

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
(YÜKSEK LİSANS TEZİ)

MODAL LOJİK İLE KLASİK LOJİK ARASINDAKİ İLİŞKİ

Zafer ÖZDEMİR

Tez Danışmanı : Yrd. Doç. Dr. Tahsin Öner

Matematik Anabilim Dalı

Bilim Dalı Kodu : 403. 05. 01

Sunuş Tarihi : 26.08.2010

Bornova-İZMİR

2010

Zafer ÖZDEMİR tarafından Yüksek Lisans Tezi olarak sunulan “ Modal Lojik ile Klasik Lojik Arasındaki İlişki” başlıklı bu çalışma, Ege Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeli ile Ege Üniversitesi Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 26.08.2010 tarihinde yapılan tez savunma sınavında aday oybirliği ile başarılı bulunmuştur.

Jüri Üyeleri

İmza

Jüri Başkanı : Yrd. Doç. Dr. Tahsin Öner

.....

Raportör Üye : Prof. Dr. Mehmet Terziler

.....

Üye : Doç. Dr. Alev Fırat

.....

ÖZET**MODAL LOJİK İLE KLASİK LOJİK ARASINDAKİ İLİŞKİ**

ÖZDEMİR, Zafer

Yüksek Lisans Tezi, Matematik Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Tahsin ÖNER

Ağustos 2010, 45 sayfa

Modal lojik formülleri Kripke çatılar üzerinde ikinci mertebeden monadic özellikler ifade etmektedir. Pek çok durumda modal lojik formüllerine karşılık gelen birinci mertebeden lojik formülleri etkili algoritmalar yardımı ile hesaplanmaktadır. Konuyla ilgili ilk çalışma H. Sahlqvist' in "Correspondence and completeness in the first and second order semantics for modal logic" isimli makalesidir. Sahlqvist 1973 tarihli makalesinde Sahlqvist formülleri olarak adlandırılan modal lojik formüllerinin belirli bir sınıfını tanımlayarak, bu sınıfın çatılar üzerinde birinci mertebeden koşullar tanımladığını ve bu koşulların Sahlqvist tekniği yardımı ile modal lojik formüllerinden etkili bir biçimde hesaplanabileceğini kanıtlamıştır.

Bir modal lojik formülüne karşılık gelen birinci mertebeden lojik formülü her zaman bulunmayabilir. Bazı durumlarda bir modal lojik formülü ikinci mertebeden lojik formülüne karşılık gelebilir. Bu tip durumlarda Sahlqvist tekniği etkinliğini kaybetmektedir. Bir modal lojik formülüne karşılık gelen birinci ve ikinci mertebeden lojik formülünü hesaplamaya yarayan farklı algoritmalar ve teknikler geliştirilmiştir. Bu algoritmalar içinde öne çıkan iki çalışma bulunmaktadır. H. J. Ohlbach ve D. Gabbay tarafından geliştirilen, temeli constraint resolution ve tekniğine dayanan SCAN algoritması ve W. Conradie, V. Goranko ve D. Vakarelov tarafından geliştirilen, modal formüller üzerinde direkt olarak çalışan SQEMA algoritmasıdır.

Bu tez giriş bölümü dahil altı ana bölümden oluşmaktadır.

İkinci bölümde önermeler lojîğinin ve modal lojîğın temel kavramları tanımlanmıştır.

Üçüncü bölümde Sahlqvist formüller sınıfı tanıtılarak Sahlqvist Teoremi verilmiş ve Sahlqvist tekniğinin modal formüller üzerindeki etkinliğı açıklanmıştır.

Dördüncü bölümde SCAN algoritmasının temelini oluşturan resolution ve Skolemizasyon tekniğı ile ilgili temel tanımlar verilmiş, SCAN algoritmasının adımları ayrıntılı olarak şekilde incelenmiş ve algoritma örneklerle açıklanmıştır. Bölümün sonunda ise algoritmanın Sahlqvist formülleri üzerindeki tamliğı ispatlanmıştır.

Beşinci bölümde SQEMA algoritmasının işleyişinde kullanılan temel kavramlar tanımlarla verilmiş, algoritmanın adımları ayrıntılı olarak incelenmiş ve örneklerle açıklanmıştır; bölüm sonunda ise algoritmanın Sahlqvist formülleri üzerindeki tamliğı ispatlanmıştır.

Altıncı bölümde Sahlqvist tekniğı, SCAN ve SQEMA algoritmaları etkinliklerine göre karşılaştırılmıştır.

Anahtar sözcükler: Sahlqvist Formülleri, SCAN Algoritması, SQEMA Algoritması.

ABSTRACT**RELATIONSHIP BETWEEN MODAL LOGIC AND CLASSICAL
LOGIC**

ÖZDEMİR, Zafer

Master Thesis in Mathematic Department

Supervisor: Asst. Prof. Tahsin ÖNER

August 2010, 45 pages

Modal formulae express monadic second-order properties on Kripke frames. In many cases modal formulas correspond to first-order formulas computed by influential algorithms. The first research of this area was “Correspondence and completeness in the first and second order semantics for modal logic” which was written by H. Sahlqvist in 1973. He defined a class of modal formulas which are determined first order conditions on Kripke frames and those conditions can be effectively computed from the modal formulas.

Sometimes a first order equivalent of a modal formula doesn't exist. In some cases a modal formulae can be correspond to a second-order formulae. Under the circumstances Sahlqvist technique loses its effectiveness. Some algorithms and techniques have been developed for computing a first-order or a second-order equivalent of a modal formula. Several algorithms and techniques are introduced but two of them come forward in these algorithms. SCAN, based on constraint resolution technique and SQEMA, works directly on modal formulae.

In addition to Introduction Chapter, the thesis consists essentially of The thesis consists of six chapters.

Chapter two starts with basic definitions and notions of proposition logic and modal logic.

In the chapter three, class of Sahlqvist formulae and Sahlqvist theorem are introduced. Then the effectiveness of Sahlqvist technique is explained on modal formulae.

In the chapter four, definitions of resolution and Skolemization technique are introduced which are basis for SCAN algorithm. Then SCAN algorithm is described in great detail, algorithm is illustrated by examples. In the end of this chapter, the completeness of SCAN algorithm is proved with respect to class of Sahlqvist formulae.

In the chapter five, basic definitions and notions of SQEMA algorithm are introduced which are used process of this algorithm. The steps of SQEMA algorithm are examined in detail and are explained with examples. In the end of this chapter, the completeness of SQEMA algorithm is proved with respect to class of Sahlqvist algorithm.

In the chapter six, Sahlqvist technique, SCAN and SQEMA algorithms are compared with respect to their effectiveness.

Key words: Sahlqvist Formulae, SCAN Algorithm, SQEMA Algorithm.

TEŐEKKÜR

Bu alıŐma sűresince gerekli kaynakların saęlanmasında yardımcı olan, zellikle kıymetli grűŐlerinden yararlandıęım ve yakın ilgisini esirgemeyen sayın Prof. Dr. Mehmet Terziler ve sayın Yrd. Do. Dr. Tahsin ner'e teŐekkűrű bir bor bilirim.

İÇİNDEKİLER

| | <u>Sayfa</u> |
|---|--------------|
| ÖZET..... | II. |
| ABSTRACT..... | III. |
| TEŞEKKÜR..... | IV |
| 1. GİRİŞ..... | 1 |
| 2. ÖN BİLGİLER..... | 2 |
| 2.1. Önergeler Mantığı..... | 2 |
| 2.2. Modal Lojik..... | 3 |
| 2.3. K Lojigi..... | 5 |
| 3. SAHLQVİST FORMÜLLERİ ve TEKNİĞİ..... | 8 |
| 3.1. Ön Bilgiler..... | 8 |
| 3.2. Sahlqvist Formülleri ve Teoremi..... | 9 |
| 3.3. Örnekler..... | 12 |
| 4. SCAN ALGORİTMASI..... | 14 |
| 4.1. Ön Bilgiler..... | 14 |
| 4.2. SCAN Algoritması..... | 17 |
| 4.3. Örnekler..... | 19 |
| 4.4. SCAN Algoritmasının Sınırları..... | 22 |
| 4.5. Sahlqvist Formülleri için SCAN Algoritmasının Tamlığı..... | 23 |

İÇİNDEKİLER (devam)

| | <u>Sayfa</u> |
|---|--------------|
| 5. SQEMA ALGORİTMASI..... | 26 |
| 5.1 Ön Bilgiler..... | 26 |
| 5.2 SQEMA Algoritması..... | 29 |
| 5.3 Örnekler..... | 34 |
| 5.4 Sahlqvist Formülleri için SQEMA Algoritmasının Tamlığı..... | 40 |
| 6. SONUÇ..... | 43 |
| KAYNAKLAR DİZİNİ..... | 44 |

1. GİRİŞ

Modal lojik, çok basit bir ifadeyle, birinci mertebeden lojiğe bir ya da daha fazla operatör eklenerek oluşturulan lojik olarak tanımlanabilir. Birinci mertebeden lojiğin dili ifade gücünün yeterli olmaması nedeni ile kuvvetli matematiksel yapıların oluşturulmasına engel olur. Modal lojik birinci mertebeden lojikle karşılaştırıldığında sahip olduğu operatörler sayesinde daha etkili bir ifade gücüne sahiptir.

Modal lojik ve birinci mertebeden lojik arasındaki ilişkiyi inceleyen ve bilinen en iyi sentaktik yaklaşım Sahlqvist formüller sınıfıdır. Bu sınıfın birinci mertebeden karşılığı Sahlqvist tekniği kullanılarak hesaplanmaktadır. Her modal formül çatılar üzerinde ikinci mertebeden bir koşul tanımlar. Çoğu zaman ikinci mertebeden niceleyicilerin elenmesi ile birinci mertebeden lojiğe karşılık gelen bir koşula indirgenebilir. İkinci mertebeden niceleyicilerin elenmesi ile ilgili olarak geliştirilen çeşitli algoritmalar bulunmaktadır.

Bu tezde Sahlqvist formülleri ve teoremi verilerek Sahlqvist teoremi ayrıntılı olarak ispatlanmış. Ardından ikinci mertebeden niceleyicilerin elenmesi ile ilgili olarak geliştirilen iki önemli algoritma SCAN ve SQEMA incelenerek, algoritmaların Sahlqvist formüllerine göre tamlığı ispatlanmıştır. Tezde ele alınan teknik ve algoritmalar tezin sonuç kısmında karşılaştırılmıştır.

2. ÖN BİLGİLER

Bu bölümde tezin okunabilirliğini kolaylaştırmak amacıyla bazı temel tanım ve kavramlar verilmiştir.

