

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

PETRİ AĞLARINDA DEĞİŞMEZ ANALİZİ

YÜKSEK LİSANS TEZİ
Müh. Sinan ÇAYIR

Anabilim Dalı : ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ

Programı : ELEKTRONİK MÜHENDİSLİĞİ

MAYIS 2004

PETRİ AĞLARINDA DEĞİŞMEZ ANALİZİ

YÜKSEK LİSANS TEZİ
Müh. Sinan ÇAYIR
(504021212)

Tezin Enstitüye Verildiği Tarih : 26 Nisan 2004
Tezin Savunulduğu Tarih : 21 Mayıs 2004

Tez Danışmanı : Doç.Dr. Mürvet ÜÇER
Diğer Jüri Üyeleri Doç.Dr. Ece Olcay GÜNEŞ
Doç. Dr. Sema OKTUĞ

MAYIS 2004

ÖNSÖZ

Sayın hocam Doç. Dr. Mürvet ÜÇER KIRCI'ya ve aileme teşekkür ederim. Tüm çalışmalarımda desteğini esirgemeyen Ayşegül K. YILDIRIM'a teşekkür ederim.

Nisan 2004

Sinan ÇAYIR

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER.....	iii
KISALTMALAR	v
TABLO LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	vii
ÖZET	ix
SUMMARY	xi
1. GİRİŞ	1
2. MODELLEME.....	2
2.1. Sistemlerin Özellikleri	2
2.2. Petri Ağlarının Erken Gelişimi	3
2.3. Petri Ağ Teorisinin Uygulanması	3
3. PETRİ AĞLARI TEMEL TANIMLARI	5
3.1. Petri Ağı Yapısı	5
3.2. Petri Ağ Grafları	7
3.3. Petri Ağı İşaretlemesi.....	12
3.4. Petri Ağlarının Çalışma Kuralları	14
3.5. Petri Ağı Durum Uzayları.....	18
3.6. Petri Ağları Alt Sınıfları	21
3.7. Petri Ağlarında Kapan (Trap) ve Savak (Siphon) Yapıları.....	26
3.8. Petri Ağlarını Modelleme Biçimleri	31
4. PETRİ AĞLARININ ANALİZ EDİLMESİ	33
4.1. Petri Ağlarındaki Analiz Soruları	33
4.1.1. Güvenlik (Safeness).....	33
4.1.2. Sınırlılık (Boundedness)	35
4.1.3. Tutarlılık (Conservation)	37
4.1.4. Canlılık (Liveness).....	40
4.1.5. Erişilebilirlik ve Kapsanabilirlik (Reachability and Coverability)	43
4.1.6. Ateşleme Dizileri	45
4.1.7. Eşitlik ve Altküme Problemleri	45
4.2. Analiz Yöntemleri.....	45
4.2.1. Erişilebilirlik Ağacı Yöntemi.....	46
4.2.2. EAY ve Problemlere Getirdiği Çözümler.....	51

4.2.3.	Erişilebilirlik Ağacının Kısıtlamaları.....	54
4.2.4.	Durum Denklemleri, Değişim-Olay Matrisleri.....	57
5.	P DEĞİŞMEZLERİN VE T DEĞİŞMEZLERİN ANALİZİ	64
5.1.	P – Değişmezlerin Özellikleri.....	64
5.2.	T – Değişmezlerin Özellikleri.....	67
5.3.	P – Değişmez ve T – Değişmez Tanımları	70
5.4.	P ve T Değişmezlerin Kullanım Alanları	71
5.4.1.	Petri Ağındaki Paralel ve Çevre Yapıların Sayısı.....	71
5.4.2.	SSA’nın İşaret Geçiş Bileşenlerince Kapsanması	72
5.4.3.	SSA Durum Makineleri Bileşenlerince Kapsanması.....	72
5.4.4.	PA Analizlerinin P-değişmezler Kullanılarak Kolaylaştırılması.....	73
5.4.5.	P-Değişmezler ile Haberleşme Sistemlerinde Belirsizlik Analizi	73
5.5.	P ve T Değişmez Kümelerinin Lineer Cebirsel İncelemesi.....	74
5.6.	P – Değişmezlik ve T – Değişmezlik Algoritmaları.....	83
5.6.1.	Satır Eliminasyonu Yöntemi Algoritması	83
5.6.2.	Fourier-Motzkin Yöntemi.....	84
5.6.3.	Önerilmiş Diğer Değişmez Bulma Algoritmaları.....	89
6.	PADA PROGRAMI VE ÖZELLİKLERİ.....	91
6.1.	PADA Programı Çalıştırılması	91
6.2.	PADA Programı Çalışma Ortamı	91
6.3.	PADA Programının Dosya Giriş Formatı.....	91
6.4.	PADA Programının Akışı.....	92
6.5.	Satır Eliminasyonları Algoritması	94
6.6.	PADA Programı Analiz Sonuçları.....	98
7.	SONUÇLAR VE ÖNERİLER.....	100
	KAYNAKLAR	101
	EKLER	104
	ÖZGEÇMİŞ	105

KISALTMALAR

PA	: Petri Ağı
PN	: Petri Net
DM	: Durum Makinesi
İG	: İşaret Grafı
SSA	: Serbest Seçim Ağı
GSSA	: Genişletilmiş Serbest Seçim Ağı
ASA	: Asimetrik Serbest Seçim Ağı
EAY	: Erişilebilirlik Ağacı Yöntemi
PADA	: Petri Ağları Değişmez Analizleri

TABLO LİSTESİ

Sayfa No

Tablo 3.1 Geçiş ve Yerlerin Tipik Yorumları.....	32
Tablo 6.1 PADA Programı Hesaplama Zamanları	99

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 Petri Ağlarını Kullanarak Modelleme	3
Şekil 3.1 Örnek 3.1'deki Petri Ağının Grafi	8
Şekil 3.2 Örnek 3.2'deki Petri Ağının Grafi	9
Şekil 3.3 Giriş ve Çıkış Fonksiyonlarından Petri Ağı Grafına Geçiş	10
Şekil 3.4 Giriş Çıkış Fonksiyonları Bulanacak Petri Ağı	10
Şekil 3.5 Öz Çevrenin Kaldırılması	11
Şekil 3.6 İşaretlenmiş Petri Ağı Grafi	13
Şekil 3.7 Aynı Petri Ağına Ait Farklı Bir İşaretleme	13
Şekil 3.8 Çok Sayıda Jeton Barındıran Bir Petri Ağı	14
Şekil 3.9 İşaretlemlerin Değişimi	15
Şekil 3.10 İlk İşaretleme, t_1 , t_3 ve t_4 Geçişleri Ateşlenebilir Durumda, M_0	16
Şekil 3.11 t_4 'ün Ateşlenmesi ile Oluşan İşaretleme M_1	17
Şekil 3.12 t_1 'in Ateşlenmesi ile Oluşan İşaretleme M_2	17
Şekil 3.13 t_3 'ün Ateşlenmesi ile Oluşan İşaretleme M_3	17
Şekil 3.14 İşaretli Petri Ağı	20
Şekil 3.15 Örnek Petri Ağı Grafi	20
Şekil 3.16 Şekil 3.15'teki Petri Ağının Erişilebilirlik Ağacı	21
Şekil 3.17 Öncü ve İzleyen Kümeler	22
Şekil 3.18 Petri Ağları ve Alt Sınıfları Arasındaki İlişki	24
Şekil 3.19 Çeşitli Petri Ağı Alt Kümeleri	25
Şekil 3.20 GSSA Ağından SSA Ağına Dönüşüm	25
Şekil 3.21 Örnek Petri Ağı A1	28
Şekil 3.22 Örnek Petri Ağı A2	28
Şekil 4.1 Güvenli Olmayan Petri Ağı	34
Şekil 4.2 Güvenli Petri Ağı	35
Şekil 4.3 Sınırsız Petri Ağı	37
Şekil 4.4 Sınırlı Hale Getirilmiş Petri Ağı	37
Şekil 4.5 Tutarsız Petri Ağı	38
Şekil 4.6 Tutarlı Hale Getirilmiş Petri Ağı	39
Şekil 4.7 Tutarsız Petri Ağı	40
Şekil 4.8 Sistemin Petri Ağı İfadesi	41
Şekil 4.9 Petri Ağındaki Çeşitli Canlılık Seviyeleri	43
Şekil 4.10 Mutual Exclusion Problemine Çözüm Olan Petri Ağı	44
Şekil 4.11 İşaretli Bir Petri Ağı	46
Şekil 4.12 Erişilebilirlik Ağacının Oluşturulmasında İlk Adım	46
Şekil 4.13 Erişilebilirlik Ağacının Oluşturulmasında İkinci Adım	47
Şekil 4.14 Erişilebilirlik Ağacının Oluşturulmasında Üçüncü Adım	47
Şekil 4.15 Sonsuz Erişilebilirlik Ağacına Sahip Bir Petri Ağı	48
Şekil 4.16 Şekil 4.11'deki Petri Ağının Erişilebilirlik Ağacı	50
Şekil 4.17 Erişilebilirlik Ağacı İncelenecek Petri Ağı	50
Şekil 4.18 Şekil 4.17'deki Petri Ağının Erişilebilirlik Ağacı	51

Şekil 4.19 Erişilebilirlik Ağacı Kullanılarak Petri Ağının Sınırlılığını Belirleme	52
Şekil 4.20 İncelenecek Petri Ağı.....	54
Şekil 4.21 Şekil 4.20'deki Petri Ağının Erişilebilirlik Ağacı	54
Şekil 4.22 Şekil 4.24'te Erişilebilirlik Ağacı Görülen Petri ağı	55
Şekil 4.23 Şekil 4.24'te Erişilebilirlik Ağacı Görülen Petri Ağı	55
Şekil 4.24 Şekil 4.22 ve Şekil 4.23'teki Petri Ağlarının Erişilebilirlik Ağacı	55
Şekil 4.25 Muhtemel Kilitlenme Barındıran Petri Ağı.....	56
Şekil 4.26 Kilitlenme Barındırmayan Petri Ağı	56
Şekil 4.27 Şekil 4.25 ve Şekil 4.26'daki Petri Ağlarının Erişilebilirlik Ağacı	57
Şekil 4.28 Durum Denklemleri İncelenecek Petri Ağı Grafı	59
Şekil 4.29 Durum Denklemleri İncelenen Petri Ağı Grafı	61
Şekil 4.30 Durum Denklemleri Çözümünü Yapılan Petri Ağı Grafı.....	62
Şekil 5.1 İncelenecek Petri Ağı Grafı	72
Şekil 5.2 Vektör Uzayları Arasındaki İlişki.....	77
Şekil 5.3 İncelenecek Petri Ağı.....	85
Şekil 6.1 Örnek Petri Ağı Giriş Dosyası.....	92
Şekil 6.2 PADA Programında P Değişmez Analizi.....	93
Şekil 6.3 PADA Programında T Değişmez Analizi	94
Şekil 6.4 Satır Eliminasyonu Uygulama Algoritması.....	96
Şekil 6.5 Satır Eliminasyonu Uygulama Algoritma Adımları	97
Şekil 6.6 Satır Eliminasyonu Fonksiyonu.....	97
Şekil 6.7 Satır Değişimi Fonksiyonu	98
Şekil 6.8 Sütunda Sıfır Kontrol Fonksiyonu.....	98

PETRİ AĞLARINDA DEĞİŞMEZ ANALİZİ

ÖZET

Bu çalışmada, ayrık olaylı ve eş zamanlı sistemleri modelleme ve analiz etme konusunda başarılı araçlardan biri olan Petri ağı sistemleri ele alınmış, Petri ağı yapısal özelliklerinden olan değişmezler konusu ayrıntılı olarak incelenmiş, değişmezlerin bulunması konusunda bir algoritma önerilmiş ve bu algoritmayı uygulayan bir bilgisayar programı geliştirilmiştir.

İkinci bölümde modelleme konusu ele alınmış, Petri ağlarının modelleme döngüsü içinde nerede yer aldığı ve nasıl kullanılabileceği tartışılmış, Petri ağı teorisinin erken gelişiminden kısaca bahsedilmiştir.

Üçüncü bölümde Petri ağları tanımlanmış, yapı taşları olan yer, geçiş ve yönlü ağırlıklı dallar ve bunların grafiksel gösterimi verilmiş, sistemlerin dinamik davranışlarını modelleyen jetonlar ve geçişlerin ateşlenmesi ele alınmış, ilk işaretleme, erişilebilirlik, erişilebilirlik ağacı gibi yapılar tanımlandıktan sonra Petri ağı alt sınıfları incelenmiştir. Bölümün sonunda tanımları desteklemek amacıyla çeşitli sistemlerin Petri ağlarında nasıl modellenebileceği ele alınmıştır.

Dördüncü bölümde ilk olarak Petri ağlarının analizinde karşılaşılabilecek güvenlik, sınırlılık, canlılık, erişilebilirlik, alt küme gibi temel sorular incelenmiş, ardından bu soruları çözebilecek yöntemlerden biri olan erişilebilirlik ağacı yöntemi ve algoritması tanımlanmış, erişilebilirlik ağacının çözebildiği ve çözemediği sorular incelenmiş, en son olarak Petri ağlarının özelliklerini incelemek amacıyla kullanılan durum denklemleri yöntemi ele alınmış ve buna bağlı olarak olay-değişim matrisi tanımlanmış, bu yöntemin güçlü ve zayıf yönlerine değinilmiştir.

Beşinci bölümde Petri ağlarının yapısal özelliklerinden olan ve olay matrisinden elde edilen Petri ağı yer (P) ve geçiş (T) değişmezleri tanımlanmış, Petri ağı analizinde değişmezlerin kullanılabildikleri alanlar incelenmiş, değişmezlerin lineer cebirsel

anlamı irdelenmiş, buna dayanan ve Petri ağının değişmezlerini bulmak amacıyla kullanılabilecek bir teorem ve teoreme dayanan bir algoritma önerilmiştir. Ardından Petri ağı değişmez analizlerinde en fazla kullanılan Fourier-Motzkin algoritması tanımlanmış, son bölümde ise farklı değişmez bulma algoritmaları, değişmez analizinde farklı yaklaşımları gösterebilmek için ele alınmıştır.

Altıncı bölümde önerilen teoreme dayanan Petri Ağı Değişmez Analizi (PADA) programı tanıtılmış, programın akış şemaları ve algoritmaları ele alınmış, ardından çeşitli boyutlardaki Petri ağlarının PADA programı kullanılarak yapılan değişmez analizlerinin hesaplama zamanları verilmiştir.

INVARIANT ANALYSIS IN PETRI NETS

SUMMARY

Petri Nets, one of the best defined approaches to modeling of discrete and concurrent systems, were researched during this study. The scope of this thesis is to study properties of Petri Nets' invariants, to present an algorithm of finding the invariants and to develop a computer program implementing the algorithm.

Modeling a system and the value of using Petri Nets in modeling systems were presented. Then, the focus of the study was to review how and where Petri Nets can function in a model. In addition to all, early development of Petri Net Theory was mentioned briefly.

The terminology of Petri Nets is defined in the third chapter: Basic definitions of Petri Nets, their building blocks; places, transitions and weighted directed arcs; then, the definitions of tokens and transition firing that simulate dynamic behavior of Petri Nets were presented. Essential concepts such as initial marking, reachability, and reachability tree, subclasses of Petri Nets (i.e. State Machine and Marked Graphs) were explained. In order to strengthen the definitions some typical and useful interpretations of transitions and places in system modeling were illustrated.

Six fundamental problems which may arise during analysis of Petri Nets safeness, boundedness, liveness, reachability, conservation and finding sub-networks were clarified. Then, Reachability Graph Method, one of the most common methods used to solve these problems was defined and an algorithm implementing the method was introduced. Finally, an additional method based on the incidence matrix and the state equations was proposed. Nevertheless, strengths and weaknesses of the latter method were presented. Two methods were took up in terms of the problems which can/cannot be solved by each method

The structural properties of the Petri Nets, place (P) and transition (T) invariants were examined on the basis of the definitions and possible uses of these invariants. Then, linear algebraic properties of the invariants were proved; a theorem and an algorithm to find the invariants were proposed. A well-known method to find Petri Net invariants; Fourier-Motzkin method was studied through examples. Finally different algorithms of invariant analysis were briefly mentioned to envision a range of approaches.

Final section of this study was dedicated to develop a computer program based on the theorem proposed in chapter 5. The computer program; Petri Net Invariant Analysis (PNIA) was developed. The program's workflows and algorithms were presented. In conclusion, performance analysis of the program for different sized Petri Nets were performed and tabulated.

1. GİRİŞ

Elektronik sistemlerin tasarımı genel olarak dört temel adımdan oluşur; sistemin tanımlanması, sistemin modellenmesi, modelin analizi ve indirgenmesi, indirgenmiş modelin elektronik sistem olarak gerçekleştirilmesi. Bu adımlardan her biri farklı araçlar kullanılarak gerçekleştirilebilir. Kullanılabilecek araçlardan biri Petri ağlarıdır. Petri ağları tasarım adımlarından modelleme ile analiz ve indirgemeyi kapsar.

Petri ağları kullanılarak ayırık olaylı ve eş zamanlı her alandan sistemler modellenebilir. Modellenen sistemler arasında senkron ve asenkron devreler, haberleşme sistemleri, haberleşme protokolleri, bilgi sistemleri, sebep sonuç ilişkisine dayanan sistemler, çok işlemcili sistemler, iş akışları, biyolojik sistemler, kimyasal reaksiyonlar, ekolojik sistemler, stokastik sistemler, otomatik üretim sistemleri, gerçek zamanlı sistemler gösterilebilir.

Petri ağları ile ilgili birçok yöntem araç ve algoritma geliştirilmiş olmasına karşın bunların elektronik sistemlerin tasarımına yönelik alanlarda kullanımları henüz standartlaşmayı tamamlamamıştır. Bu yüzden Petri ağlarının elektronik sistemlerin tasarımında kullanımına yönelik çalışmalar yapılabilir.

Petri ağları sistemlerdeki nedensel ilişkileri, yarış durumlarını ve seçim durumlarını kolay ve doğru modelleyip çeşitli analizler yapılmasını olanaklı kılmaktadır. Bu analizler arasında canlılık, sınırlılık, güvenlik, tutarlılık vardır. Kilitlenmesiz bir elektronik sistem tasarlanmak istendiğinde bu durumda modellenen sisteme ait Petri ağının her durumda canlı kalması gerekir, bazı durumlarda canlılığın bozulması durumunda model üzerinde gerekli müdahaleler yapılarak sistemin her zaman canlı kalması dolayısıyla kilitlenmesiz olması sağlanabilir. Petri ağı değişmezleri sistemlerin tasarımında bu tarz yapısal özelliklerini ortaya çıkarmak amacıyla kullanılabilir.

2. MODELLEME

Petri ağlarının uygulaması modelleme ile olmaktadır. Birçok araştırma alanında incelemeler doğrudan olaylar üzerinde yapılmaz, bunun yerine dolaylı olarak modeller kullanılarak analizler gerçekleştirilir. Bir model incelenmekte olan nesnenin veya sistemin önemli olduğu hissedilen yönlerinin bir gösterimidir. Bu gösterim çoğu zaman matematiksel ifadeler ile yapılmaktadır. Elektrik, Elektronik, Astronomi, Nükleer Fizik, Sosyoloji, Biyoloji ve daha birçok alanda modelleme teknikleri kullanılır [1].

Çoğu modelleme yönteminde matematik kullanır. Birçok fiziksel olayın özellikleri sayılarla, eşitliklerle veya eşitsizliklerle ifade edilir. Özellikle doğa bilimlerinde ve mühendislikte kütle, konum, momentum, ivme ve kuvvetler matematiksel eşitliklerle ifade edilebilir. Modellemenin başarılı olabilmesi için incelenen olaya ve modelleme tekniğine hakim olmak gerekir. Matematikteki birçok kavram diğer bilimlerdeki olayları modellemek amacıyla geliştirilmiştir.

2.1. Sistemlerin Özellikleri

Bir sistem birden fazla parçanın bir araya gelmesinden oluşur. Bu parçalar birbirileri ile karmaşık etkileşimler kurarlar. Bazen bu parçaların eş zamanlı ve paralel çalışması istenebilir. Bir parçadaki işlem, başka bir parçadaki işlem ile aynı zamanda gerçekleşebilir. Bir bilgisayar sisteminde diskler, yazıcılar, ağ bağlantı kartları ve diğer bazı çevresel birimler kontrollü olarak aynı anda çalışabilirler. Benzer şekilde bir ekonomik sistemde üreticiler belli ürünleri üretebilir, perakendeciler başka ürünleri satabilir ve tüketiciler de tamamen farklı bir ürünü aynı zamanda kullanabilirler.

Bir sistemin doğasında var olan ve eş zamanlı gerçekleşen olaylar, modellemede güçlükler oluşturur. Sistemlerin parçaları etkileştiklerinden aralarında uyumluluk (senkronizasyon) olmalıdır. Değişimin gerçekleşeceği zamanlarda sistemin parçaları eşzamanlı davranmak zorunda olabilir. Bu durum bir parçanın başka bir parçayı

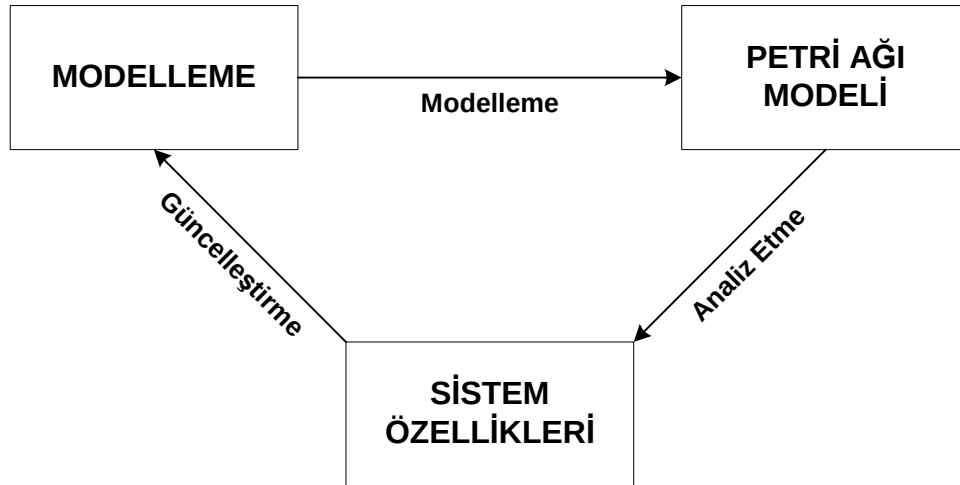
beklemesine sebep olur. Farklı parçaların zamanlamasını ve etkileşimlerini tanımlamak zor olabilir.

2.2. Petri Ağlarının Erken Gelişimi

Petri ağları, özellikle etkileşen ve eş zamanlı sistemleri modellemek amacıyla tasarlanmışlardır. Petri ağları ilk defa Carl Adam Petri'nin "Kommunikaton mit Automaten" (Communication with Automata) isimli doktora tezinde geliştirilmiştir. Petri çalışmasında bir bilgisayarın asenkron parçaları arasında bir iletişim teorisi geliştirmiştir. Çalışmasında özellikle olaylar arasındaki nedensel ilişkinin üzerinde durulmaktadır.

2.3. Petri Ağ Teorisinin Uygulanması

Petri ağlarını sistemlerin tasarım ve analizinde kullanmanın birkaç yöntemi vardır. İlk yöntemde bilinen tasarım yöntemleri kullanılarak sistem tanımlanır. Bu sistem daha sonra Petri ağı olarak modellenir ve neticede bu Petri ağı incelenir. Analizde görülen hatalardan yola çıkarak tasarım yeniden ve yeniden incelenir. Bu adımlar sorunsuz çalışma elde edilene kadar tekrarlanır. Bu yöntem Şekil 2.1'de anlatılmaktadır.



Şekil 2.1 Petri Ağlarını Kullanarak Modelleme

Yukarıda anlatılan klasik tasarım yönteminde; tasarlanan sistem ile Petri ağı modeli arasında sürekli bir uyumlaştırma gerekir. Bir diğer yöntemde ise tüm tasarım ve tanımlama işlemleri Petri ağları üzerinde yürütülür. Analiz yöntemleri ise sadece

hatasız bir Petri ağı tasarımı bulmaya yöneliktir. Bundan sonraki aşama Petri ağını çalışan gerçek bir sisteme dönüştürmek olacaktır. Bu işleme gerçekleştirme denir.

Her iki yöntem de sistem tasarımı açısından farklı problemlere sebep olur. İlk yöntemde sistemleri Petri ağına dönüştüren modelleme teknikleri, ikinci yöntemde ise Petri ağlarını gerçek sistemlere çeviren tasarım teknikleri geliştirilmelidir. Her iki yöntemde de ortak nokta, Petri ağlarının özelliklerini çözecek analiz yöntemlerini geliştirmektir. Bu durumda ilk önce Petri ağlarının özellikleri iyi bilinmelidir.

3. PETRİ AĞLARI TEMEL TANIMLARI

Petri ağı bir grafiksel ve matematiksel modelleme aracıdır. Daha geniş bir tanımlamayla; Petri ağları eş zamanlı sistemleri modelleme, tanımlama, analiz etme ve sentezlemede kullanılan bir biçimsel grafik aracıdır. Eş zamanlı ve ayrık olaylı sistemler Petri ağları kullanılarak analiz edilebilirler. Petri ağlarını kullanarak birçok sistemde yer alan bilgi akışı soyut ve biçimsel olarak modellenenir. [1]

Bu bölümde Petri ağlarının temel tanımları ve özellikleri incelenecektir.

3.1. Petri Ağı Yapısı

Bir Petri ağı dört temel parçadan oluşur; **P** yerler kümesini, **T** geçişler kümesini, **F** bağıntısı yer ve geçiş kümeleri arasındaki ilişkiyi, **W** yerleri ve geçişleri birbirine bağlayan dalların ağırlıklarını tanımlar.

Bir Petri ağı yerler, geçişler, ilişkiler bağıntısı ve ağırlık fonksiyonu ile birlikte tanımlanır.

Tanım 3.1: Petri Ağının Temel Tanımı

Bir Petri ağı yapısı, **C**, dört bileşenli bir kümedir, **C = (P, T, F, W)**. [2]

P = {**p**₁, **p**₂, ..., **p**_n} sonlu sayıda yer olan kümeyi, $n \geq 0$

T = {**t**₁, **t**₂, ..., **t**_m} sonlu sayıda geçiş olan kümeyi, $m \geq 0$

F = (**P** x **T**) $\cup$ (**T** x **P**) dallardan oluşan kümeyi (akış bağıntısı),

W: **F** $\rightarrow$ {1, 2, 3, ...} ağırlık fonksiyonunu tanımlar.

Yerler kümesi ve geçişler kümesi ayrıktır, **P** $\cap$ **T** = $\emptyset$.

Yerler kümesi ve geçişleri kümesinin bileşkesi boş küme değildir, **P** $\cup$ **T** $\neq \emptyset$.

P kümesinin eleman sayısı, n , **T** kümesinin eleman sayısı da m 'dir. **P**'deki herhangi bir eleman **p**_{*i*}, $i = 1, 2, \dots, n$, **T**'deki herhangi bir eleman **t**_{*j*}, $j = 1, 2, \dots, m$ ile ifade edilecektir.

Örnek 3.1. Aşağıda bir **C = (P, T, F, W)** Petri ağ yapısı örneklenmiştir,

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

$$F = \{(t_1, p_1), (t_2, p_2), (t_2, p_3), (t_2, p_5), (t_3, p_3), (t_4, p_4), (p_2, t_1), (p_3, t_1), (p_5, t_1), (p_5, t_2), (p_4, t_3), (p_2, t_4), (p_3, t_4)\}$$

$W = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ – toplam dal sayısı kadar (13) dalların ağırlıkları.

Örnek 3.2. Aşağıda dört yer ve üç geçişten oluşan ikinci bir $C = (P, T, F, W)$ Petri ağ yapısı örneklenmiştir,

$$P = \{p_1, p_2, p_3, p_4\}$$

$$T = \{t_1, t_2, t_3\}$$

$$F = \{(t_1, p_1), (t_2, p_2), (t_2, p_4), (t_3, p_3), (p_2, t_1), (p_3, t_1), (p_1, t_2), (p_1, t_3), (p_4, t_3)\}$$

$$W = (1, 1, 1, 1, 1, 1, 1, 1)$$

Tanım 3.2: Giriş ve Çıkış Fonksiyonları

Giriş ve çıkış fonksiyonları bir Petri ağında bulunan geçişleri ve yerleri ilişkilendirir. Giriş fonksiyonu **I**, çıkış fonksiyonu da **O** ile ifade edilir. I giriş fonksiyonu bir t_j geçişinden ilişkili olduğu yerlere **I(t_j)**'ye bir eşlemedir. Bunlara o geçişin giriş yerleri olarak bakılabilir. Benzer şekilde **O** çıkış fonksiyonu t_j geçişinden çıkan tüm **O(t_j)** çıkış yerlerine bir eşlemedir. Bu eşlemelerde yer alan değerler geçiş ile yer arasındaki dalın ağırlığı kadar değer alır. [1]

I: $T \rightarrow P$, bir geçişe gelen yerlerin eşleşmesi ile oluşan giriş fonksiyonunu tanımlar.

O: $T \rightarrow P$, bir geçişten ulaşılan yerlerin eşleşmesi ile oluşan çıkış fonksiyonunu tanımlar.

Örnekleme yapılırsa; t_3 geçişine p_1 ve p_4 'ten giriş dalları olsun. Bu dalların ağırlıkları sırasıyla 1 ve 2 olarak tanımlansın. Bu durumda $I(t_3) = \{p_1, 2p_4\}$ olarak tanımlanacaktır. Aşağıdaki örneklerde giriş ve çıkış fonksiyonları ile Petri ağları tanımlanmaktadır.

Örnek 3.3. Örnek 2.1'deki Petri ağının giriş ve çıkış fonksiyonları kullanılarak $C = (P, T, I, O)$ olarak tanımlanması,

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

$$I(t_1) = \{p_1\}, I(t_2) = \{p_2, p_3, p_5\}, I(t_3) = \{p_3\}, I(t_4) = \{p_4\}$$

$$O(t_1) = \{p_2, p_3, p_5\}, O(t_2) = \{p_5\}, O(t_3) = \{p_4\}, O(t_4) = \{p_2, p_3\}$$

Örnek 3.4. Örnek 2.2'deki Petri ağının giriş ve çıkış fonksiyonları kullanılarak $C = (P, T, I, O)$ olarak tanımlanması,

$$P = \{p1, p2, p3, p4\}$$

$$T = \{t1, t2, t3\}$$

$$I(t1) = \{p1\}, I(t2) = \{p2, p4\}, I(t3) = \{p3\}$$

$$O(t1) = \{p2, p3\}, O(t2) = \{p1\}, O(t3) = \{p1, p4\}$$

Tanım 3.1'de bir Petri ağı $C = (P, T, F, W)$ olarak tanımlanmıştı. Örnek 3.3 ve 3.4 incelendiğinde aynı Petri ağlarının P, T kümelerinin yanı sıra I, O giriş ve çıkış fonksiyonlarının eklenmesiyle tanımlanabilmektedir, $C = (P, T, I, O)$. Her iki tanım da aynı yapıyı tanımladığına göre bu durumda aralarında doğrudan bir ilişki bulunmalıdır. Birinci ve ikinci tanım arasında P ve T kümeleri ortak olduğundan, I giriş fonksiyonu ve O çıkış fonksiyonunun, F ve W 'nin yerini aldığını düşünülebilir. $C = (P, T, I, O)$ tanımında yer alan giriş ve çıkış fonksiyonlarını kullanarak, F ilişki bağıntısı ve W ağırlık fonksiyonu elde edilebilir. Bunun tersi de doğrudur. Sonuç itibarıyla $C = (P, T, I, O)$ ve $C = (P, T, F, W)$ birbirinin eşdeğeri olan tanımlardır.

3.2. Petri Ağ Grafları

Bir Petri ağının grafiksel gösterimi, Petri ağı teorisinin kavramlarını göstermesi bakımından çok daha kullanışlıdır. Bir Petri ağı grafi, Petri ağı yapısını gösteren iki elemanlı, yönlü ve ağırlıklı bir graftır.

Bir Petri ağı yapısı yerler ve geçişlerden oluşur. Buradan hareketle bir Petri ağı grafinda iki tür düğüm bulunması gerektiği söylenebilir. Bir daire (**O**) yere işaret ederken, bir çizgi veya dikdörtgen bir şekil (**|**, **⊞**) geçişleri temsil eder. Daireler yerleri temsil ettiğinden bunlar *yer* adını alır, benzer şekilde çizgiler veya dikdörtgenler de *geçiş* adını alır.

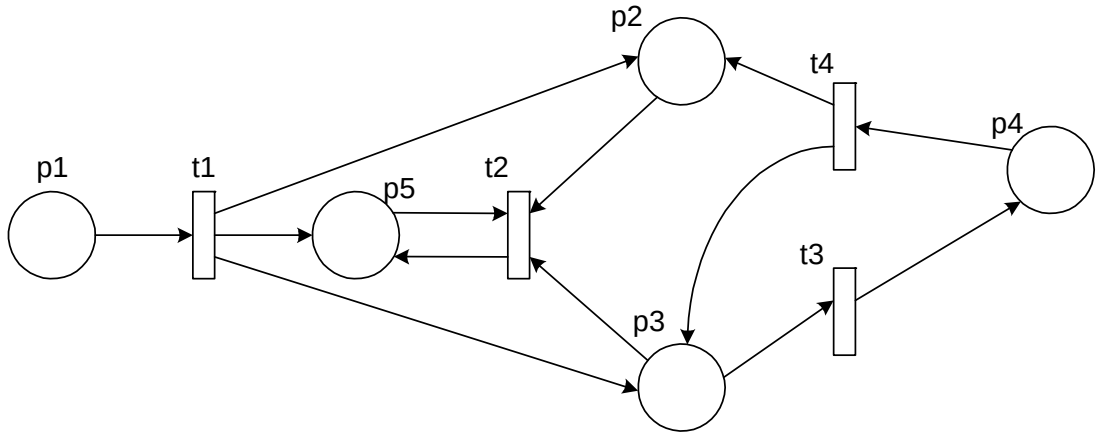
Yönlü çizgiler (dallar) yerleri ve geçişleri birbirine bağlar. Bir p_i yerinden t_j geçişine yönelen bir dal o geçişe bir giriş yeri olduğunu işaret eder. Benzer şekilde bir geçişten yerlere yönelen dallar da o geçişin çıkışlarını ifade eder. Bir Petri ağında bir düğümden bir diğerine çizilen dallar farklı ağırlıklarda olabileceğinden ağırlıklı bir graftır. Yine bu dallar yönlü olduğundan oluşan yapı *yönlü ağırlıklı graf* ismini alır. Bir Petri ağı grafinin düğümleri yerler ve geçişler olmak üzere iki kümeye ayrılır,

öyle ki her bir kümenin bir elemanından diğer kümenin bir elemanına yönelen yönlü dallar tanımlanabilir. Bu yapıya *iki parçalı yönlü ağırlıklı graf* denir. Kısaca bileşenlerini tanımlanan bu yapılara *Petri ağı grafı* ismi verilir.

Tanım 3.3: Petri Ağı Grafı

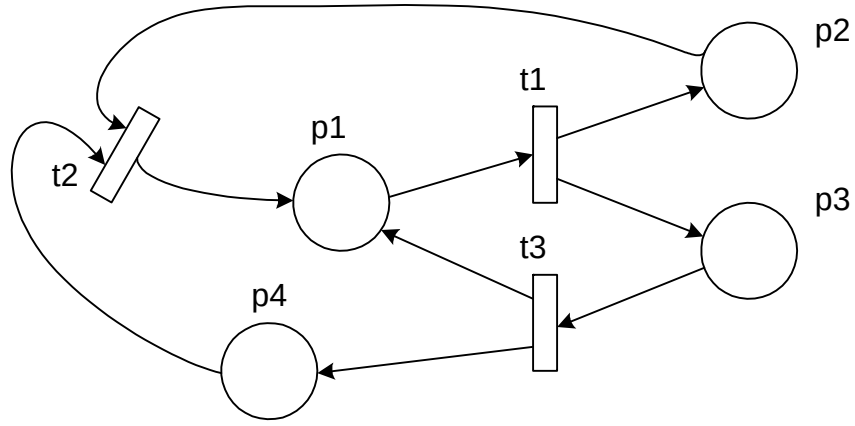
Bir Petri ağı grafı iki parçalı yönlü ve ağırlıklı bir graftır. $G = (V, A)$ 'da $V = \{v_1, v_2, \dots, v_s\}$ bir bileşenler kümesi, $A = \{a_1, a_2, \dots, a_s\}$ ise yönlü dallardan oluşan bir küme olsun öyle ki $a_i = (v_j, v_k)$ ve $v_j, v_k \in V$. V kümesi ayrık P ve T alt kümelerine bölünebilir, öyle ki $V = P \cup T$, $P \cap T = \emptyset$, ve her yönlü $a_i \in A$ dalında $a_i = (v_j, v_k)$ ise bu durumda ya $v_j \in P$ ve $v_k \in T$ 'dir veya $v_j \in T$ ve $v_k \in P$ 'dir. v_j ve v_k elemanlarının her ikisi aynı anda aynı kümenin elemanı olamaz (P veya T).

Örnek 3.5. Örnek 3.1'de yer alan Petri ağı yapısının grafı Şekil 3.1'de görülmektedir.



Şekil 3.1 Örnek 3.1'deki Petri Ağının Grafı

Örnek 3.6. Örnek 3.2'de yer alan Petri ağı yapısının grafı Şekil 2.2'de görülmektedir.



Şekil 3.2 Örnek 3.2’deki Petri Ağının Grafi

Tanım 3.4: Petri Ağı Grafi ve Petri Ağı Yapısı Arasındaki İlişki

Bileşenler kümesi $V = P \cup T$ olarak tanımlansın. A da bir yönlü dallar bağıntısı olarak tanımlansın, öyle ki her $p_i \in P$ ve $t_j \in T$ için,

$\#((p_i, t_j), A) = \#(p_i, I(t_j))$, burada $\#$ (diyez) işareti dal ağırlıklarını ifade etmektedir

$$\#((t_j, p_i), A) = \#(p_i, O(t_j))$$

bu durumda $G(V, A)$ Petri ağı grafi, $C = (P, T, I, O)$ Petri ağı yapısına eşdeğerdir.

