G.Ü. Fen Bilimleri Dergisi,
16(3):627-640, 2003

ISSN 1303-9709

G.U. Journal of Science,
16(3):627-640, 2003

KISALTICI MEKANİZMASININ BİR BOYUTLU HÜCRESEL HAREKETLİLERDE GERÇEK ZAMANDA SİMULASYONU

Zeki ÇİFTÇİ
Proje I Müdürlüğü, Türkiye Kalkınma Bankası,Necatibey, Ankara, TÜRKİYE,
zeki-c@tkb.com.tr

Doğan ÇALIKOĞLU
Gazi Üniversitesi,Endüstriyel Sanatlar Eğitim Fakültesi, Bilgisayar Eğitimi Bölümü, Ankara, TÜRKİYE

Halit OĞUZTÜZÜN
Orta Doğu Teknik Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Ankara, TÜRKİYE

ÖZET

Girdi şeridi üzerinde, kısaltım işlemleri yaparak girdiyi tanıyan ve "kısaltıcı" olarak adlandırılan mekanizmalar vardır. Bu çalışmada, zincir adı verilen bir-boyutlu hücresel hareketlilerin belirlenimli kısaltıcıları gerçek zamanda simule edebildiği ortaya konmuştur. Zincirler, sonlu durumlu Moore makinelerinden oluşan bir-boyutlu hücresel yapılardır. Zincirde her hücrenin girişi, her iki yanındaki komşularının çıkışlarına bağlıdır. Burada, verilen herhangi bir belirlenimli kısaltıcıya karşılık, onu simule eden bir zincir olduğu gösterilirken, yapımsal ispat yöntemi kullanılmıştır. Kısaltıcı mekanizmasından hareketle kurulan zincir, dengeli parantezleri ve ortası belli palindromları kabul etmektedir. Zincir, bu dizgileri kabul ederken kısaltıcıya göre daha az sayıda geçiş yapmaktadır.

Anahtar Kelimeler: Zincir, kısaltıcı, bir-boyutlu hücresel hareketli, yığmalı hareketli

THE SIMULATION OF SHRINKERS BY ONE DIMENSIONAL CELLULAR AUTOMATA IN REAL TIME

ABSTRACT

There are some mechanisms called shrinkers which recognize the input by performing shrinking operations on the input tape. In this work, it is shown that one-dimensional cellular automata called chains can simulate deterministic shrinkers in real-time. Chains are one-dimensional cellular structures consisting of finite-state Moore machines. In a chain the input of each cell is connected to the outputs of the two neighboring cells on each side. Here a constructive proof method is employed in showing that a chain exists corresponding to any given deterministic shrinker which simulates it. The chain which is constructed corresponding to a shrinker can accept balanced parenthesis and palindromes with distinguished centers. In accepting these strings a chain performs less number of transitions with respect to shrinkers.

Key Words : Chain, shrinker, one-dimensional cellular automata, pushdown automata.

1.GİRİŞ

Hücresel hareketliler üzerindeki çalışmalar von Neumann'ın kendini üretme konusunda yaptığı çalışmayla başlamıştır. Thatcher, von Neumann modeli üzerinde çalışarak ortaya koyduğu iki boyutlu, 29 durumlu ve beş komşulu hücresel hareketlinin kendini üretme kapasitesi ve evrensel hesaplama özelliği bulunduğunu göstermiştir. Aynı sonuçlar durum sayısı sekize indirilerek gösterilmiştir. Sonunda iki boyutlu hücresel hareketlide, başlangıç yerleşiminde boşluk olmayan durumların sayısını kısıtlamamak şartıyla evrensel hesaplama için sadece iki durumlu hücre yapısının yeterli olduğu gösterilmiştir. Evrensel hesaplamalı bir boyutlu hücresel hareketli için kendini üretme özelliği olduğu Smith tarafından gösterilmiştir (1).

Belirlenimli bağlam-bağımsız dillerin bir-boyutlu hücresel hareketliler tarafından gerçek zamandan daha kısa bir zamanda kabul edildiği Çalıkoğlu tarafından gösterilmiştir (1).

Dyer (2) "Bir belirlenimsiz bir-yönlü kısıtlı hücresel tanıyıcının belirlenimsiz veya belirlenimli kısıtlı hücresel tanıyıcıyı simule ettiği"ni bir teoremle ispat etmiştir. Bu teoremi ve Smith'in "belirlenimsiz kısıtlı hücresel tanıyıcı dillerinin bağlam-duyarlı dillere eşdeğer olduğu" sonucunu kullanarak belirlenimsiz bir-yönlü kısıtlı hücresel tanıyıcı diller sınıfının bağlam-duyarlı diller sınıfına eşdeğer olduğunu göstermiştir.

Dyer (2) bir-yönlü kısıtlı hücresel tanıyıcının bağlam-duyarsız dillerin önemli alt sınıflarından Dyck dilini, doğrusal bağlam-bağımsız dilleri ve parantezli bağlam-bağımsız dilleri gerçek zamanda kabul ettiğini göstermiştir.

Son zamanlarda bir-boyutlu hücresel hareketlilerle ilgili yapılan çalışmalarda zamanla kurulabilir fonksiyonlar üzerinde durulmuştur (3, 4). Tersinir hücresel hareketlilerle ilgili bir simulasyonu ise Durand-Lose (5) gerçekleştirmiştir.

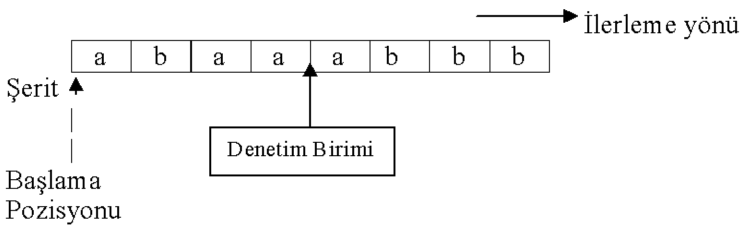
Bu makalede girdi şeridi üzerinde kısaltım işlemleri yaparak girdiyi tanıyan kısaltıcıların (6) bir-boyutlu hücresel hareketliler (1) tarafından gerçek zamanda simulasyonu verilmiştir. Makalede simulasyonla ilgili kavramlar ve simulasyon aşağıdaki sıra izlenerek açıklanmıştır. İkinci bölümde kısaltıcılar üzerinde durulmuştur. Kısaltıcıların tanımı ve kısaltıcılarla ilgili temel kavramlar verilmiştir. Aynı bölümde kısaltıcı dilleri ve belirlenimli kısaltıcının tanımları da yer almaktadır.

Üçüncü bölümde zincir yapıları ile ilgili temel tanımlar vardır. Zincir, akış zinciri, hesaplama, yerleşim, etkinlik gibi kavramlar üzerinde durulmuştur.

Dördüncü bölümde, belirlenimli kısaltıcıların kabul ettikleri dili zincir yapılarının da gerçek zamanda kabul ettikleri gösterilmiştir (7). Bunu göstermek için bir belirlenimli kısaltıcı mekanizmasından bir zincir inşa edilmiş ve belirlenimli kısaltıcıların yaptığı her hamleyi zincir yapısının da yapabildiği tümevarım yoluyla ispatlanmıştır. Yapısal ispat yoluyla kurulan zincir, dengeli parantezleri ve ortası belli palindromları, kısaltıcıların bu dilleri kabul etmek için yaptığı hamle sayısından daha az sayıda geçiş yaparak kabul etmektedir.

2. KISALTICILAR

Kısaltıcılar, bir denetim birimi ve bir kafadan oluşmaktadır. Denetim birimi aslında bir belirlenimsiz sonlu durumlu hareketli ("nondeterministic finite automata") olarak çalışmaktadır. Kısaltıcı mekanizması, girdiyi karelere bölünmüş sonlu bir şerit üzerinden okumaktadır (Şekil 2.1.). Girdi şeridi üzerindeki her karenin içinde girdi dizgisinin bir simgesi yer almaktadır. Kısaltıcının kafası ardışık iki kareyi okuyabilmektedir. Kafa herhangi bir anda iki karenin arasındaki sınırda ya da şeridin sol veya sağ tarafındaki sınırlarından birinde durmaktadır.



Şekil 2.1. Bir kısaltıcının yapısı

Her hamlede kafa, sol ve sağ karelerden birini veya ikisini okur ya da hiçbirini okumaz. Bir karenin okunmadığı durumda, bu kareden girdi olarak boş dizgi geldiği kabul edilir. Sol ve sağ karelerden gelen girdilere ve denetim biriminin içinde bulunduğu duruma göre kısaltıcıda aşağıdaki işlemler meydana gelir.

1) sol taraftaki kareyi yok eder,

2) sağ taraftaki kare ya yok edilir veya kafa bu karenin üzerinden bir kare sağa doğru ilerler,

3) denetim birimi için bir durum değişikliği olabilir.

1.sırada anlatılan işlemin olabilmesi ancak sol taraftan bir simge okunması halinde mümkündür. 2.sıradaki işlem için de sağ taraftan bir simge okunması gerekir. Kısaltma işlemi sonucunda girdi şeridi kısalır. Kısaltma hem sağ hem de sol taraftan olabilir. Şerit üzerinde ilerleme ise soldan sağa olmak üzere tek yönlüdür.