2.1 Önermeler Mantığı

Önermeler mantığı her muhakemenin ya doğru ya da yanlış olduğu varsayımına dayalı en basit muhakeme modelini temsil eden bir lojiktir. Diğer lojiklerin çoğu bu lojik tarafından kapsanır ya da dili yeni bağlaçlarla zenginleştirilerek onun üzerine inşa edilir.

Tanım 2.1.1 L_0 önermeler mantığının *dili* aşağıdaki sembollerden oluşan belirli bir önermesel dildir:

- (a) Önerme değişkenleri: p_0, p_1, \dots
- (b) Önerme bağlaçları: $\wedge, \vee, \rightarrow, \neg, \leftrightarrow, \perp$,
- (c) Yardımcı semboller: $(,)$.

L_0 dilinin *formülleri* veya L_0 -*formüller* aşağıdaki gibi tanımlanır:

- (a) L_0 daki tüm değişkenler ve \perp sabiti birer L_0 -formüldür.
- (b) φ ve ψ L_0 -formüller ise $\varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi, \neg \varphi$ lerde L_0 -formüllerdir.
- (c) L_0 daki sembollerin bir dizisi bir formüldür eğer ve yalnız eğer bu dizi (a) ve (b) nin sonlu kez uygulanmasının bir sonucudur.

Önerme değişkenleri p, q, \dots (veya indisileri) ve formüller φ, ψ, \dots ile gösterilecektir. L_0 dilindeki tüm değişkenlerin kümesi $VarL_0$ ve L_0 daki tüm formüllerin kümesi de $ForL_0$ ile gösterilir.

Türetim Kuralları

Modus Ponens φ ve $\varphi \rightarrow \psi$ formüllerinden ψ türetilebilir.

Substitution (İkame) φ den bir φ_s türetilebilir; burada bir s ikamesi $VarL_0$ dan $ForL_0$ a bir dönüşümdür ve φ_s , φ nin inşası üzerine tümevarımla aşağıdaki gibi tanımlanır:

Her $p \in VarL_0$ için $p_s := s(p)$;

$\perp_s := \perp$;

$\Theta \in \{\wedge, \vee, \rightarrow\}$ için $(\varphi\Theta\psi)_s := \varphi_s\Theta\psi_s$

ve $(\neg p)_s := \neg p_s$

dir.

L_0 dilindeki bir lojik Modus Ponens ve substitution türetim kurallarına kapalı herhangi bir $L_0 \subseteq ForL_0$ kümesidir.

2. 2 Modal Lojik

Modal lojik Aristonun “olasıdır” ve “gereklidir” kelimelerini içeren durum analizleri ile başlamıştır. Bu kavramlar doğal ve teknik dillerde geniş çalışma alanları bulmuştur. Modal lojiğin bir matematik disiplini olarak kabul edilmesi C. I. Lewis’in 1918 tarihinde yaptığı “Survey of Symbolic Logic” adlı çalışmasıyla gerçekleşmiştir. Lewis önermeler lojiğine “olanaksızdır” anlamındaki birli I operatörünü ve “kesin gerektirir” anlamını ifade eden ikili operatör \prec operatörünü ekleyerek yeni bir aksiyom sistemi geliştirmiştir. 1932 yılında Lewis’in düşüncelerinden hareketle Lewis ve C. H. Langford modal lojiğin modern anlamdaki ilk aksiyomatik sistemlerini tanımlamışlardır.

Modal lojik 1918 den günümüze yalnızca bir matematik disiplini olmaktan çıkarak gerek dil bilimcilerin gerekse bilgisayar bilimcilerin üzerinde çalıştığı önemli bir bilim dalı olmuştur.

Tanım 2.2.1 \mathcal{O} boştan farklı bir küme ve $\rho: \mathcal{O} \rightarrow \mathbb{N}$ bir fonksiyon olmak üzere bir *modal benzerlik tipi* $\tau(\mathcal{O}, \rho)$ ikilidir; burada \mathcal{O} nun elemanları sonlu ariteli modal operatörlerdir.

Tanım 2.2.2 Bir *modal dil* ML önerme harflerinin kümesi Φ ve modal benzerlik tipi $\tau(\mathcal{O}, \rho)$ kullanılarak inşa edilir.

τ ve Φ üzerindeki *modal formüllerin kümesi* $For(\tau, \Phi)$ aşağıda verilen kurallarla tanımlanır:

$$\phi := p \mid \top \mid \neg \phi \mid \phi_1 \wedge \phi_2 \mid \Diamond \phi$$

Modal Çatılar ve Modeller

W boştan farklı bir küme ve R, W üzerinde bir ikili bağıntı olmak üzere bir *modal Kripke çatı* $\mathcal{F} = (W, R)$ ikilisinden oluşur. xRy ise y ye x den *ulaşılabilir* veya x, y yi *görür* denir. ML belirli bir önermesel dil olsun. ML nin $\mathcal{F} = (W, R)$ deki bir V *valuasyonu* (değer ataması) $VarML$ den 2^W ya bir dönüşümdür.

ML nin bir *Kripke modeli* bir $M = (\mathcal{F}, V)$ ikilidir.

Tanım 2.2.3 $x, M = (\mathcal{F}, V)$ Kripke modelinde bir nokta olsun. Bir $(M, x) \models \phi$ doğruluk bağıntısı, ϕ nin oluşumu üzerinde tümevarımla, aşağıdaki gibi tanımlanabilir:

$$(M, x) \models p \Leftrightarrow \text{Her } p \in \text{Var için } x \in V(p),$$

$$(M, x) \models \perp,$$

$$(M, x) \models \neg\varphi \Leftrightarrow (M, x) \not\models \varphi,$$

$$(M, x) \models \varphi \wedge \psi \Leftrightarrow (M, x) \models \varphi \text{ ve } (M, x) \models \psi,$$

$$(M, x) \models \diamond\varphi \Leftrightarrow \exists y \in W, Rxy \text{ ve } (M, y) \models \varphi,$$

$$(M, x) \models \Box\varphi \Leftrightarrow \forall y \in W, Rxy \text{ ise } (M, y) \models \varphi.$$

$(M, x) \models \varphi$ yerine daha yalın olarak $x \models \varphi$ yazılabilir.

Tanım 2.2.4 Bir φ formülü \mathcal{F} çatısı üzerindeki her M modelinin her x noktasında doğru ise o zaman bu x noktasına \mathcal{F} çatısında *geçerlidir* denir.

Bir φ formülünün bir F çatılar sınıfında geçerli olması için gerek ve yeter koşul bu φ formülünün F sınıfının her \mathcal{F} çatısında geçerli olmasıdır.

2.3 K Lojği

Kripke semantiğine göre sağlam ve tam olan modal lojğin minimal K lojğinin aksiyomları ve türetim kuralları aşağıdaki gibi tanımlanır:

Aksiyomları

(A1) Önergeler lojğinin tüm totolojileri,

(A2) $\Box(p_0 \rightarrow p_1) \rightarrow (\Box p_0 \rightarrow \Box p_1)$.

Türetim Kuralları:

(T1) Modüs Ponens,

(T2) Substitution,

(T3) Gereklilik Kuralı: Bir φ formülünden $\Box\varphi$ formülü türetilebilirdir.

Böylece *modal lojik* Modus Ponens ve substitution kullarına altında kapalı ve tüm klasik totolojileri içeren herhangi bir modal formüller kümesidir.

Tanım 2.3.1 φ bir modal formül, x birinci mertebeden lojiğin değişkeni ve P birli yüklem sembolü olmak üzere, *standart çeviri* $ST(\varphi)$, φ modal formülünün oluşumu üzerinde tümevarımla aşağıdaki gibi birinci mertebeden lojiğin diline çevirir.

$$ST_x(p) = Px,$$

$$ST_x(\top) = x = x,$$

$$ST_x(\neg\varphi) = \neg ST_x(\varphi),$$

$$ST_x(\varphi \wedge \psi) = ST_x(\varphi) \wedge ST_x(\psi),$$

$$ST_x(\Diamond\varphi) = \exists y(Rxy \wedge ST_x(\varphi)).$$

Örnek 2.3.2 $\Diamond(p \rightarrow q)$ formülünün standart çevirisi aşağıdaki şekilde yapılır:

$$ST_x(\Diamond(p \rightarrow q)) = \exists y(Rxy \wedge ST_y(\Box p \rightarrow q))$$

$$= \exists y(Rxy \wedge ST_y(\Box p) \rightarrow ST_y(q))$$

$$= \exists y(Rxy \wedge \forall z(Ryz \rightarrow Qz) \rightarrow Qy)$$

Tanım 2.3.3 Bir φ formülünde bir p önermesel harfinin deęillemelerinin sayısı çift ise p nin geçiřleri *pozitif* ve tek ise p nin geçiřleri *negatif* olarak adlandırılır.

Bir φ formülünde p nin tüm geçiřleri pozitif ise φ formülü p de pozitif ve tüm geçiřleri negatif ise φ formülü negatiftir. Bir φ formülü tüm önermesel harflerine göre pozitif ise φ formülü pozitif ve negatif ise φ formülü negatiftir.

Tanım 2.3.4 (Bir Formülün Monotonluęu) Her \mathfrak{F} çatısı, her $w \in \mathfrak{F}$ durumu ve \mathfrak{F} üzerindeki tüm V, V' deęer atamaları için her $q \in AT, p \neq q$ için $V(p) \subseteq V'(p)$ ve $V(q) = V'(q)$ ise φ formülü *yukarı doęru monotondur* denir. Eęer $(\mathfrak{F}, V), w \models \varphi$ ise $(\mathfrak{F}, V'), w \models \varphi$ dir veya denk olarak $\llbracket \varphi \rrbracket_{(\mathfrak{F}, V)} \subseteq \llbracket \varphi \rrbracket_{(\mathfrak{F}, V')}$ dir; burada $\llbracket \varphi \rrbracket_{(\mathfrak{F}, V)}$, Kripke model (\mathfrak{F}, V) de φ nin genişlemesidir.

Benzer olarak, her \mathfrak{F} çatısı için, her $w \in \mathfrak{F}$ durumu ve \mathfrak{F} üzerindeki her V, V' deęer atamaları her $q \in AT, p \neq q$ için $V'(p) \subseteq V(p)$ ve $V(q) = V'(q)$ ise φ formülü *ařaęıya doęru monotondur* denir. Eęer $\mathfrak{F}, V \models \varphi$ ise $\mathfrak{F}, V' \models \varphi$ veya denk olarak $\llbracket \varphi \rrbracket_{(\mathfrak{F}, V)} \subseteq \llbracket \varphi \rrbracket_{(\mathfrak{F}, V')}$ dir.

Bir pozitif (yukarı doęru monoton) formülün deęillemesi negatif (ařaęıya doęru monoton) formüldür.

p de ki her pozitif formül p de yukarı doęru monotondur.

