

156312

PETRI AĞLARINDA
ÖRTÜŞMELİ AYRIŞTIRMA VE GENLEŞTİRME
KULLANILARAK KONTROLÖR TASARIMI

AYDIN AYBAR

Doktora Tezi

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği
Anabilim Dalı
Şubat-2001

JÜRİ VE ENSTİTÜ ONAYI

Aydın Aybar'ın ``Petri Ağlarda Örtüşmeli Ayrıştırma ve Genleştirme Kullanılarak Kontrolör Tasarımı`` başlıklı Elektrik-Elektronik Mühendisliği Anabilim Dalındaki, Doktora tezi 12.02.2001 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	İmza
Üye (Tez Danışmanı)	: Prof. Dr. Altuğ İFTAR	
Üye	: Prof. Dr. Kemal İNAN	
Üye	: Prof. Dr. Ersin TULUNAY	
Üye	: Doç. Dr. Abdurrahman KARAMANCIOĞLU	
Üye	: Doç. Dr. Hüseyin AKÇAY	

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun
14.02.2001.. tarih ve ...6/1..... sayılı kararıyla onaylanmıştır.

ÖZET

Doktora Tezi

PETRI AĞLARINDA ÖRTÜŞMELİ AYRIŞTIRMA VE GENLEŞTİRME KULLANILARAK KONTROLÖR TASARIMI

AYDIN AYBAR

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Altuğ İFTAR

2001

Kesikli olay sistemlerin modelleme yöntemlerinden, Petri ağı, vektörel, otomata ve dil gösterimleri incelenmiştir. Bugüne kadar sürekli durum sistemleri için kullanılmış olan ve kesikli durum sistemlerinin yalnızca vektörel gösterimi için uygulanmış olan örtüşmeli ayrıştırma ve genleştirme işlemleri, Petri ağları, otomata ve dil gösterimlerine ilk kez bu çalışmada uygulanmıştır. İçerme prensibi yine bu çalışmada Petri ağları, otomata ve dil gösterimleri için ilk kez tanımlanmıştır. Petri ağı ve otomata gösterimlerinin her biri için, örtüşmeli ayrıştırmaya dayalı dışmerkezli kontrolör tasarım yöntemleri geliştirilmiştir. Bu yöntemlerle tasarlanan kontrolörler, sistem çıkmazı meydana gelmemesini garanti etmektedir. Ayrıca bu çalışmada, sistem çıkmazı meydana gelmesine izin vermeyen bir dil tasarım yöntemi de önerilmiştir.

Anahtar Kelimeler: Kesikli Olay Sistemler, Petri Ağları, Örtüşmeli Ayrıştırma, İçerme Prensibi, Dışmerkezli Kontrol

ABSTRACT

Ph.D. Dissertation

**CONTROLLER DESIGN ON PETRI NETS USING OVERLAPPING
DECOMPOSITIONS AND EXPANSION**

AYDIN AYBAR

Anadolu University
Graduate School of Natural and Applied Sciences
Electrical and Electronics Engineering Program

Supervisor: Prof. Altuğ İFTAR

2001

Petri nets, vector-forms, automatas, and formal languages, which are modeling methods for discrete event systems, are studied. Overlapping decompositions and expansions which have been used for continuous state systems and for discrete state systems modelled using vector-forms are applied to Petri nets, automatas and formal languages in this study for the first time. Inclusion principle for Petri nets, automatas and formal languages is also defined in this study for the first time. Decentralized supervisory controller design methods based on overlapping decompositions are developed for Petri nets and automatas. The controllers which are designed using these methods guaranty deadlock avoidance. Furthermore a design method for a deadlock-free formal language is also purposed in this study.

Keywords: Discrete Event Systems, Petri Nets, Overlapping Decompositions, Inclusion Principle, Decentralized Control

TEŐEKKÜR

Çalıőmalarım sırasında bana yol gösteren ve yardımlarını esirgemeyen hocam Prof. Dr. Altuğ İftar'a ve beni destekleyen aileme sonsuz teőekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	viii
SİMGELER ve KISALTMALAR DİZİNİ	ix
1. GİRİŞ	1
2. PETRI AĞLARI ve KESİKLİ OLAY SİSTEMLER İÇİN DİĞER BAZI MODELLEME YÖNTEMLERİ	6
2.1. Petri Ağı	6
2.1.1. Petri ağının tanımlanması	6
2.1.2. Petri ağının özellikleri	10
2.1.3. Petri ağı ile sistem modellemesi	11
2.2. Kesikli Olay Sistemlerin Diğer Bazı Modelleme Yöntemleri ile Petri Ağı Modellemesi Arasındaki İlişki	19
2.2.1. Otomata ve dil gösterimleri ile Petri ağı gösterimi arasındaki ilişki	19
2.2.2. Vektörel gösterim ile Petri ağı gösterimi arasındaki ilişki	25
3. KONTROLÖR TASARIMI	28
3.1. Petri Ağında Kontrolör Tasarımı	28
3.2. Kesikli Olay Sistemler İçin Kullanılan Bazı Kontrol Yaklaşımları Arasındaki İlişkiler	32

3.2.1. Dil gösterimi için bir kontrol yaklaşımı	33
3.2.2. Otomata gösterimi için bir kontrol yaklaşımı	33
3.2.3. Vektörel gösterim için bir kontrol yaklaşımı	35
4. İÇERME PRENSİBİ	36
4.1. Petri Ağlarında İçerme Prensibi	36
4.2. Biri Diğerini İçeren İki Petri Ağı Arasındaki İlişkiler	38
4.3. Otomata ve Dil Gösterimlerinde İçerme Prensibi	43
5. ÖRTÜŞMELİ AYRIŞTIRMA ve GENLEŞTİRME	45
5.1. Petri Ağlarında Örtüşmeli Ayrıştırma ve Genleştirme	45
5.2. Genleştirilmiş Petri Ağı ile Özgün Petri Ağı Arasındaki Bazı İlişkiler	54
5.3. Otomata ve Dil Gösterimlerinde Örtüşmeli Ayrıştırma ve Genleştirme	65
5.4. Genleştirilmiş Otomata ve Dil ile Özgün Otomata ve Dil Arasındaki Bazı İlişkiler	70
5.5. Değişik Gösterimler İçin Genleştirme İşleminin Karşılaştırılması	75
6. DIŞMERKEZLİ KONTROLÖR TASARIMI	76
6.1. Petri Ağlarında Dışmerkezli Bir Kontrolör Tasarım Yöntemi ..	76
6.1.1. Alt Petri ağları için kontrolör tasarımı	76
6.1.2. Genleştirilmiş Petri ağı için kontrolör tasarımı	81
6.1.3. Özgün Petri ağı için kontrolör tasarımı	86
6.2. Otomata Gösteriminde Dışmerkezli Bir Kontrolör Tasarım Yöntemi	91
6.2.1. Alt otomatalarda kontrolör tasarımı	91
6.2.2. Genleştirilmiş otomatada kontrolör tasarımı	92
6.2.3. Özgün otomatada kontrolör tasarımı	92

6.2.4. Sistem çıkmazının meydana gelmesini önleyen dil tasarımı	95
6.3. Önerilen Yöntemlerin Hesaplama Karmaşıklığı	96
7. SONUÇLAR	101
KAYNAKLAR	105

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
2.1. Petri ağı	7
2.2. Örnek 2.2'de ele alınan üretim sistemi	12
2.3. Örnek 2.2'de verilen sistemin Petri ağ modeli	16
2.4. Örnek 2.3'de ele alınan üretim sistemi	17
2.5. Örnek 2.3'de verilen sistemin Petri ağ modeli	18
2.6. Örnek 2.3'de verilen sistemin otomata modeli	21
2.7. Şekil 2.1'de Petri ağı ile gösterilen sistemin otomata modeli ...	24
3.1. Kapalı döngü Petri ağı	29
5.1. Örtüşmeli ayrıştırma uygulanmış Petri ağ	47
5.2. Genleştirilmiş Petri ağı	49
5.3. Şekil 2.7'deki otomatanın örtüşmeli ayrıştırma uygulanmış hali	66
5.4. Şekil 5.3'deki otomatanın genleştirilmiş hali	66
5.5. Örnek bir otomata	69
5.6. Şekil 5.5'deki otomatanın örtüşmeli ayrıştırma uygulanmış hali	69
5.7. Şekil 5.6'daki otomatanın genleştirilmiş hali	70
6.1. Merdiven otomatalar	98

ÇİZELGELER DİZİNİ

Sayfa

2.1. Şekil 2.3'deki Petri ağının girdi matrisi	14
2.2. Şekil 2.3'deki Petri ağının çıktı matrisi	15
2.3. Örnek 2.3'de verilen sistemin $R(G, m_0)$ ve Q kümelerinin elemanları	22
2.4. Şekil 2.1'de Petri ağı ile gösterilen sistemin $R(G, m_0)$ ve Q	24
2.5. Örnek 2.3'de verilen sistemin \mathcal{X}_u kümesinin elemanları	26
2.6. Örnek 2.3'de verilen sistemin vektörel gösterimdeki çalışması ..	27
5.1. Şekil 5.1'de verilen Petri ağı ile Şekil 5.2'de verilen Petri ağı arasındaki içerme prensibinde kullanılan Q matrisi	63
5.2. Şekil 5.1'de verilen Petri ağı ile Şekil 5.2'de verilen Petri ağı arasındaki içerme prensibinde kullanılan V matrisi	64
6.1. V_2 matrisi	77
6.2. Şekil 6.1'de verilen otomataların hesaplama karmaşıklığı	99

SİMGELER ve KISALTMALAR DİZİNİ

\mathcal{N} : Doğal sayılar kümesi

$|\cdot|$: (\cdot) kümesinin eleman sayısı

$(\cdot)^T$: (\cdot) matrisinin transpozu

$\mu = \{0, 1\}$

$\lambda(M)$: M vektörünün en büyük elemanı

$köşegen\{\dots\}$: köşegeninde $\{\dots\}$ kümesinin elemanları olan köşegen matris

$\mathcal{A}^{m \times n}$: \mathcal{A} alanı üzerinde tanımlı $m \times n$ boyutlu matrisler kümesi

I : birim matris

Z : sıfır matrisi

$R(G, M)$: G ile gösterilen Petri ağında, M işaretleme vektöründen sonra ulaşılabilir işaretleme vektörlerinin oluşturduğu küme

$\mathcal{Y}(G)$: G ile gösterilen Petri ağında, kendisinden sonra mümkün olan tüm ateşleme dizileri kullanıldığında, sistem çıkmazı meydana gelen, işaretleme vektörlerinin oluşturduğu küme

APA: Alt Petri Ağı

GPA: Genleştirilmiş Petri Ağı

ÖPA: Özgün Petri Ağı

1. GİRİŞ

Günümüz sistemlerinin büyümesi, genişlemesi ve karmaşıklaşması nedeniyle, sistemlerin modellenmesi ve kontrolü büyük önem kazanmıştır. Geleneksel yöntemler yerine, günümüz sistemlerinin bir çoğunun modelleme, analiz, kontrol vb. amaçlarla incelenmesi için, olay etkileşimli sistemler (event-driven) kullanılmaya başlanmıştır. Olay etkileşimli sistemler; bir olayın ortaya çıkışının, başka olayların meydana gelişine bağlı olduğu sistemlerdir. Otomatik üretim sistemleri, haberleşme ağları, bilgisayar ağları, vb. sistemler olay etkileşimli sistemlerin kapsamına alınabilir. Söz konusu sistemler, *kesikli olay [dinamik] sistemler* (discrete event [dynamic] systems) olarak da adlandırılmaktadırlar.

Kesikli olay sistemler, çok fazla değişken ile ifade edildiği ve olayların meydana gelişi, diğer olay ya da olayların oluşuna bağlı olduğundan, bu tür sistemleri diferansiyel veya fark denklemleri ile göstermek zor, hatta imkansızdır. Kesikli olay sistemlerin modellenmesinde, Markov zincirlerinin, otomata modellerinin (bu gruba, Petri ağları ve sonlu durum makineleri (finite-state machines) dahildir), minimum-maksimum cebir modellerinin, kuyruk sıralama ağ modellerinin (queueing network models), sıralı işlem modellerinin (communicating sequential process models) ve genelleştirilmiş yarı Markov işlem modellerinin (generalized semi Markov process models) kullanıldığından bahsedilmiştir [1].

Bu çalışmada, kesikli olay sistemlerin modellenmesinde, analizinde ve kontrolör tasarımında Petri ağları kullanılmıştır. Ayrıca Petri ağı modelinin, kesikli olay sistemlerin diğer bazı modellemeleriyle olan ilişkisi de ortaya konmuştur.

Petri ağı yapısı, ilk kez 1962'de Carl Adam Petri tarafından ortaya konmuştur. Petri ağı, grafiksel ve matematiksel bir modelleme aracıdır. Bir Petri ağının genel yapısını; yerler, geçişler ve bunların arasındaki bağlantılar oluşturur [2].

Petri ağı yönteminin gelişiminin başlangıcında, sistemin durum özelliklerinin analizi büyük bir öneme sahip olmuştur. Ağdaki iletimin analizi için,

işaretlenmiş kapsayabilirlik (coverability) grafikleri veya ulaşılabilirlik ağaçları (reachability trees) şeklindeki özgün araçlar ve bu işaretlemeler arasındaki ilişki, Karp ve Miller [3] tarafından gösterilmiştir. Lautanbach [4], Petri ağlarında *canlılığı* (liveness) incelemiştir ve ilk olarak geliştirdiği cebirsel yaklaşımları, *sınırlılığın* (boundedness) ve canlılığın analizinde kullanmıştır. Commoner [5] ise, Petri ağları kullanarak *sistem çıkmazını* (deadlock) incelemiştir. Diğer yapısal özelliklerin (yapısal özellikler, yalnızca Petri ağının yapısına bağlı özelliklerdir; yapısal sınırlılık vb.), analiz yöntemleri ise Sifakis [6] tarafından geliştirilmiştir. Günümüzde de Petri ağı ile ilgili çalışmalar artarak devam etmektedir.

Petri ağı, eş zamanlı olmayan, istatikselsel ve aynı zamanda meydana gelen olayların birbirlerine yapısal etkilerini ve ilişkilerini ortaya koymuş, karmaşık sistemlerin sistematik analizinin yapılmasına ve gerçek zamanlı kontrolör gerçekleştirilmesine imkan tanımıştır. Petri ağı modellemesinden geliştirilmiş, zamanlanmış (timed) Petri ağı, reklendirilmiş (coloured) Petri ağı, kontrol edilen (controlled) Petri ağı, vb. Petri ağı tipleri vardır [7–8]. Bu çalışmada kullanılan Petri ağı yapısı, Bölüm 2’de verilmiştir.

Kesikli olay sistemler için kontrol teorisi, Ramadge ve Wonham [9] tarafından ortaya konmuştur. İlgili çalışmada, bir kesikli olay sistem, matematiksel olarak *biçimsel dil* (formal language) ile ifade edilmiş ve sistemin özellikleri yine bu dil ile tanımlanmıştır. Ayrıca [9]’da, merkezi kontrolör için, yerel bilgi ve yerel kontrolörlerin kullanıldığı, kısmi gözlenebilen olaylar esas alınarak oluşturulan dışmerkezli kontrolden de bahsedilmiştir. Özellikle Lin ve Wonham [10], kapalı döngü sistemde dışmerkezli kontrolörü iyileştirmek amacıyla yerel kontrolörler arasında eş güdüm sağlayan bir kontrolör ortaya koymuşlardır. Ancak [10]’da kesikli olay sistemler, [9]’daki gibi biçimsel dil ile gösterilmek yerine *otomata* (automata) olarak gösterilmiştir. [11]’de ise, yalnız başına dışmerkezli kontrolörün yeterli olmadığı durumda, dışmerkezli kontrolör ile merkezi kontrolörün beraber kullanımı gösterilmiştir.

Holloway ve Krogh [12], kontrol edilen Petri ağı için, yasaklanmış durum problemlerini incelemiştir. Kontrol edilen Petri ağı, fazladan kontrol yer-

lerinin dahil edildiği Petri ağı olarak ifade edilmektedir. Bu makalede, en fazla geçişe izin veren durum geri beslemesi için, bağlantı yolu tabanlı bir uygulama geliştirilmiştir. [12]'de geliştirilen bu uygulama, yine aynı yazarlar tarafından, kesikli olay sistemlerde kapalı döngü canlılığı sağlamak için kullanılmıştır [13]. [12]'de verilen uygulama, Haxoun [14] tarafından geliştirilerek, [12]'deki kontrol yaklaşımı için, S-azaltılması (S-decreases) adlı bir yöntem ortaya konmuştur. İlgili makaledeki, S-azaltılması yöntemi, herhangi bir geçişin ateşlenmesiyle belirtilerin toplam ağırlığı asla artmayan ağdaki yerlerde ağırlık vektörlerinin tanımlanması olarak verilmiştir. Bir Petri ağının kontrolü için gerek ve yeter şartların varlığının ortaya çıkarılması Giua ve DiCesera tarafından gerçekleştirilmiştir [15]. Sreenivas ve Krogh [16] tarafından, sonsuz durumlu kesikli olay sistemler için, otomata ve önleyici bağlantı ilave edilmiş Petri ağı modellemeleri kullanılarak elde edilen kontrolörler karşılaştırılmıştır.

Sreenivas [17], kontrol edilen Petri ağı için canlılığı sağlayan bir kontrolörün gerek ve yeter şartlarını ortaya koymuştur. Sreenivas bu çalışmasında; sınırlı veya her bir geçişi denetlenebilen kontrol edilen Petri ağları için, canlılığı sağlayabilen bir kontrolörün varlığından söz etmiştir. Bu kontrol tarzının varlığı ise, Petri ağının ana yapısına, başlangıç işaretlemesine ve denetlenemeyen işaretlemelerin kümesine bağlıdır. İlgili çalışmadaki kontrolörün yaptığı işlem ise, verilen bir işaretleme için geçişlere izin verilip verilmemesi olarak gösterilmiştir.

Ezpeleta vd. [18] tarafından, esnek (flexible) üretim sistemleri için Petri ağı tabanlı sistem çıkmazını önleyen bir kontrol önerilmiştir. Ancak, ilgili makaledeki çalışmada, Petri ağının özel bir sınıfı olan ve kaynakları ile birlikte basit sıralı işlemlerin sistemi olarak adlandırılan bir model kullanılmıştır. Ayrıca bu makalede esnek üretim sistemlerindeki canlılık, herhangi bir ateşleme dizisinden sonra ateşlenebilecek bir geçiş olarak ifade edilmiş ve her üretim sürecinin yeni ürünler üretebilmesi olarak gösterilmiştir. Ancak yine bu makalede, sistem canlı olmasa bile, sistem çıkmazını önleyen kontrolör yardımıyla, sistemin çalışmasının sağlanabileceği ve yeni ürünler üretilabileceği belirtilmiştir.

Bu tezde, Petri ağı ve otomata gösteriminin herbiri için ayrı olarak, örtüşmeli ayrıştırmaya dayalı birer dışmerkezli kontrol tasarım yöntemi geliştirilmeye çalışılacaktır. *Örtüşmeli ayrıştırmaya* (overlapping decomposition) dayalı dışmerkezli kontrol tasarım yöntemi, Ikeda ve Šiljak [19] tarafından ortaya konmuştur. İlgili çalışmada, sürekli durum sistemleri (diferansiyel veya fark denklemleri ile gösterilen sistemler) olarak modellenen büyük ölçekli sistemler, örtüşen alt sistemlere ayrıştırılarak, dışmerkezli kontrol tasarımı yapılmıştır. Bir dinamik sistemi alt sistemlere ayrıştırabilmek için ilk önce o sistemin zayıf bağlarla bağlanmış kısımlarını belirlemek gerekmektedir. Ancak, pek çok büyük ölçekli sistem birbirlerine belli dinamik parçalar (örtüşen kısım) üzerinden kuvvetli bağlarla, bunlar dışında zayıf bağlarla bağlı alt sistemlerden oluşmaktadır [19]. Bu şekildeki büyük ölçekli sistemlerde, alt sistemlerin birbirlerinden tamamen ayrışık olduğu “ayrışık ayrıştırma” yerine, alt sistemlerin ortak (örtüşen) kısımlarının bulunduğu “örtüşmeli ayrıştırma” uygulamanın daha iyi olabileceği ortaya konulmuştur (örneğin bkz. [20]). Örtüşmeli ayrıştırma yöntemi, trafik ağlarında ve bilgisayar ağlarında başarı ile uygulanmıştır (örneğin bkz. [21–27]).

Hibrid sistemler için dışmerkezli kontrol tasarımında da, İftar ve Özgüner [28] örtüşmeli ayrıştırma yönteminden yararlanmışlardır. Bu çalışmadaki hibrid sistemler, sürekli durum sistemleri ile kesikli durum sistemlerinin beraber kullanıldığı sistemlerdir. İlgili çalışmadaki, hibrid sistemleri oluşturan sistemlerden, kesikli durum sistemler, genellikle sonlu durumlarla ifade edilen sistemlerdir ve ayrıca bu sistemler, kesikli olay sistemlerin de bir alt sınıfıdır. Ele alınan hibrid sistemler vektörel gösterimle ifade edilmiştir. Bu gösterim, otomata gösterimi ile ilişkilendirilebilir. [28]’de ayrıca, *içerme* (inclusion) [19] ve *genişletme* (extension) [29] prensipleri hibrid sistemler için incelenmiş ve hibrid sistemler için kararlılık tanımları verilmiştir.

Bu tezde, Petri ağı ile modellenen kesikli olay sistemler için bir dışmerkezli kontrol algoritması geliştirilmek istenmiştir. Özellikle, bilgisayar ağlarında, örtüşmeli ayrıştırmaya dayalı dışmerkezli kontrol tasarımının gerçekleştirilmiş olmasından [22–26] ve bilgisayar ağlarının grafiksel gösteriminin Petri

ağlarının grafiksel gösterimine benzerliğinden dolayı, bu çalışmada örtüşmeli ayrıştırma ve genleştirme yöntemi kullanımı seçilmiştir. Bu amaca yönelik yapılan araştırma ve çalışmalar aşağıda belirtildiği şekilde ana başlıkları ile ifade edilmiştir.

Bölüm 2’de, daha önce yapılmış olan, Petri ağı yapısının, grafiksel ve matematiksel gösterimi ile ilgili tanımların verildiği çalışmalar [2,7], detaylı olarak incelenerek elde edilen sonuçlar verilmiştir. Bu bölümde, kesikli olay sistemlerin bazı modelleme yöntemleri arasındaki ilişkiler de ortaya konmuştur.

Sreenivas [17] tarafından kullanılan kontrol tekniğinin geliştirilmesi ile elde edilen kontrol yaklaşımı ve elde edilen bu kontrol yaklaşımı ile kesikli olay sistemler için kullanılan diğer bazı kontrol yaklaşımları arasındaki ilişkiler, Bölüm 3’de verilmiştir.

İçerme prensibinin [19], Petri ağlarına uygulanması, Bölüm 4’de verilmiştir. Bu bölümde, otomata ve dil gösterimleri için bir içerme prensibi tanımı da verilmiştir.

Örtüşmeli ayrıştırma işleminin Petri ağlarına uygulanması ve sonra Petri ağının genişletilmesi ile geliştirilmiş Petri ağı (GPA) elde edilmesi, Bölüm 5’de verilmiştir. Ayrıca bu bölümde, geliştirilmiş Petri ağı ile özgün Petri ağı (ÖPA) arasında içerme prensibinin varlığı ve Petri ağına ait olan bazı ilişkiler incelenmiştir. Bölüm 5’de, otomata ve dil gösterimleri içinde örtüşmeli ayrıştırma ve genleştirme işlemlerinin nasıl yapılabileceği anlatılmıştır.

Petri ağı ve otomata gösterimlerinin herbiri için, sistem çıkmazının meydana gelmesini önleyen örtüşmeli ayrıştırmaya dayalı birer dışmerkezli kontrolör tasarım yöntemi Bölüm 6’da verilmiştir. Bu bölümde ayrıca sistem çıkmazına izin vermeyen bir dil tasarım yöntemi de önerilmiştir.

Bu çalışmada elde edilen sonuçlar ise Bölüm 7’de tartışılmıştır.

2. PETRI AĞLARI ve KESİKLİ OLAY SİSTEMLER İÇİN DİĞER BAZI MODELLEME YÖNTEMLERİ

Bu bölümde, Zhou ve F. DiCesare [2] ve Desrochers ve Al-Jaar [7] tarafından yapılan çalışmalarda Petri ağları ile ilgili elde edilen sonuçlar anlatılacaktır. Kesikli olay sistemlerinin diğer gösterimleri de [9,10,28] ince lenerek, bu gösterimler ile Petri ağı gösterimi arasındaki ilişkiler verilecektir.

2.1. Petri Ağı

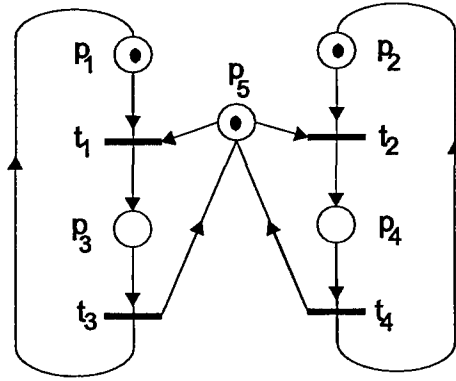
Petri ağı, kesikli olay sistemler için, formal ve grafiksel bir yapıdır. Petri ağı modeli, sistemlerin performans analizinde, simülasyon çalışmalarında ve kontrolör tasarımında kullanılmaktadır.

2.1.1. Petri ağının tanımlanması

Bir önceki bölümde bahsedildiği gibi, Petri ağları yerler, geçişler ve bun lar arasındaki bağlantılardan oluşmaktadır. Yer, bir işlemi veya bir kaynağın durumunu, geçiş ise, bir olayın ya da işlemin, başlangıcını veya bitişini göstermektedir. Burada kaynak, sistemde gerçekleştirilmek istenen olaylar için gerekli olan koşulları ifade etmektedir. Bu koşullar, personel, makine, araç-gereç, vb. olabilir.

Petri ağının grafiksel gösterimi; yerlerin, dairelerle, geçişlerin çizgilerle ve yerlerden geçişlere ve geçişlerden yerlere olan bağlantıların da oklarla ifade edilmesidir. Bir olayın veya bir işlemin başlamasını gösteren geçişler ile ilgili yerler arasındaki bağlantı; geçişlerden yerlere doğru olan oklarla gösterilirken, bir olayın veya bir işlemin bitişini gösteren geçişler ile ilgili yerler arasındaki bağlantı ise; yerlerden geçişlere doğru olan oklarla gösterilmektedir (bkz. Şekil 2.1).

Petri ağının grafiksel gösteriminde yerlere, geçişlere ve oklara ilave olarak belirtiler de vardır. Belirtilerin de bulunduğu Petri ağına, işaretlenmiş Petri ağı denir. Bir belirti, grafiksel olarak yerlerde “•” ile gösterilir. Petri ağında, bir yerde belirti olması, o yerin gösterdiği işlemin yapılıyor olduğunu veya belirtinin sayısına göre kaynağın durumunu gösterir. Kesikli olay sistemin kay-



Şekil 2.1. Petri ağı

naklarının başlangıçtaki durumunun (sayısı, pozisyonu, vb.), Petri ağındaki gösterimine başlangıç işaretlemesi denir. Bu da, Petri ağında kaynakları gösteren yerlere, uygun sayıda belirti konulması ile olur (bkz. Şekil 2.1).

Petri ağıının çalışması; belirti veya belirtilerin bir geçişin ateşlenmesiyle bir yerden başka bir yere iletilmesi olarak ifade edilir. Bir geçişin ateşlenmesi için, geçişe girdi olarak (oklarla) bağlı olan yerlerin her birinde en az bir belirti olmalıdır. Böyle bir geçişin ateşlenmesiyle, bu geçişe çıktı olarak (oklarla) bağlı olan her bir yere bir belirti ulaşırken, girdi olarak bağlı olan yerlerden de birer belirti eksilmiş olacaktır.

P : yerlerin kümesini, T : geçişlerin kümesini ($P \cap T = \emptyset$ ve $P \cup T \neq \emptyset$), $N : P \times T \rightarrow \{0, 1\} =: \mu$ yerlerden geçişlere doğru olan bağlantıların gösterildiği girdi matrisini, $O : P \times T \rightarrow \mu$ geçişlerden yerlere doğru olan bağlantıların gösterildiği çıktı matrisi ve $m_0 : P \rightarrow \mathcal{N}$ Petri ağıında başlangıç işaretleme vektörünü göstermek üzere (\mathcal{N} : doğal sayılar kümesi), $G(P, T, N, O, m_0)$ gösterimi ile bir Petri ağı ifade edilebilir. N ve O matrislerinin satırlarına, sırasıyla, T kümesinin elemanları ve sütunlarına, sırasıyla, P kümesinin elemanları karşılık gelmektedir. $N(p, t)$ ($O(p, t)$), N (O) matrisinde p yeri ile t geçişi arasındaki bağlantının değerini göstermek üzere, p yerinden t geçişine, bir ok ile bağlantı var ise, $N(p, t) = 1$, aksi durumda $N(p, t) = 0$ ve t geçişinden p yerine, bir ok ile bağlantı var ise, $O(p, t) = 1$, aksi durumda $O(p, t) = 0$ olur. Ayrıca, $M : P \rightarrow \mathcal{N}$ Petri ağıında işaretleme vektörünü, $M(p)$: M işaretleme vektörünün p yerindeki belirtilerin sayısını,

$|\cdot| : (\cdot)$ kümesinin eleman sayısını ve $\lambda(M)$: M vektörünün en büyük elemanını göstermektedir.

Petri ağının çalışması matematiksel olarak;

$$M'(p) = M(p) + O(p, t) - N(p, t), \quad p \in P, \quad t \in T$$

şeklinde gösterilebilir [2]. Burada, $M(p)$, p yerindeki bir önceki belirti sayısını ve $M'(p)$, t geçişinin ateşlendikten sonra p yerindeki belirti sayısını göstermektedir. Petri ağında, eğer ve yalnızca eğer $N(p, t) = 1$ iken $M(p) > 0$ ($\forall p \in P$) oluyorsa, t ateşlemesi yapılabilir.

Bu matematiksel gösterim, Petri ağı denklemi olarak adlandırılarak, aşağıdaki şekilde vektör formunda düzenlenebilir:

$$M' = M + O_t - N_t, \quad t \in T \quad (2.1)$$

Burada, M' ; p yerlerindeki son belirtilerin sayısını gösteren işaretleme vektörü ($M' \in \mu^{|P| \times 1}$), M ; p yerlerindeki bir önceki belirtilerin sayısını gösteren işaretleme vektörü ($M \in \mu^{|P| \times 1}$), O_t ; O matrisinin t geçişine karşılık gelen sütunu ($O_t \in \mu^{|P| \times 1}$) ve N_t ; N matrisinin t geçişine karşılık gelen sütunu ($N_t \in \mu^{|P| \times 1}$) göstermektedir.

Petri ağında, peşpeşe ateşlenen geçişlerin oluşturduğu diziye, ateşleme dizisi denir ve $g \triangleq t_{i_1}, t_{i_2}, \dots, t_{i_k}; \quad t_{i_1}, t_{i_2}, \dots, t_{i_k} \in T$ şeklinde gösterilir. Bu g ateşleme dizisindeki bütün geçişlerin sırasıyla (t_{i_1} geçişinden başlayarak en son t_{i_k} geçişinin) ateşlenmesiyle M gibi herhangi bir işaretlemeden, bir M' işaretlemesi elde edilebiliyor ise, bu işlem denklem (2.1)'den yararlanılarak,

$$\begin{aligned} M^1 = M + O_{t_{i_1}} - N_{t_{i_1}} &\implies \dots \implies M' = M^{k-1} + O_{t_{i_k}} - N_{t_{i_k}} \\ M \xrightarrow{t_{i_1}} M^1 &\implies \dots \implies M^{k-1} \xrightarrow{t_{i_k}} M' \end{aligned}$$

$$M \xrightarrow{t_{i_1}, t_{i_2}, \dots, t_{i_k}} M' \implies M \xrightarrow{g} M'$$

şeklinde ifade edilir. Elde edilen bu ifadeyi, T^* , T kümesindeki geçişlerden oluşan tüm sonlu dizilerin oluşturduğu kümeyi göstermek üzere, $\rho : T^* \times P \rightarrow \mathcal{N}$ şeklinde bir kısmi fonksiyon ile gösterebiliriz¹. Bu kısmi fonksiyon,

¹ ρ , tanım kümesi olan $T^* \times P$ 'nin her elemanı için tanımlı olmadığından dolayı,

iletim fonksiyonu olarak adlandırılabilir. İletim fonksiyonu kullanılarak, M işaretleme vektöründen sonra, ateşlenebilen geçiş ya da geçişlerin oluşturduğu bir g ateşleme dizisi kullanılmasıyla M' işaretleme vektörü elde edilmektedir.

Yukarıda verilen ifade düzenlenerek,

$$M' = [(M + O_{t_{i_1}} - N_{t_{i_1}}) + O_{t_{i_2}} - N_{t_{i_2}}] + \dots + O_{t_{i_k}} - N_{t_{i_k}} \quad (2.2a)$$

$$= M + O_{t_{i_1}} - N_{t_{i_1}} + O_{t_{i_2}} - N_{t_{i_2}} + \dots + O_{t_{i_k}} - N_{t_{i_k}} \quad (2.2b)$$

$$= M + (O_{t_{i_1}} + O_{t_{i_2}} + \dots + O_{t_{i_k}}) - (N_{t_{i_1}} + N_{t_{i_2}} + \dots + N_{t_{i_k}}) \quad (2.2c)$$

$$= M + \hat{O}_g - \hat{N}_g =: \rho(g, M) \quad (2.2d)$$

elde edilir. Böylece $\rho(g, M)$ fonksiyonu, vektör matris formunda ifade edilmiş olur. Burada $\hat{O}_g \triangleq (O_{t_{i_1}} + O_{t_{i_2}} + \dots + O_{t_{i_k}})$, $\hat{N}_g \triangleq (N_{t_{i_1}} + N_{t_{i_2}} + \dots + N_{t_{i_k}})$ olarak gösterilmektedir. Ayrıca, $A \triangleq (O - N)$, $A : P \times T \rightarrow \{-1, 0, 1\}$ ve $U_g \in \mathcal{N}^{|T| \times 1}$, g ateşleme dizisini meydana getiren geçişlerin, bu dizi içinde kaç kez kullanıldığını gösteren vektör olmak üzere, denklem (2.2d), aşağıdaki şekilde düzenlenebilir:

$$M' = M + (O - N)U_g = M + AU_g \quad (2.3)$$

Burada kullanılan, A matrisi *bağlantı matrisi* ve U_g vektörü de *ağırlık vektörü* olarak adlandırılabilir. Denklem (2.3)'de, g dizisinin bir ateşleme dizisi olduğu, yani sırası geldiğinde her bir geçişin ateşlenebildiği varsayılmıştır. Aksi durumda, yukarıdaki denklem $M'(p) \geq 0$, $\forall p \in P$ şartını sağlasa bile geçerli değildir, yani $M' \notin R(G, m_0)$ 'dir.

Bu çalışmada Petri ağı topolojisinde, $M = M + AU_t$ eşitliğini sağlayan bir $t \in T$ geçişinin olmadığı kabullenmesi yapılmıştır. Böyle bir geçişin var olduğu Petri ağında, bu geçiş ateşlendiği zaman bir değişiklik meydana gelmemektedir.

bir fonksiyon değil kısmi fonksiyondur. Ancak bu çalışmada, ρ iletim fonksiyonu olarak adlandırılmaktadır.

Örnek 2.1:

Şekil 2.1'deki Petri ağında, $P = \{p_1, p_2, p_3, p_4, p_5\}$, $T = \{t_1, t_2, t_3, t_4\}$ ve $m_0 = [1 \ 1 \ 0 \ 0 \ 1]^T$ 'dir. m_0 , başlangıç işaretlemesindeki elemanlar, sırasıyla P kümesindeki yerlerin belirti sayılarını göstermektedir. Şekil 2.1'de gösterilen Petri ağının başlangıç işaretlemesi, hem t_1 hem de t_2 geçişinin ateşlenmesine olanak vermektedir. Bu durumda t_1 geçişi ateşlendiği zaman $M_1 = [0 \ 1 \ 1 \ 0 \ 0]^T$ 'e, t_2 ateşlendiği zaman $M_2 = [1 \ 0 \ 0 \ 1 \ 0]^T$ 'ye ulaşılır. t_1 ateşlendikten sonra t_3 , t_2 ateşlendikten sonra t_4 ateşlenebilir. Ayrıca $t_1 t_3 t_2 t_4$ (veya $t_2 t_4 t_1 t_3$) ateşleme dizisiyle $M_f = m_0 = [1 \ 1 \ 0 \ 0 \ 1]^T$ işaretleme vektörüne ulaşılır. Bu Petri ağının, girdi ve çıktı matrisleri aşağıdaki şekilde elde edilmektedir.

$$N = \begin{array}{c} \left[\begin{array}{cccc|c} t_1 & t_2 & t_3 & t_4 & \\ \hline 1 & 0 & 0 & 0 & p_1 \\ 0 & 1 & 0 & 0 & p_2 \\ 0 & 0 & 1 & 0 & p_3 \\ 0 & 0 & 0 & 1 & p_4 \\ 1 & 1 & 0 & 0 & p_5 \end{array} \right] \end{array} \quad \text{ve} \quad O = \begin{array}{c} \left[\begin{array}{cccc|c} t_1 & t_2 & t_3 & t_4 & \\ \hline 0 & 0 & 1 & 0 & p_1 \\ 0 & 0 & 0 & 1 & p_2 \\ 1 & 0 & 0 & 0 & p_3 \\ 0 & 1 & 0 & 0 & p_4 \\ 0 & 0 & 1 & 1 & p_5 \end{array} \right] \end{array}$$

2.1.2. Petri ağının özellikleri

G ile gösterilen Petri ağında, M işaretleme vektöründen başlayarak elde edilen tüm işaretleme vektörlerinin oluşturduğu kümeye *ulaşılabilirlik kümesi* denir ve $R(G, M)$ ile gösterilir. Şekil 2.1'de gösterilen Petri ağının ulaşılabilirlik kümesi, $R(G, m_0) = \{[1 \ 1 \ 0 \ 0 \ 1]^T, [0 \ 1 \ 1 \ 0 \ 0]^T, [1 \ 0 \ 0 \ 1 \ 0]^T\}$ olarak elde edilir. [2]'de bahsedilen, Petri ağı ile ilgili bazı tanımlar aşağıda verilmiştir.