Kısaltıcı, verilen bir girdi şeridi üzerinde işlem yapmaya kafa başlangıç durumundayken ("initial state") şeridin sol kenarından başlar. Kısaltıcı, birtakım hamlelerden sonra şeridi kısaltarak bitirir ve bir sonuç durumuna ("final state") girerse verilen şeridi kabul etmiştir. Bu şekilde kabul ettiği şeritler (dizgiler) bu kısaltıcı tarafından kabul edilen dili oluşturur.

2.1. Kısaltıcılara İlişkin Temel Kavramlar

Tanım 2.1: Bir kısaltıcı beş elemandan oluşmaktadır: $K = (Q, \Sigma, \delta, q_0, F)$, burada

Q sonlu durumlar kümesi,

Σ bir alfabe,

δ: $(\Sigma \cup \{\varepsilon\}) \times Q \times (\Sigma \cup \{\varepsilon\})$' den $Q \times \{., \leftarrow\}$ 'in alt kümelerine durum geçiş fonksiyonu öyle ki $(q', \leftarrow) \in \delta(a, q, b)$ ise $b \neq \varepsilon$,

$q_0 \in Q$ başlangıç durumu,

$F \subseteq Q$ sonuç durumları kümesidir.

Tanım 2.2: K'nın *an-tarifi* ("instantaneous description") $\Sigma^* Q \Sigma^*$'ın bir elemanıdır.

Tanım 2.3: K'nın an-tarifleri üzerindeki ikili ilişki aşağıdaki gibi tanımlanmıştır.

$u, w \in \Sigma^*$, $a, b \in \Sigma \cup \{\varepsilon\}$, $c \in \Sigma$, $q, r \in Q$

1) $(r, .) \in \delta(a, q, b)$ ise $waqbu \vdash wru$,

2) $(r, \leftarrow) \in \delta(a, q, c)$ ise $waqcu \vdash wcru$.

Tanım 2.4: K'nın an-tariflerinin bir serisi $I_0, \ldots, I_k$ $(k \geq 0)$ K'da bir hesaplama olarak adlandırılır. Burada ya $k = 0$ ya da $k \geq 1$ ve $0 \leq i < k$ için $I_i \vdash_K I_{i+1}$ olur.

k = 1 için hesaplama bir hamle ("move") olarak adlandırılır.

Tanım 2.5: K'nın I ve I′ an-tarifleri için K'da bir hesaplama ilişkisi varsa $I_0, \ldots, I_k$ $(k \geq 0)$ an-tarifler serisinde $I_0 = I$ ve $I_k = I'$ olarak alınırsa "K'da I′'yı, I'dan k adımda elde ederiz" denir $(I \vdash^k_K I')$.

$\vdash_K^*$ ilişkisi, $\vdash_K$ ilişkisinin yansımalı ve geçişli kapanışı olarak tanımlanır.

$u_1, u_2, w_1, w_2, x, y \in \Sigma^*$, $q_1, q_2 \in Q$ için $u_1 q_1 w_1 \vdash_K^* u_2 q_2 w_2$ eğer ve ancak $xu_1 q_1 w_1 y \vdash_K^* xu_2 q_2 w_2 y$.

Tanım 2.6: K tarafından kabul edilen dil, L(K) olarak gösterilir ve aşağıdaki gibi tanımlanmıştır.

$$L(K) = \{u \in \Sigma^* \mid q_0 u \vdash_K^* q, \text{ bazı } q \in F \text{ için}\}$$

Kısaltıcıların bir dili kabul etmesi için kısaltıcının hem sonuç durumuna girmesi hem de teybin yok edilmesi gerekmektedir. Yani yığmalı hareketlilerin kabul etme şartlarının ikisinin de aynı anda gerçekleşmesi gerekmektedir. Kısaltıcılar tarafından kabul edilen diller kısaltıcı dilleri olarak adlandırılır.

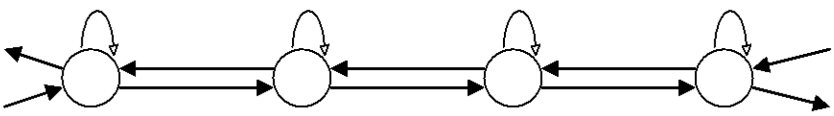
Tanım 2.7: Bir K kısaltıcı mekanizması aşağıdaki koşulları sağlıyorsa belirlenimli bir kısaltıcıdır.

1. $a, b \in \Sigma \cup \{\varepsilon\}$, $q \in Q$ olmak üzere $\delta(a, q, b)$ en fazla bir elemana sahiptir.
2. $c \in \Sigma$, $b \in \Sigma \cup \{\varepsilon\}$, $q \in Q$ olmak üzere $\delta(c, q, b) \neq \varnothing$ ise $\delta(\varepsilon, q, b) = \varnothing$ olmalıdır.
3. $d \in \Sigma$, $a \in \Sigma \cup \{\varepsilon\}$, $q \in Q$ olmak üzere $\delta(a, q, d) \neq \varnothing$ ise $\delta(a, q, \varepsilon) = \varnothing$ olmalıdır.

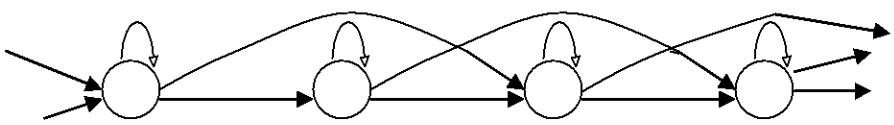
Birinci kural kısaltıcının aynı girdi için birden çok seçenekle karşılaşma ihtimalini ortadan kaldırır. İkinci kural, sol taraftan bir girdi okunması durumunda, buradan okuma yapmama (boş girdi gelmesi) seçeneğinin olamayacağını ifade eder. Üçüncü kural aynı durumu sağ taraf için yapar.

3. ZİNCİRLER

Bu bölümde, *Evrensel Zincir (E-zincir)* ve tipleri anlatılmıştır (1). Bir *E-zincir*, sonlu durumlu Moore makinelerinin oluşturduğu 1-boyutlu hücresel hareketlidir. Bir E-zincir iki yolla bağlanabilir. 1) Her hücre, sağ ve solundaki iki komşu hücrenin çıktılarına bağlanır (Şekil 3.1.), 2) her hücre, solunda kalan en yakın iki komşu hücrenin çıktılarına bağlanır (Şekil 3.2.). Birinci durumdaki E-zincir, zincir olarak adlandırılır. İkinci durumdaki zincir ise 1-yönlü Akış Zinciri veya kısaca A-zincir olarak adlandırılır.



Şekil 3.1. Bir zincirin bağlantısı



Şekil 3.2. Bir A-zincirin bağlantısı

Her hücre her bir adımda, o hücrenin geçiş fonksiyonu tarafından tanımlanan bir duruma geçer. Her hücrenin durum simgeleri *"boşluk"* veya *"durgun durum"* diye ifade edilen özel bir simge içerir. Sol veya sağ taraftan boşluk simgeleri tarafından sınırlandırılan boşluk olmayan durumlara sahip hücreler serisine bir *etkinlik* denir. Her E-zincirde tam olarak bir etkinlik olduğu ve bu etkinliğin sonlu olduğu varsayılır. Bir hücre boşluk durumunda ve durum geçiş fonksiyonu ile belirtilen komşuları da boşluk durumundaysa bu hücre, aynı durumda kalmaya devam eder. E-zincirin $t = 0$ anındaki ilk etkinlik *girdi*dir ve etkinlikte o andaki durumlar girdi durumları veya girdi simgeleridir. E-zincirin bir girdiyle başlayan yerleşim serisine bu girdi üzerindeki *hesaplama* denir. Hesaplamalar, *"tanıma"* ya yani girdinin belirlenmiş bir kümenin (sonuç durumları kümesi) elemanı olup olmadığına karar verir. Bu şekilde tanınan girdiler kümesi bir dili oluşturur.

Tanım 3.1: Evrensel-zincir

i) Bir E-zincir Ç altı elemandan oluşur.
$Ç = (\Sigma, ƀ, Q, S, F, f)$
Σ sonlu, boş olmayan girdi simgeleri kümesi,
ƀ (boşluk) özel bir simge, $ƀ \notin \Sigma$,
Q sonlu bir küme, $Q \cap (\Sigma \cup \{ƀ\}) = \emptyset$,
$S = \Sigma \cup \{ƀ\} \cup Q$ durum simgeleri kümesi,
$F \subseteq Q$ sonuç durumları kümesi,
f: $S^3 \to S$ durum geçiş fonksiyonu $f(ƀ, ƀ, ƀ) = ƀ$ olduğu varsayılır.
ii) Σ^+ 'nın (Σ üzerinde boş olmayan bütün dizgilerin kümesi) bir elemanı, bir *girdi* olarak adlandırılır.
iii) Z bütün tamsayılar kümesi olsun ve H'i aşağıdaki şekilde tanımlayalım.
$H = \{h: Z \to S \mid k_2 \geq 1$ olmak üzere k_1 ve k_2 tamsayıları vardır; $k_1 < i \leq k_1+k_2$ için $h(i) \neq ƀ$ ve diğer durumlarda $h(i) = ƀ\}$
H'nin elemanları Ç'nin *yerleşim*leri olarak adlandırılır. h, Ç'nin bir yerleşimi olsun. Bu yerleşimde h(i), i. hücrenin durumunu verir.
h'nin boşluk olmayan kısmına, $x_1x_2...x_{k2}$, h'nin *etkinliği* denir. Eğer etkinlik girdi ise o zaman yerleşim, *başlangıç yerleşimi*dir.
$h(k_1+k_2) \in F$ ise $h[k_1, k_2]$ yerleşimi bir *kabul*dür ("acceptance"). Kabulün olması için en sağdaki hücrenin sonuç durumuna girmesi gerekir.