Keyfi yukarı doęru monoton (özellikle pozitif) formülde deęiřkenler için yukarı doęru (ařaęıya doęru) monoton formülün ikamesi sonucu yukarı doęru (ařaęıya doęru) monoton formül elde edilir. Özellikle $\varphi(p)$, p de yukarı doęru (ařaęıya doęru) monotondur (eye) $\varphi(\neg p)$, p de ařaęıya doęru (yukarı doęru) monotondur.

3. SAHLQVİST FORMÜLLERİ ve TEKNİĞİ

Modal lojikte birinci mertebeden tanımlanabilirlik ve tamlığın en genel sonuçlarından biri Sahlqvist teoremidir. Sahlqvist, 1970 tarihli makalesinde, Sahlqvist formülleri olarak adlandırılan modal formüllerin belirli bir sınıfını tanımlayarak, bu sınıfın Kripke çatılar üzerinde birinci mertebeden koşulları tanımladığını ve bu koşulların modal formüllerden etkili bir biçimde hesaplanabildiğini göstermiştir. Ayrıca tüm Sahlqvist formüllerinin kanonik formüller olduğunu ve Sahlqvist formüllerinin kanonik çatılar sınıfına göre geçerli olduğunu ispatlamıştır. Sahlqvist formüller sınıfı alternatif ispatları ve genelleştirilmeleri kapsamlı bir şekilde incelenmiştir.

Bu bölümde Sahlqvist formülü tanıtılarak Sahlqvist teoremi verilecek ve Sahlqvist tekniğinin modal formüller üzerindeki etkinliği açıklanacaktır.

3.1. Ön Bilgiler

Tanım 3.1.1 Bir φ formülünde bir p önerme harfinin deęillemelerinin sayısı çift (tek) ise p nin geişlerine *pozitif* (*negatif*) tir denir.

Bir φ formülünde p nin tüm geişleri pozitif (negatif) ise o zaman φ formülü p de pozitif (negatif) tir.

Bir φ formülü tüm önerme harflerine göre pozitif (negatif) ise φ formülü pozitif (negatif) tir.

Tanım 3.1.2 p bir önerme harfi olmak üzere $\square \dots \square p$ biçimli bir formüle *box-lı formül* denir. $\square \dots \square$ zinciri boş olabilir.

3.2 Sahlqvist Formülleri ve Teoremi

Tanım 3.2.1 Bir φ formülü, \top , \perp , negatif formüller ve \Box lı atomlara, \wedge , \vee önerme bağlaçları ve \Diamond operatörü uygulanarak elde edilen bir formül ise φ formülüne bir *Sahlqvist öncülü* denir.

Tanım 3.2.2 φ bir Sahlqvist öncülü ve ϕ bir pozitif formül olmak üzere $\varphi \rightarrow \phi$ şeklindeki bir formüle *Sahlqvist gerektirmesi* denir.

Tanım 3.2.3 Bir φ formülü Sahlqvist gerektirmelerine \Box operatörü, \wedge önerme bağlacı ve hiçbir ortak önerme harfi içermeyen iki formülün \vee bağlacı uygulanmasıyla elde edilen bir formül ise φ formülüne bir *Sahlqvist formülü* denir.

Sahlqvist öncüllerine aşağıdaki işlemler uygulanmaz:

- Veya bağlacı üzerine box operatörü:

$$\Box(p \vee q)$$

- Diamond operatörü üzerine box operatörü:

$$\Box \Diamond p \rightarrow q$$

Teorem 3.2.4 (Sahlqvist Teoremi, 1975) τ , bir modal benzerlik tipi ve χ, τ üzerinde bir Sahlqvist formülü olsun. O zaman χ , çatılar üzerinde birinci mertebeden bir $c_x(\chi)$ formülüne yerel olarak karşılık gelir.

Kanıt (Taslak) $\chi, \varphi \rightarrow \phi$ Sahlqvist formülü formunda olsun; burada φ , bir Sahlqvist öncülü ve ϕ ise pozitif bir formüldür.

Adım 1 χ formülünün ikinci mertebeden standart çevirisi yapılır. Elde edilen formül

$$\forall P_1 \dots \forall P_n (ST_x(\varphi) \rightarrow ST_x(\phi))$$

biçimindedir. Eğer formülün öncülünde varlıksal niceleyici mevcut ise o zaman aşağıdaki mantıksal denklikler yardımıyla niceleyici parantezin dışına alınır.

$$\exists x_i \alpha(x_i) \wedge \beta \equiv \exists x_i (\alpha(x_i) \wedge \beta),$$

$$(\exists x_i \alpha(x_i) \rightarrow \beta) \equiv \forall x_i (\alpha(x_i) \rightarrow \beta).$$

Adım 2 Birinci adım sonunda elde edilen formül üç formda olabilir:

$$(i) \quad \forall P_1 \dots \forall P_n \forall x_1 \dots \forall x_n (Bağ \wedge At \rightarrow Poz),$$

$$(ii) \quad \forall P_1 \dots \forall P_n \forall x_1 \dots \forall x_n (Bağ \wedge Box - At \rightarrow Poz),$$

$$(iii) \forall P_1 \dots \forall P_n \forall x_1 \dots \forall x_n (Bağ \wedge At \wedge Neg \rightarrow Poz).$$

(iii) deki form

$$\alpha \wedge Neg \rightarrow \beta \equiv \alpha \rightarrow \beta \vee \neg Neg$$

denkliği yardımıyla birinci forma dönüştürülebilir.

Adım 3 (Minimal atama)

(i) ve (iii) formundaki formüller için

$$V(P_i) \equiv (u = x_{i_1} \vee \dots \vee u = x_{i_k})$$

ataması yapılır. (ii) formundaki

$$\forall P_1 \dots \forall P_n \forall x_1 \dots \forall x_n (Bağ \wedge Box - At \rightarrow Poz)$$

formül box-lı atomların çevirisi

$$\forall y (R_{\beta_i} x_{i_1} y \rightarrow Py)$$

şeklindedir. Bu formüller için

$$V(P_i) \equiv \forall x_{i_1} (R_{\beta_i} x_{i_1} u \vee \dots \vee R_{\beta_k} x_{i_k} u)$$

ataması yapılır.

3.Adım sonunda elde edilen formülün sadeleştirilmesiyle verilen χ Sahlqvist formülüne karşılık gelen birinci mertebeden formül $c_x(\chi)$ i elde ederiz.□

3.3 Örnekler

Örnek 3.3.1. $(p \wedge \diamond \neg p) \rightarrow \diamond p$ Sahlqvist formülüne karşılık gelen birinci mertebeden formülü bulalım.

Formülün ikinci mertebeden standart çevirisi

$$\forall P(Px \wedge \exists y(Rxy \wedge \neg Py) \rightarrow \exists z(Rxz \wedge Pz))$$

dir. Varlıksal niceleyiciyi parantezin dışına alınırsa

$$\forall P \forall y (Px \wedge (Rxy \wedge \neg Py) \rightarrow \exists z (Rxz \wedge Pz))$$

↑ ↑ ↑ ↑

Box-At Bağ Neg Poz

elde edilir. $\neg P$ yi \rightarrow bağlacının sağına alınırsa

$$\forall P \forall y (Px \wedge Rxy \rightarrow Py \vee \exists z (Rxz \wedge Pz))$$

bulunur. Px i doğru yapan minimal atama $u = x$ dir. Atamadan sonra

$$\forall y (Rxy \rightarrow y = x \vee \exists z (Rxz \wedge z = x))$$

elde edilir. İfadeyi sadeleştirirsek, $(p \wedge \diamond \neg p) \rightarrow \diamond p$ Sahlqvist formülüne karşılık gelen birinci mertebeden

$$\forall y(Rxy \wedge x \neq y \rightarrow Rxx)$$

formül elde edilir.

Örnek 3.3.2. $\diamond p \rightarrow \square \diamond p$ Sahlqvist formülüne karşılık gelen birinci mertebeden formül aşağıdaki gibi elde edilebilir.

Formülün ikinci mertebeden çevirisi yapılırsa

$$\forall P(\exists y(Rxy \wedge Py) \rightarrow \forall z(Rxz \rightarrow \exists v(Rzv \wedge Py)))$$

elde edilir. Formüldeki varlıksal niceleyici parantezin dışına alınırsa

$$\forall P \forall y((Rxy \wedge Py) \rightarrow \forall z(Rxz \rightarrow \exists v(Rzv \wedge Py)))$$

↑ ↑

REL Box-At

bulunur. Py yi doğru yapan minimal atama

$$V(p) \equiv y = y$$

dir. Atamadan sonra elde edilen formül

$$\forall y((Rxy \wedge y = y) \rightarrow \forall z(Rxz \rightarrow \exists v(Rzv \wedge v = y)))$$

olur. Formül sadeleştirilirse verilen Sahlqvist formülüne karşılık gelen birinci mertebeden formül elde edilir:

$$\forall x \forall y \forall z(Rxz \wedge Rxz \rightarrow Rzy).$$

Bu formül birinci mertebede lojikte Euclidean bağıntısına karşılık gelir.

4. SCAN

Bir modal formülün her zaman birinci mertebeden bir çatı koşulu tanımlamadığı bir önceki bölümde gösterilmiştir. Bir modal formüle karşılık gelen birinci mertebeden formül hesaplanırken evrensel monadic ikinci mertebeden niceleyicilerin elenmesi o formülün bir çatıda geçerliliğini, varlıksal monadic niceleyicilerin elenmesi formülün sağlanabilirliğini ifade eder. Varlıksal monadic ikinci mertebeden niceleyicilerin elenmesi yöntemi ile ilgili en bilinen algoritma SCAN algoritmasıdır. H. J. Ohlbach ve D. Gabbay tarafından geliştirilen SCAN algoritmasının ana fikri W. Ackermann'ın makalelerinde görünmektedir.

Bu bölümde SCAN algoritmasında kullanılan temel tanımlar ve kavramlar verilecek. SCAN algoritmasının adımları ayrıntılı bir şekilde sunularak, algoritmanın etkililiği örneklerle açıklanacaktır. Bölüm sonunda algoritmanın sınırları verilecek ve SCAN algoritmasının Sahlqvist formüllerine göre tamlığı kanıtlanacaktır.

4.1. Ön Bilgiler

Tanım 4.1.1 L_1, \dots, L_n önerme değişkenleri olmak üzere $\{L_1, \dots, L_n\}$ sonlu kümesine *clause* denir.

Tanım 4.1.2 C ve D clause ve P bir önerme değişkeni olmak üzere *resolution kuralı*

$$\frac{P(s_1, \dots, s_n) \vee C \quad \neg P(t_1, \dots, t_n) \vee D}{C \vee D \vee s_1 \neq t_1 \vee \dots \vee s_n \neq t_n}$$

dir. Burada iki clause P üzerinden silinmiştir ve $C \vee D$ *resolvent* (çözücü) olarak adlandırılır.