Bir Petri ağı grafindan Petri ağı yapısı, Petri ağı yapısından da Petri ağı grafi elde edilebilir.

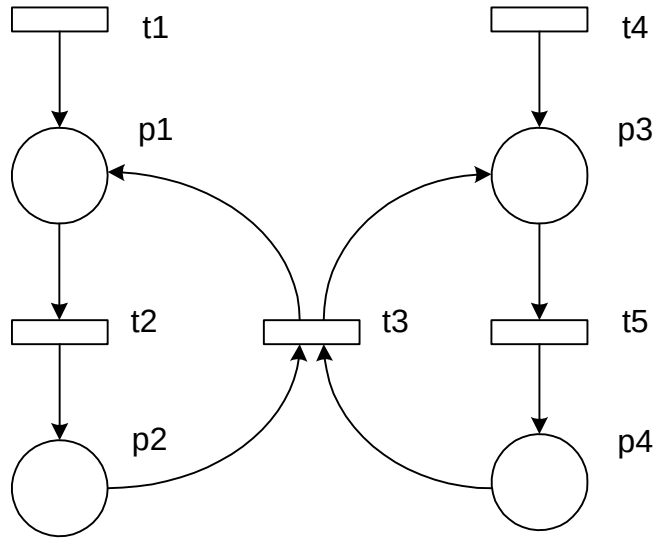
Örnek 3.7. Dört yer ve beş geçişten oluşan bir Petri ağının giriş ve çıkış fonksiyonları aşağıdaki gibi tanımlanmıştır.

$$P = \{p1, p2, p3, p4\}, T = \{t1, t2, t3, t4, t5\}$$

$$I(t1) = \{\}, I(t2) = \{p1\}, I(t3) = \{p2, p4\}, I(t4) = \{\}, I(t5) = \{p3\}$$

$$O(t1) = \{p1\}, O(t2) = \{p2\}, O(t3) = \{p1, p3\}, O(t4) = \{p3\}, O(t5) = \{p4\}$$

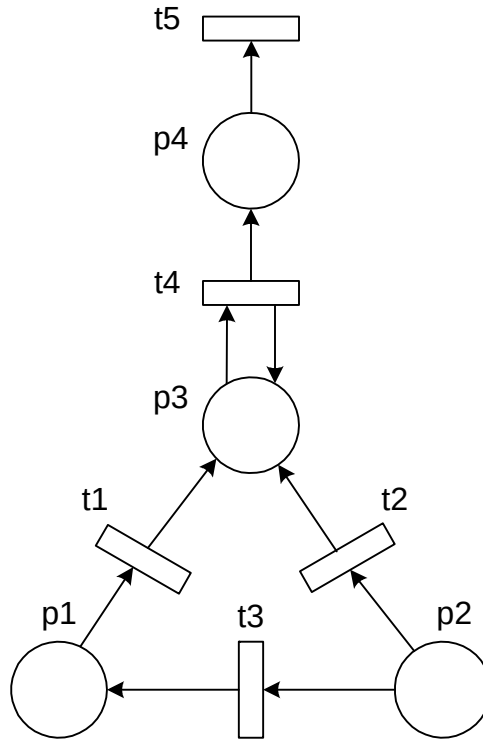
Bu bilgiler kullanılarak Petri ağına ilişkin graf Şekil 3.3’te gibi elde edilir,



Şekil 3.3 Giriş ve Çıkış Fonksiyonlarından Petri Ağı Grafına Geçiş

Benzer şekilde grafı bilinen bir Petri ağından o ağa ilişkin giriş, çıkış fonksiyonları elde edilebilir.

Örnek 3.8. Şekil 3.4’te görülen Petri ağı grafından yararlanarak giriş ve çıkış fonksiyonlarının bulunması.



Şekil 3.4 Giriş Çıkış Fonksiyonları Bulunacak Petri Ağı

Şekil 3.4'teki Petri ağı grafında dört yer ve beş adet geçiş görülmektedir. Bunlar arasındaki yönlü dallardan yararlanarak giriş ve çıkış fonksiyonları aşağıdaki gibi bulunur,

$$P = \{p1, p2, p3, p4\}$$

$$T = \{t1, t2, t3, t4, t5\}$$

$$I(t1) = \{p1\}, I(t2) = \{p3\}, I(t3) = \{p2\}, I(t4) = \{p3\}, I(t5) = \{p4\}, O(t1) = \{p3\}$$

$$O(t2) = \{p2\}, O(t3) = \{p1\}, O(t4) = \{p3, p4\}, O(t5) = \{\}$$

Tanım 3.5. Sıradan (Ordinary) Petri Ağları

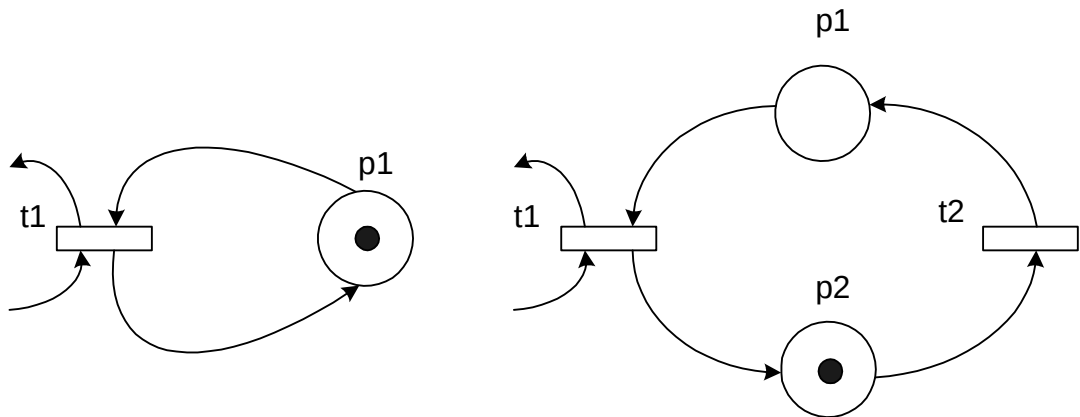
Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu ağdaki birbirini izleyen her $p_i \in P$ ve $t_j \in T$ çiftinde aradaki dalın ağırlığı bir ise bu durumda bu *sıradan* Petri ağı denir.

$$\#(p_i, I(t_j)) = 1 \text{ ve } \#(p_i, O(t_j)) = 1$$

Tanım 3.6. Temiz (Pure) Petri Ağları

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu ağda bir $p_i \in P$ yeri, $t_j \in T$ geçişin hem giriş fonksiyonunda hem de çıkış fonksiyonunda yer alıyorsa bu durumda ağda bir öz çevre vardır denir. Hiçbir özçevre barındırmayan ağlara *temiz Petri ağı* denir.

Öz çevre barındıran ağlar bir yer-geçiş çifti eklenerek temiz hale getirilebilir. Şekil 3.5'te, bir yer-geçiş çiftinin eklenmesiyle öz-çevre barındıran bir ağın nasıl temiz bir ağa dönüştürülebileceği görülmektedir.



Şekil 3.5 Öz Çevrenin Kaldırılması

3.3. Petri Ağı İşaretleme

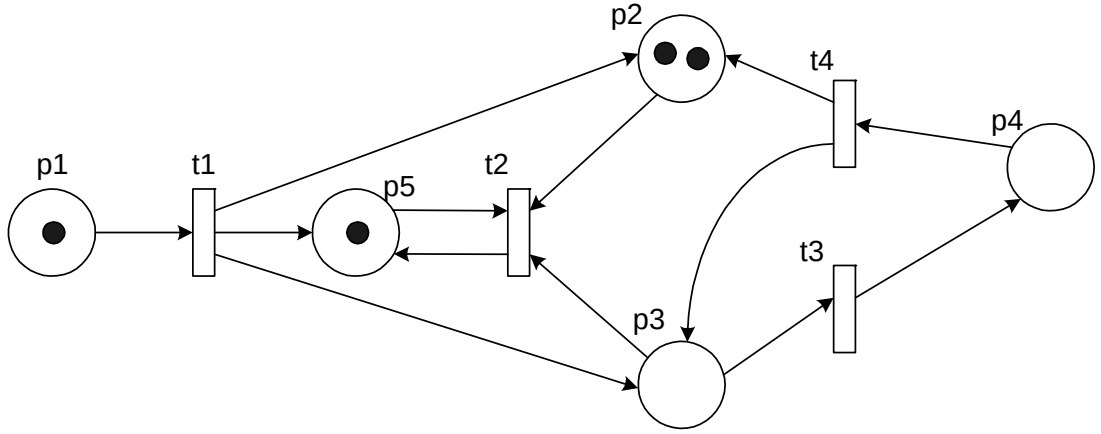
Bir M işaretleme bir Petri ağındaki yerlere jetonların atanması işlemidir. Atanan jetonların Petri ağındaki yerlerde bulundukları düşünülebilir. Yer, geçiş ve yönlü dallar gibi jetonlar da Petri ağlarındaki temel kavramlardan biridir. Bir Petri ağının çalışması esnasında jetonların sayısı ve yerleri değişir. Jetonlar bir Petri ağının çalışmasını tanımlamak amacıyla kullanılırlar. Bu şekilde Petri ağı statik bir yapıdan *dinamik bir sisteme* dönüştürülmüş olur. İncelenen sistemlerin yapıları modellenirken Petri ağı yapısına dönüştürülür, buna karşın sistemlerin dinamik davranışları işaretleme kullanılarak Petri ağında modellenir. Petri ağı yapıları tek başına sistemlerin davranışlarını incelemeye yetersiz kalırlar, bu yüzden sistemin dinamik davranışları ile incelenmesi önemlidir.

Tanım 3.7: Petri Ağı İşaretleme (M)

$C = (P, T, F, W)$ Petri ağındaki **M** işaretleme, **P** yerler kümesinden N^+ negatif olmayan tamsayılar kümesine bir fonksiyondur,

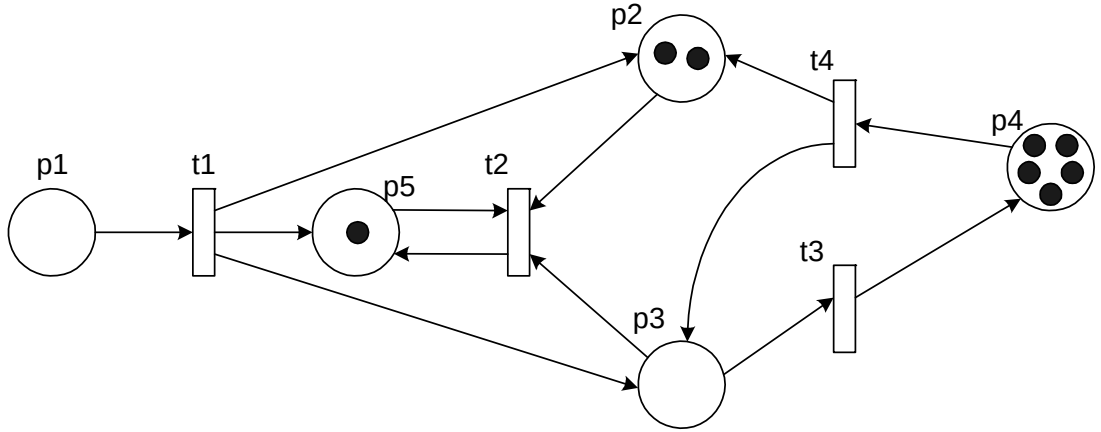
$$M: P \rightarrow N^+$$

M işaretleme n-boyutlu bir sütun vektör olarak tanımlanabilir, $M = [m_1, m_2, \dots, m_n]^T$, bu tanımda $n = |P|$ ve her $m_i \in N$, $i = 1, 2, \dots, n$ 'dir. **M** vektörü, Petri ağında yer alan her p_i yerinde bulunan jetonların sayısını belirtir. Bir p_i yerinde bulunan jetonların sayısı $M(p_i)$ veya $M(i)$ ile ifade edilir, $i = 1, 2, \dots, n$. İşaretli bir Petri ağı $P.N = (C, M)$ ile ifade edilir ve $C = (P, T, F, W)$ bir Petri ağı yapısı ile M işaretlemeinden oluşur. Bazen bir Petri ağı $C = (P, T, F, W, M)$ olarak da ifade edilir. Petri ağı grafında jetonlar yerlerin içinde noktalar ($\bullet$) ile temsil edilir. Petri ağındaki ilk işaretleme M_0 ile gösterilir. Şekil 3.6'da Örnek 3.5'te verilen Petri ağı grafi üzerinde işaretleme yapılmıştır.



Şekil 3.6 İşaretlenmiş Petri Ağı Grafi

Burada ilk işaretleme $M_0 = (1, 2, 0, 0, 1)$ ile ifade edilir. Şekil 3.7'de aynı Petri ağı grafına uygulanmış farklı bir işaretleme görülmektedir.

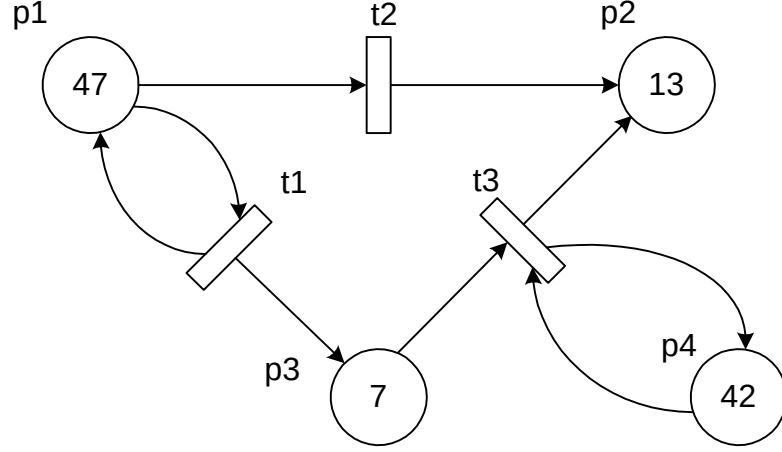


Şekil 3.7 Aynı Petri Ağına Ait Farklı Bir İşaretleme

Bu durumda ilk işaretleme $M_0 = (0, 2, 0, 5, 1)$ ile ifade edilir.

Bir Petri ağındaki bir yere atanabilecek jetonların sayısı sınırsız olduğundan, o Petri ağında sonsuz işaretleme bulunur. n sayıdaki yerden oluşan bir Petri ağında $\mathbb{N}^n$ farklı işaretleme vardır. Bu işaretlemelerin oluşturduğu küme sonsuz olmasına rağmen numaralandırılabilir. Bunun anlamı her işaretlemeye karşı gelen bir tamsayının var olmasıdır.

Örnek 2.9. Yerlerde bulunan jetonların sayısı çok fazla olduğunda jetonları ifade etmek için noktaların yerine sayılar kullanılmaktadır. Aşağıdaki şekilde ilk işaretlemesi $M_0 = (47, 13, 7, 42)$ olan Petri ağında bunun uygulaması görülmektedir.



Şekil 3.8 Çok Sayıda Jeton Barındıran Bir Petri Ağı

3.4. Petri Ağlarının Çalışma Kuralları

Bir Petri ağının çalışması, o Petri ağında bulunan jetonların sayısı ve dağılımı ile kontrol edilir. Jetonlar yerlerde bulunur ve ağdaki geçişlerin çalışmasını kontrol eder. Bir Petri ağı geçişlerin ateşlenmesi ile çalışır. Bir ateşleme işlemi giriş yerlerinden jeton eksiltir, çıkış yerlerine de jeton ekler.

Bir geçişin ateşlenmesi ancak izinli olması ile mümkündür. Bir geçişin izinli olması ancak giriş yerlerinin her birinde en az o yer ile geçiş arasındaki dalların ağırlıkları kadar jeton bulunması ile mümkün olur. Giriş yerlerinde bulunan ve bir geçişi ateşlenebilir hale getiren jetonlara *izin jetonları* denir.

Tanım 3.8: Ateşlenebilme Koşulu

Bir $C = (P, T, F, W)$ Petri ağında ve M işaretlemesi altında $t_j \in T$ geçişi ancak ve ancak her $p_i \in P$ için,

$$M(p_i) \geq \#(p_i, I(t_j))$$

koşulu sağlanıyorsa ateşlenebilirdir.

Bir geçişin ateşlenmesi ile giriş yerlerinden dalın ağırlığı kadar jeton alınır ve yine çıkış yerlerine dalların ağırlığı kadar jeton eklenir. Bir t_3 geçişinin $I(t_3) = \{p_2\}$ ve $O(t_3) = \{p_7, p_{13}\}$ koşulları altında izinli olabilmesi için p_2 'de en az bir jeton bulunmalıdır. t_3 'ün ateşlenmesi ile p_2 'den bir jeton eksilir, p_7 ve p_{13} 'e de birer jeton eklenir. t_2 'de bulunan ekstra jetonlar t_3 'ün ateşlenmesinden etkilenmez.

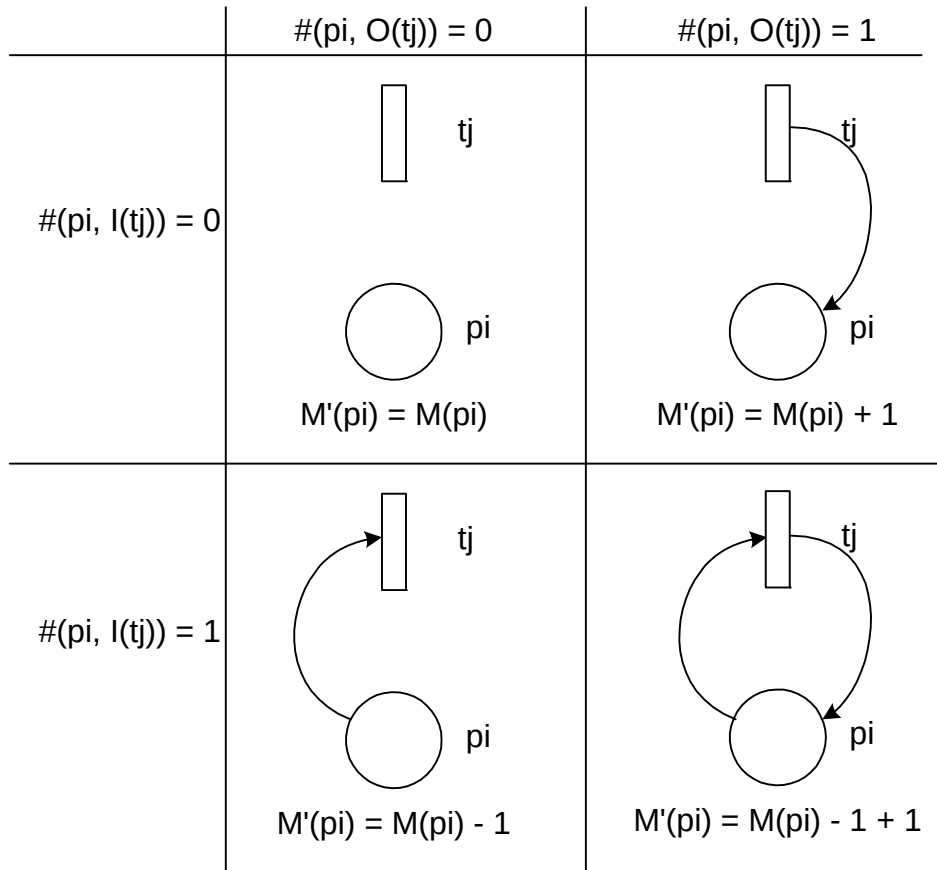
Bir geçişin ateşlenmesi ile Petri ağının M_0 ilk işaretlemesi değişip yeni bir M_1 işaretlemesine gider. Bir geçişin ateşlenmesi olmayan jetonları kaldırmaya çalışmaz. Giriş yerlerinden en az birinde yeteri sayıda jeton yoksa bu durumda o geçiş ateşlenemez.

Tanım 3.9: İşaretlemelerin Değişimi

M işaretlemesi ile işaretlenmiş bir Petri ağındaki t_j geçişi izinli olduğunda ateşlenebilir. t_j 'nin ateşlenmesi Petri ağının işaretleme durumunu M_1 'e getirir. Bu durum

$$M_1(p_i) = M(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j)) \quad (3.1)$$

bağıntısı ile ifade edilebilir. Şekil 3.9'da değişik koşullar altında bu bağıntının sonuçları görülmektedir.



Şekil 3.9 İşaretlemelerin Değişimi

İşaretleme akışı Örnek 3.10 üzerinde adım adım gösterilmektedir.

Örnek 3.10. Şekil 3.10'da Örnek 3.1'deki Petri ağı grafi ve ilk işaretlemesi görülmektedir. Bu grafi kullanarak üzerinde bir dizi ateşleme uygulanıp sonuçları incelensin,

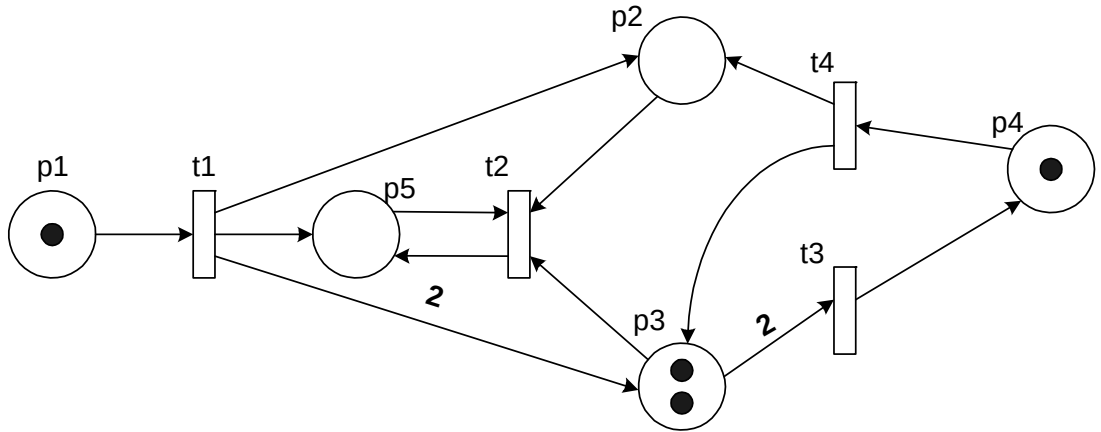
a) İlk durumda t_1 , t_3 ve t_4 geçişleri ateşlenebilir durumdadır. İlk işaretleme $M_0 = [1 \ 0 \ 2 \ 1 \ 0]^T$ 'dir.

b) t_4 'ün ateşlenmesi durumunda p_4 'teki jetonu eksiltilip, p_2 'e ve p_3 'e birer jeton ekler. Bu durumda Şekil 3.11'de görünen işaretleme elde edilir. Yeni işaretleme altında $M_1 = [1 \ 1 \ 3 \ 0 \ 0]^T$ 'dir.

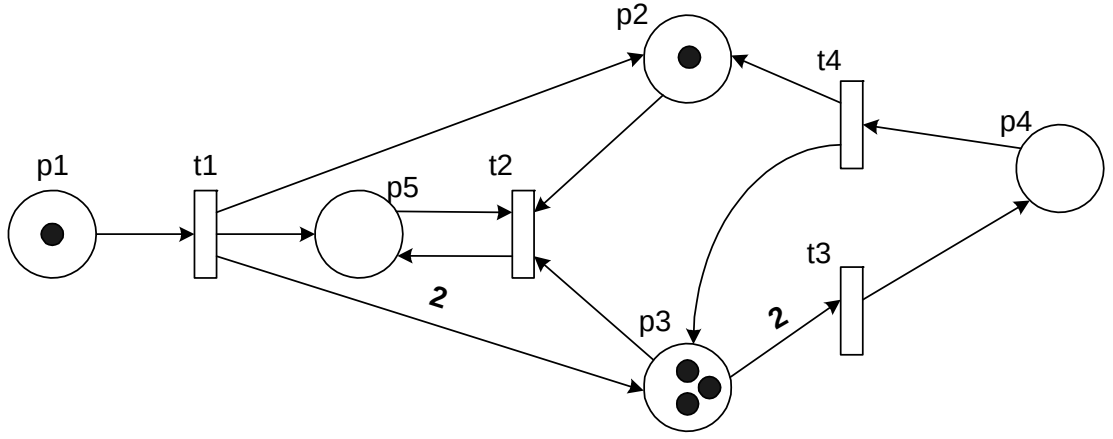
c) Oluşan bu yeni durumda t_1 ve t_3 geçişleri ateşlenebilir durumdadır. t_1 'in ateşlenmesi ile p_1 'den bir jeton eksilir, p_2 ve p_5 'e birer, p_3 'de ise iki jeton eklenir. Bu yeni durum Şekil 3.12'de görülmektedir. Yeni işaretleme altında $M_2 = [0 \ 2 \ 5 \ 0 \ 1]^T$ 'dir.

d) Oluşan yeni durumda t_2 ve t_3 ateşlenebilirdir. t_3 'ün ateşlenmesi ile p_3 'ten iki jeton eksilir, p_4 'e de bir jeton eklenir. Bu yeni durum Şekil 3.13'te görülmektedir. Yeni işaretleme altında $M_3 = [0 \ 2 \ 3 \ 1 \ 1]^T$ 'dir.

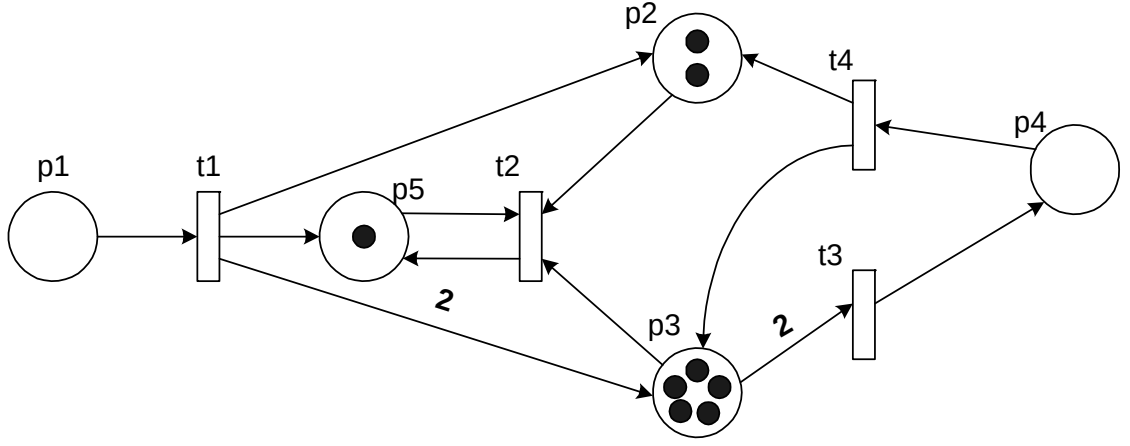
Bu şekilde ateşlemeler en azından bir tane ateşlenebilir geçiş olduğu sürece devam eder. Ateşlenebilir geçiş kalmadığı zaman çalışma durur.



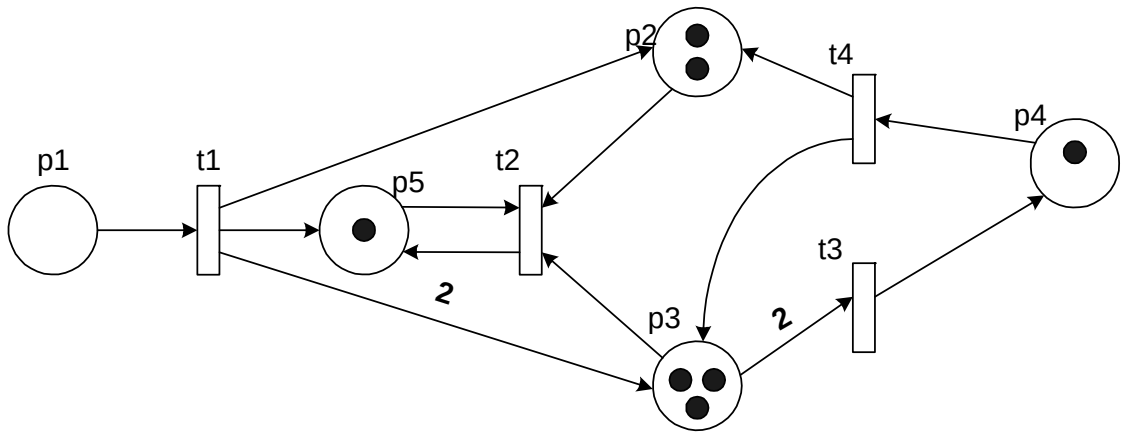
Şekil 3.10 İlk İşaretleme, t_1 , t_3 ve t_4 Geçişleri Ateşlenebilir Durumda, M_0



Şekil 3.11 t_4 'ün Ateşlenmesi ile Oluşan İşaretleme M_1



Şekil 3.12 t_1 'in Ateşlenmesi ile Oluşan İşaretleme M_2



Şekil 3.13 t_3 'ün Ateşlenmesi ile Oluşan İşaretleme M_3

3.5. Petri Ağı Durum Uzayları

Bir Petri ağının durumu işaretlemesi tarafından tanımlanır. Bir geçişin ateşlenmesi ile Petri ağının işaretlemesi değiştiğinden durumu da değişir. n adet yer olan bir Petri ağının durum uzayı mümkün olan tüm işaretlemelerin (N^n) kümesidir. Bir geçişin ateşlenmesi ile durumda oluşan değişiklikler *sonraki-durum* fonksiyonu ile tanımlanır ve δ ile gösterilir. Bu fonksiyon, M işaretlemesi altında t_j ateşlemesi ile Petri ağını yeni bir M_2 durumuna götürür. t_j 'nin M işaretlemesi altında ateşlenebilir olması $\delta(M, t_j)$ 'nin tanımlanabilmesi için yeter koşuldur. t_j geçişi izinli ise bu durumda ateşlenmesi ile $\delta(M, t_j) = M_2$ *sonraki-durum* fonksiyonu ile elde edilen M_2 , t_j geçişinin giriş yerlerinden jeton eksiltilmesi ve çıkış yerlerine jeton eklenmesi ile oluşan yeni işaretlemedir.

Tanım 3.10: Sonraki-Durum Fonksiyonu (δ)

Bir $C = (P, T, F, W)$ Petri ağında M işaretlemesi altında t_j işaretlemesi ile sonraki-durum fonksiyonu her $p_i \in P$ için

$$M(p_i) \geq \#(p_i, I(t_j)),$$

koşulu sağlanıyorsa tanımlanabilir. $\delta(M, t_j)$ tanımlanabilir ise bu durumda $\delta(M, t_j) = M_2$ ile ilgili,

$$M_2(p_i) = M(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))$$

koşulu her $p_i \in P$ için sağlanacaktır.

Bir $C = (P, T, F, W)$ Petri ağı ve M_0 ilk işaretlemesi verilmiş olsun. Bu durumda Petri ağı üzerinde ardışık ateşlemeler gerçekleştirilebilir. İlk işaretleme altında izinli bir t_j geçişi ateşlendiğinde $M_1 = \delta(M_0, t_j)$ işaretlemesi elde edilir. Bu yeni işaretleme altında herhangi bir t_k izinli geçişinin ateşlenmesi Petri ağını $M_2 = \delta(M_1, t_k)$ durumuna götürmektedir. Bu ateşleme dizisi en azından bir tane ateşlenebilir geçiş olduğu sürece devam eder. Hiçbir geçişin ateşlenemediği bir işaretleme elde edilirse bu durumda çalışma duracaktır.

Bir Petri ağının çalışmasından iki dizi elde edilir; *işaretlemeler dizisi* ($M_0, M_1, M_2, \dots$) ve ateşlenen *geçişlerin dizisi* ($t_1, t_2, t_3, \dots$). Bu iki dizi arasında $\delta(M_k, t_j) = M(k+1)$ bağıntısı vardır, ($k = 0, 1, 2, \dots$). Belli bir geçiş dizisi ve M_0 kullanılarak kolayca işaretleme dizisi elde edilebilir. Benzer şekilde bir işaretleme dizisinden de

ateşlenen geçişlerin dizisi elde edilebilir. Her iki dizi de Petri ağının çalışması konusunda ipuçları verir.

Bir M işaretlemesi altında geçişlerden bazıları izinli olup ateşlenebilir olacaktır. Geçişlerden birinin ateşlenmesi sonucunda M_2 durumuna ulaşılabilecektir. Bu durumda M_2 'ye M 'den *hemen erişilebilir* denir.

Tanım 3.11: Hemen Erişilebilirlik

$C = (P, T, F, W)$ Petri ağı yapısı ve M işaretlemesi verilmiş olsun. M_2 işaretlemesine M 'den *hemen-erişilebilir* denmesi için öyle bir $t_j \in T$ geçişi izinli olmalıdır ki $\delta(M, t_j) = M_2$ olsun.

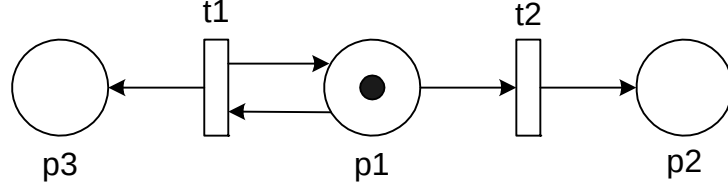
Hemen-erişilebilir işaretleme kavramı genişletilip belirli bir Petri ağında erişilebilen tüm işaretlemeler kümesi incelenebilir. M_1 'e M_0 'dan hemen erişiliyorsa, M_2 'ye de M_1 'den hemen erişiliyorsa bu durumda M_0 'dan M_2 'ye erişilebilir denir. Belirli bir C Petri ağını ve M_0 ilk işaretlemesini göz önüne alarak M_0 'dan ulaşılacak tüm işaretlemeler kümesini $R(C, M_0)$ veya $R(M_0)$ olarak tanımlanır. Ağın durumunu M_0 'dan M_k 'ya ($k = 0, 1, 2, \dots$) götüren bir ateşleme dizisi varsa $M_k \in R(C, M_0)$ 'dır. Erişilebilirlik bağıntısı, hemen-erişilebilirlik bağıntısının yansımali geçişken örtendir.

Tanım 3.12: Erişilebilirlik Kümesi $R(C, M_0)$

Bir $C = (P, T, F, W)$ Petri ağı ve M_0 ilk işaretlemesi verilmiş olsun. Bu durumda bir Petri ağının $R(C, M_0)$ ulaşılabilirlik kümesi aşağıdaki koşulları sağlayan işaretlemelerin kümesidir;

1. $M \in R(C, M_0)$
2. Eğer $M_k \in R(C, M_0)$ ve belirli bir $t_j \in T$ için $M_l = \delta(M_k, t_j)$ ise bu durumda $M_l \in R(C, M_0)$ 'dır.

Örnek 3.11. Şekil 3.14'deki Petri ağı ve $M_0 = (1, 0, 0)$ ilk işaretlemesinden sonra $(0, 1, 0)$ ve $(1, 0, 1)$ hemen ulaşılabilir işaretlemelerdir. $(0, 1, 0)$ 'da hiçbir geçiş izinli olmadığından yeni işaretlemelere erişilemez. $(1, 0, 1)$ 'den ise $(0, 1, 1)$ ve $(1, 0, 2)$ işaretlemelerine erişilebilir. Bu Petri ağının erişilebilirlik kümesi $R(C, M_0)$, $\{(1, 0, n), (0, 1, n) | n \geq 0\}$ olarak tanımlanabilir.



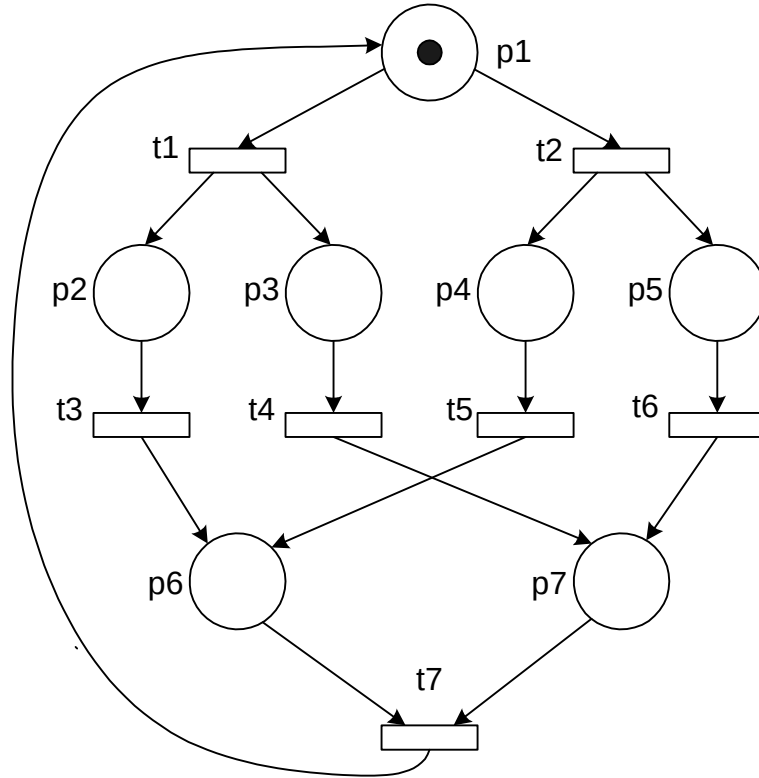
Şekil 3.14 İşaretli Petri Ağı

Bir sonraki-durum fonksiyonunu genişleterek belirli bir durumdan başka bir duruma ulaşmak amacıyla ateşlemeler dizisi gerçekleştirilebilir. Bu durumda M_0 ilk işaretlemesi ve $t_{j1}, t_{j2}, \dots, t_{jk}$ ateşleme dizisini kullanarak $M_k = \delta(M_0, t_{j1}, t_{j2}, \dots, t_{jk})$ olarak tanımlanabilir.