Tanım 2.1: Eğer ve yalnızca eğer $M(p) \leq k$, $\forall M \in R(G, m_0)$ sağlayacak şekilde herhangi bir $k \in \mathcal{N}$ varsa, $p \in P$ k ile *sınırlıdır*.

Tanım 2.2: Eğer ve yalnızca eğer p ($\forall p \in P$), k ile sınırlı ise, Petri ağı da k ile *sınırlıdır*. Burada eğer ve yalnızca eğer $k = 1$ ise, Petri ağı *güvenlidir*.

Tanım 2.3: Eğer ve yalnızca eğer $\forall M \in R(G, m_0)$ işaretlemesinden sonra t ($\forall t \in T$) geçişine izin verecek şekilde bir g ateşleme dizisi varsa Petri ağı *canlıdır*.

Tanım 2.4: Eğer, herhangi bir $M \in R(G, m_0)$ işaretlemesinden sonra, hiç

bir $t \in T$ geçişi yapılamıyorsa, Petri ađında *sistem çıkmazı* meydana gelir.

Not 2.1: Petri ađının canlı olması, sistem çıkmazı olmamasını garanti eder (bkz. [2]).

Tanım 2.5: Eđer ve yalnızca eđer $m_o \in R(G, M)$, $\forall M \in R(G, m_o)$ ise, G ile gösterilen Petri ađı *tersine dönüřebilir*dir.

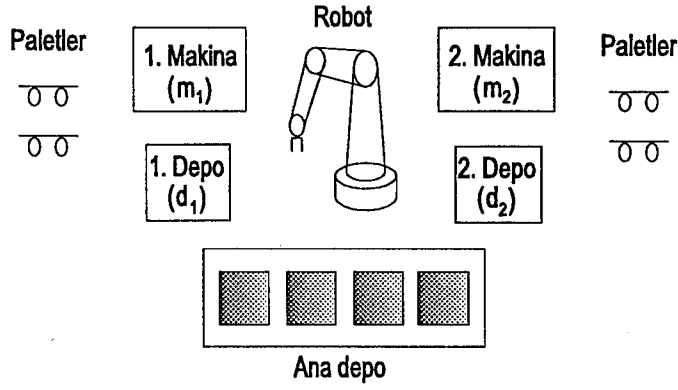
Tanım 2.6: Eđer ve yalnızca eđer, $\sum_{p \in P} w(p)M(p) = K$, $\forall M \in R(G, m_o)$ denklemini sađlayan bir w vektörü ($w(p) > 0$, $\forall p \in P$) ve $K \in \mathcal{N}$ var ise, G ile gösterilen Petri ađı *korunumludur*. Eđer $w = [1 \ 1 \ \dots \ 1]^T$ ise, iřaretlenmiř Petri ađı *tam korunumlu* olarak adlandırılır.

Not 2.2: Eđer bir Petri ađı tam korunumlu ise belirtilerin toplamı bütün ulařılabilir iřaretlemeler için deđiřmeden kalacaktır.

2.1.3. Petri ađı ile sistem modellemesi

Kesikli olay sistemlere bakıldıđı zaman birbirlerini etkileyen çeřitli olaylar görüldür. Bu olayların belirli bir sırası vardır. Bazı olayların, bazı olaylardan önce olması gerekir [2]. Bir sistemin modellenmesi [2];

- 1) Sistemin her ařaması için gerekli olan olaylar ve kaynaklar belirlenmelidir.
- 2) Sistemin çalışması esnasında, olayların meydana geliř sıralaması belirlenmelidir.
- 3) Sıradaki her bir olay için:
Olayın durumunu gösteren bir yer oluřturulur. Bu yere girdi ve çıktı olarak birer geçiř ilave edilir. Bađlantı řekli kendisinden yere dođru olan geçiř (olayı bařlatma), çıktı olarak ve bađlantı řekli yerden kendisine dođru olan geçiř (olayı sonlandırma), girdi olarak kabul edilir. Bir olay için sonlandırma geçiři, bařka bir olay için bařlangıç geçiři olabilir. Ayrıca ađdaki bazı olaylar için birden fazla belirtinin olması gerekebilir.
- 4) Sıradaki her bir olay için:
Eđer yukarıda anlatıldıđı řekilde bir yer oluřturulmamıř ise, her bir kaynak bařlangıç olayını göstermek üzere bir yer olarak ele alınır. Bütün



Şekil 2.2. Örnek 2.2'de ele alınan üretim sistemi

uygun kaynaklar diğer yerlere girdi olarak bağlanır. Ayrıca, sonlandırma geçişleri de olayların gösterildiği yerlerden kaynaklara doğru bağlanır.

5) En son olarak bu sistem için başlangıç işaretlemesi yapılır.

Örnek 2.2:

Şekil 2.2'de verilen sistemde, iki tane makine (m_1 ve m_2), bir tane robot ve dört tane de palet vardır. Bu paletlerden, ikisi m_1 için, diğer ikisi ise m_2 için kullanılmaktadır. Ayrıca sistemde, her bir makine için bir mamül kapasiteli iki depo (d_1 ve d_2) ve her biri bir mamül kapasiteli dört bölmeden oluşan bir ana depo (d) bulunmaktadır.

Sistemin çalışması, sırasıyla aşağıdaki şekilde olmaktadır:

- m_1 (m_2), bir palet üzerine sabitleştirilmiş parçaların üretimini yapar.
- Robot, m_1 'de (m_2) üretilen mamülleri d_1 'e (d_2) yükler.
- d_1 (d_2) robot tarafından boşaltılır. Bu sırada, kullanılan palet serbest kalmaktadır.
- Paletten ayrılan mamüller ana depoda saklanır. Ana depo yine robot tarafından boşaltılmaktadır.

İşleyişi verilen sistemin, Petri ağı modelini oluşturmak üzere yukarıda verilen yöntem uygulanarak aşağıdaki Petri ağı modeli elde edilir.

- 1) Sistemin kaynakları, robot, paletler (dört paletten, ikisi m_1 , ikisi de m_2 için kullanılmaktadır), depolar (sistemde, her biri bir makinenin ürettiği mamülleri saklayan birer depo ile üretilen tüm mamüllerin saklandığı dört ayrı bölmeden oluşan bir ana depo bulunmaktadır) ve 1. ve 2. makinelerdir. Sistemdeki olaylar, m_1 'de veya m_2 'de, bir palet üzerine sabitleştirilmiş parçaların üretiminin yapılması, robotun m_1 'de (m_2) üretilen mamülleri d_1 'e (d_2) yüklemesi, d_1 'de (d_2) mamüllerin saklanması, robotun d_1 'i (d_2) boşaltması ve paletlerin serbest kalması, paletten ayrılan mamüllerin ana depoda saklanması ve ana deponun yine robot tarafından boşaltılmasıdır.
- 2) Sistemde, m_1 'de (m_2), bir palet üzerine sabitleştirilmiş parçaların üretimi yapıldıktan sonra robot, m_1 'de (m_2) üretilen mamülleri d_1 'e (d_2) boşaltır. Daha sonra d_1 'deki mamül de ana depoya boşaltılır. Bu sırada, kullanılan palet serbest kalır. En son olarak, ana depo yine robot tarafından boşaltılır.
- 3) p_1 ; m_1 için kullanılabilir paletlerin sayısını, p_2 ; m_1 'in üretim yaptığını, p_3 ; m_1 'in uygun olup olmadığını, p_4 ; robotun m_1 'e hizmet vermesini, p_5 ; m_1 tarafından üretilen mamüllerin d_1 'de saklanmasını, p_6 ; d_1 'in boş olup olmadığını, p_7 ; robotun dolan d_1 'i boşaltmasını, p_8 ; paletten ayrılmış mamüllerin ana depoda saklanmasını, r ; robotun uygun olup olmadığını, d ; ana deponun kullanılabilir bölme sayısını, p_9 ; m_2 için kullanılabilir paletlerin sayısını, p_{10} ; m_2 'nin üretim yaptığını, p_{11} ; m_2 'nin uygun olup olmadığını, p_{12} ; robotun m_2 'ye hizmet vermesini, p_{13} ; m_2 'de üretilen mamüllerin d_2 'de saklanmasını, p_{14} ; d_2 'nin boş olup olmadığını, p_{15} ; robotun dolan d_2 'yi boşaltmasını göstermek üzere, $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, r, d, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$ şeklinde oluşturulur. t_1 ; m_1 'in üretime başlamasını, t_2 ; p_2 'nin bitişini ve robotun m_1 'e hizmet vermeye başlamasını, t_3 ; p_4 'ün bitiş ve m_1 'de üretilen

mamüllerin robot tarafından d_1 'e yüklenmesini, t_4 ; p_5 'in bitişini ve robotun d_1 'i boşaltmaya başlamasını, t_5 ; p_7 'nin bitişini ve robotun d_1 'deki mamülü ana depodaki bir bölmeye yüklemesini, t_6 ; robotun ana depodaki bir bölmeyi boşaltmasını, t_7 ; m_2 'nin üretime başlamasını, t_8 ; p_{10} 'nun bitişini ve robotun m_2 'ye hizmet vermeye başlamasını, t_9 ; p_{12} 'nin bitişini ve m_2 'de üretilen mamüllerin robot tarafından d_2 'ye yüklenmesini, t_{10} , p_{13} 'ün bitişini ve robotun d_2 'yi boşaltmaya başlamasını, t_{11} ; p_{15} 'in bitişini ve robotun d_2 'deki mamülü ana depodaki bir bölmeye yüklemesini gösteren geçişlerdir. $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}\}$ şeklinde oluşturulur.

- 4) Yerler ile geçişler arasındaki bağlantılarsa; $N : P \times T \rightarrow \mu$ ve $O : P \times T \rightarrow \mu$ olmak üzere elde edilir (bkz. Çizelge 2.1 – 2.2).

Çizelge 2.1. Şekil 2.3'deki Petri ağının girdi matrisi

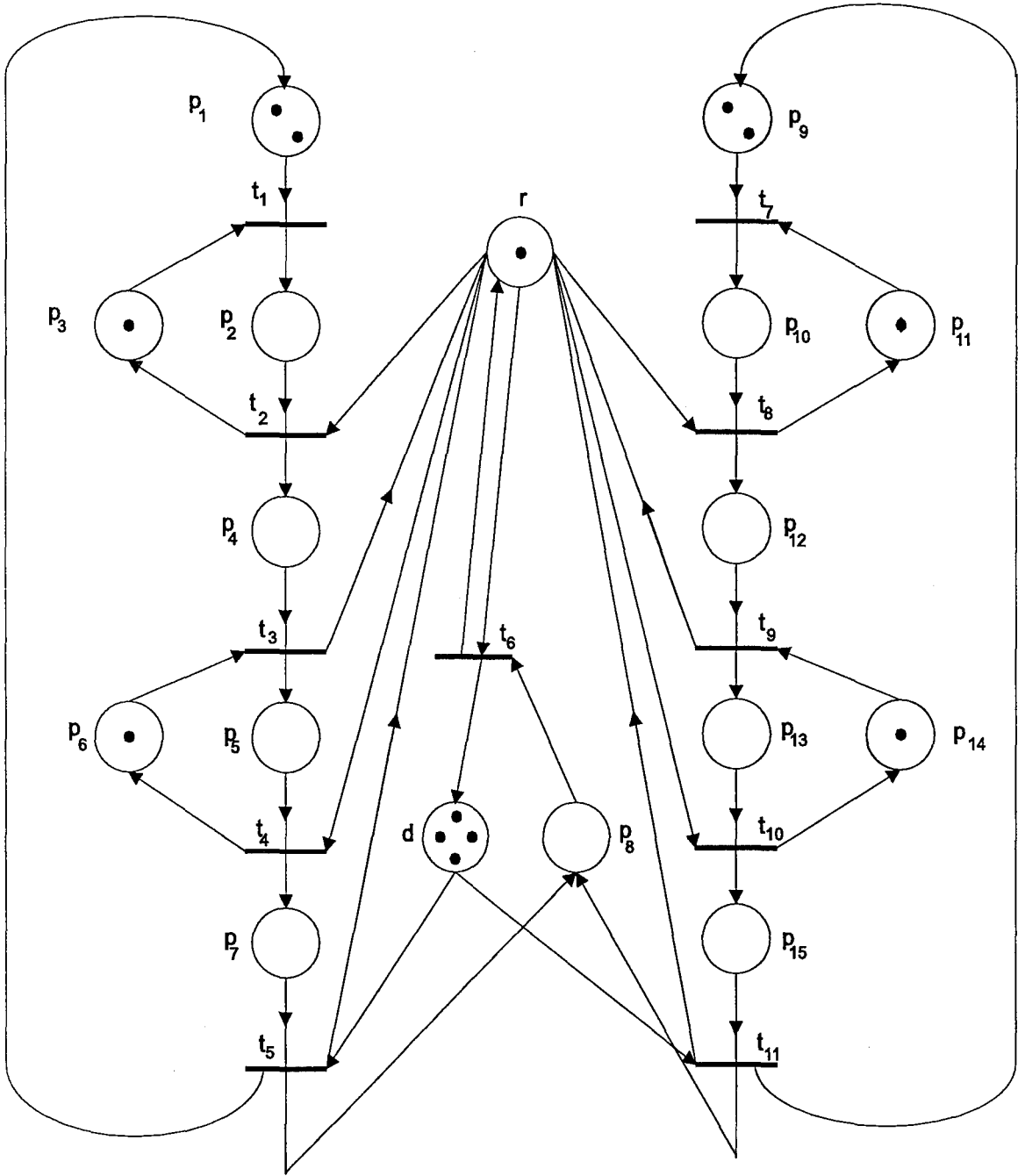
$$N = \begin{array}{c} \left[\begin{array}{cccccccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \begin{array}{l} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ r \\ d \\ p_9 \\ p_{10} \\ p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{15} \end{array} \end{array}$$

Çizelge 2.2. Şekil 2.3'deki Petri ağının çıktı matrisi

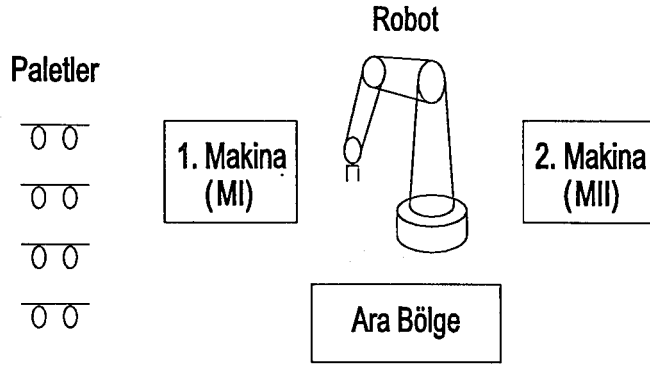
$$O = \begin{array}{c|ccccccccccc|c} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} & \\ \hline & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & p_1 \\ & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_2 \\ & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_3 \\ & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_4 \\ & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_5 \\ & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_6 \\ & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_7 \\ & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & p_8 \\ & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & r \\ & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & d \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & p_9 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & p_{10} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & p_{11} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & p_{12} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & p_{13} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & p_{14} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & p_{15} \end{array}$$

- 5) Şekil 2.2'de gösterilen sistemin başlangıç durumunu ifade etmek için; paletlerin sayısını ve ana deponun bölmelerinin sayısını gösteren yerler ile robot, makine ve depoların uygun olmalarını gösteren yerlere belirti atanmasıyla Petri ağının başlangıç işaretlemesi elde edilir ve $m_0 = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 4 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ olarak gösterilir. m_0 vektöründeki elemanlar, sırasıyla P kümesindeki yerlerin belirti sayılarını göstermektedir.

Böylece Şekil 2.2'de gösterilen sistem, Şekil 2.3'de gösterilen Petri ağı ile modellenmiş olur.



Şekil 2.3. Örnek 2.2'de verilen sistemin Petri ağı modeli



Şekil 2.4. Örnek 2.3'de ele alınan üretim sistemi

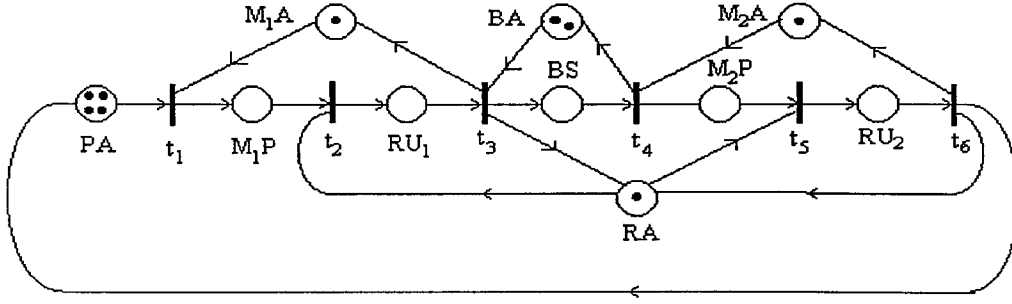
Örnek 2.3:

Şekil 2.4'de gösterilen sistemde, iki tane makine (MI ve MII), bir tane robot, dört tane palet ve bir tane de ara bölge (ara bölgede, her biri bir ürün saklayabilen 2 tane depo bulunmaktadır) vardır. Sistemin işleyişi aşağıda verilmiştir:

- MI, MII'den önce çalışmaya başlamakta,
- MI yüklenerken, palet sabitleştirilmekte,
- Ürün meydana geldikten sonra robot ürünü ara bölgeye boşaltmakta,
- MII yüklenerken çalışmaya başlamakta,
- MII robot tarafından boşaltılmakta ve bu sırada kullanılan palet serbest kalmaktadır.

İşleyişi verilen sistemin, Petri ağı modelini oluşturmak üzere yukarıda verilen yöntem uygulanarak aşağıdaki Petri ağı modeli elde edilir.

- 1) Sistemin kaynakları, robot, paletler (dört tane), ara bölge ve 1. ve 2. makinelerdir. Sistemdeki olaylar, MI'in yüklenmesi (paletler sabitleştirilerek), MI'de ürün meydana geldikten sonra robotun ürünü ara bölgeye boşaltması, MII'nin yüklenerken çalışmaya başlaması ve robotun, üretilen son mamülü boşaltmasıdır (paletler de serbet kalırlar).
- 2) MI, MII'den önce çalışmaya başlamaktadır.



Şekil 2.5. Örnek 2.3'de verilen sistemin Petri ağı modeli

- 3) M_1P ; MI'in yüklenmesi, palet sabitleştirilmesi, üretimi, RU_1 ; robotun MI'i boşaltması, BS ; ara bölgede mamüllerin saklanması, M_2P ; MII'nin yüklenmesi ve üretimi, RU_2 ; robotun MII'yi boşaltması ve palet çözümü, PA ; kullanılabilir palet sayısı (örnek sistemde 4 tane palet vardır), M_1A ; MI'in uygun olduğu, RA ; robotun uygun olduğu, M_2A ; MII'nin uygun olduğu, BA ; ara bölgede kullanılabilir depo sayısı, şeklinde Petri ağının yerleri gösterilmektedir. t_1 ; M_1P 'nin başlama geçişi, t_2 ; M_1P 'nin bitiş ve RU_1 'nin başlama geçişi, t_3 ; RU_1 'in bitiş ve BS 'nin başlama geçişi, t_4 ; BS 'nin bitiş ve M_2P 'nin başlama geçişi, t_5 ; M_2P 'nin bitiş ve RU_2 'nin başlama geçişi, t_6 ; RU_2 'nin bitiş geçişi, olarak Petri ağının geçişleri ifade edilmektedir. Buradan, $P = \{PA, M_1P, M_1A, RU_1, RA, BS, BA, M_2P, M_2A, RU_2\}$ ve $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ olarak elde edilir.

- 4) Yerler ile geçişler arasındaki bağlantılarsa;

$N : P \times T \rightarrow \mu$ ve $O : P \times T \rightarrow \mu$ olmak üzere, girdi ve çıktı matrisleri aşağıdaki şekilde elde edilir.

$$N = \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{array} \\ \begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \begin{array}{l} PA \\ M_1P \\ M_1A \\ RU_1 \\ RA \\ BS \\ BA \\ M_2P \\ M_2A \\ RU_2 \end{array} \end{array} \quad \text{ve} \quad O = \begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

- 5) Petri ağının başlangıç işaretlemesi, $m_0 = [4 \ 0 \ 1 \ 0 \ 1 \ 0 \ 2 \ 0 \ 1 \ 0]^T$ 'dir. m_0 vektöründeki elemanlar, sırasıyla P kümesindeki yerlerin belirti sayılarını göstermektedir.

Böylece, Şekil 2.4'de verilen sistem, Şekil 2.5'de gösterilen Petri ağı ile modellenmiş olur.

2.2. Kesikli Olay Sistemlerin Diğer Bazı Modelleme Yöntemleri İle Petri ağı Modellemesi Arasındaki İlişki

Tezin giriş bölümünde belirtildiği gibi, kesikli olay sistemlerin çok çeşitli modelleme yöntemleri vardır. Bu modellemelerden, kesikli olay sistemlerin ifadesinde kullanılan otomata ve dil gösterimleri ile Petri ağı gösterimi arasındaki ilişki ve ayrıca vektörel gösterim [28] ile Petri ağı gösterimi arasındaki ilişki bu bölümde incelenecektir.

2.2.1. Otomata ve dil gösterimleri ile Petri ağı gösterimi arasındaki ilişki

[9] ve [10]'daki çalışmalarda, Σ , olayların oluşturduğu kümeyi, Σ^* , Σ kümesindeki olaylardan oluşan tüm sonlu dizilerin oluşturduğu kümeyi (bu kümeye, ϵ ile gösterilen boş dizi de ilave edilmiştir), Q , durumların

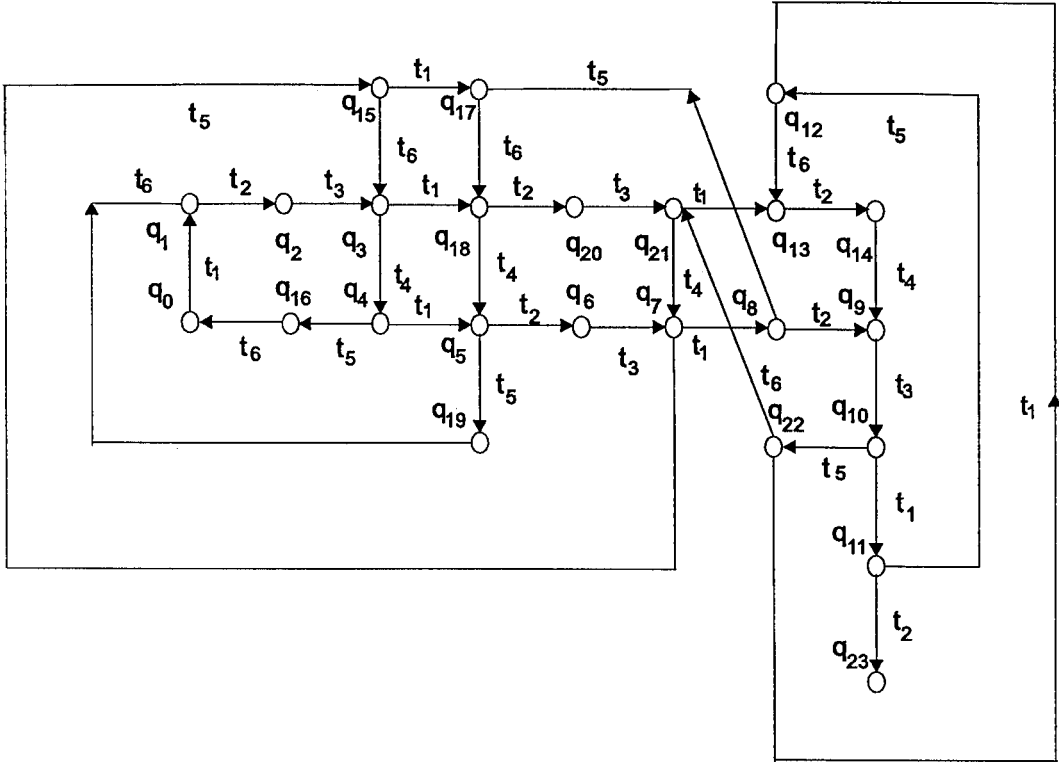
oluşturduğu kümeyi, q_0 , başlangıç durumunu ve $\gamma : \Sigma^* \times Q \rightarrow Q$, bir kısmi fonksiyonu göstermektedir². Bu kısmi fonksiyon, *geçiş fonksiyonu* olarak adlandırılmaktadır. $\gamma(s, q) = q'$, γ geçiş fonksiyonu kullanılarak, $q \in Q$ durumundan sonra $s \in \Sigma^*$ olay dizisi kullanılmasıyla $q' \in Q$ durumu elde edilmektedir. Dil de, $L = \{s \in \Sigma^* \mid \gamma(s, q_0) \text{ tanımlı ise}\}$ olarak tanımlanmıştır.

Otomata gösterimi, kesikli olay sistemlerin dil gösterimindeki durumlar ($q \in Q$) ile bu durumların, hangi olay ($t \in \Sigma$) ve olayların hangi sırada meydana geldiğini gösteren bir grafiksel gösterimdir. Otomata gösteriminde, herbir durum “o” ile gösterilirken, bir durumdan başka bir durumun elde edilmesini sağlayan olaylar ise oklar ile gösterilmektedir (okun yönü, ulaşılan yeni durumu gösterecek şekilde alınır; bkz. Şekil 2.6). Örnek 2.3’de verilen sistemin otomata olarak gösterimi Şekil 2.6’da verilmiştir.

Bölüm 2.1.1’de verilen Petri ağı modeli ile otomata ve dil gösterimleri arasındaki ilişki aşağıda verilmiştir.

- Petri ağında, bir olayın ya da bir işlemin başlatılması ve bitirilmesi geçişler kullanılarak yapılmaktadır. Bu nedenle, Σ kümesi, T kümesi ile ilişkilendirilir. Ayrıca Σ^* kümesinin elemanları da, T kümesinin elemanlarından oluşturulan ateşleme dizileri ile ilişkilendirilir. Burada, Σ^* kümesinin elemanı olan ϵ durumuna, Petri ağında hiçbir geçişin ateşlenmemesine karşılık gelmektedir. Örnek 2.3’de verilen sistem için, $\Sigma = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ ($\Sigma = T$) ve $\Sigma^* = \{\epsilon, t_1, t_1t_2, t_2t_1, t_1t_2t_3, \dots\}$ olarak bulunur.
- Petri ağında ulaşılabilirlik kümesi, başlangıç işaretlemesinden başlayarak elde edilen tüm işaretleme vektörlerinin oluşturduğu kümedir (bkz. Bölüm 2.1.2). Ulaşılabilirlik kümesinin elemanı olan her bir işaretleme vektörü, kesikli olay sistemlerindeki bir duruma karşılık gelmektedir. Dolayısıyla, Q kümesi, $R(G, m_0)$ kümesi ile ilişkilendirilir. Örnek 2.3’de verilen sistem için, $Q = \{q_0, q_1, q_2, \dots, q_{23}\}$ olarak bulunur (bkz. Çizelge 2.3).

² γ , tanım kümesi olan $\Sigma^* \times Q$ ’nun her elemanı için tanımlı olmadığından dolayı, bir fonksiyon değil kısmi fonksiyondur. Ancak bu çalışmada, ([9,10] vb. pek çok çalışmada olduğu gibi) γ geçiş fonksiyonu olarak adlandırılmaktadır.



Şekil 2.6. Örnek 2.3'de verilen sistemin otomata modeli

- Ulaşılabilirlik kümesi ile Q kümesi ilişkilendirildiğinden, m_0 da q_0 ile ilişkilendirilir.
- Petri ağında bir işaretleme vektöründen, başka bir işaretleme vektörü; T kümesinin bir elemanı olan ve ele alınan işaretleme vektöründen sonra ateşlenebilecek bir geçişin, ρ iletim fonksiyonunda kullanılması ile elde edilir. Bu nedenle ρ fonksiyonu, γ fonksiyonu ile ilişkilendirilir.
- Ulaşılabilirlik kümesinin elemanı olan işaretleme vektörlerinin elde edilmesini sağlayan tüm ateşleme dizilerinin oluşturduğu küme, L ile ilişkilendirilir. Örnek 2.3'de verilen sistem için, $L = \{\epsilon, t_1, t_1 t_2, t_1 t_2 t_3, \dots\}$ olarak bulunur.

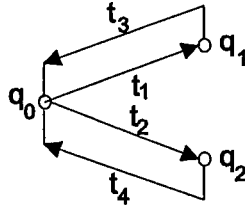
Örnek 2.2'de verilen sistemin modellenmesinde, dil veya otomata gösterimi kullanıldığında; $\Sigma = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}\}$ şeklinde bulunur. Ancak, sistemin durumlarının sayısı oldukça fazladır. Bu nedenle, Örnek 2.2'de verilen sistem için dil veya otomata modellemesi yerine Petri ağı modellemesinin kullanılması, sistem üzerinde yapılacak olan incelemelerin daha anlaşılabilir ve çalışılabilir olmasını sağlamaktadır.

Çizelge 2.3. Örnek 2.3'de verilen sistemin $R(G, m_0)$ ve Q kümelerinin elemanları
(Devamı bir sonraki sayfada verilmiştir)

$R(G, m_0)$ kümesinin elemanları	Q kümesinin elemanları
$m_0 = [4\ 0\ 1\ 0\ 1\ 0\ 2\ 0\ 1\ 0]^T$	q_0 , Sistemin başlangıç durumu
$M_1 = [3\ 1\ 0\ 0\ 1\ 0\ 2\ 0\ 1\ 0]^T$	q_1 , MI'e birinci paletin yüklendiği ve MI'in işlem yaptığı durumu
$M_2 = [3\ 0\ 0\ 1\ 0\ 0\ 2\ 0\ 1\ 0]^T$	q_2 , Robotun, birinci paletin kullanıldığı MI'e hizmet verdiği durumu (depolar boşken)
$M_3 = [3\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0]^T$	q_3 , MI tarafından üretilen mamülün (birinci palet ile birlikte) ara bölgedeki iki depodan birine boşaltılması durumu
$M_4 = [3\ 0\ 1\ 0\ 1\ 0\ 2\ 1\ 0\ 0]^T$	q_4 , Depodan yüklenerek, MII'nin işlem yaptığı durumu
$M_5 = [2\ 1\ 0\ 0\ 1\ 0\ 2\ 1\ 0\ 0]^T$	q_5 , MI'e ikinci paletin yüklendiği ve MI'in işlem yaptığı durumu
$M_6 = [2\ 0\ 0\ 1\ 0\ 0\ 2\ 1\ 0\ 0]^T$	q_6 , Robotun, ikinci paletin kullanıldığı MI'e hizmet verdiği durumu (depolar boşken)
$M_7 = [2\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0]^T$	q_7 , MI tarafından üretilen mamülün (ikinci palet ile birlikte) ara bölgedeki iki depodan birine boşaltılması durumu
$M_8 = [1\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0]^T$	q_8 , MI'e üçüncü paletin yüklendiği ve MI'in işlem yaptığı durumu
$M_9 = [1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0]^T$	q_9 , Robotun, üçüncü paletin kullanıldığı MI'e hizmet verdiği durumu (depolardan biri doluyken)
$M_{10} = [1\ 0\ 1\ 0\ 1\ 2\ 0\ 1\ 0\ 0]^T$	q_{10} , MI tarafından üretilen mamülün (üçüncü palet ile birlikte) ara bölgedeki iki depodan boş olanına boşaltılması durumu
$M_{11} = [0\ 1\ 0\ 0\ 1\ 2\ 0\ 1\ 0\ 0]^T$	q_{11} , MI'e dördüncü paletin yüklendiği ve MI'in işlem yaptığı durumu
$M_{12} = [0\ 1\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 1]^T$	q_{12} , Robotun, birinci paletin kullanıldığı MII'ye hizmet verdiği durumu (depolar doluyken)

Çizelge 2.3. (Devam) Örnek 2.3'de verilen sistemin $R(G, m_0)$ ve Q kümelerinin elemanları

$R(G, m_0)$ kümesinin elemanları	Q kümesinin elemanları
$M_{13} = [1\ 1\ 0\ 0\ 1\ 2\ 0\ 0\ 1\ 0]^T$	q_{13} , Robotun MII'yi boşaltması ve birinci paletin çözümü
$M_{14} = [1\ 0\ 0\ 1\ 0\ 2\ 0\ 0\ 1\ 0]^T$	q_{14} , Robotun, birinci paletin kullanıldığı MI'e hizmet verdiği durumu (depolar doluyken)
$M_{15} = [2\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1]^T$	q_{15} , Robotun, ikinci paletin kullanıldığı MII'ye hizmet verdiği durumu (depolardan biri doluyken)
$M_{16} = [3\ 0\ 1\ 0\ 0\ 0\ 2\ 0\ 0\ 1]^T$	q_{16} , Robotun, birinci paletin kullanıldığı MII'ye hizmet verdiği durumu (depolar boşken)
$M_{17} = [1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1]^T$	q_{17} , Robotun, üçüncü paletin kullanıldığı MII'ye hizmet verdiği durumu (depolardan biri doluyken)
$M_{18} = [2\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0]^T$	q_{18} , Robotun MII'yi boşaltması ve ikinci paletin çözümü
$M_{19} = [2\ 1\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 1]^T$	q_{19} , Robotun, ikinci paletin kullanıldığı MII'ye hizmet verdiği durumu (depolar boşken)
$M_{20} = [2\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0]^T$	q_{20} , Robotun, ikinci paletin kullanıldığı MI'e hizmet verdiği durumu (depolardan biri doluyken)
$M_{21} = [2\ 0\ 1\ 0\ 1\ 2\ 0\ 0\ 1\ 0]^T$	q_{21} , MI tarafından üretilen mamülün (ikinci palet ile birlikte) ara bölgedeki iki depodan boş olanına boşaltılması durumu
$M_{22} = [1\ 0\ 1\ 0\ 2\ 0\ 0\ 0\ 0\ 1]^T$	q_{22} , Robotun, üçüncü paletin kullanıldığı MII'ye hizmet verdiği durumu (depolar doluyken)
$M_{23} = [0\ 0\ 0\ 1\ 0\ 2\ 0\ 1\ 0\ 0]^T$	q_{23} , Robotun, dördüncü paletin kullanıldığı MI'e hizmet verdiği durumu (depolar doluyken)



Şekil 2.7. Şekil 2.1'de Petri ağı ile gösterilen sistemin otomata modeli

Çizelge 2.4. Şekil 2.1'de Petri ağı ile gösterilen sistemin $R(G, m_0)$ ve Q kümelerinin elemanları

$R(G, m_0)$ kümesinin elemanları	Q kümesinin elemanları
$M_0 = [1 \ 1 \ 0 \ 0 \ 1]^T$	q_0
$M_1 = [0 \ 1 \ 1 \ 0 \ 0]^T$	q_1
$M_2 = [1 \ 0 \ 0 \ 1 \ 0]^T$	q_2

Şekil 2.1'de Petri ağı modeli verilen sistemin otomata gösterimi Şekil 2.7'de verilmiştir. Otomata gösteriminde; q_0 , başlangıç durumunu, $Q = \{q_0, q_1, q_2\}$ durumlar kümesini (bkz. Çizelge 2.4) ve $\Sigma = \{t_1, t_2, t_3, t_4\}$ olaylar kümesini göstermektedir. Otomata gösteriminin çalışması da;

$$\gamma(t_1, q_0) = q_1 \quad \gamma(t_2, q_0) = q_2$$

$$\gamma(t_3, q_1) = q_0 \quad \gamma(t_4, q_2) = q_0$$

şeklinde olmaktadır. Birden fazla olaydan meydana gelen olay dizileri için yukarıdaki tanımlardan faydalanılarak,

$$\gamma(t_{i_1} t_{i_2} \dots t_{i_k}, q) = \gamma(t_{i_1}, \gamma(t_{i_2}, \dots, \gamma(t_{i_k}, q) \dots))$$

olarak tanımlanır ($t_{i_1}, t_{i_2}, \dots, t_{i_k} \in \Sigma$). Bu sistem için dil, $L = \{\epsilon, t_1, t_1 t_3, t_1 t_3 t_1, \dots, t_2, t_2 t_4, t_2 t_4 t_2, \dots\}$ olarak ifade edilmektedir.