Ayrıca bir zincir için en soldaki hücreye göre kabul tanımı da yapılabilir. Eğer bir dizgi veya dil normal bir zincir tarafından kabul ediliyorsa onun tersini sol kabul ile kabul eden bir zincir vardır. Bu sonuç, ileride ortaya koyacağımız teoremin ispatında kullanılacaktır.

Tanım 3.2: E-zinciri ya *zincir*dir ya da *A-zinciri*dir. Ç, bir E-zincir ve $h, h' \in H$ olsun. Eğer Ç bir zincirse o zaman $f(x, y, z) = ƀ \supset y = ƀ$ ve her i için $h \to h'$, $h'(i) = f(h(i-1), h(i), h(i+1))$ olur. Eğer Ç bir A-zincirse o zaman $f(x, y, z) = ƀ \supset z = ƀ$ ve her i için $h \to h'$, $h'(i) = f(h(i-2), h(i-1), h(i))$ olur. Zincirlerin her iki tipi için de eğer $h \to h$ ise h bir *durma yerleşimi*dir. $h \to^k h'$ olması durumunda $h_0, h_1,..., h_k \in H$ olacak şekilde yerleşimler vardır öyle ki $h = h_0$, $h' = h_k$ $k = 0$ veya $0 \leq j < k$ için $h_j \to h_{j+1}$ olur.
Tanım 3.3: Ç'nin w gibi bir girdiyi *kabul* etmesi için $\underline{ƀ}w\underline{ƀ}$ şeklindeki başlangıç yerleşiminin bir kabul vermesi gerekir. Eğer bu kabul, en az k adımda gerçekleşirse w dizgisi *k adımda kabul* edilmiş olur. Boşlukların sonsuz dizgisi $\underline{ƀ}$ ile gösterilmiştir.

e gerçek sayısı için, w, $e \geq 1$ ve $k \leq e|w|$ veya $e < 1$ ve $k \leq |w|+1$ durumunda e-zamanda kabul edilmiştir. Bazen 0,5 zaman, 1-zaman ve 2-zaman yerine *yarı-zaman*, *gerçek-zaman* ve *iki kat zaman* terimleri kullanılır. Hemen hemen yarı-zaman terimi $(0,5 + \varepsilon)$ zaman için kullanılır. Gerçek zamanda kabul, bir dizgiyi uzunluğu kadar adımda kabul etmek demektir.

Tanım 3.4: Σ'yı bir girdi kümesi olarak alalım. Σ^+'nın bir alt kümesi olan L, *dil* olarak adlandırılır. L'in tersi L^T olarak gösterilir ve aşağıdaki şekilde tanımlanır.
Herhangi bir $a \in \Sigma$, $w \in \Sigma^+$ için $a^T = a$ ve $(aw)^T = w^Ta$ olur. Bu durumda $L^T = \{w^T \mid w \in L\}$ dir.
Tanım 3.5: L(Ç), Ç tarafından kabul edilen girdilerin kümesidir. T(Ç), Ç tarafından gerçek zamanda kabul edilen girdilerin kümesidir. e gerçek sayısı için, L dili Ç tarafından e-zamanda kabul edilirse L'in her elemanı Ç tarafından e-zamanda kabul edilir. Eğer $k \geq 1$ koşulunu sağlayan bir k varsa L dili *doğrusal zamanda* kabul edilmiş olur.
L dili, Ç tarafından gerçek zamanda L = L(Ç) = T(Ç) koşulu sağlanıyorsa kabul edilir.

Tanım 3.6: Ç, aşağıdaki koşulu sağlıyorsa *sağdan kısıtlı etkinlik* (SKE) özelliğine sahiptir. h, h′ $\in$ H, eğer $h \rightarrow h'$ herhangi bir i için $h(i\text{-}1) \neq h(i) = ƀ \supset h'(i) = ƀ$.
Eğer herhangi bir i için $h(i) = ƀ \supset h'(i) = ƀ$ oluyorsa bu durumda Ç, *kısıtlı etkinlik* (KE) özelliğine sahiptir. Bu özellikler kısıt özellikleridir.
Not: Eğer bir A-zinciri Ç, SKE özelliğine sahipse o zaman KE özelliğine de sahiptir.

4. BELİRLENİMLİ KISALTICI MEKANİZMASININ ZİNCİR TARAFINDAN GERÇEK ZAMANDA SİMULASYONU

Belirlenimli kısaltıcı mekanizmasının yaptığı tüm hamlelerin zincir tarafından nasıl simule edildiği bu bölümde detaylarıyla ortaya konmuştur. Önce zincir yapısının belirlenimli kısaltıcı mekanizmasını simule ederken belirlenimli kısaltıcının kafası ve girdi şeridinin zincirdeki hücreler tarafından nasıl temsil edildiği anlatılmıştır. Sonra gerçek zamanda kabulü nasıl sağladığı ve simulasyonun genel prensipleri sözel olarak anlatılmıştır. Daha sonra simulasyon ile ilgili teorem ve matematiksel ispatı verilmiştir.

Belirlenimli kısaltıcı mekanizması zincir tarafından iki izli bir teyp kullanılarak gerçek zamanda simule edilmektedir. Girdi $w = a_1a_2...a_{n-1}a_n$ olsun. Birinci ize girdiyi yerleştiriyoruz. İkinci izde ise kısaltıcı mekanizmasının kafasını temsil eden hücre vardır. Bu hücre, bundan sonra kafa hücresi olarak anılacaktır. Kafa hücresi, başlangıçta girdi dizgisinin sol başında yer almaktadır (Şekil 4.1.). Kafa hücresinin durumu başlangıçta ikililer, daha sonra ikili veya üçlüler şeklinde, yığıt kısmında yer alan hücrelerin durumu ikililer, girdi kısmındaki hücrelerin durumu ise bir simgeyle ifade edilmiştir. Kafa hücresinin üstündeki simge, kısaltıcı mekanizmasında kafanın sağında bulunan ilk simge olarak kabul edilmiştir. Kafa hücresi, görevini sol veya sağındaki hücreye aktarmak suretiyle her iki yöne hareket edebilmektedir. Birinci izde kafa hücresine kadar olan kısım, gerçek zamanda kabulü sağlamak üzere her adımda bir simge sağdan sola doğru kaymaktadır. Kafa hücresinin durduğu adımda girdi bir simge kısalmaktadır.

Kafa hücresinin soldan sağa ilerlemesi durumunda simgeleri ikişer ikişer yığıta attığını düşünüyoruz. Kafa hücresinin solundaki (yığıt tarafı) her hücreye iki simge yerleştirir. Kafa hücresi içine bir simgeyi alabilmektedir. Kafa hücresi, bir simgenin üzerinden atladığında bu simgeyi içine alır. Kafa hücresi, yığıttan bir simge silmişse yığıttaki ikinci simgeyi de içine alır. Bu silme işleminin yapıldığı hücre özel bir duruma ("B") girmektedir. Yığıtta böyle bir durum oluştuğunda hücreler, bu durumu her adımda bir hücre sola aktarmak suretiyle etkinlik dışına taşımaktadırlar. Böylece kafa hücresinin solunda (yığıt tarafında) her zaman çift sayıda simge bulunmaktadır. Girdi kısmındaki hücrelerde birer simge, yığıt kısmındaki hücrelerde ikişer simge bulunmaktadır. Dolayısıyla bu durum özel bir simge kullanmadan girdi ve yığıta ait hücreleri ayırt etmeyi sağlar. Bu özellik, sadece girdi tarafındaki simgeleri bir adım sola kaydırmayı mümkün hale getirmektedir.

Kafa hücresi hiç kısaltmadan bu şekilde devam ederse, n/2 adım sonra (girdinin uzunluğu n) girdinin sonuna gelir. Kafa hücresi, bir müddet sağa doğru hareket edip sonra durursa yine birinci durumda olduğu gibi girdiden kısaltmaya devam eder.

Kafa hücresi bir süre sağa hareket eder, sonra sola giderse; bu yığıta atmış olduğu simgeleri ikişer ikişer kısaltması anlamına gelir. Tasarladığımız zincir E-zincir olduğundan komşu hücreler, hücrenin solunda ve sağındaki hücrelerdir.

$\leftarrow$ Girdinin hareket yönü

	ƀ	a_1	a_2	. . . .	a_{n-1}	a_n	ƀ	
		q_0						

Şekil 4.1. Zincir mekanizmasının başlangıçtaki durumu

Şekil 4.1'de kafayı temsil eden hücre sol başta, etkinliğin birinci simgesi olan a_1'i görmektedir. Hiçbir işlem yapılmadığı için yığıtta simge yoktur. Bu durum, kısaltıcı mekanizmasında kafanın girdinin en sol başında bulunduğu duruma karşılık gelir.