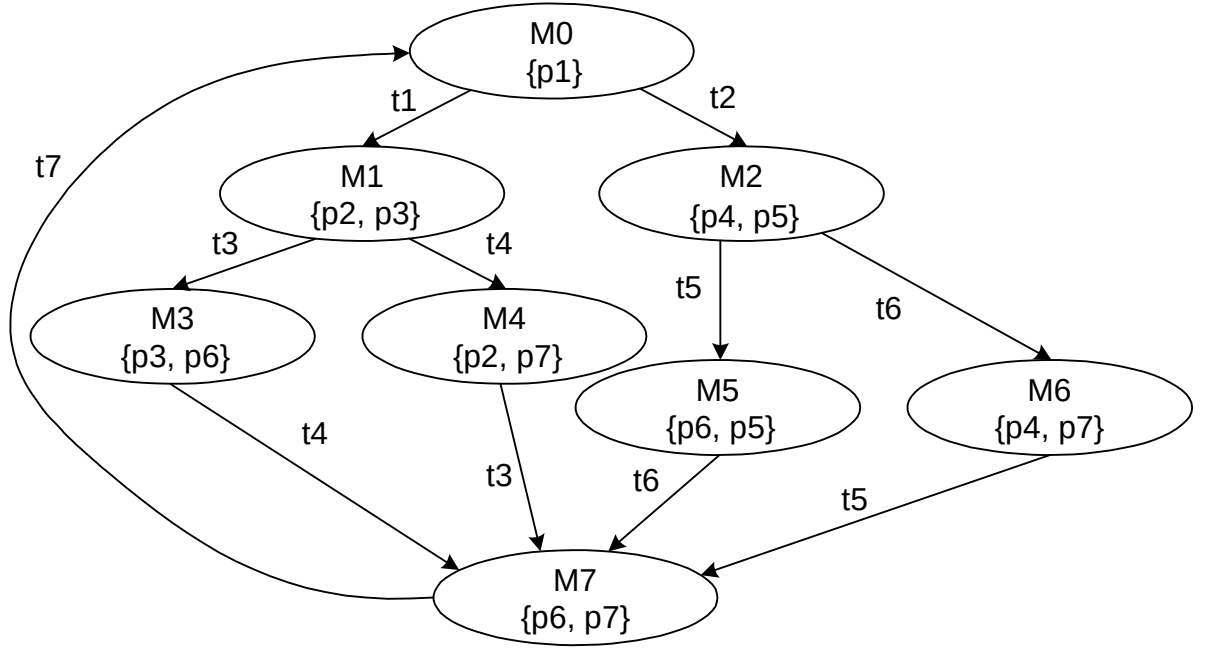
Tanım 2.13: Erişilebilirlik Ağacı

Bir $C = (P, T, F, W)$ Petri ağı ve M_0 ilk işaretlemesi verilmiş olsun. Bu ağdan $R(C, M_0)$ erişilebilirlik kümesi elde edilsin. Bu kümenin tüm elemanlarını grafiksel olarak ifade etmek amacıyla kullanılan grafa *erişilebilirlik ağacı* denir. [3]

Örnek 3.12. Aşağıda Şekil 3.15’de Petri ağı grafı görülmektedir. Bu Petri ağına ilişkin erişilebilirlik ağacı ise Şekil 3.16’da gösterilmiştir.



Şekil 3.15 Örnek Petri Ağı Grafı



Şekil 3.16 Şekil 3.15'teki Petri Ağının Erişilebilirlik Ağacı

Tanım 3.14: Genişletilmiş Sonraki Durum Fonksiyonu

Genişletilmiş sonraki durum fonksiyonu $\delta: N^n \times T^* \rightarrow N^n$, t_j geçişi ve $\sigma \in T^*$ geçiş dizisi olarak şu şekilde tanımlanabilir, [1]

$$\delta(M_0, t_j \sigma) = \delta(\delta(M_0, t_j) \sigma)$$

$$\delta(M_0, \lambda) = M$$

3.6. Petri Ağları Alt Sınıfları

Bu bölümde ilk olarak Petri ağlarının alt sınıfları tanımlanacaktır. İncelenen Petri ağlarının sıradan oldukları varsayılmaktadır. [2]

Tanım 3.2'de giriş ve çıkış fonksiyonları tanımlanmıştı. Bu bölümde bu tanıma benzer olmakla birlikte bazı noktalarda ayrılan bir tanım yapılacaktır. O tanımda geçişlere ait ağırlıklı giriş ve çıkış yerleri tanımlanmıştı, yeni tanımda ise bir geçiş veya yerin öncü veya izleyen elemanlarından oluşan kümeler tanımlanacaktır;

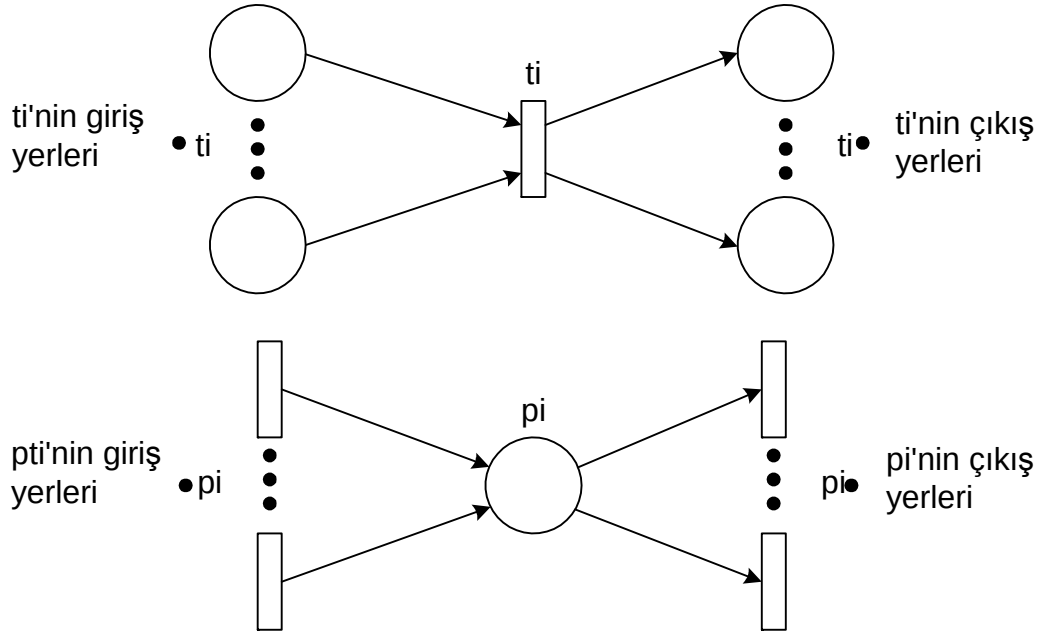
• $t = \{p \mid (p, t) \in F\}$, t geçişinin öncü yerlerinin oluşturduğu küme

$t^\bullet = \{p \mid (t, p) \in F\}$, t geçişinin izleyen yerlerinin oluşturduğu küme

• $p = \{t \mid (t, p) \in F\}$, p yerinin öncü geçişlerinin oluşturduğu küme

$p^\bullet = \{t \mid (p, t) \in F\}$, p yerini izleyen geçişlerin oluşturduğu küme

Yukarıdaki kümelerin grafiksel ifadesi Şekil 3.17’de gösterilmiştir. Bu notasyonlar kolaylıkla bir alt küme kavramına genişletilebilir. $S_1 \subseteq P$ olsun, bu durumda $\bullet S_1$ kümesi $p \in S_1$ olan tüm $\bullet p$ ’lerin birleşiminden oluşur. Yukarıda tanımlanan notasyonlar kullanılarak Petri ağlarının üzerinde bazı kısıtlamalar tanımlanabilir. Bu bölümde yapılacak analizlerde incelenen ağlarda tek başına yer veya geçişin olmadığı varsayılacaktır, $\bullet p_i = p_i \bullet \neq \emptyset$ veya $t_i \bullet = t_i \bullet \neq \emptyset$.



Şekil 3.17 Öncü ve İzleyen Kümeler

Tanım 3.15: Durum Makinesi (State Machine), DM

Sıradan (ordinary) bir Petri ağında her t_j geçişinin tam bir giriş yeri ve tam bir çıkış yeri varsa, bu ağ *durum makinesi* tanımlar,

$$|\bullet t_i| = |t_i \bullet| = 1, \text{ her } t_i \in T$$

Durum makineleri çakışmalara izin verirler, öyle ki farklı çıkış geçişlerine sahip bir yerde yeterli sayıda jeton bulunduğu zaman onların tamamını izinli hale getirebilir.

Tanım 3.16: İşaretli Graflar (Marked Graph), İG

Sıradan bir Petri ağında her p_i yerinin tam bir giriş geçişi ve tam bir çıkış geçişi varsa, bu ağ *işaretli graf* tanımlar,

$$|\bullet p_i| = |p_i \bullet| = 1, \text{ her } p_i \in P$$

İşaretli graflar eş zamanlı çalışmaya izin verirler. İki yer aynı çıkış geçişini paylaşabilir bu durumda geçiş ateşlendiğinde her iki yer de etkilenir. İşaretli graflarda asla çakışma olmaz, bunun sebebi her yeri izleyen tek bir geçişin olmasıdır. Bir yerdeki jeton çıkışındaki tek bir yer tarafından tüketilebilir.

Tanım 3.17: Serbest Seçim Ağı (Free Choice Nets), SSA

Sıradan bir Petri ağında, bir yerden çıkan veya o yere gelen her dal tek ise bu ağ *serbest seçim ağı* denir.

$$\forall p_i \in P, |p_i \bullet| \leq 1 \text{ veya } \bullet(p_i \bullet) = p_i, \text{ veya}$$

$$\forall p_i, p_j \in P, p_i \bullet \cap p_j \bullet \neq \emptyset \Rightarrow |p_i \bullet| = |p_j \bullet| = 1$$

Serbest seçim ağları durum makinelerini ve işaret graflarını kapsayan genelleştirilmiş bir sınıftır. Bu yüzden hem durum makinelerinin hem de işaret graflarının bazı özelliklerini barındırırlar. Serbest seçim ağları işaretli graflar tipinde eş zamanlı çalışmaya ve durum makineleri tipinde çakışmalara izin verirler. Her durum makinesi ve her işaretli grafi bir serbest seçim ağıdır. Serbest seçim ağında ortak bir giriş yerini paylaşan t_1 ve t_2 geçişleri her zaman aynı anda ateşlenebilir hale gelebilirler.

Tanım 3.18: Genişletilmiş Serbest Seçim Ağı (Extended Free Choice Nets), GSSA

Sıradan bir Petri ağında,

$$\forall p_i, p_j \in P, p_i \bullet \cap p_j \bullet \neq \emptyset \Rightarrow p_i \bullet = p_j \bullet$$

koşulu sağlanıyorsa buna *genişletilmiş serbest seçim ağı* denir.

Genişletilmiş serbest seçim ağları, serbest seçim ağlarından farklı olarak iki yerin aynı çıkış geçişlerini paylaşmasına da olanak sağlarlar. Genişletilmiş serbest seçim ağları, serbest seçim ağlarını kapsar.

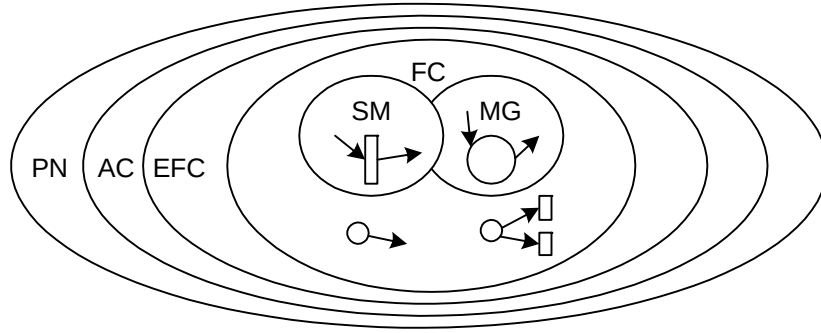
Tanım 3.19: Asimetrik Seçim Ağları (Asymmetric Choice Nets), ASA

Sıradan bir Petri ağında,

$$\forall p_i, p_j \in P, p_i \bullet \cap p_j \bullet \neq \emptyset \Rightarrow p_i \bullet \subseteq p_j \bullet \text{ veya } p_j \bullet \subseteq p_i \bullet$$

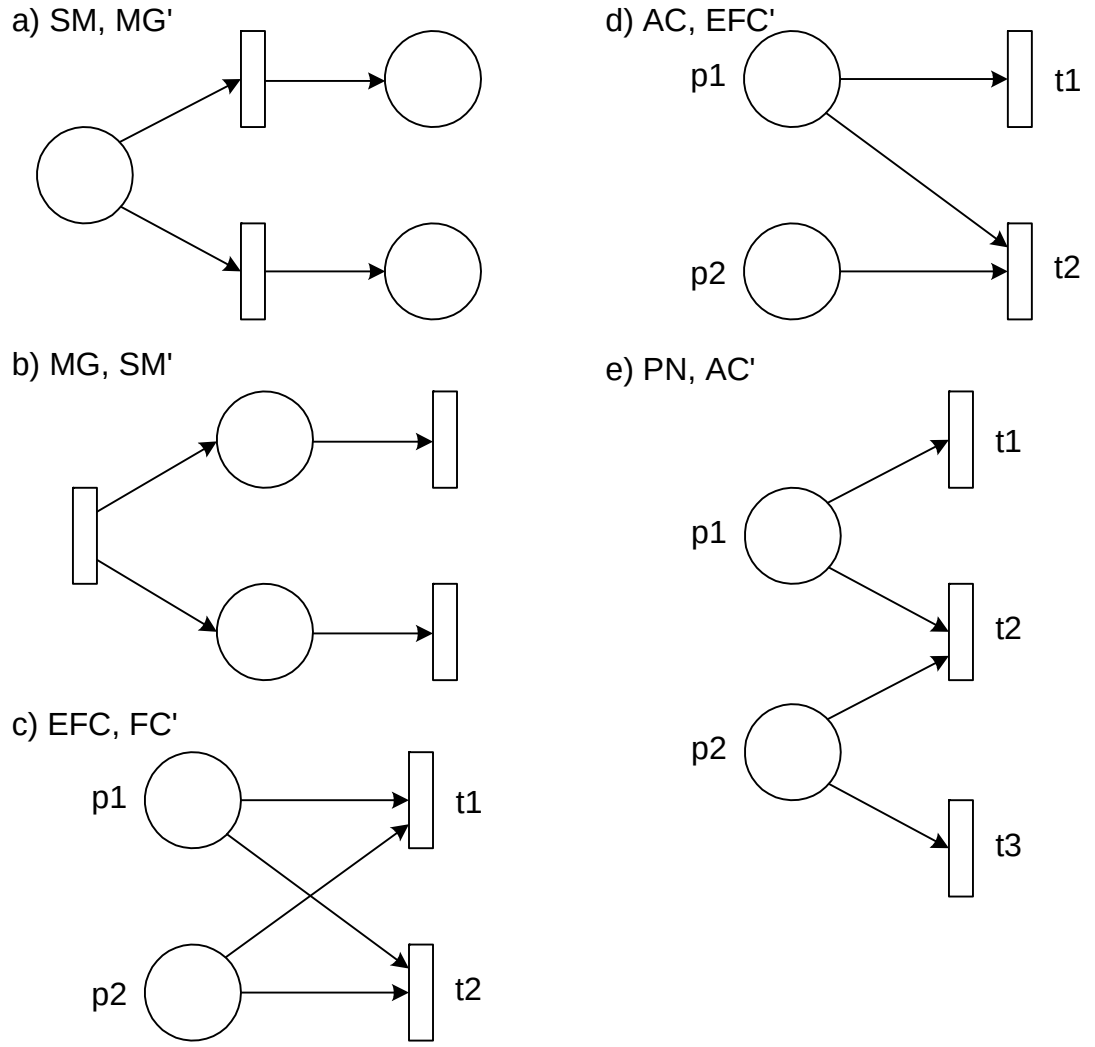
koşulu sağlanıyorsa buna *asimetrik seçim ağı* denir.

Asimetrik seçim ağları, genişletilmiş serbest seçim ağlarını kapsar. Dolayısıyla daha önce tanımlanan durum makinelerini, işaret graflarını ve serbest seçim ağlarını da kapsar. Serbest seçim ağlarından farklı olarak iki yerin ortak çıkış yerlerinin boş küme olmaması durumunda çakışmasını zorunlu kılmaz. İki p_1 ve p_2 yerlerinin ortak bir geçişi t_1 olabilirken, p_2 'nin ayrıca t_2 'ye giden bir çıkışı bulunabilir. Oysa genişletilmiş serbest seçim ağlarında p_1 'in de t_2 'ye çıkışı olmalıdır, yani çıkışları ortak olmalıdır. Serbest seçim ağlarında bu çıkış tek bir adet ile sınırlandırılmıştır. Şekil 3.18'de çeşitli Petri ağı sınıfları arasındaki ilişki görülmektedir.

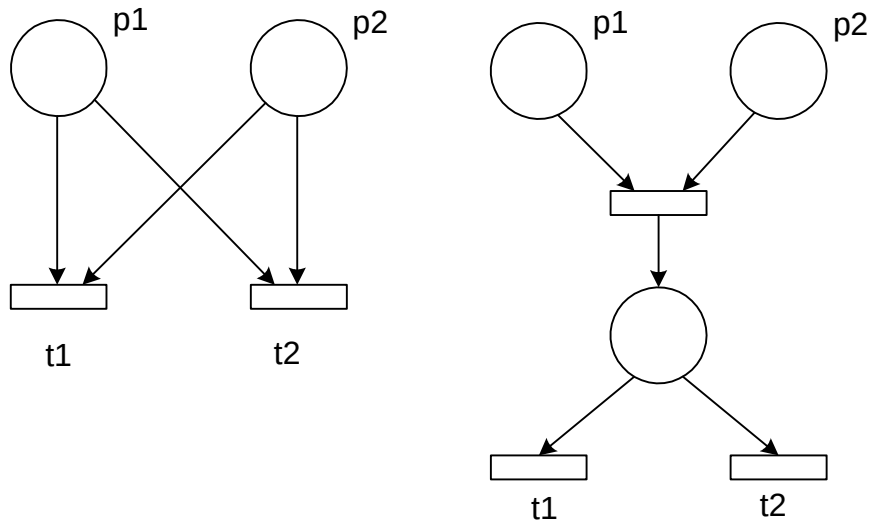


Şekil 3.18 Petri Ağları ve Alt Sınıfları Arasındaki İlişki

Şekil 3.19'de çeşitli Petri ağı sınıfları ile ilgili örnekler yer almaktadır. Şekil 3.19.a'da durum makineleri, Şekil 3.19.b'de ise işaretli graflar görülmektedir. Serbest seçim ağları Şekil 3.19.c'deki $p_1 \bullet = p_2 \bullet = \{t_1, t_2\}$ gibi yapılara izin vermiyor. Genişletilmiş serbest seçim ağları Şekil 3.19.c'yi kapsarken, Şekil 3.19.d'de görülen $p_1 \bullet = \{t_1\} \subseteq p_2 \bullet = \{t_1, t_2\}$ tipinde yapılara izin vermiyorlar. Hem serbest seçim ağlarında, hem de genişletilmiş serbest seçim ağlarında, aynı yeri paylaşan iki geçiştten biri ateşlenebilir olduğunda diğerinin de ateşlenebilir olması şartı vardır. Şekil 3.19.c'deki yapı eşdeğer serbest seçim ağına dönüştürülebilir. Şekil 3.20'de bu dönüşüm işlemi gösterilmiştir. Asimetrik seçim ağları Şekil 3.19.d'de görülen yapılara izin verirken, Şekil 3.19.e'de görülen ve karışıklık içeren ağlara izin vermezler, $p_1 \bullet = \{t_1, t_2\}$ ve $p_2 \bullet = \{t_2, t_3\}$. Asimetrik ağlarda daha öncekilerden farklı olarak aynı yeri paylaşan t_1 ve t_2 yerlerinden birinin çalıştırılmaz olması kabul edilebilir bir durumdur.



Şekil 3.19 Çeşitli Petri Ağı Alt Kümeleri



Şekil 3.20 GSSA Ağından SSA Ağına Dönüşüm

Özetlenirse; durum makineleri senkronizasyona izin vermezler, işaretli graflar çatışmalara izin vermezler, serbest seçim ağları karmaşaya izin vermezler, asimetrik serbest seçim ağları simetrik karmaşaya izin vermezler. Petri ağları ise bunların tamamını kapsamaktadır.

Petri ağları ile ilgili yapılabilecek bir diğer sınıflandırma çalışması ağların bağıllık durumları ile ilgilidir. Genel olarak yapılan incelemelerde ağların bağlı olması istenir. Bağlı olmayan ağ yapılarında ise en az bir yer veya geçiş düğümü ağdaki başka bir düğüme bir dal ile bağlı değildir. Bağlı olan ağlar da kendi içinde ikiye ayrılır; sıkı bağlı ağlar, zayıf bağlı ağlar.

Tanım 3.20: Sıkı Bağlı Ağ

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Eğer aşağıdaki iki koşul sağlanıyorsa o ağ *sıkı bağlı ağ* denir.

Ağdaki $\forall p_i \in P$ yeri için ağda yer alan $\forall t_j \in T$ geçişine yönlü bir yol varsa,

Ağdaki $\forall t_j \in T$ geçişi için ağda yer alan $\forall p_i \in P$ yerine yönlü bir yol varsa.

Tanımda yer alan $i = 1, 2, 3, \dots, n$, $j = 1, 2, \dots, k$, $n = |P|$, $k = |T|$ 'dir [3].

Tanım 3.21: Zayıf Bağlı Ağ

Eğer bağlı bir ağ sıkı bağlı değil ise ona zayıf bağlı ağ denir. [3]

3.7. Petri Ağlarında Kapan (Trap) ve Savak (Siphon) Yapıları

Hem kapanlar hem de savaklar bir Petri ağındaki belli kuralları sağlayan yerlerin bir araya gelmesiyle oluşan yer kümeleridir.

Kapanlar ve savaklar Petri ağlarının işaretlemesi ile ilgili yapısal özellikleri tanımlamada kullanılırlar. Bu yönleriyle Petri ağlarının dinamik davranışını belirleyen yapılar oldukları söylenebilir. Kapanlar bir M_k işaretlemesi altında jeton kazanmaları durumunda M_k işaretlemesini takip eden işaretlemelerin her birinde en az bir jeton barındırır ve canlılıklarını korurlar. Buna karşın, M_k işaretlemesi altında bir savakta yer alan tüm yerler jetonsuz kalırsa takip eden işaretlemelerin hiçbirinde işaretli hale getirilemezler. Buradan hareketle kapanların ve savakların bir ağın canlılığı ile doğrudan ilgili olduğu söylenebilir. [4]

Kapanlar ve savaklar genel olarak sıradan Petri ağlarının yapısal analizlerinde kullanılırlar. Burada kapanlar ve savaklar kullanılarak incelenebilecek birkaç örnek verilmektedir,

- Sıradan bir Petri ağında kilitli (deadlock) bir sistem varsa bu durumda o ağdaki jetonsuz yerlerin oluşturduğu yerler kümesi bir savak tanımlar.
- Kapanların bir defa jeton kazanması durumunda her zaman işaretli kaldıkları göz önüne alınırsa, ve sıradan bir Petri ağında ilk M_0 işaretlemesi altında her kapan işaretli ise bu durumda bu sistem kilitlenmesizdir. Asimetrik seçim ağlarında her kapanın işaretli olması durumu ağın canlılığı için yeter koşul oluşturur. Serbest seçim ağlarında da ise yeter ve gerek koşuldur.
- M_0 işaretlemesi canlı bir sistemin ev işaretlemesi ise bu durumda her kapan işaretli olmalıdır, aksi durumda bir defa işaretlenen bir kapan her zaman canlı kalacağından asla ilk duruma ulaşamaz.

Kapanlar ve savaklarla ilgili temel tanım ve teoremler aşağıda ele alınmıştır.

Tanım 3.22: Savak (Siphon) Tanımı

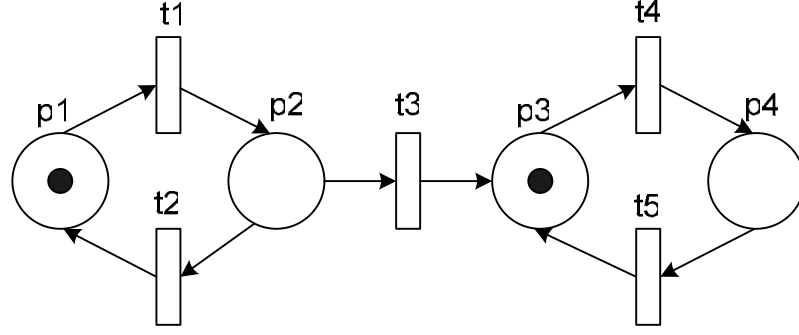
Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu Petri ağında S yerlerden oluşan bir küme olsun, öyleki $S \subseteq P$. S yerler kümesi $\bullet S \subseteq S \bullet$ koşulunu sağlıyorsa bu bir savak belirtir.

Teorem 3.1:

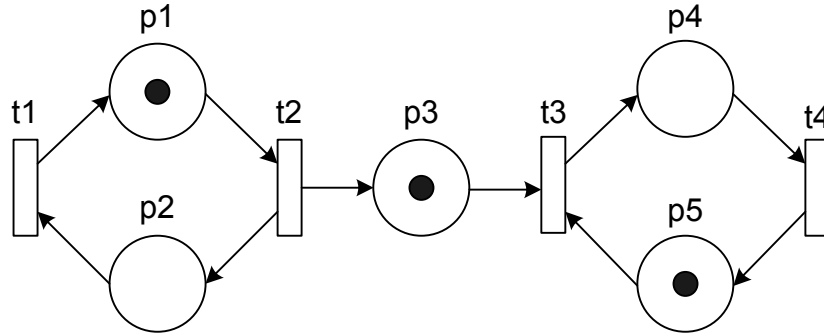
İşaretli bir Petri ağında S savağı tanımlı olsun. Eğer S savağı M_0 ilk işaretlemesinde işaretli değil ise bu durumda erişilebilir hiçbir işaretleme altında işaretli duruma getirilemez. [5]

İspat:

$R(M_0)$, S 'de hiçbir yeri izinli hale getiremeyen işaretlemelerin kümesi olsun. $R(M_0)$ 'de bir M_k işaretlemesini ve buna bağlı bir t geçişi tanımlansın, öyle ki t geçişi $M_k \rightarrow M_{k+1}$ durum değişimini sağlasın. Bu durumda $t \notin S \bullet$ 'tir, çünkü M_k t geçişini izinli hale getirir ve M_k işaretlemesi S 'teki hiçbir yeri işaretleyemez. S bir savak olduğundan $t \notin \bullet S$ 'i gerektirir. Bu yüzden hiçbir $p \in S$ yeri t geçişinin ateşlenmesiyle jeton kazanamaz.



Şekil 3.21 Örnek Petri Ağı A1



Şekil 3.22 Örnek Petri Ağı A2

Örnek 3.13: Şekil 3.21’deki A1 Petri ağında aşağıdaki savaklar bulunmaktadır, [5]

$$\emptyset, \{p1, p2\}, \{p1, p2, p3, p4\}$$

t1 ve t3 geçişlerinin sırasıyla ateşlenmesi sonucunda {p1, p2} işaretli hale gelir ve bir daha hiçbir işaretleme altında jeton kazanamaz.

Şekil 3.22’de görülen A2 Petri ağında aşağıdaki savaklar bulunmaktadır,

$$\emptyset, \{p1, p2\}, \{p1, p2, p3\}, \{p1, p2, p3, p4\}, \{p1, p2, p3, p4, p5\}, \{p4, p5\}$$

Savaklar bir Petri ağındaki geçişlerin ölü olduğunu göstermek amacıyla kullanılabilirler.

Teorem 3.2:

En az bir yeri içinde barındıran bir S savağı $C = (P, T, F, W)$ Petri ağında tanımlı olsun. İlk işaretlemede eğer hiçbir geçiş ölü değil ise bu durumda S’de yer alan ve ayrık olmayan bir yer işaretlidir. [5]

İspat:

S'de ilk işaretleme altında ayırık olmayan hiçbir yerin işaretli olmadığını varsayalım. p , S'de yer alan bir yeri tanımlasın ($p \in S$). Teorem 3.1 gereği p yeri hiçbir erişilebilir işaretleme altında işaretlenemez. Bu yüzden $(\bullet p \cup p \bullet)$ 'de yer alan tüm geçişler ölüdür. p ayırık bir yer olmadığından ilk işaretleme altında ölü bir geçiş vardır ki bu teoremin koşullarıyla çelişir.

Tanım 3.23: Kapan (Trap) Tanımı

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu Petri ağında S yerlerden oluşan bir küme olsun, öyle ki $S \subseteq P$. S yerler kümesi $S \bullet \subseteq \bullet S$ koşulunu sağlıyorsa bu bir kapan belirtir.

Kapanlar bir defa jeton kazandıklarında bir daha asla jetonları kaybetmezler. Bu yönleri ile kapanlar Petri ağlarının canlılık analizlerinde kullanılabilirler.

Teorem 3.3:

İşaretli bir Petri ağında S kapanı tanımlı olsun. Eğer S kapanı ilk işaretlemeye işaretli ise bu durumda her erişilebilir işaretlemeye de işaretli olacaktır. [6]

İspat:

$R(M_0)$, S'de en az bir yeri izinli hale getiren işaretlemelerin kümesi olsun. Varsayım gereği ilk M_0 işaretlemesi de $R(M_0)$ 'da yer almalıdır. M_k 'da $R(M_0)$ 'da yer alan bir işaretleme olsun ve buna bağlı bir t geçişi tanımlansın, öyle ki t geçişi $M_k \rightarrow M_{k+1}$ durum değişimini sağlasın. Eğer $t \notin S \bullet$ ise bu durumda M_k 'da işaretli olan ve S'te yer alan p yerleri işaretli kalır. Eğer $t \in S \bullet$ ise aynı zamanda $t \in \bullet S$ olmalıdır. Bu koşul kapanın tanımından ileri gelir ($S \bullet \subseteq \bullet S$). Bu yüzden M_{k+1} işaretlemesi altında $(t \bullet \cap S)$ 'te yer alan en az bir yer işaretli olmalıdır. Sonuç itibarıyla $R(M_0)$ erişilebilir tüm işaretlemeleri barındırır.

Örnek 3.14: Şekil 3.21'deki A1 Petri ağında aşağıdaki kapanlar bulunur,

$$\emptyset, \{p_3, p_4\}, \{p_1, p_2, p_3, p_4\}$$

ve Şekil 3.22'deki A2 Petri ağında aşağıdaki kapanlar bulunur,

$$\emptyset, \{p_1, p_2\}, \{p_1, p_2, p_3, p_4, p_5\}, \{p_2, p_3, p_4, p_5\}, \{p_3, p_4, p_5\}$$

Teorem 3.4:

Tersinir (reversible) ve ayrık yerler barındırmayan bir S kapanının bulunduğu $C = (P, T, F, W)$ Petri ağı göz önüne alınsın. İlk işaretlemede hiçbir geçiş ölü değil ise bu durumda S kapanı işaretlidir. [5]

İspat:

$p \in S$ olan bir yeri, t 'de $(\bullet p \cup p \bullet)$ koşuluna uyan bir geçiş tanımlasın. t geçişi ilk işaretlemede ölü olmadığından, p yerini işaretleyen bir M_k işaretlemesi bulunmalıdır. Bu durumda S kapanı erişilebilir bir işaretleme altında işaretli hale gelecektir. M_0 ilk işaretlemesi M_k işaretlemesinden tersinirlik gereği erişilebilir olduğundan S kapanı M_0 işaretlemesi altında işaretli olmalıdır.

Şekil 3.22'deki A2 Petri ağı tersinirdir ve tüm kapanlar ilk işaretlemede işaretli olduğundan hiçbir ölü geçiş yoktur. Bu teorem bir Petri ağının tersinir olup olmadığını göstermek amacıyla kullanılabilir.

Teorem 3.5:

En az bir geçişi barındıran ve işaretli bir $C = (P, T, F, W)$ Petri ağı göz önüne alınsın. Ayrık yerler barındırmayan ve boş olmayan her savak ilk işaretlemede işaretlenmiş bir kapan barındırıyorsa bu durumda işaretlenmiş ağ kilitlemesizdir (deadlock free). [5]

İspat:

İşaretli bir Petri ağının kilitlenebilir oluşu varsayılınsın. M_k bu Petri ağında erişilebilen ölü işaretlemeyi tanımlasın. S ayrık olmayan ve M_k 'da işaretli olmayan yerlerin tanımladığı küme olsun. Bu durumda S'in boş olmayan ve ilk işaretlemede işaretli olmayan bir savak barındırdığı gösterilecektir.

M_k işaretlemesi altında her t geçişi ölüdür bu yüzden işaretli olmayan giriş yerleri bulundurulur. $S \bullet$ aynı durumda olan tüm t geçişlerinin kümesini tanımlar. Bu yüzden $(\bullet S \subseteq S \bullet)$ 'tir. S varsayım gereği ağda bir geçiş ve bu geçiş öncüsü bir yer bulundurduğundan boş küme değildir. Bu durumda savak tanımından S'in boş olmayan ve ayrık yerler barındırmayan bir savak olduğu sonucuna varılır. Yine varsayım gereği S savağı M_k işaretlemesi altında işaretli değildir. Bu yüzden S savağı M_k işaretlemesi altında bir kapan barındıramaz. İlk işaretlemede işaretli olan kapanlar her zaman canlı kaldığından, S ilk işaretleme altında bir kapan barındıramaz. Bu çelişki aranan sonucu verir.

Şekil 3.21'deki A1 Petri ağında $\{p_1, p_2\}$ sifonu boş olmayan bir kapan barındırmaz bu yüzden belli ateşlemelerden sonra ölü hale gelir. Şekil 3.22'deki A2 Petri ağı Teorem 5'in koşullarını sağlar bu yüzden kilitlenmesizdir.

Tanım 3.24: Minimal Savak

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu Petri ağında bir S savağı başka hiçbir savak tarafından içerilmiyor ise buna minimal savak denir. [7]

Tanım 3.25: Minimal Kapan

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu Petri ağında bir S kapanı başka hiçbir kapan tarafından içerilmiyorsa buna minimal kapan denir. [7]

İki savağın birleşimi yine bir savaktır. Benzer şekilde iki kapanın birleşimi yine bir kapan tanımlar. Bu bilgiler ışığında aşağıdaki tanımlar yapılabilir.

Tanım 3.26: Baz Kapan ve Savaklar

Bir savak (kapan) başka savakların (kapanların) birleşimi olarak ifade edilemiyorsa o bir baz savaktır (baz kapandır). [7]

Her minimal savak (kapan) bir baz savaktır (baz kapandır) ama tersi her zaman doğru değildir.

3.8. Petri Ağlarını Modelleme Biçimleri

Bölüm 2'de Petri ağları bir modelleme ve analiz aracı olarak tanımlanmıştı. Bu bölümde de Petri ağlarının temel bileşenleri ve bunların özellikleri ele alındı. Çeşitli bilim dallarında farklı bileşenleri modellerken Petri ağı bileşenlerinden neye karşı geldiklerini bilmek gerekir. [2]

Koşullar ve olayları göz önüne alıp bunları Petri ağlarını kullanarak modellemek gerektiğinde yerler koşulları, geçişler de olayları temsil edeceklerdir. Bir geçiş (olay) belli sayıda giriş ve çıkış yerine sahiptir, bunlar sırasıyla ön koşulları ve sonraki koşulları temsil ederler. Bir işaret işleme devresinde giriş yerleri ham veri girişini, çıkış yerleri işlenmiş veriyi temsil ederken, aradaki geçiş ise filtreleme yapan bir işaret işlemciye karşı gelebilir. Bir yerdeki bir jetonun varlığı o yerle ilgili koşulların oluşmuş olduğunu gösterir. Başka bir yorumda, bir yerdeki k adet jeton, k adet veri

veya kaynağın kullanılabilir olduğunu gösterir. Bazı tipik Petri ağı modelleme biçimleri Tablo 3.1’de gösterilmiştir.

Tablo 3.1 Geçiş ve Yerlerin Tipik Yorumları

Giriş Yerleri	Geçişler	Çıkış Yerleri
Ön koşullar	Olay	Sonraki Koşullar
Ham veri girişi	Hesaplama adımı	Çıkan veri
Giriş işareti	İşaret işlemci	Çıkış işareti
Gerekli kaynaklar	Görev veya iş	Kaynakların bırakılması
Koşullar	Lojik koşul	Sonuçlar
Ara bellek	İşlemci	Ara bellek

Burada verilen tabloda yer alan modelleme biçimleri dışında kalan daha pek çok alandan incelenecek sistemler bir Petri ağı olarak modellenebilir. Bu konuda daha ayrıntılı bilgilere Agerwala [8] makalesinden ulaşılabilir.

4. PETRİ AĞLARININ ANALİZ EDİLMESİ

Petri ağları birçok sistemin çalışma yapısını doğru bir şekilde yansıtabilen bir modelleme aracıdır. Özellikle eş zamanlı çalışma gerektiren sistemlerin modellenmesi doğal ve uygun yollarla gerçekleştirilebilmektedir. Bir Petri ağı eş zamanlı sistemlerin çalışma yapısını incelemek amacıyla kullanılabilir. [1]

Gerçekte modelleme tek başına bir çözüm değil, çözüme giden yolda bir ilk adım oluşturur. Önemli olan modellenen sistemin analiz edilmesidir. Dolayısıyla bu bölümde ağırlıklı olarak Petri ağlarını özelliklerini irdeleyen yöntemler tanımlanacaktır. Zaman içinde Petri ağlarının özelliklerini inceleyen birçok yöntem geliştirilmiştir, buna karşın cevaplanması gereken sorular da vardır. Geliştirilen analiz yöntemlerine geçmeden önce Petri ağları ile ilgili cevaplanması istenen sorular tespit edilmelidir. Analiz yöntemleri bu sorulara çözüm getirmeye çalışırlar. Bu amaçla ilk önce çözümü aranan sorular tanımlanacaktır.