Şekil 2.7'de verilen otomata ve dil gösterimleri, Şekil 2.1'de verilen Petri ağı gösteriminden daha basit gösterimlerdir. Bu sistemin modellemesinde Petri ağı yerine, otomata ve dil gösterimleri tercih edilebilir.

2.2.2. Vektörel gösterim ile Petri ağı gösterimi arasındaki ilişki

İftar ve Özgüner tarafından yapılan çalışmada [28], kesikli durum sistemleri vektörel olarak; \mathcal{X} , durumların kümesi, \mathcal{X}_u , gerçekleşmesi mümkün olan durumların kümesi ($\mathcal{X}_u \subset \mathcal{X}$), \mathcal{U} olayların kümesi, X_0 , başlangıç durumu ve F , *sistem fonksiyonu* olmak üzere $X^+ = F(X, U)$ şeklinde ifade edilmiştir ($X, X^+ \in \mathcal{X}, U \in \mathcal{U}$). Sistem fonksiyonu olarak adlandırılan F fonksiyonu, tanım kümesinden dolayı, gerçekte bir kısmi fonksiyondur. $X^+ = F(X, U)$ ifadesinde, X^+ , X durumundan sonra U olayının olmasıyla elde edilen bir sonraki durumu göstermektedir. \mathcal{X} kümesinin elemanı olan her bir durum vektörü, sistemdeki bir durumu göstermektedir. U olayları kullanılarak, bir durumdan başka bir duruma geçilmesi sağlanmış olur. Bu da, F fonksiyonuyla olmaktadır.

Örnek 2.3'de verilen sistemin, vektörel gösterimdeki durumları aşağıda verilmiştir: Sistemin durum vektörü elemanları cinsinden, $X = [P \ MI \ R \ B \ MII]^T$ şeklinde tanımlanabilir. Burada, $P \in \{0, 1, 2, 3, 4\}$; sistemde kullanılan palet sayısını, $MI \in \{0, 1\}$; 1. makinenin çalışıp, çalışmadığını ("0", MI'in çalışmadığını, "1", MI'in çalıştığını göstermektedir), $R \in \{0, 1, 2\}$; robotun MI veya MII'ye hizmet vermesini ya da kullanılmamasını ("0", robotun kullanılmadığını, "1", robotun MI'e hizmet verdiğini ve "2", robotun MII'ye hizmet verdiğini göstermektedir), $B \in \{0, 1, 2\}$; ara bölgede kullanılan depo sayısını, $MII \in \{0, 1\}$; 2. makinenin çalışıp, çalışmadığını ("0", MII'nin çalışmadığını, "1", MII'nin çalıştığını göstermektedir) gösteren değişkenlerdir. Bu sistemin başlangıç durumu $X_0 = [0 \ 0 \ 0 \ 0 \ 0]^T$ ile gösterilir.

Vektörel gösterim ile Petri ağı modellemesi arasındaki ilişki aşağıda verilmiştir.

- Petri ağında, bir olayın ya da bir işlemin başlatılması ve bitirilmesi geçişler kullanılarak yapılmaktadır. Bu nedenle, \mathcal{U} kümesi, T kümesi ile ilişkilendirilir. Örnek 2.3'de verilen sistem için, $\mathcal{U} = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ olarak bulunur.
- Petri ağında ulaşılabilirlik kümesi, başlangıç işaretlemesinden başlayarak

elde edilen tüm işaretleme vektörlerinin oluşturduğu kümedir (bkz. Bölüm 2.1.2). Ulaşılabilirlik kümesinin elemanı olan her bir işaretleme vektörü, kesikli durum sistemlerdeki bir duruma karşılık gelmektedir. Dolayısıyla, \mathcal{X}_u kümesi, $R(G, m_0)$ kümesi ile ilişkilendirilir. Örnek 2.3'de verilen sistem için, $\mathcal{X}_u = \{X_0, X_1, X_2, \dots, X_{23}\}$ olarak bulunur (\mathcal{X}_u kümesinin elemanları Çizelge 2.5'de verilmiştir).

- Ulaşılabilirlik kümesi ile \mathcal{X}_u kümesi ilişkilendirildiğinden, m_0 da X_0 ile ilişkilendirilir.
- Petri ağında bir işaretleme vektöründen, başka bir işaretleme vektörü; T kümesinin bir elemanı olan ve ele alınan işaretleme vektöründen sonra ateşlenebilecek bir geçişin, ρ iletim fonksiyonunda kullanılması ile elde edilir. Bu nedenle ρ fonksiyonu, F fonksiyonu ile ilişkilendirilir. Örnek 2.3'de verilen sistem için, F fonksiyonu Çizelge 2.6'da tanımlanmıştır.

Çizelge 2.5. Örnek 2.3'de verilen sistemin \mathcal{X}_u kümesinin elemanları

$X_0 = [0\ 0\ 0\ 0\ 0]^T$	$X_1 = [1\ 1\ 0\ 0\ 0]^T$
$X_2 = [1\ 0\ 1\ 0\ 0]^T$	$X_3 = [1\ 0\ 0\ 1\ 0]^T$
$X_4 = [1\ 0\ 0\ 0\ 1]^T$	$X_5 = [2\ 1\ 0\ 0\ 1]^T$
$X_6 = [2\ 0\ 1\ 0\ 1]^T$	$X_7 = [2\ 0\ 0\ 1\ 1]^T$
$X_8 = [3\ 1\ 0\ 1\ 1]^T$	$X_9 = [3\ 0\ 1\ 1\ 1]^T$
$X_{10} = [3\ 0\ 0\ 2\ 1]^T$	$X_{11} = [4\ 1\ 0\ 2\ 1]^T$
$X_{12} = [4\ 1\ 2\ 2\ 0]^T$	$X_{13} = [3\ 1\ 0\ 2\ 0]^T$
$X_{14} = [3\ 0\ 1\ 2\ 0]^T$	$X_{15} = [2\ 0\ 2\ 1\ 0]^T$
$X_{16} = [1\ 0\ 2\ 0\ 0]^T$	$X_{17} = [3\ 1\ 2\ 1\ 0]^T$
$X_{18} = [2\ 1\ 0\ 1\ 0]^T$	$X_{19} = [2\ 1\ 2\ 0\ 0]^T$
$X_{20} = [2\ 0\ 1\ 1\ 0]^T$	$X_{21} = [2\ 0\ 0\ 2\ 0]^T$
$X_{22} = [3\ 0\ 2\ 2\ 0]^T$	$X_{23} = [4\ 0\ 1\ 2\ 1]^T$

Çizelge 2.6. Örnek 2.3'de verilen sistemin vektörel gösterimdeki çalışması

X	U	$F(X, U)$
X_0	t_1	X_1
X_1	t_2	X_2
X_2	t_3	X_3
X_3	t_4	X_4
	t_1	X_{18}
X_4	t_1	X_5
	t_5	X_{16}
X_5	t_2	X_6
	t_5	X_{19}
X_6	t_3	X_7
X_7	t_1	X_8
	t_5	X_{15}
X_8	t_2	X_9
	t_5	X_{17}
X_9	t_3	X_{10}
X_{10}	t_1	X_{11}
	t_5	X_{22}

X	U	$F(X, U)$
X_{11}	t_2	X_{23}
	t_5	X_{12}
X_{12}	t_6	X_{13}
X_{13}	t_2	X_{14}
X_{14}	t_4	X_9
X_{15}	t_1	X_{17}
	t_6	X_3
X_{16}	t_6	X_0
X_{17}	t_6	X_{18}
X_{18}	t_4	X_5
	t_2	X_{20}
X_{19}	t_6	X_1
X_{20}	t_3	X_{21}
X_{21}	t_1	X_{13}
	t_4	X_7
X_{22}	t_1	X_{12}
	t_6	X_{21}

3. KONTROLÖR TASARIMI

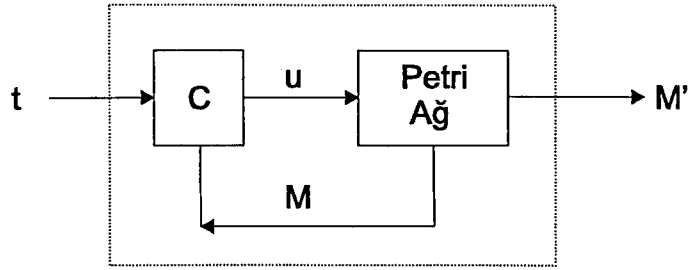
Bu bölümde, Petri ağları, otomata, dil ve vektörel gösterimlerinin her biri için sistem çıkmazının meydana gelmesini önleyen birer kontrolör tasarım yöntemi verilecektir. Ayrıca, otomata, dil ve vektörel gösterimler için elde edilen kontrol yaklaşımları ile Petri ağları için elde edilen kontrol yaklaşımı arasındaki ilişkilerin incelenmesi de bu bölümde yapılacaktır.

3.1. Petri Ağında Kontrolör Tasarımı

Sistem çıkmazının meydana gelmesini önlemek için; ya [2]'de belirtildiği gibi, Petri ağının başlangıç işaretlemesi değiştirilerek sistem çıkmazı önlenmeye çalışılabilir ya da bir kontrolör tasarlanarak sistem çıkmazının meydana gelmesi engellenebilir. [2]'de bahsedilen başlangıç işaretlemesinin değişimi, eğer mümkün ise, kesikli olay sistemdeki kaynak (makine, personel, vb.) veya kaynakların sayısının arttırılıp azaltılmasıdır. Bu da sistemin fiziksel olarak değiştirilmesi demektir; dolayısıyla, bu yöntemi gerçekleştirmek her zaman mümkün olmayabilir. Bu durumda, Petri ağında sistem çıkmazının meydana gelmesini önlemek için kontrolör tasarımı önem kazanmaktadır.

$G(P, T, N, O, m_0)$ ile gösterilen bir Petri ağında sistem çıkmazını önleyecek bir kontrolör tasarlayabilmek için $m_0 \notin \mathcal{Y}(G)$ olmalıdır. Burada $\mathcal{Y}(G)$, kendisinden sonra mümkün olan tüm ateşleme dizileri kullanıldığında, sistem çıkmazı meydana gelen, işaretleme vektörlerinin oluşturduğu kümeyi göstermektedir. $m_0 \in \mathcal{Y}(G)$ ise, Petri aği için bir kontrolör tasarlamak mümkün değildir. Bu durumda, $\mathcal{Y}(G)$ kümesinin elemanı olmayacak bir başlangıç işaretlemesi değişikliği mümkün ise, başlangıç işaretlemesi değiştirilerek kontrolör tasarlanabilir.

Petri ağları için, Sreenivas [17] tarafından yapılan çalışmada kullanılan kontrol tekniği incelenmiş ve bu yaklaşım temel alınarak bir kontrol yaklaşımı geliştirilmiştir. Sreenivas [17] tarafından kontrol edilen Petri ağları için, ortaya konulan kontrol tasarımı, kontrol edilen Petri ağının başlangıç işaretlemesinden başlayarak ulaşılan her işaretleme vektöründen sonra hangi geçişlerin ateşlenip ateşlenemeyeceği şeklinde gerçekleştirilmiştir. Burada, ele alınan



Şekil 3.1. Kapalı döngü Petri ağı

geçiş, ateşlenmesine izin veriliyorsa “1”, izin verilmiyorsa “0” olarak gösterilmiştir.

Eğer M işaretleme vektöründen ya bir $t \in T$ geçişi ya da bu geçişle başlayan tüm ateşleme dizileri kullanılarak elde edilen işaretleme vektörlerinden sonra, Petri ağında hiç bir geçiş ateşlenemiyorsa (sistem çıkmazı meydana geliyorsa), bu M işaretleme vektörüne sistem çıkmazına yol açabilecek işaretleme vektörü denir. Bu tür işaretleme vektörlerinden oluşan küme, \mathcal{D} ile gösterilmektedir.

$M \in \mathcal{D}$ olan işaretleme vektöründen sonra, ateşlenmesiyle sistem çıkmazı meydana gelmesine neden olan geçiş ya da geçişlere izin vermeyip, diğer geçişlere izin veren bir kontrolör; C , kontrol matrisini göstermek üzere, $C(M) \triangleq$ köşegen $\{c(M, t)\}$ olarak tanımlanabilir ($C : R(G, m_0) \rightarrow \mu^{|T| \times |T|}$). Burada,

$$c(M, t) = \begin{cases} 0, & M \in \mathcal{D}, t \text{ ateşlenmesine izin verilmiyorsa} \\ 1, & \text{diğer durumlarda} \end{cases}, \quad t \in T$$

olarak tanımlanmaktadır. Böylece, her M vektörüne karşılık bir C matrisi elde edilmiş olur.

$G_c(P, T, N, O, m_0, C)$; sistem çıkmazını önleyen kontrolörle birlikte işaretlenmiş bir Petri ağı göstermek üzere, kontrolörle birlikte Petri ağı denklemi;

$$M' = M + AC(M)U_t, \quad t \in T \quad (3.1)$$

şeklinde yeniden düzenlenebilir.

Şekil 3.1’de gösterilen kapalı döngü sistemde, t , ateşlenmek istenen geçişi

($t \in T$), $u \triangleq C(M)U_t$ ve M' , M işaretleme vektöründen sonra gelen işaretleme vektörünü göstermektedir. Tasarlanan kontrolörün çalışma şekli aşağıda verilmiştir.

- $M \notin \mathcal{D}$ ise, $C(M) = I$ olacağından, $M' = \rho(t, M)$ ile gösterilen yeni bir işaretleme vektörüne ulaşılır.
- $M \in \mathcal{D}$ ise,
 - * $c(M, t) = 1$ ise, kontrolör t geçişinin ateşlenmesine izin verdiği için, $M' = \rho(t, M)$ ile gösterilen yeni bir işaretleme vektörüne ulaşılır.
 - * $c(M, t) = 0$ ise, kontrolör t geçişinin ateşlenmesine izin vermediğinden, Petri ağında bir değişiklik meydana gelmeyecektir, $M' = M$.

Not 3.1: Yukarıda verilen kontrolörün çalışma şeklinden, $R(G_c, m_0) \subset R(G, m_0)$ olduğu bulunmaktadır.

Şekil 2.3'deki Petri ağında, elde edilen aşağıdaki işaretleme vektörlerinden sonra hiç bir geçiş ateşlenmemektedir. Bu nedenle, Tanım 2.4'e göre Şekil 2.3'de verilen Petri ağında sistem çıkmazı meydana gelmektedir.

- $\hat{M}_{ij}^1 = [0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ i \ 0 \ j \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$
- $\hat{M}_{ij}^2 = [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ i \ 0 \ j \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0]^T$
- $\hat{M}^3 = [0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 4 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$
- $\hat{M}^4 = [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 4 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1]^T$

Burada, M_{ij} işaretleme vektöründe, i ; p_8 'deki belirtilerin sayısını ve j ; d 'deki belirtilerin sayısını göstermektedir ($(i, j) \in \{(0, 4), (1, 3), (2, 2), (3, 1), (4, 0)\}$).

Şekil 2.3'de verilen Petri ağında, meydana gelen bu sistem çıkmazını önlemek için ise, yukarıda verilen kontrolör tasarımı kullanılarak, bu sistem için bir kontrolör tasarlanabilir:

Petri ağında sistem çıkmazına yol açabilecek işaretleme vektörleri aşağıdaki şekilde elde edilmiştir.

- $M_{xy}^1 = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ x \ 1 \ y \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$, $(x, y) \in \{(0, 4), (1, 3), (2, 2), (3, 1)\}$. Bu işaretleme vektöründen sonra t_2 geçişinin ateşlenmesi ile sistem çıkmazı meydana gelmektedir, $\hat{M}_{xy}^1 = \rho(t_2, M_{xy}^1)$.

- $M_{xy}^2 = [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ x \ 1 \ y \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T$ ($x, y \in \{(0, 4), (1, 3), (2, 2), (3, 1)\}$). Bu işaretleme vektöründen sonra t_8 geçişinin ateşlenmesi ile sistem çıkmazı meydana gelmektedir, $\hat{M}_{xy}^2 = \rho(t_8, M_{xy}^2)$.
- $M^3 = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$. Bu işaretleme vektöründen sonra t_2 veya t_4 geçişlerinden birinin ateşlenmesi ile sistem çıkmazı meydana gelmektedir, $\hat{M}_{40}^1 = \rho(t_2, M^3)$ ve $\hat{M}^3 = \rho(t_4, M^3)$.
- $M^4 = [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 4 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T$ Bu işaretleme vektöründen sonra t_8 veya t_{10} geçişlerinden birinin ateşlenmesi ile sistem çıkmazı meydana gelmektedir, $\hat{M}_{40}^2 = \rho(t_8, M^4)$ ve $\hat{M}^4 = \rho(t_{10}, M^4)$.
- $M^5 = [1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$. Bu işaretleme vektöründen sonra t_4 geçişi ateşlenirse, bir sonraki ateşlenebilecek geçiş yalnızca t_1 geçisidir ($\rho(t_4, M^5) = [1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 4 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$). Bu da sistem çıkmazının meydana gelmesine neden olur, $\hat{M}^3 = \rho(g_1, M^5)$ ($g_1 = t_4, t_1$).
- $M^6 = [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 4 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]^T$. Bu işaretleme vektöründen sonra t_{10} geçişi ateşlenirse, bir sonraki ateşlenebilecek geçiş yalnızca t_7 geçisidir ($\rho(t_{10}, M^6) = [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 4 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1]^T$). Bu da sistem çıkmazının meydana gelmesine neden olur, $\hat{M}^4 = \rho(g_2, M^6)$ ($g_2 = t_{10}, t_7$).

Dolayısıyla, $\mathcal{D} \triangleq \{M_{xy}^1, M_{xy}^2, M^3, M^4, M^5, M^6\}$ olarak bulunmaktadır. Bu bölümde verilen kontrol algoritmasına göre; M_{xy}^1 'den sonra t_2 geçişinin ateşlenmesine, M_{xy}^2 'den sonra t_8 geçişinin ateşlenmesine, M^3 'den sonra t_2 ve t_4 geçişlerinin ateşlenmesine, M^4 'den sonra t_8 ve t_{10} geçişlerinin ateşlenmesine, M^5 'den sonra t_4 geçişinin ateşlenmesine, M^6 'dan sonra t_{10} geçişinin ateşlenmesine izin verilmeyecek şekilde kontrolör matrisleri oluşturulur. Kontrolör tasarımında, Şekil 2.3'deki Petri ağının $M_{xy}^1, M_{xy}^2, M_{xy}^3, M^4, M^5$ ve M^6 işaretleme vektörleri hariç, diğer işaretleme vektörleri için kontrolör matrisi birim matris olarak alınmaktadır ($C = I$).

$$C(M_{xy}^1) = \text{köşegen}\{1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$$

$$C(M_{xy}^2) = köşegen\{1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1\}$$

$$C(M^3) = köşegen\{1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1\}$$

$$C(M^4) = köşegen\{1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1\}$$

$$C(M^5) = köşegen\{1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1\}$$

$$C(M^6) = köşegen\{1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1\}$$

Burada, $(x, y) \in \{(0, 4), (1, 3), (2, 2), (3, 1)\}$.

Örneğin, M_{13}^1 , işaretleme vektöründen sonra, t_2 ve t_6 geçişlerinin ateşlenmesini inceleyelim.

$$\dot{M} = M_{13}^1 + (O - N)C(M_{13}^1)U_{t_2}$$

$\dot{M} = M_{13}^1$ olarak elde edilmiştir. Burada kontrolör t_2 geçişinin ateşlenmesine izin vermemiştir. Dolayısıyla, Petri ağında sistem çıkmazı meydana gelmesi önlenmiştir.

Aşağıda gösterilen ifade, M_{13}^1 işaretleme vektöründen sonra, kontrolörün t_6 geçişinin ateşlenmesine izin vermesiyle, Petri ağında farklı bir işaretleme vektörüne ulaşıldığını göstermektedir.

$$\ddot{M} = M_{13}^1 + (O - N)C(M_{13}^1)U_{t_6} = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 4 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$$

3.2. Kesikli Olay Sistemler İçin Kullanılan Bazı Kontrol Yaklaşımları Arasındaki İlişkiler

Tezin giriş bölümünde belirtildiği gibi, kesikli olay sistemlerinde kontrol yaklaşımlarının ortaya konduğu ve incelendiği çeşitli çalışmalar yapılmıştır. 1990 yılına kadar bu konuda yapılmış olan çalışmalar, Hayman [30] tarafından incelenmiştir. Ancak o tarihten, günümüze kadar çeşitli yaklaşımların ortaya konulduğu çalışmalar yapılmıştır [2,7,10–18,28].

Burada, dil ve otomata gösterimleri ile vektörel gösterimde sistem çıkmazını önleyen birer kontrol yaklaşımı ve bu kontrol yaklaşımları ile Petri ağları için elde edilen kontrol yaklaşımı arasındaki ilişkiler verilecektir.

3.2.1. Dil gösterimi için bir kontrol yaklaşımı

[9]'da yapılan çalışmada (bu çalışmada kullanılan notasyon Bölüm 2.2.1 de verilmiştir), kesikli olay sistemin otomata modelinde olayların kümesi, denetlenebilen ve denetlenemeyen olarak ikiye ayrılarak incelenmiştir.

[9]'da olması istenilen durumlardan bir küme oluşturulmuştur. Bu küme, Q_m ile gösterilmektedir, $Q_m \subset Q$. Yalnızca Q_m durum kümesinin elemanları meydana gelecek şekilde, $L_m = \{s \in L \mid \gamma(s, q_0) \in Q_m\}$ bir dil seçilmektedir ($L \subset \Sigma^*$, $q_0 \in Q$, $L_m \subset L$).

Örnek 2.3'ün otomata gösteriminde, sistem q_{23} ile gösterilen duruma geldikten sonra başka bir duruma geçememektedir. $Q_m = Q \setminus \{q_{23}\}$ şeklinde alınarak, öncelikle q_{23} durumuna ulaşılmasını sağlayacak bir dil elde edilebilir.

$$\begin{aligned} L_{q_{23}} &\triangleq \{u \in L \mid \gamma(u, q_0) = q_{23}\} \\ &= \{t_1 t_2 t_3 t_4 t_1 t_2 t_3 t_1 t_2 t_3 t_1 t_2, t_1 t_2 t_3 t_1 t_2 t_3 t_4 t_1 t_2 t_3 t_1 t_2, \dots\} \end{aligned}$$

Bu dilden yararlanılarak da L_m ile gösterilen dil,

$$L_m = L \setminus L_{q_{23}} = \{\epsilon, t_1, t_1 t_2, t_1 t_2 t_3, \dots\}$$

olarak tanımlanabilir. Sonuç olarak, L_m olarak gösterilen dil kullanılarak, q_{23} durumuna ulaşılması önlenmiş olur.

Bu tezde yapılan kontrol tasarımında, sistem çıkmazının meydana geldiğini gösteren işaretleme vektörlerinin, ulaşılabilirlik kümesinden çıkartılmasıyla elde edilen yeni küme, $R(G_c, m_0)$, Q_m ile ilişkilendirilir. $R(G_c, m_0)$ kümesinin elemanı olan işaretleme vektörlerinin elde edilmesini sağlayan tüm ateşleme dizilerinin oluşturduğu küme, L_m ile ilişkilendirilir.

3.2.2. Otomata gösterimi için bir kontrol yaklaşımı

Burada, otomata gösteriminde sistem çıkmazının meydana gelmesini önleyen bir kontrolör tasarımı verilecektir. Burada kullanılan notasyon Bölüm 2.2.1 de verilmiştir. Otomata gösteriminde, sistem çıkmazının tanımı aşağıda verilmiştir.

Tanım 3.1: Bir otomatada, $q \in Q$ durumundan sonra hiç bir olay meydana gelmiyorsa veya yapılabilecek tüm olay dizileri kullanılarak elde edilen durumlardan sonra hiç bir olay meydana gelmiyorsa, q durumuna, *sistem çıkmazı gösteren durum* denir. Bu durumların oluşturduğu küme Q^d ile gösterilir.

Not 3.2: $q \in Q^d$ durumunun olması, otomatada sistem çıkmazının meydana geldiğini gösterir.

$Q^b \triangleq \{q \mid \exists t \in \Sigma, \text{ öyle ki } \gamma(t, q) \in Q^d\}$, sistem çıkmazı gösteren durumlara ulaşılmasını sağlayan durumların oluşturduğu küme olarak tanımlanmaktadır. $D(q)$, $q \in Q^b$ durumundan sonra, sistem çıkmazı gösteren durumlara ulaşılmasını sağlayan olaylarının oluşturduğu kümeyi göstermektedir. $q \notin Q^b$ ise, $D(q) = \emptyset$ olarak alınmaktadır.

$q \in Q$ durumundan sonra, olmasıyla sistem çıkmazı gösteren durumlara ulaşılmasına neden olan olayların olmasına izin vermeyip, diğer olayların olmasına izin veren bir kontrolör;

$$C(t, q) = \begin{cases} 0, & t \in D(q) \\ 1, & \text{aksi durumda} \end{cases}, q \in Q, t \in \Sigma$$

şeklinde tanımlanmaktadır. $C(t, q) = 0$ ise, kontrolör q durumundan sonra t olayının olmasına izin vermemektedir. $C(t, q) = 1$ ise, kontrolör q durumundan sonra t olayının olmasına izin vermektedir.

Burada tasarlanan kontrolör, Bölüm 3.1'de verilen Petri ağları için tasarlanan kontrolöre benzemektedir. Otomata gösterimi (Petri ağları) için tasarlanan kontrolörde ele alınan bir durumdan (işaretleme vektöründen) sonra bir olayın olmasına (geçişin ateşlenmesine) izin verilip verilmeyeceği belirlenmektedir.

Örnek 2.3'de verilen sistemin otomata gösteriminde, q_{23} sistem çıkmazı gösteren tek durumdur. Dolayısıyla, $Q^d = \{q_{23}\}$ olmaktadır. Buradan, $Q^b = \{q_{11}\}$ ve $D(q_{11}) = \{t_2\}$ elde edilmektedir. Dolayısıyla, $C(t_2, q_{11}) = 0$ ve $C(t_5, q_{11}) = 1$ olmaktadır. Kontrolör, Q kümesinin geri kalan elemanlarına bağlı olan olayların olmasına izin verecek şekilde seçilmiştir ($C(t, q) = 1, \forall q \neq q_{11}, \forall t$).

Şekil 2.7'de gösterilen otomatada sistem çıkmazı meydana gelmemektedir. Dolayısıyla, $Q^d = \emptyset$, $D(q) = \emptyset$ ve $C(t, q) = 1$, $\forall q, \forall t$ olmaktadır.

3.2.3. Vektörel gösterim için bir kontrol yaklaşımı

İftar ve Özgüner tarafından yapılan çalışmada [28], vektörel gösterim kullanılarak modellenen kesikli durum sistemlerinde, sistem çıkmazını önlemek amacıyla bir kontrolör tasarım yöntemi verilmemiştir. Ancak hibrid sistemler için verilen kontrolör tasarımından yararlanılarak; vektörel gösterimde girdiler kullanılarak, sistemde meydana gelmesi istenilmeyen durumlara ulaşılması engellenebilir.

Bu yöntem geliştirilerek, vektörel gösterimde, $U = C(X)$, $X \in \mathcal{X}_u$ şeklinde bir kontrolör tasarlanabilir. $C(X)$, X durumundan sonra izin verilen girdileri göstermektedir. Tasarlanan bu kontrolör kullanıldığında, meydana gelmesine izin verilen durumların kümesi, \mathcal{X}_{cu} ile gösterilir ($\mathcal{X}_{cu} \subset \mathcal{X}_u$).

Örnek 2.3'de verilen sistemin vektörel gösteriminde, X_{23} durumundan sonra, herhangi bir girdi kullanılarak başka bir duruma ulaşılması mümkün olmamaktadır. Bu nedenle, sistemin X_{23} durumuna ulaşmasını engellemek için, X_{11} durumundan sonra t_2 girdisi kullanılması izin verilmemelidir. X_{11} durumundan sonra, t_2 girdisi yerine t_5 girdisi kullanıldığında sistem farklı bir duruma ulaşır, $F(X_{11}, t_5) = X_{12}$. Çizelge 2.6'dan yararlanılarak \mathcal{X}_u kümesinin bütün elemanları için birer kontrolör tasarlanabilir. Örneğin, X_{11} için; $C(X_{11}) = t_5$ ve X_7 için; $C(X_7) = t_1$ veya $C(X_7) = t_5$ olarak alınmaktadır.

Kontrolör tasarlanmış Petri ağının ulaşılabilirlik kümesi, $R(G_c, m_0)$, \mathcal{X}_{cu} ile ilişkilendirilir. Vektörel gösterimde sistemde meydana gelmesi istenilmeyen durumlar, Petri ağında sistem çıkmazının meydana geldiğini gösteren işaretleme vektörleri ile ilişkilendirilir.

4. İÇERME PRENSİBİ

Bu bölümde Ikeda ve Šiljak [19] tarafından sürekli durum sistemleri için ortaya konan içerme prensibinin, kesikli olay sistemlerin modellenmesinde kullanılan Petri ağlarında ve otomata ile dil gösterimlerinde nasıl tanımlanabileceği verilecektir.

4.1. Petri Ağlarında İçerme Prensibi

Burada, $\tilde{G}(\tilde{P}, \tilde{T}, \tilde{N}, \tilde{O}, \tilde{m}_0)$ ile gösterilen bir Petri ağı ve $G(P, T, N, O, m_0)$ ile gösterilen bir Petri ağı arasında, içerme prensibinin tanımı ve özellikleri verilecektir. G ile gösterilen Petri ağında, ρ ; iletim fonksiyonunu, A ; bağlantı matrisini, U_g ; ağırlık vektörünü göstermektedir. Benzer şekilde, \tilde{G} ile gösterilen Petri ağında, $\tilde{\rho}$; iletim fonksiyonunu, \tilde{A} ; bağlantı matrisini, \tilde{U}_g ; ağırlık vektörünü göstermektedir.

Tanım 4.1: Aşağıdaki şartları sağlayan Q ve V matrisleri varsa, \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerir.

- (i) $QV = I$,
- (ii) $\tilde{m}_0 = Vm_0$,
- (iii) herhangi bir $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ için, eğer $\tilde{M}' \in R(\tilde{G}, \tilde{M})$ ise $Q\tilde{M}' \in R(G, Q\tilde{M})$ 'dir.
- (iv) herhangi bir $M \in R(G, m_0)$ için, $M = Q\tilde{M}$ olan $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ vardır.

Burada I , birim matrisi göstermektedir.

Bu tanım kullanılarak, içerme prensibinin sağlanması için gereken şartlar, bir teorem ile aşağıdaki şekilde verilmiştir.

Teorem 4.1: Eğer ve yalnızca eğer,

$$QV = I \tag{4.1a}$$

$$\tilde{m}_0 = Vm_0 \tag{4.1b}$$

ve

$$Q\tilde{A}\tilde{U}_g = AU_g \tag{4.2}$$

şartlarını sağlayacak şekilde V ve Q matrisleri ile m_0 başlangıç işaretlemesinden başlayan herhangi bir g ateşleme dizisine karşılık, \tilde{m}_0 başlangıç işaretlemesinden başlayan herhangi bir \tilde{g} ateşleme dizisi ve herhangi bir $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ işaretleme vektöründen başlayan \tilde{g} ateşleme dizisine karşılık, $Q\tilde{M}$ işaretleme vektöründen başlayan g ateşleme dizisi bulunabiliyorsa \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerir.

İspat: Denklem (4.1a) ve (4.1b) sırasıyla, Tanım 4.1'in (i). ve (ii). şartları ile aynı ifadeyi göstermektedir.

Yalnızca eğer kısmının ispatı:

$\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olmak üzere, \tilde{M} işaretleme vektöründen sonra \tilde{g} ateşleme dizisi kullanıldığında,

$$\tilde{M}' = \tilde{M} + \tilde{A}\tilde{U}_{\tilde{g}} \quad (4.3)$$

elde edilir. Tanım 4.1'in (iii). şikkından,

$$Q\tilde{M}' = Q\tilde{M} + AU_g \quad (4.4)$$

olacak şekilde $Q\tilde{M}'$ 'den başlayan g ateşleme dizisi vardır. Denklem (4.3)'ün her iki tarafı soldan Q matrisi ile çarpılırsa,

$$Q\tilde{M}' = Q\tilde{M} + Q\tilde{A}\tilde{U}_{\tilde{g}} \quad (4.5)$$

elde edilir. Denklem (4.4) ve (4.5)'den dolayı, denklem (4.2) sağlanır.

$M \in R(G, m_0)$ olmak üzere, m_0 başlangıç işaretlemesinden sonra g ateşleme dizisi kullanıldığında,

$$M = m_0 + AU_g \quad (4.6)$$

elde edilir. Tanım 4.1'in (iv). şikkından dolayı, $Q\tilde{M} = M$ olan ve

$$\tilde{M} = \tilde{m}_0 + \tilde{A}\tilde{U}_{\tilde{g}} \quad (4.7)$$

eşitliğini sağlayan \tilde{m}_0 'dan başlayan \tilde{g} ateşleme dizisi vardır. Bu denklemin her iki tarafı soldan Q matrisi ile çarpılırsa,

$$Q\tilde{M} = Q\tilde{m}_0 + Q\tilde{A}\tilde{U}_{\tilde{g}} = M \quad (4.8)$$

elde edilir. Burada Tanım 4.1'in (i). ve (ii). şıkları kullanılarak,

$$M = m_0 + Q\tilde{A}\tilde{U}_{\tilde{g}} \quad (4.9)$$

elde edilir. Denklem (4.6) ve (4.9)'dan dolayı, denklem (4.2) sağlanır.

Eğer kısmının ispatı:

Denklem (4.5)'de denklem (4.2) kullanılarak, denklem (4.4) elde edilir. Denklem (4.4) de, $Q\tilde{M}$ işaretleme vektöründen sonra g ateşleme dizisi kullanılarak $Q\tilde{M}' \in R(G, Q\tilde{M})$ olan bir işaretleme vektörüne ulaşıldığını göstermektedir. Dolayısıyla, Tanım 4.1'in (iii). şikkı sağlanmış olur.

Denklem (4.6)'da, denklem (4.1a), (4.1b) ve (4.2) kullanılarak, denklem (4.7) yi sağlayan $M = Q\tilde{M}$ elde edilir. Denklem (4.7) de, $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olan bir işaretleme vektörüne ulaşıldığını göstermektedir. Dolayısıyla, Tanım 4.1'in (iv). şikkı sağlanmış olur. \square

4.2. Biri Diğerini İçeren İki Petri Ağı Arasındaki İlişkiler

Bu bölümde, \tilde{G} ile gösterilen Petri ağı ile G ile gösterilen, \tilde{G} 'nin içerdiği, Petri ağı arasında sınırlılık, tersine dönüşebilirlik, canlılık ve sistem çıkmazı ilişkileri incelenecektir. Sistem çıkmazı ilişkisinin incelenmesi sırasında, \tilde{G} ile gösterilen Petri ağında daha farklı bir özellik olması gerektiğinden, Petri ağlarında, *çalışma engeli* olarak adlandırılan yeni bir özelliğin tanımı da verilecektir.

Teorem 4.2: Eğer \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içeriyorsa ve \tilde{G} ile gösterilen Petri ağı sınırlı ise, G ile gösterilen Petri ağı da sınırlıdır.

İspat: \tilde{G} ile gösterilen Petri ağı sınırlı ise, $\lambda(\tilde{M}) \leq \tilde{\kappa}$, $\forall \tilde{M} \in R(\tilde{G}, \tilde{m}_0)$, $\tilde{\kappa} \in \mathcal{N}$ olmaktadır. Başka bir deyişle, bütün $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ için $\|\tilde{M}\|$ sınırlıdır ($\|\cdot\|$, (\cdot) 'nin ∞ -normunu göstermektedir). \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerdiğinden, herhangi bir $M \in R(G, m_0)$ için, $M = Q\tilde{M}$ şartını sağlayan $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ vardır. Dolayısıyla, herhangi bir $M \in R(G, m_0)$ için, $\|M\| = \|Q\tilde{M}\| \leq \|Q\| \|\tilde{M}\|$ olur. Q matrisi, sabit bir matris olduğundan normu sınırlıdır. Buradan, G ile gösterilen Petri ağının sınırlı olduğu bulunur. \square

Teorem 4.3: Eğer \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içeriyorsa ve \tilde{G} ile gösterilen Petri ağı tersine dönüşebilir ise, G ile gösterilen Petri ağı da tersine dönüşebilir.

