelerinin öğretim programlarında, Bilgisayar Bilimleri Bölümü Öğretim Programı olarak yer almaktadır. Ülkemizde bu konu Bilgisayar Mühendisliği Bölümlerinin öğretim programları içine gömülmüştür. Bu nedenle yarının uygulamalarına ışık tutacak teorik bilgiler çalışılıp üretilememiştir. Onun için ülkemizde, bilgisayarla problem çözümlemelerinde tutumlu ve biçimsel bildirişim²--çok yönlü iletişim yapabilmek, Bilgisayar Bilimlerinin en önemli amaçlarından biridir. Bu yöntem değişik düzeylerde belli kurallarla geliştirilmiş programlama--taşırıcaklama³ veya taşırıcıklama dilleri ile yapılır. Programlama bir bilgi kümesinin BTBN--“Bilgi Tabanlı Bilgi Nesnesi” öğelerini, sonlu sayıda alt algoritmalarından oluşturulmuş algoritmalar aracılığında, bir başka kümenin içine taşır. Yani bir programlama dilidir.

Programlama dili, bilgisayar ile problem çözümlemesi yapılırken, bildirişim yapan kişi ile bilgisayar arasında bildirişim yapabilme yöntemlerini oluşturmada kullanılan bir biçimsel-ergin⁴ dildir. Tıpkı bir doğal dil gibi, belli kurallara uygun olarak doğar. Onun da bir bilgi-nesnel yaşamı vardır. Belli bilgi-nesnel yaşam süreçlerinden geçer. Aşamalar yaparak amacı doğrultusunda gelişir. Çoğu kez kullanışsız durumda ölür. Genel olarak, programlama dillerinin ömürleri doğal dillerinkinden daha kısadır. Bilgisayar Bilimlerinin son 60 yıl içindeki gelişimine paralel olarak, değişik bildirişim yapma gücünde çok ayrı amaçlara

yönelik birçok programlama dili tanımlanmıştır ve değişik sorunların çözümlerinde bilgisayarları programlamak için kullanılmıştır. Örneğin, Bilgisayar Bilimlerinin çocukluk döneminde, Sammet [53] yalnız A.B.D.'de geliştirilen yaklaşık olarak 129 değişik programlama dilinin irdelenmesini yapmıştır. Bunlardan 20'si ölü ya da hiç kullanılmayan dillerdir. 40 tanesi çok az kullanılmıştır. 50 tanesi özel amaçların gerçekleştirilmesinde kullanılırken, yalnız **FORTRAN, ALGOL, COBOL, SNOBOL, LISP, JAVA**, v.b. gibi programlama dilleri geniş uygulama alanları bulmuşlardır. Son gruba ait dillerin özellikleri için USA [57;58], Backus [30], Heising [39;40;41], Robins [51], Bottenburch [32] Baumann [31], Haynam [38], Bromberg [33], Cunningham [34], Saxon [54], Desautels [35], McCarthy [48;49], Halsted [36;37] v.b.'ne bakılmasında, okuyucu açısından yarar vardır.

Programlama dillerinin içeriğinde yer alan temel bağıntıların yapıları ayrıntılı olarak incelendiğinde, yapıbilim--syntax, anlambilim--semantics ve kılınbilim--pragmatics yönünden önemli olan ortak özelliklerin olduğu açık olarak görülür. Church ve Kleene [3], Kleene [9], ve Church [4] v.b.'nin verdikleri matematik yöntemleri ile hemen her programlama dilini yapıbilim yönünden etkilemişlerdir. Bu gerçeği Landin [45;46] ilk programlama dilinin tanımından on yıl sonra, ALGOL için göstermiştir. Yazar, 1962 – 1972, Texas Üniversitesi, Auistin, Texas'ın Bilgisayar Bilimleri Bölümünde doktora ve doktora üstü çalışmaları yaptığı yıllarda, Prof. Dr. N. M. Martin'in uyarıları ile aynı gerçeğin üstüne eğilerek ALGOL dışındaki programlama dilleri için bilimsel araştırmalar yapmıştır. Hükmün doğru olduğunu görmüştür. Yazara göre bu çok önemli olan bir yorumdur. Ünlü [59].

Bu yazıda, TASIM tanıtılmak istenmektedir. TASIM bir çok programlama dillerinden önce geliştirilmiştir. Programlama dillerini geniş ölçüde etkileyen yöntemlere dayalıdır. Biçimsel olarak özel bildirişimli--çok yönlü iletişim yapabilme özelliğine sahip ilk algoritma tanımlama dilidir. Bu biçimsel dili diğer biçimsel dillerden ayıran en önemli özellik her süreçte kendi kendisini yenileyebilen bir dil olmasıdır. Bir Türk bilim adamı tarafından geliştirilmiştir. Güdülen amaç, us⁵--mantık değerleri ile tanımlanabilen TASIM-işlevlerini⁶ yapıbilim ile bir bildirişimli algoritma gibi türeterek, gerektiğinde bu algoritmaların değerlerini bir TASIM-bilgisayarında, bir TASIM dili ile programlayarak, sonuç üretecek biçimde düzenlemektir. Sözü edilen algoritmalar genelleştirilmiş TASIMlar aracılığında diğer programlama dillerine kolaylıkla taşınabilir olacaktır. Bunun için gerekli olan kılınbilim—pragmatics geliştirilmiştir. TASIMla ortaya konan yöntemlerin anlam yönünde aydınlığa

kavuşturulması için bu biçimsel algoritma türlerinin yapıbilim yönünden dayandırıldığı ilkelerin tarihi gelişimini vermektten yarar umuyoruz.

