OKU Fen Bilimleri Enstitüsü Dergisi
5(1):401-416, 2022

OKU Journal of The Institute of Science and Technology 5(1):401-416, 2022

Osmaniye Korkut Ata Üniversitesi
Fen Bilimleri Enstitüsü
Dergisi

Osmaniye Korkut Ata University
Journal of The Institute of Science
and Technology

Modal Mantık için Algoritmik Tekabül

Zafer ÖZDEMİR[1*]

[1]Nişantaşı Üniversitesi, İktisadi, İdari ve Sosyal Bilimler Fakültesi, Yönetim Bilişim Sistemleri Bölümü, 34398, İstanbul

[1]https://orcid.org/0000-0001-7090-373X
*Sorumlu yazar: zafer.ozdemir@nisantasi.edu.tr

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 11.08.2021
Kabul tarihi: 09.11.2021
Online Yayınlanma:08.03.2022

Anahtar Kelimeler:
Modal mantık
Scan algoritması
Sqema algoritması

ÖZ

Modal mantık formülleri Kripke çatılar üzerinde ikinci mertebeden özellikler ifade etmektedir. Pek çok durumda modal mantık formüllerine karşılık gelen birinci mertebeden mantık formülleri etkili algoritmalar yardımı ile hesaplanmaktadır. Bu alandaki ilk araştırma makalesi, 1975 yılında H. Sahlqvist tarafından yazılan "Modal mantık için birinci ve ikinci dereceden semantikler için tekabül ve tamlık" idi. Yaptığı çalışmada modal mantık formüllerinin belirli bir sınıfını tanımlayarak, bu sınıfın çatılar üzerinde birinci mertebeden koşullar tanımladığını ve bu koşulların da geliştirdiği tekniği yardımı ile modal mantık formüllerine tekabül eden birinci mertebeden formülleri hesaplamıştır. Ancak bir modal mantık formülüne karşılık gelen birinci mertebeden mantık formülü her zaman bulunmayabilir. Bazı durumlarda bir modal mantık formülü ikinci mertebeden mantık formülüne tekabül edebilir. Bu tip durumlarda Sahlqvist tekniği etkinliğini kaybetmektedir. Literatürde bir modal mantık formülüne tekabül eden birinci ve ikinci mertebeden mantık formülünü hesaplamaya yarayan farklı algoritmalar ve teknikler geliştirilmiştir. Bu algoritmalar içinde öne çıkan iki çalışma bulunmaktadır. Bu algoritmalardan biri Gabbay ve Ohlbach (1992) tarafından geliştirilen, temeli kısıtlama çözümleme ve tekniğine dayanan SCAN algoritması diğeri ise Condradie ve ark. (2006) tarafından geliştirilen, modal formüller üzerinde direkt olarak çalışan SQEMA algoritmasıdır. Bu çalışmada SCAN ve SQEMA algoritmaları ayrıntılı olarak incelip, karşılaştırması yapılacaktır.

Algorithmic Correspondence for Modal Logic

Research Article

Article History:
Received: 11.08.2021
Accepted: 09.11.2021
Published online: 08.03.2022

Keywords:
Modal logic
Scan algortihm
Sqema algorithm

ABSTRACT

Modal logic formulas express second-order properties on Kripke frameworks. In many cases, first-order logic formulas corresponding to modal logic formulas are calculated with the help of efficient algorithms. The first research paper in this field was "Correspondence and completeness for first and second order semantics for modal logic" written by H. Sahlqvist in 1975. In his study, he defined a certain class of modal logic formulas, this class defines first-order conditions on frames, and with the help of the technique he developed for these conditions, he calculated first-order formulas corresponding to modal logic formulas. However, a first-order logic formula corresponding to a modal logic formula may not always be found. In some cases, a modal logic formula may correspond to a second-order logic formula. In such cases, the Sahlqvist technique loses its effectiveness. In the literature, different algorithms and techniques have been developed to calculate the first and second order logic formula corresponding to a modal logic formula. There are two prominent studies among these algorithms. One of these algorithms is the SCAN algorithm developed by Ohlbach and Gabbay, which is based on constraint analysis and technique. The other is the SQEMA algorithm developed by Condradie et al., (2006), which works directly on modal formulas. In this study, SCAN and SQEMA algorithms will be examined in detail and compared.