İspat: \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerdiğinden, herhangi bir $M \in R(G, m_0)$ için, $M = Q\tilde{M}$ şartını sağlayan $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ vardır. \tilde{G} ile gösterilen Petri ağı tersine dönüşebilir olduğundan; $\tilde{m}_0 \in R(\tilde{G}, \tilde{M})$, $\forall \tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olmaktadır. Bu ifade,

$$\tilde{m}_0 = \tilde{M} + \tilde{A}\tilde{U}_{\tilde{g}} \quad (4.10)$$

şartını sağlayacak \tilde{g} ateşleme dizisinin varlığını gösterir. Bu ifadenin her iki tarafı soldan Q matrisi ile çarpılırsa,

$$Q\tilde{m}_0 = Q\tilde{M} + Q\tilde{A}\tilde{U}_{\tilde{g}} \quad (4.11)$$

elde edilir. Burada (4.1b), (4.1a) ve (4.2) kullanılarak,

$$m_0 = M + AU_g \quad (4.12)$$

elde edilir. Bu denklemde, M işaretleme vektöründen sonra, g ateşleme dizisi

kullanılarak $m_0 \in R(G, M)$ başlangıç işaretlemesine ulaşıldığı bulunmaktadır. Dolayısıyla, G ile gösterilen Petri ağı tersine dönüşebilir. \square

Sınırlık ve tersine dönüşebilirlik gibi sistem çıkmazını da içerme prensibi kullanılarak incelemek için, çalışma engeli olarak adlandırılan yeni bir özellik tanımlanacaktır.

Tanım 4.2: $\bar{T} \triangleq \{\tilde{t} \in \tilde{T} \mid \exists t \in T \text{ öyle ki } Q\tilde{A}\tilde{U}_{\tilde{t}} = AU_t\}$ ve $\mathcal{T} \triangleq \tilde{T} \setminus \bar{T}$ olmak üzere, \tilde{M} ($\forall \tilde{M} \in R(\tilde{G}, \tilde{m}_0)$) işaretlemesinden sonra; en az bir $\tilde{t} \in \bar{T}$ geçişi ateşlenebiliyorsa veya en çok sonlu kere $\tilde{t} \in \mathcal{T}$ geçişi ateşlendikten sonra en az bir $\tilde{t} \in \bar{T}$ geçişi ateşlenebiliyorsa, \tilde{G} ile gösterilen Petri ağında *çalışma engeli* meydana gelmez.

Lemma 4.1: $\tilde{t} \in \mathcal{T}$ ise, $Q\tilde{A}\tilde{U}_{\tilde{t}} = 0$ olarak bulunur.

İspat:

$\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olmak üzere, \tilde{M} işaretleme vektöründen sonra $\tilde{t} \in \bar{T}$ ateşleme geçişi kullanılarak,

$$\tilde{M}' = \tilde{M} + \tilde{A}\tilde{U}_{\tilde{t}} \quad (4.13)$$

elde edilir. Bu denklemin her iki tarafı soldan Q matrisi ile çarpılırsa,

$$Q\tilde{M}' = Q\tilde{M} + Q\tilde{A}\tilde{U}_{\tilde{t}} \quad (4.14)$$

olarak bulunur. Bu denklemde kullanılan geçiş $\tilde{t} \in \mathcal{T}$ ise, Tanım 4.2'den dolayı, \tilde{t} geçişine karşılık $Q\tilde{A}\tilde{U}_{\tilde{t}} = AU_t$ eşitliğini sağlayan bir $t \in T$ geçişi yoktur. Bu nedenle, $Q\tilde{M}' = Q\tilde{M}$ olmaktadır. Denklem (4.14)'de, $Q\tilde{M}' = Q\tilde{M}$ olması, $Q\tilde{A}\tilde{U}_{\tilde{t}} = 0$ olmasını gerektirir. \square

Teorem 4.4: Eğer \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içeriyorsa ve \tilde{G} ile gösterilen Petri ağında çalışma engeli meydana gelmiyorsa, G ile gösterilen Petri ağında da sistem çıkmazı meydana gelmez.

İspat: \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerdiğinden, herhangi bir $M \in R(G, m_0)$ için, $M = Q\tilde{M}$ şartını sağlayan $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$

vardır. \tilde{G} ile gösterilen Petri ağında çalışma engeli meydana gelmiyorsa, $\tilde{M}' = \tilde{\rho}(\tilde{t}, \tilde{M}) \in R(\tilde{G}, \tilde{m}_0)$ olmasını sağlayan en az bir $\tilde{t} \in \bar{T}$ geçişi ya da $\tilde{M}' = \tilde{\rho}(\tilde{g}, \tilde{M}) \in R(\tilde{G}, \tilde{m}_0)$ olmasını sağlayan $\tilde{g} \triangleq \tilde{t}_{i_1}, \tilde{t}_{i_2}, \dots, \tilde{t}_{i_{n-1}}, \tilde{t}_{i_n}$, $(\tilde{t}_{i_1}, \tilde{t}_{i_2}, \dots, \tilde{t}_{i_{n-1}} \in \mathcal{T}, \tilde{t}_{i_n} \in \bar{T})$ ateşleme dizisi vardır.

- $\tilde{M}' = \tilde{\rho}(\tilde{t}, \tilde{M}) \in R(\tilde{G}, \tilde{m}_0)$ olmasını sağlayan en az bir $\tilde{t} \in \bar{T}$ geçişi olduğu durum:

$\tilde{M}' = \tilde{M} + \tilde{A}\tilde{U}_{\tilde{t}}$ denklemin her iki tarafı soldan Q matrisi ile çarpılırsa,

$$Q\tilde{M}' = Q\tilde{M} + Q\tilde{A}\tilde{U}_{\tilde{t}} \quad (4.15)$$

elde edilir. Burada, $\tilde{M}', \tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olduğundan ve \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerdiğinden, $M' = Q\tilde{M}'$, $M' \in R(G, m_0)$ olur. Böylece,

$$M' = M + Q\tilde{A}\tilde{U}_{\tilde{t}} \quad (4.16)$$

elde edilir. Denklem (4.16)'da $\tilde{t} \in \bar{T}$ ise, \bar{T} kümesinin tanımından dolayı, $Q\tilde{A}\tilde{U}_{\tilde{t}} = AU_t$ olduğu bir t geçişi vardır. Dolayısıyla, $M' = M + AU_t$ elde edilir.

- $\tilde{M}' = \tilde{\rho}(\tilde{g}, \tilde{M}) \in R(\tilde{G}, \tilde{m}_0)$ olmasını sağlayan $\tilde{g} \triangleq \tilde{t}_{i_1}, \tilde{t}_{i_2}, \dots, \tilde{t}_{i_{n-1}}, \tilde{t}_{i_n}$, $(\tilde{t}_{i_1}, \tilde{t}_{i_2}, \dots, \tilde{t}_{i_{n-1}} \in \mathcal{T}, \tilde{t}_{i_n} \in \bar{T})$ bir ateşleme dizisi olduğu durum:

$\tilde{g}^+ \triangleq \tilde{t}_{i_1}, \tilde{t}_{i_2}, \dots, \tilde{t}_{i_{n-1}}$ olarak tanımlanmak üzere, \tilde{G} ile gösterilen Petri ağında,

$$\tilde{M}'' = \tilde{M} + \tilde{A}\tilde{U}_{\tilde{g}^+} \quad (4.17)$$

olarak elde edilir. Burada denklemin her iki tarafı soldan Q matrisi ile çarpılırsa,

$$Q\tilde{M}'' = Q\tilde{M} + Q\tilde{A}\tilde{U}_{\tilde{g}^+} \quad (4.18)$$

elde edilir. \tilde{g}^+ ateşleme dizisi yalnızca \mathcal{T} kümesinin elemanlarından olduğundan ve Lemma 4.1'den dolayı,

$$Q\tilde{M}'' = Q\tilde{M} \quad (4.19)$$

olur. $\tilde{M}' = \tilde{\rho}(\tilde{t}_{i_n}, \tilde{M}'')$ işaretleme vektörü de,

$$\tilde{M}' = \tilde{M}'' + \tilde{A}\tilde{U}_{\tilde{t}_{i_n}} \quad (4.20)$$

elde edilir. Burada denklemin her iki tarafı soldan Q matrisi ile çarpılırsa,

$$Q\tilde{M}' = Q\tilde{M}'' + Q\tilde{A}\tilde{U}_{\tilde{t}_{i_n}} \quad (4.21)$$

elde edilir. Denklem (4.19) kullanılarak,

$$Q\tilde{M}' = Q\tilde{M} + Q\tilde{A}\tilde{U}_{\tilde{t}_{i_n}} \quad (4.22)$$

elde edilir. $\tilde{M}', \tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olduğundan ve \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerdiğinden, $M' = Q\tilde{M}'$, $M' \in R(G, m_0)$ olur. Böylece,

$$M' = M + Q\tilde{A}\tilde{U}_{\tilde{t}_{i_n}} \quad (4.23)$$

elde edilir. Denklem (4.23)'de $\tilde{t}_{i_n} \in \tilde{T}$ ise, \tilde{T} kümesinin tanımından dolayı, $Q\tilde{A}\tilde{U}_{\tilde{t}_{i_n}} = AU_t$ olduğu bir t geçişi vardır. Dolayısıyla, $M' = M + AU_t$ elde edilir.

Her iki durumda elde edilen ifadeler M işaretlemesinden sonra bir t geçişinin ateşlenebildiğini, dolayısıyla Petri ağında sistem çıkmazının meydana gelmediğini göstermektedir. Bu nedenle, \tilde{G} ile gösterilen Petri ağında çalışma engeli meydana gelmiyorsa, G ile gösterilen Petri ağında da sistem çıkmazı meydana gelmez. \square

Petri ağının önemli özelliklerden biri de canlılıktır. İçerme prensibi kullanılarak, canlılık özelliğinin incelenmesi aşağıda verilecektir.

Teorem 4.5: Herhangi bir $t \in T$ için, $AU_t = Q\tilde{A}\tilde{U}_{\tilde{t}}$ şartını sağlayan bir $\tilde{t} \in \tilde{T}$ geçişi var ise, \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içeriyorsa ve \tilde{G} ile gösterilen Petri ağı canlı ise, G ile gösterilen Petri ağı da canlıdır.

İspat: \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerdiğinden, herhangi bir $M \in R(G, m_0)$ için, $M = Q\tilde{M}$ şartını sağlayan $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ vardır. Herhangi bir $t \in T$ için, $\tilde{t} \in \tilde{T}$ $AU_t = Q\tilde{A}\tilde{U}_{\tilde{t}}$ şartını sağlayan geçiş olmak üzere, \tilde{G} ile gösterilen Petri ağı canlı olduğundan,

$$\tilde{M}'' = \tilde{M}' + \tilde{A}\tilde{U}_{\tilde{t}} = [\tilde{M} + \tilde{A}\tilde{U}_{\tilde{g}}] + \tilde{A}\tilde{U}_{\tilde{t}} \quad (4.24)$$

şartını sağlayan \tilde{g} ateşleme dizisi, $\tilde{M}' \in R(\tilde{G}, \tilde{M})$ ve $\tilde{M}'' \in R(\tilde{G}, \tilde{M}')$ olan işaretleme vektörleri vardır. \tilde{G} ile gösterilen Petri ağı, G ile gösterilen Petri ağını içerdiğinden, $M' = Q\tilde{M}'$ ve $M'' = Q\tilde{M}'' = Q[\tilde{M}' + \tilde{A}\tilde{U}_{\tilde{t}}] = M' + AU_t$ olan $M' \in R(G, M)$ ve $M'' \in R(G, M')$ işaretleme vektörleri vardır. Dolayısıyla, $M' \in R(G, M)$ işaretleme vektöründen sonra t geçişi ateşlenebilmektedir. Buradan da G ile gösterilen Petri ağının canlı olduğu bulunur. \square

4.3. Otomata ve Dil Gösterimlerinde İçerme Prensibi

Bölüm 4.1'de verildiği şekilde Petri ağlarında ve [28]'de gösterildiği şekilde de vektörel gösterimde içerme prensibi tanımlanmıştır. Bu bölümde, kesikli olay sistemlerin modelleme yöntemlerinden olan otomata ve dil gösterimleri için, içerme prensibi tanımlanacaktır.

Otomata veya dil gösterimi ile modellenmiş, \mathcal{S} ve $\tilde{\mathcal{S}}$ sistemlerini ele alalım.

\mathcal{S} sisteminin, durumların kümesini Q , olayların kümesini Σ , Σ kümesinin elemanlarından oluşan tüm sonlu dizilerin kümesi Σ^* , geçiş fonksiyonunu γ ve tanımlanan dili de $L = \{s \in \Sigma^* \mid \gamma(s, q_0) \text{ tanımlı ise}\}$ ile gösterelim.

$\tilde{\mathcal{S}}$ sisteminin, durumların kümesini \tilde{Q} , olayların kümesini $\tilde{\Sigma}$, $\tilde{\Sigma}$ kümesinin elemanlarından oluşan tüm sonlu dizilerin kümesi $\tilde{\Sigma}^*$, geçiş fonksiyonunu $\tilde{\gamma}$ ve tanımlanan dili de $\tilde{L} = \{\tilde{s} \in \tilde{\Sigma}^* \mid \tilde{\gamma}(\tilde{s}, \tilde{q}_0) \text{ tanımlı ise}\}$ ile gösterelim.

Tanım 4.3: Aşağıdaki şartları sağlayan $\Theta : \tilde{Q} \rightarrow Q$ ve $H : Q \rightarrow \tilde{Q}$ dönüşümleri varsa, \tilde{S} , S 'yi içerir.

$$(i) \Theta(H(q)) = q, \forall q \in Q$$

$$(ii) \tilde{q}_0 = H(q_0)$$

(iii) herhangi bir $\tilde{q} = \tilde{\gamma}(\tilde{s}, \tilde{q}_0)$, $\tilde{s} \in \tilde{\Sigma}^*$ için, eğer $\tilde{q}' = \tilde{\gamma}(\tilde{s}', \tilde{q})$, $\tilde{s}' \in \tilde{\Sigma}^*$ ise, $q = \Theta(\tilde{q})$ ve $q' = \Theta(\tilde{q}')$ olmak üzere, $q' = \gamma(s, q)$ şartını sağlayan $s \in \Sigma^*$ vardır.

(iv) herhangi bir $q = \gamma(s, q_0)$, $s \in \Sigma^*$ için, $q = \Theta(\tilde{q})$ olan $\tilde{q} = \tilde{\gamma}(\tilde{s}, \tilde{q}_0)$, $\tilde{s} \in \tilde{\Sigma}^*$ vardır.

5. ÖRTÜŞMELİ AYRIŞTIRMA ve GENLEŞTİRME

Sürekli durum sistemleri olarak modellenen büyük ölçekli sistemlerde, dışmerkezli kontrolör tasarımı için; sistemlerin, alt sistemlerine ayrıştırılmasından yararlanılmıştır [19]. Bu tür sistemlerde, alt sistemlerin birbirlerinden tamamen ayrışık olduğu “ayrışık ayrıştırma” yerine, “örtüşmeli ayrıştırma” uygulanmıştır (bkz. [20]). Ayrıca, vektörel gösterimle ifade edilen, sürekli durum sistemleri ile kesikli durum sistemlerinin birlikte kullanıldığı hibrid sistemlerde de, örtüşmeli ayrıştırmaya dayalı dışmerkezli kontrolör tasarımı, İftar ve Özgüner [28] tarafından gerçekleştirilmiştir. Bu yaklaşımı, vektörel gösterimle modellenmiş kesikli olay sistemlere uygulamak mümkündür.

Petri ağı ile gösterilen kesikli olay sistemler için merkezi bir kontrolör tasarlamak, Petri ağındaki yerlerin ve geçişlerin sayısı arttıkça zorlaşmaktadır. Bunun için bu çalışmada, Petri ağları için örtüşmeli ayrıştırma yöntemi geliştirilerek, dışmerkezli kontrolör tasarımı yapılmıştır. Petri ağına, örtüşmeli ayrıştırma ve genişletme işlemleri uygulanarak alt ağların ayrışık olarak elde edilmesi mümkün olmuştur. Ayrışık olarak elde edilen alt Petri ağlarının (APA'larının) her biri için (birbirinden bağımsız olarak) Bölüm 3.1'de verildiği şekilde, kontrolör tasarlanmaktadır. Bölüm 6.1'de detayları verildiği şekilde, tasarlanan bu kontrolörlerden bir dışmerkezli kontrolör elde edilmektedir.

Bu bölümde otomata ve dil gösterimlerinde de örtüşmeli ayrıştırma ve genişletmenin nasıl olabileceği ortaya konulacaktır. Otomata gösterimi için de, detayları Bölüm 6.2'de verildiği şekilde örtüşmeli ayrıştırma ve genişletmeye dayalı bir dışmerkezli kontrolör tasarım yöntemi geliştirilmiştir.

5.1. Petri Ağlarında Örtüşmeli Ayrıştırma ve Genleştirme

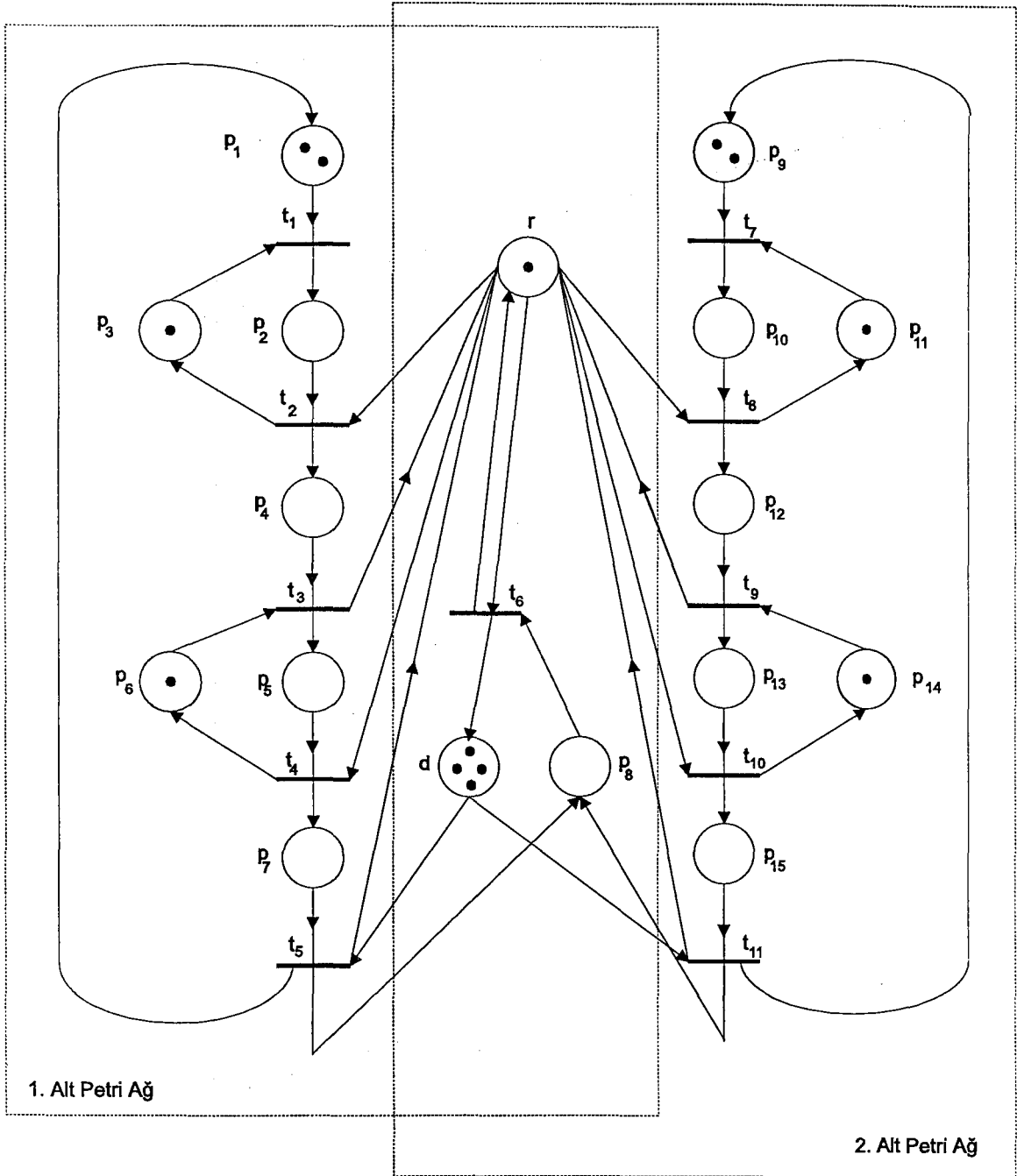
Örtüşmeli ayrıştırma işlemi, Petri ağının topolojisine bakılarak yapılmaktadır. Kesikli olay sistemlerde gerçekleşen olay veya işlemler, Petri ağında yerler ile gösterildiğinden; Petri ağında örtüşmeli ayrıştırma işlemi, yerlerin girdi-çıkış bağlantılarından yararlanılarak gerçekleştirilmektedir. Örtüşmeli alt Petri ağları aşağıda verilen özellikten yararlanılarak belirlenmektedir.

Örtüşmeli alt Petri ağlarının arasındaki bağlantı yalnızca örtüşmeli kısım üzerinden olacak şekilde olmalıdır. Yani, bir alt Petri ağındaki yerden/geçişten bir diğer alt Petri ağındaki geçişe/yere, örtüşmeli kısımdaki bağlantılar hariç, bağlantı olmamalıdır. Petri ağının örtüşmeli kısımda bulunan yer(ler) ve geçiş(ler)e, sırasıyla ortak yer(ler) ve ortak geçiş(ler) denir. Şekil 2.3'de verilen Petri ağının örtüşmeli ayrıştırılması sonucu Şekil 5.1 elde edilmiştir. Şekil 5.1'de verilen Petri ağı, iki alt Petri ağından meydana gelmektedir ve burada p_8 , r ve d ile gösterilen yerler, ortak yerlerdir. Ayrıca, Şekil 5.1'de verilen örnek sistemin Petri ağı modelinde, p_8 , r ve d ile gösterilen yerler arasında bağlantıyı sağlayan t_6 geçişi, ortak geçiştir.

Elde edilen alt Petri ağları, kesikli olay sistemlerdeki alt sistemlere karşılık gelebilirler. Örnek 2.2'de verilen sistemin iki alt sistemine, Şekil 5.1'de gösterilen alt Petri ağları karşılık gelmektedir. Buna göre, m_1 'de, bir palet üzerine sabitleştirilmiş parçaların üretiminin yapılması, robotun m_1 'de üretilen mamülleri d_1 'e yüklemesi, d_1 'de mamüllerin saklanması, robotun d_1 'i boşaltması ve paletlerin serbest kalması, paletten ayrılan mamüllerin ana depoda saklanması ve ana deponun yine robot tarafından boşaltılması şeklinde bir alt sistem ve m_2 'de, bir palet üzerine sabitleştirilmiş parçaların üretiminin yapılması, robotun m_2 'de üretilen mamülleri d_2 'ye yüklemesi, d_2 'de mamüllerin saklanması, robotun d_2 'yi boşaltması ve paletlerin serbest kalması, paletten ayrılan mamüllerin ana depoda saklanması ve ana deponun yine robot tarafından boşaltılması şeklinde de bir diğer alt sistem vardır. Bu alt sistemler, Şekil 5.1'de verilen Petri ağının sırasıyla 1. ve 2. alt Petri ağlarına karşılık gelmektedir. Şekil 5.1'de verilen Petri ağının örtüşmeli kısmı, robotun ve ana deponun ortak kullanımının olduğunu göstermektedir.

Petri ağının alt Petri ağları belirlendikten sonra genişletme işlemi yapılarak, alt Petri ağların ayrışık olarak elde edilmesi sağlanır. Bunun için Petri ağlarında genişletme işlemi aşağıda verildiği şekilde yapılmaktadır:

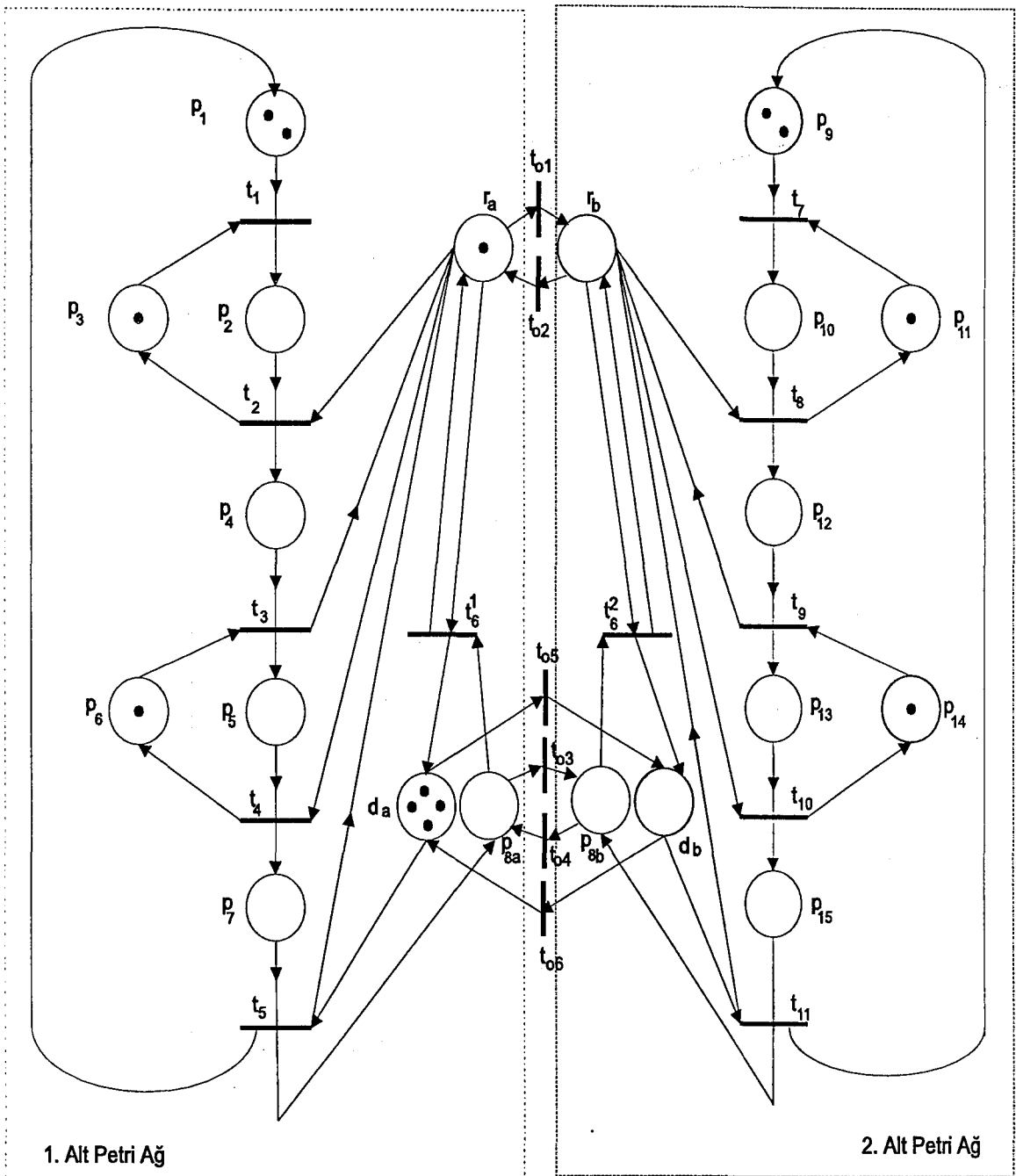
Petri ağının alt Petri ağları belirlendikten sonra, örtüşen kısımda bulunan ortak yerler dahil oldukları alt Petri ağlarında tekrarlanırlar. Daha sonra tekrarlanan bu yerler, alt Petri ağlarındaki bağlantıları aynı kalırken,



Şekil 5.1. Örtüsmeli ayrıştırma uygulanmış Petri ağı

ayrıştırılan yerlerin birbirleri ile bağlantılarını sağlamak için yeni geçişler eklenir. Örneğin, özgün Petri ağındaki p_o , genişletilmiş Petri ağında p_{oa} ve p_{ob} olarak elde edilir. Dolayısıyla, $P = \{\dots, p_k, \dots, p_{o1}, p_{o2}, p_{o3}, \dots, p_l, \dots\}$, genişletilince $\tilde{P} = \{\dots, p_k, \dots, p_{o1a}, p_{o1b}, p_{o1c}, \dots, p_{o2a}, p_{o2b}, \dots, p_l, \dots\}$ ($k, l \in \mathcal{N}$) şeklinde olmaktadır. Şekil 5.1'de verilen Petri ağında, p_8 , r ve d dahil oldukları alt Petri ağlarında tekrarlanmaları sonucunda, p_{8a} , p_{8b} , r_a , r_b , d_a ve d_b yerleri elde edilmektedir. Genişletilmiş Petri ağının yerler kümesi, $\tilde{P} = \bigcup_{i=1}^S P_i$ olarak ifade edilmektedir. Burada, P_i , i . alt Petri ağının yerlerinin oluşturduğu kümeyi ve S , alt Petri ağlarının sayısını göstermektedir. Şekil 5.2'de verilen genişletilmiş Petri ağında, $P_1 = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_{8a}, r_a, d_a\}$, $P_2 = \{p_{8b}, r_b, d_b, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$ ve $\tilde{P} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_{8a}, r_a, d_a, p_{8b}, r_b, d_b, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$ olmaktadır.

Örtüşen kısımda bulunan ortak geçişler, dahil oldukları alt Petri ağlarında tekrarlanırlar. Örtüşen kısımda bulunan bir t geçişi, t^r , $r \in \{1, \dots, n\}$ şeklinde tekrarlanmış geçişler ile gösterilir (n , t geçişinin dahil olduğu alt Petri ağı sayısıdır). Şekil 5.1'deki t_6 geçişi, genişletilmiş Petri ağında t_6^1 ve t_6^2 geçişleri ile gösterilmektedir (bkz. Şekil 5.2). Tekrarlanan bu geçişler, aynı alt Petri ağında bulunan tekrarlanmış yerlerin birbirleri ile olan bağlantılarını sağlarlar. Bir ortak yerin, tekrarlanmış yerleri arasındaki bağlantı ise yeni geçişler ile sağlanır. Örneğin, özgün Petri ağındaki p_o , genişletilmiş Petri ağında p_{oa} ve p_{ob} olarak elde edildiğinde, p_{oa} ile p_{ob} arasında, p_{oa} 'dan p_{ob} 'ye bir geçiş ve p_{ob} 'den p_{oa} 'ya bir geçiş eklenir. Örneğin, Şekil 5.1'deki Petri ağında r , Şekil 5.2'de gösterildiği şekilde alt Petri ağlarında tekrarlanması sonucu r_a ve r_b yerleri elde edilmiştir. Bu iki yer arasında bağlantıyı sağlamak için, t_{o1} ve t_{o2} geçişleri ilave edilmiştir. Alt Petri ağlarında tekrarlanan yerler hariç diğer yerler arasındaki geçişler, özgün Petri ağındaki şekliyle korunmaktadır. Böylece genişletilmiş Petri ağının geçişlerinin kümesi, $\tilde{T} = \bar{T} \cup \mathcal{T}$ olarak elde edilir. Burada, $\bar{T} = \bigcup_{i=1}^S T_i$ ve \mathcal{T} , alt Petri ağları arasında bağlantıyı sağlamak için ilave edilen geçişlerin oluşturduğu kümeyi göstermektedir. Ayrıca \tilde{T} kümesi, $\tilde{T} = T_1 \cup T_1^{12} \cup T_1^{13} \cup \dots \cup T_2 \cup T_2^{21} \cup T_2^{23} \cup \dots$ olarak da elde edilir. Burada, T_i , i . APA'nın geçişlerinin oluşturduğu kümeyi, T_i^{ij} , bağlantı



Şekil 5.2. Genleştirilmiş Petri ağı

şekli i . APA'dan, j . ($j \neq i$) APA'na doğru olan geçişlerin oluşturduğu kümeyi göstermektedir. Şekil 5.2'deki genişletilmiş Petri ağ için, $\bar{T} = \{t_1, t_2, t_3, t_4, t_5, t_6^1, t_6^2, t_7, t_8, t_9, t_{10}, t_{11}\}$, $\mathcal{T} = \{t_{o1}, t_{o2}, t_{o3}, t_{o4}, t_{o5}, t_{o6}\}$, $T_1 = \{t_1, t_2, t_3, t_4, t_5, t_6^1\}$, $T_1^{12} = \{t_{o1}, t_{o3}, t_{o5}\}$, $T_2 = \{t_6^2, t_7, t_8, t_9, t_{10}, t_{11}\}$, $T_2^{21} = \{t_{o2}, t_{o4}, t_{o6}\}$ ve $\tilde{T} = \{t_1, t_2, t_3, t_4, t_5, t_6^1, t_{o1}, t_{o3}, t_{o5}, t_6^2, t_7, t_8, t_9, t_{10}, t_{11}, t_{o2}, t_{o4}, t_{o6}\}$ olarak elde edilir.

Elde edilen genişletilmiş Petri ağının yerler ve geçişler kümeleri kullanılarak, genişletilmiş Petri ağının girdi-çıkış matrisleri de aşağıdaki şekilde elde edilir.

$$\tilde{N} = \begin{bmatrix} N_1 & N_1^{12} & \dots & 0 & N_1^{21} & \dots & 0 & \dots \\ 0 & N_2^{12} & \dots & N_2 & N_2^{21} & \dots & 0 & \dots \\ 0 & \vdots & \dots & 0 & \vdots & \dots & N_3 & \dots \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots \end{bmatrix}, \quad \tilde{N} : \tilde{P} \times \tilde{T} \rightarrow \mu$$

$$\tilde{O} = \begin{bmatrix} O_1 & O_1^{12} & \dots & 0 & O_1^{21} & \dots & 0 & \dots \\ 0 & O_2^{12} & \dots & O_2 & O_2^{21} & \dots & 0 & \dots \\ 0 & \vdots & \dots & 0 & \vdots & \dots & O_3 & \dots \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots \end{bmatrix}, \quad \tilde{O} : \tilde{P} \times \tilde{T} \rightarrow \mu$$

Burada, N_i ve O_i ($i \in S$), i . alt Petri ağının girdi ve çıkış matrislerini, N_i^{ij} ve O_i^{ij} ($i, j \in S$), bağlantı şekli i . APA'dan, j . ($j \neq i$) APA'na doğru olan geçişlerin yerlerle olan bağlantılarını gösteren girdi ve çıkış matrislerini göstermektedir. Şekil 5.2'deki genişletilmiş Petri ağının, girdi çıkış matrisleri aşağıdaki şekilde gösterilmektedir.

$$\tilde{N} = \begin{bmatrix} N_1 & N_1^{12} & Z & N_1^{21} \\ Z & N_2^{12} & N_2 & N_2^{21} \end{bmatrix} \quad \tilde{O} = \begin{bmatrix} O_1 & O_1^{12} & Z & O_1^{21} \\ Z & O_2^{12} & O_2 & O_2^{21} \end{bmatrix}$$

Z : 6×9 'luk sıfır matrisi, N_1 , O_1 ve N_2 , O_2 ; 1. ve 2. APA'larının girdi-çıkış matrisleridir.

$$N_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad N_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$O_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad O_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Yukarıda anlatıldığı şekilde özgün Petri ağına, örtüşmeli ayrıştırma ve genişletme uygulanarak geliştirilmiş Petri ağ elde edilmektedir. Elde edilen geliştirilmiş Petri ağı ile özgün Petri ağı arasında Bölüm 4.1'de verilen içerme prensibinin olup olmadığını araştırmak için her iki Petri ağının, $\tilde{A} = \tilde{O} - \tilde{N}$ ve $A = O - N$ bağlantı matrisleri arasındaki ilişki incelenmiştir. Sonuç 5.1'de gösterilen iki matris arasındaki ilişki ise, Lemma 5.1 ve 5.2'nin ispatında kullanılmaktadır.

Sonuç 5.1: A ve \tilde{A} bağlantı matrisleri elemanları cinsinden sırasıyla $A(p, t)$ ve $\tilde{A}(\tilde{p}, \tilde{t})$ ile gösterilmektedir. Burada, $A(p, t)$, A matrisinde p yeri ile t geçişi

arasındaki bağlantının değerini ve $\tilde{A}(\tilde{p}, \tilde{t})$, \tilde{A} matrisinde \tilde{p} yeri ile \tilde{t} geçişi arasındaki bağlantının değerini göstermektedir. A ve \tilde{A} bağlantı matrisleri elemanları arasındaki ilişki aşağıda verilmiştir.

