

# Kilitlenebilir denetimsel gözetleyicilerin performansının iyileştirilmesi

Özgür Turay KAYMAKÇI\*, Salman KURTULAN

İTÜ Fen Bilimleri Enstitüsü, Kontrol ve Otomasyon Mühendisliği Programı, 34469, Ayazağa, İstanbul

## Özet

Son otuz yıl içerisinde bilgisayar, haberleşme ve elektronik teknolojisindeki hızlı gelişimin bir sonucu olarak yeni bir tür dinamik sistem ortaya çıkmıştır. Öyle ki bu tür dinamik sistemler incelendiğinde, büyük bir bölümünün çoğu zaman tamamının ayırık, zamandan bağımsız değişkenlerle kontrol edildiği görülmektedir. Üretim sistemleri, bilgisayar sistemleri, haberleşme sistemleri, hava trafik sistemleri gibi birçok sistem bu tür sistem sınıfına girmektedir. Bu tür sistemlerin davranışı genellikle eş zamansız olarak oluşan ayırık olaylara bağlı olarak değişir. Bu özellikleri nedeniyle bu sistemler Ayırık Olay Sistemleri olarak adlandırılmaktadır. Ayırık olay sistemi çok fazla olası kilitlenme içerdiğinde denetimsel gözetleyici olarak en az kısıtlamalı kilitlenmesiz çözümü seçmek tutucu bir çözüm oluşturabilir. O zaman toplam performansı arttırmak için kilitlenmesiz olma koşulunu esnetmek kaçınılmaz bir olgudur. Diğer bir taraftan tam başarılı çözümü denetimsel gözetleyici olarak seçmek bazı olası kilitlenmelerden dolayı ciddi sistem arızalarına sebebiyet verebilir. O zaman bu iki sonucu bağlayan bir denetimsel gözetleyiciye ihtiyaç vardır. Bu yüzden bu çalışmada optimizasyon yaklaşımını kullanarak kilitlenme ve başarımlar arasındaki denge araştırılmıştır. İlk olarak kilitlenme ve başarımlara karşılık gelen kelimelerin nümerik değerlerine dayanan yeni bir performans ederi ortaya çıkartılmıştır. Önerilen formülasyon klasik optimizasyon yaklaşımını barındıran temel değiş tokuş özelliğine sahiptir. Aynı zamanda istenilen işaretli dili olabildiğince üreten optimal kilitlenebilir çözümü araştıran yeni bir algoritma önerilmiştir.

**Anahtar Kelimeler:** Ayırık olay sistemleri, kilitlenme, düzenli diller, performans değerlendirme.

\*Yazışmaların yapılacağı yazar: Özgür Turay KAYMAKÇI. kaymakcio@itu.edu.tr; Tel: (212) 285 67 12.

Bu makale, birinci yazar tarafından İTÜ Fen Bilimleri Enstitüsü, Kontrol ve Otomasyon Mühendisliği Programı'nda tamamlanmış olan "Ayırık olay sistemleri için optimal denetimsel gözetleyici tasarımı" adlı doktora tezinden hazırlanmıştır. Makale metni 28.12.2007 tarihinde dergiye ulaştırılmış, 30.01.2008 tarihinde basım kararı alınmıştır. Makale ile ilgili tartışmalar 31.01.2010 tarihine kadar dergiye gönderilmelidir.

## Improving the performance of blocking supervisors

### Extended abstract

*We can not model discrete event systems by employing ordinary differential equations, this being their most important difference from time-varying dynamic systems. Rather discrete event systems evolve in time in the form of events occurring at possibly irregular time intervals. Therefore, we often model them as regular languages represented by deterministic finite state automata. In addition, in order to examine the behavior of the system, we modify the behavior by a control action. Ramadge and Wonham's supervisory control theory is a powerful tool to build variety of modifications in the model through a supervisor. However, the supervisor can not usually mark all the desired behaviors of the system, which are built according to nonblocking and controllability constraints. Thus blocking becomes a crucial task in discrete event systems. Industrial examples give support to this observation too. We observe that supervisory control problems with blocking are generally more widespread than those without blocking, and blocking supervisors are preferred to nonblocking ones in several applications. Among these applications, database concurrency control, the protocols coupled with deadlock detection schemes are most popular since deadlock prevention and avoidance are impractical.*

*When we examine a discrete event system within the corresponding admissible language, too many blockings may be realized. We then usually select minimally restrictive nonblocking solution as the supervisor, even though it is inadequate due to its restrictive behavior. Since it prevents all uncontrollable events that lead to blocking minimal restrictive nonblocking solution can provide a conservative result. As a result, the supervisor can generate only a small part of admissible marked language. As such this strategy may constrain the behavior of the system considerably. On the other hand, the completely satisfying solution generates all the admissible marked strings, but its price adds too many blockings to the supervised language. Therefore, neither of these two results can be acceptable in many discrete event system problems. Here the success of the supervisor is related to the generated admissible*

*marked language. Then it will be of interest to relax the nonblocking requirement and consider the synthesis of blocking supervisors. What motivates us to consider blocking solutions is that by allowing a certain amount of blocking, we can increase the part of admissible marked language that can be achieved under control. At this point the following question needs to be answered. "How many blockings do arise?" This work aims to find a formal answer to this question.*

*In the literature blocking supervisors were first studied by Chen and Lafortune. If possible, the proposed solution may improve the performance either by reducing the supervised language's blocking without affecting its achievement or by increasing the supervised language's achievement without affecting its blocking. But, the literature did not concern with improving the performance by reducing the blockings and achievements at the same time. Furthermore, the generated strings on the system have different meanings in practice. For example, some strings, which symbolize critic tasks, have high importance compared to other strings. Thus the differences between the strings have to be taken into consideration to select the best supervisor.*

*In this work, we propose a new approach to overcome these drawbacks. Firstly, we suggest a new metric space and introduce a new performance measure including blocking. The performance of blocking supervisors hold on two concepts: blocking and failure. The defined performance measure captures the fundamental concept, as it is given over sum of blocking measure and non-satisfying measure. Afterwards we propose a new method to select an optimal blocking but also maximally permissive supervisor from the acceptable solution set. The proposed algorithm removes some strings in a sequence from the initial language in order to optimize the performance. As it is usually required to generate as many as possible admissible marked strings, the complete satisfying solution is selected as the initial language. Also it is proved that the final solution obtained will be the maximally permissive and the optimal blocking supervisor. Also, the method is explained over a database management system.*

**Keywords:** Discrete event systems, blocking, regular languages, performance evaluation.

## **Giriş**

Kapsadığı uygulama alanındaki beklentilerin her geçen gün biraz daha artması ve bu alandaki hızlı gelişmenin sonucunda, ayrık olay sistemleri yeni, etkileyici ve ümit vadeden bir araştırma alanı olarak hem akademisyenlerin hem de mühendislerin ilgisini çekmektedir. Bu bağlamda bu güne kadar farklı çalışma grupları tarafından ayrık olay sistemlerini modellemek ve elde edilen modeller üzerinden analiz yapabilmek için değişik yöntemler önerilmiştir. Bu sistemleri “kontrol etmek” için ise genellikle Ramadge ve Wonham tarafından temelleri atılan denetimsel gözetleyici kuramı bünyesinde elde edilen sonuçlar kullanılmakta ve bu kuram üzerinden denetimsel gözetleyici yapıları önerilmektedir (Ramadge ve Wonham 1987; Ramadge ve Wonham 1989). Bu kuramın akademik ve mühendislik çevreleri tarafından kabul görmesindeki en büyük etken, klasik kontrol kuramındaki kavramların bu kuram ile ayrık olay sistemlerine taşınmış olması ve kontrol kuramı bakış açısıyla sistemi analiz etmenin mümkün olmasıdır.