4.1. Petri Ağlarındaki Analiz Soruları

4.1.1. Güvenlik (Safeness)

Gerçek donanımları modellemeyi amaçlayan bir Petri ağında, en önemli özelliklerden biri güvenlidir. Bir Petri ağındaki herhangi bir işaretlemede bir yerde bulunabilecek jeton sayısı biri aşmıyorsa o yer güvenlidir. Sadece güvenli yerler barındıran bir Petri ağına *güvenli* denir. [1]

Tanım 4.1: Güvenlik

Bir $C = (P, T, F, W)$ Petri ağı ve bu ağa ilişkin M_0 ilk işaretlemesi tanımlı olsun. Bu ağdaki bir $p_i \in P$ yerinin güvenli olması her $M' \in R(C, M_0)$ için $M'(p_i) \leq 1$ koşulunun sağlanması gerekir. Bir ağdaki yerlerin tamamı güvenli ise bu Petri ağı güvenlidir.

Donanımsal cihazlarda güvenlik kavramı çok önemlidir. Bir yer güvenli ise o yerdeki jetonların sayısı ya 0 ya da 1'dir. Bu durumda o yer tek flip-flop ile uygulanabilir.

Yerler koşul, geçişler de olay olarak yorumlandığında, güvenlik özelliğinin sağlandığını görürüz. Bir koşul ya doğrudur (jeton vardır) ya da yanlıştır (jeton yok). Bu durumda birden çok jeton olması o koşul için anlamlı değildir. Koşul-olay sistemlerinde güvenlik kendiliğinden sağlanır.

Sıradan (tüm dal ağırlıkları 1 olan) bir Petri ağında bir yer güvenli olmaya zorlanabilir. Güvenli olmasını istenen her bir p_i yeri için yardımcı bir p_i' yeri tanımlanır. Güvenli bir Petri ağı için mevcut ağ üzerinde p_i yerini giriş ve çıkış yeri olarak kullanan geçişlere aşağıdaki işlemler uygulanmalıdır;

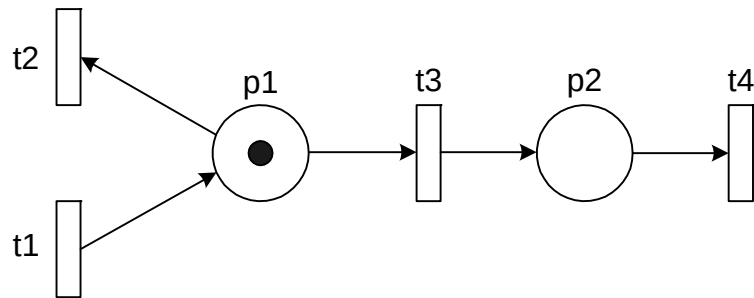
1. adım: $p_i \in I(t_j)$ ve $p_i \notin O(t_j)$ ise bu durumda,

p_i' 'nü $O(t_j)$ 'ye eklenir,

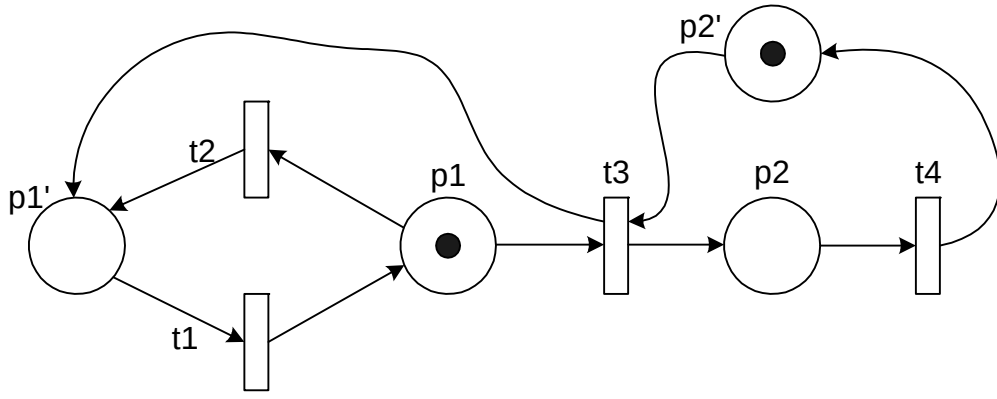
2. adım: $p_i \in O(t_j)$ ve $p_i \notin I(t_j)$ ise bu durumda,

p_i' 'nü $I(t_j)$ 'ye eklenir.

Yeni p_i' yerinin amacı “ p_i boş” koşulunu temsil etmektir. Bu durumda p_i ve p_i' yerleri eşleniktir denebilir. p_i 'den jeton eksiltlen bir geçiş p_i' 'ne jeton eklemelidir, benzer şekilde p_i' 'nden jeton eksiltlen bir geçiş p_i 'ye jeton eklemelidir. Sistemin düzgün çalışabilmesi için ilk işaretleme de güvenli olmalıdır, yani p_i ve p_i' çiftlerinden her birinde en az bir jeton bulunmalıdır. Şekil 4.1'de güvenli olmayan bir Petri ağı, Şekil 4.2'de ise aynı ağın eşlenik yerlerin eklenmesiyle güvenli hale getirildiği görülmektedir. [2]



Şekil 4.1 Güvenli Olmayan Petri Ağı



Şekil 4.2 Güvenli Petri Ağı

Şekil 4.1'deki Petri ağı güvenli hale getirilirken p_2' yerine bir adet jeton eklenmiştir. Yardımcı yerlerin eklenmesi Petri ağlarının çalışmasını etkilememektedir.

4.1.2. Sınırlılık (Boundedness)

Güvenlik, sınırlılık özelliğinin özel bir halidir. Bazı uygulamalarda yerlerin uygulanmasıyla ilgili gerçek kısıtlamanın güvenlik olmadığı görülmektedir. Güvenlik bir yerin flip-flop kullanılarak gerçekleştirilmesini sağlarken, bunun yerine sayaç kullanılabilir. Sayaç kullanıldığında ise kısıtlama sayacın alabileceği en büyük değer olacaktır. Bir yerin k -güvenli veya k -sınırlı olması, o yerde k 'dan daha fazla jeton bulunamayacağı anlamına gelir.

Tanım 4.2: Sınırlı Petri Ağı

Bir $C = (P, T, F, W)$ Petri ağı ve M_0 ilk işaretlemesi tanımlı olsun. Bu Petri ağında bir $p_i \in P$ yeri her $M' \in R(C, M_0)$ için $M' \leq k$ koşulunu sağlıyorsa o yere k -sınırlıdır denir.

1-sınırlı olan Petri ağlarına güvenli denir. Yukarıda yapılan tanımlamaya göre bir Petri ağında farklı yerlerin farklı üst sınırları olabilir. Bir ağdaki p_3 yeri 5-sınırlı iken aynı ağdaki p_8 yeri 3-sınırlı olabilir. Petri ağının sınırı ise en yüksek sınırı sahip yerdir. Bu yerin sınırı k olsun, bu durumda Petri ağı k -sınırlıdır denir. Bir Petri ağının sınırlı olması için tüm yerler sınırlı olmalıdır. Sınırlı bir Petri ağı donanımsal olarak gerçekleştirilebilir iken, sınırsız bir Petri ağı donanımsal olarak gerçekleştirilemezdir. Buna karşın sınırsız olan her Petri ağı yardımcı yerler yardımıyla sınırlı hale getirilebilir. Sınırsız bir (C, M) Petri ağından yardımcı yerler yöntemiyle sınırlı (C', M') sınırlı Petri ağına dönüştürmek için aşağıdaki adımlar izlenmelidir,

1. adım: Her p_i yeri için yardımcı bir p_i' yeri eklenir. p_i' 'nün ilk işaretleme

$$M_0'(p_i') = K(p_i) - M_0(p_i) \quad (4.1)$$

denklemini ile bulunur. 4.1 denkleminde yer alan $K(p_i)$, p_i yerinin sınırlandır.

2. adım: p_i' yerleri ile p_i 'nin giriş ve çıkış olduğu geçişler arasında

$$w(t, p_i') = w(p_i, t) \text{ ağırlıklı } (t, p_i') \text{ dalı,}$$

$$w(p_i', t) = w(t, p_i) \text{ ağırlıklı } (p_i', t) \text{ dalı tanımlanır.}$$

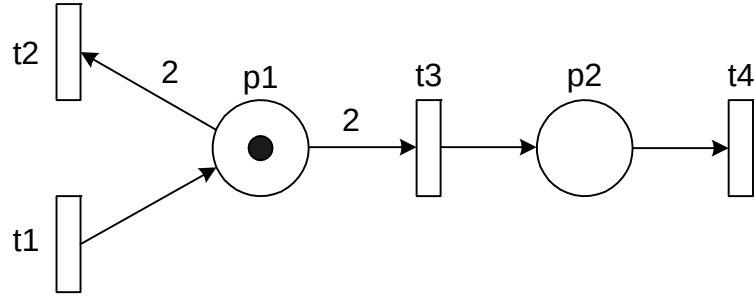
Bu durumda p_i ve p_i' yer çiftlerinde ateşlemelerden sonraki ve önceki jeton sayıları hep aynı kalacak ve $K(p_i)$ 'ye eşit olacaktır.

Örnek 4.1. Şekil 4.3'te görülen Petri ağına yardımcı yer yöntemini uygulansın. İlk $M_0 = (1, 0)$ işaretlemesinde sadece t_1 geçişi ateşlenebilir durumdadır. t_1 ateşlendikten sonra, $M_1 = (2, 0)$ işaretlemesi gerçekleşir ki bu durumda t_2 ve t_3 ateşlenebilir. t_2 ateşlenirse M_1 'den $M_2 = (0, 0)$ işaretlemesine, t_3 ateşlenirse $M_3 = (0, 1)$ işaretlemesine erişilir. Bu adımlar devam ettirilerek erişilebilecek tüm durumlar tespit edilebilir. Şekil 4.4'te bu ağın yardımcı yer yöntemiyle sınırlı hale dönüştürülmüş hali görülmektedir. İlk adımda p_1' ve p_2' yerlerindeki jeton sayısı hesaplanır, $K(p_1) = 2$ ve $K(p_2) = 1$ olarak verilmiştir,

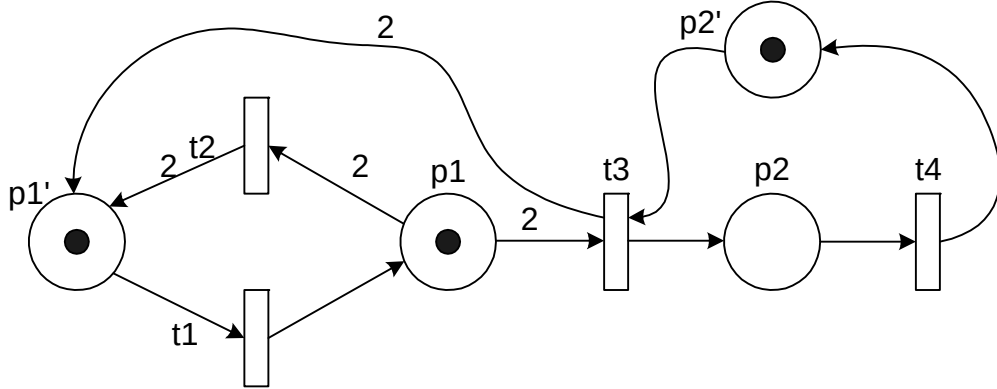
$$M_0(p_1') = K(p_1) - M_0(p_1) = 2 - 1 = 1$$

$$M_0(p_2') = K(p_2) - M_0(p_2) = 1 - 0 = 1$$

Bundan sonraki adım bazı geçiş ile bazı yardımcı yerler arasındaki dalları tanımlamaktır. $w(t_1, p_1) = 1$ olduğundan $w(p_1', t_1) = 1$ olacaktır. Benzer şekilde $w(t_3, p_1') = w(p_1, t_3) = 2$ ve $w(p_2', t_3) = w(t_3, p_2) = 1$ bulunacaktır. t_3 ateşlemesi p_1' 'den iki jeton alıp, p_2' 'ye bir jeton verdiğinden, t_3 'ten p_1' 'ne iki ağırlıklı bir dal, p_2' 'nden t_3 'e de bir ağırlıklı bir dal çizilir. Benzer şekilde (t_2, p_1') , (t_4, p_2') ve (p_1', t_1) dalları çizildiğinde yeni (C', M_0') ağı elde edilir.



Şekil 4.3 Sınırsız Petri Ağı



Şekil 4.4 Sınırlı Hale Getirilmiş Petri Ağı

4.1.3. Tutarlılık (Conservation)

Petri ağları kaynakların paylaşıldığı sistemleri modellemek amacıyla kullanılabilirler. Bir bilgisayardaki giriş/çıkış cihazlarına istek gönderme, kaynak tahsis etme ve serbest bırakma işlemleri Petri ağları kullanılarak modellenebilir. Bu sistemlerde kaynaklar jetonlarla ifade edilebilir. Üç tane yazıcıdan oluşan bir yazıcı havuzu tek bir yer ve üç jeton ile temsil edilebilir. Burada havuz yer olarak, yazıcılar da jeton olarak modellenmiştir. Bir yazıcıya gelen istek bu yerin giriş olduğu bir geçiştir, buna karşın da serbest bırakılan bir yazıcı bir geçiştten havuzu temsil eden yere bir çıkış dalı ile temsil edilebilir.

Bu tarz Petri ağlarında tutarlılık önemli bir özelliktir. Tutarlı sistemlerde kaynaklar jetonlarla ifade edilir, dolayısıyla jetonların çoğalıp kaybolmaması esastır. Bunu gerçekleştirmenin en basit yolu ağdaki jeton sayısının sabit kalmasını sağlamak olacaktır.

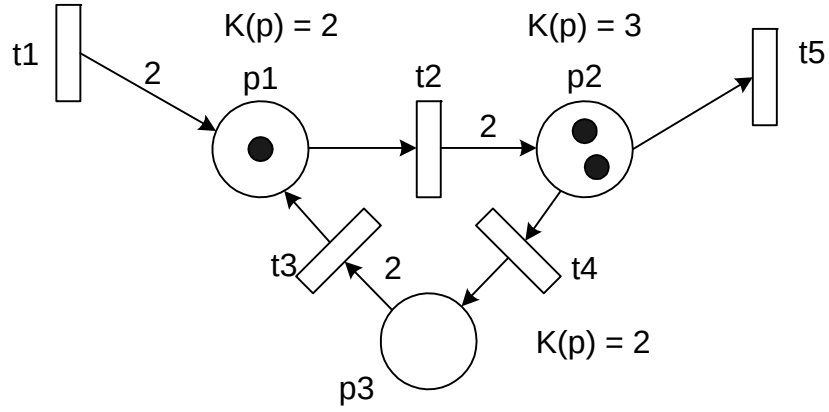
Tanım 4.3: Tutarlılık

Bir $C = (P, T, F, W)$ Petri ağı ve M_0 ilk işaretlemesi tanımlı olsun. Bu Petri ağı her $M' \in R(C, M_0)$ için,

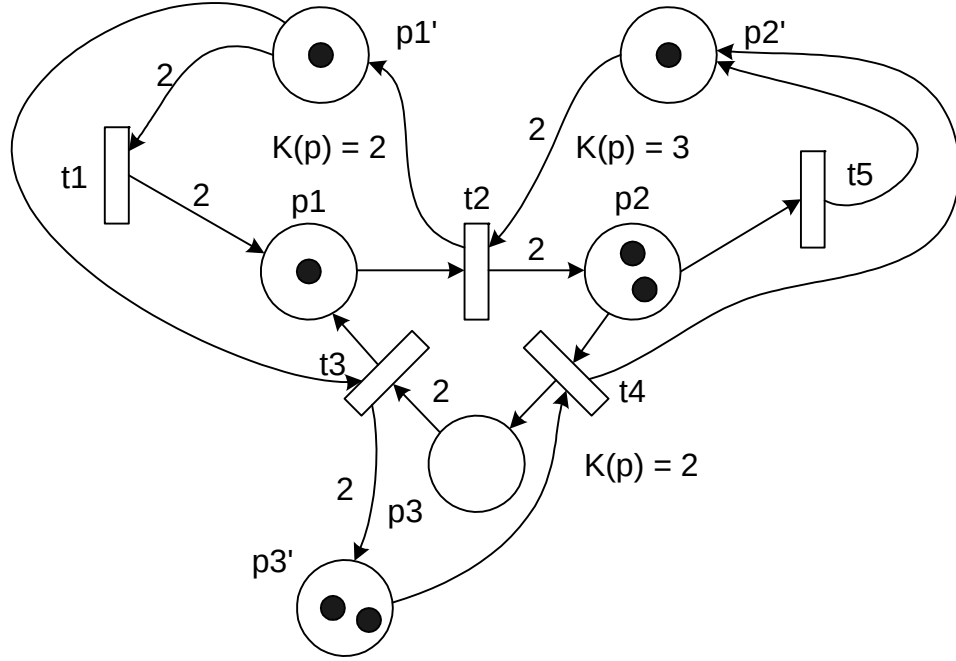
$$\sum M'(p_i) = \sum M_0(p_i), p_i \in P$$

koşulunu sağlıyorsa Petri ağına *katı tutarlı* denir.

Katı tutarlılık çok güçlü bir ilişkidir. Bu bağıntıdan hareketle bir geçişe ait çıkış dallarının ağırlıkları toplamı ile giriş dallarının ağırlıkları toplamının eşit olması gerektiği gösterilebilir ($I(t_j) = O(t_j)$ veya $\sum w(I(t_j), t_j) = \sum w(t_j, O(t_j))$). Bu varsayım yanlış olsaydı t_j geçişinin ateşlenmesi ağıdaki jeton sayısının değişmesine sebep olurdu. Şekil 4.5'te tutarsız bir Petri ağı görülmektedir. t_1 , t_2 ve t_3 'ün ateşlenmesi Petri ağını tutarsız hale getirecektir. Buna karşın yardımcı yer yöntemi kullanılarak elde edilen ve Şekil 4.6'da görülen Petri ağı her zaman tutarlı olacaktır.



Şekil 4.5 Tutarsız Petri Ağı



Şekil 4.6 Tutarlı Hale Getirilmiş Petri Ağı

Tutarlı bir Petri ağı modellediği kaynakları koruyabilmelidir. Yalnız her zaman jetonlar ve kaynaklar arasında bire bir eşleme yapılamaz. Bazı jetonlar program sayaçları ve diğer elemanları temsil ederken, bazı durumlarda tek bir jeton birkaç kaynağı temsil edebilir. Bazı durumlarda bu tek jeton bölünerek birden fazla jeton oluşturabilir. Bu durumda jeton ağırlıkları tanımlanabilir. Yalnız bir jetonun ağırlığı bulunduğu yer ile ilgilidir. Aynı yerde bulunan jetonlar arasında değer olarak bir farklılık yoktur. Bu durumda ağırlıkları jetonlarla ilişkilendirmek yerine ağdaki yerle ilişkilendirmek daha doğru olacaktır. Bir yer ağırlık vektörü $w_p = (w_{p1}, w_{p2}, \dots, w_{pn})$, ($n = 1, 2, \dots$) ile tanımlanabilir. w_{pi} her $p_i \in P$ yerinin ağırlığıdır. Bu durumda yerin özelliğine bağlı olarak $w_{pi} = 0, 1, 2, \dots$ değerlerini alabilir. 0 ağırlıklı bir yer etkisizdir. Ağırlıklı jetonların bulunduğu bir ağda bu ağırlıklar toplamı sabit kalmalıdır. Burada tanımlanan ağırlık kavramı, dalların ağırlıkları ile karıştırılmamalıdır. Benzer şekilde Renkli Petri Ağları ile de karıştırılmamalıdır.

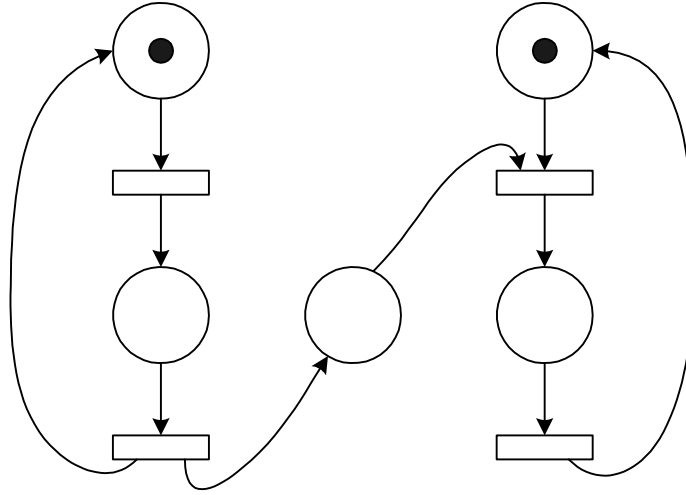
Tanım 3.4. Yer Ağırlık Vektörü (w_p)

Bir $C = (P, T, W, F)$ Petri ağı ve M_0 ilk işaretlemesi tanımlı olsun. Bu durumda Petri ağı $w_p = (w_{p1}, w_{p2}, \dots, w_{pn})$, $n = |P|$, $w_p \geq 0$, yer ağırlık vektörüne göre her $M' \in R(C, M_0)$ işaretlemesi için,

$$\sum w_{pi} M'(p_i) = \sum w_{pi} M(p_i), i = 1, 2, \dots, n$$

koşulu sağlanıyorsa bu Petri ağına yer ağırlık vektörüne göre tutarlı denir.

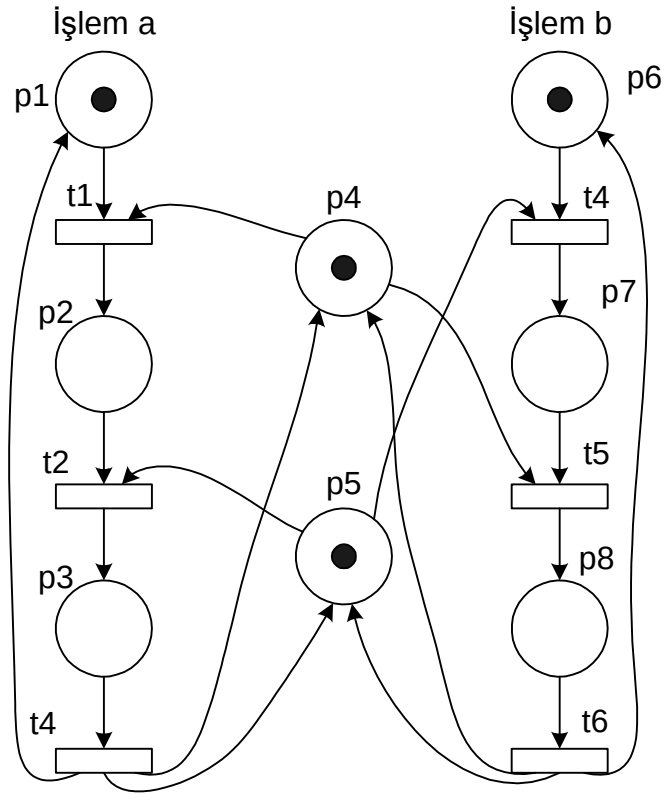
Katı tutarlı bir Petri ağı $(1, 1, \dots, 1)$ yer ağırlık vektörüne göre tutarlılık ifade eder. $(0, 0, \dots, 0)$ yer ağırlık vektörüne sahip tüm ağlar tutarlıdır. Dolayısıyla 0 yer ağırlık vektörü Petri ağları için bir tutarlılık ölçüsü oluşturamaz. Yer ağırlık vektörleri 0 olmamalıdır. Şekil 4.7’de tutarsız bir Petri ağı görülmektedir.



Şekil 4.7 Tutarsız Petri Ağı

4.1.4. Canlılık (Liveness)

Tutarlılığın göz önüne alınmasının bir sebebi sistemlerdeki kaynak dağılımıdır. Kaynak dağılımından kaynaklanabilecek başka bir problem ise kilitlenmelerdir (deadlock). Elektronik sistemlerde ve bilgisayar sistemlerinde kilitlenme sorunu birçok kez farklı açılardan incelenmiştir. Bu sorunu daha iyi anlamak için basit bir örneğe başvurulacaktır. Örnek sistemde q ve r kaynakları, a ve b ise işlemleri temsil etmektedir. Her iki işlem de aynı kaynağı kullanacağı durumlarda kaynaklar paylaşılmalıdır. Bunu başarabilmek için bir işlem kaynağı talep etmelidir, kullandıktan sonra da serbest bırakmalıdır. a işleminin ilk başta q kaynağını daha sonrada r kaynağını kullanıp serbest bıraktığı varsayalım. b işlemi ise ilk başta r kaynağını daha sonra da q kaynağını kullanıp serbest bıraktığı varsayalım. Şekil 4.8’de bu sistem Petri ağı ile ifade edilmektedir.



Şekil 4.8 Sistemin Petri Ağı İfadesi

İlk işaretlemeye göre $q(p_4)$ ve $r(p_5)$ kaynakları kullanılabilir durumunda iken hem a hem de b işlemi çalıştırılabilir. Bu ağın bir çalışması $t_1, t_2, t_3, t_4, t_5, t_6$ ateşleme dizisi olabilir, bir diğer çalışması da $t_4, t_5, t_6, t_1, t_2, t_3$ ateşleme dizisi olabilir. Her iki çalışma da kilitlenmeyen çalışmalardır. Buna karşın t_1, t_4 ateşlemeleri ile başlayan işlem göz önüne alındığında a işlemi q kaynağına sahip ve r'yi istemekte, b işlemi de r kaynağına sahip ve q'yü istemektedir. Bu durumda sistem kilitlenir ve her iki işlem de durur.

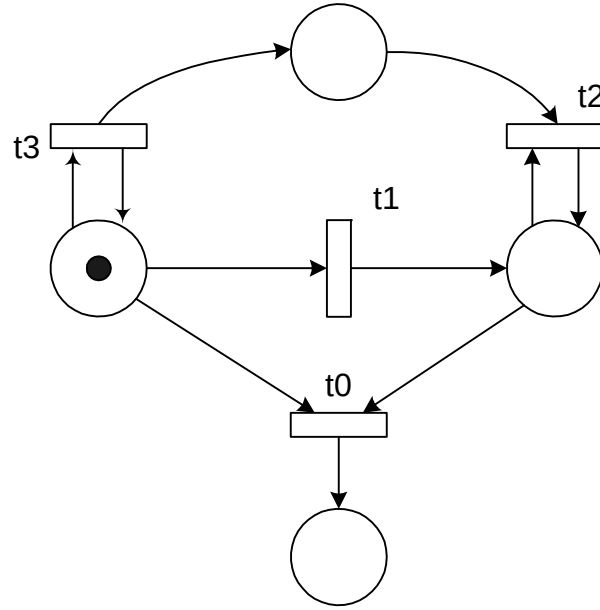
Bir Petri ağında kilitlenme ateşlenemeyen bir veya birkaç tane geçiştir. Şekil 4.8'de t_2 ve t_5 geçişleri ateşlenemediği durumlarda kilitlenme oluşur. Bir geçiş kilitlenmemişse canlıdır. Bu tanım izinli bir geçiş ile karıştırılmamalıdır. Canlı bir geçiş izinli hale getirilebilecek geçiş anlamına gelmektedir. Bir C Petri ağında ve M_0 ilk işaretlemesi göz önüne alındığında, bir t_j geçişinin potansiyel olarak ateşlenebilmesi için öyle bir $M' \in R(C, M_0)$ olmalıdır ki t_j, M' işaretlemesi altında izinli olsun. Bir geçişin canlı olabilmesi için $R(C, M_0)$ işaretlemelerinin her birinde potansiyel olarak ateşlenebilir olmalıdır. Bu durumda bir geçiş canlı ise, bir Petri ağındaki M işaretlemesinden M' işaretlemesine giden bir ateşleme dizisi vardır.

Commoner tarafından 1972 yılında canlılıkla ilgili farklı seviyeler tanımlanmıştır. Bu seviyeler şu şekildedir;

0. Seviye: bir t_j geçişi asla ateşlenemez ise 0. seviyede canlıdır.
1. Seviye: bir t_j geçişinin ateşlenebilme ihtimali varsa 1. seviyede canlıdır. Bu durumda öyle bir $M' \in R(C, M_0)$ olmalıdır ki bu işaretleme altında t_j ateşlenebilir olsun.
2. Seviye: bir t_j geçişi belli bir ateşleme dizisi altında n defa ateşlenebiliyorsa 2. seviyede canlıdır.
3. Seviye: bir t_j geçişi, sonsuz bir ateşleme dizisi altında sonsuz defa ateşlenebiliyorsa 3. seviyede canlıdır.
4. Seviye: her $M' \in R(C, M_0)$ işaretlemesi altında t_j 'yi ateşlenebilir yapan bir σ ateşleme dizisi varsa, $\delta(M', \sigma)$, bu durumda t_j geçişi 4. seviyede canlıdır.

“0.” seviyede canlı bir geçiş çalıştırılmaz, yani ölüdür. “4.” seviyedeki bir geçiş ise canlıdır. Bir ağdaki her geçiş i. seviyede canlı ise o Petri ağı da i. seviyede canlıdır.

Şekil 4.9’da bu seviyelerin hepsi gözlenebilmektedir. t_0 geçişi asla ateşlenemez dolayısıyla 0. seviyede canlıdır (ölüdür). t_1 ateşlemesi sadece bir kez ateşlenebilir, dolayısıyla 1. seviyede canlıdır. t_2 geçişi t_1 ateşlendiği sürece t_3 ’ten sonra ateşlenebilir. t_1 bir defa ateşlenirse t_2 bir daha ateşlenemez, yine ateşlenebilmesi için t_3 ’ün de ateşlenmesi gerekir bu durumda t_2 2. seviyede canlıdır. t_3 ise t_1 ateşlenmediği sürece sonsuz defa ateşlenebilir. Buna karşın t_1 ateşlendikten sonra bir daha ateşlenemez, bu yüzden t_3 3. seviyede canlıdır.



Şekil 4.9 Petri Ağındaki Çeşitli Canlılık Seviyeleri

4.1.5. Erişilebilirlik ve Kapsanabilirlik (Reachability and Coverability)

Daha önceden incelenen problemlerde erişilebilecek tüm işaretlemeler göz önüne alınmıştı. Bu bölümde erişilebilirlik sorusu daha derinlemesine incelenecektir.

Tanım 4.5: Erişebilirlik Problemi

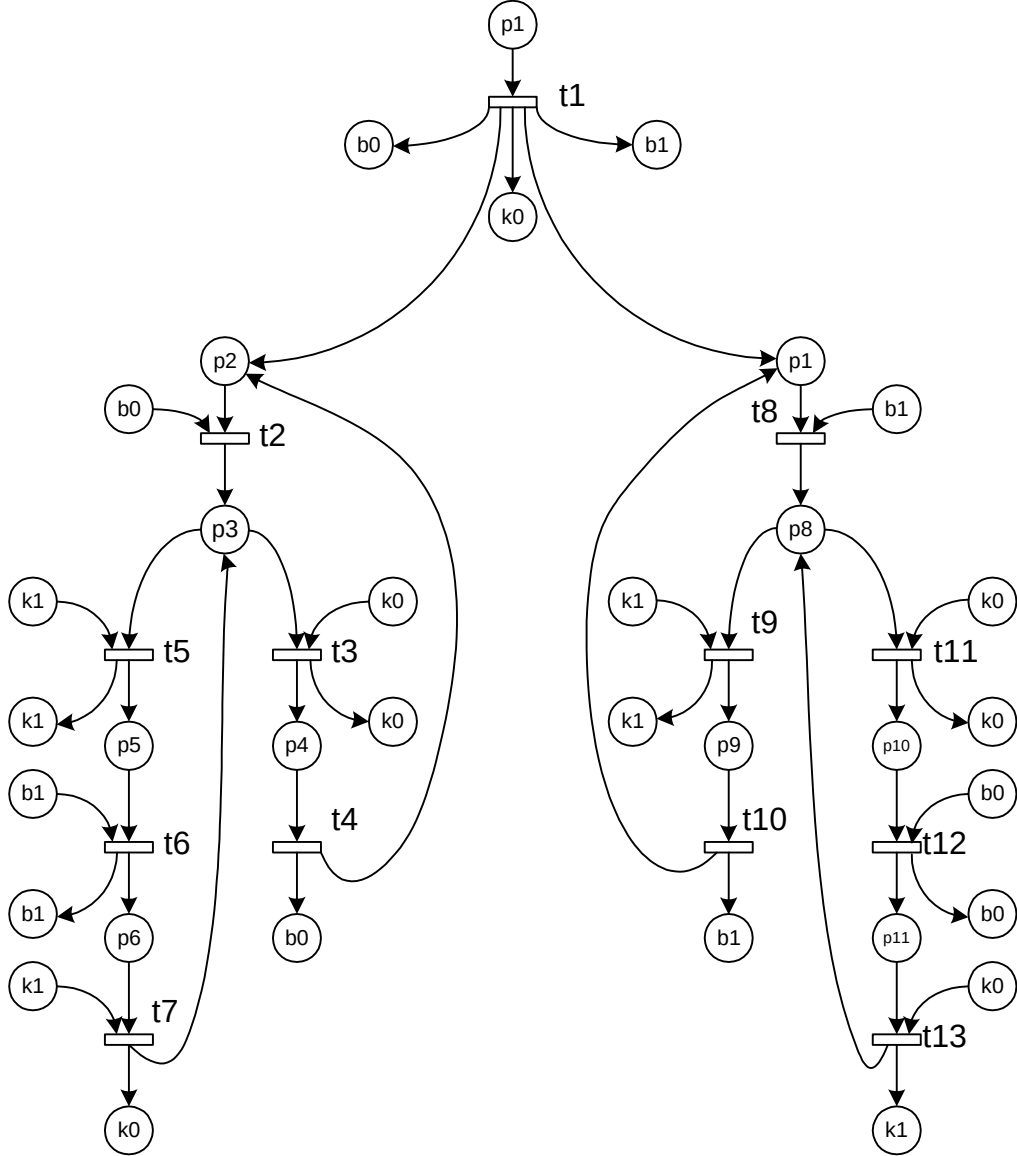
Bir $C = (P, T, F, W)$ Petri ağı ve M_0 ilk işaretlemesi verilmiş olsun. Bu durumda bir $M' \in R(C, M_0)$ mıdır?

Birçok problem erişebilirlik göz önüne alınarak çözülebilir. Şekil 4.8'deki Petri ağında $(0, 1, 0, 0, 0, 0, 1, 0)$ işaretlemesine ulaşılabilirse bu durumda kilitlenme olacaktır.

Şekil 4.10'da mutual-exclusion problemi çözülmeye çalışılmaktadır. Mutual-exclusion problemi çok işlemcili sistemlerde, işlemci kaynaklarına erişimi denetlemek amacıyla geliştirilmiştir. p_4 ve p_9 yerleri karşılıklı olarak birbirilerinin çalışmasını engellemekteler. Bu şekilde denetim sağlanmış olur. Bu durumda bu ağda $p_4 = 1$ ve $p_9 = 1$ işaretlemesine erişilip erişilemeyeceği incelenmelidir. Bu soru biraz daha farklı bir problem olan *kapsanabilirlik* problemine dönüşecektir. Bir M'' işaretlemesi M' işaretlemesini kapsayabilmesi için $M'' \geq M'$ olması yeterlidir.

Tanım 4.6: Kapsanabilirlik Problemi

Bir $C = (P, T, F, W)$ Petri ağı, M_0 ilk işaretlemesi ve $M' \in R(C, M_0)$ işaretlemesi tanımlı olsun. Bu durumda $M'' \geq M'$ koşulunu sağlayan bir $M'' \in R(C, M_0)$ işaretlemesi bulunabilir mi?



Şekil 4.10 Mutual Exclusion Problemine Çözüm Olan Petri Ağı

Şekil 4.10'da Hyman'nın 1966 yılında mutual-exclusion problemi ile ilgili geliştirdiği çözüm görülmektedir. k_0 , k_1 , b_1 ve b_0 yerleri sürekli kullanıldığından bunlar tekrar tekrar kullanılmıştır.

Yukarıda bahsedilen problemler daha da genişletilebilir. İncelemelerde sadece ağıdaki belli yerlerin durumu incelenmek istenebilir. Bu durumda problem alt işaretlemelerde erişilebilirlik ve alt işaretlemelerde kapsanabilirlik kavramlarına

doğru genişletilebilir. Bunlara erişebilirlik kümesi problemi ve kapsanabilirlik kümesi problemleri denir.

4.1.6. Ateşleme Dizileri

Analizlerde durumları kullanmak yerine ateşleme dizileri kullanılabilir. t_j ateşlenebilir mi sorusu sorulduğunda bu doğrudan canlılık kavramı ile ilişkilendirilir. Bunun yanı sıra, belli bir ateşleme dizisi gerçekleştirilebilir mi veya belli bir geçiş kümesi ateşlenebilir mi sorulabilecek diğer sorulardır.

4.1.7. Eşitlik ve Altküme Problemleri

Bu sorular optimizasyonla ilgili konulardan kaynaklanır. Bir Petri ağı, ateşleme dizisi ve erişebilirlik kümesi tarafından belirlenen bir davranışı gösterirken bu davranış değiştirilmeden o Petri ağı değiştirilip optimize edilebilir mi? Bu işlem kullanılmayan veya anlamsız geçiş ve yerlerin silinmesini, bazı geçişlerin de baştan tanımlanmasını içerebilir. Aynı sayıda geçişe, farklı sayıda yere sahip iki Petri ağının aynı ateşleme dizisine, veya aynı sayıda yere, farklı sayıda geçişe sahip iki Petri ağının aynı erişebilirlik kümesine sahip olduğu gösterilebilir mi? Bu durumda Petri ağı değiştirilerek paralellikler artırılabilir veya başka ilerlemeler kaydedilebilir. Bu durumda iki Petri ağının eşit olup olmadığı veya biri diğerinin alt kümesi olup olmadığını incelenmelidir.

