Eryılmaz ve Albayrak, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 27(4):292-295

Erciyes University Journal of the Institute of Science and Technology
Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi
ISSN 1012-2354
Cilt (Volume) 27, Sayı (Issue) 4, Ekim/October-2011
http://fbe.erciyes.edu.tr/

Lamda algoritmasında kongruanslarının türetimi

[1]Aytekin ERYILMAZ*, [2]Lütfullah ALBAYRAK

[1]*Nevşehir Üniversitesi, Fen Edebiyat Fakültesi, Matematik Bölümü,* eryilmazaytekin@gmail.com
[2]*Süleyman Demirel Üniversitesi, Fen Edebiyat Fakültesi, Matematik Bölümü, lalbayrak@hotmail.com*

ÖZET

Lamda Kalkülüs 1930'larda mantık bilimcisi Alanzo Church tarafından temelleri kurulan, bir fonksiyon notasyonuna bağlı, çeşitli sistemlerin bir kümesidir (Hindley ve Seldin, 1986). Bilgisayar bilimlerinde Lamda Kalkülüs'ün açık ve sistematik kullanımı Peter Landin, Christopher Strachey ve Lamda Kalkülüs üzerine kurulmuş programlama dillerinin teorik anlatımını geliştiren diğer bilim insanları tarafından başlatılmıştır (Revesz, 1988). λ-algoritmasının alfabesi, yazılım yapısı Church (1965), Kleene (1936), Barendregt (1984), Bakker(1975), Byrkıt (1970), Krivine (1993), Levy (1975) tarafından ortaya konuldu. Daha sonraları Ünlü (1976), Mirasyedioğlu (1982, 1987) ve Albayrak (1982,1985,1986,1989,1993) tarafından çalışıldı. Bu çalışmada; teorik λ-algoritması adı altında bir formal sistem geliştirilmiştir. Sistemin önemli öğeleri: bağıntılar, fonksiyonlar ve operatörlerdir. λ-algoritması yazılım yapısı, türetim teknikleri ve λ-kongruansının tanımı verildi. λ-algoritması kuralları içinde λ-kongruansları türetildi.

Anahtar Kelimeler

Lamda Kalkülüs, λ-algoritması, λ-kongruansları

The production of congruences in lamda algorithm

ABSTRACT

Lamda Calculus was founded in 1930's by Alanzo Church (Hindley and Seldin, 1986). The systematic and explicit usage of Lamda Calculus were initiated by Peter Landin, Christopher Strachey (Revesz, 1988). The software structure of λ-algorithm has been studied from the point ofdervational techniques by Church (1965), Kleene (1936), Barendregt (1984), Bakker(1975), Byrkıt (1970), Krivine (1993), Levy (1975). In this study, a formal system under the name of theoritical λ-algorithms has been developed.The important elements in it are expressions, functions and operators. The software structure of λ-algorithm, productional techniques and definitions of λ-congruence are given. Under the rules of λ-algorithm, λ-congruences are producted. Later, Ünlü (1976), Mirasyedioğlu (1982, 1987) and Albayrak (1982,1985,1986,1989,1993) were studied λ-algorithms expanded upon algebraic structures.

Keywords

Lamda Calculus, λ-algorithm, λ-congruences

** Sorumlu yazar (Corresponding author) e-posta: eryilmazaytekin@gmail.com*

1. Giriş

Bilgisayar bilimlerinin gelişmesi ile birlikte Lamda Kalkülüs'e de büyük ilgi olagelmiştir. Lamda Kalkülüs ile bilgisayar bilimleri arasındaki ilgi çok açıktır. Örnek olarak, 1958'de John McCarthy tarafından icat edilen ve günümüzde ANSI Common Lisp olarak bilinen LISP programlama dilinin planlaması λ-algoritmasından etkilenmiştir (Revesz, 1988). Ayrıca auto cad dilindeki formüllerin kodlanması λ-algoritmasındaki gibidir (Albayrak, 1986).