Denetimsel gözetleyici altında çalışan sistemin davranışının her zaman hem kontrol edilebilir hem de kilitlenmesiz olması istenir. Fakat bu koşulları sağlayan bir denetimsel gözetleyici olmayabilir ya da bu koşulları sağlayan bir denetimsel gözetleyici olsa bile tutucu bir davranışa sahip olabilir. Bu noktada bir çözüm elde edebilmek ya da tutucu davranışa sahip olan bu denetimsel gözetleyiciden daha az tutucu bir denetimsel gözetleyiciyi seçebilmek için kilitlenme ve/veya kontrol edilebilirlik koşullarının yeniden değerlendirilmesi gerekir. Kontrol edilebilirlik koşulu denetimsel gözetleyiciler için vazgeçilmezdir ve kontrol edilemeyen bir davranışa sahip çözüm gerçekleştirilebilir bir çözüm olmadığından kesinlikle kabul edilebilir değildir. O zaman bu tür ayrık olay sistemlerinde beklenen davranışa yakınsayabilmek için sistemin belli bir oranda kilitlenmesine izin verilmelidir.

Kilitlenmenin kolaylıkla algılanabildiği ve çözülebildiği veri yönetim sistemleri gibi sistemlerde ise kilitlenmeye genellikle izin verilir. Bu sebepten bu tür sistemlerde kilitlenmesiz çö-

zümünden çok istenilen davranışa daha yakın fakat kilitlenebilir çözümler tercih edilir. Fakat kilitlenmesiz olma koşulunu yumuşatmak, tüm kilitlenmelere denetimsel gözetleyici tarafından izin verileceği anlamına gelmez. Sistemdeki kilitlenmeler belli ölçülere göre değerlendirilmelidir. Öyle ki bir kilitlenmeye izin verilmesi durumunda, bu kilitlenmenin sisteme olan etkisi ve kazandırılan yeni davranışın istenilen davranışa olan yakınlığı beraber incelenmelidir. Genellikle kilitlenme ile istenilen davranışın elde edilen kısmı arasında bir değiş tokuş vardır. Bu noktada “Ne oranda bir kilitlenmeye izin verilecek?” sorusunun yanıtı aranmaktadır. Bu çalışmanın temel amacı da kilitlenebilir denetimsel gözetleyici problemini araştırmak ve bu probleme biçimsel bir cevap bulmaktır. Bu çalışmada sisteme ait tüm çözümlerin sonlu sayıda kelimedenden oluştuğu varsayılmıştır.

Literatürde optimizasyon problemine farklı açılardan yaklaşılmış ve çözüm bulunmaya çalışılmıştır. Ayrık olay sistemlerinde optimum denetimsel kontrol ilk olarak Ratnesh Kumar ve Vijay K. Garg tarafından da ele alınmıştır. Bu çalışmada optimizasyon yöntemi olarak ağ akış teknikleri kullanılmıştır (Kumar vd. 1993). Passino ve Antsaklis tarafından sistemdeki olay geçişleri üzerinden yeni bir eder fonksiyonu tanımlanmış, sistemin davranışını kısıtlayacak şekilde bir kontrol amacı ortaya atılmış ve optimizasyon problemi bir yörünge izleme problemine dönüştürülmüştür (Passino ve Antsaklis 1989; Passino ve Antsaklis 1990). Diğer taraftan Brave ve Heymann tarafından ayrık olay sistemlerinde optimizasyon, bir optimal çekim problemi olarak tanımlanmış ve sistemin belli bir durum kümesine yakınsaması ve orada kararlı kılınması olarak ifade edilmiştir. (Brave ve Heymann 1990; Brave ve Heymann 1993). Sengupta ve Lafortune tarafından Kumar ve Garg’ın çalışmasına benzer olarak kontrol edilen sistemin ederini belirlemek üzere kontrol ve eder fonksiyonları tanımlanmıştır. Sistemin istenilen davranış kümesi içinde en fazla davranışına izin veren minimum dili bulmayı amaçlamışlardır. Optimizasyon yöntemi olarak dinamik programlama yapılmıştır (Sengupta ve Lafortune 1998). Bunların yanı

sıra kilitlenebilir denetimsel gözetleyici problemi ise ilk olarak Chen ve Lafortune tarafından ifade edilmiştir (Chen ve Lafortune 1991).

Bu çalışma dört bölümden oluşmaktadır. İkinci bölümde ayrık olay sistemleri ile ilgili temel kavramlar aktarılmış, üçüncü bölümde kilitlenebilir denetimsel gözetleyici problemi detaylandırılmış ve istenilen dile en yakın ve en düşük performans ölçüsüne sahip dilin nasıl elde edilebileceği önerilmiştir. Dördüncü bölümde ise sonuçlar verilmiştir.

### Temel kavramlar

Ayrık olay sistemlerinin davranışını incelemek ve modellemek için geliştirilmiş biçimsel yollardan birisi de diller ve otomatlardır. Bu kuramsal yapının başlangıç noktası, sistemi süren sonlu elemanlı olay kümesidir ve  $\Sigma$  ile gösterilir, öyle ki bu küme  $\Sigma = \Sigma_c \cup \Sigma_{uc}$  ( $\Sigma_c \cup \Sigma_{uc} = \{\}$ ) olmak üzere kontrol edilebilen ve kontrol edilemeyen iki farklı kümenin bir birleşimidir. Olay kümesindeki bütün elemanların sonlu sayıda birbiri ardına eklenmesiyle oluşan küme olay kümesinin Kleene kapanışıdır ve  $\Sigma^*$  ile gösterilir. Bu kümenin her bir elemanına da kelime denir.  $L \subseteq \Sigma^*$  ile  $\Sigma$  üzerine tanımlı bir dil ifade edilir.  $\bar{L}$  ile dilin önek kapalı hali gösterilir.

Deterministik sonlu durumlu otomat,  $G = (X, \Sigma, f, \Gamma, x_0, X_m)$  altılısı ile ifade edilmektedir. Burada  $X$  ile otomata ait durum kümesi,  $\Sigma$  ile de olay kümesini ifade edilmektedir.  $f: X \times \Sigma \rightarrow X$  ile durum geçiş fonksiyonu,  $\Gamma: X \rightarrow 2^\Sigma$  ile de aktif olay fonksiyonunu tanımlanır.  $x_0$  otomata ait ilk durumu göstermektedir.  $X_m \subseteq X$  ise işaretlenmiş durum kümesidir.  $G$  otomatının ürettiği dil ve işaretlediği diller sırasıyla  $L(G)$  ve işaretlediği dil  $L_m(G)$ 'dir. Eğer  $L(G) \neq \bar{L}_m(G)$  ise otomatta kilitlenme söz konusudur. Aynı zamanda  $p_j(s)$  fonksiyonu  $s$  kelimesinin  $j$  uzunluklu önekini ve  $q_i(s)$  fonksiyonu ise  $s$  kelimesinin  $i$ . olayını verir. Mesela  $s = e_1 e_2 e_1 e_3$  için  $p_3(s) = e_1 e_2 e_1$  ve  $q_3(s) = e_1$ 'dir.

Okuyucunun bundan sonra yazılanları rahatlıkla takip edebilmesi için ayrık olaylı sistemleri modellemek ve kontrol etmek için ortaya atılan ve sıklıkla tercih edilen Ramadge-Wonham yaklaşımı hakkında temel seviyede bilgiye sahip olduğu kabul edilmiştir. İstenirse detaylı bilgi için (Cassandras ve Lafortune 1999; Caillaud vd. 2002) başvurulabilir.