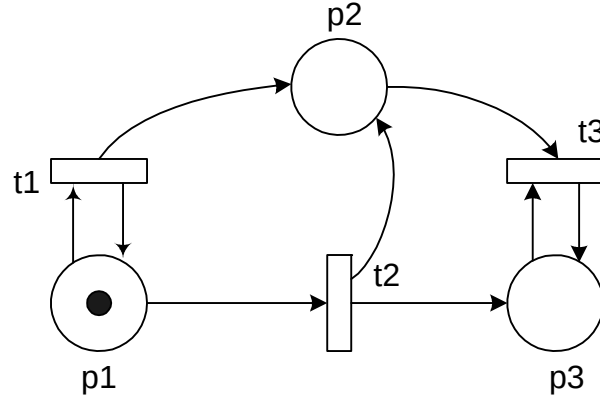
4.2. Analiz Yöntemleri

Önceki bölümde bahsedilen problemlerin yanı sıra gerçekleştirilmek istenen uygulamaya veya analizlere bağlı olarak birçok başka problemten bahsedilebilir. Bu bölümde daha önce tanımlanan problemleri çözmek amacıyla kullanılabilecek yöntemler incelenecektir. Özellikle bilgisayar programları ile kolay uygulanabilen yöntemler öncelikli olarak ele alınacaktır. [1]

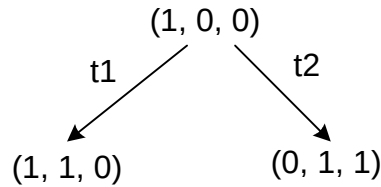
Petri ağı analizlerinde kullanılan iki temel analiz yöntemi geliştirilmiştir. Petri ağları ile ilgili ilk yöntem erişebilirlik ağacı yöntemidir, ikinci yöntem ise lineer cebirsel denklemlere dayanan yöntemdir. Sırasıyla yöntemler tanıtılacak ve bunların problemlerin çözümünde nasıl kullanılabileceği tartışılacaktır.

4.2.1. Erişilebilirlik Ağacı Yöntemi

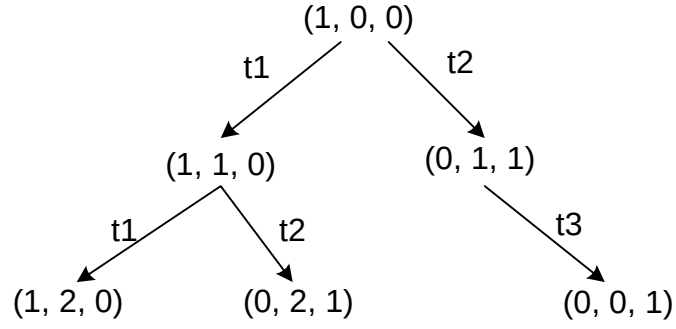
Erişilebilirlik ağacı bir Petri ağındaki erişilebilen işaretlemeler kümesini ifade eder. Şekil 4.11'deki Petri ağının ilk işaretlemesi $M_0 = (1, 0, 0)$ olarak verilmiştir. İlk işaretleme altında t_1 ve t_2 geçişleri izinlidir. Şekil 4.12'de her iki geçişin ateşlenmesi sonucu oluşan yeni işaretlemeler görülmektedir. Bu yeni işaretlemeler, ilk işaretlemeye birer dal ile bağlanır ve dalın yanına ateşlenen geçiş işaretlenir. Bir sonraki adımda oluşan yeni işaretlemelerden erişilebilecek diğer işaretlemeler göz önüne alınmalıdır. $(1, 1, 0)$ işaretlemesi altında t_1 'i ateşlenerek $(1, 2, 0)$ durumuna ve t_2 'yi ateşlenerek $(0, 2, 1)$ durumunda gidilebilir. $(0, 1, 1)$ işaretlemesi altında izinli t_3 geçişi ateşlenerek $(0, 0, 1)$ işaretlemesine gidilebilir. Bu da Şekil 4.13'te görülen ağacı oluşturur. Elde edilen üç yeni işaretleme üzerinde bir önceki adımlar tekrar edilirse Şekil 4.14'te görülen işaretleme ağacı yapısı elde edilebilir. $(0, 0, 1)$ işaretlemesi ölü bir işaretlemedir. $(0, 2, 1)$ işaretlemesi altında izinli t_3 geçişi ateşlendiğinde $(0, 1, 1)$ işaretlemesi elde edilir, oysa ilk işaretlemeden sonra bu durum elde edilmişti.



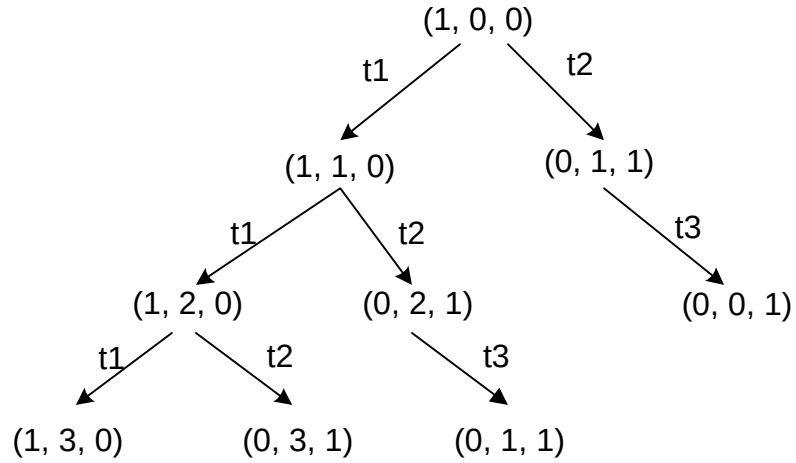
Şekil 4.11 İşaretli Bir Petri Ağı



Şekil 4.12 Erişilebilirlik Ağacının Oluşturulmasında İlk Adım

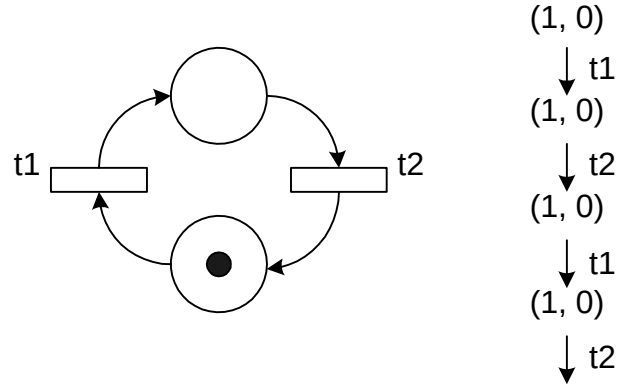


Şekil 4.13 Erişilebilirlik Ağacının Oluşturulmasında İkinci Adım



Şekil 4.14 Erişilebilirlik Ağacının Oluşturulmasında Üçüncü Adım

Bu işlem sürekli olarak yapıldıkça erişilebilecek tüm işaretlemeler üretilecektir. Bazı durumlarda ulaşılan erişilebilirlik ağacı sonsuz elemanlı olabilir. Sonlu erişilebilirlik kümesine sahip bir Petri ağı bile sonsuz bir ağaca sahip olabilir. Ağaçta aynı zamanda mümkün olan tüm ateşleme dizilerinin bilgisi de bulunacaktır. Ağacın kökünden başlayan her yol bir ateşleme dizisine işaret eder. Ağacı kullanışlı hale getirmek amacıyla boyutunu sonlu hale getirmenin bir yolu bulunmalıdır. Şekil 4.15'te sonlu elemanlı bir Petri ağı ve ona ilişkin sonsuz elemanlı ağaç görülmektedir.



Şekil 4.15 Sonsuz Erişilebilirlik Ağacına Sahip Bir Petri Ağı

Sonsuz elamanlı bir ağacın gösterimi sonlu hale indirgemek birkaç yolla gerçekleştirilebilir. Oluşturulan öncü düğümleri sınırlamak amacıyla bir yöntem belirlenmelidir. İlk olarak ölü olan işaretlemeler belirlenir, bunlar aynı zamanda terminal olarak adlandırılır. Bir diğer işaretleme sınıfı da ağaçta daha önceden oluşmuş olan düğümlerdir. Bunlara tekrarlanan düğümler denir ve ilkinden sonra oluşan düğümler ilkinde temsil edilir. Şekil 4.14'te görülen $(0, 1, 1)$ düğümü daha önceden oluştuğundan dolayı tekrar üretilmeyecektir. Bunun yerine ilk oluşan durum kullanılacaktır.

Son incelenecek işaretleme sınıfı sonsuz defa tekrar eden işaretlemelerdir. M işaretlemesinden M' işaretlemesine giden bir σ ateşleme dizisini göz önüne alınsın, öyle ki $M' > M$. Burada büyüklük işareti ile M' 'nin bazı yerlerinde M 'den daha çok jeton bulunduğu belirtilmektedir, yalnız bu M' 'de jeton olmayan yerlere jeton gelen durumları içermez. Bu durumda σ ateşleme dizisi tekrar ateşlenebilir. Bu dizinin ilk defa ateşlenmesi M işaretlemesine $(M' - M)$ jeton eklenmiştir. İkinci defa ateşlendiğinde de M' işaretlemesine $(M' - M)$ jeton ekleyecektir. σ ateşleme dizisi n kez ateşlenirse ağın durumu $M + n(M' - M)$ olacaktır. Bu durumda Petri ağına jeton ekleyen σ ateşleme dizisi istenildiği kadar çok defa ateşlenebilir. Şekil 4.11'de görülen Petri ağında t_1 birçok kez ateşlenebilir, her ateşleme de p_1 'deki jetonların sayısını bir artıracaktır. Bu şekilde döngüye girip sonsuza varan yerlerdeki jeton sayısı “ w ” ile ifade edilir. “ w ” sonsuz anlamına gelip gerektiği kadar büyük yapılabilecek bir sayıyı ifade etmektedir. w sayısı bir a sabit sayısı ile karşılaştırılırsa aşağıdaki sonuçlar elde edilir.

$$w + a = w$$

$$w - a = w$$

$$a < w$$

$$w \leq w$$

Bu temel kavramlar tanımlandıktan sonra erişilebilirlik ağacı oluşturulabilir. Ağaçtaki her i düğümü bir genişletilmiş $M[i]$ işaretlemesini temsil eder. Genişletmenin amacı işaretlemenin w sembolünü de barındırabilmesidir. Ağaçtaki tüm düğümler öncü düğüm, terminal düğüm, tekrar düğüm veya iç düğüm olarak sınıflandırılabilir. Öncü düğümler algoritma tarafından henüz sınıflandırılmamış düğümleri temsil eder ve algoritma işlemleri sonucunda terminal, tekrar veya iç düğümlerden birine dönüştürülür.

Erişilebilirlik ağacı algoritması ilk işaretlemeyi ağacın kökü olarak yerleştirerek işe başlar ve bir öncü düğüm yerleştirir. Öncü düğüm kaldığı sürece algoritma çalışır.

Tanım 4.7: Erişilebilirlik Ağacı Algoritması

“ x ” ile algoritma tarafından işlenecek olan öncü düğüm temsil edilsin.

1. Eğer ağaçta öncü düğüm olmayan ve $M[x] = M[y]$ koşulunu sağlayan bir y düğümü varsa bu durumda x düğümü *tekrar* düğümdür.
2. $M[x]$ işaretlemesinde hiçbir geçiş izinli değil ise bu durumda x bir *terminal* düğümdür.
3. $M[x]$ işaretlemesi altında ateşlenebilir durumda olan her $t_j \in T'$ 'ye ($\delta(M[x], t_j)$ tanımlı olan) erişilebilirlik ağacında yeni bir z düğümü tanımlansın. Bu durumda yeni $M[z]$ işaretlemeleri ile ilgili p_i yerlerinin durumu şu şekilde belirlenir,

a. Eğer $M[x][p_i] = w$ ise bu durumda $M[z][p_i] = w$ olur,

b. Kök düğümünden x düğümüne dek, $M[y] < \delta(M[x], t_j)$ ve $M[y]_i < \delta(M[x], t_j)$ koşulunu sağlayan bir y düğümü varsa $M[z][p_i] = w$ olur.

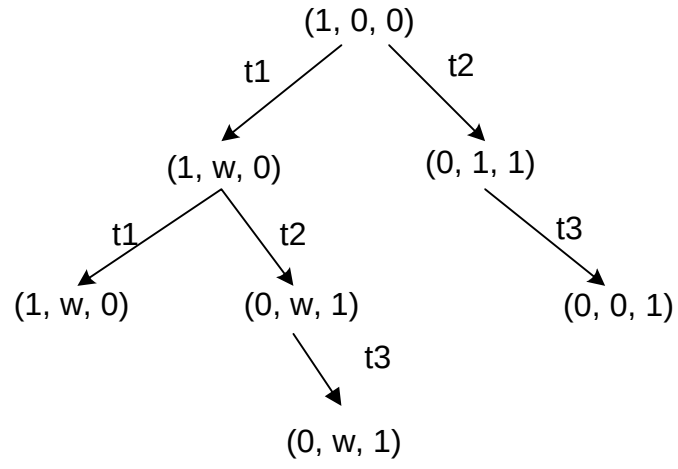
Bu tanım döngüleri tespit eder,

c. Diğer durumlarda $M[z][p_i] = \delta(M[x], t_j)$ olur.

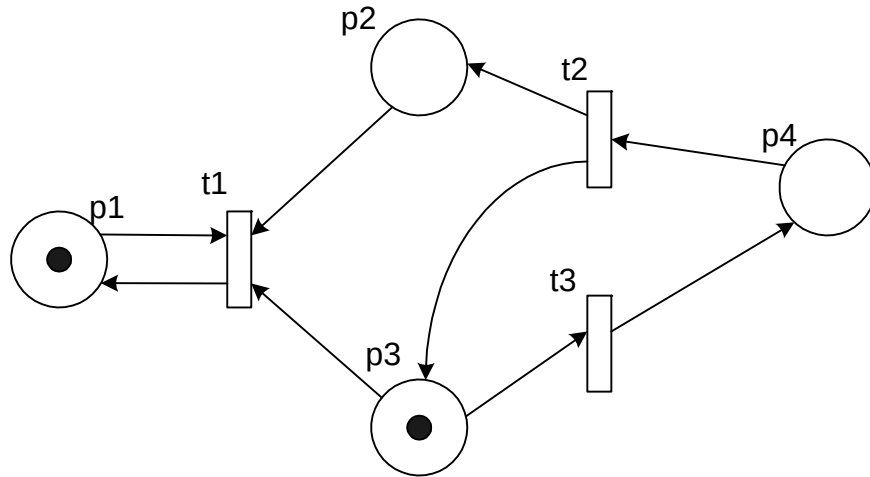
x düğümünden z düğümüne t_j isimli dal çizilir. X düğümü iç düğüm olarak, z düğümü de öncü düğüm olarak tanımlanır.

Tüm düğümler terminal, tekrar veya iç düğüm olarak belirlendiğinde algoritma tamamlanır. [2]

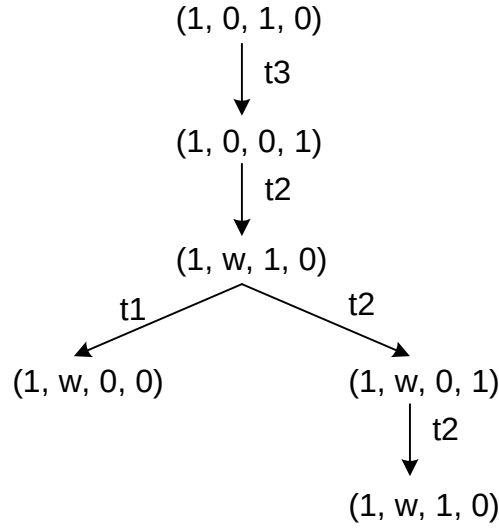
Şekil 4.16’da, Şekil 4.11’deki Petri ağının erişilebilirlik ağacı görülmektedir. Şekil 4.18’de, Şekil 4.17’deki Petri ağının erişilebilirlik ağacı görülmektedir.



Şekil 4.16 Şekil 4.11’deki Petri Ağının Erişilebilirlik Ağacı



Şekil 4.17 Erişilebilirlik Ağacı İncelenecek Petri Ağı



Şekil 4.18 Şekil 4.17'deki Petri Ağının Erişilebilirlik Ağacı

Erişilebilirlik ağacı Petri ağlarının özelliklerini analiz etmek için kullanılabilecek oldukça iyi bir araçtır. Bundan sonraki bölümlerde bu yöntemin çeşitli problemleri çözmek amacıyla nasıl kullanılabileceği araştırılacaktır.

4.2.2. EAY ve Problemlere Getirdiği Çözümler

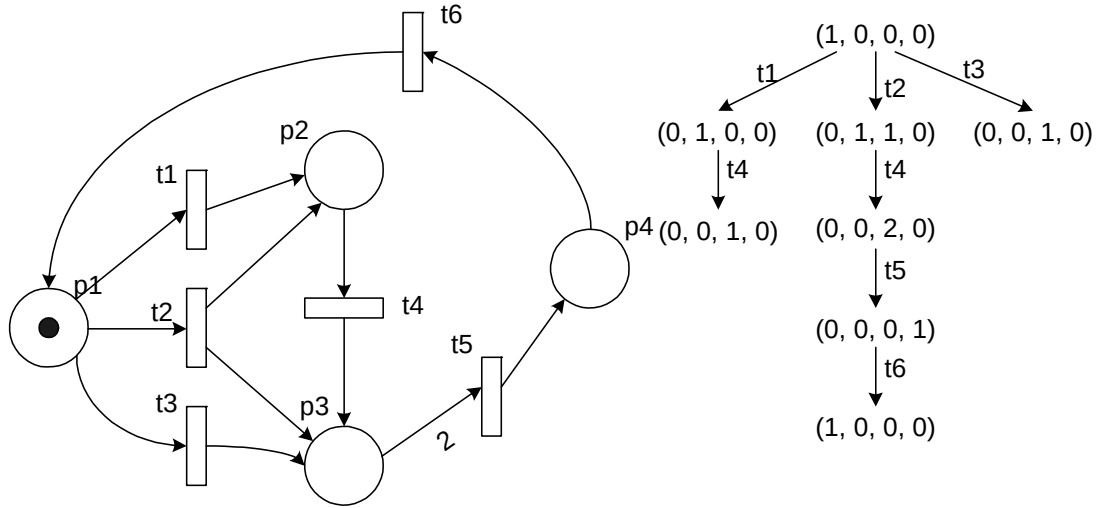
Güvenlik ve Sınırlılık Sorunu

Bir Petri ağının herhangi bir yerindeki jetonların sayısı asla 1'i geçmediği ağlar güvenlidir, benzer şekilde jetonların sayısı k 'yı geçmediği durumlarda Petri ağları k -sınırlıdır. Her iki özellik de erişilebilirlik ağacı kullanılarak test edilebilir. “Bir Petri ağının sınırlı olması, erişilebilirlik ağacında asla w işareti olmamasına bağlıdır”. w 'nın varlığı ağdaki jeton sayısının sınırsız olduğunu gösterir. w işareti aynı zamanda hangi yerlerin sınırsız olduğunu da gösterir.

Benzer şekilde, bir Petri ağı sınırsız ise bu durumda erişilebilecek işaretleme sayısı da sınırsızdır. Erişilebilirlik ağacı sonlu olduğundan, w işareti sonsuz sayıdaki erişilebilir işaretlemeleri temsilen bulunmalıdır. Kısaca w 'nin varlığı o ağın sonsuz sayıda işaretlemeye sahip olduğunu gösterir.

Eğer Petri ağı sınırlı ise ve w sembolü işaretleme ağacında yoksa bu durumda Petri ağı sonlu durumlu bir sistemi ifade eder. Bu durumda erişilebilirlik ağacı bir durum grafi olup erişilebilecek tüm işaretlemeleri barındıracaktır. Bundan sonra diğer tüm sorular bu sonlu yapı incelenerek cevaplanabilir.

Bir yerin alabileceği en yüksek jeton sayısını tespit etmek için erişilebilirlik ağacındaki tüm işaretlemeler taranıp o yerle ilgili en yüksek değer bulunur. Bu sayı o yerin sınır değeridir. Eğer ağdaki tüm yerlerin sınırları 1 ise bu durumda ağ güvenli bir ağıdır. Şekil 4.19’da erişilebilirlik ağacı kullanılarak sınırlılığın tespiti yapılabilir.



Şekil 4.19 Erişilebilirlik Ağacı Kullanılarak Petri Ağının Sınırlılığını Belirleme

Sınırsız olan Petri ağlarında bile erişilebilirlik ağacı kullanılarak tek tek veya belli bir yerler kümesinin sınırlılık durumu kesin olarak belirlenebilir.

Tutarlılık

Bir Petri ağı jeton kaybetmiyorsa veya jeton kazanmıyorsa buna karşın jetonları ağda dolaştırabiliyorsa tutarlıdır. Daha önceden yer ağırlık vektörü ile tanımlanan Petri ağlarında, tutarlılığın sağlanabilmesi amacıyla jetonların ağırlıklı toplamının değişmemesi gerektiği belirtilmişti.

Tutarlılık, erişilebilirlik ağacı kullanılarak test edilebilir. Erişilebilirlik ağacı sonlu olduğundan her işaretleme ile ilgili ağırlıklı toplam bulunabilir. Toplamlar her işaretleme için eşit ise bu durumda ağ tutarlıdır, değil ise tutarsızdır denir.

w sembolünün olduğu ağaçlar dikkatle değerlendirilmelidir. w 'nın olduğu yerlerin ağırlığı 0 ise bu ağ tutarlıdır. w 'nın olduğu en az bir yerin ağırlığı 0 değil ise bu durumda ağ tutarsız olur.

Bir işaretlenmiş ağdaki yer ağırlık vektörü w_p 'ye göre x işaretleme altında jetonların ağırlıklı toplamı sabit kalmalıdır,

$$w_{p1}.M[x]_1 + w_{p2}.M[x]_2 + ... + w_{pn}.M[x]_n = S \quad (4.2)$$

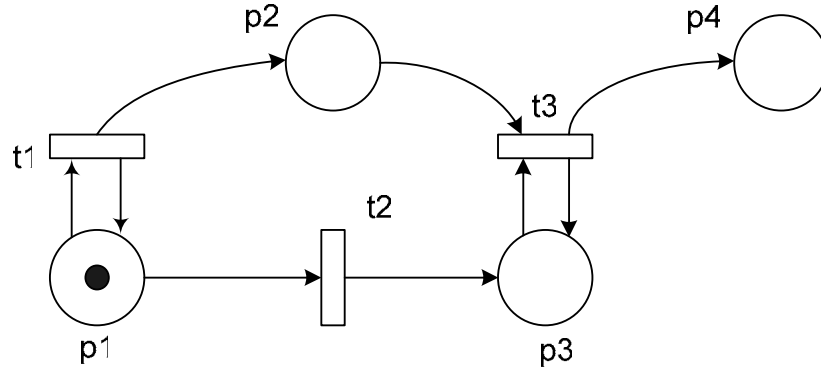
Bu durumda erişilebilirlik ağacında k düğüm bulunan bir Petri ağında yukarıdaki her x işaretlemesinde toplam aynı S değeri olmalıdır.

Kapsanabilirlik

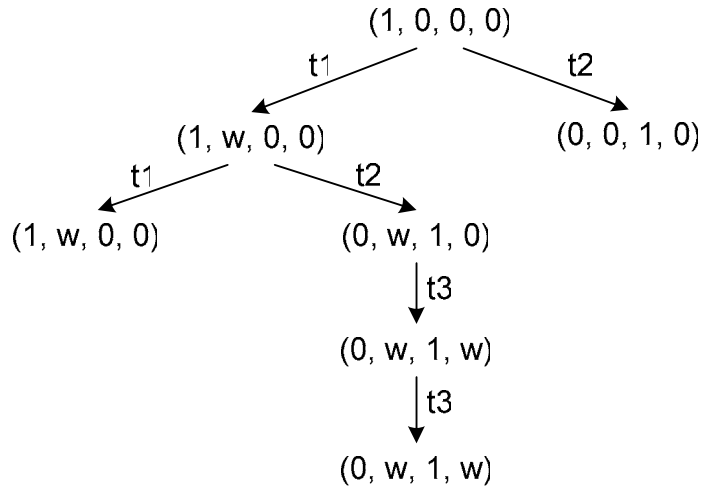
Erişilebilirlik ağacı kullanılarak tam olarak çözülebilecek son problem kapsanabilirlik problemidir. Bu problemde, verilen bir M' işaretlemesi için $M'' \geq M'$ koşulunu sağlayan bir M'' işaretlemesi olup olmadığı araştırılır. Erişilebilirlik ağacı kullanılarak bu problem çözülebilir. M_0 ilk işaretlemesinden yola çıkarak erişilebilirlik ağacı tanımlansın. Bu durumda öyle bir x düğümü olmalıdır ki $M[x] \geq M'$ olsun. Böyle bir düğüm bulunamazsa M' hiçbir erişilebilir işaretleme tarafından kapsanamaz, buna karşın bu koşulu sağlayan bir işaretleme varsa $M[x]$, M' işaretlemesini kapsar.

Kökten kapsayan işaretlemeye giden yol bir ateşlenen geçişler dizisi tanımlar. w sembolü olan bir işaretleme sonsuza işaret eder. Kapsayan işaretlemenin bir elemanı w ise bu durumda kökten o işaretlemeye bir döngü tanımlanmalıdır. Bu durumda döngü, incelenen yerdeki jeton sayısı istenen değerden daha büyük olana kadar tekrarlanmalıdır.

Kapsayan işaretlemelerde birden fazla w varsa, bu durumda istenen değerleri elde etmekte kullanılan döngüler hesaplanırken, bunların birbirine etkisi de göz önüne alınmalıdır. Şekil 4.20'de görülen Petri ağının erişilebilirlik ağacı Şekil 4.21'de görülmektedir. (0, 14, 0, 7) işaretlemesi erişilebilirlik ağacında kapsanabilirdir. Bu işaretlemeyi kapsayabilmek için birkaç tane t_1 , onu izleyen bir t_2 ve ondan sonra da birkaç tane t_3 ateşlenmelidir. Burada sorun t_1 ve t_3 'lerin ateşleme sayısını belirlemektir. p_2 'de 14 adet jeton istediğimizden ve t_1 p_2 'ye bir jeton eklediğinden 14 adet t_1 denenebilir. Buna karşın 7 adet t_3 'e ihtiyacımız var ki bu da her defasında p_2 'den 1 jeton eksildir. Bu durumda çözüm en az 21 adet t_1 , 1 adet t_2 ve 7 adet t_3 ateşlemesi olacaktır.



Şekil 4.20 İncelenecek Petri Ağı

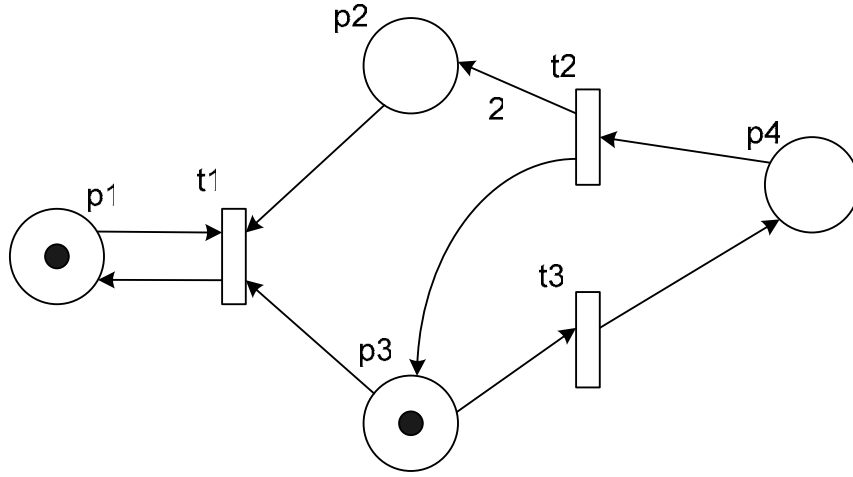


Şekil 4.21 Şekil 4.20'deki Petri Ağının Erişilebilirlik Ağacı

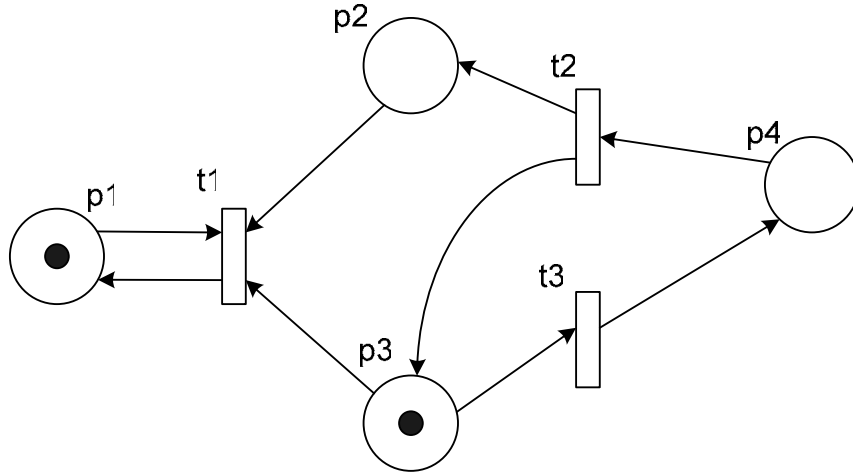
4.2.3. Erişilebilirlik Ağacının Kısıtlamaları

Önceki bölümlerde gösterildiği gibi erişilebilirlik ağacı yöntemi güvenlik, sınırlılık, tutarlılık ve kapsanabilirlik problemlerinin çözümünde kullanılabilir. Buna karşın genel olarak erişilebilirlik veya canlılık problemlerini çözmede veya ateşleme dizilerinin tespitinde kullanılamaz. Bunun sebebi w sembolünün kullanımından kaynaklanır. w sembolü bilgi kaybı anlamına gelir.

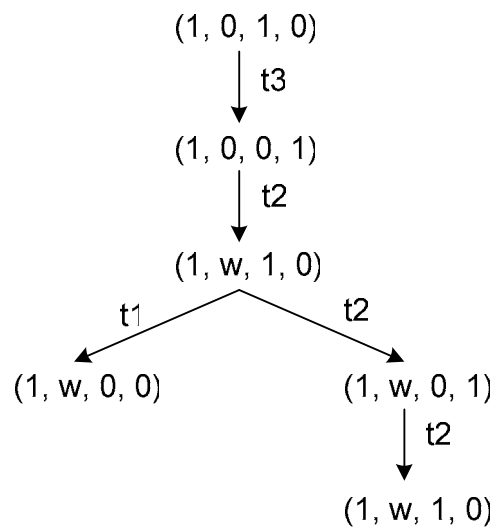
Şekil 4.22 ve 4.23'te görülen Petri ağlarının erişilebilirlik ağacı Şekil 4.24'te görülmektedir. Aynı erişilebilirlik ağacı benzer (ama farklı) iki Petri ağacını temsil ediyor. Buna karşın erişilebilirlik kümeleri aynı değildir. Şekil 4.22'de görülen Petri ağında p_2 yerinde her zaman tek sayıda jeton bulunur (t_1 ateşlenmediği sürece), buna karşın Şekil 4.23'teki Petri ağacında bu değer herhangi bir tamsayı olabilir. w sembolü bu tarz bilgilerin ayırt edilmesini engeller.



Şekil 4.22 Şekil 4.24'te Erişilebilirlik Ağacı Görülen Petri ağı

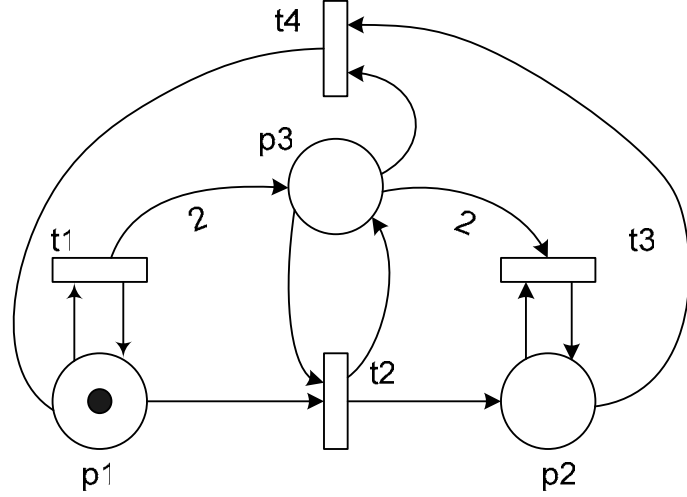


Şekil 4.23 Şekil 4.24'te Erişilebilirlik Ağacı Görülen Petri Ağı

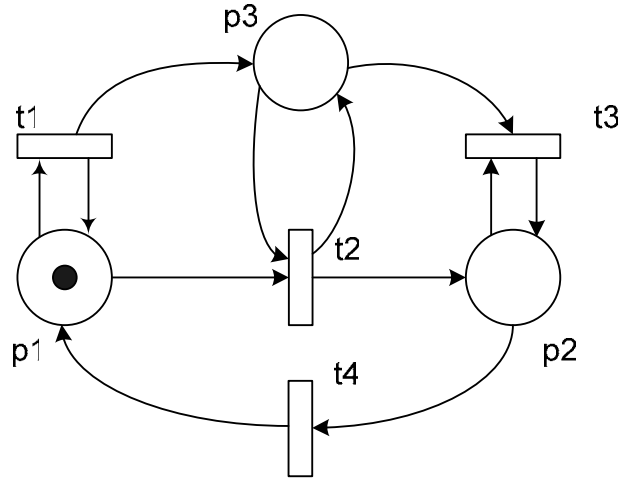


Şekil 4.24 Şekil 4.22 ve Şekil 4.23'teki Petri Ağlarının Erişilebilirlik Ağacı

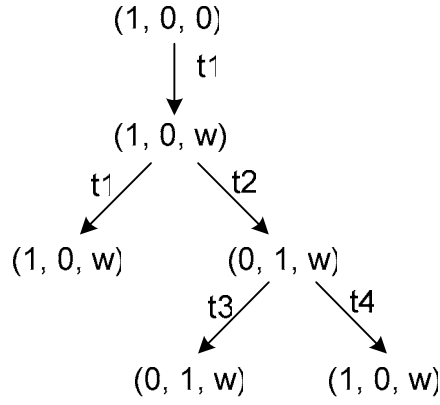
Benzer bir sorun canlılık probleminin çözümünde de gözlenir. Şekil 4.25 ve 4.26'daki Petri ağlarının erişilebilirlik ağacı Şekil 4.25 ve 4.26'daki Petri ağlarının erişilebilirlik ağacı Şekil 4.27'de görülmektedir. Şekil 4.25'teki ağ $t_1t_2t_3$ ateşleme dizisi sonucu kilitlenir, buna karşın Şekil 3.26'daki ağ canlılığını korur. Erişilebilirlik ağacı bunların ikisi arasındaki farkı da ayırt edemez.



Şekil 4.25 Muhtemel Kilitlenme Barındıran Petri Ağı



Şekil 4.26 Kilitlenme Barındırmayan Petri Ağı



Şekil 4.27 Şekil 4.25 ve Şekil 4.26'daki Petri Ağlarının Erişebilirlik Ağacı

Erişilebilirlik ağacı kullanılarak canlılık ve erişilebilirlik ile ilgili bazı analizler yapılabilirken ağaç yöntemi kullanılarak bu problemler tam olarak çözülemez. Bu amaçla değişik analiz yöntemlerine ihtiyaç duyulabilir.

4.2.4. Durum Denklemleri, Değişim-Olay Matrisleri

Bu bölümde lineer cebirden yararlanarak daha önce ele alınan soruların cevaplanmasına çalışılacaktır. Bir $C = (P, T, F, W)$ Petri ağı tanımlanmış olsun. Bu Petri ağındaki giriş ve çıkış fonksiyonlarını sırasıyla D^- ve D^+ matrisleri ile ifade edilecektir. Her matris m satır ve n sütundan oluşur. Satırlar geçişleri, sütunlar da yerleri ifade eder. Bu durumda,

$$D^-[j,i] = \#(p_i, I(t_j)), i \text{ yerinden } j \text{ geçişine gelen dalın ağırlığı}$$

$$D^+[j,i] = \#(p_i, O(t_j)), j \text{ geçişinden } i \text{ yerine giden dalın ağırlığı}$$

D^- geçişlere olan girişleri ifade ederken, D^+ çıkışları ifade eder.

$e[j]$ vektörü j 'inci elamanı 1, diğer elemanlı ise sıfır olan bir m -satır vektörü tanımlasın. Bu durumda M işaretlemesi altında izinli olan bir t_j geçişinin ateşlenmesi ile,

$$\delta(M, t_j) = M - e[j]D^- + e[j]D^+$$

$$\delta(M, t_j) = M + e[j](D^+ - D^-)$$

$$\delta(M, t_j) = M + e[j]D \quad (4.3)$$

4.3 denkleminde görülen $D = (D^+ - D^-)$ *değişim matrisi* olarak tanımlanır. Yukarıdaki işlemi şu şekilde tanımlanabilir;

$$M_k = M_{k-1} + e[j]D \quad (4.4)$$

4.4 denklemi incelenen Petri ağının *durum denklemini* ifade eder. Bir $\sigma = t_{j1}, t_{j2}, \dots, t_{jk}$ ateşleme dizisi göz önüne alındığında durum denklemleri genelleştirilebilir,

$$M_1 = M_0 + e[t_{j1}]D$$

$$M_2 = M_1 + e[t_{j2}]D = M_0 + (e[t_{j1}] + e[t_{j2}]) D$$

...

$$M_k = M_0 + (\sum_i e[t_{ji}]) D$$

$f(\sigma) = e[t_{j1}] + e[t_{j2}] + \dots + e[t_{jk}]$ vektörüne $t_{j1}, t_{j2}, \dots, t_{jk}$ ateşleme dizisinin *ateşleme vektörü* denir. $f(\sigma)$ 'daki i 'inci eleman t_{ji} geçişinin kaç kez ateşlendiğini gösterir. $f(\sigma)$ 'ya, σ dizisinin *Parikh vektörü* de denir.