← Girdinin hareket yönü

	ƀ	a_2	a_4	. . . .	a_{i-2}	a_i	a_{i+1}	. . . .	a_n	ƀ	
		a_1	a_3	. . . .	a_{i-3}						
						q_i					
						a_{i-1}					

Şekil 4.2. Zincir mekanizmasında kafa bir müddet ilerledikten sonraki muhtemel bir durum

Şekil 4.2.'de kafa hücresi girdiden hiç kısaltmadan a_i simgesine kadar ilerlemiştir. Kafa hücresinin içinde a_{i-1} simgesi vardır. Bu nedenle yığıtta tek sayıda simge bulunmaktadır. Bu durum, kısaltıcı mekanizmasında kafanın sağındaki simgenin a_i, kafanın hemen solundaki simgenin a_{i-1} olduğu duruma karşılık gelir.

Teorem 4.1: K bir belirlenimli kısaltıcı olsun. L(K) dilini gerçek zamanda kabul eden KE bir Ç-zinciri vardır.

İspat: K = (Q, Σ, δ, q_0, F) bir kısaltıcı olsun. Buradan hareketle bir KE-zincir Ç inşa edelim. K'nın her bir an-tarifine karşılık Ç'de onu temsil eden bir etkinlik olacaktır.

- Ç = (Σ, ƀ, Q', S, F', f)
- $Q' = \Sigma \cup \Sigma^2 \cup Q \times \Sigma \cup (\Sigma \times Q) \times \Sigma \cup \{B\}$
- $F' = F \times$ {ƀ}

Zincir yapısındaki hücreler, Q' da bulunan beş ayrı durumdan birinde olabilirler. Bu durumların açıklaması aşağıdadır:

1) Girdinin kafa hücresinin üstünde olan simgesi hariç, girdi simgelerini temsil eden hücrelerin durumları Σ 'nın elemanları ile ifade edilir. Örnek: a_2 simgesi tek başına bir hücre durumuysa, bu durum a_2'nin kısaltıcı kafasının hemen sağında olmayan bir girdi simgesi olduğunu gösterir.

2) Σ^2'nin elemanları ikililer şeklinde olup kafa hücresinin solunda bulunan yığıttaki simgeleri ifade etmekte kullanılır. Kafa hücresinin solunda bulunan simgeler, kısaltıcı kafasının solunda bulunan simgelerdir.

3) $Q\times\Sigma$ kümesinin elemanları olan ikililer, kafa ve gördüğü ilk girdi simgesinin bulunduğu hücrenin durumunu belirler. Örnek: (q, a_1) durumundaki bir kafa hücresi, kısaltıcıda kafanın durumunun q olduğunu, hemen sağındaki simgenin (ilk girdi simgesi) a_1 olduğunu gösterir.

4) $\Sigma\times Q$ kümesinin elemanları üçlüler şeklinde olup kafa hücresinin içine aldığı simgeyi ve baktığı ilk girdi simgesini ifade eder. Örnek: ((a_1, q), a_2) durumundaki bir kafa hücre, kısaltıcı kafasının q durumunda olduğu, girdide a_2 simgesine baktığı ve yığıttaki ilk simge olan a_1'i kafanın içine aldığı durumu temsil eder.

5) B, yığıtın özel bir durumu olup yığıttaki hücrelerin boşaldıklarında girdiği durumdur. Bu duruma giren hücre, her adımda B'yi bir soluna aktarmak suretiyle etkinlik dışına taşır.

Ç zincirinin durum geçiş fonksiyonları dört grupta toplanmıştır. K1 grubunda kısaltıcının bir hamlesine karşılık gelen durum geçiş fonksiyonları, K2 grubunda kısaltıcının iki hamlesine karşılık gelen durum geçiş fonksiyonları bulunmaktadır. Kafa hücresinin içinde yığıt simgesi olup olmadığını kural numarasının yanındaki "b" ve "d" harfleri gösterir. "b" harfi kafa hücresinin boş, "d" harfi ise dolu olduğunu gösterir. Örnek: K1.1b, K1 grubunun 1 nolu kuralı olup kafa hücresinin boş olduğunu göstermektedir.

B grubunda, yığıttaki bir hücrenin "B" özel durumuna girmesi halinde, bu durumun etkinlik dışına taşınmasını sağlayan kurallar vardır. G grubunda, gerçek zamanda kabulü sağlamak üzere teybin birinci izindeki girdiyi her adımda sağdan sola bir simge kaydıran kurallar vardır.

$a_1, a_2, a_3 \in \Sigma$; $p, q, r \in Q$ için Ç zincirinin durum geçiş fonksiyonları aşağıda verilmiştir. Durum geçiş fonksiyon kurallarında, durum simgesi olarak kullanılan "-" simgesi "fark etmez, ilgili komşu herhangi bir durumda olabilir" anlamındadır.

K1 Grubu Kuralları ve Açıklaması:

K1.1b) $\delta(\varepsilon, q, a_1) = (r, .)$ ise her bir $a_2 \in \Sigma \cup \{\text{ƀ}\}$ için $f(-, (q, a_1), a_2) = (r, a_2)$ — Girdiden kısaltma

K1.1d) $\delta(\varepsilon, q, a_2) = (r, .)$ ise her bir $a_3 \in \Sigma \cup \{\text{ƀ}\}$ için $\delta(\varepsilon, r, a_3) = (p, \leftarrow)$ olmayacak şekilde $f(-, ((a_1, q), a_2), a_3) = ((a_1, r), a_3)$ — Girdiden kısaltma (Kafanın içinde simge var)

K1.2b) $\delta(\varepsilon, q, a_1) = (r, \leftarrow)$ ise her bir $a_2 \in \Sigma \cup \{\text{ƀ}\}$ için $\delta(\varepsilon, r, a_2) = (p, \leftarrow)$ olmayacak şekilde $f(-, (q, a_1), a_2) = ((a_1, r), a_2)$ — Girdi simgesini yığıta (kafanın içine) yerleştirme

K1.3b) $\delta(a_2, q, a_3) = (r, .)$ ise her bir $a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(a_1a_2, (q, a_3), a_4) = ((a_1, r), a_4)$, $f(-, a_1a_2, (q, a_3)) = B$ — Girdi ve yığıttan birer simge kısaltma

K1.3d) $\delta(a_1, q, a_2) = (r, .)$ ise her bir $a_3 \in \Sigma \cup \{\text{ƀ}\}$ için $f(-, ((a_1, q), a_2), a_3) = (r, a_3)$ — Girdi ve yığıttan birer simge kısaltma (Kafanın içinde simge var)

K1.4b) $\delta(a_2, q, a_3) = (r, \leftarrow)$ ise her bir $a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(a_1a_2, (q, a_3), a_4) = (r, a_4)$, $f(-, a_1a_2, (q, a_3)) = a_1a_3$ — Girdi simgesinin yığıta yerleştirilmesi ve yığıttan bir simge silinmesi

K1.4d) $\delta(a_1, q, a_2) = (r, \leftarrow)$ ise her bir $a_3 \in \Sigma \cup \{\text{ƀ}\}$ için $f(-, ((a_1, q), a_2), a_3) = ((a_2, r), a_3)$ — Kafanın içindeki simgenin silinmesi ve girdi simgesinin kafanın içine alınması

K1.5b) $\delta(a_2, q, \varepsilon) = (r, .)$ ise her bir $a_3 \in \Sigma \cup \{\text{ƀ}\}$ için $f(-, a_1a_2, (q, a_3)) = ((a_1, r), a_3)$ ve her bir $a_3, a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(a_1a_2, (q, a_3), a_4) = a_4$ — Yığıttan bir simge kısaltma

K2 Grubu Kuralları ve Açıklaması:

K2.1b) $\delta(\varepsilon, q, a_1) = (p, \leftarrow)$ ve $\delta(\varepsilon, p, a_2) = (r, \leftarrow)$ ise $f(-, (q, a_1), a_2) = a_1a_2$ ve her bir $a_3 \in \Sigma \cup \{\text{ƀ}\}$ için $f((q, a_1), a_2, a_3) = (r, a_3)$ — İki simgeyi yığıta yerleştirme

K2.1d) $\delta(\varepsilon, q, a_2) = (p, \leftarrow)$ ve $\delta(\varepsilon, p, a_3) = (r, \leftarrow)$ ise $f(-, ((a_1, q), a_2), a_3) = a_1a_2$ ve her bir $a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(((a_1, q), a_2), a_3, a_4) = ((a_3, r), a_4)$ — İki simgeyi yığıta yerleştirme (Kafanın içinde simge var)

K2.2d) $\delta(\varepsilon, q, a_2) = (p, .)$ ve $\delta(\varepsilon, p, a_3) = (r, \leftarrow)$ veya $\delta(\varepsilon, q, a_2) = (p, \leftarrow)$ ve $\delta(a_2, p, a_3) = (r, \leftarrow)$ ise $f(-, ((a_1, q), a_2), a_3) = a_1a_3$ ve her bir $a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(((a_1, q), a_2), a_3, a_4) = (r, a_4)$ — Birinci simgeyi kısaltıp ikinci simgeyi yığıta yerleştirme (Kafanın içinde simge var)

K2.3d) $\delta(\varepsilon, q, a_2) = (p, \leftarrow)$ ve $\delta(\varepsilon, p, a_3) = (r, .)$ ise her bir $a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(((a_1, q), a_2), a_3, a_4) = (r, a_4)$, $f(-, ((a_1, q), a_2), a_3) = a_1a_2$ — Girdiden birinci simgeyi yığıta yerleştirme, ikinciyi kısaltma (Kafanın içinde simge var)