To Cite: Özdemir Z. Modal Mantık için Algoritmik Tekabül. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2022; 5(1):401-416.

Giriş

Modal mantık, çok basit bir ifadeyle, birinci mertebeden mantığa bir ya da daha fazla operatör eklenerek oluşturulan mantık olarak tanımlanabilir (Chellas, 1980). Birinci mertebeden mantığın dili ifade gücünün yeterli olmaması nedeni ile kuvvetli matematiksel yapıların oluşturulmasına engel olur. Modal mantık birinci mertebeden mantıkla karşılaştırıldığında sahip olduğu operatörler sayesinde daha etkili bir ifade gücüne sahiptir (Blackburn ve ark., 2001). Modal mantıkta birinci mertebeden tanımlanabilirlik ve tamlığın en genel sonuçlarından biri Sahlqvist teoremidir. Sahlqvist, 1975 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. Bazı durumlarda bir modal mantık formülü ikinci mertebeden mantık formülüne karşılık gelebilir (Vaananen, 2001). Bu tip durumlarda Sahlqvist tekniği etkinliğini kaybetmektedir. Bu durum daha etkili tekniklerin geliştirilmesine sebep olmuştur. Bir modal formüle karşılık gelen birinci mertebeden formül hesaplanırken evrensel monadik ikinci mertebeden niceleyicilerin elenmesi o formülün bir çatıda geçerliliğini, varlıksal monadik niceleyicilerin elenmesi formülün sağlanabilirliğini ifade eder. Varlıksal monadik ikinci mertebeden niceleyicilerin elenmesi yöntemi ile ilgili en bilinen algoritma SCAN algoritmasıdır. Gabbay ve Ohlbach (1992) tarafından geliştirilen SCAN algoritmasının ana fikri Ackermann'ın (1972) makalesinde görünmektedir. SCAN algoritmasının geliştirilmesi katkı da bulunan bir diğer isim Szalas yaptığı çalışmalarda birinci mertebeden aksiyomlara karşılık gelen modal aksiyomlar üzerinde çalışmıştır (1993, 1999). Bir diğer teknik ise SQEMA algoritmasıdır. SQEMA ikinci mertebeden niceleyicilerin elenmesine yönelik geliştirilen bir algoritmadır. SQEMA algoritması Ackermann lemmasının modal versiyonunu kullanarak diğer algoritmaların aksine modal formüller üzerinde direkt olarak çalışır. Bu nedenden dolayı diğer algoritmalardan çok daha etkili ve yalın bir algoritmadır. Bu çalışmada SCAN ve SQEMA algoritmaları ayrıntılı olarak incelenip, bu algoritmaların Sahlqvist formüllerine göre tamlığı ispatlanacak ve algoritmaların karşılaştırması yapılacaktır.

Bu bölümde SCAN algoritmasında kullanılan temel tanımlar ve kavramlar verilecek. (Hustadt ve ark., 2004) SCAN algoritmasının adımları ayrıntılı bir şekilde sunularak, algoritmanın etkililiği örneklerle açılanacaktır. Bölüm sonunda algoritmanın sınırları verilecek ve SCAN algoritmasının Sahlqvist formüllerine göre tamlığı kanıtlanacaktır.

$L_1, \dots, L_n$ önerme değişkenleri olmak üzere $\{L_1, \dots, L_n\}$ sonlu kümesine *cümle* denir. C ve D cümle ve P bir önerme değişkeni olmak üzere çözünme *kuralı*

$$\frac{\begin{array}{c}P(s_1,\dots,s_n)\vee C\\ \neg P(t_1,\dots,t_n)\vee D\end{array}}{C\vee D\vee s_1\neq t_1\vee\dots\vee s_n\neq t_n}$$

dir. Burada iki cümle P üzerinden silinmiştir ve $C\vee D$ *çözücü* olarak adlandırılır. C bir cümle ve P bir önerme değişken olmak üzere *faktorizasyon kuralı*

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

dir. Faktorizasyon kuralı yalnızca farklı cümleler arasında yapılabilir.