Tanım 4.8: Olay Matrisi

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu Petri ağında n adet geçiş ve m adet yer bulunsun. $A = [a_{ij}]$ olay matrisi $m \times n$ boyutlu ve tüm elemanları tamsayılardan oluşan bir matristir. Bir elemanın değeri,

$$a_{ij} = a_{ij}^+ - a_{ij}^- \quad (4.5)$$

ile tanımlanır. 4.5 ifadesinde yer alan $a_{ij}^+ = w(i, j) = \#(p_j, O(t_i))$, i geçişinden j çıkış yerine olan dalın ağırlığı, $a_{ij}^- = w(j, i) = \#(p_j, I(t_i))$ 'de i geçişinden j giriş yerine olan dalın ağırlığıdır. Olay matrisi kavramı işaretli graflardan (Petri ağlarının bir alt kümesi) gelmektedir. Tanımından da anlaşıldığı gibi A olay matrisi değişim matrisine eşittir,

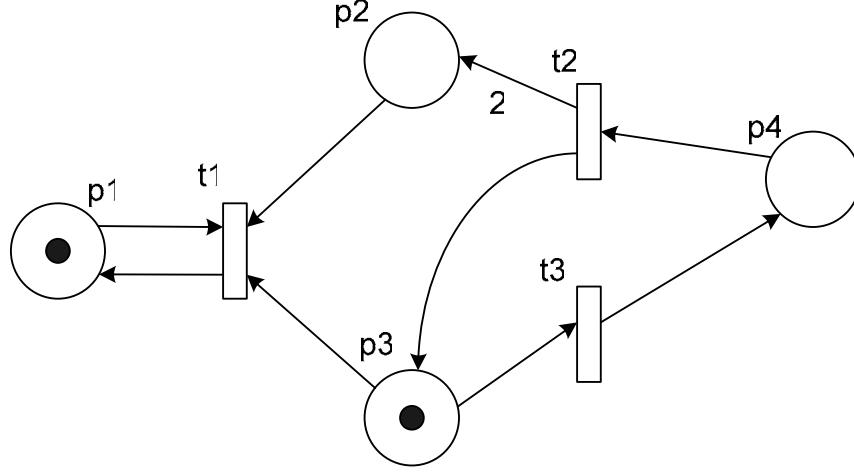
$$D = A \quad (4.6)$$

Olay matrisi soruların çözümünde kullanımı ayrıntılı olarak incelenecektir.

Matrisler kullanılarak erişilebilirlik problemi incelenebilir. M' işaretlemesine M 'den erişilebileceği varsayılın. Bu durumda öyle bir σ ateşleme dizisi vardır ki ağın durumunu M 'den M' durumuna getirsin. Bu durumda aşağıdaki durum denkleminin bir çözümü, x vardır.

$$M' = M + x.D \quad (4.7)$$

Örnek 4.1: Şekil 4.28’de görülen Petri ağı grafının değişim matrislerinin elde edilmesi, ilk işaretlemenin tanımlanması ve erişilebilecek işaretlemelerin incelenmesi.



Şekil 4.28 Durum Denklemeleri İncelenecek Petri Ağı Grafı

Şekildeki Petri ağına ilişkin D^- ve D^+ matrisleri,

$$D^- = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ ve } D^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ve değişim matrisi, $D = D^+ - D^-$

$$D = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Ağın ilk işaretlemesi $M_0 = [1 \ 0 \ 1 \ 0]$ 'dır ve bu durumda t_3 geçişi izinlidir. Bu ateşleme gerçekleşirse,

$$M_1 = [1 \ 0 \ 1 \ 0] + [0 \ 0 \ 1] \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$M_1 = [1 \ 0 \ 1 \ 0] + [0 \ 0 \ -1 \ 1]$$

$$M_1 = [1 \ 0 \ 0 \ 1]$$

Benzer şekilde $\sigma = t_3 t_2 t_3 t_2 t_1$ ateşleme dizisi $f(\sigma) = (1, 2, 2)$ ile ifade edilir. Bu durumda son işaretleme,

$$M' = [1 \ 0 \ 1 \ 0] + [1 \ 2 \ 2] \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$M' = [1 \ 0 \ 1 \ 0] + [0 \ 3 \ -1 \ 0]$$

$$M' = [1 \ 3 \ 0 \ 0]$$

Matrisleri kullanarak erişilebilirlik sorusunun çözülüp çözilemeyeceği araştırılsın; Yukarıda verilen Petri ağında $[1 \ 8 \ 0 \ 1]$ son işaretlemesine $[1 \ 0 \ 1 \ 0]$ ilk işaretlemesinden erişilebilir mi? Bu durumda aşağıdaki denklem takımı kurulur, burada bilinmeyen $\mathbf{x}$ ile ifade edilmiştir,

$$[1 \ 8 \ 0 \ 1] = [1 \ 0 \ 1 \ 0] + \mathbf{x} \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$[0 \ 8 \ 1 \ 1] = \mathbf{x} \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Yukarıda yer alan denklem takımının çözümünden $\mathbf{x} = [0 \ 4 \ 5]$ olarak bulunur. Bu da $\sigma = t_3 t_2 t_3 t_2 t_3 t_2 t_3$ ateşleme dizisine karşılık gelir.

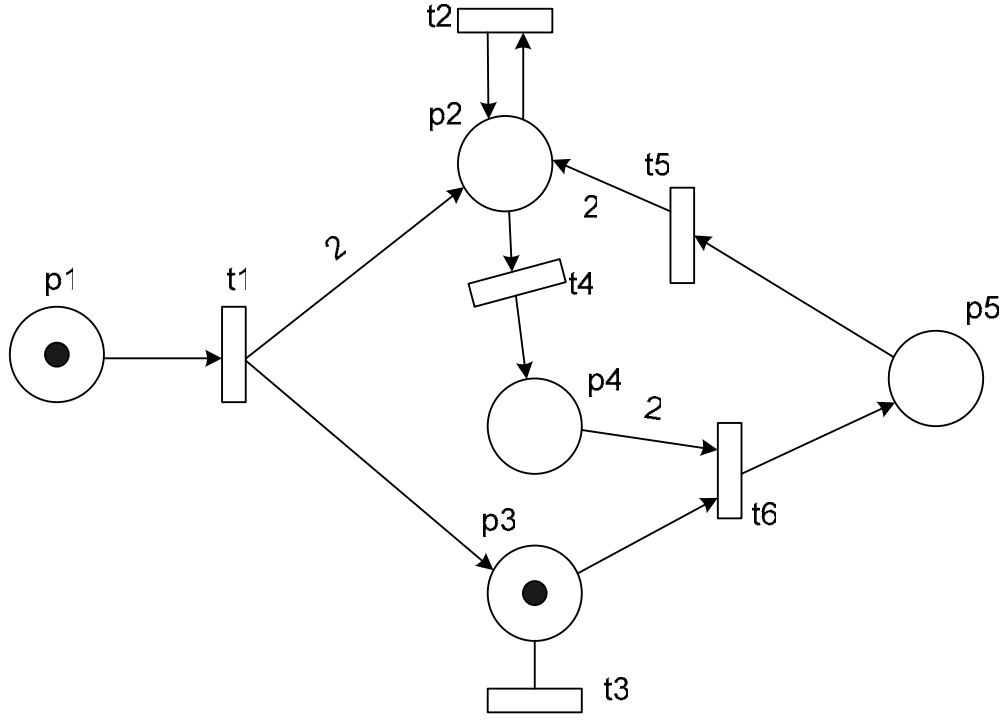
Yine incelenen Petri ağında $[1 \ 0 \ 1 \ 0]$ işaretlemesinden $[1 \ 7 \ 0 \ 1]$ işaretlemesine erişilemeyeceği gösterilebilir,

$$[1 \ 7 \ 0 \ 1] = [1 \ 0 \ 1 \ 0] + \mathbf{x} \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$[0 \ 7 \ 1 \ 1] = \mathbf{x} \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Yukarıdaki denklem takımının çözümü yoktur.

Durum denklemlerini kullanarak erişilebilirlik sorusu tam olarak cevaplanamaz. D değişiklik matrisi ağın durumunu doğru olarak yansıtmaz. Bunun sebebi ağdaki öz-çevrelere bilgilerinin ($D^+ - D^-$) işlemi ile kaybolmasıdır. Bu sorun, Petri ağına her bir öz-çevreye karşılık bir yer-geçiş çifti ekleyerek temiz hale getirilmesiyle aşılr.

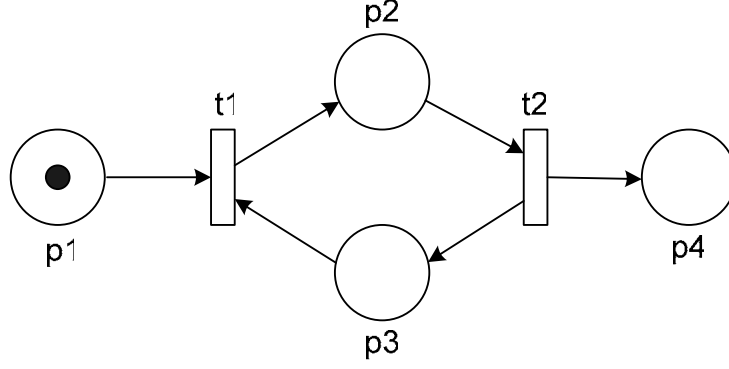


Şekil 4.29 Durum Denklemleri İncelenen Petri Ağı Grafı

Bir diğer sorun ateşleme dizilerinde sıralama bilgisinin kaybolmasıdır. Şekil 4.29’da görülen ve $(1, 0, 0, 0, 0)$ işaretlemesi ile işaretlenmiş ağın $(0, 0, 0, 0, 1)$ işaretlemesine getirilip getirilemeyeceği sorgulansın.

$$[0 \ 0 \ 0 \ 0 \ 1] = [1 \ 0 \ 0 \ 0 \ 0] + X \begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & -1 & 1 \end{bmatrix}$$

Yukarıdaki denklem takımının genel çözümü $X = (1, x_2, x_6-1, 2x_6, x_6-1, x_6)$ olarak bulunur. $x_2 = 1$ ve $x_6 = 1$ seçilirse $f(\sigma) = (1, 1, 0, 2, 0, 1)$ bulunur. Bu sonuçlar ateşlenen geçişlerin ve ateşleme sayılarının bilgisini verirken, ateşleme sırasının bilgisini vermez. Bu dizi $t_1t_2t_4t_4t_6$ olabileceği gibi $t_1t_4t_2t_4t_6$ da olabilir.



Şekil 4.30 Durum Denklemleri Çözümünü Yapılan Petri Ağı Grafi

Durum denklemleriyle uygulanan çözümlerde eksik kalan bir nokta da erişilebilirlik ağacı ile ilgilidir. Şekil 4.30'daki Petri ağında (1, 0, 0, 0) durumundan (0, 0, 0, 1) durumun gidilip gidilemeyeceğini araştırılsın.

$$[0 \ 0 \ 0 \ 1] = [1 \ 0 \ 0 \ 0] + X \begin{bmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix}$$

Burada çözüm $X = (1, 1)$ olarak bulunur. Muhtemel $t_1 t_2$ ve $t_2 t_1$ ateşleme dizilerinin her ikisi de mümkün değil, çünkü (1, 0, 0, 0) ateşlemesi altında izinli değildir. Dolayısıyla durum denklemleri bu şekilde erişilebilirliği göstermek amacıyla kullanılamaz.

Durum denklemleri birçok farklı analizde kullanılabilir. R. Johnsonbaugh ve T. Murata [9]'daki makalesinde durum denklemleri kullanılan ve işaretli graflara uygulanabilen bir erişilebilirlik testi yöntemi geliştirilmiştir. Bu amaçla bir işaretli grafin ilişkilerinden türetilen bir B_f temel matrisi tanımlanır.

$$\Delta M = M' - M_0, M' \text{ son işaretlemeyi, } M_0 \text{ ilk işaretlemeyi tanımlar}$$

$$B_f . \Delta M = 0 \quad (4.8)$$

4.8 koşulu sağlanıyorsa bu durumda işaretli grafta M' işaretlemesi M_0 işaretlemesinden erişilebilirdir.

Durum denklemlerine dayanan ve Petri ağlarının yapısı hakkında bilgileri açığa çıkaran P ve T değişmez analizleridir. Bir sonraki bölümde bu analizlerin üzerinde

durulacak, kullanım alanları tanıtılacak ve bunları bulmak amacıyla kullanılabilecek bir algoritma önerilecektir.

5. P DEĞİŞMEZLERİN VE T DEĞİŞMEZLERİN ANALİZİ

5.1. P – Değişmezlerin Özellikleri

Tanım 5.1: P – Değişmez Tanımı

Bir $C = (P, T, F, W)$ Petri ağı ve buna ait A $m \times n$ boyutlu olay-değişim matrisi tanımlı olsun. Bu sistemde bir $i: P_N \rightarrow Z$ tanımlanan bir i bağıntısı aşağıdaki homojen sistemin x çözümlerinden biri ise i 'ye P değişmezlik vektörü denir.

$$A \cdot x = 0 \quad (5.1)$$

Yerleri negatif sayılara götürmeyen değişmezlere negatif olmayan değişmezler denir. Bu temel P -değişmez tanımından bazı tanım ve sonuçlara varılabilir.

Tanım 5.2: P – Değişmez Özelliği

Bir $C = (P, T, F, W)$ Petri ağı tanımlanmış olsun. Bu durumda bir $i: P \rightarrow Z$ yer değişim vektörü aşağıdaki koşulu her $t \in T$ için sağlamaktadır,

$$\sum_{p \in \bullet t} i(p) \cdot w(p, t) = \sum_{p \in t \bullet} i(p) \cdot w(t, p) \quad (5.2)$$

Bunun anlamı bir t geçişi ateşlendikten sonra ağdaki jetonların sayısının i yer değişmezlik vektörüne bağlı olarak toplamalarının değişmeyeceğidir.

Sonuç: Yukarıdaki tanım sıradan Petri ağlarına uygulandığında bir t geçişinin ateşlenmesi i yer değişmezlik vektörüne göre aynı sayıda jeton eksiltip eklediği sonucu ortaya çıkar. Bu durumda $w(p, t)$ $w(t, p)$ 1 veya 0 değerlerini alabileceklerinden aşağıdaki bağıntı yazılabilir.

$$\sum_{p \in \bullet t} i(p) = \sum_{p \in t \bullet} i(p) \quad (5.3)$$

Teorem 5.1: Bir $C = (P, T, F, W)$ Petri ağı ve i yer değişmezi tanımlı olsun. Bir M işaretlemesine M_0 işaretlemesinden erişilebiliyorsa bu durumda [6];

$$\mathbf{i.M} = \mathbf{i.M}_0 \quad (5.4)$$

İspat: Durum denkleminde çözümü başlansın;

$$\mathbf{M} = \mathbf{M}_0 + \mathbf{A}^T \cdot \mathbf{u} \quad , \mathbf{u} \text{ ateşleme dizisi vektörünü gösterir}$$

Bu denklemde her iki tarafı sağdan bir $\mathbf{i}$ yer değişmez vektörü ile çarpılırsa

$$\mathbf{i.M} = \mathbf{i.M}_0 + \mathbf{i.A}^T \cdot \mathbf{u}$$

Tanım gereği $\mathbf{i.A}^T = \mathbf{0}$ olduğundan aşağıdaki sonuç elde edilir.

$$\mathbf{i.M} = \mathbf{i.M}_0 \quad (5.5)$$

Görüldüğü gibi 5.4 ve 5.5 ifadeleri birbirinin eşidir.

P değişmezler kullanılarak jetonların ağırlıklı toplamlarının farklı işaretlemeler altında aynı olduğu gözlenebilir. Teorem 5.2 bu konuyu kapsamaktadır.

Teorem 5.2: Bir $C = (P, T, F, W)$ Petri ağı ve $\mathbf{i}$ yer değişmezi tanımlı olsun. Bu ağda M, M_0 'dan erişilebilen bir işaretleme ise, [5]

$$\sum_{p \in P} i(p).M(p) = \sum_{p \in P} i(p).M_0(p) \quad (5.6)$$

İspat: R ile üstteki koşulu sağlayan işaretlemelerin kümesi tanımlansın. Açık olarak $M \in R$ ve $M_0 \in R$ olduğu görülmektedir. M işaretlemesi altında bir t geçişinin ateşlendiği varsayılınsın, bu durumda her p yeri için;

$$M'(p) = M(p) - w(p, t) + w(t, p) \quad (5.7)$$

olacaktır. Bu bağıntıda $p \in \bullet t$ ise $w(p, t)$ bir tam sayı değeri alacak, aksi durumda değeri 0 olacaktır. Benzer şekilde $p \in t \bullet$ ise $w(t, p)$ bir tam sayı değeri alacak, aksi durumda aldığı değer 0 olacaktır. Yukarıdaki bağıntıdan hareketle,

$$\sum_{p \in P} i(p).M'(p) = \sum_{p \in P} i(p).M(p) - \sum_{p \in \bullet t} i(p).w(p, t) + \sum_{p \in t \bullet} i(p).w(t, p) \quad (5.8)$$

5.8 eşitliğinde yer $\sum i(p).w(p, t)$ ve $\sum i(p).w(t, p)$ toplamaları birbirine eşit olduğundan,

$$\sum_{p \in P} i(p).M'(p) = \sum_{p \in P} i(p).M(p) \quad (5.9)$$

elde edilir. M' işaretlemesi de erişilebilir bir işaretleme olduğundan istenen sonuç iterasyonla elde edilir.

P değişmezler bu özellikleri ile bir Petri ağında tutarlı (conservative) yer ağırlık vektörünü belirlemede kullanılabilirler.

Teorem 5.3: Bir $C = (P, T, F, W)$ işaretli ve canlı Petri ağı tanımlı olsun. M_0 ağdaki ilk işaretlemeyi, $i:P \rightarrow Z$ bir bağıntı tanımlasın. Eğer bu ağdaki her erişilebilir M işaretlemesi aşağıdaki koşulu sağlarsa,

$$\sum_{p \in P} i(p).M(p) = \sum_{p \in P} i(p).M_0(p) \quad (5.10)$$

bu durumda i bir yer değişmezidir.

İspat: Bir t geçişi ağda tanımansın. t ölü olmadığından o ağda öyle iki erişilebilir M ve M' işaretlemeleri vardır ki $M' = \delta(M, t)$ olsun. Bu varsayımdan 5.11 eşitliği yazılabilir,

$$\sum_{p \in P} i(p).M(p) = \sum_{p \in P} i(p).M'(p) \quad (5.11)$$

Teorem 5.2'de kullanılan yöntem burada da kullanılabilir,

$$\sum_{p \in P} i(p).M'(p) = \sum_{p \in P} i(p).M(p) - \sum_{p \in \bullet t} i(p).w(p, t) + \sum_{p \in t \bullet} i(p).w(t, p) \quad (5.12)$$

5.12 eşitliğinde yer alan $\sum i(p).w(p, t)$ ve $\sum i(p).w(t, p)$ toplamaları eşit olduğundan istenen sonuç elde edilir.

$$\sum_{p \in P} i(p).M'(p) = \sum_{p \in P} i(p).M(p) \quad (5.13)$$

Elde edilen bu sonuca iterasyon uygulayarak $R(C, M_0)$ erişilebilirlik kümesinde yer alan tüm işaretlemeler için 5.14'teki toplamın aynı olduğu sonucuna varılır, bunlara $M' \in R(C, M_0)$ işaretlemesi de dahildir.

$$\sum_{p \in P} i(p).M'(p) \quad (5.14)$$

Teorem 5.4: Bir $C = (P, T, F, W)$ Petri ağı ve bu ağdaki bir p yeri tanımlı olsun. $i(p) \geq 1$ koşulunu sağlayan ve negatif olmayan bir yer değişmezlik vektörü varsa bu durumda ağdaki bir p yeri en fazla

$$\frac{1}{i(p)} \sum_{p' \in P} i(p').M(p') \quad (5.15)$$

jeton barındırabilecektir.

İspat: i negatif olmadığından $i(p') \geq 0$ her $p' \in P$ yeri için geçerlidir.

$$i(p).M(p) \leq \sum_{p' \in P} i(p').M(p') \quad (5.16)$$

olduğundan ve Teorem 5.2'deki sonucu kullanarak 5.17'deki eşitsizlik yazılabilir,

$$i(p).M(p) \leq \sum_{p' \in P} i(p').M_0(p') \quad (5.17)$$

Bu son eşitsizlik $i(p)$ 'ye bölünerek aranan sonuç 5.15'teki ifade bulunmuş olur.

Sonuç: Bir önceki teoremden yola çıkarak, işaretli bir Petri ağının sınırlı olabilmesi için öyle bir i yer değişmezine sahip olmalıdır ki tüm yerleri pozitif tam sayılara eşlesin (5.18).

$$i \geq 1 \quad (5.18)$$

Teorem 5.5: $C = (P, T, F, W)$ bir Petri ağını, M_0 ilk işaretlemeyi, i negatif olmayan ($i \geq 0$) bir yer değişmezlik vektörünü tanımlasın. i vektöründe $i(p) > 0$ koşulunu sağlayan yerlerin kümesi P_i olsun. Eğer her $p \in P_i$ için $M_0(p) = 0$ ise bu durumda ilk işaretlemede $(\bullet P_i \cup P_i \bullet)$ 'deki her geçiş ölüdür.

İspat: i bir yer değişmezi olduğundan erişilebilir hiçbir işaretleme P_i 'deki yerleri işaretleyemez. Her $t \in \bullet P_i$ geçişi de ölüdür aksi takdirde t 'nin ateşlenmesi sisteme jeton ekleyebilir.

Sonuç: İşaretli bir ağda, ilk işaretlemede hiçbir geçiş ölü değil ise bu durumda negatif olmayan her i yer değişmezi ayrı olmayan yerleri 0'a veya ilk işaretlemede jeton bulunduran en az bir yeri $i(p) \geq 1$ 'e götürür.

5.2. T – Değişmezlerin Özellikleri

Tanım 5.3: T – Değişmez Tanımı

Bir $C = (P, T, F, W)$ Petri ağı ve buna ait A $m \times n$ boyutlu olay matrisi tanımlı olsun. Bu sistemde bir $j: T_N \rightarrow Z$ tanımlanan bir j bağıntısı aşağıdaki homojen sistemin y çözümlerinden biri ise j 'ye T değişmezlik vektörü denir.

$$\mathbf{A}^T \cdot \mathbf{y} = \mathbf{0} \quad (5.19)$$

5.19 denklemin çözümü olan vektörlere geçiş değişmezleri denir.

Teorem 5.6: Bir $C = (P, T, F, W)$ Petri ağı ve bu ağdaki bir $\sigma: M \rightarrow M'$ ateşleme dizisi tanımlı olsun. P_σ Parikh vektörü $M = M'$ ise geçiş değişmezidir.

İspat: $P_\sigma = \mathbf{x}$ olsun. Durum denklemi bağıntısını yazarsak,

$$\mathbf{M}' = \mathbf{M} + \mathbf{A}^T \cdot \mathbf{x} \quad (5.20)$$

$\mathbf{x}$ bir geçiş değişmezi olduğundan tanım gereği $\mathbf{A}^T \cdot \mathbf{x} = \mathbf{0}$ 'dır. Dolayısıyla istenen sonuç ortaya çıkar,

$$\mathbf{M}' = \mathbf{M} \quad (5.21)$$

Teorem 5.7: Eğer sonlu elemanlı, sınırlı ve ilk işaretli bir Petri ağında sonsuz bir σ ateşleme dizisi varsa bu durumda bu ağın j geçiş değişmezi aşağıdaki özellikleri sağlar,

$j(t) = 0$, σ 'de yer almayan her geçiş için

$j(t) = 0$, σ 'de sonlu defa yer alan her geçiş için

$j(t) \geq 1$, σ 'de sonsuz defa yer alan her geçiş için

İspat: T_w , σ 'da sonsuz kez görülen geçişlerin oluşturduğu küme olsun. $\sigma = t_1 t_2 t_3 \dots$ olduğu varsayılın ve aşağıdaki işaretleme ilişkileri tanımlansın,

$$M_1 = \delta(M_0, t_1)$$

$$M_2 = \delta(M_1, t_2)$$

$$M_3 = \delta(M_2, t_3)$$

...

k_1 indeksi ile $t_{k_1+1}, t_{k_1+2}, t_{k_1+3} \dots$ T_w 'nın tüm elemanlarının görüldüğü ilk sıralama olsun.

Sonsuz bir k_1, k_2, k_3 indeks dizisi tanımlansın, öyle ki $m = 1, 2, 3, \dots$ için,

$$t_{k_m+1}, t_{k_m+2}, t_{k_m+3}, \dots, t_{k_{m+1}}$$

geçiş dizisi T_w 'daki her geçişi en az bir defa içersin.

İşaretli ağ, sonlu ve sınırlı olduğundan erişilebilecek tüm durumların kümesi sınırlıdır. Bu durumda işaretli ağın erişilebilirlik kümesinde $R(C, M_0)$ farklı i ve j indislerine sahip öyle iki işaretleme olacaktır ki bunlar birbirine eşit olsun, $M_{kj} = M_{ki}$. Bu durumda Teorem 5.6 gereği $t_{ki+1}, t_{ki+2}, \dots, t_{kj}$ geçiş dizisinin tanımladığı Parikh vektörü bir geçiş değişmezidir.

Teorem 5.8: Pozitif yer değişmezlik (P – değişmez) ve pozitif geçiş değişmezlik (T – değişmez) vektörlerine sahip her bağlı ağa *sıkı bağlı* bir ağdır.

İspat: C bağlantılı bir ağ tanımlasın. $\mathbf{i}$, C’de tanımlı ve $\mathbf{i} > \mathbf{0}$ olan bir yer değişmezi, $\mathbf{j}$ yine C’de tanımlı ve $\mathbf{j} > \mathbf{0}$ olan bir geçiş değişmezini tanımlasın. Sıkı bağlı (strongly connected) ağın tanımında C’deki her (u, v) dalına karşın, v düğümünden u düğümüne giden bir yol bulunur.

Durum 1: $u \in P$ ve $v \in T$ olsun,

$\mathbf{j}' : T \rightarrow \mathbf{N}$ bağıntısı aşağıdaki gibi tanımlansın;

$$\mathbf{j}' = \begin{cases} \mathbf{j}(t) & , v' \text{ den } t' \text{ ye giden bir yol varsa} \\ 0 & , \text{ diğer durumlarda} \end{cases}$$

Bu durumda $\mathbf{A}^T \cdot \mathbf{j}'$ yer vektörü göz önüne alındığında bir p yeri için bileşeni aşağıdaki bağıntıdan bulunur,

$$\sum_{u \in \bullet p} \mathbf{j}'(u) - \sum_{u \in p \bullet} \mathbf{j}'(u)$$

Herhangi bir p yeri için $\mathbf{A}^T \cdot \mathbf{j}' \leq \mathbf{0}$ olduğunu göstermek gerekir.

İlk başta p ’nin tüm öncü geçişlerinde ($t \in \bullet p$) $\mathbf{j}'(t) = \mathbf{0}$ olduğunu varsayalım. Tanım gereği her t geçişi için $\mathbf{j}'(t) \geq \mathbf{0}$ olduğundan 5.22’deki eşitsizlik yazılabilir.

$$0 = \sum_{t \in \bullet p} \mathbf{j}'(t) \leq \sum_{t \in p \bullet} \mathbf{j}'(t) \quad (5.22)$$

Herhangi bir $t \in \bullet p$ ’de geçişi için $\mathbf{j}'(t) = \mathbf{j}(t) > \mathbf{0}$ koşulu sağlanıyorsa bu durumda v ’den p ’ye giden bir yol vardır. Bir tane geçiş için bu yol varsa tüm $t \in p \bullet$ geçişlerine v ’den giden bir yol vardır. Bu yüzden her $t \in p \bullet$ için $\mathbf{j}'(t) = \mathbf{j}(t) > \mathbf{0}$ koşulu sağlanır. Bu durumda 5.23’teki ilişkiler yazılabilir.

$$0 < \sum_{t \in \bullet p} \mathbf{j}'(t) \leq \sum_{t \in \bullet p} \mathbf{j}(t) = \sum_{t \in p \bullet} \mathbf{j}(t) = \sum_{t \in p \bullet} \mathbf{j}'(t) \quad (5.23)$$

Her iki durum için 5.24'te yer alan ifadenin sağlandığı gösterildi.

$$\sum_{t \in \bullet p} j'(u) \leq \sum_{t \in p \bullet} j'(t) \quad (5.24)$$

5.24'te yer alan eşitsizlik $A^T \cdot j' \leq 0$ eşitsizliğinin doğruluğunu gösterir.

i bir yer değişmezi olduğundan $i \cdot A^T = 0$ olduğu bilinmektedir. Bu durumda $i \cdot A^T \cdot j' = 0$ olmalıdır. Tanım gereği $i > 0$ olduğundan ve $A^T \cdot j' \leq 0$ elde edildiğinden, $A^T \cdot j'$ vektörü negatif bileşen içermemelidir, bu yüzden tek çözüm $A^T \cdot j' = 0$ çözümüdür. j' bir geçiş değişmezi olduğundan aşağıdaki eşitsizlikler elde edilir,

$$\begin{aligned} \sum_{t \in \bullet u} j'(t) &= \sum_{t \in u \bullet} j'(t) & , j' \text{ bir geçiş değişmezidir} \\ &\geq j'(v) & , v \in u \bullet \\ &= j(v) & , j' \text{ nün tanımından} \\ &> 0 & , j \text{ pozitif bir geçiş değişmezlik vektörüdür.} \end{aligned}$$

Bu durumda $j'(t) > 0$ koşulunu sağlayan bir $t \in \bullet u$ geçişi vardır. j' nün tanımından v 'den t 'ye giden bir yol vardır. Bu yol u yerine kadar uzatılabilir.

Durum 2: $u \in T$ ve $v \in P$ olsun,

Bu durumda C' eşlenik ağı tanımlansın. Eşlenik ağda yerler ve geçişler yer değiştirmiştir. Elde edilen eşlenik ağ için yukarıdaki adımlar uygulanırsa istenen sonuç elde edilir.

5.3. P – Değişmez ve T – Değişmez Tanımları

Tanım 5.4: Yarı Pozitif P – Değişmezler

Bir $C = (P, T, F, W)$ Petri ağı ve buna ait A $m \times n$ boyutlu olay matrisi tanımlı olsun. $A \cdot x = 0$ denklem takımının çözümü olan ve her $p \in P$ için $i(p) \geq 0$ koşulu sağlanıyorsa bu P – değişmezlere yarı pozitif denir. Yarı pozitif P – değişmezlerin bir diğer adı da destek değişmezleridir.

Tanım 5.5: Yarı Pozitif T – Değişmezler

Bir $C = (P, T, F, W)$ Petri ağı ve buna ait A $m \times n$ boyutlu olay matrisi tanımlı olsun. $A^T \cdot y = 0$ denklem takımının çözümü olan ve her $t \in T$ için $j(t) \geq 0$ koşulu sağlanıyorsa

bu T – deęişmezler yarı pozitif denir. Yarı pozitif T – deęişmezlerin bir dięer adı da destek deęişmezleridir.

Tanım 5.6: Minimal Destek Deęişmezleri

Ele alınan bir i yer (j geçiř) destek deęişmezlik vektöründe $i_1(p) \leq i(p)$, her $p \in P$ için ($j_1(t) \leq j(t)$, her $t \in T$ için) saęlayan bir i_1 yer (j_1 geçiř) deęişmezlik vektörü yoksa bu durumda i (j) bir minimal destek deęişmezidir.

Tanım 5.7: Minimal Yer Destek Deęişmezleri Kümesi

$(n-r)$ adet minimal yer deęişmezinden oluřan kümeye yer minimal destek deęişmezleri kümesi denir. Bu kümenin elemanları birbirinden baęımsızdır. Dięer tüm yer deęişmezleri bu deęişmezlerin lineer kombinasyonundan türetilebilir.

Minimal yer destek deęişmezleri A 'nın sıfırı uzayını tanımlayan baz vektörlerinin span ettięi uzayda yer alan ama sadece pozitif deęerleri taşıyan vektörlerdir.

Tanım 5.8: Minimal Geçiř Destek Deęişmezleri Kümesi

$(m-r)$ adet minimal yer deęişmezinden oluřan kümeye minimal geçiř destek deęişmezleri kümesi denir. Bu kümenin elemanları birbirinden baęımsızdır. Dięer tüm geçiř deęişmezleri bu deęişmezlerin lineer kombinasyonundan türetilebilir.

Minimal geçiř destek deęişmezleri A 'nın sol sıfırı uzayını tanımlayan baz vektörlerinin span ettięi uzayda yer alan ama sadece pozitif deęerleri taşıyan vektörlerdir.

5.4. P ve T Deęişmezlerin Kullanım Alanları

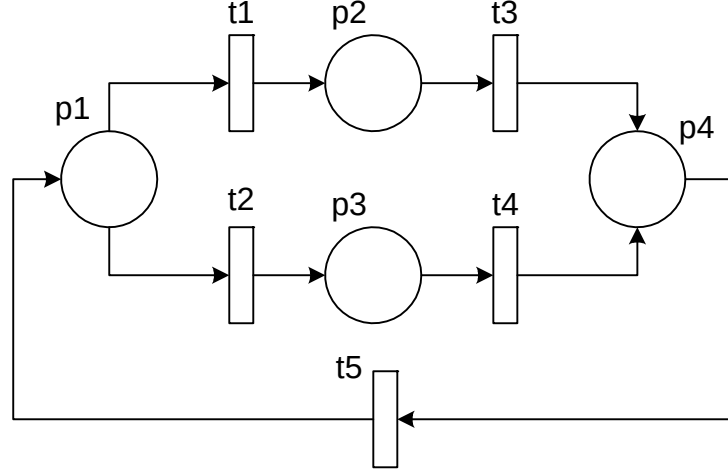
Bu bölümde P ve T deęişmezlerin deęişik amaçlarla kullanımları incelenecektir.

5.4.1. Petri Ağındaki Paralel ve Çevre Yapıların Sayısı

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. İncelenen Petri ağının sıkı baęlı olduęu varsayılmaktadır. Bu Petri ağına ait $m \times n$ boyutlu geçiř-yer olay matrisi A tanımlansın, $m = |T|$, $n = |P|$. A olay matrisinin rankı r olsun. Bu durumda Petri ağında $(m-r)$ adet paralel ve/veya çevre bulunacaktır [10]. Bulunan $(m-r)$ paralel ve/veya çevre A matrisinin sol sıfır uzayının spaninde yer alacaklardır. A 'nın sol sıfır

uzayı $A^T \cdot y = 0$ çözümünde elde edilen (m-r) adet T-değişmez ile span edilebilir. Bu (m-r) adet çözümde tüm elemanları 0'dan büyük olan yapılar çevreleri, hem negatif hem de pozitif değerlere sahip olanlar ise paralel yapıları vereceklerdir [11]. Örnek 5.1'de Petri ağına bu çözümlemeler incelenmiştir.

Örnek 5.1: Şekil 5.1'deki Petri ağına ait çevre ve/veya paralel yapıların sayısının bulunması.



Şekil 5.1 İncelenecek Petri Ağı Grafı

Yapılan çözümlerden sonra Petri ağına ilişkin T-değişmezlerin baz çözümleri $[1 \ 0 \ 1 \ 0 \ 1]$ ve $[-1 \ 1 \ -1 \ 1 \ 0]$ olarak bulunur. $[1 \ 0 \ 1 \ 0 \ 1]$ 'in desteği t_1 , t_3 ve t_5 'e karşı gelir. Bunlar gerçekten de bir çevre oluşturur. $[-1 \ 1 \ -1 \ 1 \ 0]$ 'in destek kümesi t_1 , t_2 , t_3 ve t_4 'te bir paralel oluşturur. İki çözüm toplanırsa $[0 \ 1 \ 0 \ 1 \ 1]$ elde edilir ki bu da t_2 , t_4 ve t_5 'in oluşturduğu çevreye karşı gelir. Bu durumda Petri ağına bir paralel yapı ve aynı çevre yapısını paylaşan iki çevre yolu bulunur. [10]

5.4.2. SSA'nın İşaret Geçiş Bileşenlerince Kapsanması

Canlı ve güvenilir bir serbest seçim Petri ağı İşaret Geçiş bileşenlerine ayrılabilir ve bu bileşenler bir araya geldiklerinde Petri ağını oluştururlar. Bu işaret geçiş alt ağları, A olay matrisinin sol sıfır uzayında ($A^T \cdot y = 0$) yer alan ve T-değişmez minimal desteklerinde $j(t) > 0$ olan geçişlerin bir arada kalmasıyla elde edilir. [3]

5.4.3. SSA Durum Makineleri Bileşenlerince Kapsanması

Canlı ve güvenilir bir serbest seçim Petri ağı Durum Makineleri bileşenlerine ayrılabilir ve bu bileşenler bir araya geldiklerinde Petri ağını oluştururlar. Bu durum

makinesi alt ağları, $\mathbf{A}$ olay matrisinin sıfır uzayında ($\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$) yer alan ve P-değişmez minimal desteklerinde $\mathbf{i}(p) > 0$ olan yerlerin bir arada kalmasıyla elde edilir. [3]

Pastor ve Cortadella [12] yaptıkları çalışmada minimal P değişmezlerine dayanan ve Petri ağlarını logaritmik sayıda değişken kullanarak kodlayan bir yöntem önermişlerdir. Yöntemde Petri ağları minimal P değişmezler kullanılarak durum makinesi bileşeni oluşturan alt ağlara ayrılıp bunlar üzerinde uygun kodlamalar uygulanıyor. İzlenen yöntemde amaç durum kodlamasında kullanılacak değişken sayısını azaltmak bu şekilde analizler ve kodlama aşamalarında daha az bellek ve işlemci gücüne ihtiyaç duymaktır.

5.4.4. PA Analizlerinin P-değişmezler Kullanılarak Kolaylaştırılması

Petri ağı modelleri eşzamanlı, asenkron, paralel özellikler sergileyebilir. Modeller hem çok büyük hem de analizleri çok karmaşık olabilir. Bu gibi durumlarda analizleri kolaylaştırmak amacıyla Petri ağının alt ağları incelenebilir. [13]'da gerçek Petri ağından anlamlı bir alt ağ tanımlanabilmesi amacıyla P değişmezlerden yararlanılarak oluşturulmuş bir geçiş matrisi tanımlanmıştır. Önerilen yöntem 1-sınırlı Petri ağlarında çalışmaktadır. [14]'te ise yine aynı geçiş matrisi Petri ağlarının davranışsal özelliklerini incelemek amacıyla kullanılmıştır. Bu amaçla P değişmezler ve geçiş matrisi Petri ağını temel eşzamanlı parçalara bölmek üzere algoritmalar önerilmiştir.