### Optimal kilitlenebilir denetimsel gözetleyici tasarım problemi

Çok fazla kilitlenme içeren ayrık olay sistemlerinde sistem modellerini kontrol edecek olan denetçinin yapısı ilk olarak Chen ve Lafortune tarafından incelenmiştir (Chen ve Lafortune 1989; Chen ve Lafortune 1991). Çalışma kapsamında temel kilitlenebilir denetimsel gözlemleyici tasarım problemi ortaya koyulmuş ve çözüm olarak iki işlem önerilmiştir. Teori gereği işlemlerin uygulanmasıyla ulaşılan son dil kapsama anlamında en iyi çözüm olduğundan farklı bir başlangıç dili ile ulaşılan son dil daha önce elde edilen dilden genellikle farklı olacaktır. Burada elde edilen çözümler küme teorisine göre kapsama anlamında elde edildiğinden birbirleriyle karşılaştırılması olanaksızdır. Elde edilen çözüm, teoriye göre en iyi çözümdür fakat çoğu zaman bu yeterli değildir. Bunun sebebi elde edilen çözüme ait bütün kelimelerin sistem için aynı anlamı taşımasıdır. Fakat gerçek hayatta her bir olayın ederi aynı olmadığı gibi her bir kilitlenmenin ve başarılmanın de eri aynı değildir. Bunu dikkate alıp belirtilen performans ölçüleri içerisinde çözüm bulmaya çalışan yeni bir algoritma vermek bizi daha anlamlı çözümlere götürecektir.

Yukarıda genel olarak ifade edilen kilitlenebilir denetimsel gözetleyici tasarım problemi için aşağıda verilen ön koşullar dâhilinde çözüm aranacaktır:

$$L_m(S/G) := L(S/G) \cap L_m(G),$$

$$L_m(S/G) \subseteq L_{am}, \quad L(S/G) \subseteq L_a = \bar{L}_a \subseteq L(G),$$

$$L_{am}^{\downarrow C} \subseteq L_a \text{ ve } L_{am} = L_a \cap L_m(G) = L_a \cap L_{am} \text{ 'dir.}$$

Buna göre  $L_{am}$  ve  $L_m(G)$  kapalıdır. Olası tüm çözümleri içeren dil ailesi kümesi ise

$$L_{cand} := \{\Theta : (L_{am}^{\uparrow C} \subseteq \Theta L_{am}^{\downarrow C}) \wedge (\bar{\Theta} \Sigma_{uc} \cap L(G) \subseteq \bar{\Theta})\}$$

şeklinde verilir. Bir  $S$  denetimsel gözetleyicisi

ile çalışan  $G$  sisteminde sistemi kilitlenmeye götüren kelimeler ve başarısız kelimeler sırasıyla  $BM(L(S/G)) := \{L(S/G) \setminus \overline{L_m(S/G)}\}$  ve  $SM^C(L(S/G)) := \{\overline{L_{am} \setminus L_m(S/G)}\}$ 'dir.  $L(S/G)$  olası tüm çözüm kümesinin bir elemanı olduğundan düzenli bir dildir ve sonuçta  $BM(L(S/G))$  ve  $SM^C(L(S/G))$  kümeleri sonlu elemanlı iki kümedir. Kilitlenebilir denetimsel gözetleyici problemi için önerilen çözümlerde üretilen kelimelerin hepsi aynı öneme sahipmişçesine değerlendirilmekte, onların sistem üzerindeki etkileri göz ardı edilmektedir. Bu aşamada kelimelerin önem farklılaşmasını içerecek yeni bir yapı ortaya koyulacaktır.

### Olay ederi

$G$  deterministik sonlu durumlu bir otomat,  $\Sigma$  bu otomata ilişkin olay kümesi,  $s \in \Sigma^*$  ve  $e_i \in \Sigma$  herhangi bir olay olmak üzere olmak üzere bu olayın ederi  $c_e$  olarak verilir öyle ki  $c_e(s, e_i) \rightarrow \mathfrak{R}^+$ .

Bir olay bir otomatta birden fazla durumda aktif olabilir ve her aktif olduğu durumda aynı edere sahip olmak zorunda değildir.  $c_e$  olay ederi ile sadece olayın oluşmasının ederi ortaya konulmamış aynı zamanda olayın hangi kelimedenden sonra oluştuğunun bilgisi de tanıma eklenmiştir. Ayrıca  $c_e(s, \varepsilon) = \varepsilon$  olarak kabul edilmiştir (Kaymakci vd., 2005).

### Kelime ederi

$G$  deterministik sonlu durumlu bir otomat,  $\Sigma$  bu otomata ilişkin olay kümesi ve  $s_k, s_l \in \Sigma^*$  bu otomatın ürettiği iki farklı kelime olmak üzere bir kelimenin ederi  $c_s(s_k, s_l) : \Sigma^* \times \Sigma^* \rightarrow \mathfrak{R}^+$  öyle ki

$$c_s(s_k, s_l) := \sum_{i=1}^{\|s_l\|} \frac{c_e[s_k \cdot p_{i-1} s_l, q(s_l)]}{\|s_l\|} \quad (1)$$

Burada  $c_s(s_k, s_l)$  ile  $s_l$  kelimesinin  $s_k$  kelimesinden sonra üretilmesinin sisteme olan ederi ifade edilmiştir.

### Dilin ederi

$L = \{s_1, s_2, \dots, s_n\} \in \Sigma^*$  döngüsel olmayan bir dil olmak üzere dilin ederi

$\beta(L) : (\Sigma^*)^n \rightarrow \mathfrak{R}^+ + \{0\}$  şeklindedir, öyle ki

$$\beta(L) := \begin{cases} \sum_{i=1}^n c_s(s_i) & L \neq \{\} \\ 0 & L = \{\} \end{cases}$$

Bir dilin ederi kendisini oluşturan kelimelerin ederlerinin toplamı olacak şekilde verilmiştir. Her bir kelimenin ederi pozitif olduğundan dilin ederi de her zaman pozitif olacaktır. Ancak elde edilen pozitif sayısal değer herhangi bir belirleyici özelliğinin olabilmesi için metrik uzay gibi biçimsel bir yapı kurmak gerekir (Kaymakci ve Kurtulan 2009).

### Mesafe fonksiyonu

$\forall L_1, L_2 \in \Sigma^*$  olmak üzere  $d : \Sigma^* \times \Sigma^* \rightarrow \mathfrak{R}^+ + \{0\}$  olacak şekilde bir mesafe fonksiyonu tanımlanabilir öyle ki  $d(L_1, L_2) := \beta(\{L_1 \setminus L_2\} \cup \{L_2 \setminus L_1\})$ 'dir.

Ayrık olay sistemlerinde iki dil arasındaki mesafe birbirlerine göre farklı olan kelimelerle ilişkilidir. Fakat bu farklı olan kelimelerin sadece adedi ya da benzer bir özelliği üzerinden işlem yapmak yukarıda anlatılanlar dikkate alındığında yetersizdir. Bu farklı kelimelerin toplam ederleri aslında iki farklı dilin arasındaki gerçek mesafeye karşılık düşmektedir, öyle ki bu mesafe fonksiyonu ile bir metrik uzay oluşturmak ve farklı diller arasındaki mesafeyi ifade etmek mümkündür.