- 1) p , ÖPA'da örtüşmeli kısımda bulunmayan bir yeri ve \tilde{p} , bu yere GPA'da karşılık gelen yeri, t , ÖPA'da örtüşmeli kısımda bulunmayan bir geçişi ve \tilde{t} , bu geçişe GPA'da karşılık gelen geçişi gösteriyorsa,

$$\tilde{A}(\tilde{p}, \tilde{t}) = A(p, t)$$

- 2) p , ÖPA'da örtüşmeli kısımda bulunan bir yeri ve \tilde{p} , bu yerin GPA'da tekrarlanmış yerlerinden birini, t , ÖPA'da örtüşmeli kısımda bulunan bir geçişi ve \tilde{t} , bu geçişin GPA'da tekrarlanmış geçişlerinden birini gösteriyorsa,

$$\tilde{A}(\tilde{p}, \tilde{t}) = \begin{cases} A(p, t), & \tilde{p} \text{ ile } \tilde{t} \text{ aynı APA'da bulunuyorlar ise} \\ 0, & \text{aksi durumda} \end{cases}$$

- 3) p , ÖPA'da örtüşmeli kısımda bulunan bir yeri ve \tilde{p} , bu yerin GPA'da tekrarlanmış yerlerinden birini, t , ÖPA'da örtüşmeli kısımda bulunmayan bir geçişi ve \tilde{t} , bu geçişe GPA'da karşılık gelen geçişi gösteriyorsa,

$$\tilde{A}(\tilde{p}, \tilde{t}) = \begin{cases} A(p, t), & \tilde{p} \text{ ile } \tilde{t} \text{ aynı APA'da bulunuyorlar ise} \\ 0, & \text{aksi durumda} \end{cases}$$

- 4) p , ÖPA'da örtüşmeli kısımda bulunmayan bir yeri ve \tilde{p} , bu yere GPA'da karşılık gelen yeri, t , ÖPA'da örtüşmeli kısımda bulunan bir geçişi ve \tilde{t} , bu geçişin GPA'da tekrarlanmış geçişlerinden birini gösteriyorsa,

$$\tilde{A}(\tilde{p}, \tilde{t}) = \begin{cases} A(p, t), & \tilde{p} \text{ ile } \tilde{t} \text{ aynı APA'da bulunuyorlar ise} \\ 0, & \text{aksi durumda} \end{cases}$$

Genleştirilmiş Petri ağının başlangıç işaretlemesi, özgün Petri ağındaki yerlerin belirti sayıları kullanılarak aşağıdaki şekilde belirlenir.

- özgün Petri ağında örtüşmeli kısımda bulunmayan bir yerin belirtileri, genleştirilmiş Petri ağında bu yere karşılık gelen yerin belirtileri olarak alınır.

- özgün Petri ağında örtüşmeli kısımda bulunan bir yerdeki belirtilerin tamamı, geliştirilmiş Petri ağında bu yerin tekrarlanmış yerlerinin herhangi birinde olacak şekilde alınır.

Özgün Petri ağının başlangıç işaretlemesi (m_0) kullanılarak, geliştirilmiş Petri ağı için birden fazla başlangıç işaretlemesi elde edilebilir. Şekil 5.2'de verilen geliştirilmiş Petri ağının başlangıç işaretlemesi, başlangıçta robotun ve ana deponun tüm bölmelerinin 1. alt Petri ağına hizmet vermesi şeklinde bir seçim yapılması ile, $\tilde{m}_0 = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 4 \ 0 \ 0 \ 0 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ olarak elde edilir. Başka bir başlangıç işaretlemesi de, başlangıçta robotun ve ana deponun tüm bölmelerinin 2. alt Petri ağına hizmet verecek şekilde seçilmesi ile elde edilir: $\tilde{m}'_0 = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 4 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$. Başlangıç işaretlemesi de seçildikten sonra, Şekil 5.2'de gösterilen geliştirilmiş Petri ağı $\tilde{G}(\tilde{P}, \tilde{T}, \tilde{N}, \tilde{O}, \tilde{m}_0)$ ile gösterilir.

5.2. Geliştirilmiş Petri Ağı ile Özgün Petri Ağı Arasındaki Bazı İlişkiler

Burada, Bölüm 5.1'de bahsedildiği şekilde elde edilen geliştirilmiş Petri ağı ile özgün Petri ağı arasındaki ilişkiler incelenecektir. $\mathcal{E}(p)$, ÖPA'daki p yerine, GPA'da karşılık gelen yerlerin oluşturduğu kümeyi ve $\mathcal{E}(t)$, ÖPA'daki t geçişine, GPA'da karşılık gelen geçişlerin oluşturduğu kümeyi göstermektedir. Ayrıca, $\mathcal{E}^{-1} : \tilde{T} \rightarrow T$ dönüşümü $t = \mathcal{E}^{-1}(\tilde{t}) \iff \tilde{t} \in \mathcal{E}(t)$ olarak tanımlanmıştır. Geliştirilmiş Petri ağında tanımlanan ateşleme dizileri ile özgün Petri ağında tanımlanan ateşleme dizileri arasında aşağıda verildiği gibi iki özellik bulunmaktadır.

Lemma 5.1: GPA'da herhangi bir $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ ve $\tilde{M}' \in R(\tilde{G}, \tilde{M})$ işaretleme vektörleri için, \tilde{g} ateşleme dizisi $\tilde{M}' = \tilde{\rho}(\tilde{g}, \tilde{M})$ eşitliğini sağlayan ateşleme dizisi olsun. Bu durumda ÖPA'da $M' = \rho(g, M)$ şartını sağlayan bir g ateşleme dizisi vardır. Burada M ve M' ,

$$M(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}(\tilde{p}), \forall p \in P \quad (5.1)$$

ve

$$M'(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}'(\tilde{p}), \forall p \in P \quad (5.2)$$

eşitliklerini sağlayan işaretleme vektörleridir.

İspat: \tilde{g} ateşleme dizisini oluşturan, ilk geçiştten başlayarak son geçişe kadar, her bir geçiş için, aşağıda verilen uygun durumlardan elde edilen geçişlerden, g dizisi oluşturulur. \tilde{g} ateşleme dizisinde ele alınan geçiş, \tilde{t} ,

- i) $\tilde{t} \in \mathcal{T}$ ise, ÖPA'da bu geçiş karşılık gelen bir geçiş alınmaz.
- ii) $\tilde{t} \in \bar{\mathcal{T}}$ ise, $t = \mathcal{E}^{-1}(\tilde{t})$ geçişi alınır.

Elde edilen g dizisinin, ÖPA için bir ateşleme dizisi olduğu (g ateşleme dizisindeki her bir geçişin, sırası geldiğinde ateşlenebildiği) ve ÖPA'da kullanıldığında (5.2) eşitliğini sağladığı aşağıda gösterilecektir.

$\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olmak üzere, $\tilde{M}' = \tilde{\rho}(\tilde{g}, \tilde{M}) \in R(\tilde{G}, \tilde{M})$ işaretleme vektörünün elde edilmesini sağlayan ateşleme dizisi, $\tilde{g} = \tilde{t}$ olsun.

a) $\tilde{t} \in \mathcal{T}$ ise,

- \tilde{t} geçişi yalnızca tekrarlanmış yerlere girdi-çıkıtı olarak bağlı olduğundan, \tilde{p} tekrarlanmış bir yer değil ise,

$$\tilde{A}(\tilde{p}, \tilde{t}) = 0 \quad (5.3)$$

olmaktadır.

- \tilde{t} geçişi, tekrarlanmış iki yer arasında ise, birine girdi (çıkıtı) olarak bağlı, diğerine çıkıtı (girdi) olarak bağlıdır ya da tekrarlanmış iki yerle de bağlantısı yoktur. Bu nedenle,

$$\sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{A}(\tilde{p}, \tilde{t}) = 0 \quad (5.4)$$

olmaktadır.

$\tilde{M}' = \tilde{\rho}(\tilde{t}, \tilde{M})$ işaretleme vektörü elemanları cinsinden,

$$\tilde{M}'(\tilde{p}) = \tilde{M}(\tilde{p}) + \tilde{A}(\tilde{p}, \tilde{t}), \tilde{p} \in \tilde{P} \quad (5.5)$$

olarak ifade edilmektedir. Burada, denklem (5.3) ve (5.4) kullanılarak,

$$\sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}'(\tilde{p}) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}(\tilde{p}), \forall p \in P \quad (5.6)$$

elde edilir. ÖPA'da \tilde{t} geçişine karşılık hiç bir geçiş olmadığından, (5.6) ve (5.1) eşitliklerinden dolayı, aşağıdaki verilen eşitlik bulunmaktadır.

$$M'(p) = M(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}(\tilde{p}) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}'(\tilde{p}), \forall p \in P$$

Dolayısıyla, (5.2) eşitliği sağlanmaktadır.

- b) $\tilde{t} \in \bar{T}$ ise, \tilde{t} geçişi ateşlenebildiğinden, \tilde{t} geçişinin çıktı olarak bağlı olduğu yerlerin her birinde en az bir belirti vardır. Bu ifade aşağıdaki şekilde gösterilmektedir.

$$\tilde{M}(\tilde{p}) > 0, \forall \tilde{p} \in \{\tilde{p} \in \tilde{P} \mid \tilde{A}(\tilde{p}, \tilde{t}) = -1\} =: \tilde{\mathcal{P}}(\tilde{t})$$

t geçişinin ateşlenebilmesi için de,

$$M(p) > 0, \forall p \in \{p \in P \mid A(p, t) = -1\} =: \mathcal{P}(t)$$

olması gerekir. Sonuç 5.1'den dolayı, $A(p, \mathcal{E}^{-1}(\tilde{t})) = -1$ ise, bir $\tilde{p} \in \mathcal{E}(p)$ için $\tilde{A}(\tilde{p}, \tilde{t}) = -1$ olmaktadır. Buradan, \tilde{t} ateşlenebilir bir geçiş olduğundan ve (5.1)'den dolayı, $M(p) > 0$ olmaktadır. Dolayısıyla t geçişinin ateşlenebilmesi için gerekli şart sağlanmaktadır.

\tilde{A} ve A bağlantı matrislerinin elemanları arasında, \tilde{t} ve $t = \mathcal{E}^{-1}(\tilde{t})$ geçişleri kullanılarak, Sonuç 5.1'den,

$$A(p, t) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{A}(\tilde{p}, \tilde{t}), \forall p \in P \quad (5.7)$$

eşitliği bulunmaktadır. $M' = \rho(t, M)$ işaretleme vektörü elemanları cinsinden,

$$M'(p) = M(p) + A(p, t), p \in P \quad (5.8)$$

olarak ifade edilmektedir. Burada, denklem (5.7), (5.1) ve (5.5) kullanılarak, (5.2) eşitliği elde edilir.

$\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olmak üzere, $\tilde{M}' = \tilde{\rho}(\tilde{g}, \tilde{M}) \in R(\tilde{G}, \tilde{M})$ işaretleme vektörünün elde edilmesini sağlayan ateşleme dizisi, $\tilde{g} = \tilde{t}_{k_1} \tilde{t}_{k_2} \dots \tilde{t}_{k_m}$ olsun. Bu durumda, \tilde{M}' işaretleme vektörü, $\tilde{M}' = \tilde{\rho}(\tilde{t}_{k_m}, (\dots, \tilde{\rho}(\tilde{t}_{k_1}, \tilde{M})))$ olarak elde edilmektedir.

A) $\tilde{\rho}(\tilde{t}_{k_1}, \tilde{M})$ işaretleme vektörü, \tilde{M}_{k_1} ile gösterilsin. \tilde{t}_{k_1} geçişi için (i). veya (ii). durumlardan uygun olanı alınarak, yukarıda verilen işlemler yapıldığında, $M_{k_1}(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}_{k_1}(\tilde{p})$, $\forall p \in P$ eşitliği sağlanmaktadır.

B) \tilde{t}_{k_j} geçişinin ateşlenmesiyle, ulaşılan \tilde{M}_{k_j} işaretleme vektörü ile M_{k_j} işaretleme vektörü arasında, $M_{k_j}(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}_{k_j}(\tilde{p})$, $\forall p \in P$ eşitliği sağlandığı kabul edilsin. $\tilde{t}_{k_{j+1}}$ geçişi için (i). veya (ii). durumlardan uygun olanı alınarak, yukarıda verilen işlemler yapıldığında, $M_{k_{j+1}}(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}_{k_{j+1}}(\tilde{p})$, $\forall p \in P$ eşitliği sağlanmaktadır.

(A) ve (B) durumlarından endüksiyon yöntemiyle istenilen sonuç elde edilmektedir. \square

Lemma 5.2: ÖPA'da herhangi bir $M \in R(G, m_0)$ işaretleme vektörü için, g ateşleme dizisi $M = \rho(g, m_0)$ eşitliğini sağlayan dizi olsun. Bu durumda GPA'da $\tilde{M} = \tilde{\rho}(\tilde{g}, \tilde{m}_0)$ eşitliğini sağlayan bir \tilde{g} ateşleme dizisi vardır. Burada \tilde{M} , (5.1) eşitliğini sağlayan bir işaretleme vektörüdür.

İspat: \tilde{m}_0 başlangıç işaretlemesinin seçiminden (bkz. Bölüm 5.1) dolayı,

$$m_0(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{m}_0(\tilde{p}), \forall p \in P \quad (5.9)$$

olmaktadır. $M = \rho(g, m_0)$ işaretleme vektörünün elde edilmesini sağlayan g ateşleme dizisi, $g = t$ olsun. t ateşlenebilir bir geçiş olduğundan dolayı, $m_0(p) > 0$, $\forall p \in \mathcal{P}(t)$ sağlanmaktadır ($\mathcal{P}(t)$ ve $\tilde{\mathcal{P}}(\tilde{t})$ Lemma 5.1'in ispatında tanımlanmıştır). t geçişine karşılık, aşağıda verilen uygun durumlardan elde edilen geçişlerden, \tilde{g} ateşleme dizisi oluşturulur.

i) Herhangi bir $\tilde{t} \in \mathcal{E}(t)$ geçişi, $\tilde{m}_0(\tilde{p}) > 0, \forall \tilde{p} \in \tilde{\mathcal{P}}(\tilde{t})$ şartını sağlıyorsa, t geçişine karşılık $\tilde{g} = \tilde{t}$ ateşleme dizisi alınır.

ii) Hiç bir $\tilde{t} \in \mathcal{E}(t)$ geçişi, $\tilde{m}_0(\tilde{p}) > 0, \forall \tilde{p} \in \tilde{\mathcal{P}}(\tilde{t})$ şartını sağlamıyorsa, herhangi bir $\tilde{t} \in \mathcal{E}(t)$ geçişi için, bu şartın sağlanması için aşağıdaki şekilde bir \tilde{g}^+ ateşleme dizisi tanımlanabilir. Sonuç 5.1'den dolayı, $A(p, \mathcal{E}^{-1}(\tilde{t})) = -1$ ise bir $\tilde{p} \in \mathcal{E}(p)$ için $\tilde{A}(\tilde{p}, \tilde{t}) = -1$ olmaktadır. Buradan ve t ateşlenebilir bir geçiş olduğundan,

a) $p \in \mathcal{P}(t)$ örtüşmeli kısımda bulunmuyorsa, $\mathcal{E}(p) = \{\tilde{p}\}$, denklem (5.9)'dan dolayı, $\tilde{m}_0(\tilde{p}) = m_0(p) > 0$ olmaktadır.

b) $p \in \mathcal{P}(t)$ örtüşmeli kısımda bulunuyorsa, denklem (5.9)'dan dolayı, en az bir $\tilde{p}_r \in \mathcal{E}(p)$ için $\tilde{m}_0(\tilde{p}_r) > 0$ olmaktadır. \tilde{p}_r ile $\tilde{t} \in \mathcal{E}(t)$ geçişinin çıktısı olarak bağlı olduğu $\tilde{p} \in \mathcal{E}(t)$ arasında bağlantıyı sağlayan ve \mathcal{T} kümesinin elemanı olan geçiş ateşlenerek, \tilde{p} 'ya belirti ulaştırılır.

$\tilde{t} \in \mathcal{E}(t)$ geçişinin, çıktısı olarak bağlı olduğu belirti olmayan tekrarlanmış yerlerin her biri için (b) şıkında verilen işlem ayrı ayrı yapılır. Bu işlemde kullanılan geçişlerden \tilde{g}^+ ateşleme dizisi oluşturur. t geçişine karşılık $\tilde{g} = \tilde{g}^+ \tilde{t}$ ateşleme dizisi alınır.

$M = \rho(t, m_0)$ ile $\tilde{M} = \tilde{\rho}(\tilde{g}, \tilde{m}_0)$ 'nın, (5.1) eşitliğini sağladığı gösterilecektir.

- t geçişine karşılık $\tilde{g} = \tilde{t} \in \mathcal{E}(t)$ ateşleme dizisi geliyorsa; $M = \rho(t, m_0)$ ve $\tilde{M} = \tilde{\rho}(\tilde{t}, \tilde{m}_0)$ işaretleme vektörleri, sırasıyla, elemanları cinsinden aşağıdaki şekilde gösterilir.

$$M(p) = m_0(p) + A(p, t), \forall p \in P \quad (5.10)$$

$$\tilde{M}(\tilde{p}) = \tilde{m}_0(\tilde{p}) + \tilde{A}(\tilde{p}, \tilde{t}), \forall \tilde{p} \in \tilde{P} \quad (5.11)$$

Denklem (5.10)'da, denklem (5.9), (5.7) ve (5.11) kullanılarak, (5.1) eşitliği elde edilir.

- t geçişine karşılık $\tilde{g} = \tilde{g}^+ \tilde{t}$ ($\tilde{t} \in \mathcal{E}(t)$) ateşleme dizisi geliyorsa; $\tilde{g} = \tilde{g}^+ \tilde{t}$ ateşleme dizisi kullanılarak, $\tilde{M} = \tilde{\rho}(\tilde{t}, \tilde{\rho}(\tilde{g}^+, \tilde{m}_0))$ olmaktadır. $\tilde{\rho}(\tilde{g}^+, \tilde{m}_0)$ işaretleme vektörü, \tilde{M}'' ile gösterilsin. \tilde{M}'' ile \tilde{m}_0 arasında, \tilde{g}^+ ateşleme dizisi \mathcal{T} kümesinin elemanlarından oluştuğundan ve denklem (5.3), (5.4)'den dolayı,

$$\sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}''(\tilde{p}) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{m}_0(\tilde{p}), \quad \forall p \in P \quad (5.12)$$

eşitliği elde edilir. $\tilde{M} = \tilde{\rho}(\tilde{t}, \tilde{M}'')$ işaretleme vektörü, elemanları cinsinden aşağıdaki şekilde gösterilir.

$$\tilde{M}(\tilde{p}) = \tilde{M}''(\tilde{p}) + A(\tilde{p}, \tilde{t}), \quad \forall \tilde{p} \in \tilde{P} \quad (5.13)$$

Denklem (5.10)'da, denklem (5.9), (5.7), (5.12) ve (5.13)'den dolayı, (5.1) eşitliği elde edilir.

$M = \rho(g, m_0)$ işaretleme vektörünün elde edilmesini sağlayan ateşleme dizisi, $g = t_{k_1} t_{k_2} \dots t_{k_m}$ olsun. Bu durumda, M işaretleme vektörü, $M = \rho(t_{k_m}, (\dots, \rho(t_{k_1}, m_0)))$ olarak elde edilmektedir.

A) $\rho(t_{k_1}, m_0)$ işaretleme vektörü, M_{k_1} ile gösterilsin. t_{k_1} geçişine karşılık, (i). veya (ii). durumlardan biri kullanılarak, \tilde{g}_{k_1} ateşleme dizisi gelir. Yukarıda verilen işlemler yapıldığında, $M_{k_1}(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}_{k_1}(\tilde{p}), \quad \forall p \in P$ eşitliği sağlanmaktadır.

B) t_{k_j} geçişinin ateşlenmesiyle, ulaşılan M_{k_j} işaretleme vektörü ile \tilde{M}_{k_j} işaretleme vektörü arasında, $M_{k_j}(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}_{k_j}(\tilde{p}), \quad \forall p \in P$ eşitliği sağlandığı kabul edilsin. $t_{k_{j+1}}$ geçişine karşılık, (i). veya (ii). durumlardan biri kullanılarak, $\tilde{g}_{k_{j+1}}$ ateşleme dizisi gelir. Yukarıda verilen işlemler yapıldığında, $M_{k_{j+1}}(p) = \sum_{\tilde{p} \in \mathcal{E}(p)} \tilde{M}_{k_{j+1}}(\tilde{p}), \quad \forall p \in P$ eşitliği sağlanmaktadır.

(A) ve (B) durumlarından endüksiyon yöntemiyle istenilen sonuç elde edilmektedir. \square

Genleştirilmiş Petri ağının özgün Petri ağını içerdiği aşağıda verilen teoremden gösterilecektir.

Teorem 5.1: Bölüm 5.1’de anlatıldığı şekilde elde edilen genleştirilmiş Petri ağı, özgün Petri ağını içerir.

İspat:

Q ve V dönüşüm matrislerinin seçimi:

\mathcal{Z} , ÖPA’da örtüşmeli kısımda bulunan yerlerin, ÖPA’nın işaretleme vektörünün kaçınıcı elemanı olduğunu gösteren sayılardan oluşan kümeyi,

$\beta(i)$, i . ortak yerin ($i \in \mathcal{Z}$), GPA’daki tekrarlanmış yerlerinin ÖPA’nın işaretleme vektörünün kaçınıcı elemanı olduğunu gösteren sayılardan oluşan kümeyi,

$\bar{\mathcal{Z}}$, ÖPA’da örtüşmeli kısımda bulunmayan yerlerin, ÖPA’nın işaretleme vektörünün kaçınıcı elemanı olduğunu gösteren sayılardan oluşan kümeyi,

$\bar{\beta}_k$, $\bar{\mathcal{Z}}$ kümesinin elemanı olan sayıya karşılık gelen bir yerin ($k \in \bar{\mathcal{Z}}$), ÖPA’nın işaretleme vektörünün kaçınıcı elemanı olduğunu ifade eden sayıyı göstermektedir.

V matrisi, Bölüm 5.1’de anlatıldığı şekilde elde edilen genleştirilmiş Petri ağının başlangıç işaretlemesini özgün Petri ağının başlangıç işaretlemesinden nasıl elde edildiğini gösteren matristir. Burada özgün Petri ağında örtüşmeli kısımda bulunmayan yerlerin belirti sayısı, genleştirilmiş Petri ağında aynen alınır; $V(\bar{\beta}_k, k) = 1, \forall k \in \bar{\mathcal{Z}}$ ve özgün Petri ağında örtüşmeli kısımda bulunan yerlerin belirtilerinin tamamı, Bölüm 5.1’de anlatıldığı şekilde, bu yerlerin genleştirilmiş Petri ağında tekrarlanmış yerlerinden herhangi birinde olacak şekilde alınır; $\sum_{j \in \beta(i)} V(j, i) = 1, \forall i \in \mathcal{Z}$ olacak şekilde, her bir $V(j, i)$ ’nin değeri tamsayı olarak seçilmektedir (yani, içlerinden bir tanesi “1” alınırken, diğerleri “0” alınmaktadır). V matrisinin geri kalan elemanları da sıfır olarak seçilir ($V \in \mu^{|\bar{P}| \times |P|}$).

V matrisi seçimi yapıldıktan sonra, $QV = I$ olacak şekilde, $Q(i, j) =$

1, $\forall i \in \mathcal{Z}, j \in \beta(i)$ ve $Q(k, \bar{\beta}_k) = 1, \forall k \in \bar{\mathcal{Z}}$ olarak seçilir. Q matrisinin geri kalan elemanları da sıfır olarak seçilir ($Q \in \mu^{|\mathcal{P}| \times |\bar{\mathcal{P}}|}$).

$$\sum_{q=1}^{|\bar{\mathcal{P}}|} Q(n, q)V(q, r) = I(n, r), \quad n, r \in \{1, 2, \dots, |P|\}$$

olarak seçildiğinden, $QV = I$ olur. Böylece Tanım 4.1'in (i). şıkkı (denklem (4.1a)) sağlanmış olur.

V matrisinin seçimi, ÖPA'nın ve GPA'nın başlangıç koşullarından yararlanılarak yapıldığından Tanım 4.1'in (ii). şıkkı (denklem (4.1b)) sağlanmış olur.

Lemma 5.1'den dolayı, $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ ve $\tilde{M}' = \tilde{\rho}(\tilde{g}, \tilde{M})$ ise (5.1) ve (5.2) eşitliklerini sağlayan $M \in R(G, m_0)$ ve $M' = \rho(g, M)$ vardır. Q matrisinin seçiminden dolayı, (5.1) ve (5.2) eşitlikleri, sırasıyla $M = Q\tilde{M}$ ve $M' = Q\tilde{M}'$ olmaktadır. Bu da, $Q\tilde{M}$ işaretleme vektöründen sonra g ateşleme dizisi kullanılarak $Q\tilde{M}' \in R(G, Q\tilde{M})$ olan bir işaretleme vektörüne ulaşıldığını göstermektedir. Dolayısıyla, Tanım 4.1'in (iii). şıkkı sağlanmış olur.

Lemma 5.2'den dolayı, $M = \rho(g, m_0)$ ise (5.1) eşitliğini sağlayan $\tilde{M} = \tilde{\rho}(\tilde{g}, \tilde{m}_0)$ vardır. Q matrisinin seçiminden dolayı, (5.1) eşitliği, $Q\tilde{M} = M$ olmaktadır. Dolayısıyla, Tanım 4.1'in (iv). şıkkı sağlanmış olur.

Böylece genişletilmiş Petri ağı, özgün Petri ağını içerir. \square

Petri ağlarında tanımlanan örtüşmeli ayrıştırma ve genişletme işlemleri sonucu, Teorem 4.5'de ilave olarak istenen şartın sağlandığı aşağıda gösterilecektir.

Lemma 5.3: $t \in T$ olmak üzere, $AU_t = Q\tilde{A}\tilde{U}_{\tilde{t}}, \forall \tilde{t} \in \mathcal{E}(t)$ şartı sağlanmaktadır.

İspat: $AU_t = Q\tilde{A}\tilde{U}_{\tilde{t}}$ ifadesi, (5.7) eşitliği ile gösterilebilir. t geçişi ve herhangi bir $\tilde{t} \in \mathcal{E}(t)$ geçişi alındığında, Sonuç 5.1'den ve Q matrisinin seçiminden dolayı, (5.7) eşitliği sağlanmaktadır. Sonuç olarak, $t \in T$ olmak üzere, $\mathcal{E}(t)$ kümesinin bütün elemanları, $AU_t = Q\tilde{A}\tilde{U}_{\tilde{t}}$ şartını sağlamaktadır. \square

Yukarıda verilen tüm sonuçlar, genişletilmiş Petri ağı ile özgün Petri ağı arasında aşağıdaki özellikleri gerektirmektedir.

Sonuç 5.2: Bölüm 5.1’de anlatıldığı şekilde elde edilen genişleştirilmiş Petri ağı, özgün Petri ağını içerdiğinden (Teorem 5.1), Teorem 4.2 – 4.4’den ve Lemma 5.3 ile Teorem 4.5’den dolayı, genişleştirilmiş Petri ağı ile özgün Petri ağı arasında ilişkiler sırasıyla aşağıda verilmiştir:

- Genleştirilmiş Petri ağı sınırlı ise, özgün Petri ağı da sınırlıdır.
- Genleştirilmiş Petri ağı tersine dönüşebilir ise, özgün Petri ağı da tersine dönüşebilir.
- Genleştirilmiş Petri ağında çalışma engeli meydana gelmiyorsa, özgün Petri ağında da sistem çıkmazı meydana gelmez.
- Genleştirilmiş Petri ağı canlı ise, özgün Petri ağı da canlıdır.

alt Petri ağları arasında

Şekil 2.3’de gösterilen özgün Petri ağı ile, bu Petri ağının örtüşmeli ayrıştırma ve genişleştirilmesi ile elde edilen ve Şekil 5.2’de gösterilen Petri ağı arasındaki içerme prensibini sağlayan Teorem 5.1’in ispatında gösterildiği şekilde oluşturulan, Q ve V matrisleri Çizelge 5.1 ve 5.2’de gösterilmiştir.

Örtüşmeli ayrıştırma ve genişleştirme işlemleri, ayrışık olarak alt Petri ağlarının elde edilmesine ve genişleştirilmiş Petri ağı elde edilerek, özgün Petri ağı ile arasındaki ilişkilerin ortaya çıkartılmasına olanak vermektedir. Sonuç olarak, örtüşmeli ayrıştırma ve genişleştirme işlemleri, Petri ağlarında bir dış-merkezli kontrolör tasarım yöntemi geliştirmek için kullanılmaktadır (bkz. Bölüm 6.1).

Aşağıda verilen lemma, Teorem 6.1’in ispatında kullanılmaktadır.

Lemma 5.4: Bölüm 5.1’de tanımlanan \mathcal{T} ve $\bar{\mathcal{T}}$ kümeleri, Tanım 4.2’de verilen sırasıyla, \mathcal{T} ve $\bar{\mathcal{T}}$ kümeleriyle aynıdır.

İspat:

- \tilde{t} geçişi, Bölüm 5.1’de tanımlanan \mathcal{T} kümesinin bir elemanı ise; (5.3)–(5.4) eşitliklerinden ve Q matrisinin seçiminden dolayı, $Q\tilde{A}\tilde{U}_{\tilde{t}} = 0$

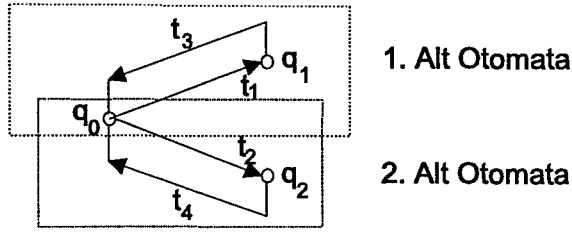
5.3. Otomata ve Dil Gösterimlerinde Örtüşmeli Ayırıştırma ve Genleştirme

Kesikli olay sistemlerin modelleme yöntemlerinden, Petri ağlarında (bkz. Bölüm 5.1) ve vektörel gösterimde [28], örtüşmeli ayırıştırma ve genleştirme işlemleri tanımlanmıştır. Bu bölümde ise, kesikli olay sistemlerin diğer modelleme yöntemlerinden, otomata ve dil gösterimleri için örtüşmeli ayırıştırma ve genleştirme işlemleri incelenecektir.

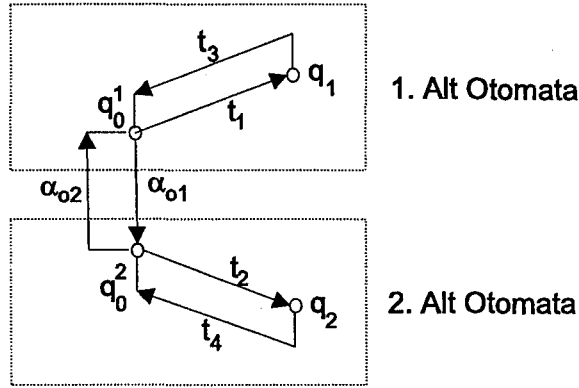
Otomata gösteriminde örtüşmeli ayırıştırma işlemi, otomata gösteriminde durumlar üzerinden yapılmaktadır. Örtüşmeli alt otomatalar arasındaki bağlantı yalnızca örtüşmeli kısım üzerinde olacak şekilde olmalıdır. Yani, bir alt otomatadaki durum ile bir diğer alt otomatadaki durum arasında (örtüşmeli kısımdaki bağlantılar hariç) bağlantı olmamalıdır. Bu özelliğe göre belirlenen örtüşmeli kısımda bulunan durum(lar) ve olay(lar)a, sırasıyla ortak durum(lar) ve ortak olay(lar) denir. Örneğin Şekil 2.7'deki otomata gösterimi, Şekil 5.3'de gösterildiği şekilde iki alt otomatadan meydana gelmektedir.

Otomata gösteriminde, ortak durumların dahil oldukları alt otomatalarda tekrarlanmalarıyla yeni durumlar elde edilir. Ortak durumların tekrarlanmasıyla elde edilen yeni durumlar ve geri kalan diğer durumların oluşturduğu küme $\tilde{Q} = \bigcup_{i=1}^K \tilde{Q}_i$ ile gösterilmektedir. \tilde{Q}_i , ayrışık olarak elde edilmiş i . alt otomatanın durumlarının oluşturduğu kümeyi ve K alt otomata sayısını göstermektedir. Şekil 5.3'deki otomata gösteriminde q_0 durumu, bir ortak durumdur. Bu durumun dahil olduğu alt otomatalarda tekrarlanması sonucu; $\tilde{Q}_1 = \{q_0^1, q_1\}$, $\tilde{Q}_2 = \{q_0^2, q_2\}$ ve $\tilde{Q} = \{q_0^1, q_0^2, q_1, q_2\}$ elde edilir (genleştirilmiş otomata için bkz. Şekil 5.4).

Özgün otomatada başlangıç durumu örtüşen kısımda bulunuyor ise, genleştirilmiş otomatada tekrarlanan durumlarından biri seçilerek genleştirilmiş otomatanın başlangıç durumu olarak alınır. Eğer özgün otomatada başlangıç durumu örtüşen kısımda bulunmuyor ise genleştirilmiş otomatada da aynı durum başlangıç durumu olarak alınır. Genleştirilmiş otomatada başlangıç durumu \tilde{q}_0 olarak gösterilir. Şekil 5.4'de gösterilen genleştirilmiş otomatada başlangıç durumu ise q_0^1 veya q_0^2 durumlarından biri seçilmelidir. q_0^1 durumu,



Şekil 5.3. Şekil 2.7'deki otomatanın örtüşmeli ayrıştırma uygulanmış hali



Şekil 5.4. Şekil 5.3'deki otomatanın genişletilmiş hali

başlangıç durumu olarak seçilmiş olsun, $\tilde{q}_0 = q_0^1$.

Tekrarlanmış durumlar arasındaki bağlantının sağlanması için ise, yeni olaylar ilave edilir. Bu olayların oluşturduğu küme $\bar{\Sigma}$ ile gösterilir. Örneğin, q_0^1 durumundan, q_0^2 durumuna ulaşılmasını sağlayan olay " α_{o1} " ve benzer şekilde q_0^2 durumundan, q_0^1 durumuna ulaşılmasını sağlayan olay " α_{o2} " olacak şekilde bu iki olay ilave edilmiştir, $\bar{\Sigma} = \{\alpha_{o1}, \alpha_{o2}\}$. Örtüşen kısımda bulunan ortak olaylar, dahil oldukları alt otomatalarda tekrarlanırlar. Örtüşen kısımda bulunan bir t olayı, t^j , $j \in \{1, \dots, n\}$, şeklinde tekrarlanmış olaylar ile gösterilir (n , t olayının dahil olduğu alt otomata sayısıdır). $\hat{\Sigma} = \bigcup_{i=1}^K \Sigma_i$, Σ_i i . alt otomatanın olaylarının oluşturduğu kümeyi göstermektedir. Böylece, genişletilmiş otomatada olaylar kümesi, $\tilde{\Sigma} = \hat{\Sigma} \cup \bar{\Sigma}$ olarak elde edilir. Bölüm 2.2.1'de kullanılan notasyona benzer şekilde, $\tilde{\Sigma}^*$, $\tilde{\Sigma}$ kümesindeki olaylardan oluşan tüm sonlu dizilerin oluşturduğu kümedir (bu kümeye, ϵ ile gösterilen boş dizi de ilave edilmiştir). Şekil 5.4'deki genişletilmiş otomatada, $\Sigma_1 = \{t_1, t_2\}$, $\Sigma_2 = \{t_3, t_4\}$, $\hat{\Sigma} = \{t_1, t_2, t_3, t_4\}$, $\tilde{\Sigma} = \{t_1, t_2, t_3, t_4, \alpha_{o1}, \alpha_{o2}\}$,

$\tilde{\Sigma}^* = \{\epsilon, t_1, t_1 t_3, \alpha_{o1}, \alpha_{o1} t_2, \dots\}$ olmaktadır.

$\tilde{\gamma} : \tilde{\Sigma}^* \times \tilde{Q} \rightarrow \tilde{Q}$ kısmi fonksiyonu, γ kısmi fonksiyonundan yararlanılarak aşağıdaki şekilde tanımlanmaktadır.

- 1) q , özgün otomatada örtüşmeli kısımda bulunmayan bir durumu ve \tilde{q} , bu duruma geliştirilmiş otomatada karşılık gelen durumu, t , özgün otomatada örtüşmeli kısımda bulunmayan bir olayı ve \tilde{t} , bu olaya geliştirilmiş otomatada karşılık gelen olayı gösteriyorsa,

$$\tilde{\gamma}(\tilde{t}, \tilde{q}) = \gamma(t, q)$$

- 2) q , özgün otomatada örtüşmeli kısımda bulunan bir durumu ve \tilde{q} , bu durumun geliştirilmiş otomatada tekrarlanmış durumlarından birini, t , özgün otomatada örtüşmeli kısımda bulunan bir olayı ve \tilde{t} , bu olayın geliştirilmiş otomatada tekrarlanmış olaylarından birini gösteriyorsa,

$$\tilde{\gamma}(\tilde{t}, \tilde{q}) = \begin{cases} \gamma(t, q), & \tilde{q} \text{ ile } \tilde{t} \text{ aynı alt otomatada bulunuyorlar ise} \\ \text{tanımlı değildir,} & \text{aksi durumda} \end{cases}$$

- 3) q , özgün otomatada örtüşmeli kısımda bulunan bir durumu ve \tilde{q} , bu durumun geliştirilmiş otomatada tekrarlanmış durumlarından birini, t , özgün otomatada örtüşmeli kısımda bulunmayan bir olayı ve \tilde{t} , bu olaya geliştirilmiş otomatada karşılık gelen olayı gösteriyorsa,

$$\tilde{\gamma}(\tilde{t}, \tilde{q}) = \begin{cases} \gamma(t, q), & \tilde{q} \text{ ile } \tilde{t} \text{ aynı alt otomatada bulunuyorlar ise} \\ \text{tanımlı değildir,} & \text{aksi durumda} \end{cases}$$

- 4) q , özgün otomatada örtüşmeli kısımda bulunmayan bir durumu ve \tilde{q} , bu duruma geliştirilmiş otomatada karşılık gelen durumu, t , özgün otomatada örtüşmeli kısımda bulunan bir olayı ve \tilde{t} , bu olayın geliştirilmiş otomatada tekrarlanmış olaylardan birini gösteriyorsa,

$$\tilde{\gamma}(\tilde{t}, \tilde{q}) = \begin{cases} \gamma(t, q), & \tilde{q} \text{ ile } \tilde{t} \text{ aynı alt otomatada bulunuyorlar ise} \\ \text{tanımlı değildir,} & \text{aksi durumda} \end{cases}$$

- 5) Geliştirilmiş otomatada bir $\tilde{t} \in \tilde{\Sigma}$ olayının olmasıyla, \tilde{t} olayının çıktısı olarak bağlı olduğu \tilde{q} durumundan, girdi olarak bağlı olduğu \tilde{q}' duruma

ulaşıldığından dolayı, $\tilde{\gamma}(\tilde{t}, \tilde{q}) = \tilde{q}'$ olarak tanımlanır. Burada \tilde{q} ve \tilde{q}' durumları, özgün otomatada örtüşmeli kısımda bulunan bir durumun tekrarlanmış durumlarıdır.