Lamda Kalkülüs 1930'larda mantık bilimcisi Alanzo Church tarafından temelleri kurulan, bir fonksiyon notasyonuna bağlı, çeşitli sistemlerin bir kümesidir (Hindley ve Seldin, 1986). Bilgisayar bilimlerinde Lamda Kalkülüs'ün açık ve sistematik kullanımı Peter Landin, Christopher Strachey ve Lamda Kalkülüs üzerine kurulmuş programlama dillerinin teorik anlatımını geliştiren diğer bilim insanları tarafından başlatılmıştır (Revesz, 1988). λ-algoritmasının alfabesi, yazılım yapısı Church (1965), Kleene (1936), Barendregt (1984), Bakker(1975), Byrkıt (1970), Krivine (1993), Levy (1975) tarafından ortaya konuldu. Daha sonraları Ünlü (1976), Mirasyedioğlu (1982, 1987) ve Albayrak (1982,1985,1986,1989,1993) tarafından çalışıldı.

Lamda Kalkülüs'ün amacı fonksiyonların en genel özelliklerini incelemektir. Bu, matematiğin çeşitli bölümlerinde kullanılan fonksiyon çeşitleri ile bütünleştirilmiş bir fonksiyonlar teorisi geliştirilmek istendiği anlamına gelir. Bunu yapma yollarından biri, temel olarak küme teorisini kullanmak ve bunun üzerine bir fonksiyon fikrini inşa etmektir. Bu fikre göre fonksiyon, sıralı ikililerin (genellikle sonsuz) kümesidir. Fakat Lamda Kalkülüs'de fonksiyonlar sıralı ikililerin kümesi ile değil, lamda bağıntıları denilen sembolik notasyonlarla gösterilir.
Bu çalışmada; λ-algoritması yazılım yapısı, türetim teknikleri ve λ-terimleri aracılıyla λ-kongruansının tanımı verildi. λ-algoritması kuralları içinde λ-kongruansları türetildi.

2.Ön bilgiler

Tanım 1: λ-terimleri "$(\ ,\ ),\ \lambda$" sembolleri ve $x, y, z,$ değişkenlerinden oluşan sonlu dizilerdir. Bu diziler λ-teriminin sonlu bir sayıda uygulanması ile elde edilir. Ayrık sembollerin sonsuz bir dizisinin değişkenler; ayrık sembollerin sonlu, sonsuz veya boş bir dizisinin de sabitler olduğu kabul edilirse, λ-bağıntılarının kümesi de denilen λ-terimleri tekrarlı olarak aşağıdaki gibi tanımlanır.
i) Bütün değişkenler ve sabitler λ-terimleridir. Sabitlerin ve değişkenlerin ikisine birden atomlar denir.
ii) M ve N iki λ-terimi ise, (MN) de λ-terimidir.
Bir λ-bağıntısının başka bir λ-bağıntısına uygulanmasına application denir. (MN) ye application veya komut da denir. (MN) bağıntısında M ye operatör (işleyen), N ye operand (işlenen) denir.
Bu demektir ki her λ-bağıntısı hiçbir kısıtlama olmaksızın hem operatör hem de operand olarak kullanılabilir.
iii) Eğer bir M bir λ-terimi ve x de bir değişken ise, $(\lambda x.M)$ de bir λ-bağıntısıdır. $(\lambda x.M)$eabstraction veya M gövde bağıntılı λ-bağıntısıda denir.

Abstractionın amacı, verilen bir λ-bağıntısından daha farklı bir λ-bağıntısı elde etmektir. Atomlar en basit λ-bağıntılarıdır. Daha karmaşık λ-bağıntıları application ve abstractiondan oluşan iki bağıntıyı kullanarak elde edilir (Eryılmaz, 1996).

Tanım2: λ-algoritmasındaki λ-terimlerinin bulunduğu kümeye λ-terim uzayı denir ve $I\!E$ ile gösterilir (Albayrak, 1982).
λ- algoritması kuralları ile $I\!E$ de türetilen her λ-bağıntısına λ-terimi uzayı denir (Albayrak, 1982).

Tanım 3: Bir P bağıntısında $\lambda x.M$ bağıntısı için, M nin bulunuşuna λ nın faaliyet alanı denir. Örneğin $P = (\lambda y.yx(\lambda x.y(\lambda y.z)x))$ bağıntısında en soldaki λ nın faaliyet alanı $yx(\lambda x.y(\lambda y.z)x)$ dır. İkinci λ nın faaliyet alanı $y(\lambda y.z)x$ dır ve üçüncü λ nın faaliyet alanı x dir (Eryılmaz, 1996).