### Önerme 1

$2^{\Sigma^*}$  kümesi ve mesafe fonksiyonu  $(2^{\Sigma^*}, d)$  şeklinde bir metrik uzay oluşturur. İspat için istenirse (Kaymakci ve Kurtulan 2009)'e bakılabilir.  $2^{\Sigma^*}$  kümesi ve  $d$  mesafe fonksiyonu önerme 1 gereği metrik uzay aksiyomlarını sağladığından,  $(2^{\Sigma^*}, d)$  bir metrik uzay oluşturur.

### Başarısızlık ve kilitlenme ölçüleri

$S$  denetimsel gözetleyicisinin kontrolü altında bir  $G$  otomatının ürettiği dil  $L(S/G)$  olsun ve aynı zamanda  $L(S/G)$ ,  $L_{cand}$  kümesinin bir elemanı olsun. O zaman  $L(S/G)$  diline ait başa-

rısızlık ölçüsü ve kilitlenme ölçüsü sırasıyla şu şekilde tanımlanır:

$$\hat{SM}^c(L(S/G)) := d(L_{am}, L_m(S/G)) \quad (2)$$

$$\hat{BM}(L(S/G)) := d(L(S/G), \overline{L_m(S/G)}) \quad (3)$$

Kilitlenme ölçüsü G otomatının S denetimsel gözetleyicisi ile beraber çalışması halinde, G'nin sahip olduğu olası tüm kilitlenmelerin toplam ederini vermektedir. Benzer bir şekilde başarısızlık ölçüsü ise S denetimsel gözetleyicisinin kontrolü altında G otomatının üretmediği kabul edilebilir işaretli kelimelerin toplam ederini verir.

### Performans ölçüsü

S denetimsel gözetleyicisinin kontrolü altında bir G otomatının ürettiği dil  $L(S/G)$  olsun. Ayrıca  $L(S/G)$ ,  $L_{cand}$  kümesinin bir elemanı olsun. O zaman  $L(S/G)$  diline ait performans ölçütü şu şekilde tanımlanır:

$$\hat{J}(L(S/G)) := \hat{SM}^c(L(S/G)) + \hat{BM}(L(S/G)) \quad (4)$$

Kilitlenebilir denetimsel gözetleyici tasarım probleminde genellikle kilitlenmelerle başarısızlık arasında bir zıtlık vardır. Öyle ki kilitlenmeleri azaltmak sistemin başarısızlığında artışa neden olurken, başarısızlığını azaltmak sistemin girebileceği olası kilitlenmelerde artışa neden olur. Örnek olarak en az kısıtlanmalı kilitlenmesiz çözüm ile tam başarılı çözüm incelenirse kilitlenme ile başarısızlık arasındaki bu zıtlık hemen kendini gösterir. Kilitlenebilir denetimsel gözetleyici tasarım probleminde sistemin performansı iki ölçüye bağlıdır: başarısızlık ve kilitlenme. Bu açıdan bakıldığında verilen performans ölçüsü tanımı bu iki kavramı da içermektedir. Açık ki kilitlenebilir denetimsel gözetleyici problemi için bu performans ölçütüne göre sayısal olarak en düşük performans ederine sahip olan dil en iyi dildir. O zaman optimal kilitlenebilir denetimsel gözetleyici tasarım problemi şu şekilde tanımlanabilir.

### Optimal kilitlenebilir denetimsel gözetleyici problemi

Sisteme ait tüm davranışları içeren dil  $L(G)$  ve sisteme ait tüm olası çözümleri içeren küme  $L_{cand} \neq \{\}$  olmak üzere S denetimsel gözetleyicisinin kontrolü altında G otomatının ürettiği dil ise  $L(S/G)$  olsun.  $L(S/G) \in L_{cand}$  için performans ölçüsü  $\hat{J}(L(S/G))$  olsun. Bu koşullar altında optimal kilitlenebilir denetimsel gözetleyici problemi şu şekilde  $\arg\{\min_{L \in L_{cand}} \hat{J}(L(S/G))\}$  tanımlanır.

Performans ölçüsü kitleme ve başarısızlık üzerine tanımlandığından, idealde en iyi çözüm bütün istenilen işaretli kelimeleri herhangi bir kilitlenmeye sebebiyet vermeden üreten çözümdür. Lakin kontrol edilebilirlik ve kabul edilebilirlik koşulu gereği bu çözüm her zaman olası değildir. O zaman olası çözümleri içeren  $L_{cand}$  kümesinden bir kilitlenebilir denetimsel gözetleyici seçmek gerekir. Bu noktada seçilen denetimsel gözetleyicinin performansının en düşük olması beklenir. Fakat uygulamada bu yeterli değildir. Zira denetimsel gözetleyiciden amaç elimizdeki sistemden maksimum beklenen davranışı elde etmektir. O zaman denetimsel gözetleyicinin ürettiği kabul edilebilir işaretli kelimelerin de olabildiğince fazla olması istenir. Sonuç olarak dil en düşük performans ölçüsüne sahip olmasının yanı sıra istenilen işaretli dili de olabildiğince işaretleyebilir olmalıdır (Kaymakci ve Kurtulan, 2009).

### Kayıp faktörü

S denetimsel gözetleyicisinin kontrolü altında bir G otomatının ürettiği dil  $L(S/G) \in L_{cand}$  olmak üzere  $L(S/G)$  diline ait kayıp faktörü şu şekilde tanımlanır:

$$F(L) := \beta(L \setminus \overline{L_m}) - \beta(L_m) \quad (5)$$

### Özellik 1

Eğer  $L = \overline{L}$  ve  $s_i \in BM(L)$  ise  $(L \setminus s_i)^{\uparrow C} = (L \setminus s_i)^{\uparrow C}$ .

### Özellik 2

Eğer  $L \in L_{cand}$ ,  $L = \bar{L}$  ve  $s_i \in BM(L)$  ise  $s_i \notin [(L \setminus s_i)^{\uparrow C} \cap L_{am}]^{\downarrow C}$ .

### Özellik 3

$s_i \in BM(L)$  ve  $L \in L_{cand}$  için  $[(L \setminus s_i)^{\uparrow C} \cap L_{am}]^{\downarrow C}$  her zaman  $L_{cand}$  kümesinin bir elemanıdır.

Özellik 1, örnek kapalı herhangi bir dilden kendisine ait bir kilitlenme çıkartılması halinde elde edilen sonuç dilin en büyük kontrol edilebilir kısmının her zaman örnek kapalı olduğunu ifade ederken, özellik 2 kontrol edilebilirlik ve kabul edilebilirlik koşulu gereği atılan kelimenin sonuç dile geri gelmediğini ifade etmektedir. Özellik 3 ise yapılan işlem neticesinde elde edilen dilin her zaman  $L_{cand}$  kümesinin bir elemanı olduğu ifade etmektedir.

### T işlemi

$L \in L_{cand}$  ve  $s_i \in BM(L(S/G))$  olmak üzere,  $T: L_{cand} \rightarrow L_{cand}$  olacak şekilde bir işlem olarak tanımlanır öyle ki

$$T(L, \alpha_i) := \begin{cases} ((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C} & \text{if } \Delta \\ L & \text{other} \end{cases} \quad (6)$$

Burada  $\Delta := \hat{J}(((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}) < \hat{J}(L)$  koşuluna karşılık düşmektedir. İşlemin sonucunu elde edebilmek için ilk olarak  $((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}$  diline elde edilir ve bu dile ait performans ölçüsü hesaplanır. Eğer  $((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}$  ait performans ölçüsü  $L$  diline ait performans ölçüsünden küçük ise işlem sonucunda  $((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}$  elde edilir. Eğer büyük ise  $L$  dili işlemin sonucu olarak kabul edilir. Burada işlem kesin bir performans değişimine bağlanmıştır.  $T$  işlemi özellik 3 gereği her zaman  $L_{cand}$  kümesine göre kapalılık gösterir. Bu da her zaman  $L_{cand}$  kümesinden  $L_{cand}$  kümesine olmasını garanti altına alır.