\tilde{s} olay dizisini oluşturan, ilk olaydan başlayarak son olaya kadar, her bir olay için, yukarıda verilen uygun durumlar kullanılarak, $\tilde{\gamma}(\tilde{s}, \tilde{q})$ ($\tilde{q} \in \tilde{Q}$) fonksiyonu tanımlanır.

Genleştirilmiş dil de, $\tilde{L} = \{\tilde{s} \in \tilde{\Sigma}^* \mid \tilde{\gamma}(\tilde{s}, \tilde{q}_0) \text{ tanımlı ise}\}$ olarak elde edilmektedir. Örneğin, Şekil 5.4'de verilen genleştirilmiş otomata gösterimi verilen sistemin dili, $\tilde{L} = \{\epsilon, t_1, t_1 t_3, \alpha_{o1}, \alpha_{o1} t_2, \alpha_{o1} t_2 t_4, t_1 t_3 \alpha_{o1} t_2, \dots\}$ şeklinde ifade edilir.

Örnek 5.1:

Bir kesikli olay sistemin, otomata gösterimi Şekil 5.5'de verilmiştir. Burada, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $\Sigma = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$, $\Sigma^* = \{\epsilon, \dots, t_1, t_1 t_2, \dots, t_5, \dots\}$ olarak elde edilmektedir. Otomata gösteriminin geçiş fonksiyonu aşağıda tanımlanmıştır.

$$\gamma(t_1, q_0) = q_1 \quad \gamma(t_5, q_0) = q_4 \quad \gamma(t_2, q_1) = q_2 \quad \gamma(t_3, q_2) = q_0$$

$$\gamma(t_4, q_2) = q_3 \quad \gamma(t_6, q_4) = q_5 \quad \gamma(t_7, q_5) = q_0 \quad \gamma(t_8, q_5) = q_1$$

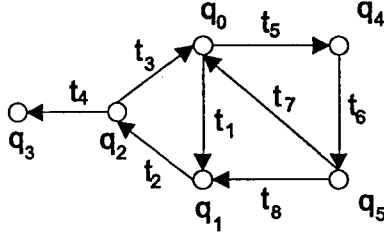
Birden fazla olaydan meydana gelen olay dizileri için yukarıdaki tanımlardan faydalanılarak,

$$\gamma(t_{i_1} t_{i_2} \dots t_{i_k}, q) = \gamma(t_{i_1}, \gamma(t_{i_2}, \dots, \gamma(t_{i_k}, q) \dots))$$

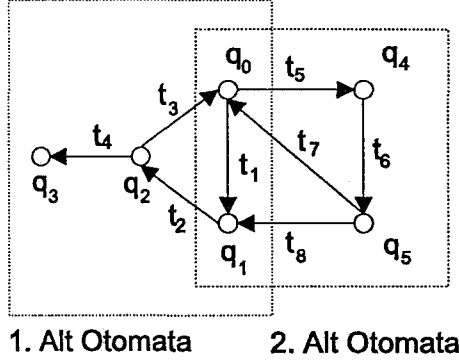
olarak tanımlanır ($t_{i_1}, t_{i_2}, \dots, t_{i_k} \in \Sigma$). Buradan yararlanılarak bu otomata gösterimi için dil, $L = \{s \in \Sigma^* \mid \gamma(s, q_0) \text{ tanımlı ise}\}$ olarak elde edilir.

Şekil 5.5'de verilen otomata gösteriminde örtüşmeli ayrıştırma uygulandığında, örtüşen iki alt otomata elde edilir (bkz. Şekil 5.6). Burada, q_0 ve q_1 durumları ile t_1 olayı, örtüşmeli kısımda bulunmaktadır.

Şekil 5.5'de verilen otomata gösteriminde örtüşmeli ayrıştırma uygulandıktan sonra, genleştirme işlemi yapıldığında, Şekil 5.7'de verilen otomata elde edilir. Şekil 5.6'da örtüşmeli kısımda bulunan, q_0 ve q_1 durumları ile t_1



Şekil 5.5. Örnek bir otomata



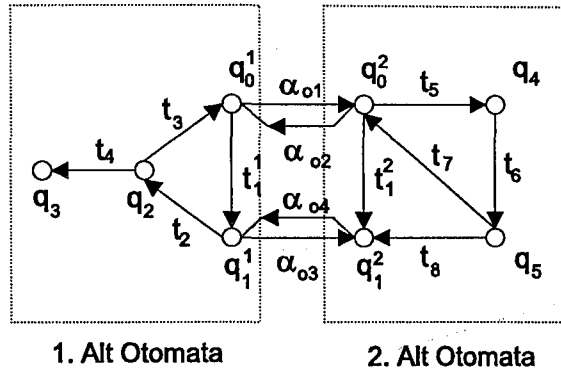
Şekil 5.6. Şekil 5.5'deki otomatanın örtüşmeli ayrıştırma uygulanmış hali

olayı, 1. ve 2. alt otomatalarda tekrarlanırlar. $\bar{Q}_1 = \{q_0^1, q_1^1, q_2, q_3\}$ ve $\bar{Q}_2 = \{q_0^2, q_1^2, q_4, q_5\}$ kümelerinden, $\tilde{Q} = \bar{Q}_1 \cup \bar{Q}_2$ elde edilir. $\Sigma_1 = \{t_1^1, t_2, t_3, t_4\}$ ve $\Sigma_2 = \{t_1^1, t_5, t_6, t_7, t_8\}$ kümelerinden, $\hat{\Sigma} = \Sigma_1 \cup \Sigma_2$ elde edilir. q_0^1 ve q_0^2 durumları arasında bağlantıyı sağlamak için ilave edilen olaylar $(\alpha_{o1}, \alpha_{o2})$ ile q_1^1 ve q_1^2 durumları arasında bağlantıyı sağlamak için ilave edilen olaylardan $(\alpha_{o3}, \alpha_{o4})$, $\bar{\Sigma} = \{\alpha_{o1}, \alpha_{o2}, \alpha_{o3}, \alpha_{o4}\}$ elde edilir (bkz. Şekil 5.7). Genleştirilmiş otomata için, $\tilde{\Sigma} = \hat{\Sigma} \cup \bar{\Sigma}$ ve $\tilde{\Sigma}^* = \{\epsilon, \alpha_{o1}, \dots, t_1^1, t_1^1 \alpha_{o3}, \dots\}$ bulunur. Genleştirilmiş otomatanın geçiş fonksiyonu aşağıda tanımlanmıştır.

$$\begin{aligned} \tilde{\gamma}(t_1^1, q_0^1) &= q_1^1 & \tilde{\gamma}(t_1^2, q_0^2) &= q_1^2 & \tilde{\gamma}(t_5, q_0^2) &= q_4 & \tilde{\gamma}(t_2, q_1^1) &= q_2 \\ \tilde{\gamma}(t_3, q_2) &= q_0^1 & \tilde{\gamma}(t_4, q_2) &= q_3 & \tilde{\gamma}(t_6, q_4) &= q_5 & \tilde{\gamma}(t_7, q_5) &= q_0^2 \\ \tilde{\gamma}(t_8, q_5) &= q_1^2 & \tilde{\gamma}(\alpha_{o1}, q_0^1) &= q_0^2 & \tilde{\gamma}(\alpha_{o2}, q_0^2) &= q_0^1 & \tilde{\gamma}(\alpha_{o3}, q_1^1) &= q_1^2 \\ \tilde{\gamma}(\alpha_{o4}, q_1^2) &= q_1^1 & & & & & & \end{aligned}$$

Birden fazla olaydan meydana gelen olay dizileri için yukarıdaki tanımlardan faydalanılarak,

$$\tilde{\gamma}(\tilde{t}_{i_1} \tilde{t}_{i_2} \dots \tilde{t}_{i_k}, \tilde{q}) = \tilde{\gamma}(\tilde{t}_{i_1}, \tilde{\gamma}(\tilde{t}_{i_2}, \dots, \tilde{\gamma}(\tilde{t}_{i_k}, \tilde{q}) \dots))$$



Şekil 5.7. Şekil 5.6'daki otomatanın genişletilmiş hali

olarak tanımlanır ($\tilde{t}_{i_1}, \tilde{t}_{i_2}, \dots, \tilde{t}_{i_k} \in \tilde{\Sigma}$). Buradan yararlanılarak genişletilmiş dil, $\tilde{L} = \{\tilde{s} \in \tilde{\Sigma}^* \mid \tilde{\gamma}(\tilde{s}, q_0^1) \text{ tanımlı ise}\}$ bulunur.

5.4. Genleştirilmiş Otomata ve Dil ile Özgün Otomata ve Dil Gösterimleri Arasındaki Bazı İlişkiler

Burada, Bölüm 5.3'de bahsedildiği şekilde elde edilen genişletilmiş otomata ve dil ile özgün otomata ve dil arasındaki ilişkiler incelenecektir.

Özgün otomata veya dil gösterimi ile modellenmiş, \mathcal{S} sistemini ve bu sisteme örtüşmeli ayrıştırma ve genişletirme uygulanmasıyla elde edilen sistemi, $\tilde{\mathcal{S}}$, ele alalım.

\mathcal{S} sisteminin, durumların kümesini Q , olayların kümesini Σ , Σ kümesinin elemanlarından oluşan tüm sonlu dizilerin kümesi Σ^* , geçiş fonksiyonunu γ ve tanımlanan dili de $L = \{s \in \Sigma^* \mid \gamma(s, q_0) \text{ tanımlı ise}\}$ ile gösterelim.

$\tilde{\mathcal{S}}$ sisteminin, durumların kümesini \tilde{Q} , olayların kümesini $\tilde{\Sigma}$, $\tilde{\Sigma}$ kümesinin elemanlarından oluşan tüm sonlu dizilerin kümesi $\tilde{\Sigma}^*$, geçiş fonksiyonunu $\tilde{\gamma}$ ve tanımlanan dili de $\tilde{L} = \{\tilde{s} \in \tilde{\Sigma}^* \mid \tilde{\gamma}(\tilde{s}, \tilde{q}_0) \text{ tanımlı ise}\}$ ile gösterelim.

\mathcal{S} ve $\tilde{\mathcal{S}}$ sistemlerinde tanımlanan olay dizileri arasındaki ilişkiler Lemma 5.5 ve 5.6'da verilecektir. $\mathcal{E}(q)$, \mathcal{S} 'deki q durumuna, $\tilde{\mathcal{S}}$ 'da karşılık gelen durumların oluşturduğu kümeyi, $\mathcal{E}(t)$, \mathcal{S} 'deki t olayına, $\tilde{\mathcal{S}}$ 'da karşılık gelen

olayların oluşturduğu kümeyi göstermektedir. $\mathcal{E}^{-1} : \tilde{\Sigma} \rightarrow \Sigma$ dönüşümü $t = \mathcal{E}^{-1}(\tilde{t}) \iff \tilde{t} \in \mathcal{E}(t)$ olarak tanımlanmaktadır. Ayrıca $\Theta(\tilde{q})$, $\tilde{\mathcal{S}}$ 'da \tilde{q} durumuna, \mathcal{S} 'de karşılık gelen durumu göstermektedir. $\Theta : \tilde{Q} \rightarrow Q$ dönüşümü aşağıdaki şekilde tanımlanmaktadır.

- \tilde{q} , \mathcal{S} 'deki q ortak durumunun, $\tilde{\mathcal{S}}$ 'da tekrarlanmış durumlarından birisi ise, $\Theta(\tilde{q}) = q$ olarak seçilir.
- \tilde{q} , tekrarlanmış bir durum değilse, $\Theta(\tilde{q}) = \tilde{q}$ olarak seçilir.

Lemma 5.5: $\tilde{\mathcal{S}}$ 'da $\tilde{q} \in \tilde{Q}$ ve $\tilde{q}' = \tilde{\gamma}(\tilde{s}, \tilde{q}) \in \tilde{Q}$ olsun. Bu durumda \mathcal{S} 'de $q' = \gamma(s, q)$ şartını sağlayan bir s olay dizisi vardır. Burada q ve q' ,

$$q = \Theta(\tilde{q}) \tag{5.14}$$

ve

$$q' = \Theta(\tilde{q}') \tag{5.15}$$

eşitliklerini sağlayan durumlardır.

İspat: \tilde{s} olay dizisini oluşturan, ilk olaydan başlayarak son olaya kadar, her bir olay için, aşağıda verilen uygun durumlar kullanılarak elde edilen olaylardan, s dizisi oluşturulur. \tilde{s} olay dizisinde ele alınan olay, \tilde{t} ,

- i) $\tilde{t} \in \tilde{\Sigma}$ ise, \mathcal{S} 'de bu olaya karşılık gelen bir olay alınmaz.
- ii) $\tilde{t} \in \hat{\Sigma}$ ise, $t = \mathcal{E}^{-1}(\tilde{t})$ olayı alınır.

Elde edilen s dizisinin, bir olay dizisi olduğu (s dizisindeki her bir olayın, sırası geldiğinde olabilirliği) ve \mathcal{S} 'de kullanıldığında (5.15) eşitliğinin sağlandığı aşağıda gösterilecektir.

$\tilde{\mathcal{S}}$ 'da, $\tilde{q} \in \tilde{Q}$ olmak üzere, $\tilde{q}' = \tilde{\gamma}(\tilde{s}, \tilde{q}) \in \tilde{Q}$ durumunun elde edilmesini sağlayan olay dizisi, $\tilde{s} = \tilde{t}$ olsun.

- a) $\tilde{t} \in \tilde{\Sigma}$ ise; $\tilde{\mathcal{S}}$ 'da, tekrarlanmış bir durumdan sonra \tilde{t} olayı olmasıyla diğer bir tekrarlanmış durum meydana gelmektedir. Buradan ve Θ dönüşümünden dolayı,

$$\Theta(\tilde{q}) = \Theta(\tilde{\gamma}(\tilde{t}, \tilde{q})) \quad (5.16)$$

olmaktadır. \mathcal{S}' 'de \tilde{t} olayına karşılık hiç bir olay gelmediğinden, $q' = q$ olmaktadır. Buradan ve denklem (5.14)'den dolayı,

$$q' = \Theta(\tilde{q}) \quad (5.17)$$

olmaktadır. $\tilde{q}' = \tilde{\gamma}(\tilde{s}, \tilde{q})$ 'da, denklem (5.16) ve (5.17) kullanılarak, (5.15) eşitliği elde edilir.

- b) $\tilde{t} \in \bar{\Sigma}$ ise; \tilde{t} olabilir bir olay olduğundan, \tilde{t} olayının çıktığı olarak bağlı olduğu durum, \tilde{q} durumudur. Bölüm 5.3'de verilen örtüşmeli ayrıştırma ve genişletme işlemlerinden ve Θ dönüşümünün tanımından dolayı, $t = \mathcal{E}^{-1}(\tilde{t})$ olayının çıktığı olarak bağlı olduğu durum, $q = \Theta(\tilde{q})$ durumudur. Bu nedenle t olayı, olabilir bir olaydır.

$t = \mathcal{E}^{-1}(\tilde{t})$ olabilir bir olay olduğundan, $q' = \gamma(t, q)$ elde edilmektedir. \tilde{t} olayının olmasıyla da $\tilde{q}' = \tilde{\gamma}(\tilde{t}, \tilde{q})$ meydana geldiği bilinmektedir. \mathcal{S}' 'nin olayları ve durumları ile $\tilde{\mathcal{S}}$ 'nin olayları ve durumları arasındaki ilişkiden dolayı, \tilde{q}' , q' 'nün tekrarlanmış durumlarından biridir ($\gamma(t, q) = \Theta(\tilde{\gamma}(\tilde{t}, \tilde{q}))$). Dolayısıyla, (5.15) eşitliği sağlanmaktadır.

$\tilde{\mathcal{S}}$ 'da, $\tilde{q} \in \tilde{Q}$ olmak üzere, $\tilde{q}' = \tilde{\gamma}(\tilde{s}, \tilde{q}) \in \tilde{Q}$ durumunun elde edilmesini sağlayan olay dizisi, $\tilde{s} = \tilde{t}_{k_1} \tilde{t}_{k_2} \dots \tilde{t}_{k_h}$ olsun. \tilde{q}' durumu, $\tilde{q}' = \tilde{\gamma}(\tilde{t}_{k_h}, (\dots, \tilde{\gamma}(\tilde{t}_{k_1}, \tilde{q})))$ olarak elde edilmektedir.

- A) $\tilde{\gamma}(\tilde{t}_{k_1}, \tilde{q})$ durumu, \tilde{q}_{k_1} ile gösterilsin. \tilde{t}_{k_1} olayı için (i). veya (ii). şıklardan uygun olanı alınarak, yukarıda verilen işlemler yapıldığında, $q_{k_1} = \Theta(\tilde{q}_{k_1})$ eşitliği sağlanmaktadır.
- B) \tilde{t}_{k_j} olayının olmasıyla, meydana gelen \tilde{q}_{k_j} durumu ile q_{k_j} durumu arasında, $q_{k_j} = \Theta(\tilde{q}_{k_j})$ eşitliği sağlandığı kabul edilsin. $\tilde{t}_{k_{j+1}}$ olayı için (i). veya (ii). şıklardan uygun olanı alınarak, yukarıda verilen işlemler yapıldığında, $q_{k_{j+1}} = \Theta(\tilde{q}_{k_{j+1}})$ eşitliği sağlanmaktadır.

(A) ve (B) şıklarından endüksiyon yöntemiyle istenilen sonuç elde edilmektedir. \square

Lemma 5.6: \mathcal{S} 'de herhangi bir $q \in Q$ durumu için, s olay dizisi $q = \gamma(s, q_0)$ eşitliğini sağlayan dizi olsun. Bu durumda $\tilde{\mathcal{S}}$ 'da $\tilde{q} = \tilde{\gamma}(\tilde{s}, \tilde{q}_0)$ eşitliğini sağlayan bir \tilde{s} olay dizisi vardır. Burada \tilde{q} , (5.14) eşitliğini sağlayan bir durumdur.

İspat: \tilde{q}_0 başlangıç durumunun seçiminden (bkz. Bölüm 5.3) dolayı, $q_0 = \Theta(\tilde{q}_0)$ olmaktadır. \mathcal{S} 'de, $q = \gamma(s, q_0)$ durumunun elde edilmesini sağlayan olay dizisi $s = t$ olsun. t olayına karşılık, aşağıda verilen uygun durumlardan elde edilen olaylardan, \tilde{s} olay dizisi oluşturulur.

- i) Herhangi bir $\tilde{t} \in \mathcal{E}(t)$ olayı için, $\tilde{\mathcal{S}}$ 'da \tilde{t} olayının çıktı olarak bağlı olduğu durum \tilde{q}_0 ise; t olayına karşılık $\tilde{s} = \tilde{t}$ olay dizisi alınır.
- ii) Hiç bir $\tilde{t} \in \mathcal{E}(t)$ olayı için, $\tilde{\mathcal{S}}$ 'da $\tilde{t} \in \mathcal{E}(t)$ olayının çıktı olarak bağlı olduğu durum \tilde{q}_0 değil ise; herhangi bir $\tilde{t} \in \mathcal{E}(t)$ geçişi için, bu şartın sağlanması için aşağıdaki şekilde bir \tilde{t}^+ olayı tanımlanabilir.

Bölüm 5.3'de verilen örtüşmeli ayrıştırma ve genişletme işlemlerinden dolayı, $\tilde{t} \in \mathcal{E}(t)$ olayının çıktı olarak bağlı olduğu bir $\tilde{q}_{0r} \in \mathcal{E}(q_0)$ durumu vardır. \tilde{q}_{0r} ile $\tilde{t} \in \mathcal{E}(t)$ olayının çıktı olarak bağlı olduğu \tilde{q}_0 arasında bağlantıyı sağlayan ve $\bar{\Sigma}$ kümesinin elemanı olan \tilde{t}^+ olayı olduktan sonra, \tilde{q}_0 durumu meydana gelir. t olayına karşılık $\tilde{s} = \tilde{t}^+\tilde{t}$ olay dizisi alınır.

$q = \gamma(t, q_0)$ ile $\tilde{q} = \tilde{\gamma}(\tilde{s}, \tilde{q}_0)$ 'nun, (5.14) eşitliğini sağladığı gösterilecektir.

- t olayına karşılık $\tilde{s} = \tilde{t} \in \mathcal{E}(t)$ olay dizisi geliyorsa; $q = \gamma(t, q_0)$ ve $\tilde{q} = \tilde{\gamma}(\tilde{t}, \tilde{q}_0)$ durumları elde edilir. \mathcal{S} 'nin olayları ve durumları ile $\tilde{\mathcal{S}}$ 'nin olayları ve durumları arasındaki ilişkiden dolayı, \tilde{q} , q 'nun tekrarlanmış durumlarından biridir ($\gamma(t, q_0) = \Theta(\tilde{\gamma}(\tilde{t}, \tilde{q}_0))$). Dolayısıyla, (5.14) eşitliği sağlanmaktadır.
- t olayına karşılık $\tilde{s} = \tilde{t}^+\tilde{t}$ ($\tilde{t} \in \mathcal{E}(t)$) olay dizisi geliyorsa; $\tilde{s} = \tilde{t}^+\tilde{t}$ olay dizisi kullanılarak, $\tilde{q} = \tilde{\gamma}(\tilde{t}, \tilde{\gamma}(\tilde{t}^+, \tilde{q}_0))$ olmaktadır. $\tilde{\gamma}(\tilde{t}^+, \tilde{q}_0)$ durumu,

\tilde{q}_{0n} ile gösterilsin. $\tilde{t}^+ \in \bar{\Sigma}$ olduğundan ve Lemma 5.5'in ispatında verilen (a) şikkından dolayı, $\Theta(\tilde{q}_0) = \Theta(\tilde{q}_{0n})$ elde edilir. \mathcal{S} 'nin olayları ve durumları ile $\tilde{\mathcal{S}}$ 'nin olayları ve durumları arasındaki ilişkiden dolayı, $\tilde{q} = \tilde{\gamma}(\tilde{t}, \tilde{q}_{0n})$, $q = \gamma(t, q_0)$ 'in tekrarlanmış durumlarından biridir ($\gamma(t, q_0) = \Theta(\tilde{\gamma}(\tilde{t}, \tilde{q}_{0n}))$). Dolayısıyla, (5.14) eşitliği sağlanmaktadır.

$q = \gamma(s, q_0)$ durumunun elde edilmesini sağlayan olay dizisi, $s = t_{k_1} t_{k_2} \dots t_{k_h}$ olsun. q durumu, $q = \gamma(t_{k_h}, (\dots, \gamma(t_{k_1}, q_0)))$ olarak elde edilmektedir.

- A) $\gamma(t_{k_1}, q_0)$ durumu, q_{k_1} ile gösterilsin. t_{k_1} olayına karşılık, (i). veya (ii). şıklardan biri kullanılarak, \tilde{s}_{k_1} olay dizisi gelir. Yukarıda verilen işlemler yapıldığında, $q_{k_1} = \Theta(\tilde{q}_{k_1})$ eşitliği sağlanmaktadır.
- B) t_{k_j} olayının olmasıyla, meydana gelen q_{k_j} durumu ile \tilde{q}_{k_j} durumu arasında, $q_{k_j} = \Theta(\tilde{q}_{k_j})$ eşitliği sağlandığı kabul edilsin. $t_{k_{j+1}}$ olayına karşılık, (i). veya (ii). şıklardan biri kullanılarak, $\tilde{s}_{k_{j+1}}$ olay dizisi gelir. Yukarıda verilen işlemler yapıldığında, $q_{k_{j+1}} = \Theta(\tilde{q}_{k_{j+1}})$ eşitliği sağlanmaktadır.

(A) ve (B) şıklarından endüksiyon yöntemiyle istenilen sonuç elde edilmektedir. \square

\mathcal{S} 'nin genişletilmiş hali olan $\tilde{\mathcal{S}}$ 'nin \mathcal{S} 'yi içerdiği aşağıda verilen teoremde gösterilecektir.

Teorem 5.2: $\tilde{\mathcal{S}}$, \mathcal{S} 'yi içerir.

İspat:

$H : \mathcal{Q} \rightarrow \tilde{\mathcal{Q}}$ dönüşümü aşağıdaki şekilde tanımlanmaktadır.

- q , \mathcal{S} 'de ortak durum ise, \tilde{q} $\tilde{\mathcal{S}}$ 'de q durumunun tekrarlanmış durumlarından, herhangi biri olmak üzere, $H(q) = \tilde{q}$ olarak seçilir. q_0 , başlangıç durumu \mathcal{S} 'de bir ortak durum ise, $H(q) = \tilde{q}_0$ olarak seçilmelidir.
- q , \mathcal{S} 'de bir ortak durum değilse, $H(q) = q$ olarak seçilir.

H ve Θ dönüşümleri Tanım 4.3'ün (i). ve (ii). şıklarını sağlayacak şekilde seçilmiştir.

Lemma 5.5'den dolayı, $\tilde{q} \in \tilde{Q}$ ve $\tilde{q}' = \tilde{\gamma}(\tilde{s}, \tilde{q})$ ise (5.14) ve (5.15) eşitliklerini sağlayan $q \in Q$ ve $q' = \gamma(s, q)$ vardır. Θ dönüşümünün tanımından dolayı, $q = \Theta(\tilde{q})$ ve $q' = \Theta(\tilde{q}')$ olmaktadır. Bu da, $q = \Theta(\tilde{q})$ durumundan sonra s olay dizisi kullanılarak $q' = \Theta(\tilde{q}') = \gamma(s, q)$ olan bir duruma ulaşıldığını göstermektedir. Dolayısıyla, Tanım 4.3'ün (iii). şıkkı sağlanmış olur.

Lemma 5.6'dan dolayı, $q = \gamma(s, q_0)$ ise denklem (5.14) eşitliğini sağlayan $\tilde{q} = \tilde{\gamma}(\tilde{s}, \tilde{q}_0)$ vardır. Θ dönüşümünün tanımından dolayı (5.15) eşitliği sağlanmaktadır. Dolayısıyla, Tanım 4.3'ün (iv). şıkkı sağlanmış olur.

Böylece $\tilde{\mathcal{S}}$, \mathcal{S} 'yi içerir. □

Örneğin, Şekil 5.3 ve 5.4'de verilen otomataların durumları için, Θ ve H dönüşümleri; $\Theta(q_0^1) = q_0$, $\Theta(q_0^2) = q_0$, $\Theta(q_1) = q_1$ ve $\Theta(q_2) = q_2$. $H(q_0) = q_0^1$, $H(q_1) = q_1$ ve $H(q_2) = q_2$ olarak alınmıştır.

5.5. Değişik Gösterimler İçin Genleştirme İşleminin Karşılaştırılması

Bu bölümde Petri ağları, otomata ve dil gösterimleri için ve [28]'de de vektörel gösterim için, örtüşmeli ayrıştırma ve genleştirme işlemlerinin, nasıl yapılabileceği verilmiştir. Sonlu durumlu bir kesikli olay sistemi, Petri ağı, otomata ve dil gösterimleri ve vektörel gösterim ile modellenebilir. Bu gösterimler arasında bire bir ilişki olmasına rağmen (ayrıntılar için bkz. Bölüm 2.2), Petri ağlarında örtüşmeli ayrıştırma ve genleştirme yerler üzerinden, otomata ve dil gösterimlerinde durumlar üzerinden ve vektörel gösterimde ise durum vektörlerinin belli elemanları üzerinden yapılmaktadır. Aynı sistemin değişik gösterimleri üzerinde, bu çalışmada konu edilen yöntemler kullanılarak örtüşmeli ayrıştırma ve genleştirme işlemleri uygulandığında, ortaya çıkan genleştirilmiş sistemler ve alt sistemler hangi gösterimin kullanıldığına bağlı olarak birbirlerinden farklı olmaktadır.

6. DIŐMERKEZLİ KONTROLÖR TASARIMI

Bu bölümde, Petri ağı ve otomata gösterimleri için birer dışmerkezli kontrolör tasarımı verilecektir. Her iki dışmerkezli kontrolör tasarımıda, örtüşmeli ayrıştırma ve genleştirmeye dayalı olarak gerçekleştirilecektir. Bölüm 5.1'de verildiği şekilde Petri ağları için, Bölüm 5.3'de verildiği şekilde otomata ve dil gösterimleri için, örtüşmeli ayrıştırma ve genleştirme işlemleri yapılmıştır.

6.1. Petri Ağlarında Dışmerkezli Bir Kontrolör Tasarım Yöntemi

Dışmerkezli kontrolör tasarımı için sırasıyla; elde edilen her bir ayrışık alt Petri ağı için kontrolör tasarlanmakta, tasarlanan bu kontrolörler kullanılarak, genleştirilmiş Petri ağı için kontrolör tasarlanmakta ve son olarak genleştirilmiş Petri ağı için tasarlanan kontrolör büzülerek özgün Petri ağı için bir kontrolör elde edilmektedir.

6.1.1. Alt Petri ağları için kontrolör tasarımı

Örtüşmeli ayrıştırma ve genleştirme işlemleri sonucu elde edilen her bir alt Petri ağı için, tek başına bir Petri ağı gibi Bölüm 3.1'de verildiği şekilde sistem çıkmazını önleyici bir kontrolör tasarlanabilir. Bu kontrolörün tasarımı sırasında, alt Petri ağlarının başlangıç işaretlemelerinden yararlanılarak, ulaşılabilirlik kümeleri oluşturulduğundan, alt Petri ağlarının, tek başlarına çalışabilecek şekilde, başlangıç işaretlemelerine sahip olmaları gerekmektedir, yani $m_{0k} \notin \mathcal{Y}(G_k)$, $\forall k \in \{1, 2, \dots, S\}$ olmalıdır.

k. alt Petri ağı için kontrolör tasarımı aşağıdaki şekilde yapılmaktadır.

- *k.* alt Petri ağı için başlangıç işaretlemesi belirlenmelidir. Teorem 5.1'in ispatında verildiği şekilde V matrisinin seçimi, *k.* alt Petri ağının tek başına çalışabileceği şekilde olmalıdır ($m_{0k} \notin \mathcal{Y}(G_k)$). Seçilen V matrisi, V_k ile gösterilir.

$$V_k m_0 = [\dots | \dots | m_{0k}^T | \dots]^T$$

Örneğin Şekil 5.2'de verilen genleştirilmiş Petri ağında; 1. alt Petri

Çizelge 6.1. V_2 matrisi

$$V_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

ağı için başlangıç koşulu, $V_1 m_0 = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 4 \ | \ 0 \ 0 \ 0 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$, $m_{10} = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 4]^T$ ve 2. alt Petri ağı için başlangıç koşulu, $V_2 m_0 = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ | \ 0 \ 1 \ 4 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$, $m_{20} = [0 \ 1 \ 4 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ olmaktadır. Burada, V_1 matrisi, Çizelge 5.1'de verilen V matrisi ile aynıdır. V_2 ise Çizelge 6.1'de verilmiştir.

- k . alt Petri ağının başlangıç koşulu seçildikten sonra, Bölüm 3.1'de verilen tasarım kullanılarak; k . alt Petri ağı, $G_k(P_k, T_k, N_k, O_k, m_{k0})$ ile gösterilmek üzere, \mathcal{D}_k kümesi ve C_k kontrolör matrisi bulunur ve

k . alt Petri ağ kontrolörü ile birlikte $G_{ck}(P_k, T_k, N_k, O_k, m_{k0}, C_k)$ olarak gösterilir.

Genleştirilmiş Petri ağının tüm alt Petri ağları için yukarıda verildiği şekilde kontrolör tasarlanır.

Şekil 5.2'deki genleştirilmiş Petri ağı, iki alt Petri ağından meydana gelmektedir ($G_i(P_i, T_i, I_i, O_i, m_{i0})$, $i = 1, 2$). 1. ve 2. alt Petri ağının girdi ve çıktı matrisleri Bölüm 5.1'de verilmiştir. Alt Petri ağlarının yerlerinin ve geçişlerinin kümeleri ile başlangıç işaretlemeleri, $P_1 = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_{8a}, r_a, d_a\}$, $T_1 = \{t_1, t_2, t_3, t_4, t_5, t_6^1\}$, $m_{10} = [2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 4]^T$, $P_2 = \{p_{8b}, r_b, d_b, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$, $T_2 = \{t_6^2, t_7, t_8, t_9, t_{10}, t_{11}\}$ ve $m_{20} = [0 \ 1 \ 4 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ şeklindedir. Bu iki alt Petri ağı için aşağıdaki şekilde kontrolör tasarlanmaktadır:

- 1. alt Petri ağının incelenmesi sonucu sistem çıkmazı meydana geldiği görülmektedir. Bunu önlemek için, Bölüm 3.1'de anlatıldığı şekilde bir kontrolör tasarlanabilir. Öncelikle, sistem çıkmazına yol açabilecek işaretleme vektörlerinin bulunması gerekmektedir. Buradan,

$$M_{xy}^{11} = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ x \ 1 \ y]^T \ ((x, y) \in \{(0, 4), (1, 3), (2, 2), (3, 1)\}),$$

$$M^{12} = [1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0]^T \text{ ve } M^{13} = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0]^T$$

olarak bulunur, $\mathcal{D}_1 = \{M_{xy}^{11}, M^{12}, M^{13}\}$. M_{xy}^{11} işaretleme vektörlerinden sonra t_2 geçişinin ateşlenmesini önlemek için $c_1(M_{xy}^{11}, t_2) = 0$ alınır ve diğer geçişler Petri ağının topolojisi müsait ise ateşlenebilir, bunun için $c_1(M_{xy}^{11}, t_j) = 1$, $j \in \{1, 3, 4, 5\}$ ve $c_1(M_{xy}^{11}, t_6^1) = 1$ alınır. Kontrolör matris formunda,

$$C_1(M_{xy}^{11}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde elde edilir. M^{12} işaretleme vektöründen sonra t_4 geçişinin ateşlenmesini önlemek için $c_1(M^{12}, t_4) = 0$ alınır ve diğer geçişler Petri

ağının topolojisi müsait ise ateşlenebilir, bunun için $c_1(M^{12}, t_j) = 1$, $j \in \{1, 2, 3, 5\}$ ve $c_1(M^{12}, t_6^1) = 1$ alınır. Kontrolör matris formunda,

$$C_1(M^{12}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde elde edilir. M^{13} işaretleme vektöründen sonra t_2 ve t_4 geçişlerinin ateşlenmesini önlemek için $c_1(M^{13}, t_2) = 0$ ve $c_1(M^{13}, t_4) = 0$ alınır ve diğer geçişler Petri ağının topolojisi müsait ise ateşlenebilir, bunun için $c_1(M^{13}, t_j) = 1$, $j \in \{1, 3, 5\}$ ve $c_1(M^{13}, t_6^1) = 1$ alınır. Kontrolör matris formunda,

$$C_1(M^{13}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde elde edilir. 1. alt Petri ağında, M_{xy}^{11} , M^{12} ve M^{13} işaretleme vektörleri hariç diğer işaretleme vektörleri için kontrolör matrisi birim matris olarak alınmaktadır ($C_1 = I$). Böylece 1. alt Petri ağı için sistem çıkmazını önleyecek bir kontrolör tasarlanmış olur ve $G_{c_1}(P_1, T_1, N_1, O_1, m_{10}, C_1)$ şeklinde gösterilir.