Prenex formdaki birinci mertebeden bir formüldeki varlıksal niceleyiciler sistematik olarak elenerek yerlerine yeni sabit sembolleri ve fonksiyon sembolleri atanması işlemine *Skolemleştirme* adı verilir. Yeni sabit sembollerine ve fonksiyon sembollerine sırasıyla *Skolem sabitleri* ve *Skolem fonksiyonları* adı verilir.$\exists x\forall y\forall zA$birinci mertebeden mantığın bir formülü ve c Skolem sabiti olmak üzere birinci mertebeden formülün Skolemizasyon sonucu$\forall y\forall zA[c/x]$dir. $\forall y\exists zP(y,z)$ birinci mertebeden mantığın bir formülü ve f Skolem fonksiyonu olmak üzere formülün Skolemizasyon sonucu $\forall yP(y,f(y))$ dir.

Tanım (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 mantığın bir formüldür.

SCAN algoritmasında çıktı: $P_1,\dots,P_n$ yüklem değişkenlerini 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 *Skolemleştirme* kuralı uygulanarak cümle formuna dönüştürülür. İşlem sonucunda f_i ler Skolem fonksiyonları, ψ' atomların veya onların değillemelerinin sonlu 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 cümle olmak üzere çözümleme ve faktorizasyon kuralları uygulanarak cümleler silinir. Eğer tüm cümleler silinmiş ise bunun anlamı çelişkidir.

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

Bir örnekle SCAN algoritmasının adımlarını detaylı olarak gösterelim. Algoritmada girdi olarak

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

ele alalım formülünü alalım. Birinci adımda ikinci mertebeden skolemleştirme kuralı kullanılarak cümle formları hesaplanır.

f bir skolem fonksiyonu, ikinci mertebeden değişkenler ve niceleyicileri $\exists P, \exists f \forall x, y$ olmak üzere elde edilen cümleler 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 çözümlemeye 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 cümlesi silindi.

Geriye kalan cümleler:

$$C_2 \quad P(y), Q(a)$$
$$C_3 \quad P(f(x,y))$$
$$C_4 \quad Q(a)$$
$$C_5 \quad a \neq f(x,y), Q(x)$$

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

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

formülü elde edilir.

Bir örnekle SCAN algoritmasının farklı adımlarını detaylı olarak gösterelim. Algoritmada girdi olarak

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

formülünü alalım.

Birinci adımda ikinci mertebeden skolemizasyon kuralı kullanılarak atomların veya onların değillemelerinin sonlu kümesi hesaplanır.

f bir skolem fonksiyonu, ikinci mertebeden değişkenler ve niceleyicileri $\exists P, \exists f \forall x, y$ olmak üzere elde edilen atomların veya onların değillemelerinin sonlu kümesi 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 çözümlemeye 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 satırı silindi.

Geriye kalan satırlar:

$$C_2 \quad P(y), Q(a)$$
$$C_3 \quad P(f(x, y))$$
$$C_4 \quad Q(a)$$
$$C_5 \quad a \neq f(x, y), Q(x)$$

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.

SCAN Algoritmasının Sınırları

Modal mantık 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 ∀P($\square \lozenge P \longrightarrow \lozenge \square P$) McKinsey aksiyomunun denk olduğu formül yalnızca ulaşılabilirlik bağıntısının ikinci mertebeden özelliği kullanılarak bulunabilir (SCAN algoritması ters Skolleştirme sırasında ikinci mertebeden Henkin niceleyicilerine ihtiyaç duyar). Geçişme aksiyomu ile birleştirilirse, bu iki tanım atomik olarak $\forall x \exists y (Rxy) \wedge \forall z Ryz \longrightarrow z = y)$ denktir. Açıkca görülmektedir ki bu birinci mertebeden tanımlanabilen bir özelliktir. 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 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 veyalaması 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.

Shalqvist 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 çözücüler ve C faktörlerinin hesaplanması ψ ile ilgili cümlelerin kümesi $Cls(\psi)$ uygulandığında sonlanır. Örneğin SCAN, işlemde yalnızca sonlu sayıda yeni cümle olduğunda çalışır.
2. Sonuçta oluşan birinci mertebeden formüle (genel olarak Skolem fonksiyonlarını içerir) başarılı bir şekilde ters Skolemleştirme işlemi yapılmış olur.

Her Sahlqvist formül temel Sahlqvist formülün $\vee$ 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 atomların veya onların değillemelerinin sonlu bir kümesidir; $(\neg)R_{k_i}st$, $(\neg)Q_j(u)$ 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.

Teorem. SCAN**,** keyfi bir Sahlqvist gerektirmesi φ nin birinci mertebeden mantıksal denk olduğu formül α_φ yi hesaplar.