Skolem [23] ve Schönfinkel [32] us matematiğinin temel ilkelerini araştırdılar. Curry [5] bazı değişkenler tanımlayarak birbirleriyle değiştirebilme özelliklerini incelemiştir. Church [1;2] us matematiğinin belitlerini--aksiyomlarını ortaya koymuştur. Kleene [6;7;] daha önce Curry [5] ve Church [1;2]'nin verdiği sonuçları değerlendirerek pozitif tam sayılara uygulamıştır. Rosser [21] değişkensiz bir us matematiğinin kurulup kurulamama nedenlerini araştırmış ve eleştirmiştir. Kleene [8;9;10] bir λ - tanımlanabilirliği kavramını gündeme getirmiş ve yinelemeli⁷--recursive bağıntı geliştirme yönteminin ufkunu açmıştır. Turing [24;25;26] sayıların λ - tanımlanabilirliği ile tanımlanmasını ve dar anlamda biçimsel TASIMlama yöntemlerini incelemiştir. Church ve Kleene [3] belli λ -bağıntıları ile sıralama sayılarını tanımlamıştır. Church [4] λ - dönüşümlerinin dayandığı temel ilkeleri incelemiştir. Nelson [12;14], Robinson, R. M. [19;20], Post [14;15;16] ve Robinson, J. [17;18] bağıntılar kuramına biçimleştirme ve erginleştirme yönünde söz etmeye değer yenilikler getirmişlerdir. Günümüze değin sayısız matematikçiler ve bilgi işlem bilimcileri aynı konuya katkıda bulunmuşlardır. Rogers [52] ve Hermes [42]'yi görünüz.

İkinci dünya savaşını izleyen 20 yıl içinde, bilgisayar bilimleri ve bilgisayar yapıcılığını destekleyen bilgisayar mühendisliği teknolojileri gelişti. Çeşitli güçte bilgi işleme becerisine sahip bilgisayarlar yapılmıştır. Bilgisayarlar ile biçimsel bildirişim yapabilme önem kazanmıştır. Bilgisayar Bilimleri doğmuştur. UNIVAC [56], Holberton [44], Speeding [55], Backus [28;29], Adam [27], Laning [47] ve Preliminary [50]'yi görünüz. İlk yayınlanan programlama dilinden günümüze değin, bilgisayarlarla bildirişim yapma uğruna, çok ince ve uzun yolda yürünmüş ve değişik süreçlerden geçilerek çok büyük aşamalar katedilmiştir. Elde edilen sonuçlar kayıtlara geçilmiştir ve mevcuttur. Çok sayıda, yapıbilimi, anlambilimi ve kılınbilimi zengin dilbilim-gramer ile denetlenebilen programlama dili tasarımı ve gerçekleştirilmesi yapılmıştır. Biz ancak önemli gördüğümüz birkaç programlama dilinin doğup gelişmesi ile ilgili makale ve kitaptan bazılarının listesini kaynak olarak verdik. Amacımıza ulaşmada bu bilimsel kaynakların gerek sayıca, gerekse bilgi yönünde yeterli olduğu inancındayız.

“TASIM Günlüğü A: Tanıtım” adlı bu BTBN beş bölümden oluşmaktadır. Birinci bölüm giriştir. İkinci bölümde biçimsel TASIM BTBN’ni oluşturan TASIM öğeler tanıtılmıştır. Üçüncü bölümde TASIM dilbilimi, TASIM türleri ve örnekleri sergilenmiştir. Dördüncü bölümde yazının amacı ile ilgili eleştiri yapılmıştır. Beşinci bölümde ulamalar verilmiştir. İkinci bölüm dışındaki bölümler çok kısa bölümlerdir. İkinci bölüm 3 kesime ayrılmıştır. Birinci kesimde TASIMın yapı taşları tanımlanmıştır. İkinci kesimde TASIMın TIQ: “TASIM Infirmation Quantifier” ve TOQ: “TASIM Organized-information Quantifier” türleri verilmiştir. Üçüncü kesimde TASIM işlevleri ve TASIM işleyici⁸-işleçleri, yapıbilim ve anlambilim yolu ile örneklenmiştir. Bu yazının hazırlanmasında, anlaşılması kolay olan sözlük ve deyimlerin kullanılmasına özen gösterilmiştir. Yazı içeriğinde, özel anlamda kullanılmış bazı sözcüklerin bu anlamları ulama dediğimiz ekler aracılığında açılmıştır. Ulamalar her özel sözcüğün ya da deyimın sağ üst köşesine konan sıramak⁹ sayıları ile tanımlanmıştır. Yapılan ulamaların listesi beşinci bölümde en son bölüm olarak kaynaklar listesinden önce yer almaktadır.

TASIMın tanıtımı geliştirilirken, tanımlar, kural ve kuramlar x.x.x. şeklinde üçlü sıralama çetelesi--kodu ile sıraya konulmuşlardır. Bu çetelede, birinci x söz konusu ögenin bulunduğu bölümü, ikinci x bu bölüm içindeki ait olduğu kesimi ve üçüncü x’de kesim içindeki sırasını göstermektedir. Kaynaklar, [] ayraç çifti içine alınmış sıralama sayıları ile sıraya konulmuşlardır. Bir kaynaktan söz edildiğinde, kaynak listesinden bulmayı kolaylaştırmak için, bu sıra sayıları kaynağın yazar adından sonra [] ayraç çifti içinde verilmiştir. Aynı anda bir yazarın birden çok yayını söz konusu ise, bu yayınların kaynak listesindeki sıra sayıları aynı [] ayraç çifti içinde noktalı virgüllerle ayrılarak verilmiştir. Bir kaynak belli bir kurum adına, bir yazar ya da yazarlar grubu çevresi ile hazırlanmış ise, kaynağın adındaki ilk sözcük yazar adı gibi kullanılmıştır.

Bir bilimsel yazıda okuyucu ile tutumlu bildirişim yapmak her zaman her yazarın öncelikli amacıdır. Konuları işlerken, tanım, kural ve kuramlardan sonra örnekler verilmiştir. Bu örneklerin incelenmesi ile soyut görünen tanım, kural ve kuramların sanıldığı gibi karmaşık olmadığı görülür. Yani kolay ve iyi anlaşılabilirliği inancındayız.