Örnek 4.1.3 $\{\neg P, Q\}, \{\neg Q, \neg R, S\}, \{P\}, \{R\}, \{\neg S\}$ clause kümeleri üzerinde resolution türetimi ile boş clause elde edilir.

1. $\{\neg S\}$

2. $\{\neg Q, \neg R, S\}$

3. $\{\neg Q, \neg R\}$ (resolution 1,2)

4. $\{R\}$

5. $\{\neg Q\}$ (resolution 3,4)

6. $\{\neg P, Q\}$

7. $\{\neg P\}$ (resolution 5,6)

8. $\{P\}$

9. $\{\}$ (resolution 7,8)

Tanım 4.1.4 C bir clause ve P bir önerme değişken olmak üzere *faktörizasyon kuralı*

$$\frac{P(s_1, \dots, s_n) \vee P(t_1, \dots, t_n) \vee C}{P(s_1, \dots, s_n) \vee C \quad \vee s_1 \neq t_1 \vee \dots \vee s_n \neq t_n}$$

dir.

Faktörizasyon kuralı yalnızca farklı clauselar arasında yapılabilir.

Skolemizasyon Tekniđi

Tanım 4.1.5 Prenex formdaki birinci mertebeden bir formüldeki varlıksal niceleyiciler sistematik olarak elenerek yerlerine yeni sabit sembolleri ve fonksiyon sembolleri atanması işleme *Skolemizasyon (Skolemleştirme)* adı verilir. Yeni sabit sembollerine ve fonksiyon sembollerine sırasıyla *Skolem sabitleri* ve *Skolem fonksiyonları* adı verilir.

$\exists x \forall y \forall z A$ birinci mertebeden lojiđin bir formülü ve c Skolem sabiti olmak üzere birinci mertebeden formülün Skolemizasyon sonucu

$$\forall y \forall z A[c/x]$$

dir.

$\forall y \exists z P(y, z)$ birinci mertebeden lojiđin bir formülü ve f Skolem fonksiyonu olmak üzere formülün Skolemizasyon sonucu

$$\forall y P(y, f(y))$$

dir.

4.2 SCAN Algoritması

Tanım 4.2.1 (SCAN Algoritması)

SCAN algoritmasında girdi: $\alpha = \exists P_1, \dots, P_n \psi$ şeklinde bir α formülüdür; burada P_1, \dots, P_n yüklem sembolleri ve ψ keyfi birinci mertebeden lojiđin bir formüldür.

SCAN algoritmasında çıktı: P_1, \dots, P_n yüklem deđişkenlerini içermeyen ve α formülüne mantıksal denk olan φ_α formülüdür.

SCAN algoritması aşağıdaki üç adımdan oluşmaktadır:

1. Verilen formülün değillemesi alınır ve ikinci mertebeden standart çevirisi yapılarak $\exists P_1, \dots, P_n \psi$ biçimine indirgenir. ψ formülüne ikinci mertebeden Skolemizasyon kuralı uygulanarak clause forma dönüştürülür. İşlem sonucunda f_i ler Skolem fonksiyonları, ψ' clause ların bir kümesi olmak üzere ψ formülü

$$\exists P_1, \dots, P_n, \exists f_1, \dots, f_n \psi'$$

biçimli formüle indirgenir.

2. C_1, \dots, C_n ler clause olmak üzere resolution, faktörizasyon kuralları uygulanarak clause' lar silinir. Eğer tüm clause' lar silinmiş ise bunun anlamı çelişkidir.

3. Eğer bir önceki adımda algoritma sonlanırsa ve kalan clause varsa, ters Skolemizasyon uygulanır ve elde edilen formülün değillemesi alınır.

Bir örnekle SCAN algoritmasının farklı adımlarını detaylı olarak göstereyim.

Algoritmada girdi olarak

$$\exists P \forall x, y \exists z (\neg P(a) \vee Q(x)) \wedge (P(y) \vee Q(a)) \wedge P(z)$$

formülünü alalım.

Birinci adımda ikinci mertebeden skolemizasyon kuralı kullanılarak clause formlar hesaplanır.

f bir skolem fonksiyonu, ikinci mertebeden değişkenler ve niceleyicileri $\exists P, \exists f \forall x, y$ olmak üzere elde edilen clause form aşağıdaki şekildedir:

$$C_1 \quad \neg P(a), Q(x)$$

$$C_2 \quad P(y), Q(a)$$

$$C_3 \quad P(f(x, y))$$

Algoritmanın ikinci adımında $\neg P(a)$ seçilerek çözümlenmeye başlanır.

C_1 ve C_2 arasındaki çözücü

$$C_4 = Q(x), Q(a)$$

dır. Burada x değişkeni yerine a ataması yapılırsa $C_4, Q(a)$ ya denk olur.

C_1 ve C_3 arasındaki çözücü

$$C_5 = a \neq f(x, y), Q(x)$$

dür. $\neg P(a)$ ile daha fazla çözücü kalmadı. Böylece C_1 clause silindi.

Geriye kalan clause' lar:

$$C_2 \quad P(y), Q(a)$$

$$C_3 \quad P(f(x, y))$$

$$C_4 \quad Q(a)$$

$$C_5 \quad a \neq f(x, y), Q(x)$$

Diğer iki P önerme değişkenini çözecek yeni çözücüler yoktur. Bu nedenle C_2 ve C_3 kolayca silinebilir. Tüm niceleyicileri yeniden düzenlersek

$$\forall x \exists z Q(a) \wedge (a \neq z \vee Q(x))$$

elde ederiz.

Teorem 4.2.2 SCAN algoritması ikinci mertebeden lojiğin bir α formülünü sonuçlandırıyor α , $SCAN(\alpha)$ ya mantıksal denktir.

4.3 Örnekler

Örnek 4.3.1 K4 Aksiyomu:

$$\forall P \Box P \rightarrow \Box \Box P$$

$$\forall P \forall a (\Box P \rightarrow a \models \Box \Box P)$$

\Box in Semantiği: $\forall P \forall w (w \models \Box P) \Leftrightarrow (\forall v R w v \rightarrow v \models P)$

Aksiyomun Değili: $\exists P \exists a \neg (a \models \Box P \wedge \Diamond \Diamond \neg P)$

Standart Çeviri: $\exists P \exists a ((\forall v R a v \rightarrow P(x)) \wedge \exists b R a b \wedge \exists c R b c \wedge \neg P(c))$

Clause Form: $\neg R a v, P(v)$

$R a b$

$R a c$

$\neg P(c)$

P resolved away: $\neg R a c$

$R a b$

$R b c$

Unskolemizasyon: $\exists a, b, c \neg Rac \wedge Rab \wedge Rbc$

Değillemesi: $\forall a, b, c Rbc \wedge Rab \rightarrow Rac$

Örnek 4.3.2 McKinsey Aksiyomu :

$$\forall P(\Box \Diamond P \rightarrow \Diamond \Box P)$$

Formülün Doğruluğu $\forall P \forall a (a \models \Box \Diamond P \rightarrow \Diamond \Box P)$

Sematik: $\forall P \forall w \Box P \models w \Leftrightarrow (\forall v R w v \rightarrow \Box P \models v)$

Aksiyomun Değillemesi: $\exists P \exists a (a \models \Box \Diamond P \wedge \Box \Diamond \neg P)$

Standart Çeviri:

$$\exists a \forall x (Rax \rightarrow \exists y (Rxy \wedge P(y))) \wedge \forall x Rax \rightarrow \exists y (Rxy \wedge \neg P(y))$$

Clause Form: $\neg Rax, R(x, f(x))$

$$\neg Ray, R(y, g(y))$$

$$\neg Rax, P(f(x))$$

$$\neg Ray, \neg P(g(y))$$

P Resolved Away: $\neg Rax, R(x, f(x))$

$$\neg Ray, R(y, g(y))$$

$$\neg Rax, \neg Ray, f(x) \neq g(y)$$

İkinci Mertebeden Niceleyiciler için Unskolemizasyon:

$$\exists a \left(\begin{array}{l} \exists f \quad \forall x \\ \exists g \quad \forall y \end{array} \right) (Rax \rightarrow Rxf(x)) \wedge (Ray \rightarrow Ryg(y)) \wedge (\neg Rax \vee \neg Ray \vee f(x) \neq g(y))$$

Değillemesi:

$$\forall a \left(\begin{array}{l} \forall f \exists x \\ \forall g \exists y \end{array} \right) (Ray \rightarrow Rxf(x)) \wedge (Ray \rightarrow Ryg(y)) \rightarrow \\ (Rax \wedge Ray \wedge f(x) = g(y))$$

Örnek 4.3.3 Euclidean

$$\forall P \diamond P \rightarrow \square \diamond P$$

Formülün Doğruluğu $\forall P \forall a (a \models \diamond \square P \rightarrow \square \diamond P)$

Semantik: $\forall P \forall w w \models \diamond P \Leftrightarrow (\exists v R w v \wedge (v \models P))$

Aksiyomun Değillemesi: $\exists P \exists a (a \not\models \diamond \square P \wedge \square \neg P)$

Standart Çeviri: $\exists P \exists a ((\exists x R a x \wedge P(x)) \wedge (\exists y R a y \wedge \forall z R y z \rightarrow \neg P(z)))$

Clause Form: Rax

$$P(x)$$

$$Ray$$

$$\neg Ryz, \neg P(z)$$

P Resolved Away : Rax

$$Ray$$

$$\neg Ryx$$

Unskolemizasyon: $\exists a, x, y (Rax \wedge Ray \wedge \neg Ryx)$

Değillemesi: $\forall a, y, x \neg (Rax \wedge Ray) \vee Ryx$

Sadeleştirirsek

$$\forall a, x, y (Rax \wedge Ray \rightarrow Ryx)$$

elde ederiz.

4.4 SCAN Algoritmasının Sınırları

Modal lojik bilindiği gibi Hilbert aksiyomlarının sahip olduğu semantik özellikler nedeniyle yalnızca ikinci mertebeden aksiyomatize edilebilirler. İlave edilen aksiyomlarla birlikte verilen bir formülün denk olduğu formül, ulaşılabilirlik bağıntısının birinci mertebeden aksiyomatizasyonu sayesinde bulunabilir. Örnek olarak $\forall P(\Box \Diamond P \rightarrow \Diamond \Box P)$ McKinsey aksiyomunun denk olduğu formül yalnızca ulaşılabilirlik bağıntısının ikinci mertebeden özelliği kullanılarak bulunabilir (SCAN algoritması ters Skolemizasyon sırasında ikinci mertebeden Henkin niceleyicilerine ihtiyaç duyar). Geçişme aksiyomu ile birleştirilirse, bu iki tanım atomik olarak $\forall x \exists y (Rxy) \wedge \forall z Ryz \rightarrow z = y$ denktir. Açıkça görülmektedir ki bu birinci mertebeden tanımlanabilen bir özelliktir.