Tanım 4: $\lambda x.M$ bağıntısındaki λ ya yönetici, x e bağlayıcı değişken denir. Eğer x değişkeni $P = \lambda x.M$ biçiminde bir bağıntının içinde bulunursa, bu durumda x e bağımlı değişken denir. Aksi takdirde x serbest değişkendir. Serbest değişkeni olmayan λ-terimlerine kapalı terim denir. Örneğin, $P = (\lambda x.(z)x)\lambda y.(y)x$ bağıntısında ilk x bağlayıcı değişken, ikinci x bağımlı değişken, üçüncü x de serbest değişkendir. Birinci$y$ bağlayıcı değişken, ikinci y bağımlı değişken, z de serbest değişkendir. Bu örnekten bir değişkenin, bir bağıntıda hem serbest değişken hem de bağımlı değişken olarak bulunabileceği görülmektedir (Barendregt, 1984; Revesz, 1988).

3. λ-Kongruanslarının türetimi

Tanım 5: $a, b, m \in I\!E$ olsun ve $m \leftarrow Y^m D$, $a \leftarrow Y^a D$, $b \leftarrow Y^b D$ biçiminde tanımlayalım. Eğer $(a-b) \leftarrow ((\text{çıkar})a)b$ sayısı m ile tam bölünebiliyorsa a ve b sayılarına $\text{modül } m$ ye göre λ-kongruenti denir.

$$a \equiv b \ (mod\ m) \leftarrow (((\sim a)b)m)$$

biçiminde gösterilir. Bu tanıma göre

$$m|(a-b) \Leftrightarrow a \equiv b \ (mod\ m)$$

olur. Yani, λ-algoritmasında

$$m|((\text{çıkar } a)b) \Leftrightarrow a \equiv b \ (mod\ m)$$

olur. Eğer $((\text{çıkar})a)b$ sayısı m sayısına bölünmüyorsa a sayısı $\text{modül } m$ ye göre b sayısına λ-kongruenti değildir denir ve $(((\sim a)b)m)$ şeklinde gösterilir. Burada ~ sembolü *çıkar* anlamında kullanılmıştır.

Örnek: $a \leftarrow 32,\ b \leftarrow 4$ ve $a, b, m \in I\!E$ olmak üzere, λ-algoritması kuralları ile $m|((\text{çıkar } 32)4) \Leftrightarrow \leftarrow (((\sim 32)4)m)$ veya

$m|(32-4)$ dir. Buna göre m, 28 in pozitif bölenleri olacaktır. Şu halde, $m \in \{1,2,4,7,14,28\}$ olur.

Teorem 6: $a,b,m \in IB$, λ-algoritmasında sayılar olsun. $(((\sim a)b)m)$ λ-kongruenti olmak üzere,

$$(((\sim a)b)m) \Leftrightarrow a = ((\text{topla}(\text{çarp}\,q)m))b) \qquad (3.1)$$

olacak şekilde λ-algoritmasında bir $q \in IB$ sayısı vardır.

İspat: $(((\sim a)b)m)$ olsun. Bunun anlamı $m|((\text{çıkar}\ a)b)$ veya $((\text{çıkar}\ a)b) = (\text{çarp}\ m)q)$ olacak şekilde bir $q \in IB$ vardır. Bu ise $a = ((\text{topla}(\text{çarp}\,q)m))b)$ olacak şekilde bir $q \in IB$ sayısının varlığını gerektirir.

Teorem 7: $a,b,c,m \in IB$ λ-algoritmasında sayılar olsun. $(((\sim a)b)m)$ λ-kongruenti ise,

i) $(((\sim((\text{topla}\ a)c))((\text{topla}\ b)c))m)$

ii) $(((\sim((\text{çarp}\ a)c))((\text{çarp}\ b)c))m)$

dir.

İspat: i) (3.1) ifadesinde bir $q \in IR$ sayısının varlığı bilindiğinden, bu eştliğin her iki tarafına c sayısı λ-algoritması kuralları ile eklenirse,

$$(((\sim((\text{topla}\ a)c))((\text{topla}\ b)c))m)$$

olur. Bu durumda

$$((\text{çıkar}((\text{topla}\ a)c))((\text{topla}\ b)c)) = ((\text{çarp}\ q)m)$$

elde edilir ve ispat tamamlanır.

ii) (3.1) ifadesinde, eşitliğin her iki tarafını c sayısı, λ-algoritması kuralları ile çarpılırsa

$$((\text{çarp}\ a)c) = ((\text{topla}((\text{çarp}\ b)c))((\text{çarp}((\text{çarp}\ q)c))m))$$

olur ve ispat tamamlanır.