TASIM günlüğünü etkileyen kaynak bilginin içeriği ve uygulama alanı geniştir. Biz bu geniş alanı biçimsel tanımlarla daraltmaya çalıştık.

2.TASIM BTBN

Bu bölümde TASIM BTBN tanıtılacaktır.

2.1 TASIMın Yapı Taşları :

TASIM soyut anlamda bir biçimsel dildir. Bu dili algılayan bir TASIM makinesi--bilgisayarı vardır. Yani TASIM ile TASIM-makinesi birlikte bir iki durumlu sistem gibi çalışır. Birinci durumda makine ikinci durumda bir programlama dili anlamı taşıyarak işlev görevini gerçekleştirir. Çünkü TASIM, TASIM makinesi ile onun duyarlı olduğu dil, dil derleyicisini ve dil yorumlayıcısını bütünleştirmiştir. Bunun için, biçimsel olarak, onu ortaya koyacak dilbilim--gramer yapısının bilinmesi ve incelenmesi zorunluğudur. Görüldüğü üzere TASIM, programlama dilinden daha kapsamlı bir matematiksel yapıdır. Bir soyut programlama dili tarafından üretilen makine algoritmalarının kapanışını tanımlayan biçimsel mekanizma ile yine bu biçimsel mekanizmanın oluşturduğu programlama dili çatısı gibi düşünülebilir.

Tanım 2.1.1 : TASIM, bir soyut programlama dili tarafından üretilen algoritmaların kapanışını tanımlayan mekanizma ile bu mekanizmanın oluşturduğu programlama dili çatısı gibi düşünüldüğünde, bu programlama dilinin çatısı içinde gözlenen $[1] +, -, *, \div, \wedge, \vee, \square, =, \rightarrow,$ ve \leftarrow simgeleri--işaretleri veya imleri; toplama, çıkarma, çarpma, bölme, ve, veya, değil, eşitlik, indirgeme ve değer bağlama işlemlerinin işleçlerini temsil eder. Simge, işaret ve im aynı anlamlı kelimelerdir. [2] $\lambda, \emptyset,)$ ve $($ simgelerine katkısız simgeler denir. [3] Bir matematiksel TASIM içindeki temel yapı taşlarından olan, TASIM ve TASIMlama süreçlerinde en az iki farklı biçimsel bilgiyi değer olarak alabilen, her BTBN'ne değişken denir. Her değişken BTBN'nin bir değişken veya durağan adı vardır. TASIM BTBN'leri, diğer TASIM BTBN'lerini sonlu sayıda farklı dilbilimsel özellik aracılığında tanımlar. Eğer bir TASIM BTBN'nin bir TASIM-işlem sürecinde en az bir özelliği değişiyor ise söz konusu BTBN'si değişkendir. Aksi halde o TASIM BTBN'si bir durağan--sabit TASIM BTBN'dir. Değişken TASIM BTBN'leri küçük sesimek--harf ve küçük sesimekleri izleyen sayımak--

rakam imdizisi--string ile gösterilir veya adlandırılır. İmdizi, işaretdizi, semboldizi ve string aynı anlamlı kelimelerdir. Burada önemle belirtiriz. [4] Bir TASİM içinde kullanılan simgeler kümesine TASİM'in $a b e c e^{10}$ kümesi veya daha kısa olarak $a b e c e s i$ --alfabesi denir. Kümeler büyük sesimeklerle adlandırılır. [5] Her işaret, sembol veya simgeye kısaca bir im denir. İm TASİM bilgi işlem süreçlerinde alt parçalarına ayrılamaz. Böylece $+$, a , veya \wedge bir imdir. İmlerden oluşan anlamlı veya anlamsız her kelime bir imdizi oluşturur. Yani "arı" bir imdizidir. Hiç im içermeyen imdiziye boş imdizi denir. Boş imdizi "ε" ile temsil edilecektir. Amaç işaret, simge, sembol ve im kelimelerinden hangisinin kullanılması daha uygun olur sorusunu yanıtlamak olursa, cevap im olabilir. Çünkü tutumlu bildirişim yapmak önemli bir kavramdır. Bu anlamda, im kelimesinin kullanılmasının uygun olacağını önermek doğru olur. Aynı anlamlı kelimelerin en kısa çetelesine--koduna o sahiptir. Yani daha tutumlu bildirişim yapmamızı sağlar. Ama amacımız anlambilimsel bildirişim yapmaksa bu kelimelerden uygun görülen bir başkası kullanılabilir.

Biz Türkçe $a b e c e$ --alfabe'sine $\lambda, \emptyset,), (, +, -, *, \div, \wedge, \vee, \Gamma, =, \rightarrow,$ ve \leftarrow simgelerinin ve $0,1,2,3,4,5,6,7,8,9$ sayımaklarının katılması ile elde edilen yeni simgeler kümesini TASİM'in abecesi olarak kullanacağız ve genellikle bir büyük harf ile adlandıracağız. Bu alfabeğe yeni simgeler ekleyerek, alfabenin genişletilmesinin yapılabileceği açıktır.

Tanım 2.1.2 : TASİM içeriğinde: 1. Bir işlemin yaptığı işi yüklenerek yürüten bir işleyici vardır. 2. Üzerine işlem uygulanan her öğeye bir işlenen denir. 3. İşlenenlerden oluşan bir kümeye işlenenler kümesi denir. 4. İki işlenenler kümesinin öğeleri arasında ilişki kuran her kurala bir TASİM-bağıntısı veya kısaca TASİM denir. 5. Fonksiyon özelliğine sahip her TASİM-bağıntısına, bir TASİM-işlevi denir. Altı çizilmiş her simge ve sözcük bir TASİM- işlevinin adıdır yada biçimsel TASİM-çatı sistemi içinde özel bir anlam taşır.