McKinsey aksiyomunun uygulanmasında, eğer kritik clause ters Skolemizasyona engel olsaydı onun factorleri normal yolla yeniden yerleştirilirdi. Geçişme özelliğinin belirli durumlarda neden bu operasyona olanak verdiğine dair bazı fikirler olmasına rağmen, konu hakkında genel bir teoriye sahip olunmamaktadır. Aslında McKinsey aksiyomu geçişme aksiyomu ile birlikte atomik olarak seçme aksiyomuna karşılık gelmektedir. Beklenildiği üzere bu problemin basit bir çözümü bulunmamaktadır.

SCAN algoritması tam değildir, birinci mertebeden denk formülü her zaman hesaplayamaz. İkinci mertebeden iki formülün veya laması alındığında, bu formüllerden birinin denk olduğu birinci mertebeden formül bulunmasa bile formüllerden birine ya da her ikisine denk olan birinci mertebeden formülün bulunabilir olduğu açıktır.

4.5 Sahlqvist Formüller için SCAN Algoritmasının Tamlığı

Bu bölümde SCAN algoritmasının Sahlqvist formüllere göre tamlığı gösterilecektir. Verilen bir Sahlqvist formül φ ye önişlemler uygulanarak elde edilen ikinci mertebeden keyfi ψ formülünün, birinci mertebeden mantıksal denk olduğu formülün SCAN tarafından hesaplanabildiği kanıtlanmalıdır. Bu nedenle aşağıdaki iki özelliğin varlığı bilinmelidir.

1. C resolventler ve C faktörlerinin hesaplanması ψ ile ilgili clauseların kümesi $Cls(\psi)$ uygulandığında sonlanır. Örneğin SCAN, işlemde yalnızca sonlu sayıda yeni clause olduğunda çalışır.

2. Sonuçta oluşan birinci mertebeden formül (genel olarak Skolem fonksiyonlarını içerir) başarılı bir şekilde ters Skolemizasyon edilebilir.

Her Sahlqvist formül temel Sahlqvist formülün \forall bağlacıyla bağlanmış formuna denk olduğu için her temel Sahlqvist formül φ için ispatlamak yeterlidir.

İlk olarak φ nin bir Sahlqvist gerektirmesi olduğu göz önünde bulundurulmalıdır. İspatın köşe taşı zincir notasyonudur.

(t_1, \dots, t_n) ayrık terimlerin sıralı dizisi olsun. Bir C zinciri, (t_1, \dots, t_n) üzerinde bir clause' dır; $(\neg)R_{k_i}st$, $(\neg)Q_j(u)$ formundaki önermesel değişkenleri içerir ve aşağıdaki üç koşulu sağlar:

- (1) Her i , $1 \leq i \leq n - 1$ için ya $\neg R_{k_i}t_i t_{i+1}$ ya da $R_{k_i}t_i t_{i+1}$, C de dir;
- (2) Her $(\neg)R_{k_i}uv \in C$ için $u = t_j$ ve bazı j , $1 \leq j \leq n - 1$ ler için $v = t_{j+1}$;
- (3) Her $(\neg)Q_j uv \in C$ ve bazı j , $1 \leq j \leq n - 1$ ler için $u = t_j$ dir.

Lemma 4.5.1 $C(t_1, \dots, t_n)$ üzerinde bir zincir ise (t_1, \dots, t_n) lerden ayrık bir (s_1, \dots, s_n) sıralı dizisi yoktur öyle ki $C, (s_1, \dots, s_n)$ üzerinde de bir zincirdir.

Bir C zincirinin (t_1, \dots, t_n) üzerindeki uzunluğu n dir.

□

Lemma 4.5.1 de C zinciri tek bir şekilde (t_1, \dots, t_n) i belirler. Bu nedenle bir zincirin uzunluğu iyi tanımlanmış notasyondur.

Lemma 4.5.2 Σ , ikili ve birli yüklem sembollerinin, sabit sembollerin ve birli fonksiyon sembollerinin bir sonlu kelimesi olsun. Eğer bir zincirin üzerinde bir J_b sınırı ve terimlerin derinlikleri üzerinde bir d_b sınırı varsa Σ üzerinde zincirlerin sayısı ayrık modulo değişkenleridir ki bu sonlu olarak sınırlandırılmıştır.

Teorem 4.5.3 (Sahlqvist formüllere göre tamlık) SCAN, keyfi bir Sahlqvist gerektirmesi φ nin birinci mertebeden mantıksal denk olduğu formül α_φ yi hesaplar.

Kanıt $N, ST(\neg\varphi, \alpha)$ nın clause formu olsun. Tüm clause'lar N den türetilir öyle ki N, N nin kelimeleri üzerindeki zincirlerden meydana gelir. Zincirlerin uzunluğu, N deki zincirlerin uzunluğu tarafından sınırlandırılır.

Göstermeliyiz ki her ne zaman bir clause çok sayıda zincir ile türetilirse bu clause yoğunlaşır. Burada çok sayıda clause olduğunun belirtilmesi Lemma 4.5.2 deki sınırlı kelimesini vurgulamak içindir.

Ters Skolemizasyona durumunda tümevarımsal argümanlar kullanılarak eşitsizlikler ve terimlerde meydana gelen türetilmiş clause'ların başarılı bir şekilde unskolemize edilebileceği gösterilebilir. Sonuç olarak niceleyicilerin ters Skolemizasyon sırasında her zaman başarı ile düzenlenebileceği gösterilir.

Son olarak temel bir Sahlqvist formül φ , box operatörleri ve hiçbir ortak önerme harfi içermeyen iki formüle veya bağlacı uygulanarak elde edilir.

Teorem 4.5.4 SCAN, keyfi bir Sahlqvist formülü φ nin, birinci mertebeden mantıksal denk olduğu formül α_φ yi etkili bir biçimde hesaplar.

5. SQEMA

SQEMA ikinci mertebeden niceleyicilerin elenmesine yönelik geliştirilen en yeni algoritmadır. SQEMA algoritması Ackermann lemmasının modal versiyonunu kullanarak diğer algoritmaların aksine modal formüller üzerinde direkt olarak çalışır. Bu nedenden dolayı diğer algoritmalarından çok daha etkili ve yalın bir algoritmadır. Bu bölümde SQEMA algoritmasının temel kavramlarına ve tanımlarına değinilecektir. Ardından algoritmanın işleyişi ayrıntılı olarak sunularak modal formüller üzerindeki etkinliği örneklerle açıklanacaktır. Bölümün sonunda ise SQEMA algoritmasının Sahlqvist formüllere göre tamlığı ispatlanacaktır.

5.1 Ön Bilgiler

Algoritmanın işleyişini güçlendirmek için aşağıdakiler modal dile eklenir:

- Ters modalite \Box^{-1} :

$$M, u \Vdash \Box^{-1}\varphi \text{ (eye) her } w \in M \text{ için } R^{-1}uw$$

Ters modalite için standart çeviri :

$$ST(\Box^{-1}, \varphi) := \forall y (Ryx \rightarrow ST(\varphi, y))$$

dir. \Box^{-1} in duali \Diamond^{-1} olarak tanımlanır.

- Nominaller, özel bir tür önermesel değişkenler $Nom = \{i_1, i_2, \dots\}$ in değer atamalarının singleton-lara kısıtlanmasıdır. Nominallerin doğruluk tanımları ve standart çevirileri aşağıdaki şekilde verilmiştir:

$$(w, V), u \Vdash i \text{ (eye) } V(i) = \{u\}.$$

$ST(i_i, x) := x = y_i$ öyle ki burada y_0, y_1, \dots ler i_0, i_1, \dots nominalleri ile ilgili korunmuş değişkenlerdir.

Modal dilin bu genişlemesi ML^+ ile gösterilir.

M bir model ve φ, ML^+ nin bir formülü olsun. $\llbracket \varphi \rrbracket_M$, bir M modelinde φ formülünün genişlemesidir ve $\llbracket \varphi \rrbracket_M = \{m \in M : M, m \Vdash \varphi\}$ biçiminde tanımlanır.

ML^+ da bir saf formül önermesel değişkenleri içermeyen fakat nominalleri içerebilen bir formüldür. Her saf formül $\gamma, \forall \bar{y} ST(\gamma, x)$ formülü tarafından yerel birinci mertebeden tanımlanabilir; burada \bar{y}, γ da meydana gelen tüm y_i değişkenlerinin tuple-larına karşılık gelen i_i nominalleridir.

$A, B(p), ML^+$ da bir formül olsun. $B(A/p), p$ nin tüm geçişleri için A nin aynı ikameleri tarafından $B(p)$ den elde edilen formüldür.

ML^+ biçimsel olarak algoritmanın çalışması için yeterli olsa da bazen evrensel modalite $[U]$ ile dili güçlendirmek gerekli olabilir.

Evrensel modalite $[U]$ nun semantiği:

$$\begin{aligned} M, u \Vdash [U]\varphi & \text{ (eye) her } w \in M \text{ için } M, w \Vdash \varphi \\ & \text{ (eye) } M \Vdash \varphi \end{aligned}$$

$\langle U \rangle, [U]$ nun duali olarak tanımlanır:

$$\begin{aligned} M, u \Vdash \langle U \rangle \varphi & \text{ (eye) bazı } w \in M \text{ için } M, u \Vdash \varphi \\ & \text{ (eye) } M, u \Vdash \neg [U] \neg \varphi \end{aligned}$$

$\langle U \rangle, [U]$ standart çevirileri:

$$ST([U]\varphi, x) := \forall x ST(\varphi, x) \text{ ve } ST(\langle U \rangle \varphi, x) := \exists x ST(\varphi, x)$$

Özellikle bir sonraki alt bölümde Ackermann lemmasının modal versiyonun tekrarlanma aşamasında evrensel modalite kullanılacaktır.

Ackermann Lemmasının Modal Versiyonu

Bu bölümde aksi belirtilmediği takdirde ML^+ da çalışılacaktır. Ayrıca $PROP$ önerme değişkenlerinin ve NOM nominallerin kümesi olmak üzere bu iki kümesinin birleşimi yerine kısaca AT kısaltması kullanılacaktır.

Not: Pozitif ve negatif formül tanımlarında nominaller göz önünde bulundurulmayacaktır. . Ayrıca bir saf formül φ i, hem *pozitif* hem de *negatif* kabul edilir.