- 2. alt Petri ağının incelenmesi sonucu sistem çıkmazı meydana geldiği görülmektedir. Bunu önlemek için, Bölüm 3.1'de anlatıldığı şekilde bir kontrolör tasarlanabilir. Öncelikle, sistem çıkmazına yol açabilecek işaretleme vektörlerinin bulunması gerekmektedir. Buradan,

$$M_{xy}^{21} = [x \ 1 \ y \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T \ ((x, y) \in \{(0, 4), (1, 3), (2, 2), (3, 1)\}),$$

$$M^{22} = [4 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]^T \ \text{ve} \ M^{23} = [4 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T$$

olarak bulunur, $\mathcal{D}_2 = \{M_{xy}^{21}, M^{22}, M^{23}\}$. M_{xy}^{21} işaretleme vektörlerinden sonra t_8 geçişinin ateşlenmesini önlemek için $c_2(M_{xy}^{21}, t_8) = 0$ alınır ve diğer geçişler Petri ağının topolojisi müsait ise ateşlenebilir, bunun için $c_2(M_{xy}^{21}, t_j) = 1$, $j \in \{7, 9, 10, 11\}$ ve $c_2(M_{xy}^{21}, t_6^2) = 1$ alınır. Kontrolör matris formunda,

$$C_2(M_{xy}^{21}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde elde edilir. M^{22} işaretleme vektöründen sonra t_{10} geçişinin ateşlenmesini önlemek için $c_2(M^{22}, t_{10}) = 0$ alınır. Diğer geçişler Petri ağının topolojisi müsait ise ateşlenebilir, bunun için $c_2(M^{22}, t_j) = 1$, $j \in \{7, 8, 9, 11\}$ ve $c_2(M^{22}, t_6^2) = 1$ alınır. Kontrolör matris formunda,

$$C_2(M^{22}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde elde edilir. M^{23} işaretleme vektöründen sonra t_8 ve t_{10} geçişlerinin ateşlenmesini önlemek için $c_2(M^{23}, t_8) = 0$ ve $c_2(M^{23}, t_{10}) = 0$ alınır. Diğer geçişler Petri ağının topolojisi müsait ise ateşlenebilir, bunun için $c_2(M^{23}, t_j) = 1$, $j \in \{7, 9, 11\}$ ve $c_2(M^{23}, t_6^2) = 1$ alınır. Kontrolör matris formunda,

$$C_2(M^{23}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde elde edilir. 2. alt Petri ağında, M_{xy}^{21} , M^{22} ve M^{23} işaretleme vektörleri hariç diğer işaretleme vektörleri için kontrolör matrisi birim matris olarak alınmaktadır ($C_2 = I$). Böylece 2. alt Petri ağı için sistem çıkmazını önleyecek bir kontrolör tasarlanmış olur ve $G_{c2}(P_2, T_2, N_2, O_2, m_{20}, C_2)$ şeklinde gösterilir.

6.1.2. Genleştirilmiş Petri ağı için kontrolör tasarımı

Genleştirilmiş Petri ağında çalışma engelinin meydana gelmesini önleyecek kontrolör tasarımında, alt Petri ağları için tasarlanan kontrolörlerden yararlanılacaktır. Burada, her alt Petri ağı için, Bölüm 6.1.1'de anlatıldığı şekilde, sistem çıkmazının meydana gelmesini önleyen bir kontrolör tasarlandığı kabul edilmiştir.

Genleştirilmiş Petri ağının işaretleme vektörü, \tilde{M} , alt Petri ağlarının işaretleme vektörleri cinsinden, $\tilde{M} = [M_1^T \mid M_2^T \mid \dots \mid M_k^T \mid \dots \mid M_S^T]^T$ şeklinde gösterilebilir. Bu şekilde ifade edilen işaretleme vektörlerinden yararlanılarak, alt Petri ağlarının kontrolör matrisleri yeniden düzenlenebilir. k . alt Petri ağı için $C_k(\tilde{M}) \triangleq \alpha_k(M_k).C_k(M_k)$ olmaktadır. Burada,

$$\alpha_k(M_k) \triangleq \begin{cases} 1, & M_k \in R(G_{ck}, m_{k0}) \\ 0, & \text{diğer durumlarda} \end{cases}$$

şeklinde tanımlanmıştır.

Genleştirilmiş Petri ağında, alt Petri ağları arasında bağlantıyı sağlayan geçişler, belirti veya belirtileri bir alt Petri ağından, başka bir alt Petri ağına ulaşmasını sağlamaktadır. Bu nedenle, tasarlanan kontrolörde, bu geçişler müsait oldukları her durumda ateşlenmesine izin verilmektedir.

Dolayısıyla, geliştirilmiş Petri ağının kontrolör matrisi, köşegeninde alt Petri ağların kontrolör matrisleri ($C_i, i \in \{1, 2, \dots, S\}$) ile birim matrislerin bulunduğu bir matris olarak oluşturulmaktadır.

$$\tilde{C}(\tilde{M}) = \begin{bmatrix} C_1(\tilde{M}) & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & I_1 & \dots & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & C_k(\tilde{M}) & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & I_k & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & C_S(\tilde{M}) & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & I_S \end{bmatrix} \quad (6.1)$$

Burada, $I_l : |\bigcup_q T_l^{lq}| \times |\bigcup_q T_l^{lq}|$ ($q \neq l, q, l \in \{1, 2, \dots, S\}$) boyutlu birim matrisi ve 0 uygun boyutlu sıfır matrisini göstermektedir. $\tilde{C}(\tilde{M})$ matrisinin, herhangi bir $\tilde{t} \in \tilde{T}$ geçişi için aldığı değer $\tilde{c}(\tilde{M}, \tilde{t})$ ile gösterilir.

Böylece, tasarlanan kontrolörle birlikte geliştirilmiş Petri ağı, $\tilde{G}_c(\tilde{P}, \tilde{T}, \tilde{N}, \tilde{O}, \tilde{m}_0, \tilde{C})$ ile gösterilir ve tasarlanan kontrolörle birlikte geliştirilmiş Petri ağının denklemi ise,

$$\tilde{M}' = \tilde{M} + \tilde{A}\tilde{C}(\tilde{M})\tilde{U}_{\tilde{t}}, \tilde{t} \in \tilde{T} \quad (6.2)$$

şeklinde düzenlenir.

Teorem 6.1: Özgün Petri ağında $m_0 \notin \mathcal{Y}(G)$ ise, geliştirilmiş Petri ağı için tasarlanan kontrolör, (6.1), çalışma engelinin meydana gelmesini önler.

İspat: Bölüm 5.1'de tanımlanan \mathcal{T} ve \bar{T} kümeleri ile Tanım 4.2'de verilen, sırasıyla, \mathcal{T} ve \tilde{T} kümelerinin aynıdır (bkz. Lemma 5.4).

$\tilde{\mathcal{F}}$; tasarlanan kontrolörün izin vermediği işaretleme vektörlerinin oluşturduğu kümeyi,

$\tilde{\mathcal{G}}$; GPA'da ya hiç bir $\tilde{t} \in \tilde{T}$ geçişinin ateşlenemediği, ya da yalnızca $\tilde{t} \in \mathcal{T}$

geçişlerinden oluşan ateşleme dizilerin kullanılabilirdiği işaretleme vektörlerinin oluşturduğu kümeyi,

\mathcal{F}_j ; $j. (\forall j \in S)$ APA için tasarlanan kontrolörün izin vermediği işaretleme vektörlerinin oluşturduğu kümeyi

göstermektedir.

Özgün Petri ağında $m_0 \notin \mathcal{Y}(G)$ ise, en az bir tane g ateşleme dizisi kullanılarak, $M = \rho(g, m_0) \notin \mathcal{Y}(G)$ işaretleme vektörü elde edilebilir. Lemma 5.2'nin ispatından dolayı, g ateşleme dizisine karşılık, genişletilmiş Petri ağda \tilde{g} ateşleme dizisi elde edilmektedir. $\tilde{m}_0 = Vm_0$ başlangıç işaretlemesinden sonra, \tilde{g} ateşleme dizisi kullanıldığında $\tilde{M} = \tilde{\rho}(\tilde{g}, \tilde{m}_0) \neq \tilde{m}_0$ işaretleme vektörüne ulaşılmaktadır. Lemma 5.2'den dolayı, $M = \rho(g, m_0)$ ile $\tilde{M} = \tilde{\rho}(\tilde{g}, \tilde{m}_0)$ arasında, (5.1) eşitliği sağlanmaktadır. (5.1) eşitliği de Q matrisinin seçiminden dolayı, $M = Q\tilde{M}$ olmaktadır. Dolayısıyla, $\tilde{M} \notin \tilde{\mathcal{G}}$ ve $\tilde{m}_0 \notin \tilde{\mathcal{G}}$ olarak elde edilmektedir.

$\tilde{M}' \in \tilde{\mathcal{G}}$ olan bir işaretleme vektörü alalım. Bu işaretleme vektörü, $\tilde{\mathcal{G}}$ kümesinin elemanı olduğundan; ya hiç bir $\tilde{t} \in \tilde{T}$ geçişi ateşlenemiyordur ya da yalnızca $\tilde{t} \in \mathcal{T}$ geçişleri ateşlenebilmektedir.

D-1 $\tilde{M}' \in \tilde{\mathcal{G}}$ olmak üzere, \tilde{M}' işaretleme vektöründen sonra hiçbir $\tilde{t} \in \tilde{T}$ geçişi ateşlenemiyorsa: $\tilde{m}_0 \notin \tilde{\mathcal{G}}$ olduğundan dolayı, \tilde{M}' işaretleme vektörü, GPA için başlangıç işaretlemesi olamaz. Dolayısıyla, \tilde{M}' işaretleme vektörüne, başka bir işaretleme vektöründen sonra ateşlenen bir geçiş ile ulaşılmaktadır. Bunu da, $\tilde{M}' = \tilde{M} + \tilde{A}\tilde{U}_{\tilde{t}_e}$, $\tilde{t}_e \in \tilde{T}$ şeklinde gösterebiliriz. Bu \tilde{t}_e geçisi, \mathcal{T} kümesinin bir elemanı ise, GPA'nın yapısına bağlı olarak, $\tilde{t} \in \mathcal{T}$ bir geçiş her zaman ateşlenebilir. \tilde{t}_e geçişi ateşlendikten sonra hiç bir $\tilde{t} \in \tilde{T}$ geçişi ateşlenemediğinden, $\tilde{t}_e \in \bar{T}$ olmalıdır.

$\tilde{t}_e \in \bar{T}$ geçişinin hangi APA'na ait bir geçiş olduğu bilinmektedir, $\tilde{t}_e \in T_q$ ($q \in \{1, 2, \dots, S\}$) olsun. $\tilde{M}' = \tilde{M} + \tilde{A}\tilde{U}_{\tilde{t}_e}$ denkleminde $\tilde{t}_e \in T_q$ kullanılarak \tilde{M}' işaretleme vektörü elde edildiğinden, $\tilde{M} = [M_1^T \mid M_2^T \mid \dots \mid M_q^T \mid \dots \mid M_S^T]^T$ ve $\tilde{M}' = [M_1^T \mid M_2^T \mid \dots \mid M_q'^T \mid \dots \mid M_S^T]^T$ şeklinde gösterilebilir.

\tilde{M}' işaretleme vektöründen sonra hiç bir geçişin ateşlenemediği bilinmektedir, bu yüzden M'_q işaretleme vektöründen sonra da hiç bir geçiş ateşlenemez. Bu da, q . APA'da sistem çıkmazı meydana geldiğini göstermektedir. Dolayısıyla, $M_q \in R(G_{cq}, m_{q0})$ ise $M'_q \in \mathcal{F}_q$ olduğundan $c_q(M_q, t_e) = 0$ ve $\tilde{c}(\tilde{M}, t_e) = 0$ olur, öte yandan $M_q \notin R(G_{cq}, m_{q0})$ ise, $\alpha(M_q) = 0$ olacağından $\tilde{c}(\tilde{M}, t_e) = 0$ olur. Bu nedenle, $\tilde{M}' \in \tilde{\mathcal{F}}$ olur.

D-2 $\tilde{M}' \in \tilde{\mathcal{G}}$ olmak üzere, \tilde{M}' işaretleme vektöründen sonra yalnızca $\tilde{t} \in \mathcal{T}$ geçişler ateşlenebiliyorsa: $\tilde{M}' \in \tilde{\mathcal{G}}$ işaretleme vektöründen sonra, GPA'nın yapısına bağlı olarak, ateşlenebilen bütün $\tilde{t} \in \mathcal{T}$ geçişleri kullanılarak elde edilen işaretleme vektörleri de, $\tilde{\mathcal{G}}$ kümesinin elemanlarıdır. \tilde{m}_0 başlangıç işaretlemesinden, \tilde{M}' işaretleme vektörüne ulaşılırken ateşlenen en son $\tilde{t} \in \bar{\mathcal{T}}$ geçişine kadar olan geçişler kullanılarak, \tilde{m}_0 başlangıç işaretlemesinden \tilde{M}^* ile gösterilen bir işaretleme vektörüne ulaşılır. \tilde{M}' işaretleme vektörüne ulaşılırken ateşlenen en son $\tilde{t} \in \bar{\mathcal{T}}$ geçişine \tilde{t}_b denirse, bu geçişin hangi APA'na ait bir geçiş olduğu bilinmektedir, $\tilde{t}_b \in T_k$ ($k \in \{1, 2, \dots, S\}$) olsun. $\tilde{M}' = \tilde{M}^* + \tilde{A}\tilde{U}_{\tilde{t}_b}$ denkleminde $\tilde{t}_b \in T_k$ kullanılarak \tilde{M}' işaretleme vektörü elde edilmektedir. Bu işaretleme vektörleri, $\tilde{M}^* = [M_1^{*T} \mid M_2^{*T} \mid \dots \mid M_k^{*T} \mid \dots \mid M_S^{*T}]^T$ ve $\tilde{M}' = [M_1^{*T} \mid M_2^{*T} \mid \dots \mid M_k^{*T} \mid \dots \mid M_S^{*T}]^T$ şeklinde gösterilebilir. \tilde{M}' işaretleme vektöründen ve \tilde{M}' işaretleme vektöründen sonra ateşlenebilen $\tilde{t} \in \mathcal{T}$ geçişleri kullanılarak elde edilen işaretleme vektörlerinden sonra, hiç bir $\tilde{t} \in \bar{\mathcal{T}}$ geçişi ateşlenemez. Başka bir deyişle, APA'ların işaretleme vektörlerinden sonra da hiç bir geçiş ateşlenemez. Böylece, tüm APA'larında sistem çıkmazı meydana geldiği görülür. Dolayısıyla, M_k^* işaretleme vektöründen sonra hiç bir $\tilde{t} \in T_k$ geçişin ateşlenemediği bilinmektedir. Bu da, k . APA'da sistem çıkmazı meydana geldiğini göstermektedir. Buradan, D-1'e benzer şekilde $\tilde{M}' \in \tilde{\mathcal{F}}$ olduğu bulunur.

Sonuç olarak, $\tilde{\mathcal{G}}$ kümesinin elemanı olan her işaretleme vektörünün, $\tilde{\mathcal{F}}$ kümesinin de elemanı olduğu bulunmuştur. Böylece, bu bölümde verildiği şekilde tasarlanan kontrolör, $\tilde{\mathcal{F}}$ kümesinin elemanı olan işaretleme vektörlerine ulaşıl-

masına izin vermediğinden, tasarlanan kontrolör GPA'da çalışma engeli meydana gelmesini önler. \square

Not 6.1: $m_0 \in \mathcal{Y}(G)$ ise özgün Petri ağında kontrolör tasarlamak mümkün değildir (bkz. Bölüm 3.1). Bu çalışmada, özgün Petri ağı için sistem çıkmasının meydana gelmesini önleyen bir dışmerkezli kontrolör tasarlamak amaçlanmıştır. Bu nedenle, $m_0 \in \mathcal{Y}(G)$ ise, özgün Petri ağına örtüşmeli ayrıştırma ve geliştirme işlemlerini uygulayarak geliştirilmiş Petri ağı elde etmek ve bu ağ üzerinde işlem yapmak da anlamlı değildir.

Şekil 5.2'deki geliştirilmiş Petri ağı için aşağıdaki şekilde kontrolör tasarlanmaktadır: GPA'daki \tilde{M} işaretleme vektörü; $\tilde{M} = [M_1^T \mid M_2^T]^T$ şeklinde gösterilirken, kontrolör matrisi de,

$$\tilde{C}(\tilde{M}) = \begin{bmatrix} C_1(\tilde{M}) & 0 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & C_2(\tilde{M}) & 0 \\ 0 & 0 & 0 & I_3 \end{bmatrix}$$

olarak tanımlanmaktadır. Burada, $I_3 : 3 \times 3$ 'lük birim matrisi ve 0: uygun boyutlu sıfır matrislerini göstermektedir. Ayrıca, $C_1(\tilde{M}) = \alpha_1(M_1).C_1(M_1)$ ve $C_2(\tilde{M}) = \alpha_2(M_2).C_2(M_2)$ olarak alınmaktadır. Burada,

$$\alpha_1(M_1) \triangleq \begin{cases} 1, & M_1 \in R(G_{c1}, m_{10}) \\ 0, & \text{diğer durumlarda} \end{cases} \quad \text{ve} \quad \alpha_2(M_2) \triangleq \begin{cases} 1, & M_2 \in R(G_{c2}, m_{20}) \\ 0, & \text{diğer durumlarda} \end{cases}$$

şeklinde tanımlanmıştır.

Şekil 5.2'de verilen geliştirilmiş Petri ağında, başlangıçta robotun ve ana deponun tüm bölmelerinin 1. alt Petri ağına hizmet verecek şekilde seçilmiştir ve bunu gösteren işaretleme vektörü de, başlangıç işaretlemesi olarak alınmıştır. Şekil 5.2'de verilen geliştirilmiş Petri ağında, $\tilde{M} = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0 \mid 0 \ 0 \ 0 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ işaretleme vektörü için kontrol matrisinin bulunmasını inceleyelim. Burada, $\tilde{M} \in R(\tilde{G}, \tilde{m}_0)$ olmak üzere, $\tilde{M} = [M_1^T \mid M_2^T]^T$, $M_1 = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0]^T$ ve $M_2 = [0 \ 0 \ 0 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]^T$ gibi iki işaretleme vektörü elde edilir. Burada, M_1 işaretleme vektörü, Bölüm 6.1'de alt Petri ağlarının kontrolör tasarımı için verilen örnekte, M^{13} olarak adlandırılmıştır.

$M_1 \in R(G_{c1}, m_{10})$ olduğundan, $\alpha_1(M_1) = 1$ olur. Dolayısıyla,

$$C_1(\tilde{M}) = C_1(M_1) = C_1(M^{13}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

olur. \tilde{m}_0 seçiminden dolayı, M_2 , 2. APA'nın ulaşılabilirlik kümesinde bulunmadığından, $M_2 \notin R(G_{c2}, m_{20})$, $\alpha_2(M_2) = 0$ olur. Dolayısıyla, $C_2(\tilde{M}) = Z_2$ (Z_2 , 6×6 'lık sıfır matrisidir) olarak elde edilir. Sonuç olarak,

$$\tilde{C}(\tilde{M}) = \begin{bmatrix} C_1(M^{13}) & 0 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & Z_2 & 0 \\ 0 & 0 & 0 & I_3 \end{bmatrix}$$

olarak tanımlanır. Buna göre, bulunan kontrolör matrisi \tilde{M} işaretleme vektöründen sonra ateşlenmesine izin verilen geçişler içinde de genişleştirilmiş Petri ağının yapısından dolayı, t_6^1 geçişi ateşlenebilir.

6.1.3. Özgün Petri ağı için kontrolör tasarımı

Özgün Petri ağı için genişleştirilmiş Petri ağı için tasarlanan kontrolörden yararlanılarak, kontrolör tasarlanacaktır.

Genleştirilmiş Petri ağı için tasarlanan kontrolör, $\tilde{t} \in \mathcal{T}$ olan bütün geçişlerin ateşlenmesine izin verdiği için, \mathcal{T} kümesinin elemanı olan bir geçiş kullanıldığında, işaretleme vektörleri arasında $Q\tilde{\rho}(\tilde{t}, \tilde{M}) = Q\tilde{M}$, $\tilde{M} \in R(\tilde{G}_c, \tilde{m}_0)$, $\tilde{t} \in \mathcal{T}$ şeklinde bir ilişki olur. Buradan yola çıkarak, özgün Petri ağında bir işaretleme vektörüne karşılık gelen tüm genişleştirilmiş Petri ağındaki işaretleme vektörlerinin oluşturduğu küme Δ_M olarak adlandırılır ve $\Delta_M = \{\tilde{M} \in R(\tilde{G}_c, \tilde{m}_0) \mid M = Q\tilde{M}\}$ olarak ifade edilir. Bu bölümde kullanılacak diğer tanımlamalar da aşağıda verilmiştir.

- \oplus işlemi μ kümesi üzerinde;

$$1 \oplus 1 = 1 \quad 1 \oplus 0 = 1$$

$$0 \oplus 1 = 1 \quad 0 \oplus 0 = 0$$

şeklinde tanımlanmaktadır.

- $X : \mathcal{K} \rightarrow \mu$, $\mathcal{K} = \{k_1, k_2, \dots, k_j\}$, $j \in \mathcal{N}$ olmak üzere;

$$\bigoplus_{k \in \mathcal{K}} X(k) = X(k_1) \oplus X(k_2) \oplus \dots \oplus X(k_j)$$

tanımlanmaktadır.

Özgün Petri ağı için, kontrolör matrisi Bölüm 3.1'de verildiği şekilde, $C(M) = \text{köşegen}\{c(M, t)\}$ olarak ifade edilmektedir. Burada önemli kısım, C kontrolör matrisinin köşegeninde bulunan elemanlarının tanımlanmasıdır. Bu elemanlar, geliştirilmiş Petri ağı için tasarlanan kontrolörün elemanlarının

$$c(M, t) = \bigoplus_{\tilde{M} \in \Delta_M} \bigoplus_{\tilde{t} \in \mathcal{E}(\tilde{t})} \tilde{c}(\tilde{M}, \tilde{t}), \quad \forall M \in R(G_c, m_0), \quad \forall t \in T \quad (6.3)$$

şeklinde *büzülmesiyle* elde edilir.

Özgün Petri ağı için kontrolör bu şekilde elde edildiğinde, kontrol edilen geliştirilmiş Petri ağın (\tilde{G}_c), kontrol edilen özgün Petri ağını (G_c) içerdiği aşağıda gösterilecektir.

Teorem 6.2: Eğer geliştirilmiş Petri ağı (\tilde{G}), özgün Petri ağını (G) içeriyorsa, \tilde{G} için \tilde{C} (6.1), G için (6.3) şartını sağlayan bir C kontrolörü tasarlanmışsa, \tilde{G}_c , G_c 'yi içerir.

İspat: İçerme prensibi tanımından; Tanım 4.1'in (i). ve (ii). şıkları hem \tilde{G}_c 'nin, G_c 'yi içermesi için hem de \tilde{G} 'nin, G 'yi içermesi için geçerlidir. Dolayısıyla \tilde{G} , G 'yi içerdiğinden, Tanım 4.1'in (i). ve (ii). şartları \tilde{G}_c ve G_c için de sağlanır. Burada, ρ_c ve $\tilde{\rho}_c$ sırasıyla, G_c ve \tilde{G}_c 'de tanımlanan iletim fonksiyonlarını göstermektedir.

- A) \tilde{G}_c 'de herhangi bir $\tilde{M} \in R(\tilde{G}_c, \tilde{m}_0)$ ve $\tilde{M}' \in R(\tilde{G}_c, \tilde{M})$ işaretleme vektörleri için, $\tilde{t} \in \tilde{T}$ geçişi $\tilde{M}' = \tilde{\rho}_c(\tilde{t}, \tilde{M})$ eşitliğini sağlayan bir geçiş olsun.

- a1) $\tilde{t} \in \mathcal{T}$ ise; Lemma 5.1'in ispatından dolayı, ÖPA'da hiç bir geçiş karşılık gelmez.

a2) $\tilde{t} \in \bar{T}$ ise; \tilde{G}_c 'de \tilde{t} geçişi ateşlenebildiğinden ve \tilde{C} kontrolörün tanımından dolayı, $\tilde{c}(\tilde{M}, \tilde{t}) = 1$ olmaktadır. Bu ifade, denklem (6.3)'de kullanıldığında, $c(M, t) = 1$ ($\tilde{t} \in \mathcal{E}(t)$, $\tilde{M} \in \Delta_M$), M işaretleme vektöründen sonra kontrolör tarafından ateşlenmesine izin verilen bir $t \in T$ geçişi bulunur. Bölüm 5.1'de verilen örtüşmeli ayrıştırma ve genişleme işlemlerinden dolayı, bulunan t geçişi Lemma 5.1'in ispatında verilen \tilde{t} geçişine karşılık gelen geçiştir. t geçişi kullanılarak, $M' = \rho_c(t, M)$ işaretleme vektörü elde edilir. Burada, (5.1) ve (5.2) eşitlikleri sağlanmaktadır. Dolayısıyla, Tanım 4.1'in (iii). şıkkı sağlanmaktadır.

\tilde{t} geçişi yerine \tilde{g} ateşleme dizisi kullanıldığında, \tilde{g} ateşleme dizisini oluşturan geçişler sırayla ele alındığında, her bir geçiş yukarıda verilen (a1) veya (a2) durumlarından birine uymaktadır. Buradan, g ateşleme dizisinin, Lemma 5.1'in ispatında verilen \tilde{g} ateşleme dizisine karşılık gelen ateşleme dizisi olduğu bulunur. Dolayısıyla, Tanım 4.1'in (iii). şıkkı sağlanmış olur.

B) G_c 'de herhangi bir $M \in R(G, m_0)$ işaretleme vektörü için, $t \in T$ geçişi $M = \rho_c(t, m_0)$ eşitliğini sağlayan bir geçiş olsun.

G_c 'de t geçişi ateşlenebildiğinden dolayı, $c(M, t) = 1$ olmaktadır. Denklem (6.3)'den yararlanılarak, en az bir tane $\tilde{c}(\tilde{M}, \tilde{t}) = 1$ ($\tilde{t} \in \mathcal{E}(t)$, $\tilde{M} \in \Delta_M$) olan ifade bulunmaktadır. \tilde{C} kontrolörün tanımından dolayı, \mathcal{T} kümesinin elemanı olan bütün geçişlerin ateşlenmesine izin verilmektedir. Bölüm 5.1'de verilen örtüşmeli ayrıştırma ve genişleme işlemlerinden ve genişletilmiş Petri ağının topolojisinden dolayı, t geçişine karşılık elde edilen ateşleme dizisi Lemma 5.2'nin ispatında verilen ateşleme dizisidir. Bu nedenle, (5.1) eşitliği sağlanmaktadır. t geçişi yerine g ateşleme dizisi kullanıldığında da, yukarıda verildiği şekilde elde edilen \tilde{g} ateşleme dizisi Lemma 5.2'in ispatında verilen g ateşleme dizisine karşılık gelen ateşleme dizisi olduğu bulunur. Dolayısıyla, Tanım 4.1'in (iv). şıkkı sağlanmış olur.

Sonuç olarak, Tanım 4.1'in (i)-(iv). şartları sağlandığından, \tilde{G}_c , G_c 'yi içerir. \square

Sonuç 6.1: Teorem 4.4, 5.1, 6.1 ve 6.2'den dolayı, \tilde{G}_c için tasarlanan \tilde{C} kontrolörünün büzülmesi ile elde edilen C kontrolörü G_c 'de sistem çıkmazı meydana gelmesini önler.

Şekil 2.3'de verilen özgün Petri ağı için, Şekil 5.2'deki genişletilmiş Petri ağının kontrolörünün denklem (6.3)'deki şekilde büzülmesi ile elde edilebilmektedir. Örneğin, $M = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 4 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$ işaretleme vektörü içi C kontrolör matrisinin bulunması (Burada M işaretleme vektörü Bölüm 3.1'de incelenen M^3 işaretleme vektörü ile aynıdır):

$\Delta_M = \{\tilde{M}^{w,z} \mid w \in \mathcal{W} \triangleq \{0, 1, 2, 3, 4\}, z \in \mathcal{Q} \triangleq \{0, 1\}\}$, burada $\tilde{M}^{w,z} = [0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ w \ z \ 0 \ \bar{w} \ \bar{z} \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0]^T$ olarak alınmaktadır ($\bar{w} = 4 - w$ ve $\bar{z} = 1 - z$). $\tilde{M}^{w,z}$ işaretleme vektörünü $\tilde{M}^{w,z} = [(\tilde{M}_1^{w,z})^T \mid (\tilde{M}_2^{w,z})^T]^T$ olarak gösterilebilir. Δ_M kümesinin elemanlarından, $\tilde{M}^{4,1}$ ve $\tilde{M}^{0,0}$ işaretleme vektörleri hariç geri kalan işaretleme vektörlerinin 1. ve 2. alt Petri ağlarına ait kısımları sırasıyla 1. ve 2. alt Petri ağlarının ulaşılabilirlik kümelerinin elemanları olmadığından, bu işaretleme vektörleri için kontrolör matrisi

$$\tilde{C} = \text{köşegen}\{0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1\}$$

olmaktadır. $\tilde{M}_1^{4,1} \in R(G_1, m_{10})$ olduğundan ($\tilde{M}_1^{4,1} = M^{13}$), $\alpha_1(\tilde{M}_1^{4,0}) = 1$ olur ve $\tilde{M}_2^{4,1} \notin R(G_2, m_{20})$ olduğundan da, $\alpha_2(\tilde{M}_2^{4,1}) = 0$ olur. Buradan,

$$\tilde{C}(\tilde{M}^{4,1}) = \begin{bmatrix} C_1(M^{13}) & 0 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & Z_2 & 0 \\ 0 & 0 & 0 & I_3 \end{bmatrix}$$

elde edilir. $\tilde{M}_1^{0,0} \notin R(G_1, m_{10})$ olduğundan, $\alpha_1(\tilde{M}_1^{0,0}) = 0$ olur ve $\tilde{M}_2^{0,0} \in$

$R(G_2, m_{20})$ olduğundan da, $\alpha_2(\tilde{M}_2^{0,0}) = 1$ olur. Buradan,

$$\tilde{C}(\tilde{M}^{0,0}) = \begin{bmatrix} Z_6 & 0 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & I_6 & 0 \\ 0 & 0 & 0 & I_3 \end{bmatrix}$$

elde edilir.

Denklem (6.3)'den yararlanılarak $C(M)$ matrisinin bulunması:

$$c(M, t_1) = \tilde{c}(\tilde{M}^{4,1}, t_1) \oplus \tilde{c}(\tilde{M}^{0,0}, t_1) = 1 \oplus 0 = 1$$

$$c(M, t_2) = \tilde{c}(\tilde{M}^{4,1}, t_2) \oplus \tilde{c}(\tilde{M}^{0,0}, t_2) = 0 \oplus 0 = 0$$

$$c(M, t_3) = \tilde{c}(\tilde{M}^{4,1}, t_3) \oplus \tilde{c}(\tilde{M}^{0,0}, t_3) = 1 \oplus 0 = 1$$

$$c(M, t_4) = \tilde{c}(\tilde{M}^{4,1}, t_4) \oplus \tilde{c}(\tilde{M}^{0,0}, t_4) = 0 \oplus 0 = 0$$

$$c(M, t_5) = \tilde{c}(\tilde{M}^{4,1}, t_5) \oplus \tilde{c}(\tilde{M}^{0,0}, t_5) = 1 \oplus 0 = 1$$

$$c(M, t_6) = \tilde{c}(\tilde{M}^{4,1}, t_6^1) \oplus \tilde{c}(\tilde{M}^{0,0}, t_6^1) \oplus \tilde{c}(\tilde{M}^{4,1}, t_6^2) \oplus \tilde{c}(\tilde{M}^{0,0}, t_6^2)$$

$$= 1 \oplus 0 \oplus 0 \oplus 1 = 1$$

$$c(M, t_7) = \tilde{c}(\tilde{M}^{4,1}, t_7) \oplus \tilde{c}(\tilde{M}^{0,0}, t_7) = 0 \oplus 1 = 1$$

$$c(M, t_8) = \tilde{c}(\tilde{M}^{4,1}, t_8) \oplus \tilde{c}(\tilde{M}^{0,0}, t_8) = 0 \oplus 1 = 1$$

$$c(M, t_9) = \tilde{c}(\tilde{M}^{4,1}, t_9) \oplus \tilde{c}(\tilde{M}^{0,0}, t_9) = 0 \oplus 1 = 1$$

$$c(M, t_{10}) = \tilde{c}(\tilde{M}^{4,1}, t_{10}) \oplus \tilde{c}(\tilde{M}^{0,0}, t_{10}) = 0 \oplus 1 = 1$$

$$c(M, t_{11}) = \tilde{c}(\tilde{M}^{4,1}, t_{11}) \oplus \tilde{c}(\tilde{M}^{0,0}, t_{11}) = 0 \oplus 1 = 1$$

şeklinde elde edilmektedir. Dolayısıyla,

$$C(M) = \text{köşegen}\{1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1\}$$

olarak elde edilir. Elde edilen $C(M)$ matrisi ile Bölüm 3.1'de bulunan $C(M^3)$ matrisinin aynı olduğu görülmektedir.

Petri ağlarında dışmerkezli kontrolör tasarımı kullanılarak, sistem çıkmazının meydana gelmesini engellemek mümkündür. Örtüşmeli ayrıştırmaya dayalı dışmerkezli kontrolör tasarımında, yerel olarak sistem çıkmazının meydana gelmesini önleyen kontrolörlerden yararlanılmaktadır. Geliştirilen kontrolör tasarımında, Petri ağın tamamında ulaşılabilirlik kümesinin belirlenmesi yerine daha küçük boyutlu Petri ağlarda ulaşılabilirlik kümeleri belirlenmektedir. Bu nedenle, özellikle büyük ölçekli sistemler (yer ve geçiş sayılarının

büyük olduğu sistemler) için, kontrolör tasarımı dışmerkezli kontrolör tasarımı kullanmak daha avantajlı olmaktadır.

6.2. Otomata Gösteriminde Dışmerkezli Bir Kontrolör Tasarım Yöntemi

Burada otomata gösterimi için, sistem çıkmazı meydana gelmesini önleyen bir dışmerkezli kontrolör tasarım yöntemi geliştirilecektir. Bu tasarımda sırasıyla; alt otomatalar, geliştirilmiş otomata ve özgün otomata için kontrolör tasarlanacaktır. Ayrıca, sistem çıkmazının meydana gelmesini önleyen bir dil tasarımı da verilecektir.

6.2.1. Alt otomatalarda kontrolör tasarımı

Örtüşmeli ayrıştırma ve genişletme işlemleri sonucu elde edilen her bir alt otomata için, tek başına bir otomata gibi, Bölüm 3.2.2'de verildiği şekilde sistem çıkmazının meydana gelmesini önleyen bir kontrolör tasarlanabilir. k . alt otomata için kontrolör tasarımı önce, Q_k^d , Q_k^b ve D_k ($k \in \{1, 2, \dots, K\}$) kümeleri bulunmalıdır. Q_k^d , k . alt otomata için sistem çıkmazı gösteren durumların oluşturduğu kümeyi (bkz. Tanım 3.1), $Q_k^b = \{q \mid \exists t \in \Sigma_k, \text{ öyle ki } \gamma(t, q) \in Q_k^d \text{ ve } D_k(q)\}$; $q \in Q_k^b$ durumundan sonra, sistem çıkmazı gösteren durumlara ulaşılmasını sağlayan olayların oluşturduğu kümeyi göstermektedir. $q \notin Q_k^b$ ise, $D_k(q) = \emptyset$ olarak alınmaktadır.

k . alt otomata için kontrolör tasarımı, aşağıdaki şekilde yapılmaktadır.

$$C_k(\tilde{t}, \tilde{q}) = \begin{cases} 0, & \tilde{t} \in D_k(\tilde{q}) \\ 1, & \text{aksi durumda} \end{cases}, \tilde{q} \in \bar{Q}_k, \tilde{t} \in \Sigma_k$$

Şekil 5.7'deki 1. alt otomatada, q_2 durumundan sonra t_4 olayının olması ile ulaşılan q_3 durumu sistem çıkmazı gösteren bir durumdur. Bu nedenle, $Q_1^d = \{q_3\}$, $Q_1^b = \{q_2\}$ ve $D_1(q_2) = \{t_4\}$ olarak alınmaktadır. Tasarlanan kontrolörde ise, $C_1(t_1^1, q_0^1) = 1$, $C_1(t_2, q_1^1) = 1$, $C_1(t_3, q_2) = 1$ ve $C_1(t_4, q_2) = 0$ olmaktadır. Şekil 5.7'deki 2. alt otomatada ise, q_0^2 durumundan sonra t_1^2 olayının olması ile ulaşılan veya q_5 durumundan sonra t_8 olayının olması ile

ulaşılan q_1^2 durumu sistem çıkmazı gösteren bir durumdur. Bu nedenle, $Q_2^d = \{q_1^2\}$, $Q_2^b = \{q_0^2, q_5\}$ ve $D_2(q_0^2) = \{t_1^2\}$, $D_2(q_5) = \{t_8\}$ olarak alınmaktadır. Tasarlanan kontrolörde ise, $C_2(t_1^2, q_0^2) = 0$, $C_2(t_5, q_0^2) = 1$, $C_2(t_6, q_4) = 1$, $C_2(t_7, q_5) = 1$ ve $C_2(t_8, q_5) = 0$ olmaktadır.

6.2.2. Genleştirilmiş otomatada kontrolör tasarımı

Alt otomatalar için tanımlanan $D_i(\tilde{q})$ ($\tilde{q} \in \tilde{Q}_i \forall i \in \{1, 2, \dots, K\}$) kümeleri kullanılarak, genleştirilmiş otomata için bir kontrolör tasarlanacaktır. Her alt otomata için bu kümelerin tanımlı olduğu kabul edilmiştir.

Genleştirilmiş otomatada, \tilde{D} kümesi,

$$\tilde{D}(\tilde{q}) \triangleq D_i(\tilde{q}), \forall \tilde{q} \in \tilde{Q}_i, \forall i \in \{1, 2, \dots, K\}$$

şeklinde tanımlanmaktadır. Burada, genleştirilmiş otomatada ele alınan bir durumdan sonra engellenmek istenilen olaylar, bulunduğu alt otomatada belirlenen olaylar olarak alınmaktadır.