Örnek 2.1.1 : Eğer R simgesi bir TASİM-işlemi olarak tanımlanırsa $b \leftarrow [Ra]$ gösterilişinde: 1. a öğesine bir işlenen, 2. a gibi öğelerden oluşan bir A kümesine işlenenler kümesi, ve 3. b gibi öğelerin oluşturduğu bir kümenin adı B ise A ve B kümelerinin öğeleri arasında fonksiyonel ilişki kuran R kuralına bir TASİM-işlevi denir. TASİM'da her TASİM-işleyeni bir TASİM-işlevi olarak tanımlanacaktır.

TASIM-işlevinin değerleri işlev komutları ile bulunur. İşlev komutlarının anlamı aşağıda verilen tanım ve örnekle aydınlatılmaya çalışılmıştır.

Tanım 2.1.3: 1. A^+ TASIM-alfabesi olsun. Eğer \underline{f} simgesi kuralları verilmiş bir TASIM-işlevinin adı ve A^+ 'da A^+ 'dan elde edilen adlandırmaya uygun imdizilerin kümesi olarak seçilirse, sonlu sayıda i doğal sayı indeksi için, $a_i \in A$ olmak üzere, $b_1 \leftarrow (\dots((\underline{f} @a_1)@a_2)\dots @a_n)$, $b_2 \leftarrow (\dots((\underline{f} a_1 a_2)@a_3)\dots @a_n)\dots$ $b_{n-1} \leftarrow ((\underline{f} a_1 a_1 \dots a_{n-1}) @a_n)$, $b_n \leftarrow (\underline{f} a_1 a_2 \dots a_n) = \underline{f} a_1 a_2 \dots a_n$ TASIM-işlevi komutlarıdır. Burada “@” bir işleçtir. TASIMda bu @-işleci bildirişim-çekimi veya bildirişimsel çekim işleci olarak adlandırılır. Burada izlenen, TASIM-işlev komutlarına sıra ile \underline{f} 'nin $(n-1)$, $(n-2)$,..., $1,0$ değişkenli işlev değerleri denir. 2. \underline{f} ve \underline{g} birbirinden değişik iki TASIM-işlevinin adları olsun. $(\underline{f}@ A)$ ve $(\underline{g}@ A)$ birbirine eşit ise bu iki TASIM-işlevine A için birbiri üzerine yayılmıştır denir. 3. Eğer $(\underline{f} @A)$ ve $(\underline{g}@ A)$ TASIM-işlev komutlarında A işlenenler kümesi \underline{f} ve \underline{g} 'nin tanım alanına ait değil ise $(\underline{f} @A)$ ve $(\underline{g} @A)$ komutları A işlenen kümesi için tanımlanmamıştır denir. Bir TASIM-işlevinin uygulanabileceği tanımlı işlenenler kümesine, bu TASIM-işlevinin tanım alanı denir. Bir TASIM-işlevinin tanım alanına uygulanması ile elde edilen işlev değerleri kümesine bu TASIM-işlevinin işlev değerleri alanı denir.

Örnek 2.1.2: Eğer x_1, x_2 ve x_3 gerçek bağımsız değişken adları için

$$\underline{f} \leftarrow x_1 + x_2 + x_3, x_1, x_2, x_3 \geq 0;$$

$$\underline{g} \leftarrow |x_1| + |x_2| + |x_3|, -\infty \leq x_1, x_2, x_3 \leq +\infty \text{ ve } A = \{a_1, a_2 = 5, a_3 = 3\} \text{ ise;}$$

$$[1] b_2 \leftarrow (((\underline{f} @1) @x_2) @x_3) = 1 + x_2 + x_3,$$

$$b'_2 \leftarrow (((\underline{g} @ 1) @ x_2) @ x_3) = 1 + |x_1| + |x_2| = 1 + x_2 + x_3$$

$$b_1 \leftarrow ((\underline{f} @15) @x_3) = 6 + |x_3| = 6 + x_3,$$

$$b'_1 \leftarrow ((\underline{g} @15) @x_3) = 6 + |x_3| = 6 + x_3$$

$$b_0 \leftarrow (\underline{f} @153) = f 153 = 9$$

$$b'_0 \leftarrow (\underline{g} @153) = g 153 = 9$$

elde ederiz.

Tanım 2. 1. 4: 1. A bir TASIM alfabesi olsun. A içeriğinde olan her öğeye bir simge, sembol, işaret veya im denir. TASIMda, TIQ diye adlandırılır. 2. Bir TASIM alfabesini öğelerinden TASIM kuralları ile elde edilen, anlamlı veya anlamsız her kelimeye veya biçime

bir Simgedizi, işaretdizi, seboldizi veya imdizi denir. TASIMda, TOQ diye adlandırılır. 3. Ögeleri arasında en az tanım alanı bilinen bir değişkeni bulunduran her TIQ veya TOQ'a bir TASIM denir. 4. Özel anlamı olan her TIQ veya TOQ'a bir s ö z c ü k denir. İlaveten: a. TIQ İngilizce “ TASIM Information Quantifier” deyiminde yer alan sözcüklerin baş harfleri ile tanımlanmış bir yeni kelimedir. b. TOQ İngilizce “TASIM Organized-integrated-information Quantifier” deyiminde yer alan kelimelerin baş harfleri ile oluşturulmuş bir yeni kelimedir. Alt yapılar ayrılabilen her TASIM bir TIQ-TASIMdır. Alt yapılarına ayrılabilen her TASIM bir TOQ-TASIMdır. Her TOQ TASIM TOQ ve TIQ TASIMlardan oluşturulabilir.

Örnek 2. 1. 3: 1. Kabul edelim ki a, d, l, r ve x TASIM alfabesinden yer alan im veya imdiziler olsun. Bunlardan x veya a bir TIQ'dır. 2. ax im-dizisi bir TOQ'dur. 3. Eğer x tanım alanı bilinen bir değişken ise, ax TOQ'u x değişkenini içine aldığından, bir TASIMdır. 4. “ardı” TASIMda özel anlam varsa bir sözcüktür.