Modal Ackermann Lemması $A, B(p)$, ML^+ da bir formül, A, p yi içermeyen keyfi bir modal formül ve $B(p)$, p ye göre pozitif bir modal formül olsun. Keyfi bir M modeli için $M \Vdash B(A)$ olması için gerek ve yeter koşul $M' \Vdash (A \rightarrow p) \wedge B(p)$ olmasıdır.

Lemmayı pozitif formüllere göre uyarlamak mümkündür.

Ackermann Lemması ve İkame Metodu

Ackermann lemmasının devrik formu:

A, p yi içermeyen keyfi bir modal formül ve B, p ye göre pozitif bir modal formül olmak üzere

$$\forall p ([U](A \rightarrow p) \rightarrow B(p)) \equiv B(A/p)$$

İfadeleri denktir.

Denklikten de görüldüğü gibi bir M modelinde $[U](A \rightarrow p)$ formülü doğrudur ancak ve ancak $\llbracket A \rrbracket_M \subseteq \llbracket p \rrbracket_M$ dir.

Yukarıdaki denklik aşağıdaki şekilde yorumlanabilir:

Bir \mathfrak{F} çatısında , $\forall p ([U](A \rightarrow p) \rightarrow B(p))$ geçerlidir ancak ve ancak $B(p)$ öncülü A nın sağlandığı minimal model için doğrudur.

5. 2 SQEMA Algoritması

Bu bölümde biçimsel olarak SQEMA algoritması temel modal dil için sunulacaktır. Fakat SQEMA, keyfi polyadic ve hybrid multi modal dillere genişletilebilir.

Temel Algoritma

Bir modal formül φ girdi olarak verildiğinde değil alınarak değil normal forma çevirilir. Daha sonra yerel çatı denkleğini koruyan dönüşüm kuralları uygulanarak verilen formül $\alpha \rightarrow \beta$ formu formüle dönüştürülür. Burada α ve β değil normal formdadır. Elde edilen formül denklem olarak adlandırılır.

Tüm denklemler global durumlarda yorumlanır ve tüm modellerde geçerlidirler.

Algoritma denklem sistemleri ile çalışmaktadır. Her bir dönüşüm kuralı denklemi bir veya daha çok yeni denklem içine dönüştürür.

Ackermann kuralı, Ackermann lemmayı temel almaktadır ve tüm denklem sistemlerine uygulanabilir.

Algoritmanın bu aşamadaki adımında bir önermesel değişken elenmek için seçilir. Elde edilen denklem sistemleri dönüşüm kuralları ve Ackermann kuralı uygulanarak yeni denklem sistemleri içine dönüştürülür. Bu işlemler sırasında seçilen önermesel değişken elenir. Tüm önermesel değişkenler aynı işlemler uygulanarak sırayla elenirler. Algoritmanın başarısı önermesel değişkenlerin eleme sıralarına bağlıdır.

Artık algoritmanın daha biçimsel bir tanımı verilebilir. Algoritma girdi olarak bir φ modal formülünü alır ve aşağıda işlemleri sırasıyla uygular:

Adım 1 φ nin değillemesi alınır ve değil normal formdaki formül $\rightarrow, \leftrightarrow$ bağlaçları ve içeride bulunan tüm değil işaretleri kaybolana kadar önermesel değişkenlerinin önüne sürülür.

Elde edilen formülü $\forall \alpha_k$ formu formüle indirgemek için

$$\diamond (\varphi \vee \psi) \equiv (\diamond \varphi \vee \diamond \psi) \text{ ve } (\varphi \vee \psi) \wedge \theta \equiv (\varphi \wedge \theta) \vee (\psi \wedge \theta)$$

denklikleri kullanılır.

Artık algoritma her bir evetleme üzerinde α_k yı ayırarak çalışmaya devam edebilir.

Adım 2 $\alpha_k, i \rightarrow \alpha_k$ olarak yeniden yazılır. Burada i, α_k da geçmeyen ve yalnızca o anki başlangıç durumunda kullanılan sabit, korunmuş nominaldir. Bu başlangıç sistemi içindeki tek denklemdir.

Adım 3 Her önermesel değişkeni elemek için sistemdeki negatif ve pozitif önermesel değişkenler yerine sırasıyla \top ve \perp yerleştirilir.

Adım 4 Eğer sistemin denklemleri içinde elenmemiş önermesel değişkenler kaldıysa, elenmek için seçilir. Aksi takdirde Adım 5 e geçilir. Eğer kalan değişkenlerin tamamı elendi ve Adım 5 başarısız olduysa elemanın sırası değiştirilerek değişkenler tekrar elenir. Eğer elemanın tüm sıralamaları kalan değişkenlere uygulandıktan sonra Adım 5 yine başarısız olursa, rapor başarısızdır. Eğer sistemdeki tüm önermesel değişkenler elendi ise Adım 6 ya geçilebilir.

Adım 5 Bu adımın amacı listedeki dönüşüm kurallarını uygulayarak seçili değişken p ye göre Ackermann kuralını uygulamak ve p yi elemek için denklemlerin sistemini yeniden yazmaktır.

Böylece her denklem ya negatif ya da $\alpha \rightarrow p$ formunun içine dönüştürülmeye çalışılır. Buradaki amaç p yi denklemden eleyerek denklemi çözmektir. Eğer bu adım başarısız olursa geri dönülür ve p yerine $\neg p$ ataması ile aynı adım tekrarlanır. Tekrar başarısız olursa ya da bir önceki adımda tamamlanırsa Adım 4 e dönülür.

Adım 6 Eğer tüm adımlar uygulanarak Adım 6 ya ulaşırsa, girdi formülündeki tüm önermesel değişkenler, sistemlerin sonuçlandırılmasıyla başarılı bir şekilde elenmiş demektir.

Son olarak her bir sistemdeki tüm denklemlerin evetlemeleri alınarak bir saf formül ya da $\forall \bar{y} \exists x_0 ST(\neg pure, x_0)$ biçimli bir formül elde edilir ve girdi formülü φ ye karşılık gelen yerel birinci mertebeden bir formül elde edilir.

Dönüşüm Kuralları

SQEMA tarafında kullanılan dönüşüm kulları aşağıda verilmiştir.

I. Mantıksal Bağlaçlar için Kurallar

Kuralın Adı

Formül

\wedge - Kuralı :

$$\begin{aligned} & \beta \rightarrow \gamma \wedge \delta \\ & \Downarrow \\ & \beta \rightarrow \gamma, \beta \rightarrow \delta \end{aligned}$$

Sola Öteleme \vee -Kuralı:

$$\begin{aligned} & \beta \rightarrow \gamma \vee \delta \\ & \Downarrow \\ & (\alpha \wedge \neg \gamma) \rightarrow \delta \end{aligned}$$

Sağa Öteleme \vee -Kuralı:

$$\begin{aligned} & (\beta \wedge \neg \gamma) \rightarrow \delta \\ & \Downarrow \\ & \beta \rightarrow \gamma \vee \delta \end{aligned}$$

| <u>Kuralın Adı</u> | <u>Formül</u> |
|------------------------------|--|
| Sola Öteleme \Box -Kuralı: | $\gamma \rightarrow \Box \delta$ \Downarrow $\Diamond^{-1} \gamma \rightarrow \delta$ |
| Sağa Öteleme \Box -Kuralı: | $\Diamond^{-1} \gamma \rightarrow \delta$ \Downarrow $\gamma \rightarrow \Box \delta$ |
| \Diamond -Kuralı: | $j \rightarrow \Diamond \gamma$ \Downarrow $j \rightarrow \Diamond k, k \rightarrow j$ |

burada j keyfi bir nominal ve k yeni bir nominaldir. Kısaltma olarak $j \rightarrow \Diamond k$ yerine Rjk kullanılabilir.

II. Ackermann Kuralı

Bu kural Ackermann lemmasındaki denkleğe dayanmaktadır. Ackermann Kuralı sadece tek bir denklem üzerinde çalışmaz, aşağıdaki gibi dönüşüm kuralları ile elde edilen yeni denklem kümeleri üzerinde de çalışır.

$$\left\| \begin{array}{l} \alpha_1 \rightarrow p, \\ \dots \\ \alpha_n \rightarrow p, \\ \beta_1(p), \\ \dots \\ \beta_m(p), \end{array} \right\| \Rightarrow \left\| \begin{array}{l} \beta_1[(\alpha_1 \vee \dots \vee \alpha_n)/p], \\ \dots \\ \beta_m[(\alpha_1 \vee \dots \vee \alpha_n)/p]. \end{array} \right\|$$

Burada;

- 1) $p, \alpha_1, \dots, \alpha_n$ lerde geçmeyen bir önermesel değişkendir.
- 2) β_1, \dots, β_n lerin her biri p de negatiftir.
- 3) p yi içeren sistemde başka bir denklem yoktur.

Bundan sonra Ackermann kuralının uygulanışında $\alpha_1, \dots, \alpha_n$ ve β_1, \dots, β_n formülleri sırasıyla α - formül ve β - formül olarak adlandırılacaktır.

III. Kutup Değişirme Kuralı

O anki sistemin içinden seçilen p değişkeninin her geçişinde kutup değiştirilir, yani p yerine $\neg p$ ve $\neg p$ yerine p yazılır.

IV. Yardımcı Kurallar

Bu kurallar bazı önermesel sonuçların kapasitelerini ve modal operatörler arasındaki dualitenin etkisini arttırmaya yöneliktir.

- 1) \wedge ve \vee bağlaçlarının dağılma ve birleşme özelliği.
- 2) $\gamma \vee \neg \gamma \equiv \top$, $\gamma \wedge \neg \gamma \equiv \perp$
- 3) $\gamma \vee \top \equiv \top$, $\gamma \vee \perp \equiv \gamma$
- 4) $\gamma \wedge \top \equiv \gamma$, $\gamma \wedge \perp \equiv \perp$
- 5) $\gamma \rightarrow \perp \equiv \neg \gamma$, $\gamma \rightarrow \top \equiv \top$
- 6) $\perp \rightarrow \gamma \equiv \top$, $\top \rightarrow \gamma \equiv \gamma$
- 7) $\neg \diamond \neg \equiv \square$, $\neg \square \neg \equiv \diamond$

Kutup değiştirme kuralının dışında önermesel değişkenin keyfi geçişinin kutupsallığını değiştiren dönüşüm kuralı yoktur.

5.3 Örnekler

Örnek 5.3.1 $\diamond \Box p \rightarrow \Box \diamond p$ formülünü göz önüne alınsın.

Adım 1 Formülün değillemesi alınır.

$$\diamond \Box p \wedge \Box \diamond \neg p$$

Adım 2 Denklemin başlangıç sistemi:

$$\| i \rightarrow (\diamond \Box p \wedge \Box \diamond \neg p)$$

Adım 3 Formül, p de ne negatif ne de pozitiftir. Formül yalnızca önermesel değişkenlerden oluşmaktadır.