5.4.5. P-Değişmezler ile Haberleşme Sistemlerinde Belirsizlik Analizi

Haberleşme sistemlerinde belirsiz davranışlar tespit edilmeli ve önlenmelidir. Standart yöntemler belirsizliği tespit etmek için erişilebilirlik analizlerini kullanırlar. Nakamura [15] ve diğerleri tarafından önerilen yöntem ise Petri ağlarına dayanmaktadır. Bu yöntemde ilk önce incelenen haberleşme sistemlerinin eşdeğer Petri ağı karşılıkları üretilir. Oluşturulan Petri ağları üzerinde belirsizliklere sebep olan alt ağlar haberleşme sistemine ait servislerin tanımlarına bağlı olarak elde edilir. Yöntem son aşamada Petri ağı modeline ait P-değişmezlerini kullanarak M_0 ilk işaretlemesinden erişilemeyen M_k işaretlemelerini tespit eder. Geliştirilen yöntemde P-değişmezler erişilebilirlik konusunda gerek koşulu denetlemek amacıyla kullanılır. İlk işaretlemeden erişilemeyen işaretlemeler durumlar kümesinden çıkartılır ve

erişilebilirlik analizleri sadece sonuç kümesine uygulanır. Bu şekilde analiz zamanı oldukça azaltılır.

Uygulanan yöntemde amaç haberleşme hizmetlerinde belirsizlikleri azaltmaktır. Haberleşme sistemlerinde bir belirsizlik şu şekilde örneklenebilir. Hem Çağrı Bekletme (ÇB) hem de Çağrı Aktarma Değişkeni (ÇAD) hizmetleri veren bir haberleşme sistemi göz önüne alınsın. ÇB özelliği konuşma yapan kişinin başka bir çağrının gelmesi durumunda onu bekletme olanağı sağlar. ÇAD özelliği ise gelen çağrıları, ÇAD kullanıcısı tarafından önceden belirlenen telefon numarasına yönlendirilmesini sağlar. Bir A kullanıcısı hem ÇAD hem de ÇB hizmetlerinden birlikte yararlanıyor olsun. ÇAD hizmeti çağrıyı D kullanıcısına yönlendirecek şekilde kurulu iken, A kullanıcısı B kullanıcısı ile konuşuyor olsun. Bu durumda bir C kullanıcısı A kullanıcısını aradığında, bu çağrı A kullanıcısı tarafından mı alınacak yoksa D'ye mi aktarılacak. Haberleşme hizmetlerindeki bu tarz belirsizlikler Petri ağı modellerinde tespit edilip çözümler geliştirilebilir.

5.5. P ve T Değişmez Kümelerinin Lineer Cebirsel İncelemesi

P ve T değişmezler bir A geçiş-yer matrisinin sırasıyla $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ ve $\mathbf{A}^T \cdot \mathbf{y} = 0$ denklemlerinin çözümü olan **i** yer değişmezleri ve **j** geçiş değişmezleridir. $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ ve $\mathbf{A}^T \cdot \mathbf{y} = 0$ denklem sistemleri birer homojen sistemdir.

İncelemeler ilk önce $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ sistemi üzerinde yapılacaktır. **A** $m \times n$ boyutlu bir matristir. Burada *n* adet bilinmeyen ve *m* adet denklem vardır. Bu denklem sisteminde olası iki farklı çözüm vardır. [16]

1. Aşıkak çözüm: $x_1 = x_2 = \dots = x_n = 0$ aşıkak çözümü oluşturur.
2. Sıfırdan farklı sonsuz tane çözüm olabilir.

Bulunmak istenen, olması durumunda sonsuz çözümlerin tamamının bulunmasıdır. $[\mathbf{A} \mid \mathbf{0}]$ bölümlenmiş matrisine Gauss eliminasyonunu uygulanarak eşelon formu haline getirildiğinde sağdaki sütun her zaman 0 kalacaktır. Bu yöntemle aşıkak çözüm dışında kalan çözümler bulunamaz.

m satır ve *n* sütundan oluşan **A** matrisinin eşelon formuna indirgenmesi sonucu satırlardan hiçbirinin tamamı sıfır değilse bu durumda bulunabilecek tek çözüm aşıkak çözümdür, $\mathbf{x} = \mathbf{0}$. Eşelon formundaki matriste satırlardan en az biri tamamen

sıfır olan elemanlardan oluşuyorsa bu durumda sıfırdan farklı olan sonsuz çözüm vardır. Bu sonsuz çözümlerin boyutu ise tamamı sıfır olan satırların sayısına bağlıdır. Eşelon formunda sıfır olmayan satırlar satır uzayını span eder.

Yukarıda yapılan analizler $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{0}$ sistemi üzerinde uygulandığında eşelon formunda elde edilen sıfır olmayan satırlar $\mathbf{A}$ 'nın sütun uzayını span ederler. Eşelon formunda sıfır olan satırların sayısı ise $\mathbf{A}$ 'nın sol sıfır uzayının boyutunu belirler. Satır uzayı, sütun uzayı, sıfır uzayı ve sol sıfır uzayı birbirini tamamlayan dört temel uzaydır.

Tanım 5.9: $\mathbf{A}$ 'nın Satır Uzayı, $\mathbf{R}(\mathbf{A}^T)$

$\mathbf{A}$ $m \times n$ boyutlu bir matrisi tanımlasın. $\mathbf{A}$ 'nın satırları tarafından span edilen ve $\mathbf{R}^n$ 'nin alt kümesi olan uzaya satır uzayı denir. $\mathbf{A}$ 'nın satır uzayı bağımsız satırların sayısı ve $\mathbf{A}$ matrisinin rankı olan r , aynı zamanda satır uzayının boyutunu tanımlar. [17]

Tanım 5.10: $\mathbf{A}$ 'nın Sütun Uzayı, $\mathbf{R}(\mathbf{A})$

$\mathbf{A}$ $m \times n$ boyutlu bir matrisi tanımlasın. $\mathbf{A}$ 'nın sütunları tarafından span edilen ve $\mathbf{R}^m$ 'nin alt kümesi olan uzaya sütun uzayı denir. $\mathbf{A}$ 'nın sütun uzayı bağımsız sütunların sayısı ve $\mathbf{A}$ matrisinin rankı olan r , aynı zamanda sütun uzayının boyutunu tanımlar. [17]

Tanım 5.11: $\mathbf{A}$ 'nın Sıfır Uzayı, $\mathbf{N}(\mathbf{A})$

$\mathbf{A}$ $m \times n$ boyutlu bir matrisi tanımlasın, $m = |\mathbf{T}|$ ve $n = |\mathbf{P}|$ 'dir. $\mathbf{A}$ 'nın sıfır uzayı $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ denklem takımının çözümü ve $\mathbf{R}^n$ 'nin alt kümesidir. Lineer Cebirin Temel Teoreminden $\mathbf{A}$ 'nın sıfır uzayı $(n-r)$ boyutlu olarak bulunur, r $\mathbf{A}$ matrisinin rankıdır. Bu $(n-r)$ adet vektör $\mathbf{A}$ 'nın sıfır uzayının baz vektörleridir ve $\mathbf{R}^n$ boyutlu uzayı span ederler.

Tanım 5.12: $\mathbf{A}$ 'nın Sol Sıfır Uzayı, $\mathbf{N}(\mathbf{A}^T)$

$\mathbf{A}$ $m \times n$ boyutlu bir matrisi tanımlasın, $m = |\mathbf{T}|$ ve $n = |\mathbf{P}|$ 'dir. $\mathbf{A}$ 'nın sol sıfır uzayı $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{0}$ denklem takımının çözümü ve $\mathbf{N}(\mathbf{A}^T)$ ve $\mathbf{R}^m$ 'in alt kümesidir. Lineer Cebirin Temel Teoreminden $\mathbf{A}$ 'nın sol sıfır uzayının boyutu $(m-r)$ olarak bulunur. Yani $\mathbf{A}$ 'nın sol sıfır uzayının $(m-r)$ adet lineer bağımsız çözümü vardır. Bu $(m-r)$ adet

vektör A 'nın sol sıfır uzayının baz vektörleridir ve $\mathbf{R}^m$ boyutlu uzayı span ederler. Burada r A matrisinin rankıdır.

Tanım 5.13: Satır ve Sütun Rankı

A $m \times n$ boyutlu bir matrisi tanımlasın. A 'nın satır uzayının ve sütun uzayının boyutu birbirine eşittir. Bu boyut A matrisinin rankı r 'dir.

$$\text{Rank}(A) = r = \text{Satır uzayı boyutu} = \text{Sütun uzayı boyutu}$$

Teorem 5.9: Lineer Cebirin Temel Teoremi (Kısım 1)

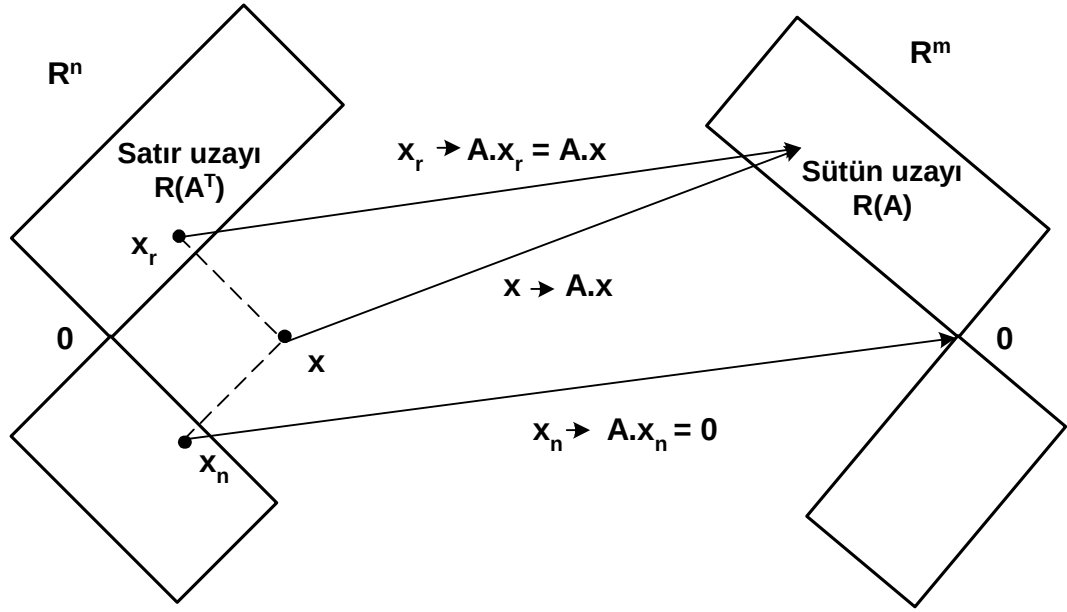
1. $R(A)$: A 'nın sütun uzayını tanımlar ve boyutu r 'dir.
2. $N(A)$: A 'nın sıfır uzayını tanımlar ve boyutu $(n-r)$ 'dir.
3. $R(A^T)$: A 'nın satır uzayını tanımlar ve boyutu r 'dir.
4. $N(A^T)$: A 'nın sol sıfır uzayını tanımlar ve boyutu $(m-r)$ 'dir. [17]

Teorem 5.10: Lineer Cebirin Temel Teoremi (Kısım 2)

Sıfır uzayı $\mathbf{R}^n$ 'de satır uzayının dik tamamlayıcıdır.

Sol sıfır uzayı $\mathbf{R}^m$ 'de sütun uzayının dik tamamlayıcıdır. [17]

Bu teoremin anlamı satır uzayında yer alan her vektörün sıfır uzayındaki herhangi bir vektöre dik olduğudur. Sütun uzayında bulunan her vektör sol sıfır uzayındaki herhangi bir vektöre diktir. Bu ilişkiler doğrudan homojen ve heterojen sistemlerin çözümleri ile ilgili ipuçlarını içinde barındırır. $A \cdot x = b$ sisteminin çözümünün olması demek b 'nin sütun uzayında olmasını gerektirir, dolaylı olarak bakıldığında b , A 'nın sol sıfır uzayına dik olmalıdır. Bu teoremin grafiksel ifadesi Şekil 5.2'de görülmektedir.



Şekil 5.2 Vektör Uzayları Arasındaki İlişki

$A.x_n = 0$ sisteminde, x_n doğrudan A 'nın sıfır uzayından çıkıp $\mathbb{R}^m$ uzayında sıfır değerine götüren bir dönüşümdür. Bu değerler aynı zamanda aranan değişmezleri de tanımlamaktadır. İzlenecek değişmezleri bulma yöntemi bu sonuçtan yararlanmaktadır.

Değişmezleri bulmak amacıyla kullanılan algoritma incelenmeden önce algoritmada kullanılan bazı işlemlerin tanımları ve bunlarla ilgili teoremler ele alınmalıdır.

Teorem 5.11:

Basit satır işlemleri bir matrisin satır uzayını değiştirmez. [18]

İspat:

A matrisinin satır vektörlerini $r_1, r_2, \dots, r_m$ olarak tanımlansın. B matrisi, A matrisine satır işlemleri uygulanarak elde edilen matris olduğu varsayılınsın. Burada B 'nin satır uzayındaki her satırın aynı zamanda A 'nın da satır uzayında olduğunun gösterilmesi gerekir. Doğal olarak bu ifadenin tersi de doğrudur. Sonuçta A ve B 'nin aynı satır uzayına sahip olduğu sonucuna varılmalıdır.

İspat yapılırken tüm satır işlemleri göz önüne alınmalıdır;

1. Yapılan işlem satır değişimi ise bu durumda $\mathbf{A}$ ve $\mathbf{B}$ 'nin satırları ve dolayısıyla satır uzayları birbirine eşittir.

2. Yapılan işlem bir satırın bir sabitle çarpımı yada bir satırı bir satırla çarpıp başka bir satıra eklenmesi işlemi ise bu durumda $\mathbf{B}$ 'nin elemanları olan $r_1', r_2', \dots, r_m'$ satırları, $\mathbf{A}$ matrisinin satırları olan $r_1, r_2, \dots, r_m$ 'in lineer birleşimidir. Bir vektör uzayı toplama işlemi ve bir sabitle çarpım işlemine göre kapalı olduğundan olası tüm $r_1', r_2', \dots, r_m'$ 'ler $\mathbf{A}$ 'nın satır uzayında yer alırlar. Dolayısıyla $\mathbf{B}$ 'nin satırları her zaman $\mathbf{A}$ 'nın satır uzayında yer alırlar.

Tüm işlemler sonucunda elde edilen $\mathbf{B}$ matrisinin satır uzayındaki her vektör $\mathbf{A}$ 'nın da satır uzayında yer alır.

Teorem 5.12:

$\mathbf{A}$ 'ya satır işlemleri uygulanmasıyla elde edilen üst üçgen formundaki $\mathbf{N}$ matrisinin satır uzayları eşdeğerdir.

İspat:

Teorem 5.12, Teorem 5.11'in bir sonucudur. $\mathbf{N}$ matrisi üst üçgen formuna indirgenmiş matrisi ifade etmektedir. $\mathbf{N}$ matrisi, $\mathbf{A}$ matrisine sürekli olarak satır işlemleri uygulanması sonucu elde edilmektedir. Satır işlemleri Teorem 5.11 gereği satır uzaylarını değiştirmedüğinden $\mathbf{N}$ matrisinin satır uzayı $\mathbf{A}$ matrisinin satır uzayına eşdeğer olmalıdır.

Sonuç: $\mathbf{A}$ matrisinin rankı $\text{Rank}(\mathbf{A}) = \mathbf{r}$ ise bu durumda aynı satır uzayına sahip $\mathbf{N}$ matrisinin rankı da $\mathbf{r}$ 'dir.

Matrislere uygulanan satır işlemleri $n \times n$ boyutlu $\mathbf{E}_i$ elemanter işlem matrisleri ile tanımlansın. $\mathbf{E}_1$ matrisi $n \times n$ boyutlu birim matristir. $\mathbf{Q}$ matrisi $n \times n$ boyutlu matristir. $\mathbf{Q}, \mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_k$ matrislerinin çarpımından oluşan sonuç matrisidir, $\mathbf{Q} = \mathbf{E}_k \dots \mathbf{E}_2 \mathbf{E}_1$. $\mathbf{A}$ $m \times n$ boyutlu bir matrisi tanımlar. $\mathbf{A}$ matrisinin sağdan $\mathbf{Q}^T$ matrisiyle çarpılması sonucu elde edilen olan sonuç matrisi $\mathbf{N}$ 'dir. $\mathbf{N}$ matrisi $n \times m$ boyutludur.

Teorem 5.13:

$\mathbf{A}$ $m \times n$ boyutlu olmak üzere $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ homojen sisteminin sıfırdan farklı çözümleri,

$$\mathbf{A} \cdot \mathbf{Q}^T = \mathbf{N} \quad (5.25)$$

koşulunu sağlayan $\mathbf{Q}^T$ matrisinin son $(\mathbf{n}-\mathbf{r})$ tane sütunudur.

İspat:

$\mathbf{A}$ matrisinin transpozu alınıp bunun üzerine satır işlemleri uygulansın. Teorem 5.11 gereği uygulanan satır işlemleri $\mathbf{A}^T$ matrisin satır uzayını değiştirmez. $\mathbf{A}^T$ 'nin satır uzayı ise $\mathbf{A}$ 'nın sütun uzayına karşı gelir. Satır işlemlerini ifade eden $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_k$ matrisleri soldan $\mathbf{A}^T$ matrisi ile çarpılsın. Sonuçta elde edilen $\mathbf{N}^T$ matrisinde sondaki $(\mathbf{n}-\mathbf{r})$ satırın tamamı sıfır olan satırlardır. Teorem 5.11 gereği $\mathbf{A}^T$ matrisi ve $\mathbf{N}^T$ matrislerinin satır uzayları birbirinin eşdeğeridir. $(\mathbf{n}-\mathbf{r})$ ifadesindeki $\mathbf{r}$ $\mathbf{A}^T$ 'nin ve aynı zamanda $\mathbf{N}^T$ 'nin rankına eşittir.

$$\begin{aligned} \mathbf{E}_1 \cdot \mathbf{A}^T &= \mathbf{N}^T_1 \\ \mathbf{E}_2 \cdot \mathbf{E}_1 \cdot \mathbf{A}^T &= \mathbf{N}^T_2 \\ \mathbf{E}_3 \mathbf{E}_2 \cdot \mathbf{E}_1 \cdot \mathbf{A}^T &= \mathbf{N}^T_3 \\ &\dots \\ \mathbf{E}_k \dots \mathbf{E}_3 \mathbf{E}_2 \cdot \mathbf{E}_1 \cdot \mathbf{A}^T &= \mathbf{N}^T \\ \mathbf{Q} \cdot \mathbf{A}^T &= \mathbf{N}^T \end{aligned} \quad (5.26)$$

Elde edilen bu eşitliğin transpozu alındığında 5.27'deki ifade elde edilir.

$$\mathbf{A} \cdot \mathbf{Q}^T = \mathbf{N} \quad (5.27)$$

Bu sonuca bakıldığında $\mathbf{N}$ matrisinin son $(\mathbf{n}-\mathbf{r})$ sütunu tamamı 0 olan sütunlardır. Teorem 5.10'da yer alan bağıntılardan, sıfır uzayına karşı gelen $\mathbf{x}_n$ değişmezlerini $\mathbf{A}$ matrisini sıfıra götürdüğü göz önüne alınsın.

$$\mathbf{x}_n \rightarrow \mathbf{A} \cdot \mathbf{x}_n = \mathbf{0}$$

Bu durumda $\mathbf{Q}^T$ matrisinin son $(\mathbf{n}-\mathbf{r})$ adet sütunu $\mathbf{A}$ 'nın satır uzayını $\mathbf{0}$ 'a götürürler. Dolayısıyla $\mathbf{Q}^T$ 'nin son $(\mathbf{n}-\mathbf{r})$ sütunlarından her biri bir değişmez vektördür. Bundan sonra bu sütunların bağımsız olup olmadığı araştırılmalıdır. Bu amaçla $\mathbf{E}_1$ nxn birim matrisine sırayla $\mathbf{E}_2, \mathbf{E}_3, \dots, \mathbf{E}_k$ satır değişim matrisleri uygulansın,

$$\mathbf{E}_2 \cdot \mathbf{E}_1$$

$$\mathbf{E}_3 \cdot \mathbf{E}_2 \cdot \mathbf{E}_1$$

...

$$\mathbf{E}_k \dots \mathbf{E}_3 \mathbf{E}_2 \cdot \mathbf{E}_1$$

Sonunda elde edilen bu ifade $\mathbf{Q}$ matrisinin kendisidir.

$$\mathbf{E}_k \dots \mathbf{E}_3 \mathbf{E}_2 \cdot \mathbf{E}_1 = \mathbf{Q}$$

$\mathbf{E}_1$ $n \times n$ birim matrisinin satırları birbirinden bağımsızdır. Dolayısıyla $\mathbf{E}_1$ matrisinin satırları n boyutlu bir satır uzayını span ederler. Bu matrise uygulanan satır işlemleri sonucunda elde edilen $\mathbf{Q}$ matrisi de Teorem 5.12 gereği aynı satır uzayını ve boyutunu span etmesi gerektiğinden tüm satırları birbirinden bağımsız olmalıdır. $\mathbf{Q}^T$ 'nin son $(n-r)$ sütunu $\mathbf{Q}$ 'nın son $(n-r)$ satırına karşı gelmektedir. Bu durumda $\mathbf{Q}^T$ 'nin tüm sütunları birbirinden bağımsızdır.

Sonuç itibariyle $\mathbf{Q}^T$ matrisinin son $(n-r)$ sütunu birbirinden bağımsız olup bunlar $\mathbf{A}$ 'nın satır uzayını $\mathbf{R}^m$ 'de yer alan sıfıra eşlerler. Bu $(n-r)$ sütun $\mathbf{A}$ 'nın sıfır uzayında yer alırlar. Bu sonuç aranan sonuçtur.

Lemma 5.1:

$\mathbf{Q}^T$ matrisinin son $(n-r)$ sütunu $\mathbf{A}$ matrisinin sıfır uzayını span ederler.

İspat:

$\mathbf{Q}^T$ matrisinin son $(n-r)$ sütunu birbirinden bağımsız olduğu bulunmuştu. Lineer Cebirin temel teoremlerinden $\mathbf{R}(\mathbf{A}^T) + \mathbf{N}(\mathbf{A}) = n$ olması gerekiyor, n $\mathbf{A}$ 'nın sütun sayısıdır. $\mathbf{A}$ 'nın satır uzayının rankı $\mathbf{R}(\mathbf{A}^T) = r$ olduğundan, $\mathbf{A}$ 'nın sıfır uzayının rankı $\mathbf{N}(\mathbf{A}) = n-r$ bulunur. $\mathbf{Q}^T$ matrisinde de $(n-r)$ adet bağımsız sütun olduğundan ve $\mathbf{A}$ 'yı sıfıra eşlediklerinden bunlar $\mathbf{A}$ 'nın sıfır uzayı için baz oluştururlar.

Matrislere uygulanan satır işlemleri $m \times m$ boyutlu $\mathbf{F}_i$ elemanter işlem matrisleri ile tanımlansın. $\mathbf{F}_1$ matrisi $m \times m$ boyutlu birim matristir. $\mathbf{P}$ matrisi $m \times m$ boyutlu matristir. $\mathbf{P}, \mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_k$ matrislerinin çarpımından oluşan sonuç matrisidir, $\mathbf{P} = \mathbf{F}_k \dots \mathbf{F}_2 \mathbf{F}_1$.

Teorem 5.14:

$\mathbf{A}$ $m \times n$ boyutlu olmak üzere $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{0}$ homojen sisteminin sıfırdan farklı çözümleri,

$$\mathbf{A}^T \cdot \mathbf{P}^T = \mathbf{N}^T \quad (5.28)$$

koşulunu sağlayan $\mathbf{P}^T$ matrisinin son $(\mathbf{m}-\mathbf{r})$ tane sütunudur.

İspat:

Teorem 5.14'ün ispatı teorem 5.13'ün ispatına paraleldir. İlkinde $\mathbf{A}^T$ matrisine uygulanan satır işlemleri burada doğrudan $\mathbf{A}$ matrisine uygulanmaktadır.

$$\mathbf{F}_1 \cdot \mathbf{A} = \mathbf{N}_1$$

$$\mathbf{F}_2 \cdot \mathbf{F}_1 \cdot \mathbf{A} = \mathbf{N}_2$$

$$\mathbf{F}_3 \mathbf{F}_2 \cdot \mathbf{F}_1 \cdot \mathbf{A} = \mathbf{N}_3$$

...

$$\mathbf{F}_k \dots \mathbf{F}_3 \cdot \mathbf{F}_2 \cdot \mathbf{F}_1 \cdot \mathbf{A} = \mathbf{N}$$

$$\mathbf{P} \cdot \mathbf{A} = \mathbf{N} \quad (5.29)$$

Teorem 5.12 gereği $\mathbf{A}$ matrisinin satır uzayı ve üst üçgen matris formundaki $\mathbf{N}$ matrisinin satır uzayı birbirinin eşdeğeridir. $\mathbf{N}$ matrisinin son $(\mathbf{m}-\mathbf{r})$ satırı tamamı 0 olan elemanlardan oluşur. $\mathbf{A}$ ve $\mathbf{N}$ matrislerinin rankı birbirine eşittir ve bu rank r ile ifade edilmiştir. Bir önceki eşitliğin transpozu alındığında

$$\mathbf{A}^T \cdot \mathbf{P}^T = \mathbf{N}^T \quad (5.30)$$

ifadesi elde edilir. Bu sonuca bakıldığında $\mathbf{N}^T$ matrisinin son $(\mathbf{m}-\mathbf{r})$ sütunu tamamı 0 olan sütunlardır. Teorem 5.10'de yer alan bağıntılardan, sol sıfır uzayına yer alan $\mathbf{x}_n$ değişmezlerini $\mathbf{A}^T$ matrisini sıfıra götürdüğü göz önüne alınsın.

$$\mathbf{x}_n \rightarrow \mathbf{A}^T \cdot \mathbf{x}_n = \mathbf{0}$$

Bu durumda $\mathbf{P}^T$ matrisinin son $(\mathbf{m}-\mathbf{r})$ adet sütunu $\mathbf{A}$ 'nın sütun uzayını $\mathbf{0}$ 'a götürürler. Dolayısıyla $\mathbf{P}^T$ matrisinin son $(\mathbf{m}-\mathbf{r})$ sütunlarından her biri bir değişmez vektörüdür. Bundan sonra bu sütunların bağımsız olup olmadığı araştırılmalıdır. Bu amaçla $\mathbf{F}_1$ $m \times m$ birim matrisine sırayla $\mathbf{F}_2, \mathbf{F}_3, \dots, \mathbf{F}_k$ satır değişim matrisleri uygulansın,

$$\mathbf{F}_2 \cdot \mathbf{F}_1$$

$$\mathbf{F}_3 \mathbf{F}_2 \cdot \mathbf{F}_1$$

...

$$\mathbf{F}_k \dots \mathbf{F}_3 \mathbf{F}_2 \mathbf{F}_1$$

Sonunda elde edilen bu ifade $\mathbf{P}$ matrisinin kendisidir.

$$\mathbf{F}_k \dots \mathbf{F}_3 \mathbf{F}_2 \mathbf{F}_1 = \mathbf{P}$$

$\mathbf{F}_1$ mxm birim matrisinin satırları birbirinden bağımsızdır. Dolayısıyla $\mathbf{F}_1$ matrisinin satırları m boyutlu bir satır uzayını span ederler. Bu matrise uygulanan satır işlemleri sonucunda elde edilen $\mathbf{P}$ matrisi de Teorem 5.12 gereği aynı satır uzayını ve boyutunu span etmesi gerektiğinden tüm satırlar birbirinden bağımsız olmalıdır. $\mathbf{P}^T$ 'nin son $(\mathbf{m}-\mathbf{r})$ sütunu $\mathbf{P}$ 'nin son $(\mathbf{m}-\mathbf{r})$ satırına karşı gelmektedir. Bu durumda $\mathbf{P}^T$ 'nin tüm sütunları birbirinden bağımsızdır.

Sonuç itibariyle $\mathbf{P}^T$ matrisinin son $(\mathbf{m}-\mathbf{r})$ sütunu birbirinden bağımsız olup bunlar $\mathbf{A}$ 'nın sütun uzayını $\mathbf{R}^n$ 'de yer alan sıfıra eşlerler. Bu $(\mathbf{m}-\mathbf{r})$ sütun $\mathbf{A}$ 'nın sol sıfır uzayında yer alır. Bu sonuç aranan sonuçtur.

Lemma 5.2:

$\mathbf{P}^T$ matrisinin son $(\mathbf{m}-\mathbf{r})$ sütunu $\mathbf{A}$ matrisinin sol sıfır uzayını span ederler.

İspat:

$\mathbf{P}^T$ matrisinin son $(\mathbf{m}-\mathbf{r})$ sütunu birbirinden bağımsız olduğu bulunmuştu. Lineer Cebirin temel teoremlerinden $\mathbf{R}(\mathbf{A}) + \mathbf{N}(\mathbf{A}^T) = m$ olması gerekiyor, m $\mathbf{A}$ 'nın satır sayısıdır. $\mathbf{A}$ 'nın sütun uzayının rankı $\mathbf{R}(\mathbf{A}) = r$ olduğundan, $\mathbf{A}$ 'nın sol sıfır uzayının rankı $\mathbf{N}(\mathbf{A}^T) = m-r$ olarak bulunur. $\mathbf{P}^T$ matrisinde de m-r adet bağımsız sütun olduğundan ve $\mathbf{A}$ 'yı sıfıra eşlediklerinden, bunlar $\mathbf{A}$ 'nın sol sıfır uzayı için baz oluştururlar.

Örnek 5.2:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \end{bmatrix}, \text{ 3x4 boyutlu matrisinin sıfır uzayının Teorem 5.13}$$

kullanılarak bulunması.

Bu çözümü gerçekleştirmek amacıyla ilk olarak $\mathbf{A}$ matrisinin transpozu alınır ve 4x4 boyutlu $\mathbf{E}_1$ birim matrisi tanımlanır.

$$\mathbf{A}^T = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & -1 \\ 1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \quad \mathbf{E}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ sonunda elde edilecek } \mathbf{Q} \text{ ve } \mathbf{N}^T$$

matrislerinin görünebilir olması amacıyla $\mathbf{E}_i$ satır işlemleri matrisleri hem $\mathbf{A}^T$ 'ye hem de $\mathbf{E}_1$ 'e uygulanacaktır. Benzer şekilde tüm elemanter işlem matrisleri uygulandıktan sonra $\mathbf{N}^T$ ve $\mathbf{Q}$ matrisleri aşağıdaki gibi elde edilir,

$$\mathbf{N}^T = \mathbf{Q} \cdot \mathbf{A}^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Eşitliğin transpozu alındığında $\mathbf{A} \cdot \mathbf{Q}^T = \mathbf{N}$ elde edilir,

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

Elde edilen $\mathbf{Q}^T$ matrisinin son iki sütunu $\mathbf{A}$ matrisinin satırlarını 0'e eşler. Bu durumda $\mathbf{v}_1 = [0 \ 1 \ 1 \ 0]^T$ ve $\mathbf{v}_2 = [1 \ 0 \ 0 \ 1]^T$ sütunları $\mathbf{A}$ 'nın sıfır uzayını span ederler.

Bir sonraki bölümde Teorem 5.13 ve Teorem 5.14'ü kullanan bir algoritma önerilmektedir.

5.6. P – Değişmezlik ve T – Değişmezlik Algoritmaları

5.6.1. Satır Eliminasyonu Yöntemi Algoritması

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu Petri ağının geçiş-yer olay matrisi $\mathbf{A}$ $m \times n$ matrisi, $|P| = n$, $|T| = m$ ve $\mathbf{I}_n$ 'da $n \times n$ boyutlu birim matrisi tanımlasın. Bu tanımlamalar ışığında P - değişmezleri bulmak amacıyla kullanılacak satır eliminasyonu yöntemi aşağıdaki adımlardan oluşur,

1. $\mathbf{A}$ matrisinin transpozu alınıp ön tarafına $n \times n$ boyutlu birim matris eklenir. Yeni oluşan matris $[\mathbf{I}_n; \mathbf{A}^T]$ şeklinde olur. Bu yeni matrise $\mathbf{B}$ ismi verilir.

$$\mathbf{B} \leftarrow [\mathbf{I}_n : \mathbf{A}^T]$$

2. Oluşan $\mathbf{B}$ matrisinin, $\mathbf{A}^T$ kısmı üzerinde satır eliminasyon işlemleri uygulanır. $\mathbf{A}^T$ üzerinde uygulanan satır işlemlerinin aynısı $\mathbf{I}_n$ 'ye de uygulanır. $\mathbf{A}^T$ kısmında uygulanan işlemler bittikten sonra $(n-r)$ adet satır sıfır olacaktır. Bu satırlara karşılık matrisin $\mathbf{I}_n$ kısmında kalan satırlar yer değişmezlerini verecektir. Eliminasyon sonucunda $\mathbf{A}^T$ matrisinde hiç tamamı sıfır olan satırlar olmaması durumunda tek çözüm aşıkardır.

T – değişmezleri elde etmek amacıyla yukarıdaki algoritma $\mathbf{A}'$ 'ya uygulanır. Sonunda elde edilen $\mathbf{B}$ matrisinde sağ tarafı sıfır olan satırların sol kısmı T değişmezleri verecektir.

P ve T değişmezlerin hesaplanmasında kullanılan ve en fazla bilinen algoritma Fourier-Motzkin algoritmasıdır.

5.6.2. Fourier-Motzkin Yöntemi

Bir $C = (P, T, F, W)$ Petri ağı tanımlı olsun. Bu Petri ağının geçiş-yer olay matrisi $\mathbf{A}$ $m \times n$ matrisi, $|P| = n$, $|T| = m$ ve $\mathbf{I}_k$ 'da k boyutlu birim matrisi tanımlasın. Bu tanımlamalar ışığında P - değişmezleri bulmak amacıyla kullanılacak Fourier – Motzkin yöntemi [4, 19] aşağıdaki adımlardan oluşur,

1. $\mathbf{C} = \mathbf{A}$,

$$\mathbf{D} = \mathbf{I}_n, n = |P| \text{ atamaları yapılır.}$$

2. $i = 1$ 'den m 'e kadar aşağıdaki adımları tekrarla, $m = |T|$,

2.1. $\mathbf{B} = [\mathbf{D} : \mathbf{C}]$ matrisine orijinal $\mathbf{B}$ matrisinin iki satırının pozitif lineer kombinasyonlarından oluşan ve yeni satırın $(n + i)$ 'inci sütun elemanını 0 yapan ürün satırını ekle.

2.2. $\mathbf{B}'$ den $(n + i)$ 'inci sütun elemanı sıfır olmayan satırları sil. Elde edilen matrisi yeni $\mathbf{B}$ matrisi olarak tanımla.

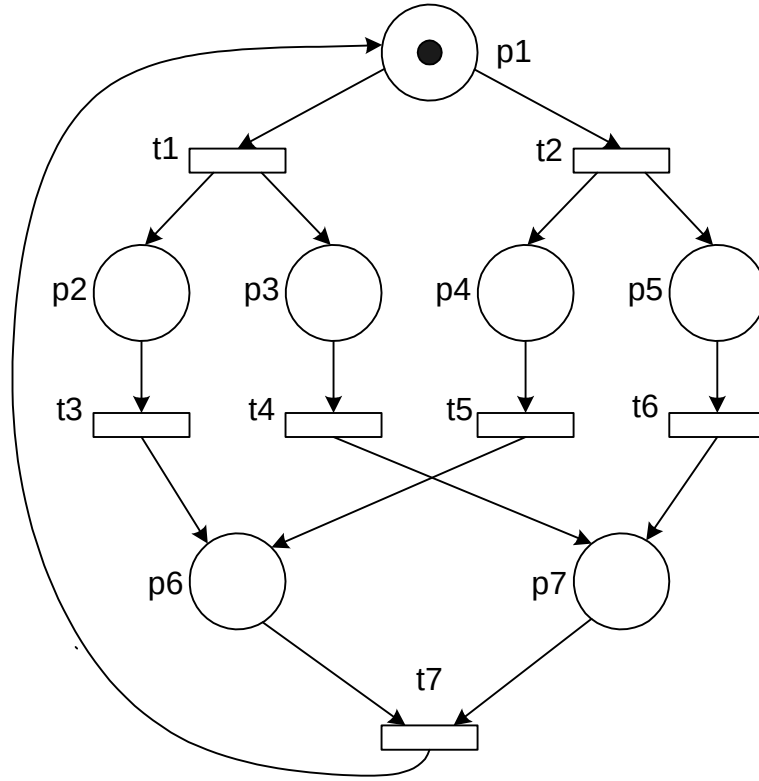
3. Geriye kalan $\mathbf{B}'$ den minimal olmayan satırları çıkart, kalan satırların elemanlarını da o satırdaki sıfır olmayan elemanların EBOB'una böl. Elde edilen son $\mathbf{B}$ matrisinin $\mathbf{D}$ kısmına karşı gelen satırlar P-değişmezleri verir.

Benzer algoritmayı A^T 'ye uygulanırsa bu durumda sonunda elde edilen B matrisinin D kısmına karşı gelen satırlar T -değişmezleri verecektir.

Daha önce tanımlanan algoritmalarından farklı olarak bu algoritma sadece pozitif P ve T değişmez vektörlerini bulacaktır. P ve T değişmez vektörlerinde sıfırdan küçük elemanların olması durumunda bunlar bulunamayacaktır. Bu algoritma minimal destek değişmezlerini bulmak amacıyla kullanılır. Yalnız her durumda tüm minimal çözümleri bulamayabilir.