“λ” simgesi TASIMın çekirdeğini teşkil eder. Şimdi λ simgesine bir anlam vereceğiz ve bu simgeyi kullanarak TASIM-işlevlerini açık olarak tanımlayacağız.

Tanım 2. 1. 5 : 1. TASIM içeriğinde, $\lambda x.G$ biçimsel gösterilişine bir TASIM-işlevi denir. Bu genel olarak, standart matematik derslerinde, adı gelişi güzel seçilebilen, $*[x]=G$ fonksiyonuna karşılık gelir. Eğer fonksiyon adını f olarak seçersek, $f[x] = G$ olur. Yani “ $f[x] = G$ ” ile “ $f = \lambda x.G$ ” aynı bilgiyi iki ayrı dilde ifade etmiş olur. Birincisi standart matematik dilinde bir ifadedir. İkincisi TASIM dilinde bir ifadedir. İki ifadenin de anlamı aynıdır. 2. TASIM-işlevi $\lambda x.G$ 'nin içinde bulunan λx 'e bir x-denetçisi veya denetleyicisi ve G' ye de bir gövde bağıntısı denir. 3. Bir x-denetçisi verildiğinde λ'ya yönetici, x'e bir bağlayıcı değişken denir. 4. Verilen herhangi bir anda, G içinde G" ve G' gibi iki gövde bağıntısı varsa, öyle ki $G' \leftarrow \lambda x.G''$ ve G" gövde-bağıntısı x' i içine alıyorsa, o zaman x değişkenine G içinde bir bağlı değişken veya bağlı görünen değişken denir. 5. Eğer x değişkeni G içinde bağlı değişken olarak bulunmuyorsa, o zaman x değişkenine G içinde bir bağımsız--erkin veya serbest değişken denir. Verilen herhangi bir anda eğer G gövde-bağıntısı x değişkenini bir bağlı değişken olarak içinde bulundurursa, x değişkenine G içinde bir bağlı görünen değişken denir. 7. Her hangi bir anda, eğer G gövde bağıntısı x değişkenini bir bağımsız değişken olarak içinde bulundurursa, x değişkenine G içinde bir bağımsız olarak görünen değişken denir. “ $f = \lambda x.G$ ” içeriğinde görülen “.” işaretine x-denetçisi ile gövde bağıntısını ayıran ayırtman işleç

simgesi denir.

Örnek 2. 1. 5a: 1. $\lambda y.(y@ \lambda x.(y@x))$ bir TASIM-işlevidir. 2. Bu TASIM-işlevinde λy bir TASIM-deneticisidir ve $\underline{G} \leftarrow (y@ \lambda x.(y@x))$ bir gövde-bağıntısıdır. 3. TASIM-deneticisi λy içinde λ yönetici ve y 'de bir bağlayıcı değişkendir. 4. Verilen bir anda, \underline{G} gövde-bağıntısında y değişkeni bir bağımsız yada serbest değişkendir. 5. x değişkeni \underline{G} içinde bir bağlı görünen değişkendir.

Örnek 2. 1. 5b: 1. $\lambda x.[\lambda x.[x@y]@ \lambda y.\lambda z.[x@y]]$ bir λ -işlevidir. 2. Bu λ -işlevinde λx bir λ -işleyicisidir. 3. λ -işleyicisi λx içinde λ yönetici ve x bir bağlayıcı değişkendir. 4. Verilen bir anda, $\underline{G} \leftarrow (\lambda x.(x@y)@ \lambda y.\lambda z.(x@y))$ gövde-bağıntısında, x değişkeni bir bağlı görünen değişkendir. 5. Aynı gövde-bağıntısında, ilk y serbest görünen değişkendir. 6. İkinci y bir bağlı görünen değişkendir. 7. $\underline{G}^1 \leftarrow \lambda y.\lambda z.(x@y) \subset \underline{G}$ içinde x bir bağımsız görünen değişkendir. Burada kullanılan, erkin, serbest ve bağımsız aynı anlamlı sözcüklerdir. TASIMda, λ katkısız simgesi ile gösterilen bir tek yönetici adı vardır.

3.KÖK TASIM DİLBİLİMİ, KÖK TASIM TÜRLER VE ÖRNEKLER

Bu kesimde TASIM dilbilimi ve TASIM türleri tanıtılacaktır. Örnekler verilecektir.

Kural 1. 3. 1: TASIMda bir TASIMın türetimi aşağıdaki kurallarla yapılır.

K0: Bir alfabe sabit ve değişken adlarından oluşur. Bu sabit ve değişken adlarına im denir. İmdizi yapısındaki imler \langle ve \rangle ayırtmanı içine yazılarak im olduğu damgalanır.

K1: $[a]$ Her sabit kendi başına bir TASIMdır. $[b]$ Her değişken kendi başına bir TASIMdır.

K2: \underline{G} bir TASIM ve x bir değişken ise $\lambda x.\underline{G}$ bir TASIMdır. TASIM işlevi olarak adlandırılır.

K3: \underline{U} ve \underline{T} iki TASIM ise $(\underline{U} @ \underline{T})$ bir TASIMdır. TASIM komutu olarak adlandırılır. TASIM komutunda \underline{U} 'ya işleyici veya işleç--operator ve \underline{T} 'ye de işlenen veya işleçdensel--operand denir.

K4: Bir TASIM yalnız yukarıdaki üç kuralla türetilebilir veya üretilebilir.

Örnek 1. 3. 1: Burada TASIM türetim veya üretim örnekler verilecektir.

[Ö1]: Eğer y bir değişken ise y bir TASIM'dır. K1b ile elde edilir.

[Ö2]: Eğer x ve y iki değişken ise $\lambda y.x$ bir TASIM işlevidir. λ -yöneticidir. TASIM işlevi $\lambda y.x$ içinde λy , y -denetcisidir ve $\underline{G} \leftarrow x$ onun gövde bağıntısıdır. K1b ve K2 ile elde edilir.