Kanıt N, $ST(\neg\varphi, a)$ nın atomların veya onların değillemelerinin sonlu bir kümesi olsun. Tüm atomların veya onların değillemelerinin sonlu kümesi N den türetilebilir öyle ki N, N nin kelimeleri üzerindeki zincirlerden meydana gelir. Zincirlerin uzunluğu, N deki zincirlerin uzunluğu tarafından sınırlandırılır.

Göstermeliyiz ki her ne zaman bir cümle çok sayıda zincir ile türetilirse bu cümle yoğunlaşır.

Ters Skolemleştirme durumunda tümevarımsal argümanlar kullanılarak eşitsizlikler ve terimlerde meydana gelen türetilmiş cümlelerin başarılı bir şekilde unskolemize edilebildiği 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.

SQEMA

SQEMA ikinci mertebeden niceleyicilerin elenmesine yönelik geliştirilen en son algoritmadır. SQEMA algoritması Ackermann lemmasının modal versiyonunu kullanarak diğer algoritmaların aksine modal formüller üzerinde direkt olarak çalışır. Bu nedenden dolayı diğer algoritmalardan 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 bir örnekle açıklanacaktır. Bölümün sonunda ise SQEMA algoritmasının Sahlqvist formüllere göre tamlığı ispatlanacaktır.

Algoritmanın işleyişini güçlendirmek için aşağıdakiler modal dile eklenir:

- Ters modalite $\Box^{-1}$:

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

Ters modalite için standart çeviri:

$$ST(\Box^{-1}, \varphi) \coloneqq \forall y(Ryx \longrightarrow ST(\varphi, y))$$

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

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

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

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

Modal dilin bu genişlemesi ML^+ ile gösterilir. M bir model ve φ, ML^+ nın bir formülü olsun. $[\![\varphi]\!]_M$,bir M modelinde φ formülünün genişlemesidir ve $[\![\varphi]\!]_M = \{m \in M: M, m \Vdash \varphi\}$ biçiminde tanımlanır.

ML^+ da bir saf formül önermesel değişkenleri içermeyen fakat nominalleri içerebilen bir formüldür. Her saf formül γ, $\forall \bar{y} ST(\gamma, x)$ formülü tarafından yerel birinci mertebeden tanımlanabilirdir; burada $\bar{y}, \gamma$ 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 nın 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:

$$M, u \Vdash [U\,]\varphi \quad (eye) \ her \ w \in M \ için \ M, w \Vdash \varphi \ (eye) \ M \Vdash \varphi$$

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

$$M, u \Vdash \langle U \rangle \varphi \ (eye) bazı \ w \in M \ için \ M, u \Vdash \varphi$$

$$(eye) M, u \Vdash \neg [U] \neg \varphi$$

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

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

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 $[\![A]\!]_M \subseteq [\![p]\!]_M$ dır.

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.

Tanım (SQEMA Algoritması)

Bu bölümde biçimsel olarak SQEMA algoritması temel modal dil için sunulacaktır. Fakat SQEMA, keyfi çok öğeli ve hibrit çoklu modal dillere genişletilebilir (Goranko ve Vakarelov, 2002).

Temel Algoritma

Bir modal formül φ girdi olarak verildiğinde değili alınarak değil normal forma çevirilir. Daha sonra yerel çatı denkliğini koruyan dönüşüm kuralları uygulanarak verilen formül $\alpha \rightarrow \beta$ formlu 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 uygulanabilirdir. 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 adımlar sırasıyla uygulanır:

Adım 1 φ nin değillemesi alınır ve değil normal formdaki formül →, ↔ bağlaçları ve içeride bulunan tüm değil işaretleri kaybolana kadar önermese değişkenlerinin önüne sürülür.

Elde edilen formülü $\bigvee \alpha_k$ formlu 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 ⊤ ve ⊥ 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 elemenin sırası değiştirilerek değişkenler tekrar elenir. Eğer elemenin 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 ulaşılırsa, 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 $\forall\overline{y}\exists x_0 ST(\neg pure, x_o)$ 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.