Eğer performans ile ilgili koşul sağlanır ise  $L$  dilinden çıkartılan kelimeler

$L \setminus ((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}$  şeklindedir. Gösterim kolaylığı sağlamak adına bundan sonra  $L \setminus ((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}$  yerine  $r(L, s_i)$  kullanılacaktır.

Ayrık olay sistemlerinde kilitlenme incelendiğinde fark edilecektir ki her bir kilitlenmenin sistem üzerine olan etkisi farklı farklıdır. Aynı zamanda kilitlenmelerin bir kısmı yukarıda açıklanan sebeplerden ötürü kabul edilebilir iken bir kısmı kesinlikle engellenmelidir. Bu engellenmesi gereken kritik kilitlenmelere ait kelimeler çıkartılması sistem performansını olumlu yönde etkiliyor ise denetimsel gözetleyiciye ait dilden uygun bir şekilde çıkartılmalıdır. Bunu yaparken sonuç dilin kontrol edilebilirlik ve kabul edilebilirlik koşullarını sağladığından emin olunmalıdır. Herhangi bir kritik kilitlenme denetimsel gözetleyicinin ürettiği dilden çıkartıldığına kontrol edilebilirlik ve kabul edilebilirlik koşulunun gereği olarak çoğu zaman beraberinde fazladan kelimeler de dilden çıkartılır. Aynı zamanda bu çıkarılan kelimeler içerisinde üretilen dilin performansını etkileyen kelimeler de olabilir.

$L \in L_{cand}$  diline ait farklı kilitlenmelerin  $T$  işlemi ile dilden çıkartılması halinde, kilitlenmeler ile beraber çıkartılan fazladan kelimelerin ya hiçbir ortak noktası yoktur ya da bir kilitlenmeye ait çıkartılan kelime grubu diğer bir kilitlenmeye ait çıkartılan kelime grubunu kapsar. Bu  $\uparrow C$  işleminin doğal bir sonucudur. Bu sonucu  $s_i, s_j \in BM(L)$  olmak üzere şu iki ifade ile göstermek mümkündür:  $r(L, s_i) \cap r(L, s_j) = \{\}$  veya  $r(L, s_i) \subseteq r(L, s_j)$ .

**Önerme 2:**  $L_1, L_2 \in L_{cand}$  iki dil olsun öyle ki  $L_1 \subset L_2$  ve  $(L_1 \cap \overline{L_{2m}}) \setminus \overline{L_{1m}} = \{\}$  koşulunu sağlasın. O zaman  $\hat{J}(L_1) = \hat{J}(L_2) - F(L_2 \setminus L_1)$  her zaman sağlanır. Bu önermenin ispatı (Kaymakci ve Kurtulan 2009) çalışmada yayınlanmıştır.

Bu önerme iki farklı dilin verilen koşulları sağlaması halinde performans ölçüleri arasındaki ilişkiyi ifade etmektedir. Burada  $\hat{J}(L_1)$  ve  $\hat{J}(L_2)$ ,

$L_1$  ve  $L_2$  dillerine ait performans ölçülerini ifade ederken  $F(L_2 \setminus L_1)$  ise  $L_2 \setminus L_1$  diline ait kayıp faktörünü ifade eder. Önerme 2 aynı zamanda  $(L_1 \cap \overline{L_{2m}}) \setminus \overline{L_{1m}} = \{\}$  koşulu  $BM(L_1) \subset BM(L_2)$  ilişkisinin her zaman sağlanmasını garanti eder.

$L \in L_{cand}$  uygun bir dil ve  $s_i \in BM(L)$  bu dile ait bir kilitlenme olsun. Her zaman  $((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C} \subset L$  olduğundan önerme 2 bu iki dile uygulanabilir. Eğer  $F(r(L, s_i)) \leq 0$  ise önerme 2'ye göre  $\hat{J}(((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}) \geq \hat{J}(L)$  olur. Diğer bir taraftan eğer  $F(r(L, s_i)) > 0$  ise  $\hat{J}(((L \setminus s_i)^{\uparrow C} \cap L_{am})^{\downarrow C}) < \hat{J}(L)$ 'dir.

### Önerme 3

Eğer  $L \in L_{cand}$ ,  $(L \cap L_{am})^{\downarrow C} = L$  ve  $s_i, s_k \in BM(L)$  ise

$$\{[(L \setminus s_i)^{\uparrow C} \cap L_{am}]^{\downarrow C} \setminus s_k]^{\uparrow C} \cap L_{am}\}^{\downarrow C} = \{[(L \setminus s_k)^{\uparrow C} \cap L_{am}]^{\downarrow C} \setminus s_i]^{\uparrow C} \cap L_{am}\}^{\downarrow C} \quad (7)$$

Bu önermenin ispatı için (Kaymakci ve Kurtulan, 2009) çalışmasına bakılabilir. Önerme 3  $L \in L_{cand}$  olacak şekilde bir dile kendisine ait herhangi iki kilitlenme üzerinden gösterildiği gibi bir işlem yapılması halinde, bu iki kilitlenmenin uygulama sırasının ulaşılan sonuç dilde bir değişikliği neden olmayacağını garanti eder. Ayrıca 3 no'lu önerme dile ait tüm kilitlenmelere genişletilebilir.

3 no.lu önermeye göre  $L \in L_{cand}$ 'ye  $BM(L)$ 'nin elemanları herhangi bir sıra ile uygulanması halinde ulaşılan son dil her zaman aynı kalır şeklinde yorumlanmamalıdır. Zira T işlemindeki koşul sağlanması halinde dile uygulanan işlem ile 3 no'lu önermede dile uygulanan işlem aynıdır. Fakat T işlemi bir ön koşula bağlı olarak işlem yaptığından  $BM(L)$ 'nin elemanlarının uygulama sırasını değiştirmek işlemin sonucunun her zaman aynı dile gideceğini göstermek için yeterli değildir. Öyle ki  $L \in L_{cand}$  olacak şekilde herhangi bir dil için dile ait kilitlenmeler iki farklı sırada ardı ardına T işlemi üzerinden

dile uygulanması halinde sonuç aynı olmayabilir. Diğer bir deyişle  $T[T(T(L, s_1), s_2), \dots, s_j]$ ,  $T[T(T(L, s_j), s_{j-1}), \dots, s_1]$  eşit olmayabilir.

### Teorem

$L \in L_{cand}$  uygun bir dil ve  $s_i, s_j \in BM(L)$  dile ait iki farklı kilitlenme olsun öyle ki  $F(r(L, s_i)) > F(r(L, s_j))$  ve  $(L \cap L_{am})^{\downarrow C} = L$  koşulları sağlanıyor ise

$\hat{J}(T(T(L, s_i), s_j)) \leq \hat{J}(T(T(L, s_j), s_i))$  her zaman sağlanır. İspat için (Kaymakci ve Kurtulan, 2009) çalışmasına bakılabilir.