[Ö3]: Eğer x ve y iki değişken ise, $(x@z)$ bir TASIM komutudur. İşleyici $\underline{U} \leftarrow x$ ve işlenen $\underline{T} \leftarrow z$ 'den oluşmuştur. K1, K1 ve K3 ile elde edilir.

[Ö4]: $(\lambda y.z@x)$ bir TASIM komutudur. İşleyici $\underline{U} \leftarrow \lambda y.z$ ve işlenen $\underline{T} \leftarrow x$ 'den oluşmuştur. K1, K2, K1 ve K3 ile elde edilir.

[Ö5]: $\lambda z.(y@x)$ bir TASIM işlevidir. TASIM denetçisi λz ile gövde-bağıntısı $\underline{G} \leftarrow (y@x)$ 'den oluşmuştur. K1, K1, K3 ve K2 ile elde edilir.

[Ö6]: $(\lambda x.z@ \lambda z.y)$ bir TASIM komutudur. İşleyici $\underline{U} \leftarrow \lambda x.z$ ve işlenen $\underline{T} \leftarrow \lambda z.y$ 'den oluşmuştur. K1, K2, K1, K2, ve K3 ile elde edilir.

[Ö7]: $\lambda x.(z@(y@x))$ bir TASIM işlevidir. TASIM x -denetçisi λx ile gövde bağıntısı $\underline{G} \leftarrow (z@(y@x))$ 'den oluşmuştur. K1, K1, K3, K1, K3 ve K2 ile elde edilir.

[Ö8]: $\lambda z. \lambda y.x$ bir TASIM işlevidir. TASIM z -denetçisi λz ile onun gövde-bağıntısı $\underline{G} \leftarrow \lambda y.x$ 'den oluşmuştur. K1, K2, ve K2 ile elde edilir.

[Ö9]: $\lambda z. \lambda x.(z@ \lambda z. \lambda x.x)$ bir TASIM'dır. TASIM z -denetçisi λz ile gövde-bağıntısı $\underline{G} \leftarrow \lambda x.[z@ \lambda z. \lambda x.x]$ 'den oluşmuştur. K1, K2, K2, K1, K3, K2 ve K2 ile elde edilir.

4.ELEŞTİRİ

Burada bulguları belirtip, irdeleme yapacağız ve önerilerde bulunacağız.

1. “TASIM günlüğü A: Tanıtım” adı altında, yazar tarafından 1970 yılların başında geliştirilen TASIM'ın biçimsel kısa tanıtımı yapılmıştır. Bu yazı TASIM tanıtımının ilk ayağını oluşturmaktadır. Bunun için “TASIM Günlüğü A: Tanıtım” diye adlandırılmıştır.
2. TASIM, yazarın İngilizce anlatımlarında “Tidy Automatic Sequential Information-processing Mechanism” söz dizisinde mevcut kelimelerin baş harflerinden oluşturulmuştur. Dilimize TASIM olarak girmiş bir kelimedir. Anlamı “sonlu sayıda bellekte var olan ham bilgilerden sonlu sayıda alt algoritmalarından oluşturulmuş algoritmaların kullanımı ile yeni oluşturulan, kullanımı amaca uygun biçimsel-ergin veya formal bilgileri tutumlu biçimde başka sonlu sayıda beleklerin amacı

doğrultusunda üreterek veya türeterek, o belleklere taşıma tekniklerinin kapanışını” içerir.

3. T a s ı m l a m a sözcüğü doğruluğu kabul edilen, bir ya da birden çok BTBN’inden yeni BTBN’leri türetme veya üretme makineleri ile onları programlayan dil çatısı anlamında kullanılmıştır. Üretilen BTBN’lerini makine ve dil denklik sınıflarına ayırarak, elde edilen makine denklik sınıfının elde edilen dil sınıfı aracılığıyla programlayarak problem çözen kapanış eylemi bütünlüğüne verilen addır.
4. Yazarın bilimsel yaşamını yenidenlikli olarak etkilemiş bir çalışma alanıdır. Temelde, Bilgisayar Bilimlerini bir tümleşik BTBN--“Bilgi Tabanlı Bilgi Nesnesi” olarak görür.
 - 01) Doğruluk--Correctness,
 - 02) Sağlamlık--Robustness,
 - 03) Genişletilebilirlik--Extendibility,
 - 04) Sıkıştırılabilirlik--Contractibility,
 - 05)Yenidenlikli-kullanılabilirlik--Recursive reusability,
 - 06) Erişilebilirlik--Acceptability,
 - 07) Yeterlilik--Efficiency,
 - 08) Taşınabilirlik--Portability,
 - 09) Kanıtlanabilirlik--Verifiability,
 - 10) Tümleştirilebilirlik--Integrity,
 - 11) Kolay-kullanılabilirlik--Easy usability,
 - 12) İçerik-işlenebilirlik--Interoperate ability,
 - 13)Sadelik--Simplicity,
 - 14) Çözültülebilirlik--Decomposability,
 - 15) Birleştirilebilirlik--Composability,
 - 16) Anlaşılabilirlik--Understandability,
 - 17) Korunabilirlik--Protectability,
 - 18) Süreklilik--Continuity,
 - 19) Uzaktan-değiştirilebilirlik--Remote changeability,
 - 20) Uzaktan-programlanabilirlik--Remote programmability,
 - 21) Uzaktan-kontrol edilebilirlik--Remote controlability, ..., vs

gibi kavramları kullanarak gerçekleştirilen bir entegre doğrusal işlem-zinciri altında oluşturulan yeni BTBN'lerini bir başka BTBN kümesi içine taşır. Bu BTBN taşıma işleminin adını programlama--taşırıklama veya taşırıklama olarak adlandırır. Bu yazıda, TASIMın anavatanında tutunabilmesi, çiçek açıp yaşama destek olması yönündeden destek istenmektedir. istenmiştir.