Mantıksal Bağlaçlar için Kurallar

Kuralın Adı	**Formül**
$\wedge$- **Kuralı :**	$\beta \to \gamma \wedge \delta$ $\Downarrow$ $\beta \to \gamma, \beta \to \delta$
Sola Öteleme $\vee$**-Kuralı:**	$\beta \to \gamma \vee \delta$ $\Downarrow$ $(\alpha \wedge \neg\gamma) \to \delta$
Sağa Öteleme $\vee$**-Kuralı:**	$(\beta \wedge \neg\gamma) \to \delta$ $\Downarrow$ $\beta \to \gamma \vee \delta$
Sola Öteleme $\square$**-Kuralı:**	$\gamma \to \square\delta$ $\Downarrow$ $\lozenge^{-1}\gamma \to \delta$
Sağa Öteleme $\square$**-Kuralı:**	$\lozenge^{-1}\gamma \to \delta$ $\Downarrow$ $\gamma \to \square\delta$
$\lozenge$**-Kuralı:**	$j \to \lozenge\gamma$ $\Downarrow$ $j \to \lozenge k, k \to j$

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

Ackermann Kuralı

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

$$\left\| \begin{array}{c} \alpha_1 \to p, \\ \ldots \\ \alpha_n \to p, \\ \\ \beta_1(p), \\ \ldots \\ \beta_m(p), \end{array} \right. \quad \Rightarrow \quad \left\| \begin{array}{c} \beta_1[(\alpha_1 \vee \ldots \vee \alpha_n)/p], \\ \ldots \\ \beta_m[(\alpha_1 \vee \ldots \vee \alpha_n)/p]. \end{array} \right.$$

Burada;

1) $p, \alpha_1, \ldots, \alpha_n$ lerde geçmeyen bir önermesel değişkendir.
2) $\beta_1, \ldots, \beta_n$ lerin her biri p de negatiftir.
3) p yi içeren sistemde başka bir denklem yoktur.

Bundan sonra Ackermann kuralının uygulanışında $\alpha_1, \ldots, \alpha_n$ ve $\beta_1, \ldots, \beta_n$ formülleri sırasıyla $\alpha - form$ül ve $\beta - form$ül olarak adlandırılacaktır.

Kutup Değiştirme Kuralı

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

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) $\gamma \vee \neg\gamma \equiv \top$, $\gamma \wedge \neg\gamma \equiv \bot$
3) $\gamma \vee \top \equiv \top$, $\gamma \vee \bot \equiv \gamma$
4) $\gamma \wedge \top \equiv \gamma$, $\gamma \wedge \bot \equiv \bot$
5) $\gamma \to \bot \equiv \neg\gamma$, $\gamma \to \top \equiv \top$
6) $\bot \to \gamma \equiv \top$, $\top \to \gamma \equiv \gamma$
7) $\neg \Diamond \neg \equiv \Box$, $\neg\Box\neg \equiv \Diamond$

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

Bir örnekle SQEMA algoritmasının adımlarını detaylı olarak gösterelim $p \wedge \Box(\Diamond p \longrightarrow \Box q) \longrightarrow \Diamond \Box\Box p$ formülünü göz önünde bulundurulsun. Verilen formül bir Sahlqvist formülüne denk değildir.

Adım 1 Formülün değillemesi alınırsa

$$p \wedge \Box[\Diamond p \longrightarrow q) \wedge \Box \Diamond\Diamond \neg q]$$

elde edilir.

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

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

Adım 4 Önermesel değişken p yi elemek için seçilir.

Adım 5 ∧- Kuralı iki kere uygulanırsa

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

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

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

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

Adım 4 Önerme değişkeni q elemek için seçilir.

Adım 5 □-kuralı, Sola öteleme ∨-kuralı ve tekrar □-kuralını uygulanırsa

$$\left\| \begin{array}{c} \Diamond^{-1} i \longrightarrow \Box\neg i \vee \Box q \\ i \longrightarrow \Box \Diamond\Diamond \neg q \end{array} \right.$$

$$\left\| \begin{array}{c} \Diamond^{-1} i \vee \neg\Box\neg i \longrightarrow \Box q \\ i \longrightarrow \Box \Diamond\Diamond \neg q \end{array} \right.$$

$$\left\| \begin{array}{c} \Diamond^{-1} (\Diamond^{-1} i \vee \neg \Box \neg i) \longrightarrow q \\ i \longrightarrow \Box \Diamond\Diamond \neg q \end{array} \right.$$

$$\left\| \begin{array}{c} \Diamond^{-1} (\Diamond^{-1} i \vee \Diamond i) \longrightarrow q \\ i \longrightarrow \Box \Diamond\Diamond \neg q \end{array} \right.$$

bulunur.