Adım 4 Elemek için önerme değişkeni p seçilir.

Adım 5 Sistem dönüşüm kurallarını kullanarak Ackermann kuralına uygun hale dönüştürülür.

\wedge - Kuralı uygulanırsa:

$$\| \begin{array}{l} i \rightarrow \diamond p \\ i \rightarrow \Box \diamond \neg p \end{array}$$

İlk olarak \diamond -kuralı ardından \Box -kuralını uygulanırsa sırasıyla aşağıdaki iki denklemi elde edilir:

$$\| \begin{array}{l} Rij \\ j \rightarrow \Box p \\ i \rightarrow \Box \diamond \neg p \end{array}$$

$$\| \begin{array}{l} Rij \\ \diamond^{-1} j \rightarrow p \\ i \rightarrow \Box \diamond \neg p \end{array}$$

Artık sisteme Ackermann lemması uygulanabilir.

$$\parallel i \rightarrow \diamond \Box \neg (\diamond^{-1} j)$$

Tüm önermesel değişkenleri başarıyla sistemden elendi, artık Adım 6 ya geçilebilir.

Adım 6 Denklemlerin evetlemesi alınır

$$Rij \wedge (i \rightarrow \diamond \Box \neg (\diamond^{-1} j))$$

ve deęillemesi alınır

$$Rij \rightarrow (i \wedge \Box \Box \diamond^{-1} j)$$

elde edilir. Denklemin çevirisini yapılırsa

$$\forall y_j \exists x_0 [Ry_i y_j \rightarrow (x_0 = y_i) \wedge \forall y (Rx_0 y \rightarrow \exists u (Ryu \wedge \exists v (Rvu \wedge v = y_j)))]$$

elde edilir. Sadeleştirilirse

$$\forall y_j [Ry_i y_j \rightarrow \forall y (Ry_i y \rightarrow \exists u (Ryu \wedge Ry_j u))]$$

Church-Rosser özellięi elde edilir.

y_j deęişkeni serbest ve i nominaline uygundur öyle ki o anki durum olarak yorumlanır. Bundan dolayı yerel özellik gösterir. Ayrıca Rij yerine $Ry_i y_j$ yi kullanarak direkt olarak standart çevirisi yapılırsa ařaęıdaki denklik elde edilir.

$$ST(i \rightarrow \diamond j, x_0) \equiv x_0 = y_i \rightarrow \exists z (Rx_0 z \wedge z = y_i)$$

Örnek 5.3.2 $p \wedge \Box (\diamond p \rightarrow \Box q) \rightarrow \diamond \Box \Box p$ formülünü göz önünde bulundurulsun. Verilen formül bir Sahlqvist formülüne denk deęildir.

Adım 1 Formülün deęillemesi alınırsa

$$p \wedge \Box [\Diamond p \rightarrow q] \wedge \Box \Diamond \Diamond \neg q]$$

elde edilir.

Adım 2 $i \rightarrow [p \wedge \Box (\Box \neg p \vee \Box q) \wedge \Box \Diamond \Diamond \neg q]$

Adım 3 Sistem p ya da q da ne pozitif ne de negatiftir.

Adım 4 Önermesel deęişken p yi elemek için seçilir.

Adım 5 \wedge - Kuralı iki kere uygulanırsa

$$\left\| \begin{array}{l} i \rightarrow p \\ i \rightarrow \Box (\Box \neg p \vee \Box q) \\ i \rightarrow \Box \Diamond \Diamond \neg q \end{array} \right.$$

elde edilir. Sistem artık Ackermann kurallının uygulanması için p izole edildi ve $i \rightarrow \Box (\Box \neg p \vee \Box q)$ formül p 'de negatiftir.

$$\left\| \begin{array}{l} i \rightarrow \Box (\Box \neg i \vee \Box q) \\ i \rightarrow \Box \Diamond \Diamond \neg q \end{array} \right.$$

Ackermann kuralının uygulanmasından sonra sistemden p önerme deęişkeni elenmiştir. Sistemdeki dięer önerme deęişkeni q ' yu elemek için Adım 4 e geçilir.

Adım 4 Önerme deęişkeni q elemek için seçilir.

Adım 5 \Box -kuralı, Sola öteleme \forall -kuralı ve tekrar \Box -kuralını uygulanırsa

$$\begin{aligned} & \parallel \Box^{-1} i \rightarrow \Box \neg i \vee \Box q \\ & \parallel i \rightarrow \Box \Box \neg q \\ & \parallel \Box^{-1} i \vee \neg \Box \neg i \rightarrow \Box q \\ & \parallel i \rightarrow \Box \Box \neg q \\ & \parallel \Box^{-1} (\Box^{-1} i \vee \neg \Box \neg i) \rightarrow q \\ & \parallel i \rightarrow \Box \Box \neg q \\ & \parallel \Box^{-1} (\Box^{-1} i \vee \Box i) \rightarrow q \\ & \parallel i \rightarrow \Box \Box \neg q \end{aligned}$$

bulunur.

Adım 6 $i \rightarrow \Box \Box [\Box^{-1} (\Box^{-1} \neg i \vee \Box \neg i)]$ değillemesi alınır
 $i \wedge \Box \Box [\Box^{-1} (\Box^{-1} i \vee \Box i)]$

elde edilir. Son olarak formülünün çevirisi yapılır

$$\begin{aligned} & \exists x_0 [x_0 = y_i \wedge \exists z_1 (R x_0 z_1 \wedge \forall z_2 (R z_1 z_2 \rightarrow \forall z_3 (R z_2 z_3 \rightarrow \\ & \exists u_1 [R u_1 z_3 \wedge \exists u_2 (R u_2 u_1 \wedge u_2 = y_i \wedge \exists u_3 (R u_1 u_3 \wedge u_3 = y_i))])))] \end{aligned}$$

elde edilir.

Bu örnekte önermesel değişkenlerin elenme sırası önemli değildir. Önce q önerme değişkeni ardından p önerme değişkeni elenseydi algoritma aynı şekilde çalışacaktı.

Örnek 5.3.3 $\Box(\Box p \leftrightarrow q) \rightarrow p$ formülü verilsin.

Adım 1 $\Box((\Box \neg p \vee q) \wedge (\neg q \vee p)) \wedge \neg p$

Adım 2 $\parallel i \rightarrow \Box((\Box \neg p \vee q) \wedge (\neg q \vee p)) \wedge \neg p$

Adım 3 Bütün değişkenler pozitif ve negatif olarak geçiyor. Adım 4 e geçilebilir.

Adım 4 Önermesel değişken p elenmek için seçilir.

Adım 5 \wedge -kuralı ve \Box -kuralı uygulanırsa

$$\left\| \begin{array}{l} i \rightarrow \Box((\Diamond \neg p \vee q) \wedge (\neg p \vee p)) \\ \quad \Diamond^{-1} i \rightarrow (\neg q \vee p) \\ \quad \quad i \rightarrow \neg p \end{array} \right.$$

elde edilir. Sola öteleme \vee -kuralı ilk denkleme uygulanırsa

$$\left\| \begin{array}{l} (\Diamond^{-1} i \wedge p) \rightarrow q \\ \Diamond^{-1} i \rightarrow (\neg q \vee p) \\ \quad \quad i \rightarrow \neg p \end{array} \right.$$

elde edilir. Sistemde q ya göre Ackermann kuralı uygulanırsa

$$\left\| \begin{array}{l} \Diamond^{-1} i \rightarrow (\neg \Diamond^{-1} i \vee \neg p \vee p) \\ \quad \quad i \rightarrow \neg p \end{array} \right.$$

elde edilir. Birinci denklem totoloji olduğu için sistemden silinir.

$$\| i \rightarrow \neg p$$

p yerine \perp yerleştirilirse sonuç sistemi olarak

$$\| \top$$

elde edilir.

Adım 6 Değili alınırsa \perp elde edilir.

Gözlem 1 Algoritmanın başarısı bazı önermesel sonuçların temel yeteneklerine bağlıdır.

Özellikle $\diamond^{-1} i \rightarrow (\neg \diamond^{-1} i \vee \neg \Box p \vee \Box p)$ denkleminde $\neg \Box p$ yerine $\diamond \neg p$ yazılmış olsaydı, elde edilen denklem $\diamond^{-1} i \rightarrow (\neg \diamond^{-1} i \vee \diamond \neg p \vee \Box p)$ olurdu. Bu durumda totoloji durumu kolayca fark edilmeyebilirdi.

Bu durum p nin kutbu değiştirildikten sonra monotonluk temelli Ackermann kuralının uygulanmasına izin verir.

$$\| \begin{array}{l} \diamond^{-1} i \rightarrow (\neg \diamond^{-1} i \vee \diamond p \vee \Box \neg p) \\ i \rightarrow p \end{array}$$

Ackermann kuralı uygulanırsa aşağıdaki denklem elde edilir:

$$\| \diamond^{-1} i \rightarrow (\neg \diamond^{-1} i \vee \diamond i \vee \Box \neg i)$$

Denklem sadeleştirilirse

$$\| \diamond^{-1} i \rightarrow (\Box^{-1} \neg i \vee \diamond i \vee \neg \diamond i)$$

elde edilir. Bu denklemin birinci mertebeden dengi deęilmeden sonra \perp dir.

Gözlem 2 Sistemden ilk olarak p elenseydi,

$$\| \diamond^{-1} i \rightarrow (\diamond p \vee q)$$

elde edilirdi.

Formülünde \diamond altında p nin geçişlerini elde etmemek için p nin kutupsallığı deęiştirilirdi. Sistem dönüştürülmeye başlandığında

$$\left\| \begin{array}{l} \diamond^{-1} i \rightarrow (\diamond \neg p \vee q) \\ \diamond^{-1} (\diamond^{-1} i \wedge q) \rightarrow p \\ i \rightarrow \neg p \end{array} \right.$$

elde edilir. Ardından sisteme Ackermann lemması uygulandığında

$$\left\| \begin{array}{l} \diamond^{-1} i \rightarrow (\diamond \neg (\diamond^{-1} (\diamond^{-1} i \wedge q)) \vee q) \\ i \rightarrow \neg (\diamond^{-1} (\diamond^{-1} i \wedge q)) \end{array} \right.$$

elde edilirdi. Artık bu sistemde SQEMA'nın takıldığını görmek değildir.

Dayanak Elemanın sırası önemli olmamasına rağmen algoritmanın geri dönüş seçeneğini içermesi teorikte adım sayısının hızla artmasına yol açar fakat uygun yol göstericiler ve ek kurallar yardımıyla elemanın doğru sıralamasına karar verilir. Bu nedenle uygulamada algoritmanın adım sayısında hızlı bir artış meydana gelmez.