Tanım 8: Elemanları λ-kültüründe, λ-algoritması kuralları ile türetilen her kümeye λ-algoritması kümesi denir.

Tanım 9: IR de tanımlanan elemanlarla yansıma, simetri ve geçişme özellikleriolan bir bağıntıya λ-denklik bağıntısı denir ve

i) $(((\sim a)a)m)$ (Yansıma özelliği)

ii) $(((\sim a)b)m)$ ise $(((\sim b)a)m)$ (Simetri özelliği)

iii) $(((\sim a)b)m)$ ve $(((\sim b)c)m)$ ise $(((\sim a)c)m)$ (Geçişme özelliği)

biçiminde tanımlanır.

Teorem 10: $a,b,c,m \in IB$ λ-algoritmasında sayılar olsun. Pozitif sayılar kümesinde tanımlanan λ-kongruentlik bağıntısı bir λ-denklik bağıntısıdır.

İspat: λ-algoritmasında negatif sayılar tanımlanmadığından $a,b \in IB$ ve $a > b$ iken $((\text{çıkar}\ b)a) < 0$ sayısı mutlak değer olarak düşünülecektir ve $c = b - a$ sayısının mutlak değeri λ-algoritmasında $\lambda[C]_{MD}$ şeklinde gösterilecektir. Şimdi Tanım 4 de verilen yansıma, simetri ve geçişme özelliklerini λ-algoritması kuralları içerisinde gösterelim.

i) $m|((\text{çıkar}\ a)a) = 0$ olduğundan $(((\sim a)a)m)$ dir. Yani, yansıma özelliği sağlanır.

ii) $a > b$ ve $((\text{çıkar}\ b)a) = c$ olsun. $(((\sim a)b)m)$ olduğundan $m|((\text{çıkar}\ a)b)$ ve $m|((\text{çıkar}\ a)b) = ((\text{çarp}\ q)m)$ yazılabilir. Yani, bir $q \in IB$ sayısı vardır. $((\text{çıkar}\ b)a) = ((\lambda[C]_{MD})m)$ olduğundan $m|((\text{çıkar}\ b)a)$ veya $m|c$ olur. Bunun anlamı ise λ-kongruentinin tanımı olan $(((\sim b)a)m)$ dir. Böylece simetri özelliği gösterilmiş olur.

iii) Tanım 4 den $(((\sim a)b)m)$ ve $(((\sim b)c)m)$ iken $(((\sim a)c)m)$ olduğunu göstermeliyiz. $(((\sim a)b)m)$ λ-kongruenti iken,

$$a = ((\text{topla}(\text{çarp}\,q)m))b) \qquad (3.2)$$

olacak şekilde bir $q \in IB$ sayısı vardır. $(((\sim b)c)m)$ λ-kongruenti iken,

$$b = ((\text{topla}(\text{çarp}\,p)m))c) \qquad (3.3)$$

olacak şekilde bir $p \in IB$ sayısı vardır. Şimdi (3.3) ifadesi (3.2) de yerine yazılırsa

$a = ((\text{topla}\ c)((\text{çarp}\,p)m))((\text{çarp}\ q)m))$ veya

$a = ((\text{topla}\ c)((\text{çarp}((\text{topla}\ p)q))m))$ olur.

Bunun anlamı ise $((\text{çıkar}\ a)c) = ((\text{çarp}((\text{topla}\ p)q)m)$ dir. Bu ise $m|((\text{çıkar}\ a)c)$, yani $(((\sim a)c)m)$ dir.

Bu teorem ile λ-kongruentlik bağıntısının λ-denklik bağıntısı olduğu gösterildi. Her denklik bağıntısı, üzerinde tanımlanan λ-algoritması kümesini denklik sınıflarına ayırır. Ayrıca bu denklik sınıfları ikişer ikişer ayrıktır. Örneğin modül 2 deki λ-kongruanslarının sınıflarının her biri IB de λ-algoritması kuralları ile türetilen,

$$K_1 = \{0,2,4,6,\ldots\} = \{((\text{çarp}\ 2)k),\ k \in IB\}$$

$$K_2 = \{1,3,5,7,\ldots\} = \{((\text{topla}((\text{çarp}\ 2)k))1),\ k \in IB\}$$

kümeleridir.