5. Yazar "TASIM Günlüğü A: Tanıtım" adlı bu yazı ile, TASIMı gündeme getirmiştir. Tartışmaya açmıştır. Yetişmekte olan kuşağa yararlı olmasını dilemektedir.

5. ULAMALAR

01. K u r a m s a l sözcüğü "teorik" sözcüğü anlamında kullanılmıştır.
02. B i l d i r i ş i m sözcüğü dar anlamda Türkçede "muhabere--iletişim" sözcüğü yerine; geniş anlamda İngilizcedeki " communication" sözcüğü anlamında kullanılmıştır.
03. T a ş ı r c a k l a m a sözcüğü "programlama" sözcüğü yerine kullanılmıştır.
04. E r g i n sözcüğü "matematik-yöntemleriyle tanımlanabilen" anlamında kullanılmıştır. Bu İngilizcedeki "formal" sözcüğünün de anlamını verir.
05. U s sözcüğünü "mantık" sözcüğü anlamında kullanılmıştır. .
06. İ ş l e v sözcüğü "fonksiyon" sözcüğü anlamında kullanılmıştır.
07. Y i n e l e m e l i sözcüğünü "kendini kendisiyle tanımlama" anlamında kullanılmıştır.. İngilizcedeki "recursive" sözcüğüne karşılık tutulmuştur.
08. İ ş l e y i c i sözcüğü İngilizcedeki "operator" sözcüğüne karşılık olarak kullanılmıştır.
09. S ı r a m a k sözcüğünü "indis" sözcüğü anlamında kullanılmıştır.
10. A b e c e sözcüğünü "alfabe" sözcüğü anlamında kullanılmıştır.

KAYNAKLAR

A. Matematik Yöntemleriyle İlgili Kaynaklar

- [1]. Church, A. : "A Set of Postulates for the Foundation of Logic", Annals of Mathematics, Seri 2, Vol. 33, 1932, pp 346-366.
- [2]. ---, -. : "A Set of Postulates for the Foundation of Logic"[Second paper], Annals of Mathematics, Seri 2, Vol. 34, 1933, pp 839-864.

- [3]. ---, -. and Kleene, S.C.: "Formal Definitions in the Theory of Ordinal Numbers",
Fundamenta Mathematica Society, Vol. 39, 1936, pp 11-21.
- [4]. ---, -.: "The Calculi of Lambda-Conversion", Princeton University Press, 1941. [Reprinted
by Kraus Reprint Corporation, New York, 1965].
- [5]. Curry, H. B.: "An analysis of logical substitution", American Journal of Mathematics.
Vol. 51, 1929, pp 363-384.
- [6]. Kleene, S. C.: "Proof by Cases in Formal logic", Annals of Mathematics, Ser. 2, Vol. 35,
1934, pp 529-544.
- [7]. ---, -.: "A Theory of Positive Integers in Formal Logic", American Journal of Mathematics,
Vol. 57, 1935, pp 153-244.
- [8]. ---, -.: "General Recursive Functions of Natural Numbers", Mathematische Annalen, Vol.
112, 1936, pp 727-742.
- [9]. ---, -.: " λ -Definability and Recursiveness", Duke Mathematical Journal, Vol. 2, pp 340-
353.
- [10]. ---, -.: "A Note on Recursive Functions", Bulletin of the American Mathematical
Society,
Vol. 42, 1936, pp. 544-546.
- [11]. ---, -.: "On Notation for Ordinal Numbers", The Journal of Symbolic Logic", Vol. 3,
1938, pp 150-155.
- [12]. Nelson, D.: "Recursive Functions and Intuitionistic Number Theory", Transactions of the
American Mathematical Society, Vol. 61, 1947, pp 307-368.
- [13]. ---, -.: "Constructible Falsity", The Journal of Symbolic Logic, Vol. 14, 1949, pp 16-26
- [14]. ---, -.: "Formal Reductions of the General Combinatorial Decision Problem", American
Journal of Mathematics, Vol. 65, 1943, pp 197-215.
- [15]. ---, -.: "Recursively Enumerable Sets of Positive Integers and Their Decision Problems",
Bulletin of the American Mathematical Society, Vol. 50, 1944, 284-316.
- [16]. ---, -.: "Recursive Unsolvability of a Problem of Thue", The Journal of Symbolic Logic,
Vol. 11, 1949, pp. 1-11.
- [17]. Robinson, J.: "Definability and Decision Problems in Arithmetic", The Journal of
Symbolic Logic, Vol. 14, 1949, pp 98-114.
- [18]. ---, -.: "General Recursive Functions" Proc. American Mathematical Society, Vol. 1,
1950, pp 703-518.
- [19]. Robinson, R.M.: "Primitive Recursive Functions", Bulletin of the American

- Mathematical Society, Vol.53, 1947, pp 925-942.
- [20].- --, -.:”Recursion and Double Recursion,”Bulletin of the American Mathematical Society, Vol. 54, 1948, pp 987-993.
- [21]. ---, -.:”A mathematical Logic Without Variables,”Annals of Mathematics, Ser, 2, Vol. 36, 1935, pp 121-150 ve Duke Mathematical Jurnal, Vol. 1, 1935, pp328-355.
- [22]. Schönfinkel, M.:”Über die Bausteine der methematischen Logic,”Mathematicshe Anelen, Vol. 92, 1942, pp 305-306.
- [23]. Skolem, T.:”Begründung der Elementaren Arithmetik Durch die Recurrierende Denkweise ohne Anwendung Scheinbarer Veranderlichen mit Unendlichem Ausdehnungsbereich, “Skrifter Utgit av Videnskapsselskapet i Kristiania, I. Mathematiks-Naturvidenskabelig Klasse, 1923, No: 6.
- [24]. Turing, A.M.:”On Computable Numbers, with an Application to Entscheidungsproblem,”Proc. London Mathematical Society, Ser. 2, Vol. 42, 1936, pp 230-265.
- [25]. ---, -.:”Computability and λ -definability, The Journal of Symbolic Logic, Vol. 2, 1937, pp 153-163.
- [26]. ---, -.:”System of Logic Based on Ordinals,”Proc. London Mathematic Society, Ser. 2, Vol. 45, 1939, pp 161-228.