K2.4d) $\delta(\varepsilon, q, a_2) = (p, .)$ ve $\delta(\varepsilon, p, a_3) = (r, .)$ veya $\delta(\varepsilon, q, a_2) = (p, \leftarrow)$ ve $\delta(a_2, p, a_3) = (r, .)$ ise her bir $a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(((a_1, q), a_2), a_3, a_4) = ((a_1, r), a_4)$, $f(-, ((a_1, q), a_2), a_3) = B$ — Girdiden iki simge kısaltma (Kafanın içinde simge var)

K2.5b) $\delta(a_2, q, \varepsilon) = (p, .)$ ve $\delta(a_1, p, \varepsilon) = (r, .)$ ise her bir $a_3, a_4 \in \Sigma \cup \{\text{ƀ}\}$ için $f(-, a_1a_2, (q, a_3)) = (r, a_3)$, $f(a_1a_2, (q, a_3), a_4) = a_4$ — Yığıttan iki simge kısaltma

K2.5d) $\delta(a_3, q, \varepsilon) = (p, .)$ ve $\delta(a_2, p, \varepsilon) = (r, .)$ ise her bir $a_4, a_5 \in \Sigma \cup \{\text{ƀ}\}$ için $f(-, a_1a_2, ((a_3, q), a_4)) = ((a_1, r), a_4)$, $f(a_1a_2, ((a_3, q), a_4), a_5) = a_5$ — Yığıttan iki simge kısaltma (Kafanın içinde simge var)

K2.6d) $\delta(a_3, q, \varepsilon) = (p, .)$ ve $\delta(a_2, p, a_4) = (r, .)$ ise her bir $a_5 \in \Sigma \cup \{\text{ƀ}\}$ için $f(a_1a_2, ((a_3, q), a_4), a_5) = ((a_1, r), a_5)$, $f(-, a_1a_2, ((a_3, q), a_4)) = B$ — Yığıttan iki simge, girdiden bir simge kısaltma (Kafanın içinde simge var)

K2.7d) $\delta(a_3, q, \varepsilon) = (p, .)$ ve $\delta(a_2, p, a_4) = (r, \leftarrow)$ ise her bir $a_5 \in \Sigma \cup \{\text{ƀ}\}$ için $f(-, a_1a_2, ((a_3, q), a_4)) = a_1a_4$, $f(a_1a_2, ((a_3, q), a_4), a_5) = (r, a_5)$ — Yığıttan iki simge kısaltma, girdi simgesini yığıta alma (Kafanın içinde simge var)

K2 grubundaki durum geçiş fonksiyonlarında, kısaltıcı mekanizmasının iki hamlesi zincir tarafından bir hamlede gerçekleştirilmektedir. Kısaltıcı mekanizması, q durumundan p durumuna, p durumundan da r durumuna geçer. Zincir ise q durumundan r durumuna direk geçer. Bu arada kısaltıcı mekanizması için meydana gelmesi mümkün tüm ihtimaller zincir için söz konusu durum geçiş fonksiyonlarında kapsanmıştır.

B Grubu Kuralları ve Açıklaması:

Ayrıca K1.3b ve K2.4d nolu durum geçiş fonksiyonlarında yığıt B ile ifade edilen özel bir duruma girmektedir. Bu durumda B durumu, her hamlede bir hücre sola kaydırılarak yığıt dışına atılmaktadır. Bu işlem aşağıdaki durum geçiş fonksiyonları ile gerçekleştirilir.

B1) $f(a_1a_2, B, -) = a_1a_2$ — B'nin yığıt dışına kaydırılması
B2) $f(-, a_1a_2, B) = B$ — B'nin yığıt dışına kaydırılması
B3) $f(\text{ƀ}, B, -) = \text{ƀ}$ — B'nin yok edilmesi

G Grubu Kuralları ve Açıklaması:

Girdi dizgisinin ilerletilmesi aşağıdaki durum geçiş fonksiyonları ile gerçekleştirilir.

G1) $f(-, a_1, a_2) = a_2$ — Girdi dizgisinin ilerletilmesi
G2) $f(-, a_1, \text{ƀ}) = \text{ƀ}$ — Girdi dizgisinin ilerletilmesi

İlk hamlede kafa ya simgeyi yok edecek, ya da üzerinden geçecektir. (Kural K1.1b ve K1.2b). Kural (K1.1b), kafa hücresinin hemen sağındaki girdi simgesini yok eder. Kural (K1.2b) kafa hücresinin sağındaki simgeyi yığıt olarak kabul edilen kafa hücresinin içine yerleştirir. Kafa her iki tarafındaki simgeyi yok ederse, kafa hücresinin içinde simge olup olmamasına göre, kural (K1.3b) veya (K1.3d) uygulanır. Sağ tarafındaki simgeleri yok ederse, kafa hücresinin içinde simge olup olmamasına göre, kural (K1.1b) veya (K1.1d), sol tarafındaki simgeyi yok ederse kural (K1.5b) uygulanır. Kafa, sağ tarafındaki simgenin üzerinden atlayıp solundaki simgeyi yok etmişse bu durumu, kafa hücresinin içinde simge olup olmamasına göre, kural (K1.4b) veya (K1.4d) gerçekleştirir. Kural (G1) ve (G2) ise girdi dizgisinin her adımda bir simge sola doğru kaymasını sağlar.

İddia 1: Eğer $q_0a_1a_2...a_n \vdash_K^k upa_i...a_n$ ise $\underline{\text{ƀ}}qa_1a_2...a_n\underline{\text{ƀ}} \to_Ç^{k'} \underline{\text{ƀ}}u(p, a_i)...a_n\underline{\text{ƀ}}$ olduğunu gösterceğiz. Hesaplamanın uzunluğu kısaltıcı için k ile zincir için k' ile gösterilmiştir. İspat k üzerinden tümevarım yoluyla yapılacaktır. Kısaltıcı mekanizmasının hamlelerinden sadece yığıttan bir simge kısaltmak ve girdiden okunacak simgeyi yığıta atmak hamleleri, zincirin yığıt kısmında tek sayıda simge varsa (kafa hücresinin içinde simge olma durumu) kısaltıcının bir sonraki hamlesi de göz önüne alınarak gerçekleştirilir. Bu durumlarda, ispat yapılırken yığıttan bir simge kısaltmak ve okunan bir simgeyi yığıta yerleştirmek kısaltıcının (k+1). hamlesi olarak alınmış, kısaltıcının bu hamleden sonra yapacağı muhtemel hamlelerin hepsi (k+2). hamle olarak ele alınmıştır. Buna karşılık zincirin durum geçişi yazılmıştır. Böylece meydana gelebilecek tüm ihtimaller tüketilmiştir.

Önce k = 1 için olan durum incelenmiştir. Eğer K kısaltıcısının başlangıç an-tarifi $qa_1a_2...a_n$ ise durum geçiş fonksiyonuna bağlı olarak ya a_1 simgesini yok edecek, ya da üzerinden atlayacaktır.

i) $(r, .) \in \delta(\varepsilon, q_0, a_1)$ ise a_1 simgesini yok edecek,

ii) $(r, \leftarrow) \in \delta(\varepsilon, q_0, a_1)$ ise a_1 simgesinin üzerinden atlayacaktır.

Buna karşılık Ç-zincirinde $\underline{\text{ƀ}}(q, a_1)a_2...a_n\underline{\text{ƀ}} \to_Ç \underline{\text{ƀ}}(r, a_2)....a_n\text{ƀ}\underline{\text{ƀ}}$ (kural (K1.1b), (G1) ve (G2)'den dolayı) veya $\underline{\text{ƀ}}(q, a_1)a_2a_3...a_n\underline{\text{ƀ}} \to_Ç \underline{\text{ƀ}}((a_1, r), a_2)a_3...a_{n-1}a_n\text{ƀ}\underline{\text{ƀ}}$ (kural (K1.2b), (G1) ve (G2)'den dolayı) yazılır.

İddianın k hamle için doğru olduğu varsayılıp kısaltıcı mekanizmasının bir sonraki hamlesi için meydana gelmesi mümkün olan tüm ihtimaller Ç-zinciri için gerçekleştirilmiştir. Zincirin durum geçiş fonksiyonları yığıtta tek veya çift sayıda simge olmasına göre yazılmıştır. Yığıtta tek sayıda simge olması kafa hücrenin içinde simge olması anlamına gelmektedir.

Bir sonraki adım için kısaltıcının durum geçiş fonksiyonu $\delta(a_{i-1}, q, a_i) = (r, .)$ ise $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ura_{i+1}...a_n$ olur. Yığıtta çift sayıda simge varsa kural (K1.3b), (G1) ve (G2)'den zincir $ƀu'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'B((a_{i-2}, r), a_{i+1})a_{i+2}...a_nƀƀƀ$ geçişini yapar. Yığıttaki "B" simgesi, kural (B1), (B2) ve (B3) kullanılarak yığıttan atılır. Yığıtta tek sayıda simge varsa kural (K1.3d), (G1) ve (G2)'den $ƀu'(a_{i-3}a_{i-2})((a_{i-1}, q), a_i)a_{i+1}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(a_{i-3}a_{i-2})(r, a_{i+1})a_{i+2}...a_nƀƀƀ$ yazılır.