Teorem 11: $a, b, c, d, m \in IB$ λ-algoritmasında sayılar olsun. $(((\sim a)b)m)$ ve $(((\sim c)d)m)$ λ-kongruentleri ise,

$$(((\sim((topla\ a)c))((topla\ b)d))m)$$

dir.

İspat : $(((\sim a)b)m)$ ise $a \blacksquare ((topla\ b)((çarp p)m))$ (3.4)

ve

$(((\sim c)d)m)$ ise $b \blacksquare ((topla\ c)((çarp q)m))$ (3.5)

olacak şekilde $p \in IB$ ve $q \in IB$ sayıları vardır. (3.4) ve (3.5) taraf tarafa toplanırken $p \blacksquare ((topla\ p)q) \in IB$ gruplaması yapılırsa

$$((topla\ a)c) \blacksquare ((topla((topla\ b)d))((çarp\ p)m))$$

olur. Bunun anlamı ise teoremde ispatı istenen,

$$(((\sim((topla\ a)c))((topla\ b)d))m)$$

dir.

4. Sonuç ve öneriler

IB kümesiyle λ-kültüründe λ-algoritması kurallarına uygun operatörler yardımıyla λ-kongruansları türetildi.

IB kümesiyle tanımlı λ-kültüründen türetilen doğal sayılar kümesi yardımıyla başlıca operatörler ve cebirde geçerli matematik sistemler ve diğer özel konular λ-algoritmasında türetilebilir.

KAYNAKLAR

1. Albayrak, L.,(1982). *λ-Kültüründen Cebirsel Yapı Türetimi,* Ege Üniversitesi, Fen Fakültesi, İzmir.
2. Albayrak, L., (1985). *λ-Algoritmasındaλ-Kültürünün Tanımı veT_{MT} Operatörünün Türetilmesi,* A.Ü.Isparta Mühendislik Fakültesi Dergisi, Isparta.
3. Albayrak, L., (1986). *Matematik Sistemlerin Bilgsayarla Ayrımı,* A.Ü.Isparta Mühendislik Fakültesi Dergisi, Isparta.
4. Albayrak, L., (1990). *λ-Algoritmasında Yarı Grup ve MonoidTüretimi,* Yüzüncü Yıl Üniversitesi III. Ulusal Matematik Sempozyumu, Van.
5. Albayrak, L., (1993). *λ-AlgoritmasındaGrup ve Değişmeli Grup Türetimi,* Doğu Akdeniz Üniversitesi IV. Ulusal Matematik Sempozyumu, Magosa.
6. Bakker, J.W.,(1975). *λ-Calculus and Compute rScience Theory,* Proceedings of Symposium in Rome, pp 27-54.
7. Barendregt, H.P., (1984). *λ-Calculus, its Syntax and Semantics,* revised ed., North Holland.
8. Byrkıt, D.R. and Petlofrezzo, A.J., (1970). *Elements of Number Theory*, Prentice-Hall, New Jersey.
9. Church, A., (1965). *The Calculi of Lambda-Conversion*, Princeton University Press, New York.
10. Eryılmaz, A., (1996), *λ-Algoritmasında Kongruansların Türetimi*, Yayınlanmamış Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta
11. Hindley, J.R. and Seldin, J.P.,(1986). *Introduction to Combinators and λ-Calculus* Cambridge Unv. Press, England.
12. Kleene, S.C.,(1936). *λ-Definability and Recursiveness,* Duke Mathematical Journal. Vol. 2, pp 340-353.
13. Krivine, J.L.,(1993). Lambda Calculus, Types and Models Masson, Ellise Harswood, England.
14. Levy, J.J., (1975). *λ-Calculus and Computer Science Theory,* Proceedings of Symposium in Rome, pp 147-163.
15. Mirasyedioğlu, Ş.,(1982). *λ-Functional Exclusive or/and Logical Equivalence,* Karadeniz University, Mathematical Journal, Vol.V. No:1, pp 64-76, Trabzon.
16. Mirasyedioğlu, Ş., (1987). *λ-Boolean Theory,* Intenational Logic Review, 35 pp 13-27.
17. Revesz, G.E.,(1988). *λ-Calculus, Combinators and Functional Programing,* Cambridge Unv. Press, England.
18. Ünlü, F.,(1976). *Kuramsal λ-Algoritması,* Atatürk Üniversitesi, Yayın No:472, Erzurum.