Teorem,  $F(r(L, s_i)) > F(r(L, s_j))$  koşulunun sağlanması halinde  $T(T(L, s_i), s_j)$  diline ait performansın her zaman  $T(T(L, s_j), s_i)$  diline ait performanstan küçük eşit olduğunu ifade etmektedir. Burada  $s_i, s_j \in BM(L)$  kilitlenmeleri rasgele seçilmiş  $L \in L_{cand}$  diline ait kilitlenmelerdir. Bu yüzden bu iki kilitlenme için elde edilmiş sonuç çok rahatlıkla diğer kilitlenmelere de genişletilebilir.

### Optimizasyon algoritması

Kilitlenebilir denetimsel gözetleyici probleminde çözüm olarak seçilecek denetimsel gözetleyiciyle ilgili iki özellik ön plana çıkar: denetimsel gözetleyicinin istenilen işaretli dilin ne kadarını işaretleyebildiği ve ürettiği dilin performansı. İstenilen işaretli dile en fazla izin veren ve en düşük performans ederine sahip kilitlenebilir denetimsel gözetleyici bu problem için en uygun çözümdür. Bu iki koşula göre çözüm üreten yeni bir algoritma verilmiştir. Algoritmada  $BM(L_{temp1}) \setminus \{i\}$ ,  $BM(L_{temp1})$  kümesinin  $i$  elemanını ifade etmektedir.

#### 1. Adım

$L_{IS} = L_{am}^{\downarrow C}$  hesapla.  $L_{temp1} = L_{IS}$  olarak kabul et.

#### 2. Adım

$BM(L_{temp1}) = \{s_1, s_2, \dots, s_n\}$ 'yi bul.

$\forall s_i \in BM(L_{temp1})$  için  $F(r(L_{temp1}, s_i))$ 'yi hesapla.

$BM(L_{temp1})$  kümesini kayıp faktörlerine göre azalacak şekilde organize et.



### 3. Adım

$i=1$ 'den  $\|BM(L_{temp1})\|$ 'e kadar

$$A = BM(L_{temp1})\{i\}$$

$$L_x = T(L_{temp1}, A)$$

$$L_{temp1+1} = L_x$$

Son

$$L_{rst} = L_{temp1+1}$$

### 4. Adım

Eğer  $L_{rst} \neq L_{temp1}$

$$L_{temp1} = L_{rst}$$

Değilse

$$L_{FS} = L_{rst}$$

Yukarıda önerilen algoritmanın yapısı karmaşık değildir. 1. adımda tam başarılı çözüm hesaplanmakta ve bu başlangıç dili olarak kabul edilmektedir. 2. adımda tam başarılı çözüme ait kilitlenmeler bulunmaktadır. Modele ait dil düzenli bir dil olduğundan tam başarılı çözüme ait kilitlenmeler de sonlu sayıda olacaktır. Aynı zamanda ikinci adımda tam başarılı çözüme ait her bir kilitlenme için kayıp faktörleri hesaplanmaktadır. Bu hesaplanan kayıp faktörleri üzerinden büyükten küçüğe doğru  $BM(L_{temp1})$  kümesi yeniden sıralanır. 3. adımda ise bu sıralanmış kilitlenmelere göre T işlemi sırayla uygulanır. Son olarak 4. adımda ise 3. adımın sonunda elde edilen  $L_{rst}$  ile 3. adımın başında girilen  $L_{temp1}$  karşılaştırılır. Eğer bu iki dil eşit ise  $L_{rst}$  sonuç dil olarak kabul edilir ve algoritma sonlandırılır. Eğer bu iki dil eşit değil ise  $L_{temp1}$  olarak  $L_{rst}$  alınır ve 2. adımdan başlayarak yeniden aynı işlemler yinelenir.

### Belirtme

Algoritmanın temelini T işlemi ve onunla ilgili elde edilen sonuçlar oluşturmaktadır. T işlemi,  $L \in L_{cand}$  olacak şekilde bir dil ve dile ait herhangi bir kilitlenme için tanımlanmıştır. Kilitlenmeyi dilden tanımlandığı gibi çıkarttıktan sonra elde edilen dilin performansı, L diline ait olan performans ile karşılaştırılır; daha iyi ise elde edilen dil sonuç dil olarak kabul edilir, iyi değil ise L dili değiştirilmez. Algoritmanın 3. adımında T işlemi, başlangıç diline kendisine ait

tüm kilitlenmeler üzerinden uygulanır. T işlemi performans değişimine bağlı olarak tanımlandığından dile ait kilitlenmeler farklı sıralamalar ile uygulanması halinde ulaşılan sonuç dil farklılık gösterebilir. Ulaşılan dilin farklı olması dillerin performanslarının da farklı olmasına neden olur. Amaçlardan birisi en düşük performans ederine sahip olan dili elde etmek olduğundan, kilitlenmelerin dile uygulama sırası önemlidir. Teorem gereği eğer dile ait iki kilitlenmeden kayıp faktörü büyük olan önce T işlemine tabii tutulursa elde edilen sonuç dil diğer şekilde elde edilecek dilden daha düşük bir performans ederine sahip olur. Buna göre algoritmadaki 2. adımda dile ait kilitlenmeler kayıp faktörlerine göre büyükten küçüğe doğru sıralanır. Bu da algoritmanın 3. adımında tüm kilitlenmeler için yapılan T işleminin sonucuna ait performans ederinin diğer sıralamalar sonucunda elde edilen dillerin performans ederlerinden küçük olmasını garanti eder.

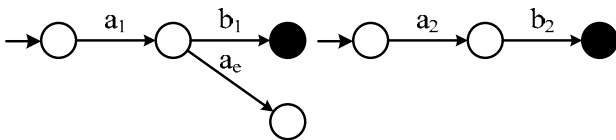
Optimal kilitlenebilir denetimsel gözetleyici probleminde amaç, sadece en düşük performans ederine sahip olan denetimsel gözetleyiciyi elde etmek değil aynı zamanda istenilen işaretli kelimenin de olabildiğince çok üretilmesini sağlamaktır. Bu noktada başlangıç dili olarak  $L_{cand}$  kümesine ait herhangi bir dili almaktansa, tam başarılı çözümü başlangıç dili olarak almak bizi en fazla işaretli kelimeyi üreten en düşük performans ederine sahip olan dile götürecektir. Zira tam başarılı çözüm tüm istenilen kelimeleri içermektedir ve tüm istenilen kelimeler algoritma içerisinde değerlendirilmiş olunur. Böylelikle sonuç dilin en fazla istenilen işaretli kelimeyi üretmesi garanti altına alınır. 2, 3 ve 4 no'lu adımlar gereğince belli bir düzen içerisinde bazı kilitlenmeler düşer ve en fazla işaretli kelimeyi üreten en düşük performans ederine sahip olan dile ulaşılır.