Adım 6 $i \longrightarrow \Box \Diamond\Diamond [\Box^{-1}(\Box^{-1} \neg i \vee \Box \neg i)]$ değillemesi alınırsa

$$i \wedge \Diamond \Box\Box[\Diamond^{-1}(\Diamond^{-1} i \vee \Diamond i)]$$

elde edilir. Son olarak formülünün çevirisi yapılırsa

$$\exists x_0[x_0 = y_i \wedge \exists z_1(Rx_0z_1 \wedge \forall z_2(Rz_1z_2 \longrightarrow \forall z_3(Rz_2z_3 \longrightarrow$$
$$\exists u_1[Ru_1z_3 \wedge \exists u_2(Ru_2u_1 \wedge u_2 = y_i \wedge \exists u_3(Ru_1u_3 \wedge u_3 = y_i)])))))]$$

elde edilir.

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

Gözlem Algoritmanın başarısı bazı önermesel sonuçların temel yeteneklerine bağlıdır. Özellikle $\Diamond^{-1} i \longrightarrow (\neg \Diamond^{-1} i \vee \neg \Box p \vee \Box p)$ denkleminde $\neg \Box p$ yerine $\Diamond \neg p$ yazılmış olsaydı, elde edilen denklem $\Diamond^{-1} i \longrightarrow (\neg \Diamond^{-1} i \vee \Diamond \neg p \vee \Box p)$ olurdu. Bu durumda totoloji durumu kolayca fark edilmeyebilirdi.

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

$$\left\| \begin{array}{c} \Diamond^{-1} i \longrightarrow (\neg \Diamond^{-1} i \vee \Diamond p \vee \Box \neg p) \\ i \longrightarrow p \end{array} \right.$$

Ackermann kuralı uygulanırsa aşağıdaki denklem elde edilir:

$$\| \Diamond^{-1} i \longrightarrow (\neg \Diamond^{-1} i \vee \Diamond i \vee \Box \neg i)$$

Denklem sadeleştirilirse

$$\| \Diamond^{-1} i \longrightarrow (\Box^{-1} \neg i \vee \Diamond i \vee \neg \Diamond i)$$

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

Gözlem Sistemden ilk olarak p elenseydi,

$$\| \Diamond^{-1} i \longrightarrow (\Diamond p \vee q)$$

elde edilirdi.

Formülünde $\Diamond$ altında p nin geçişlerini elde etmemek için p nin kutupsallığı değiştirilirdi. Sistem dönüştürülmeye başlandığında

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

elde edilir. Ardından sisteme Ackermann lemması uygulandığında

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

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

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

SQEMA' nın Sahlqvist Formüller Üzerindeki Tamlığı

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

Kanıt φ nin uzunluğu üzerinde tümevarım uygulayalım.

Eğer φ: $\alpha \longrightarrow Pos$ bir Sahlqvist gerektirme ise değili alınır ve yeniden yazılırsa $\alpha \wedge \neg Pos$ elde edilir ve Sahlqvist öncülüne dönüşür.

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

Eğer $\varphi = \psi_1 \wedge \psi_2$ formunda ise değili 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.

Lemma $E, j \longrightarrow \beta$ formlu SQEMA denklemlerinin bir sistemi ve β, veyalamalar kullanmadan oluşturulmuş, olası negatif formülleri kabul eden bir Sahlqvist öncülü olsun. p, E de negatif ve pozitif formüllerin geçtiği herhangi bir önermesel değişken ise E, yalnızca $\wedge$-kuralı, $\vee$-kuralı, $\Box$-kuralı ve Ackermann kuralı kullanılarak $j \longrightarrow \beta$ formlu 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 ayıralım başka bir ifadeyle E sistemini p nin pozitif olduğu $\gamma \longrightarrow p$ formundaki denklemlere dönüştürelim. Burada p, γ de meydana gelmesin. p nin box uygulanmış atomlarını $\Diamond$-kuralı ve $\wedge$-kuralı uygulayarak ayıralım. Elde edilen sistemdeki denklemler hala $j \longrightarrow \beta$ 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 $j \longrightarrow \Box^n p$ formundadır. $\Box$-kuralı uygulanırsa, $(\Diamond^{-1})^n j \longrightarrow p$ 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 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, $\top, \bot$, $\Box$-lı atomlar ve veyalamaların kullanılmadığı negatif formüllerden, evetlemelerin ve $\Diamond$-ların veyalamalar üzerine dağıtılmasıyla elde edilen bir formüldür. Olası bazı negatif formüller dışındaki tüm veyalamaları dağıtmak mümkündür. Çünkü φ' de bu veyalamaların hiçbiri, box-ların etkisi altında kalmaz.