5.4 SQEMA'nın Sahlqvist Formüller Üzerindeki Tamlığı

Lemma 5.4.1 φ bir Sahlqvist formül ve $\varphi', \neg\varphi$ den tüm bağlaçlar üzerindeki deęillerin içeri aktarılmasıyla elde edilmiş bir formül ise φ' bir Sahlqvist öncülüdür.

Kanıt φ nin uzunluğu üzerinde tümevarım uygulayalım.

Eđer $\varphi: \alpha \rightarrow Pos$ bir Sahlqvist gerektirme ise deęili alınır ve yeniden yazılırsa $\alpha \wedge \neg Pos$ elde edilir ve Sahlqvist öncülüne dönüşür.

Eđer $\varphi = \Box\psi$ formunda bir formül ise deęil alınarak $\neg\varphi = \Diamond\neg\psi$ dönüştürülür. Böylece φ için iddia gösterilmiş olur. Çünkü Sahlqvist öncülü, diamond operatörü üzerinde kapalıdır.

Eđer $\varphi = \psi_1 \wedge \psi_2$ formunda ise deęili alınarak $\neg\varphi = \neg\psi_1 \vee \neg\psi_2$ elde edilir. Böylece φ için iddia sağlanmış olur. Çünkü Sahlqvist öncülleri veyalamalar üzerinde kapalıdır.

$\varphi = \psi_1 \vee \psi_2$ durumu da benzer şekilde gösterilebilir.

Sonuç Teorem 5.4.2 Her Sahlqvist formül, bir değil Sahlqvist öncüle ve bir Sahlqvist gerektirmeye semantik denktir.

Lemma 5.4.3 $E, j \rightarrow \beta$ formulu SQEMA denklemlerinin bir sistemi ve β , veya-lamalar kullanmadan oluşturulmuş, olası negatif formülleri kabul eden bir Sahlqvist öncülü olsun. p, E de negatif ve pozitif formüllerin geçtiği her hangi bir önermesel değişken ise E , yalnızca \wedge -kuralı, \vee -kuralı, \Box -kuralı ve Ackermann kuralı kullanılarak $j \rightarrow \beta$ formulu fakat p yi içermeyen bir E' sistemi içine dönüştürülebilir.

Kanıt p nin tüm pozitif geçişleri, evetlemelerin ve diamond operatörlerinin etkisi altında kalan box-lı formüllerin içindedir. İlk olarak p yi ayırılım başka bir ifadeyle E sistemini p nin pozitif olduğu $\gamma \rightarrow p$ formunda ki denklemlere dönüştürelim. Burada p, γ de meydana gelmesin. p nin box uygulanmış atomlarını \Diamond -kuralı ve \wedge -kuralı uygulayarak ayırılım. Elde edilen sistemdeki denklemler hala $j \rightarrow \beta$ formundadır. Burada j , bir nominal ve β bir Sahlqvist öncülüdür. Formül bir pure formül olarak adlandırılır ve hem pozitif hem de negatif olarak kabul edilir. Oluşan tüm denklemler p de pozitif meydana gelir ve $j \rightarrow \Box^n p$ formundadır. \Box -kuralı uygulanırsa, $(\Diamond^{-1})^n j \rightarrow p$ forma dönüştürülür. Sistem, denklemleri Sahlqvist öncülleri durumuna getiremez ve tüm denklemlere Ackermann kuralı uygulanarak p elenir. Saf formüller p nin negatif geçişlerine ikame edilir, böylece ikameden sonra negatif formül ve Sahlqvist öncülleri kalır.

Teorem 5.4.4 SQEMA, her Sahlqvist formülün birinci mertebeden mantıksal dengini hesaplar.

Kanıt φ bir Sahlqvist formül olsun. Birinci adımda φ nin değilini alalım ve tüm bağlaçlar üzerine dağıtalım. Elde ettiğimiz bu formülü φ' olarak adlandıralım. Lemma 1 den φ' formülünün bir Sahlqvist öncülü olduğunu

biliyoruz. Artık φ' 'yi , $\bigvee_{j=1}^n \alpha_j$ forma dönüştürebiliriz. Burada her bir α_j Sahlqvist öncülü, \top, \perp, \square -lı atomlar ve veya-lamaların kullanılmadığı negatif formüllerden, evetlemelerin ve \diamond -ların veya-lamalar üzerine dağıtılmasıyla elde edilen bir formüldür. Olası bazı negatif formüller dışındaki tüm veya-lamaları dağıtmak mümkündür. Çünkü φ' de bu veya-lamaların hiçbiri, box-ların etkisi altında kalmaz.

Artık algoritma her bir veya-lama üzerinde ayrı olarak işler. Bunlardan biri olan α_j yi seçerek ispata devam edelim. p_1, \dots, p_n ler α_j de meydana gelen değişkenlerin keyfi sıralaması olsun. Denklemlerin başlangıç sistemi

$$\|i \rightarrow \alpha_j$$

dir.

Bu sistem Lemma 5.4.3 ün gerektirdiği formdadır. Eğer p_1 yalnızca pozitif (negatif) ise SQEMA bunun pozitif (negatif) formüllerin yerine $\top(\perp)$ ikame ederek p_1 i eler. Eğer p_1 hem pozitif hem de negatif ise Lemma 5.4.3 den SQEMA, p_1 i eler. Önermesel değişkenler p_2, \dots, p_n benzer şekilde elenir. İşlem tümevarımsal olarak her bir elemenden sonra kalanlara Lemma 5.4.3 ün uygulanmasıyla devam eder. SQEMA tarafından değillemeler ve dönüşüm kurallarıyla elde edilen saf formül, α_j ye yerel birinci mertebeden çatıya denktir. $1 \leq j \leq n$ için bu denklemlerin evetlemeleri alınırsa girdi Sahlqvist formülü φ için birinci mertebeden yerel çatıya denk olan formül elde edilir.

6. SONUÇ

Sahlqvist tekniđi, konu ile bađlantılı ilk alıřma deđildir. Gerekten, Jonsson-Tarski (Jonsson et al., 1952) yansıyan ve geiřken atıların birinci merteye dengini rneklemişlerdir. Fitch (Fitch, 1973) in yazdığı bir makale van Benthem'i – Sahlqvist' den habersiz- bugün Sahlqvist Teoremi sonucunu ispatlamaya yöneltmiştir. Sahlqvist tekniđi normal modal formüllerin yanı sıra zaman lojiđinin formülleri üzerinde de etkili bir algoritmadır.

SCAN algoritması konu ile ilgili ilk saf algoritmalarından biridir. SCAN algoritması modal formüllere karşılık gelen birinci ya da ikinci mertebeden formülleri hesaplayabilmektedir. Geliştirilen diđer algoritmalar ve tekniklerde olduđu gibi SCAN algoritması modal formüllerin sınıfına göre tam deđildir. Yani verilen her modal formüle karşılık gelen birinci ya da ikinci mertebeden formülü hesaplayamamaktadır. Algoritma ikinci mertebeden niceleyicilerin elenmesi aşamasında ters skolemizasyon tekniđinin doğası geređi sorunlarla karşılaşmaktadır. Sahlqvist tekniđinde olduđu gibi SCAN algoritması da zaman lojiđinin formülleri üzerinde etkilidir.

SQEMA algoritması içerdiği dönüşüm kuralları nedeniyle modal formüller üzerinde direkt olarak çalışabilmektedir. Bu özelliđi sayesinde diđer algoritmalarından daha yalındır. SQEMA algoritmasını diđer algoritmalar ve tekniklerden ayıran bir başka özellik ise normal modal lojiđin formülleriyle birlikte zaman lojiđi, hibrid lojik ve polyadic modal formüller üzerinde de etkili bir algoritma olmasıdır. SCAN algoritmasında olduđu gibi SQEMA algoritmasında modal formüller sınıfına göre tam deđildir. SCAN ve SQEMA algoritmaları modal formüllere karşılık gelen ikinci mertebeden formülleri hesaplayabilmeleri ve algoritmaların Sahlqvist formüllerine göre tam olması Sahlqvist tekniđinden ok daha etkili olduklarının kanıtıdır. SCAN ve SQEMA algoritmalarının uygulamalarına

<http://www.mpi-inf.mpg.de/departments/rg1/software/scan/index.html>

ve

<http://www.fmi.uni-sofia.bg/fmi/logic/sqema/sqema.jsp>

internet adreslerinden ulaşmak mümkündür.

KAYNAKLAR DİZİNİ

Rubin J. E. , 1990, Mathematical Logic: Applications and Theory, Saunders College Publishing.

Blackburn, P. de Rijke, M. , Venema, Y. , 2001, Modal Logic, Cambridge University Press.

Burris, S. N. , 1998, Logic for Mathematics and Computer Science, Prentice Hall.

Chellas, B. F. , 1980, Modal Logic: An Introduction, Cambridge University Press.

Condradie, W. Goranko V. , Vakarelov D. , 2006, Algorithmic Correspondence and Completeness in Modal Logic. I. Core Algorithm SQEMA, Logical Methods in Computer Science 2 (1;4) 1–26pp.

Condradie, W. , Goranko V. , Vakarelov D. , 2006, Algorithmic Correspondence and Completeness in Modal Logic. II. Polyadic and Hybrid Extensions of the Algorithm SQEMA, Journal of Logic and Computation Advance Access.

Darılmaz, G. , 2007, Sahlqvist Formülleri ve Tamlık, Ege Üniversitesi, Fen Bilimleri Enst., Yüksek Lisans Tezi, 29-51s.

Gabbay, D. , Ohlbach H. J. , 1992, Quantifier Elimination in Second-Order Predicate Logic, South African Computer Journal, 7: 35–43pp.

Goranko V. , Vakarelov D. , 2002, Sahlqvist formulas Unleashed in Polyadic Modal Languages, Advances in Modal Logic, 3.

Hustadt, U. , Goranko, V. , Vakarelov, D. , 2004, SCAN is compeler for all Sahlqvist formulae, In Relational and Kleene-Algebraic Methods in Computer Science.

KAYNAKLAR DİZİNİ (devam)

Nonnengart, N. , Ohlbach, H. J. , Szalas, A. , 1999, Elimination of Predicate Quantifiers. Logic and Reasoning, 159–181pp.

Sahlqvist, H. , 1973, Completeness and correspondence in the first and second order semantics for modal logic in Kranger, 110–143pp.

Szalas, A. , 1993, On the Correspondence Between Modal and Classical Logic: an Automated Approach, Journal of Logic and Computation, 605–620pp.

Vaananen, J. , 2001, Second-Order Logic and Foundation of Mathematics, The Bulletin of Symbolic Logic, 7 – 4.