### Uygulama

Algoritma bir veri yönetim sistemi üzerinde uygulanacaktır. Veri yönetim sisteminin seçilmesinin en büyük sebebi problemin yapısı gereği kilitlenmeye sahip olması ve uygulamada kilitlenebilir denetimsel gözetleyicilerin veri yöne-

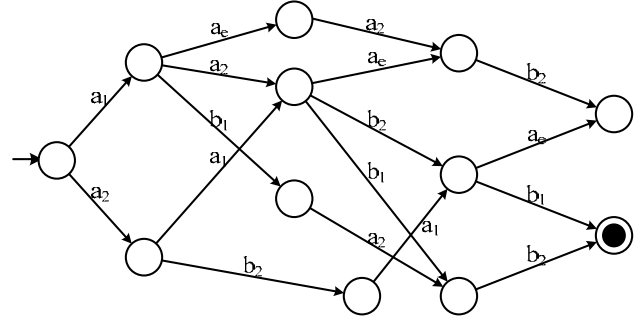
tim sistemlerinde denetimsel gözetleyici olarak tercih edilmesidir. Veri yönetim sistemleri kilitlenme bulma yapılarını da içermektedir. Böylelikle performansı arttırmak için belli bir orada kilitlenmeye izin verilmektedir. Bir veri yönetim sistemi, birden fazla kullanıcının rastlantısal olarak veri tabanı üzerinde düzenleme ve güncelleme yaparken, veri tabanının tutarlı kalmasını garanti altına alan sistemlerdir. Veri yönetim sistemlerinde ardışıl olarak her bir kullanıcının okuma-yazma yapması geçiş olarak adlandırılır. Rastlantısal olarak birden fazla geçişin oluşması halinde veritabanının tutarlılığını koruyabilmek için veritabanı yönetim sistemi geçiş işlemlerinin birbirlerine göre ardı sıra olan geçiş sıralarını tasarlar. Bu açıdan bakıldığında doğru çözüme ulaşabilmek için birden fazla geçişin birbiri içindeki hareketi tamamen engellemeye gerek yoktur. Bu da veri yönetim sisteminin performansının artmasına olanak sağlar. Bunun yanı sıra birçok eşzamanlı kontrol protokolleri kilitlenme bulma yapıları ile beraber çalıştılarından tamamen kilitlenmeyi engellemek pratik bir çözüm değildir.

$T_1 = a_1b_1$  ve  $T_2 = a_2b_2$  gibi iki geçişimizin olduğunu düşünelim öyle ki bu iki geçişin paralel çalışması ile oluşacak olan veri sistemine ilişkin olay kümesi ve kontrol edilemeyen olaylar kümesi sırasıyla  $\Sigma = \{a_1, b_1, a_2, b_2, a_e\}$  ve  $\Sigma_{uc} = \{a_e\}$  olsun.  $T_1$  ve  $T_2$  geçişlerine ilişkin olarak otomat modelleri Şekil 1'deki gibi verilebilir.  $T_1$  geçişinde  $a_1$  olayının olmasının ardından bir hatanın oluşabildiğini düşünelim öyle ki bu hatayı  $a_e$  ile gösterelim. Burada gösterimde kolaylık olması açısından yapılan işlemin okuma mı yazma mı olduğu önemsiz sayılmıştır.



Şekil 1.  $T_1$  ve  $T_2$  geçişlerine ait otomatlar

Bu iki otomat paralel çalıştığında elde edilen veri tabanı sistemine ait olan otomat modeli ise Şekil 2'deki gibidir.



Şekil 2. Veritabanına ilişkin otomat modeli

Eşzamanlı veritabanı kontrol teorisi gereğince kabul edilebilir işaretli dilde  $a_1$ 'den sonra  $a_2$  ancak  $b_1$ 'den sonra  $b_2$ 'nin üretilmesi halinde kabul edilebilir. O zaman Şekil 2'de verilen veri sistem için kabul edilebilir işaretli dil kümesi  $L_{am} = \{a_1b_1a_2b_2, a_2b_2a_1b_1, a_1a_2b_1b_2, a_2a_1b_1b_2\}$ 'dir. Sistemin olaylara ilişkin olay ederleri Tablo 1'de verilmiştir. Bu verilen olay ederleri üzerinden en fazla işaretli kelimeyi üreten en düşük performans ederine sahip olan dil elde edilecektir.

Tablo 1. Veritabanı sisteminin ürettiği olay ederleri

Olay	Eder	Olay	Eder
$c_e(\varepsilon, a_1)$	2	$c_e(a_2, a_1)$	10
$c_e(a_1, a_2)$	10	$c_e(a_2, b_2)$	12
$c_e(a_1, a_e)$	2	$c_e(a_2a_1, a_e)$	20
$c_e(a_1, b_1)$	5	$c_e(a_2a_1, b_2)$	7
$c_e(a_1a_2, b_1)$	4	$c_e(a_2b_2, a_1)$	15
$c_e(a_1b_1, a_2)$	10	$c_e(a_2a_1b_2, b_1)$	7
$c_e(a_1a_2, a_e)$	15	$c_e(a_2a_1b_2, a_e)$	5
$c_e(a_1a_2b_1, b_2)$	4	$c_e(a_2b_2a_1, b_1)$	25
$c_e(a_1b_1a_2, b_2)$	12	$c_e(a_2b_2a_1, a_e)$	2
$c_e(\varepsilon, a_2)$	10		

Bu verilen olay ederlerine göre veritabanı sisteminin ürettiği kelimelerin ederi verilen tanım yardımıyla elde edilir.

$$c_s(a_1a_2b_1b_2) = 5, c_s(a_1b_1a_2b_2) = 7.25, \\ c_s(a_1a_e) = 2, c_s(a_2a_1b_2a_e) = 8, c_s(a_1a_2a_e) = 9, \\ c_s(a_2a_1b_2b_1) = 8.5, c_s(a_2a_1a_e) = 13.33, \\ c_s(a_2b_2a_1b_1) = 15.5, c_s(a_2b_2a_1a_e) = 9.75.$$

### 1. Adım

$$L_{IS} = L_{am}^{\downarrow C}, L_{temp1} = L_{IS}.$$

### 2. Adım

$$BM(L_{temp1}) = \{a_1a_e, a_1a_2a_e, a_2a_1a_e, a_2a_1b_2a_e, a_2b_2a_1a_e\}$$

$$F(r(L_{temp1}, a_1a_e)) = -1.25$$

$$F(r(L_{temp1}, a_1a_2a_e)) = 4$$

$$F(r(L_{temp1}, a_2a_1a_e)) = 12.58$$

$$F(r(L_{temp1}, a_2b_2a_1a_e)) = -5.75$$

Elde edilen kayıp faktörleri değerlerine göre dile ait kilitlenmeleri içeren küme yeniden düzenlenirse

$$BM(L_{temp1}) = \{a_2a_1b_2a_e, a_2a_1a_e, a_1a_2a_e, a_1a_e, a_2b_2a_1a_e\} \text{ gibi olur.}$$

### 3. Adım

$$i = 1 \text{ için } A = BM(L_{temp1})\{1\} = a_2a_1b_2a_e$$

$$\hat{J}[(L_{temp1} \setminus A)^{\uparrow C} \cap L_{am}]^{\downarrow C} < \hat{J}[L_{temp1}] \text{ olduğundan}$$

$$L_{temp2} = T(L_{temp1}, A) = \{a_1b_1a_2b_2, a_1a_2b_1b_2, a_2b_2a_1b_1, a_1a_2a_e, a_1a_e, a_2b_2a_1a_e\}$$

$$i = 2 \text{ için } A = BM(L_{temp1})\{2\} = a_2a_1a_e \text{ 'dir.}$$

$$A \notin L_{temp2} \Rightarrow ((L_{temp2} \setminus A)^{\uparrow C} \cap L_{am})^{\downarrow C} = L_{temp2}$$

sonuçta  $L_{temp3} = L_{temp2}$  'dir.