Artık algoritma her bir veyalama 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

$$\| i \longrightarrow \alpha_j$$

dir.

Bu sistem Lemma 3.2.1 ün gerektirdiği formdadır. Eğer p_1 yalnızca pozitif (negatif) ise SQEMA bunun pozitif (negatif) formüllerin yerine $\top(\bot)$ 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 elemeden sonra kalanlara Lemma 3.2.1 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.

Sonuç

SCAN algoritması konu ile ilgili ilk saf algoritmalardan biridir. SCAN algoritması modal formüllere karşılık gelen birinci ya da ikinci mertebeden formülleri hesaplayabilmektedir. Geliştirilen diğer algoritmalar ve tekniklerde olduğu gibi SCAN algoritması modal formüllerin sınıfına göre tam değildir. Yani verilen her modal formüle karşılık gelen birinci ya da ikinci mertebeden formülü hesaplayamamaktadır. Algoritma ikinci mertebeden niceleyicilerin elenmesi aşamasında ters skolemleştirme tekniğinin doğası gereği sorunlarla karşılaşmaktadır. Sahlqvist tekniğinde olduğu gibi SCAN algoritması da zaman mantığının formülleri üzerinde etkilidir.

SQEMA algoritması içerdiği dönülüşüm kuralları nedeniyle modal formüller üzerinde direkt olarak çalışabilmektedir. Bu özelliği sayesinde diğer algoritmalardan daha yalındır. SQEMA algoritmasını diğer algoritmalar ve tekniklerden ayıran bir başka özellik ise normal modal mantığın formülleriyle birlikte zaman mantığı, hibrit mantık ve çok öğeli modal formüller üzerinde de etkili bir algoritma olmasıdır. SCAN algoritmasında olduğu gibi SQEMA algoritmasında modal formüller sınıfına göre tam değildir. SCAN ve SQEMA algoritmaları modal formüllere karşılık gelen ikinci mertebeden formülleri hesaplayabilmeleri ve algoritmaların Sahlqvist formüllerine göre tam olması Sahlqvist tekniğinden çok 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.

Çıkar Çatışması Beyanı

Makale yazarı herhangi bir çıkar çatışması olmadığını beyan eder.

Araştırmacıların Katkı Oranı Beyan Özeti

Yazar makaleye %100 oranında katkı sağlamış olduğunu beyan eder.

Kaynakça

Ackermann J. Der entwurf linearer regelungssysteme im zustandsraum. At Automatisierungstechnik 1972; 20(1-12): 297-300.

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

Chellas BF. Modal logic: An introduction. Cambridge University Press, 1980.

Condradie W., Goranko V., Vakarelov D. Algorithmic correspondence and completeness modal logic. I. core algorithm SQEMA. Logical Methods in Computer Science 2006; 1-26.

Condradie W., Goranko V., Vakarelov D. Algorithmic correspondence and completeness in modal logic. II. polyadic and hybrid extensions of the algorithm SQEMA. Journal of Logic and Computation Advance Access 2006.

Gabbay D., Ohlbach HJ. Quantifier elimination in second-order predicate logic. South African Computer Journal 1992; 7: 35-43.

Goranko V., Vakarelov D. Sahlqvist formulas unleashed in polyadic modal languages. Advances in Modal Logic 2002.

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

Nonnengart N., Ohlbach HJ., Szalas A. Elimination of predicate quantifiers. Logic and Reasoning 1999; 159-181.

Sahlqvist H. Completeness and correspondence in the first and second order semantics for modal logic. Studies in Logic and the Foundations of Mathematics 1975; 82: 110-143.

Szalas A. On the correspondence between modal and classical logic: an automated approach. Journal of Logic and Computation 1993; 605-620.

Vaananen J. Second-order logic and foundation of mathematics. The Bulletin of Symbolic Logic 2001; 7(4): 504-520.