Eğer $\delta(\varepsilon, q, a_i) = (r, .)$ ise $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-1}ra_{i+1}...a_n$. Yığıtta çift sayıda simge varsa kural (K1.1b), (G1) ve (G2)'den $ƀu'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(a_{i-2}a_{i-1})(r, a_{i+1})a_{i+2}...a_nƀƀƀ$. Yığıtta tek sayıda simge varsa kural (K1.1d), (G1) ve (G2)'den $ƀu'\, a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'a_{i-2}\, ((a_{i-1}, r), a_{i+1})a_{i+2}...a_nƀƀƀ$.

Eğer $\delta(a_{i-1}, q, \varepsilon) = (r, .)$ ise $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ura_ia_{i+1}...a_n$. Yığıtta çift sayıda simge varsa kural (K1.5b), (G1) ve (G2)'den $ƀu'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'((a_{i-2}, r), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ$. Yığıtta tek sayıda simge varsa ve yığıttan bir simge kısaltmak istenirse zincir, kısaltıcının bir sonraki hamlesini de dikkate alarak işlem yapar. Burada, kısaltıcının birinci hamlesi, (k+1). hamle, yığıttan kısaltma olmak üzere yapacağı muhtemel ikinci hamleler, (k+2). hamle, yazılmıştır. İkinci hamle, girdi simgesinin üstünden atlama ise kısaltıcı $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} upa_ia_{i+1}...a_n \vdash_K^{k+2} ua_ira_{i+1}...a_n$ hamlelerini yapar. Buna karşılık kural (K1.4d), (G1) ve (G2)'den zincir için $ƀu'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'((a_i, r), a_{i+1})a_{i+2}...a_nƀƀƀ$ yazılır. İkinci hamle girdiden kısaltma ise kısaltıcı $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} upa_ia_{i+1}...a_n \vdash_K^{k+2} ura_{i+1}...a_n$ hamlelerini yapar. Buna karşılık kural (K1.3d), (G1) ve (G2)'den zincir için $ƀu'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(r, a_{i+1})a_{i+2}...a_nƀƀƀ$ olur. İkinci hamle yığıttan kısaltma ise $ua_{i-2}a_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-2}pa_ia_{i+1}...a_n \vdash_K^{k+2} ura_ia_{i+1}...a_n$. Buna karşılık kural (K2.5d), (G1) ve (G2)'den zincir için $ƀu'(a_{i-3}a_{i-2})((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'((a_{i-3}, r), a_i)a_{i+1}...a_nƀƀƀ$ yazılır. İkinci hamle girdi ve yığıttan birer simge kısaltma ise $ua_{i-2}a_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-2}pa_ia_{i+1}...a_n \vdash_K^{k+2} ura_{i+1}...a_n$. Buna karşılık kural (K2.6d), (G1) ve (G2)'den zincir için $ƀu'(a_{i-3}a_{i-2})((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'B((a_{i-3}, r), a_{i+1})...a_nƀƀƀ$ yazılır. İkinci hamle yığıttan bir simge kısaltıp girdi simgesini yığıta alma ise $ua_{i-2}a_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-2}pa_ia_{i+1}...a_n \vdash_K^{k+2} ua_ira_{i+1}...a_n$. Buna karşılık kural (K2.7d), (G1) ve (G2)'den zincir için $ƀu'(a_{i-3}a_{i-2})((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(a_{i-3}a_i)(r, a_{i+1})...a_nƀƀƀ$ yazılır. Özet olarak zincirde, kısaltıcının ikinci hamlesinde girdiden gelecek simge yığıta atılacaksa kural (K1.4d), kısaltılacaksa kural (K1.3d)'ye göre işlem yapılır. Bir sonraki hamle yine yığıttan silme işlemiyse kural (K2.5d) uygulanır. İkinci hamlede kısaltıcı hem girdiden hem yığıttan kısaltmışsa zincirde kural (K2.6d), yığıttan bir simge kısaltıp girdi simgesinin üstünden atlarsa zincirde kural (K2.7d) uygulanır.

Eğer $\delta(a_{i-1}, q, a_i) = (r, \leftarrow)$ ise $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_ira_{i+1}...a_n$. Yığıtta çift sayıda simge varsa kural (K1.4b), (G1) ve (G2)'den $ƀu'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(a_{i-2}a_i)(r, a_{i+1})a_{i+2}...a_nƀƀƀ$. Yığıtta tek sayıda simge varsa kural (K1.4d), (G1) ve (G2)'den $ƀu'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'a_{i-2}((a_i, r), a_{i+1})a_{i+2}...a_nƀƀƀ$.

Eğer $\delta(\varepsilon, q, a_i) = (r, \leftarrow)$ ise $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-1}a_ira_{i+1}...a_n$. Yığıtta çift sayıda simge varsa kural (K1.2b), (G1) ve (G2)'den $ƀu'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(a_{i-2}a_{i-1})((a_i, r), a_{i+1})a_{i+2}...a_nƀƀƀ$. Yığıtta tek sayıda simge varsa ve bir simge daha yığıta atılmak istenirse zincir, kısaltıcının bir sonraki hamlesini de dikkate alarak işlem yapar. Kısaltıcının birinci hamlesi, (k+1). hamle, yığıta bir simge atma olmak üzere yapacağı muhtemel ikinci hamleler, (k+2). hamle, yazılmıştır. İkinci hamlede girdiden gelecek simge yığıta atılacaksa kısaltıcı $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-1}a_ipa_{i+1}...a_n \vdash_K^{k+2} ua_{i-1}a_ia_{i+1}r...a_n$ hamlelerini yapar. Buna karşılık zincir için kural (K2.1d), (G1) ve (G2)'den $ƀu'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(a_{i-1}a_i)((a_{i+1}, r), a_{i+2})a_{i+3}...a_nƀƀƀ$ olur. İkinci hamlede girdi simgesi kısaltılacaksa kısaltıcı $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-1}a_ipa_{i+1}...a_n \vdash_K^{k+2} ua_{i-1}a_ira_{i+2}...a_n$ hamlelerini yapar. Buna karşılık zincir için kural (K2.3d), (G1) ve (G2)'den $ƀu'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'(a_{i-1}a_i)(r, a_{i+2})a_{i+3}...a_nƀƀƀ$ yazılır. İkinci hamlede girdiden ve yığıttan birer simge kısaltılacaksa kısaltıcı $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-1}a_ipa_{i+1}...a_n \vdash_K^{k+2} ua_{i-1}ra_{i+2}...a_n$ hamlelerini yapar. Buna karşılık zincir için kural (K2.4d), (G1) ve (G2)'den $ƀu'((a_{i-1}, q), a_i)a_{i+1}a_{i+2}...a_nƀƀƀ \rightarrow_Ç^{k'+1} ƀu'B(a_{i-1}, r), a_{i+2})a_{i+3}...a_nƀƀƀ$ yazılır. İkinci hamlede yığıt kısaltılıp (ilk hamlede üzerinden atlanan) ve girdinin üzerinden atlanacaksa kısaltıcı $ua_{i-1}qa_ia_{i+1}a_{i+2}...a_n \vdash_K^{k+1} ua_{i-1}a_ipa_{i+1}...a_n \vdash_K^{k+2} ua_{i-1}a_{i+1}ra_{i+2}...a_n$ hamlelerini yapar. Buna karşılık zincir için kural (K2.2d), (G1) ve (G2)'den ƀ$u'((a_{i-1}, q), a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'(a_{i-1}a_{i+1})(r, a_{i+2})\ldots a_n$ƀƀƀ yazılır. İkinci hamle, yığıttan kısaltmaysa (birinci hamlede üzerinden atladığı simgeyi kısaltması demektir ki) Lemma 3.1'den dolayı, o simgenin bir hamlede kısaltılmasına eşdeğerdir. Bu durumda zincir kural (K1.1d)'yi uygular.□

İddia 2: Eğer ƀ$qa_1a_2\ldots a_n$ƀ $\rightarrow_{Ç}^{k'}$ ƀ$u(p, a_i)\ldots a_n$ƀ ise $q_0a_1a_2\ldots a_n \vdash_K^k upa_i\ldots a_n$ olduğunu göstereceğiz. Hesaplamanın uzunluğu kısaltıcı için k ile zincir için k′ ile gösterilmiştir. İspat k′ üzerinden tümevarım yoluyla yapılacaktır. Zincir yapısındaki bir geçiş için kısaltıcı mekanizmasında karşılık gelen hamle veya hamleler yazılacaktır.