$$i = 3 \text{ için } A = BM(L_{temp1})\{2\} = a_1a_2a_e$$

$$\hat{J}[(L_{temp3} \setminus A)^{\uparrow C} \cap L_{am}]^{\downarrow C} < \hat{J}[L_{temp3}] \text{ için}$$

$$L_{temp4} = T(L_{temp3}, A) = \{a_1b_1a_2b_2, a_2b_2a_1b_1, a_1a_e, a_2b_2a_1a_e\}$$

$$i = 4 \text{ için } A = BM(L_{temp1})\{4\} = a_2a_e \text{ 'dir.}$$

$$\hat{J}[(L_{temp4} \setminus A)^{\uparrow C} \cap L_{am}]^{\downarrow C} > \hat{J}[L_{temp4}] \text{ için}$$

$$L_{temp5} = T(L_{temp4}, A) = \{a_1b_1a_2b_2, a_2b_2a_1b_1, a_1a_e, a_2b_2a_1a_e\}$$

$$i = 5 \text{ için } A = BM(L_{temp1})\{5\} = a_2b_2a_1a_e$$

$$\hat{J}[(L_{temp5} \setminus A)^{\uparrow C} \cap L_{am}]^{\downarrow C} > \hat{J}[L_{temp5}]$$

için  $L_{temp6} = T(L_{temp5}, A) = L_{temp5}$

$$L_{rst} = L_{temp6} = \{a_1b_1a_2b_2, a_2b_2a_1b_1, a_1a_e, a_2b_2a_1a_e\}$$

### 4. Adım

$L_{rst} \neq L_{temp1}$  olduğundan  $L_{rst}, L_{temp1}$  olarak alınır ve ikinci ve üçüncü adımlar yeniden uygulanır. Sonuçta algoritmanın ikinci ve üçüncü adımları yeniden uygulandıktan sonra veritabanı sistemi için verilen olay ederi üzerinden en fazla işaretli kelimeyi üreten en düşük performans edere sahip olan dil  $L_{FS} = \{a_1b_1a_2b_2, a_2b_2a_1b_1, a_1a_e, a_2b_2a_1a_e\}$  olarak bulunmuştur. Bu dile ilişkin kilitlenme ölçüsü ve başarısızlık ölçüsü ise şu şekildedir:

$$\hat{SM}^C(L_{FS}) = \beta(a_1a_2b_1b_2, a_2a_1b_2b_1) = 13.5$$

$$\hat{BM}(L_{FS}) = \beta\{a_1a_e, a_2b_2a_1a_e\} = 11.75$$

Sonuçta dile ait performans ölçüsü

$$\hat{J}(L_{FS}) = \hat{SM}^C(L_{FS}) + \hat{BM}(L_{FS}) = 25.25 \text{ olacaktır.}$$

Başlangıç diline ait başarısızlık ve kilitlenme

ölçüleri ise  $\hat{SM}^C(L_{IS}) = 0$  ve  $\hat{BM}(L_{IS}) = 42.08$

şekindedir. O zaman başlangıç diline ait performans

ölçüsü  $\hat{J}(L_{IS}) = 42.08$  'dir.  $\hat{J}(L_{IS})$

$< \hat{J}(L_{FS})$  'dir. Sonuç dilin performansı başlangıç

diline göre daha iyidir. Aynı zamanda tam başarımlı

çözümü başlangıç dili olarak seçmekle algoritma,

bu performans ederi için en fazla işaretli kelimeyi üreten dili elde etmiştir.

## Sonuçlar

Kilitlenebilir denetimsel gözetleyici probleminde

kilitlenmeler ve başarısız kabul edilebilir işaretli

kelimeler arasında bir değiş tokuş vardır. Öyle ki

sistemin sahip olduğu kilitlenmeler azaldıkça

başarısız işaretli kelimelerin arttığı ve başarısız

kelimelerin azaldıkça kilitlenmelerin arttığı gözlemlenir.

Bu çalışmada bu probleme farklı bir bakış açısı altından

çözüm bulunmaya çalışılmıştır. Bu amaçla birbirinden

farklı başarısız işaretli kelimeler ve kilitlenmeler

içeren dilleri karşılaştırabilmek için yeni bir mesafe

fonksiyonu tanımlanmış ve  $2^{\Sigma^*}$  uzayı ile bu

mesafe fonksiyonunun bir metrik uzay oluşturduğu

gösterilmiştir. Aynı zamanda problemin doğasında

bulunan bu değiş tokuş en iyi şekilde yansıtan yeni bir perfor-

mans ölçüsü ortaya atılmış, herhangi bir denetimsel gözetleyicinin performansı pozitif reel bir sayıya karşılık düşürülmüştür. Bu performans ölçüsüne göre de optimal kilitlenebilir denetimsel gözetleyici problemi tanımlanmıştır. Bu problem için denetimsel gözetleyicinin sağlanması gereken birinci koşul en düşük performans ölçüsüne sahip olması, ikinci koşul ise istenilen işaretli kelimeleri olabildiğince çok kapsamasıdır. Sonuç olarak optimal kilitlenebilir denetimsel gözetleyici problemi için en iyi çözüm, istenilen işaretli kelimelere en fazla izin veren en düşük performans ölçüsüne sahip çözümdür.

Bu çalışmada istenilen işaretli dile de en fazla izin veren ve en düşük performansa sahip kilitlenebilir denetimsel gözetleyiciyi bulmak amaçlanmıştır. Bunun için yeni bir algoritma önerilmiş, algoritmanın belirtilen ölçütlerde çözüme ulaştığı ispatlanmış ve bir veri yönetim sistemi üzerinde uygulanmıştır.

## Kaynaklar

- Brave, Y. ve Heymann M. (1990). Stabilization of discrete-event processes, *International Journal of Control*, **51**, 5, 1101-1117.
- Brave, Y. ve Heymann M. (1993). On optimal attraction in discrete-event processes, *Information Sciences*, **67**, 3, 245-267.
- Caillaud, B., Darondeau, P., Lavagno, L. ve Xie, X. (2002). *Synthesis and Control of Discrete Event Systems*, Kluwer Academic Publishers, Dordrecht.
- Cassandras, C. ve Lafortune, S. (1999). *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, Massachusetts, USA.
- Chen, E. ve Lafortune, S. (1991). Dealing with blocking in supervisory control of discrete-event systems, *IEEE Transactions on Automatic Control*, **36**, 6, 724-735.
- Kaymakci, O. ve Kurtulan, S. (2009). A novel performance evaluation method for discrete event systems, *Journal Of Information Science And Engineering*, **25**, 105-120.
- Kaymakci, O. ve S. Kurtulan (2007). A performance evaluation algorithm for discrete event systems under blocking, *The 15th Mediterranean Conference On Control And Automation (MED'07)*, Athens, Greece.
- Kaymakci, O. ve Kurtulan, S. (2007). A performance evaluation algorithm for discrete event systems under blocking, *WSEAS Transactions on Systems*, Basımda.
- Kaymakci, O., Kurtulan, S. ve Gören, L. (2005). Improving the behaviour of supervisor under blocking, *16th IFAC World Congress*, Prague, Czech Republic.
- Kumar, R., Garg, V., (1993). Language stability and stabilizability of discrete event dynamical systems, *SIAM Journal on Control and Optimization*, **31**, 1294-1320.
- Passino, K. ve Antsaklis P. J. (1989). On optimal control of discrete event systems, *IEEE 28th Decision and Control Conference*, Tampa, Florida.
- Passino, K. M. ve Antsaklis P. J. (1990). Optimal stabilization of discrete event systems, *29th Conference on Decision and Control*, Honolulu, Hawaii.
- Ramadge, P. J. ve Wonham, W. M. (1987). Supervisory control of a class of discrete event systems, *SIAM Journal Control and Optimization*, **25**, 206-230.
- Ramadge, P. J. ve Wonham, W. M. (1989). The control of discrete event systems, *Proceedings of IEEE* **77**, 1, 81-88.
- Sengupta, R. ve Lafortune, S. (1998). An optimal control theory for discrete event systems, *SIAM Journal of Control and Optimization*, **26**, 2, 448-541.