B. Biçimsel Diller İle İlgili Kaynaklar

- [27]. Adams,C.W. and Laining,J.H.Jr.:”The M.I.T.Systems of Automatic Programming Systems,”Symposium on Automatic Programming for Digital Computers,Office of Naval Research,Department of the Navy; Washington;D.C.,1954,pp 40-68.
- [28]. Backus,J.W.and Herrick,H.:”IBM 701 Speedcoding and Other Automatic Programming Systems,” Office of Naval Research, Department of the Navy, Washington,D.C.,1954,pp 106-145.
- [29]. _____,_.:”The IBM Speeding System,”J.ACM,Vol.1,No:1,January,1954,pp 4-6.
- [30]. _____,_. And Heising,W.D.:”FORTRAN,”IEEE Trans.Elec.Comp.Vol.EC-13,No:4, August 1964,pp 304-305.
- [31]. Baumann,R.et all :”Introduction to ALGOL,” Prentice-Hall Inc., Englwood Cliffs , New Jersey, 1964.
- [32]. Battenburch, H.:”Structure and Use of ALGOL 60,” J.ACM,Vol. 9,No:2,April 1962, pp

161-221.

- [33]. Bromberg,H.:”COBOLendCompatibility, ”Datamation, V51,13,No:3,March 1963, pp 45-50.
- [34]. Cunningham,J.F.:”COBOL,” Comm. ACM ,Vol.6,No:3,March 1963, pp 79-98.
- [35]. Desautels,E.J.and Smith, D.K.:”An Introduction to the String Manipulation Language, SNOBOL,” Programming Systems and Languages [S.Rosen Ed.],Mcgraw-Hill, New York,1967,pp 419-454.
- [36]. Halsted, M.H., Uber,G.T. and Gielow,K.R.:”An Algorithmic Search Procedure for Program Generation,”Proc.SJCC vol.30,1967, pp 657-662.
- [37]._____,_.:”NELIAC “Comm.ACM, Vol.6, No.3,March,1963,PP 91-92.
- [38]. Haynam,G.E.:”An Extented ALGOL Based Language,”Proc.ACM 20 th National Conf., 1965, pp 449-454.
- [39].Heising,W.p.: “History and Summary of FORTRAN Standardization Development for U.S.A.,”Comm. ACM,Vol. 7, No. 10, October 1964.
- [40] . _____,: “FORTRAN,”Comm.ACM,Vol.6,No:3,March,1963, pp 85-86.
- [41]. -----,--:”FORTRAN:Compatibility and Standardization,”Datamation, Vol.10,No:8,August,1964,pp 24-25.
- [42]. Hermes,H.:”Enumerability.Decidability.Computability.”Spring-Veriag, Inc.,New York,1969.
- [43]. Hoernes,E.G.and Heilwell,F.M.:”Introduction to Boolean Algebra and Logic Design,”McGrow-Hill Book Company,New York,1964.
- [44]. Holberton,F.E.”Application of Automatic Coding to Lagical Process,” Symposium on Automatic Programming for Digital computers, Office of Naval Research,Department of Navy, Washington,D.C., 1954,pp 34-39.
- [45]. Landin,P.J.:”A Correspondence Between ALGOL 60 and Chuch’s Lamb- Da-Notation:Part I,”Comm.ACM,Vol.8,No:2,February,1965,pp 89-101.
- [46].-----,--:”A Corresponce Between ALGOL 60 and Church’s Lambda-Nota- tion :Part II,”Comm.ACM,Vol.8,No:3,March,1965,pp 158-165.
- [47].Laning,J.H.and Zierler,W.:”A program for Translation of Mathematical Equation for Whirlwind I,”M.I.T.Engineering Memorandum E-364,Instrumentation Lab.of Electronics,Cambridge,Mass.

January 1954.

- [48]. Mc Certhy, J. et all: "LIPS 1 Programmer's Manual," M.I.T. Computation Center and Research Lab. of Electronics, Combridge, Mass., March, 1960.
- [49]. -----, --.: "LISP 1.5 Programmer's Manual," M.I.T. Computation Center and Research Lab. of Electronics, Cambridge, Mass., March, 1960.
- [50]. Preliminary Report: "Specification for the IBM Mathematical FORMula TRANslating System, FORTRAN," IBM Corporation, Programming Research Group, Applied Science Division, 1954.
- [51]. Robins, D.K. "FORTRAN for Business Data Processing," Comm. ACM, Vol. 5, No: 7, July, 1962, pp 412-14.
- [52]. Rogers, Jr. H.: "Theory of Recursive Functions and Effective Computability, McGraw-Hill Book Company, New York, 1967.
- [53]. Sammet, J.E.: "Programming Languages," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969.
- [54]. Saxon, J.A.: "COBOL: A Self-Instructional Manual," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.
- [55]. Speeding System for the Type 701 Electronic Data Processing Machines, IBM Corporation, 24-6059-0. September, 1953.
- [56]. UNIVAC SHORT CODE, Remington Rand, Inc., Philadelphia, Oct., 1952,
- [57]. USA Standard FORTRAN, United States of American Standards Institute, USAS x 3-9-1966, New York, March 1966.
- [58]. USA Standard Basic FORTRAN, United States of American Standards Institute, USAS x 3.10-1966, New York, March 1966.
- [59]. Ünlü, F.: "On λ -calculus and its Features in Programming Languages," U.T. at Austin, 1971 [Manuscript].
- [60] Ünlü, F. : "A Remote Programming Technology on a Remote VDM Clustering in lambda-Calculus". Int. Math. Form. Vol. 2006, no. 13-16-671-685.
- [61]. Introduction to PL/I [Student text], IBM Corporation, C20-1632, Data Processing Division, White Plains, New York, 1964.