Önce k′ = 1 için zincir, birinci simgeyi kısaltır veya bir ya da iki simgeyi yığıta yerleştirir. Kural (K1.1b)'den dolayı birinci simgeyi kısaltırsa ƀ$(q, a_1)a_2\ldots a_n$ƀ $\rightarrow_{Ç}$ ƀ $(r, a_2)\ldots a_n$ƀƀ, buna karşılık kısaltıcı $(r, .) \in \delta(\varepsilon, q_0, a_1)$'den dolayı a_1 simgesini yok eder. Kural (K1.2b)'den dolayı bir simgeyi yığıta atarsa ƀ$(q, a_1)a_2a_3\ldots a_n$ ƀ $\rightarrow_{Ç}$ ƀ$((a_1, r), a_2)a_3\ldots a_{n-1}a_n$ƀƀ, buna karşılık kısaltıcı $(r, \leftarrow) \in \delta(\varepsilon, q_0, a_1)$'den dolayı a_1 simgesinin üstünden atlar. Kural (K2.1b)'den dolayı iki simge yığıta atılırsa kısaltıcı iki hamle yapar. Zincirdeki ƀ$(q, a_1)a_2a_3\ldots a_n$ƀ $\rightarrow_{Ç}$ ƀ$(a_1a_2)(r, a_3)\ldots a_{n-1}a_n$ƀƀ geçişine karşılık kısaltıcı $(p, \leftarrow) \in \delta(\varepsilon, q_0, a_1)$ ve $(r, \leftarrow) \in \delta(\varepsilon, p, a_2)$ hamlelerini yapar.

İddianın k′ hamle için doğru olduğu varsayılıp Ç-zincirinin bir sonraki yerleşimi için meydana gelmesi mümkün olan tüm ihtimaller kısaltıcı mekanizması için gerçekleştirilmiştir.

Bir sonraki adım için Ç zincirinin durum geçiş fonksiyonu kural (K1.1b), (G1) ve (G2)'den ƀ$u'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'(a_{i-2}a_{i-1})(r, a_{i+1})a_{i+2}\ldots a_n$ƀƀƀ ise kısaltıcı $\delta(\varepsilon, q, a_i) = (r, .)$ den dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_{i-1}ra_{i+1}\ldots a_n$ geçişini yapar. Ç-zincirinde kafa hücresinin içinde simge varsa (kural (K1.1d)), kısaltıcı aynı hamleyi yapar.

Eğer kural (K1.2b), (G1) ve (G2)'den ƀ$u'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'(a_{i-2}a_{i-1})((a_i, r), a_{i+1})a_{i+2}\ldots a_n$ƀƀƀ ise $\delta(\varepsilon, q, a_i) = (r, \leftarrow)$ den dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_{i-1}a_ira_{i+1}\ldots a_n$ olur.

Eğer kural (K1.3b), (G1) ve (G2)'den zincir, ƀ$u'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'B((a_{i-2}, r), a_{i+1})a_{i+2}\ldots a_n$ƀƀƀ ise $\delta(a_{i-1}, q, a_i) = (r, .)$ kuralından dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ura_{i+1}\ldots a_n$ olur. Ç-zincirinde kafa hücresinin içinde simge varsa (kural (K1.3d)), kısaltıcı aynı hamleyi yapar.

Eğer kural (K1.4b), (G1) ve (G2)'den zincir, ƀ$u'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}\ldots a_n$ ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'(a_{i-2}a_i)(r, a_{i+1})a_{i+2}\ldots a_n$ƀƀƀ geçişini yaparsa kısaltıcı $\delta(a_{i-1}, q, a_i) = (r, \leftarrow)$ kuralından dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_ira_{i+1}\ldots a_n$ olur. Ç-zincirinde kafa hücresinin içinde simge varsa (kural (K1.4d)), kısaltıcı aynı hamleyi yapar.

Eğer kural (K1.5b), (G1) ve (G2)'den zincir ƀ$u'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}\ldots a_n$ ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'((a_{i-2}, r), a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ geçişini yaparsa kısaltıcı $\delta(a_{i-1}, q, \varepsilon) = (r, .)$ kuralından dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ura_ia_{i+1}\ldots a_n$ olur.

Eğer kural (K2.1b), (G1) ve (G2)'den zincir ƀ$u'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'(a_{i-2}a_{i-1})(a_ia_{i+1})(r, a_{i+2})\ldots a_n$ƀƀƀ geçişini yaparsa kısaltıcı $\delta(\varepsilon, q, a_i) = (p, \leftarrow)$ ve $\delta(\varepsilon, p, a_{i+1}) = (r, \leftarrow)$ den dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_{i-1}a_i\ pa_{i+1}\ldots a_n\ a_n \vdash_K^{k+2} ua_{i-1}a_ia_{i+1}r\ldots a_n$ yapar. Ç-zincirinde kafa hücresinin içinde simge varsa (kural (K2.1d)), kısaltıcı aynı hamleleri yapar.

Eğer kural (K2.2d), (G1) ve (G2)'den zincir ƀ$u'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'a_{i-2}(a_{i-1}a_{i+1})(r, a_{i+2})\ldots a_n$ƀƀƀ geçişini yaparsa kısaltıcı $\delta(\varepsilon, q, a_i) = (p, .)$ ve $\delta(\varepsilon, p, a_{i+1}) = (r, \leftarrow)$ den dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_{i-1}pa_{i+1}\ldots a_n \vdash_K^{k+2} ua_{i-1}a_{i+1}r\ldots a_n$ yapar.
Eğer kural (K2.3d), (G1) ve (G2)'den zincir ƀ$u'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'a_{i-2}(a_{i-1}a_i)(r, a_{i+2})\ldots a_n$ƀƀƀ geçişini yaparsa kısaltıcı $\delta(\varepsilon, q, a_i) = (p, \leftarrow)$ ve $\delta(\varepsilon, p, a_{i+1}) = (r, .)$ den dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_{i-1}a_ipa_{i+1}\ldots a_n \vdash_K^{k+2} ua_{i-1}a_ir\ldots a_n$ yapar.
Eğer kural (K2.4d), (G1) ve (G2)'den zincir ƀ$u'a_{i-2}((a_{i-1}, q), a_i)a_{i+1}a_{i+2}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'a_{i-2}B((a_{i-1}, r), a_{i+2})\ldots a_n$ƀƀƀ geçişini yaparsa kısaltıcı $\delta(\varepsilon, q, a_i) = (p, .)$ ve $\delta(\varepsilon, p, a_{i+1}) = (r, .)$ den dolayı $ua_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_{i-1}pa_{i+1}\ldots a_n \vdash_K^{k+2} ua_{i-1}r\ldots a_n$ yapar.
Eğer kural (K2.5b), (G1) ve (G2)'den zincir ƀ$u'(a_{i-2}a_{i-1})(q, a_i)a_{i+1}\ldots a_n$ƀƀƀ $\rightarrow_{Ç}^{k'+1}$ ƀ$u'(r, a_i)a_{i+1}\ldots a_n$ƀƀƀ geçişini yaparsa kısaltıcı $\delta(a_{i-1}, q, \varepsilon) = (p, .)$ ve $\delta(a_{i-2}, p, \varepsilon) = (r, .)$ den dolayı $ua_{i-2}a_{i-1}qa_ia_{i+1}a_{i+2}\ldots a_n \vdash_K^{k+1} ua_{i-2}pa_ia_{i+1}\ldots a_n \vdash_K^{k+2} ura_ia_{i+1}\ldots a_n$ yapar. Ç-zincirinde kafa hücresinin içinde simge varsa (kural (K2.5d)), kısaltıcı aynı hamleleri yapar.□

Sonuç: Hesaplamanın uzunluğu üzerinden yapılan ispat her iki yönde de gösterilerek tamamlanmış, belirlenimli K kısaltıcı mekanizmasına karşılık bir zincir inşa edilmiştir.

5. SİMULASYON İLE İLGİLİ ÖRNEKLER

Örnek 1: Dengeli Parantezler

Dengeli parantezler kümesini kabul eden K1 = ({q}, {[,]}, δ , q, {q}) alalım.

Durum geçiş fonksiyonu:

1) δ(ε, q, [ ) = {(q, ←)},
2) δ([, q,]) = {(q, .)}.

Buna karşılık aynı kümeyi kabul eden Ç'yi inşa edelim.
Ç = (Σ, ƀ, Q′, S, F′, f)
Σ = {[,]}
F′ = {(q, ƀ)}
Q = {q}
$Q' = \Sigma \cup \Sigma^2 \cup Q \times \Sigma \cup (\Sigma \times Q) \times \Sigma \cup \{B\}$

Dengeli parantezlere ilişkin durum geçiş fonksiyonları yazılırken zincir yapısına ait kullanılan kuralların numaraları yanına yazılmıştır.

1) f (-, (q, [ ), *) = (([, q), *) * ∈ {[,]} —Kural K1.2b
2) f (-, (([, q), [ ), [) = ([[) —Kural K2.1d
3) f (([, q), [ ), [, *) = ([, q), *) * ∈ {[,]} —Kural K2.1d
4) f (([, q), [ ),], *) = (([, q), *) * ∈ {[,], ƀ} —Kural K2.4d
5) f(-, ([, q), [),]) = B —Kural K2.4d
6) f ([[,(q,]), *) = (([, q), *) * ∈ {[,], ƀ} —Kural K1.3b
7) f (-, [[, (q,])) = B —Kural K1.3b
8) f (-, ([, q),]), *) = (q, *) * ∈ {[,], ƀ} —Kural K1.3d

Yukarıdaki durum geçiş fonksiyonlarına ilave olarak B'nin yığıt dışına atılması ve girdi dizgisinin ilerletilmesine ilişkin durum geçiş fonksiyonları vardır.