Fourier – Motzkin yönteminin iki handikabı bulunmaktadır. Bunlardan ilki minimal değişmezler var olmalarına karşın algoritmada elde edilen durumların eksponensiyel olarak artmasından dolayı bellek aşımına sebep olabilir. İkinci handicap noktası ise belli sayıda değişmez elde edildikten sonra bile elde edilenlerin minimal değişmezlerin tamamını veya bir kısmını kapsamama olasılığıdır. [18]'te Fourier – Motzkin algoritmasının bu handikaplarını aşmak için yöntemle eklentiler önerilmiştir.

Örnek 5.3. Şekil 5.3'te görülen Petri ağına ait P değişmezlerin, Satır Eliminasyonu Algoritması ve Fourier-Motzkin Yöntemi kullanılarak elde edilmesi.



Şekil 5.3 İncelenecek Petri Ağı

Şekil 3'teki Petri ağının P-değişmezlerini bulmak amacıyla ilk önce Satır Eliminasyonu Algoritması uygulanacaktır. Daha sonra da Fourier-Motzkin algoritması uygulanacak ve bulunan sonuçlar karşılaştırılacaktır. İncelenen Petri ağının olay matrisi aşağıdaki gibi bulunur,

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

Satır Eliminasyonu Algoritması:

1. **B** matrisin elde edilmesi.

$$\mathbf{B} \leftarrow [\mathbf{I}_n; \mathbf{A}^T]$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}$$

2. Oluşan **B** matrisinin, $\mathbf{A}^T$ kısmı üzerinde satır eliminasyon işlemleri uygulanır. $\mathbf{A}^T$ üzerinde uygulanan satır işlemlerinin aynısı $\mathbf{I}_n$ 'ye de uygulanır. Satır eliminasyonu işlemleri sona erdikten sonra elde edilen **B** matrisinin aşağıda görülmektedir.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Elde edilen son matriste 6. ve 7. satır P yer değişmezlerini vermektedirler. Sol kısımda kalan $[1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0]$ ve $[1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1]$ vektörleri yer değişmezlerini vermektedirler. Bu iki vektör A'nın sol sıfır uzayının baz vektörleridir. Aynı zamanda minimal destek değişmezlerini de tanımlarlar.

Fourier-Motzkin Yöntemi:

$$1. C = A^T,$$

$D = I_n$, $n = |P|$ atamaları yapılır.

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

$$C = A^T = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}, D = I_n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$2. i = 1 \text{ 'den } m \text{ 'e kadar aşağıdaki adımları tekrarla, } m = |T|,$$

2.1. $B = [D:C]$ matrisine orijinal B matrisinin iki satırının pozitif lineer kombinasyonlarından oluşan ve yeni satırın $(n + i)$ 'inci sütun elemanını 0 yapan ürün satırını ekle.

2.2. B'den $(n + i)$ 'inci sütun elemanı sıfır olmayan satırları sil. Elde edilen matrisi yeni B matrisi olarak tanımla.

$$B = [D:C] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}$$

i) $i = 1$, için adım 2.1

1. satır ile 2. satırı topla ve B matrisine ekle

1. satır ile 3. satırı topla ve B matrisine ekle

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}$$

adım 2.2

Birinci sütunu sıfır olmayan satırları sil; Eski B matrisinde 1, 2 ve 3 nolu satırlar silinecek.

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}$$

Benzer şekilde $i = 2, 3, 4$ ve 5 için de adım 2.1 ve adım 2.2 uygulanır. Sonuçta aşağıda görülen B matrisi elde edilir.

$$B = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

B matrisinin sağ tarafındaki tüm satırlar sıfır olduğundan sol taraftaki satırlar pozitif P değişmezlerle karşı gelmektedir.

3. Geriye kalan **B**'den minimal olmayan satırları çıkart, kalan satırların elemanlarını da o satırdaki sıfır olmayan elemanların EBOB'una böl. Elde edilen son **B** matrisinin **D** kısmına karşı gelen satırlar P-değişmezleri verir.

Elde edilen üç P değişmezden ikisi bağımsızdır. $[2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$, P değişmezi diğer iki P değişmezin toplamından oluştuğundan bu çözümler arasından atılabilir. Geriye

kalan [1 0 1 0 1 0 1] ve [1 1 0 1 0 1 0], P değişmezleri aynı zamanda A 'nın minimal destek değişmezlerini de tanımlar.

5.6.3. Önerilmiş Diğer Değişmez Bulma Algoritmaları

Polinom Zamanlı Algoritma

Tanida [20] ve diğerlerinin makalesinde Petri ağı değişmezlerini üretmek amacıyla polinom zamanlı bir algoritma önerilmiştir. Önerilen algoritma lineer programlama tekniğine dayanmaktadır ve $O(m.n^2 + n.\tau(A))$ karmaşıklığına sahiptir. $\tau(A)$, lineer programlamada Karlmaker tarafından önerilen polinom zamanlı yöntemi ifade eder. Önerilen yöntem, olması durumunda pozitif bir baz bulmaya çalışmaktadır. Algoritmada $A.x = 0$ denkleminde ilk önce yarı-üretken bir tablo oluşturuluyor. Bu tablodan da gerçek değişmezler hesaplanıyor.

Savak ve Kapanlara Dayanan İki Algoritma

Yamauchi [21] ve diğerleri STFM (Siphon – Trap based Fourier – Motzkin) kodlu yeni bir algoritma önermişlerdir. Önerilen algoritma, Fourier-Motzkin ve minimal kovan bulma algoritmalarını (FDMS – Find Disjoint Minimal Siphons) birlikte kullanmaktadır. Algoritmada temel amaç bir minimal ayrık kovan tanımlayan $P' \subset P$ yerler kümesini veya bir minimal ayrık kapan tanımlayan $T' \subset T$ kümesini bulup bu kümenin üzerine Fourier-Motzkin algoritmasını uygulamaktır. Bu şekilde Petri ağı değişmezleri hesaplanırken aday sayısı azaltılmış olur.

Takano [22] ve diğerleri iki algoritma birden önerilmişlerdir. Algoritmalarından birincisi Fourier-Motzkin algoritmasını geliştirip performansını ve duyarlılığını artırmaktadır. İkinci algoritma ise [21]'de anlatılan STFM algoritmasını geliştirmektedir. Bu amaçla FDMS algoritmasına iyileştirmeler uygulanmakta, daha önce kullanılan Fourier-Motzkin algoritması yerine de iyileştirilmiş hali kullanılmaktadır.

Değişmez Bazı Oluşturan Algoritma

Bourjij [23] ve diğerleri, lineer cebirsel yöntemlere dayalı farklı bir algoritma önermişlerdir. Önerilen algoritma $A.x = 0$ homojen denklem takımını basit kurallar aracılığı ile basit bir forma dönüştürüyor. Yöntemde A matrisi dört parçaya bölünüp,

bu bölünen parçalar üzerinde işlemler gerçekleştiriliyor. A matrisi aşağıdaki gibi bir formda bölünüyor,

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \quad (5.30)$$

Yöntem gereği ilk adımda A mxn boyutlu matrisi 5.30'da görüldüğü gibi bir forma dönüştürölüp bölünüyor. A_1 bölmesi rxr boyutlu olup tüm rankı r'dir. Burada $\text{rank}(A) = \text{Rank}(A_1) = r$. Diğer bölmelerin boyutları buna bağlı olarak bulunabilir. İkinci adımda yöntemde 5.31'de tanımlanan bir M matrisini elde ediliyor.

$$M = \frac{-1}{\det(A_1)} (\text{Cof}(A_1))^T . A_2 \quad (5.31)$$

5.31'deki $\text{Cof}(A_1)$ A matrisinin kofaktörlerinden oluşur. A_1 matrisinin rankı tam olduğundan $\det(A_1) \neq 0$. M matrisine dayalı olarak pozitif değişmezlerin varlığına bağlı yorumlar yapılmakta ve olmaları durumunda bu değişmezler M'den hareketle elde edilmektedir. [23]

6. PADA PROGRAMI VE ÖZELLİKLERİ

Bu bölümde, yapılan çalışma kapsamında geliştirilen Petri Ağları Değişmez Analizi (PADA) programının yapısı, bu program kullanılarak değişmez analizlerinin yapılması anlatılacak ve program kullanılarak yapılmış çeşitli analizlerin zamanlarını gösteren çizelge anlatılacaktır.

6.1. PADA Programı Çalıştırılması

PADA programı komut satırından çalıştırılan bir programdır. Program çalıştırıldıktan sonra kullanıcıdan. **txt** uzantılı Petri ağı dosyasının isimi sorulmakta, daha sonra da kullanıcıdan çıktı dosyası ismi istenmektedir. Kullanıcı bunları sağladıktan sonra program P ve T değişmez analizlerini yürütmekte, sonuçları da çıktı dosyasına yazmaktadır. Sonuçlarda bulunan P ve T değişmezlerin yanı sıra, analizlerde kullanılan temel matrisler ve her analizin ortalama programlama zamanı da yer almaktadır.

6.2. PADA Programı Çalışma Ortamı

PADA programı Visual Studio .NET ortamında C# (C Sharp) programlama dili kullanılarak geliştirilmiştir. Program üzerinde Windows 9X, 2000, XP veya daha yeni bir işletim sistemi kurulu ortamlarda çalıştırılabilir. Windows 9X ve 2000 işletim sistemlerine programın hata vermeden çalıştırılabilmesi için .NET Framework 1.1 kurulu olmalıdır, ki bu yazılım web ortamından ücretsiz indirilebilmektedir.

6.3. PADA Programının Dosya Giriş Formatı

PADA programının veri girişi dosyaları Şekil 6.1’de verilen örnekteki formatta ve .txt uzantılı olarak tanımlanmalıdır. Bu formatta [P] ile yerlerin sayısı, [T] ile geçişlerin sayısı, [M] ile de satırları geçiş, sütunları da yerleri ifade eden değişim

matrisini tanımlamaktadır. Her satırda bulunan değerler birbirinden bir tab boşluk aralığı ile ayrılmış, her satır sonunda da yeni satır karakteri eklenmiştir. Yeni satır karakteri en son satırdan sonra da eklenmelidir.

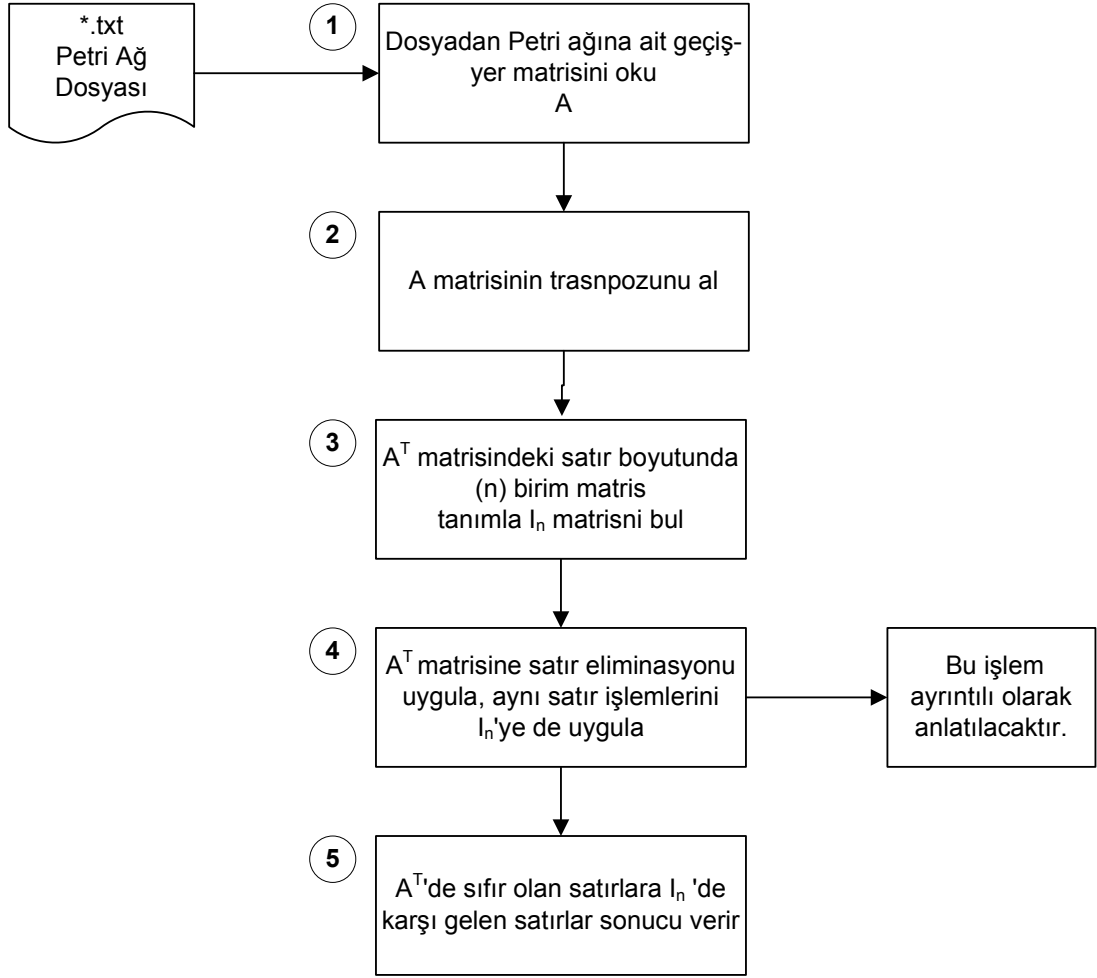
[P]						
7						
[T]						
7						
[M]						
-1	1	1	0	0	0	0
-1	0	0	1	1	0	0
0	-1	0	0	0	1	0
0	0	-1	0	0	0	1
0	0	0	-1	0	1	0
0	0	0	0	-1	0	1
1	0	0	0	0	-1	-1

Şekil 6.1 Örnek Petri Ağı Giriş Dosyası

6.4. PADA Programının Akışı

PADA programında iki temel döngü vardır. Bunlardan ilki şekil 6.2’de görülen P değişmezleri hesaplama döngüsüdür. 1. adımda döngü A $m \times n$ boyutlu matrisinin okunmasıyla başlamaktadır. 2 adımda A matrisinin transpozu alınmakta, 3. adımda ise $n \times n$ boyutlu I_n birim matrisi tanımlanmaktadır. 4. adımda A^T matrisine satır eliminasyonları uygulanmakta ve bu matrise uygulanan işlemlerin aynısı I_n birim matrisine de uygulanmaktadır. 5. adımda ise eliminasyon sonucunda indirgenmiş A matrisi elde edilmektedir. Bu döngü 5. bölümde yer alan Teorem 5.13’ü uygulayan yöntemdir. Son adımda elde edilen I_n birim matrisi yöntemdeki Q matrisine, indirgenmiş A^T matrisi de N^T ’teye karşı gelmektedir.

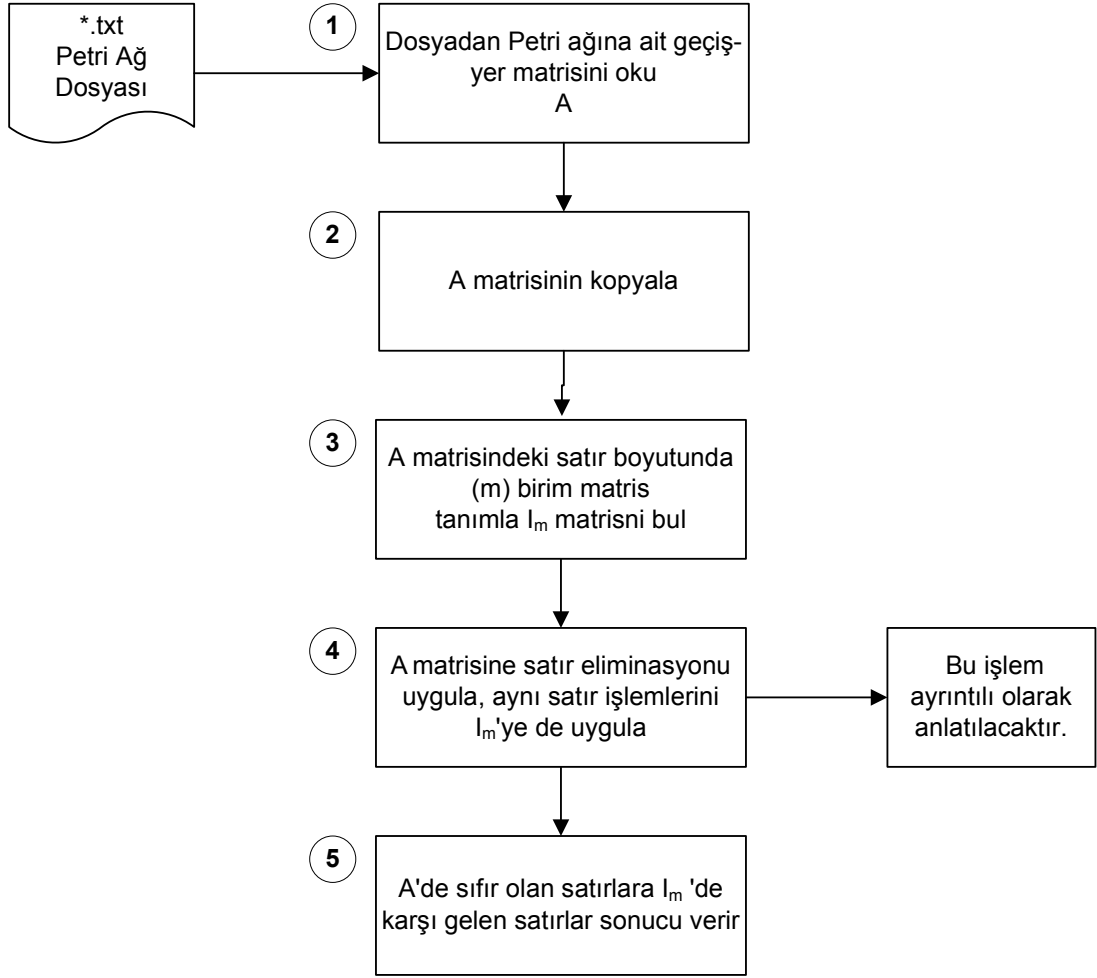
P – değişmez analizlerinin çalışma zamanı hesaplanırken 2 ve 5 arasındaki adımlar ilgili matris üzerinde 1000 kez uygulanıp ortalaması alınmaktadır.



Şekil 6.2 PADA Programında P Değişmez Analizi

PADA programındaki ikinci döngü Şekil 6.3’de görülen T değişmezleri hesaplama döngüsüdür. 1. adımda döngü A $m \times n$ boyutlu matrisinin okunmasıyla başlamaktadır. 2. adımda A matrisi kopyalanmakta, 3. adımda ise $m \times m$ boyutlu I_m birim matrisi tanımlanmaktadır. 4. adımda A matrisine satır eliminasyonları uygulanmakta ve bu matrise uygulanan işlemlerin aynısı I_m birim matrisine de uygulanmaktadır. 5. adımda ise eliminasyon sonucunda indirgenmiş A matrisi elde edilmektedir. Bu döngü 5. bölümde yer alan Teorem 5.14’ü uygulayan yöntemdir. Son adımda elde edilen I_m birim matrisi yöntemdeki P matrisine, indirgenmiş A matrisi de N ’e karşı gelmektedir.

T – değişmez analizlerinin çalışma zamanı hesaplanırken 2 ve 5 arasındaki adımlar ilgili matris üzerinde 1000 kez uygulanıp ortalaması alınmaktadır.



Şekil 6.3 PADA Programında T Değişmez Analizi

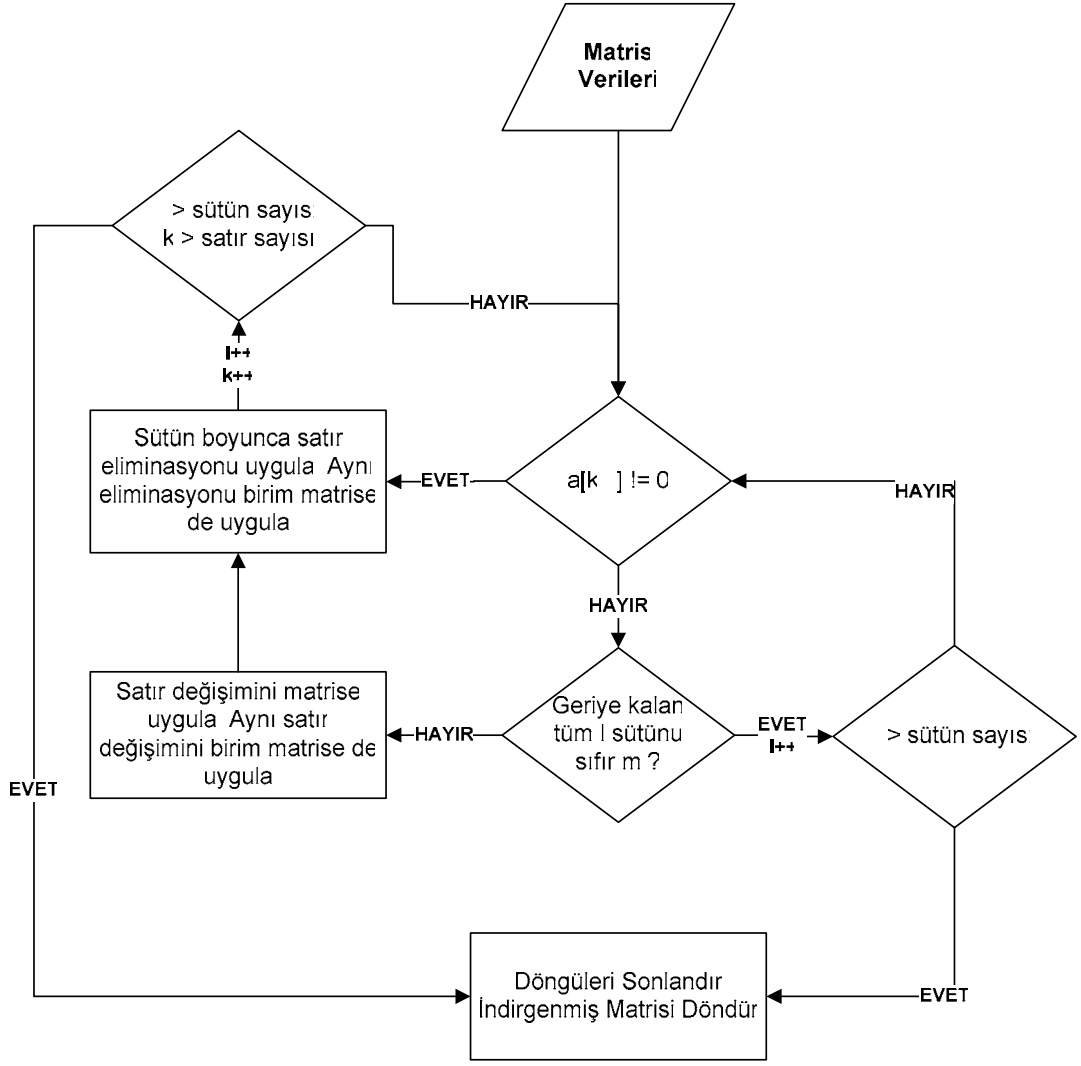
Her iki analizde de görülen 4. adımda satır eliminasyonlarının gerçekleştiği algoritma vardır. Bu algoritma bir sonraki bölümde daha yakından incelenecektir.

6.5. Satır Eliminasyonları Algoritması

Şekil 6.4'te program kodunda uygulanan satır eliminasyonu algoritması akış şeması görülmektedir. Bu algoritma program akışında dördüncü adıma karşı gelmektedir. Algoritma **P** değişmez ve **T** değişmez analizlerinde aynı şekilde çalışır. İkisinin tek farkı **P** değişmez analizinde $\mathbf{A}^T$ matrisine uygulanan algoritma adımları **T** değişmez analizinde **A** matrisine uygulanmasıdır.

Algoritmada amaç $m \times n$ boyutlu olan $\mathbf{A}^T$ matrisinde eliminasyonlar sonucunda üst üçgen matrisine dönüştürmek ve A matrisine uygulanan tüm işlemleri $\mathbf{I}_n$ birim matrisine de uygulamaktır.. Algoritma ilk olarak $\mathbf{a}[\mathbf{k}, \mathbf{I}]$ 'nin sıfıra eşit olup

olmadığını kontrol eder. Başlangıç noktası $k = 0$ ve $l = 0$ olan değerlerdir. Eğer $a[k, l]$ 'nin değeri sıfır ise algoritma bir sonraki adımda l sütununda diğer $a[i, l]$, $i = k, k+1, \dots, m$, elemanların sıfır olup olmadığını kontrol eder. Eğer sütunda tüm elemanlar sıfır ise bu durumda l 'in değeri bir artırılıp ($l++$) bir sonraki sütunda aynı işlemler uygulanır. Sütunda sıfır olmayan en az bir $a[i, l]$ elemanının bulunması durumunda $a[k, l]$ ve $a[i, l]$ satırlarının yeri değiştirilir. Bu yer değiştirme işlemi aynı şekilde birim matris üzerindeki $i[k, l]$ ve $i[i, l]$ satırlarına da uygulanır. Yer değiştirme sonucunda veya ilk kontroldeki $a[k, l]$ 'nin sıfırdan farklı bir değer alması durumunda l sütunu boyunca satır eliminasyonu uygulanır. Uygulanan satır eliminasyonu işlemlerinin aynısı I_n birim matrisine de uygulanır. Satır eliminasyonundan sonra hem k hem de l 'nin değeri bir artırılır ($k++$, $l++$). Algoritmada k ve l 'nin değer değişimlerinden sonra eşik değerlerine erişip erişmedikleri kontrol edilir. k 'nın eşik değeri m , l 'nin eşik değeri de n 'dir. Bu değişkenlerin değerleri üst sınıra ulaşmış olmaları durumunda algoritma sonlandırılır ve ilk matrisin indirgenmiş üst üçgen matrisi ile aynı satır işlemleri uygulanmış I_n matrisi geri döndürülür.



Şekil 6.4 Satır Eliminasyonu Uygulama Algoritması

Şekil 6.4'teki akış şeması görülen algoritmanın adımları Şekil 6.5'te ayrıntılı olarak ele alınmaktadır. Bu algorithmada kullanılan satır eliminasyonu algoritması Şekil 6.6'da görülmektedir. Satır eliminasyonu algoritması satır işlemleri yapılacak satırların sıradaki ilk elemanlarının işaretlerine bakıp, aynı işaretli olmaları durumunda ağırlıklı çıkarma işlemini, farklı işaretli olmaları durumunda ağırlıklı toplama işlemini yapmaktadır. Şekil 6.7'de görülen satır değiştirme algoritması, ilgili sütundaki ilk elemanı sıfır olmayan satırla ilk elemanı sıfır olan satırların yerlerini değiştirmektedir. Hem satır eliminasyonu algoritmasının hem de satır değişimi algoritmaları olay matrisine ve birim matrise paralel olarak uygulanmaktadır. Şekil 6.8'de ilgili sütun boyunca yine ilgili satırdan aşağıda kalan elemanların sıfır olup olmadıkları kontrol edilmektedir. Sıfır olması durumunda **doğru** durumu, sıfırdan farklı olması durumunda **yanlış** durumu geri döndürülmektedir.

Giriş: $m \times n$ boyutlu **a** matrisi
Çıkış: indirgenmiş **a** ve **id** matrisleri
başla:
 tanım **id** /* $m \times m$ boyutlu birim matris */
herbir ($k = 0 ; k < m$) **ve** ($l = 0 ; l < n$) **olduğu sürece yap**
 eğer ($a(k, l) \neq 0$) **ise**
 herbir ($i = k+1 ; i < m$) **için yap**
 SatırEliminasyonu(**a**, **id**)
 son
 $k++ , l++$
son
diğer (**SütunSıfırMı**(**a**) == **yanlış**) **ise**
 SıfırOlmayanSatırlaDeğiştir (**a**, **id**)
 herbir ($i = k+1 ; i < m$) **için yap**
 SatırEliminasyonu(**a**, **id**)
 son
 $k++ , l++$
son
diğer (**SütunSıfırMı**(**a**) == **doğru**) **ise**
 $l++$
son
son
bitir

Şekil 6.5 Satır Eliminasyonu Uygulama Algoritma Adımları

Fonksiyon **SatırEliminasyonu** (**a**, **id**)
başla:
 eğer ($a(k, l).id(k, l) < 0$) **ise**
 herbir ($j = 0 ; j < n$) **için yap**
 $a(j, i) \leftarrow a(j, i).a(k, l) + a(k, i).a(j, l)$
 $id(j, i) \leftarrow id(j, i).a(k, l) + id(k, i).a(j, l)$
 son
son
eğer ($a(k, l).id(k, l) > 0$) **ise**
 herbir ($j = 0 ; j < n$) **için yap**
 $a(j, i) \leftarrow a(j, i).a(k, l) - a(k, i).a(j, l)$
 $id(j, i) \leftarrow id(j, i).a(k, l) - id(k, i).a(j, l)$
 son
son
bitir

Şekil 6.6 Satır Eliminasyonu Fonksiyonu

Fonksiyon **SıfırOlmayanSatirlaDeğiştir** (a, id)

başla:

herbir (i = k+1; i < m) **için yap**

eğer (a(i, l) != 0)

a(k, l) $\leftrightarrow$ a(i, l)

id(k, l) $\leftrightarrow$ id(i, l)

dur

son

son

bitir

Şekil 6.7 Satır Değişimi Fonksiyonu

Fonksiyon **SütunSıfırMı** (a)

başla:

colSum = 0

colSumB = **yanlış**;

herbir (i = k; i < m) **için yap**

colSum $\leftarrow$ colSum + |a(i, l)|

son

eğer(colSum == 0) **ise**

colSumB = **doğru**;

son

döndür colSumB

bitir

Şekil 6.8 Sütunda Sıfır Kontrol Fonksiyonu

6.6. PADA Programı Analiz Sonuçları

Tablo 6.1’de çeşitli boyutlardaki Petri ağlarının P ve T değişmez analizlerinin hesaplama zamanları görülmektedir. Hesaplamalar Intel Pentium 4, 2.4 GHZ işlemcisi olan bir bilgisayar üzerinde yapılmıştır. Tablonun ilk sütununda analizleri yapılan **m**x**n** boyutlu olay matrisinin boyutları verilmiştir. A matrisindeki **m** değeri geçiş sayısını, **n** değeri de yer sayısını tanımlamaktadır. İkinci ve üçüncü sütunda, ilgili matrisin sırasıyla P ve T analiz zamanları verilmiştir. Verilen değerler mili saniye cinsindendir.

Tabloda kullanılan ve Petri ağlarının tanımlı olduğu giriş dosyaları, programın ürettiği sonuç dosyaları EK-C’de verilmiştir.

Tablo 6.1 PADA Programı Hesaplama Zamanları

A'nın boyutu	P analiz zamanı (ms)	T analiz zamanı (ms)
7x7	5,190 E-02	2,510 E-02
10x9	4,280 E-02	5,210 E-02
20x18	0,207	0,278
40x36	1,170	1,350
50x45	1,647	2,080
100x90	7,438	9,096

7. SONUÇLAR VE ÖNERİLER

Bu tezde, eş zamanlı sistemleri modelleme, tanımlama, analiz etme ve sentezlemede kullanılan bir araç olan Petri ağlarının yapısal özelliklerinden P ve T değişmezler incelenmiş ve bunları bulmak amacıyla bir algoritma önerilmiştir. Son bölümde önerilen bu algoritma C# programlama dili kullanılarak kodlanmış bir program geliştirilmiştir.

Program çalıştırıldığında P ve T değişmezler için birer baz elde edilmektedir. Bu bazlar kullanılarak diğer tüm P ve T değişmezler türetilebilir.

İleride önerilecek olan algoritmada bazı iyileştirmeler yapılarak pozitif minimal değişmezlerin bulunması amaçlanmaktadır.

Bir diğer adım, algoritma performansının mevcut olan diğer değişmez bulma algoritmaları ile karşılaştırılması olabilir.

Son aşamada daha geniş kapsamlı analizler yapan ve görsel olarak Petri ağlarının tanımlanabildiği bir programda, bu tezde önerilen algoritma bir analiz modülü olarak kullanılabilir.

KAYNAKLAR

- [1] **Peterson , James L.**, 1981. Petri Net Theory and The Modelling of Systemts, Prentice-Hall, Inc., Englewood Cliffs
- [2] **Murata, T.**, 1989. Petri Nets: Properties, Analysis and Applications, *Proceedings of The IEEE*, Vol. 77, No.4, 541-580
- [3] **Pastore, E.**, 1996. Structural methods for the synthesis of asynchronus circuits from signal transition graphs, *PhD Thesis*, Universitat Politecnica de Catalunya, Barcelona, Spain.
- [4] **Silva, M., Teruel, E., Colom, J. M.**, 1998, Linear Algebraic and Linear Programming Techniques for the Analysis of Place/Transition Net Systems, , in *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets*, Springer, 309-373
- [5] **Desel, J., Reising, W.**, 1998, Place or Transition Petri Nets, in *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets*, Springer, pp. 122-173
- [6] **Desel, J.**, 1998, Basic Linear Algebraic Techniques for Place/Trasition Nets, in *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets*, Springer, pp. 257-308
- [7] **Jeng, M-D., Peng, M-Y.**, 1997, Petri Net Liveness Analysis by Minimal Siphons, *Emerging Technologies and Factory Automation Proceedings, 1997. ETFA '97*, Los Angeles, CA USA, pp. 315 – 320
- [8] **Agerwala, T.**, 1979, Putting Petri Nets to Work, *IEEE Computer Magazine*, Vol.12, No.12, pp. 85-94
- [9] **Johnsonbough, R., Murata, T.**, 1982, Petri Nets and Marked Graphs – Mathematical Models of Concurrent Computing, *The American Monthly*, Vol. 89, Issue 8, 552-566
- [10] **Cai, Y., Sekiguchi, T., Tanaka, H., Hikichi, M., Maruyama, Y.**, 1993, A Method for Structural Analysis of Petri Net Models, *Proceedings of the IECON '93., International Conference*, Maui, USA, Vol. 1, 133 - 137

- [11] **Miyazawa, I., Tanaka, H., Sekiguchi, T.,** 1996, Classification of Solutions of Matrix Equation Related to Parallel Structure of a Petri Net, 1996 *IEEE Conference on Emerging Technologies and Factory Automation (EFTA)*, Kauai, USA, Vol. 2, 446 – 452
- [12] **Pastor, E., Cortadella, J.,** Efficient Encoding Schemes for Symbolic Analysis of Petri Nets, 1998, *Design, Automation and Test in Europe, 1998., Proceedings*, Paris, France, pp. 790-795
- [13] **Song, Y-J., Lee, J-K.,** 2000, Analysis of Petri net models using Transitive Matrix, 2000 *IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, USA, Vol. 4, 3122 - 3127
- [14] **Lee, J-K.,** 2002, Decomposition of Petri Nets Using The Transitive Matrix Based on P-invariant, 2002 *IEEE International Conference on Systems, Man and Cybernetics*, Hammamet, Tunisia, Vol.3
- [15] **Nakamura, M., Kakuda, Y., Kikuno, T.,** 1997, Analysing Non-determinism in Telecommunication Services Using P-Invariant of Petri-Net Model, *Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Kobe, Japan, Vol. 3, pp. 1253 – 1260
- [16] **D. Meyer, C.,** 2000, Homogeneous Systems, in *Matrix Analysis and Applied Linear Algebra*, pp. 57-61, SIAM, Philadelphia
- [17] **Strang, G.,** 1988. Linear Algebra and Its Applications, Harcourt College Publishing, Harcourt
- [18] **Anton, H., Rorres, C.,** 2000. General Vector Spaces, in *Elementary Linear Algebra*, Applications Version, John-Wiley, pp. 203-274
- [19] **Martinez, J., Silva, M.,** 1982. A Simple and Fast Algorithm to Obtain all Invariants of Generalized Petri Net, *Fachberichte Informatik*, Springer – Verlag, Berlin, Germany, Vol. 52, 301-310
- [20] **Tanida, T., Watanabe, T., Onaga, K.,** 1991, A Polynomial-Time Algorithm For Finding a Semi-Generator of Petri Net Invariants, 1991 *EEE International Symposium on Circuits and Systems*, Singapore, Singapore, Vol. 5, 2838 - 2841
- [21] **Yamauchi, M., Wakuda M., Taoka, S., Watanabe, T.,** 1999, A Fast and Space-Saving Algorithm for Computing Invariants of Petri Nets, 1999 *IEEE International Conference on Systems, Man, and Cybernetics*, Tokyo, Japan, Vol. 1, 866 - 871
- [22] **Takano, K., Taoka, S., Yamauchi, M., Watanabe, T.,** 2001, Two Efficient Methods For Computing Petri Net Invariants, 2001 *IEEE International Conference on Systems, Man, and Cybernetics*, Tucson, USA, Vol. 4, 2717 - 2722

- [23] **Bourjij, A., Boutayeb, M., Koenig, D., Cecchin, T.**, 1997, On Generating a Basis of Invariants in Petri Nets, *1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, Orlando, USA, Vol. 3, 2228 – 2233

EKLER

EK-A: Uygulamanın kaynak kodları (CD'ye kaydedilmiştir)

EK-B: Tezin dökümanlarına ait dosyalar (CD'ye kaydedilmiştir)

EK-C: Bölüm 6'da değişmezleri elde edilen Petri ağları (CD'ye kaydedilmiştir)

ÖZGEÇMİŞ

SİNAN ÇAYIR

1976 yılında Bulgaristan, Rusçuk'ta doğdu. İlk eğitimini Bulgaristan'da, orta eğitimini İstanbul'da, lise eğitimini de Adana Fen Lisesinde tamamladı. 1999 yılında İstanbul Teknik Üniversitesi'nden, Elektronik ve Haberleşme Mühendisi ünvanı aldı. 2000 yılı Mart ayından beri İstanbul Teknik Üniversitesi, Bilgi İşlem Daire Başkanlığında araştırma görevlisi olarak çalışmaktadır. 2002 yılında İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik mühendisliği yüksek lisans programına başlamıştır.