Şekil 5.7'de verilen genleştirilmiş otomatada, $\tilde{D}(q_0^1) = D_1(q_0^1) = \emptyset$, $\tilde{D}(q_0^2) = D_2(q_0^2) = \{t_1^2\}$, $\tilde{D}(q_1^1) = D_1(q_1^1) = \emptyset$, $\tilde{D}(q_2) = D_1(q_2) = \{t_4\}$, $\tilde{D}(q_4) = D_2(q_4) = \emptyset$ ve $\tilde{D}(q_5) = D_1(q_5) = \{t_8\}$ olarak elde edilmektedir. q_3 , 1. alt otomatada ve q_1^2 , 2. alt otomatada sistem çıkmazı gösteren durumlardır.

Genleştirilmiş otomata için, sistem çıkmazı gösteren durumlara ulaşılmasını önleyen kontrolör aşağıda verilmiştir.

$$\tilde{C}(\tilde{t}, \tilde{q}) = \begin{cases} 0, & \tilde{t} \in \tilde{D}(\tilde{q}) \\ 1, & \text{aksi durumda} \end{cases}, \tilde{q} \in \tilde{Q}, \tilde{t} \in \tilde{\Sigma}$$

Örneğin Şekil 5.7'de verilen genleştirilmiş otomata için tasarlanan kontrolör, $\tilde{C}(t_1^1, q_0^1) = 1$, $\tilde{C}(t_2, q_1^1) = 1$, $\tilde{C}(t_3, q_2) = 1$, $\tilde{C}(t_4, q_2) = 0$, $\tilde{C}(t_1^2, q_0^2) = 0$, $\tilde{C}(t_5, q_0^2) = 1$, $\tilde{C}(t_6, q_4) = 1$, $\tilde{C}(t_7, q_5) = 1$, $\tilde{C}(t_8, q_5) = 0$, $\tilde{C}(\alpha_{o1}, q_0^1) = 1$, $\tilde{C}(\alpha_{o2}, q_0^2) = 1$, $\tilde{C}(\alpha_{o3}, q_1^1) = 1$ ve $\tilde{C}(\alpha_{o4}, q_1^2) = 1$ olmaktadır.

6.2.3. Özgün otomatada kontrolör tasarımı

Özgün otomatada, genleştirilmiş otomata için tanımlanan \tilde{D} kümesi kullanılarak, bir kontrolör tasarlanacaktır. Özgün otomatada, D kümesi,

$$D(q) = \mathcal{H}\left(\bigcup_{\tilde{q} \in \mathcal{E}(q)} \tilde{D}(\tilde{q})\right), \quad q \in Q \quad (6.4)$$

şeklinde tanımlanmaktadır. Burada, $\mathcal{H} : \tilde{\Sigma} \rightarrow \Sigma$ dönüşümü; $\vartheta \subset \tilde{\Sigma}$ olmak üzere, $\mathcal{H}(\vartheta) = \{t \in \Sigma \mid \mathcal{E}(t) \subset \vartheta\}$ şeklinde tanımlanmaktadır.

Örneğin, Şekil 5.5'deki q_0 durumu için,

$$D(q_0) = \mathcal{H}(\tilde{D}(q_0^1) \cup \tilde{D}(q_0^2)) = \mathcal{H}(\emptyset \cup \{t_1^2\}) = \mathcal{H}(\{t_1^2\}) = \emptyset$$

olarak bulunmaktadır. q_0 durumundan sonra t_1 veya t_5 olaylarının olmasıyla, sırasıyla q_1 veya q_4 durumlarına ulaşılmaktadır. q_2 durumu için ise,

$$D(q_2) = \mathcal{H}(\tilde{D}(q_2)) = \mathcal{H}(D_1(q_2)) = \mathcal{H}(\{t_4\}) = \{t_4\}$$

olarak bulunmaktadır. q_2 durumundan sonra t_3 veya t_4 olaylarının olmasıyla, sırasıyla q_0 veya q_3 durumlarına ulaşılmaktadır. Ancak q_3 durumu sistem çıkmazı gösteren bir durumdur. Bu nedenle q_3 durumuna ulaşılmasını sağlayan t_4 olayı, $t_4 \in D(q_2)$ 'dir.

Lemma 6.2: Özgün otomatada sistem çıkmazı gösteren durumlara ulaşılmasını sağlayan olaylar, denklem (6.4)'de verilen kümenin elemanlarıdır.

İspat: İspatlanacak ifade, $D_R(q) \subset D(q)$, $\forall q \in Q$ olarak da gösterilebilir. Burada $D_R(q)$, özgün otomatada q durumundan sonra sistem çıkmazı gösteren durumlara ulaşılmasını sağlayan olayların oluşturduğu kümeyi göstermektedir. Teorem 5.2'den dolayı, genişleştirilmiş otomata, özgün otomataı içermektedir. Örtüşmeli ayrıştırma ve genişleştirme işlemlerinden dolayı, genişleştirilmiş otomatada tekrarlanmış bir olayın girdi ve çıktı olarak bağlı olduğu durumlar, tekrarlanmış durumlardır. Ayrıca, örtüşmeli kısımda birbirlerine girdi veya çıktı olarak bağlı durum ve olaylar, aynı alt otomatların ortak durum ve olaylarıdır.

Herhangi bir $q \in Q^b$ için $t \in D_R(q)$ olsun.

- a) q durumu özgün otomatada örtüşmeli kısımda bulunmuyorsa, genişleştirilmiş otomatadaki herhangi bir alt otomatada bulunmaktadır. Bu alt otomata, j . alt otomata olsun. Örtüşmeli ayrıştırma ve genişleştirme

işlemlerinden dolayı, q durumunun özgün otomatadaki bağlantıları, j . alt otomatada bir değişiklik olmaksızın aynen korunmaktadır. Bu nedenle $t \in D_R(q)$ ise, $t \in D_j(q)$ olmaktadır. $\mathcal{E}(t) = \{t\}$ olduğundan dolayı, $D(q) = \mathcal{H}(\tilde{D}(q)) = \mathcal{H}(D_j(q)) = D_j(q)$ olarak elde edilmektedir. $t \in D_j(q)$ ve $D(q) = D_j(q)$ olduğundan dolayı da, $t \in D(q)$ olmaktadır.

b) q durumu özgün otomatada örtüşmeli kısımda bulunuyorsa, elemanı olduğu alt otomatalarda tekrarlanmaktadır, $\mathcal{E}(q) = \{\tilde{q}_i^1, \tilde{q}_i^2, \dots, \tilde{q}_i^n\}$.

– t olayı özgün otomatada örtüşmeli kısımda bulunmuyorsa, $\mathcal{E}(t) = \{t\}$ olmaktadır. Örtüşmeli ayrıştırma ve genişleme işlemlerinden dolayı, özgün otomatadaki t ile q arasındaki bağlantı, genişletilmiş otomatada t ile q 'nun tekrarlanmış bir durumu arasında vardır. Tekrarlanmış bu durum \tilde{q}_i^k ile gösterilsin. Örtüşmeli ayrıştırma ve genişleme işlemlerinden ve genişletilmiş otomatanın topolojisinden dolayı, $q \in Q^b$ ise, $\tilde{q}_i^k \in Q_k^b$ olarak elde edilmektedir. Bu nedenle $t \in D_R(q)$ ise, $t \in D_k(\tilde{q}_i^k)$ olmaktadır. $t \in D_k(\tilde{q}_i^k)$ olduğundan ve denklem (6.4)'den dolayı, $t \in D(q)$ olmaktadır.

– t olayı özgün otomatada örtüşmeli kısımda bulunuyorsa, $\mathcal{E}(t) = \{\tilde{t}_m^1, \tilde{t}_m^2, \dots, \tilde{t}_m^n\}$ olmaktadır. Özgün otomatada, q durumundan sonra t olayının olması ile meydana gelen durum, sistem çıkmazı gösteren bir durumdur. Örtüşmeli ayrıştırma ve genişleme işlemlerinden ve genişletilmiş otomatanın topolojisinden dolayı, bu sistem çıkmazı gösteren durumun genişletilmiş otomatadaki tekrarlanmış durumlarının herbiri, bulunduğu alt otomatada sistem çıkmazı gösteren bir durumdur. Bu nedenle, $q \in Q^b$ ise, $\tilde{q}_i^l \in Q_l^b$ ($\forall l \in \{1, 2, \dots, n\}$) olarak elde edilmektedir. Dolayısıyla, $t \in D_R(q)$ ise, $\tilde{t}_m^l \in D_l(\tilde{q}_i^l)$ ($\forall l \in \{1, 2, \dots, n\}$) olmaktadır. $\mathcal{E}(t)$ kümesinin bütün elemanları da, \tilde{D} kümesinin tanımından dolayı, $\bigcup_{\tilde{q} \in \mathcal{E}(q)} \tilde{D}(\tilde{q})$ kümesinin elemanları olarak bulunmaktadır. Buradan ve denklem (6.4)'den dolayı, $t \in D(q)$ olmaktadır.

(a) ve (b) şıklarından, herhangi bir $t \in D_R(q)$ için, $t \in D(q)$ olduğu bulunmaktadır. Böylece $D_R(q) \subset D(q)$, $\forall q \in Q$ olmaktadır. \square

Özgün otomata için, sistem çıkmazı gösteren durumlara ulaşılmasını önleyen bir kontrolör aşağıda verilmiştir.

$$C(t, q) = \begin{cases} 0, & t \in D(q) \\ 1, & \text{aksi durumda} \end{cases}, q \in Q, t \in \Sigma \quad (6.5)$$

Şekil 5.5'deki otomatada, örneğin $C(t_8, q_5) = 0$ ve $C(t_1, q_0) = 1$ olarak bulunmaktadır.

Sonuç 6.2: Lemma 6.2'den dolayı, denklem (6.5)'de verilen kontrolör özgün otomatada sistem çıkmazının meydana gelmesini önler.

Otomata gösteriminde, örtüşmeli ayrıştırmaya dayalı bir dışmerkezli kontrolör tasarımı gerçekleştirilmiştir. Bu tasarımda otomatanın tamamı yerine, sadece alt otomatalar incelendiğinden dolayı, özellikle büyük ölçekli sistemler için dışmerkezli kontrolör tasarımının kullanılması daha kolay olmaktadır. Ayrıca, bir sonraki bölümde anlatılacağı gibi, bu dışmerkezli kontrolör tasarımından yararlanılarak, bir dil tasarımı da yapılabilmektedir.

6.2.4. Sistem çıkmazının meydana gelmesini önleyen dil tasarımı

Bölüm 6.2.3'de otomata gösteriminde, sistem çıkmazının meydana gelmesini önleyen bir kontrolör tasarımı yapılmıştır. Burada, otomata gösteriminde tanımlanan $D(q)$ ($q \in Q$) kümesinden yararlanılarak sistem çıkmazının meydana gelmesini önleyen dil tasarımı yapılacaktır.

$$L_d = \left\{ s \in L \mid e_i(s) \notin D(\gamma(\Lambda_{i-1}(s), q_0)), \forall i \in \{1, 2, \dots, E(s)\} \right\} \quad (6.6)$$

Burada, $e_i(s)$, s olay dizisinin i . elemanı olan olayı, $E(s)$, s olay dizisinin kaç tane olaydan meydana geldiğini, göstermektedir. Ayrıca, $\Lambda_1(s) = e_1(s)$ olmak üzere $\Lambda_i(s) = \Lambda_{i-1}(s)e_i(s)$, $\Lambda_{i-1}(s)$, $e_i(s)$ 'nin ön (prefix) olay dizisidir.

Şekil 5.5'de verilen otomata için Bölüm 6.2.3'de, $D(q_2) = \{t_4\}$ ve $D(q) = \emptyset, \forall q \neq q_2$ olarak bulunmuştur. D kümesinden yararlanılarak, $L_d = \{\epsilon, t_1, t_1t_2, \dots\}$ olarak bulunmaktadır.

Teorem 6.3: Denklem (6.6)'da tanımlanan dil, sistem çıkmazı gösteren durumlara ulaşmayı engeler.

İspat: $s \in L_d$ olmak üzere, $q = \gamma(s, q_0) \in Q$ durumu elde edilsin.

s olay dizisi, $s = t_{r_1} t_{r_2} \dots t_{r_{k-1}} t_{r_k}$; $t_{r_1}, t_{r_2}, \dots, t_{r_{k-1}}, t_{r_k} \in \Sigma$ olarak gösterilsin. s olay dizisini oluşturan olayların sırasıyla kullanılması ile $q_{r_1} = \gamma(t_{r_1}, q_0)$, $q_{r_2} = \gamma(t_{r_1} t_{r_2}, q_0) = \gamma(t_{r_2}, q_{r_1})$, ..., $q = \gamma(t_{r_1} t_{r_2} \dots t_{r_k}, q_0) = \gamma(t_{r_k}, q_{r_{k-1}})$ durumları elde edilmektedir. L_d kümesinin tanımına göre; $t_{r_1} \notin D(q_0)$, $t_{r_2} \notin D(q_{r_1})$, ..., $t_{r_{k-1}} \notin D(q_{r_k})$ olmaktadır. Lemma 6.2'den dolayı, $q_{r_1}, q_{r_2}, \dots, q$ durumları sistem çıkmazı göstermeyen durumlardır. Sonuç olarak, L_d kümesinin elemanları kullanılarak elde edilen durumlar, sistem çıkmazı göstermeyen durumlardır. \square

Kesikli olay sistemlerin dil gösteriminde, sistem çıkmazı gösteren durumlara ulaşmayı engelleyen ve Bölüm 3.2.1'de verilen tasarımdan farklı olan bir dil tasarlanmıştır. Bu tasarımda, ele alınan olay dizisinin elemanları kontrol edilmekte ve buna göre L_d kümesi oluşturulmaktadır. Bölüm 3.2.1'de verilen tasarımda ise, istenilmeyen durumlara ulaşmayı sağlayan olay dizileri L kümesinden çıkartılarak L_m kümesi oluşturulmaktadır. Her iki dil de, L 'nin alt kümesidir ve sistem çıkmazı gösteren durumlara ulaşılmasını engellemektedir. Bölüm 3.2.1'de verilen ilişkiye benzer şekilde, L_d kümesinin elemanları, $R(G_c, m_0)$ kümesinin elemanı olan işaretleme vektörlerinin elde edilmesini sağlayan tüm ateşleme dizileri ile ilişkilendirilir.

6.3. Önerilen Yöntemlerin Hesaplama Karmaşıklığı

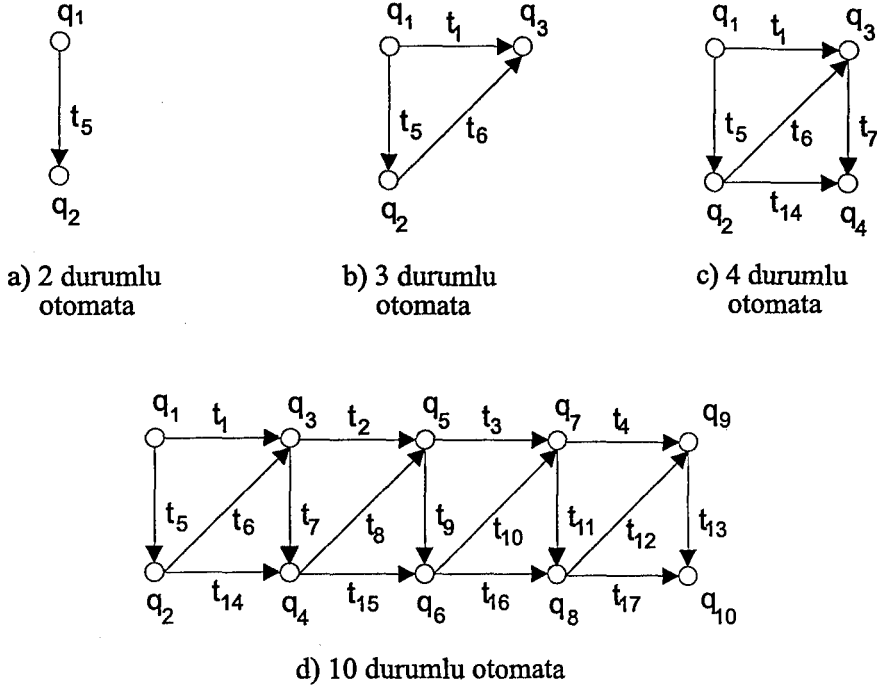
Burada otomata ve Petri ağları gösterimleri için bulunan, sistem çıkmazının meydana gelmesini önleyen dışmerkezli kontrolör tasarımları ile merkezi tasarımların hesaplama karmaşıklığı karşılaştırılacaktır.

Otomata gösterimindeki kontrolör tasarımında, her bir duruma ve ele alınan durumdan sonra olan olaylara bakılmakta ve başlangıç durumundan diğer durumlara ulaşılmasını sağlayan olay dizileri belirlenmektedir. Dolayısıyla kontrolör tasarımındaki hesaplama karmaşıklığı, başlangıç durumun-

dan, diğer durumlara ulaşılmasını sağlayan tekrarlanmayan olay dizilerinin sayısına bağlıdır. Örtüşmeli ayrıştırma ve genişletme işlemleri sonucu elde edilen alt otomatalarda, genişleştirilmiş otomatanın başlangıç durumunun bulunduğu alt otomata hariç diğer alt otomatalarda tekrarlanmayan olay dizilerini belirlemek için başlangıç durumlarının bilinmesi gerekmektedir. Bunun için, alt otomatalardaki tekrarlanan durumların herbiri olası başlangıç durumu olarak alınarak tekrarlanmayan olay dizileri belirlenir (herhangi bir alt otomatadaki bir durumun meydana gelmesi için önce ilgili olaylar ile ele alınan alt otomatanın tekrarlanmış durumlarına ulaşılmalıdır).

Örnek olarak, Şekil 5.5 ve 5.6'da verilen otomaların hesaplama karmaşıklıklarını karşılaştıralım. Şekil 5.5'de verilen otomata için, tekrarlanmayan olay dizilerinin sayısı, $C^* = 15$ olarak bulunmaktadır. Şekil 5.6'da verilen otomatanın, 1. alt otomasında başlangıç durumu, genişleştirilmiş otomatanın başlangıç durumu olan q_0^1 'dir. 2. alt otomatada olası başlangıç durumları olarak iki tane tekrarlanmış durum vardır, q_0^2 ve q_1^2 . Ancak 2. alt otomatanın topolojisinden dolayı, q_1^2 durumundan sonra hiçbir olayın olması mümkün değildir. Dolayısıyla, 1. alt otomatanın başlangıç durumu, q_0^1 ve 2. alt otomatanın başlangıç durumu, q_0^2 olmaktadır. Buradan yola çıkarak, tekrarlanmayan olay dizilerinin sayısı, 1. alt otomata için $C_1^* = 4$, 2. alt otomata için $C_2^* = 6$, toplam $C_T^* = C_1^* + C_2^* = 10$ olarak bulunmaktadır. Sonuç olarak, dışmerkezli kontrolör tasarımında, merkezi tasarıma göre yaklaşık olarak, %33 daha az hesaplama yapılmaktadır.

Durumların ve olayların sayıları arttıkça dışmerkezli kontrolör tasarlamak daha avantajlı olmaktadır. Örneğin, Şekil 6.1'de verilen "merdiven" otomatalardaki durumların ve olayların sayısı arttıkça tekrarlanmayan olay dizilerinin sayısı da artmaktadır. $n = |Q|$ olmak üzere, durumlarının sayısı $n = 1$ 'den $n = 20$ 'ye kadar olan otomatalardaki tekrarlanmayan olay dizilerinin sayısı Çizelge 6.2'de verilmiştir. Burada C_n^* , n durumlu otomatanın tekrarlanmayan olay dizilerinin sayısını göstermektedir. Bu çizelge incelendiğinde, tekrarlanmayan olay dizilerinin sayısının artışının, durumların ve olayların artışına göre, üstsel (exponential) bir artış olduğu görülmektedir. Örnek



Şekil 6.1. Merdiven otomatalar

olarak ele alınan merdiven otomatalar için, üstsel artışın tabanını, $\alpha \simeq 1.6$ olarak bulunmaktadır.

Bu örnekten yola çıkarak, durumlarının ve olaylarının sayısı çok daha fazla olan bir otomata için önerilen yöntem ile merkezi yöntemin hesaplama karmaşıklıklarını inceleyelim. Örneğin durum sayısı $n = 100$ olan bir otomata'yı ele alalım. Bu otomatada, örtüşmeli ayrıştırma ve genişletme uygulanarak 12 tane 10'ar durumlu alt otomatalar elde edilmiş ve her alt otomatada 5 tane olası başlangıç durumu olmuş olsun. Özgün otomatada ve her bir alt otomatada olayların sayısının durumların sayısına orantılı olduğu ve hesaplama karmaşıklığının durum ve olayların sayısına üstsel olarak bağlı olduğu varsayımı altında, özgün otomata için hesaplama karmaşıklığı α^{100} ile orantılı olmaktadır. Her bir alt otomatada her bir başlangıç durumu için hesaplama karmaşıklığı α^{10} ile ve olası başlangıç durumuları için toplam hesaplama karmaşıklığı $5 \times \alpha^{10}$ ile orantılıdır. Dolayısıyla önerilen hesaplama karmaşıklığının, $12 \times 5 \times \alpha^{10} = 60\alpha^{10}$ ile orantılı olduğu bulunmaktadır. Ele alınan bu örnekte, sistem çıkmazının meydana gelmesini önlemek için tasarlanan dışmerkezli kontrolör, merkezi tasarıma göre $\alpha^{100}/60\alpha^{10} = \alpha^{90}/60$ kat daha

kolay hesaplanabilmektedir. Burada $\alpha = 1.6$ için kazanç yaklaşık olarak 3.91×10^{16} 'dır.

Çizelge 6.2. Şekil 6.1'de verilen otomataların hesaplama karmaşıklığı

n	$ \Sigma $	C_n^*	C_n^*/C_{n-1}^*
2	1	1	—
3	3	3	3
4	5	6	2
5	7	11	1.83
6	9	19	1.78
7	11	32	1.68
8	13	53	1.66
9	15	87	1.64
10	17	142	1.63
11	19	231	1.63
12	21	375	1.62
13	23	608	1.62
14	25	985	1.62
15	27	1595	1.62
16	29	2582	1.61
17	31	4179	1.61
18	33	6763	1.61
19	35	10944	1.61
20	37	17709	1.61

Petri ağların kontrolör tasarımı ise, Petri ağının ulaşılabilirlik kümesinin elemanlarından sonra atışlenebilecek geçişler belirlenmektedir. Bu nedenle, Petri ağının ulaşılabilirlik kümesinin elemanlarının bulunması gerekmektedir. Bölüm 2.2.1'de verildiği şekilde, Petri ağının bir işaretleme vektörüne, otomata gösteriminde bir durum karşılık gelmektedir. Dolayısıyla, otomata gösterimindeki kontrolörün hesaplama karmaşıklığı üstsel olarak arttığından dolayı, Petri ağı için tasarlanan kontrolörün hesaplama karmaşıklığı da ulaşılabilirlik kümesinin elemanları cinsinden üstsel olarak artacaktır.

Dolayısıyla büyük ölçekli sistemler için, hem otomata (dolayısıyla dil) hem de Petri ağları gösterimlerinde, dışmerkezli kontrolör tasarımlarını kullanmak, merkezi kontrolör tasarımlarına göre daha kolay olmakta ve daha az hesaplama gerektirmektedir. Benzer kazanç vektörel gösterim için de vardır [28].

7. SONUÇLAR

Bu çalışmada yapılan incelemeler, büyük ölçekli kesikli olay sistemlerin modellenmesinde, Petri ağlarının kullanılmasının, sistemin anlaşılabilirliğini kolaylaştırdığı gibi üzerinde çalışma yapılabilmesini de kolaylaştırdığını ortaya koymaktadır. Ancak Petri ağı ve otomata gösterimlerinin birbirlerine tam bir üstünlükleri bulunmamakla beraber, bazı kesikli olay sistemlerinin modellenmesinde, biri diğerine tercih edilebilmektedir. Bu çalışmada Petri ağları ile vektörel, dil ve otomata gösterimlerinde kullanılan değişkenler ve tanımlar arasındaki ilişkiler de ortaya konmuştur. Bu yöntemlerden biri ile modellenen bir kesikli olay sistemin istenilen diğer bir yöntem ile ifade edilmesi de mümkün olmaktadır.

Bu tezde, Petri ağları için içerme prensibi tanımlanmıştır. İçerme prensibinden yararlanılarak; içeren ile içerilen Petri ağları arasında, sınırlılık, tersine dönüşebilirlik, canlılık ve bu çalışmada tanımlanan çalışma engeli adlı özellikler incelenmiştir. Eğer içeren Petri ağı sınırlı, tersine dönüşebilir ve/veya canlı (ilave bir şart ile birlikte) ise, içerilen Petri ağı da sırasıyla sınırlı, tersine dönüşebilir ve/veya canlıdır. Ayrıca, içeren Petri ağında çalışma engeli meydana gelmiyorsa, içerilen Petri ağında da sistem çıkmazı meydana gelmemektedir. Bu çalışmada, kesikli olay sistemlerin diğer modelleme yöntemlerinden olan otomata ile dil gösterimleri için de ayrı bir içerme prensibi tanımı verilmiştir.

Sürekli durum sistemleri için yapılan örtüşmeli ayrıştırma ve genişletme işlemlerinin, Petri ağları ve otomata ile dil gösterimlerinin herbiri için, uygulanması verilmiştir. Bunun sonucunda, ayrışık olarak ortaya çıkan alt Petri ağlarının (otomataların) incelenmesi sağlanmıştır. Örtüşmeli ayrıştırma ve genişletme sonucu elde edilen genişletilmiş Petri ağının (otomatanın, dilin), özgün Petri ağını (otomatayı, dili) içerdiği bulunmuştur. Bu çalışmada Petri ağları ve otomata gösteriminde ve [28]'de vektörel gösterimde yapılan, örtüşmeli ayrıştırma ve genişletme işlemleri birbirlerinden farklı sonuçlar vermektedir. Bölüm 5'in sonunda detaylı şekilde verildiği üzere, ele alınan bir sonlu durumlu kesikli olay sistemi için, farklı modelleme yöntemlerinde uygulanan

örtüşmeli ayrıştırma ve geliştirme işlemleri sonucu elde edilen geliştirilmiş sistemler ve alt sistemler, kullanılan gösterime bağlı olarak, birbirlerinden farklı olmaktadır.

Genleştirilmiş Petri ağı, özgün Petri ağını içeriyor ve içeren Petri ağına çalışma engeli meydana gelmiyorsa, özgün Petri ağına da sistem çıkmazının meydana gelmemesinden yararlanılarak, sistem çıkmazının meydana gelmesini önlemek için bir dışmerkezli kontrolör tasarlanmıştır. Bu kontrolörü tasarlamak için sırasıyla şu adımlar takip edilmiştir: i) Petri ağına, örtüşmeli ayrıştırma ve geliştirme uygulanmakta, ii) alt Petri ağları ayrışık olarak elde edilmekte, iii) her bir ayrışık alt Petri ağı için kontrolör tasarlanmakta, iv) alt Petri ağı için tasarlanan bu kontrolörler kullanılarak, geliştirilmiş Petri ağı için kontrolör tasarlanmakta, v) geliştirilmiş Petri ağı için tasarlanan kontrolörden yararlanılarak özgün Petri ağı için kontrolör elde edilmektedir. Elde edilen bu dışmerkezli kontrolör, Petri ağı ile modellenen kesikli olay sistemlerde sistem çıkmazının meydana gelmemesini garanti etmektedir.

Otomata gösterimi için, örtüşmeli ayrıştırmaya dayalı bir dışmerkezli kontrolör tasarlanmıştır. Bu tasarım için, sırasıyla şu adımlar takip edilmiştir: i) Otomatada, örtüşmeli ayrıştırma ve geliştirme uygulanmakta, ii) alt otomata ayrışık olarak elde edilmekte, iii) her bir ayrışık alt otomata için kontrolör tasarlanmakta, iv) alt otomata için tanımlanan, sistemde meydana gelmesi istenilmeyen olay dizilerinin oluşturduğu kümeler kullanılarak, geliştirilmiş otomata için kontrolör tasarlanmakta, v) geliştirilmiş otomata için tanımlanan sistemde meydana gelmesi istenilmeyen olay dizilerinin oluşturduğu küme kullanılarak, özgün otomata için sistem çıkmazının meydana gelmesini önleyen bir kontrolör elde edilmektedir. Ayrıca otomata gösterimi için tasarlanan kontrolörden yararlanılarak, sistem çıkmazının meydana gelmesini önleyen bir dil de tasarlanmıştır.

Petri ağı veya otomata gösterimi kullanılarak, kesikli olay sistemler modellenmektedir. Daha önce yapılan çalışmalarda bu modelleme yöntemleri için kontrolör tasarım yöntemleri geliştirilmiştir. Ancak ilk defa bu çalışmada, modelleme yöntemlerinin her biri için, yukarıda adımları verilen, örtüşmeli

ayrıştırılmaya dayalı kontrolör tasarım yöntemleri geliştirilmiştir. Böylece, Petri ağının (otomatanın) tamamına bakılmaksızın, ayrışık olarak elde edilmiş olan, alt Petri ağları (otomatalar) için tasarlanan kontrolör kullanılarak, Petri ağı (otomata) için bir kontrolör tasarlanabilmektedir.

Bir Petri ağı için kontrolör tasarımı, yerlerin ve geçişlerin sayısı arttıkça güçleşmektedir. Benzer şekilde bir otomata veya dil gösteriminde de durumların ve olayların sayısı arttıkça kontrolör tasarımı güçleşmektedir. Bu çalışmada geliştirilen yöntemlerle, özellikle büyük ölçekli kesikli olay sistemleri için, sistem çıkmazının meydana gelmesini önleyen kontrolör daha kolay bulunmaktadır. Bölüm 6.3'de detayları verildiği şekilde, Petri ağları ve otomata gösterimleri için dışmerkezli kontrolör tasarımının, merkezi kontrolör tasarımına göre hesaplama kolaylığı vardır ve zaman açısından dışmerkezli kontrolör, daha kısa sürede tasarlanabilir. Ancak, sürekli durum sistemleri için tasarlanan dışmerkezli kontrolörlerin, merkezi kontrolörlere göre daha korunumlu (conservative) olması gibi bu çalışmada tasarlanan dışmerkezli kontrolörler de, merkezi kontrolörlere göre daha korunumludur. Petri ağları ve otomata gösterimlerinin herbiri için ayrı şekilde tasarlanan merkezi kontrolörler tarafından ateşlenmesine (olmasına) izin verilen geçişlere (olaylara), Petri ağında (otomatada) sistem çıkmazı meydana gelmemesine rağmen, dışmerkezli kontrolörler tarafından izin verilmeyebilir.

Örtüşmeli ayrıştırma ve genleştirme işlemlerinin tanımlanması ile, sürekli durum sistemleri için tanımlanmış bazı özellikler (genişletme, vb.) Petri ağları ve/veya otomata gösterimleri için de tanımlanabilir. Petri ağlarında sistem çıkmazının meydana gelmesini önleyen dışmerkezli kontrolör tasarımından yararlanılarak, Petri ağının sınırlılığını, tersine dönüşebilirliğini ve canlılığını garanti edebilen dışmerkezli kontrolör tasarlanabileceği gibi bu özelliklerin tamamının aynı zamanda sağlanmasını garanti eden ve sistem çıkmazını da önleyen dışmerkezli kontrolör tasarlanabilir. Kesikli olay sistemlerinin diğer modelleme yöntemlerinden biri olan vektörel gösterim için de

benzer amaalarla bir dıřmerkezli kontrolör tasarımı geliřtirilebilir. Dıřmerkezli bir dil tasarımı yöntemi de geliřtirilebilir. Hibrid sistemler için de sistem ıkmazını önleyen veya benzer amaalara hizmet eden dıřmerkezli kontrolörler tasarlanabilir.

KAYNAKLAR

1. Y. Ho (Ed.), *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World*, A Selected Reprint Volume, The Institute of Electrical and Electronics Engineers, Inc., New York, 1992.
2. M. Zhou ve F. DiCesare, *Petri Net Synthesis For Discrete Event Control of Manufacturing Systems*. Norwell, Massachusetts: Kluwer Academic Publishers, 1993.
3. R. M. Karp ve R. E. Miller, "Parallel program schemata", *J. of Comp. & Sys. Sci.*, 3, 147-195, 1969. Kaynak gösteren [2].
4. K. Lautenbach, "Liveness in Petri nets", Internal Report, Gesellschaft für Mathematik und Datenverarbeitung, GMD-ISF 0.21/75-7-29, 1975. Kaynak gösteren [2].
5. F. Commoner, "Deadlock in Petri nets", Report CA-7206-2311, Massachusetts Computer Associates, Wakefield, MA, 1972. Kaynak gösteren [2].
6. J. Sifakis, "Structural properties of Petri nets", *Mathematical Foundation of Computer Science*, J. Winkowski (ed.); Springer-Verlag, 64, 474-483, 1978. Kaynak gösteren [2].
7. A. A. Desrochers ve R. Y. Al-Jaar, *Applications of Petri Nets in Manufacturing Systems*, The Institute of Electrical and Electronics Engineers, Inc., New York, 1995.
8. J. Proth ve X. Xie, *Petri Nets*, John Wiley & Sons, West Sussex, 1996.
9. P. J. G. Ramadge ve W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, 77, 81-98, 1989.
10. F. Lin ve W. M. Wonham, "Decentralized control and coordination of discrete-event systems with partial observation", *IEEE Transactions on Automatic Control*, 34, 1330-1337, 1990.

11. K. Cho ve J. Lim, "Mixed centralized/decentralized supervisory control of discrete event dynamic systems", *Automatica*, 35, 121-128, 1999.
12. L. E. Holloway ve B. H. Krogh, "Synthesis of feedback control logic for a class of controlled Petri nets", *IEEE Transactions on Automatic Control*, 35, 514-523, 1990.
13. L. E. Holloway ve B. H. Krogh, "On closed-loop liveness of discrete-event systems under maximally permissive control", *IEEE Transactions on Automatic Control*, 37, 692-697, 1992.
14. C. Haoxun, "Net structure and control logic synthesis of controlled Petri nets", *IEEE Transactions on Automatic Control*, 43, 1446-1450, 1998.
15. A. Guia ve F. DiCesera, "Blocking and controllability of Petri nets in supervisory control", *IEEE Transactions on Automatic Control*, 39, 818-823, 1994.
16. R. S. Sreenivas ve B. H. Krogh, "On Petri net models of infinite state supervisors", *IEEE Transactions on Automatic Control*, 37, 274-277, 1992.
17. S. R. Sreenivas, "On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modelled by controlled Petri nets", *IEEE Transactions on Automatic Control*, 42, 928-945, 1997.
18. J. Ezpeleta, J. M. Colom ve J. Martínez, "A Petri net based deadlock prevention policy for flexible manufacturing systems", *IEEE Transactions on Robotics and Automation*, 11, 173-184, 1995.
19. M. Ikeda ve D. D. Šiljak, "Overlapping decentralized control with input, state, and output inclusion," *Control Theory and Advanced Technology*, 2, 155-172, 1986.
20. A. İftar, "Overlapping decentralized dynamic optimal control," *International Journal of Control*, 58, 187-209, 1993.

21. A. İftar, "Overlapping decomposition of traffic networks," *Proceedings of the IEEE Conference on Decision and Control*, 1016-1017, Tucson, Arizona, A.B.D., 1992.
22. A. Aybar, A. İftar ve M. B. Uzman, "Decentralized routing control using overlapping decompositions," *Proceedings of the Mediterranean Electrotechnical Conference*, 671-674, Antalya, 1994.
23. A. Aybar ve A. İftar, "Bilgisayar ağlarında büzülebilir yönlendirme kontrolü tasarımı," *Yöneylem Araştırması/Endüstri Mühendisliği XVI. Ulusal Kongresi Bildirileri*, Ankara, 1994.
24. A. Aybar ve A. İftar, "Bilgisayar ağlarında içerme prensibinden yararlanılarak yönlendirme kontrolü tasarımı," *Birinci Bilgisayar Ağları Sempozyumu Bildiriler Kitabı*, 106-115, İstanbul, 1996.
25. A. Aybar ve A. İftar, "Büyük ölçekli bilgisayar ağlarında yönlendirme kontrolü tasarımı," *Elektrik-Elektronik-Bilgisayar Mühendisliği 7. Ulusal Kongresi Bildirileri*, 1-4, Ankara, 1997.
26. M. B. Uzman ve A. İftar, "Bilgisayar ağlarında örtüşmeli ayrıştırma kullanılarak bir dışmerkezli yönlendirme kontrolü uygulaması," *Otomasyon ve Bilgisayar Sempozyumu Bildirileri*, 127-134, İstanbul, 1994.
27. B. Ataşlar ve A. İftar, "Decentralized routing controller design using overlapping decompositions," *International Journal of Control*, 72, 1175-1192, 1999.
28. A. İftar ve Ü. Özgüner, "Overlapping decompositions, extensions, contractions, and stability of hybrid systems", *IEEE Transactions on Automatic Control*, 43, 928-945, 1998.
29. A. İftar, "Decentralized estimation and control with overlapping input, state and output decomposition", *Automatica*, 29, 511-516, 1993.
30. M. Heymann, "Concurrency and discrete event control", *IEEE Contr. Sys. Magazine*, 10 (4), 103-112, 1990.