Ç'nin [[ ] [ ]] dizgisi üzerindeki hesaplamasına bakalım.

b̲ (q, [ )[ ] [ ]] b̲ $\rightarrow_{Ç}$ b̲(([, q), [)] [ ]]ƀb̲ $\rightarrow_{Ç}$ b̲B(([, q), [)]] ƀƀb̲ $\rightarrow_{Ç}$ b̲ƀB(([, q),])ƀƀƀb̲ $\rightarrow_{Ç}$ b̲ƀƀ(q, ƀ)ƀƀƀb̲

Ç zincirinde önce birinci ([) sol parantez yığıta alınır (1. kural). Zincir yapısında kafa hücresi, iki girdi simgesine bakabildiği için sağındaki girdi dizgisinden aynı anda iki simgeyi kısaltabilir veya yığıta atabilir. Dolayısıyla kısaltıcı mekanizmasının iki hamlede yaptığını bir hamlede yapar. Zincir yapısında kafa hücresinin sağında hem sol hem de sağ parantez varsa yığıta koymadan ikisini de kısaltabilir (4.kural). Son olarak girdiden gelen sağ parantez ile yığıttaki sol parantez yok edilir (8.kural). Kısaltıcı mekanizmasının altı hamlede kabul ettiği dizgiyi, zincir dört geçiş sonucunda kabul etmiştir.

Örnek 2: Palindromlar

Önce, örnekte kullanılacak palindrom kavramı tanımlanmıştır. Palindrom, başından sonuna ve sonundan başına okunuşları aynı olan dizgilerdir. Palindromun resmi tanımı (8) şöyledir:

1) ε (boş dizgi) bir palindromdur.
2) a bir simge ise a'dan oluşmuş tek simgelik dizgi bir palindromdur.
3) a bir simge ve x bir palindrom ise axa bir palindromdur.
4) Yukarıdakiler dışında hiçbir şey palindrom değildir.

Palindrom örnekleri: kayak, kabak, mum, talat, 11011011.
Σ alfabesi üzerinde ortası belirli palindromlar kümesi $L_p = \{w\#w^T \mid w \in \Sigma^*\}$ olur.

Durum geçiş fonksiyonu:

1) $a \in \Sigma$ olmak üzere $\delta(\varepsilon, q, a) = \{(q, \leftarrow)\}$,
2) $\delta(\varepsilon, q, \#) = \{(r, .)\}$,
3) $a \in \Sigma$ olmak üzere $\delta(a, r, a) = \{(r, .)\}$.

Buna karşılık aynı kümeyi kabul eden Ç'yi inşa edelim.

Ç = $(\Sigma \cup \{\#\}$, ƀ, $Q', S, F', f)$
$F' = \{(r,$ ƀ$)\}$
$Q = \{q, r\}$
$Q' = \Sigma \cup \{\#\} \cup \Sigma^2 \cup Q \times \Sigma \cup (\Sigma \times Q) \times \Sigma \cup \{B\}$

Durum geçiş fonksiyonları:

a1, a2, a3, a4 olmak üzere,

1) $f(-, (q, a_1), a_2) = ((a_1, q), a_2)$ her $a_1, a_2 \in \Sigma$ için — Kural K1.2b
2) $f(-, ((a_1, q), a_2), a_3) = a_1a_2$ her $a_1, a_2, a_3 \in \Sigma$ için — Kural K2.1d
3) $f(((a_1, q), a_2), a_3, a_4) = ((a_3, q), a_4)$ her $a_1, a_2, a_3, a_4 \in \Sigma$ için —Kural K2.1d
4) $f(-, ((a_1, q), \#), a_1) = ((a_1, r), a_1)$ her $a_1 \in \Sigma$ için — Kural K1.1d
5) $f(-, ((a_1, q), a_2), \#) = a_1a_2$ her $a_1, a_2 \in \Sigma$ için — Kural K2.3d
6) $f(((a_1, q), a_2), \#, a_2) = (r, a_2)$ her $a_1, a_2 \in \Sigma$ için — Kural K2.3d
7) $f(-, ((a_1, r), a_1), a_2) = (r, a_2)$ her $a_1 \in \Sigma$, $a_2 \in \Sigma \cup \{$ƀ$\}$ için —Kural K1.3d
8) $f(a_1a_2, (r, a_2), a_1) = ((a_1, r), a_1)$ her $a_1, a_2 \in \Sigma$ için — Kural K1.3b
9) $f(-, a_1a_2, (r, a_2)) = B$ her $a_1, a_2 \in \Sigma$ için — Kural K1.3b

Yukarıdaki durum geçiş fonksiyonlarına ilave olarak B'nin yığıt dışına atılması ve girdi dizgisinin ilerletilmesine ilişkin durum geçiş fonksiyonları vardır.

Ç'nin 1101#1011 dizgisi üzerindeki hesaplamasına bakalım.

ƀ(q, 1)101#1011ƀ $\rightarrow_{Ç}$ ƀ((1, q), 1)01#1011ƀƀ $\rightarrow_{Ç}$ ƀ(11)((0, q), 1)#1011ƀƀƀ $\rightarrow_{Ç}$ ƀ(11)(01)(r,1)011ƀƀƀƀ $\rightarrow_{Ç}$ ƀ(11)B((0, r),0)11ƀƀƀƀƀ $\rightarrow_{Ç}$ ƀB(11)(r,1)1ƀƀƀƀƀƀ $\rightarrow_{Ç}$ ƀƀB((1, r),1)ƀƀƀƀƀƀƀ $\rightarrow_{Ç}$ ƀƀƀ(r, ƀ)ƀƀƀƀƀƀƀ

Ç zincirinde ilk "1" kafa hücresinin içine alınır (1.kural). İkinci ve üçüncü simgeler ("1" ve "0") bir defada yığıta alınır (2.ve 3.kurallar). Dördüncü simge ("1") yığıta alınırken # simgesi yok edilir ve kafa hücresi r durumuna girer (5.ve 6.kurallar). Girdi ve yığıttan bir simge yok eder, yığıtta tek kalan simgeyi kafa hücresinin içine alır ve yığıtta "B" özel durumu oluşur (8.ve 9.kurallar). Kafa hücresinin içinde simge olduğu durumda girdiden ve yığıttan silmek için 7.kural kullanılır.

6. SONUÇ

Bu makalede, kısaltıcı mekanizmasının "zincir" adı verilen bir boyutlu hücresel hareketliler tarafından gerçek zamanda simulasyonu yapılmıştır. Simulasyonda belirlenimli bir kısaltıcı mekanizması alınmış, buna karşılık bir zincir inşa edilmiştir. Kısaltıcının tüm hamleleri zincirde temsil edilmiştir. Zincirde gerçek zamanda kabulü sağlamak üzere girdi dizgisi her adımda bir hücre sağdan sola ilerletilmiştir. Zincirde yığıt olarak kullanılan hücrelere (kafa hücresinin solu) ikişer simge yerleştirilmiştir. Yığıtta oluşan boşluklar "kabarcık" yöntemiyle yığıt dışına çıkarılmıştır. Böylece zincir, kısaltıcının yaptığı tüm hamleleri simule etmiştir. Zincir, yapısından dolayı kısaltıcı mekanizmasından daha hızlı çalışmaktadır. Bazı durumlarda kısaltıcı mekanizmasının iki hamlesi, zincirde bir geçişle karşılanmıştır.

Makalede ortaya konan simulasyon tekniği ile dengeli parantezler ve ortası belli palindromlar kümelerini üreten dillerin zincir tarafından kabul edildiği gösterilmiştir. Bu diller, zincir tarafından kısaltıcı mekanizmasına göre daha az geçiş sonucu kabul edilmiştir. Bu duruma örnek olarak, zincirin dengeli parantezleri kabul ederken girdide sol ve sağ parantez gördüğünde yığıta koymadan bu simgeleri silmesi verilebilir. Kısaltıcı, böyle bir durumda önce sol parantezi yığıta alır, sonra sağ parantezle birlikte siler.

KAYNAKLAR

1. Çalıkoğlu, D., "On the computational properties of a class of cellular automata", Ph. D. Thesis, ***The University of Iowa***, Iowa, USA, 1,2,8-14,63-67 (1973).
2. Dyer, C.R., "One-way bounded cellular automata", ***Information and Control***, 44, 261-281 (1980).
3. Iwamoto, C. et al., "Constructible functions in cellular automata and their applications to hierarchy results", ***Theoretical Computer Science***, 270, 797 - 809 (2002).
4. Buchholz, T. and Kutrib, M., "On time computability of functions in one-way cellular automata", ***Acta Informatica***, 35, 329-352 (1998).
5. Durand-Lose, J.O., "Reversible space-time simulation of cellular automata", ***Theoretical Computer Science***, 246, 117-129 (2000).
6. Oğuztüzün, H., "Shrinkers: a class of recognizers for a subclass of context-free languages", Yüksek Lisans Tezi, ***Bilgisayar Mühendisliği, ODTÜ***, Ankara, 16-25 (1984).
7. Çiftçi, Z., "Kısaltıcı Mekanizmasının Bir Boyutlu Hücresel Hareketlilerde Gerçek Zamanda Simulasyonu", Doktora Tezi, ***Elektronik ve Bilgisayar Eğitimi, Gazi Üniversitesi***, Ankara, 27-42 (2002).
8. Hopcroft, J.E. and Ullman, J.D., "Introduction to automata theory, languages, and computation", ***Addison Wesley Publishing Company***, USA, 11 (1979).

Geliş Tarihi:26.06.2002 *Kabul Tarihi:07.05.2003*