ریاضی و جامعه
شاپا (چاپی): ۲۳۴۵-۶۴۹۳، شاپا(الکترونیکی): ۶۵۰۷ — ۲۳۴۵
جلد ۶، شماره ۳ (۱۴۰۰)، صص. ۳۵-۵۵

http://www.ui.ac.ir

http://math-sci.ui.ac.ir

کاربردهایی از منطق گزاره‌ای

سمیه تاری

چکیده. پیدایش بسیاری از نظریه‌های ریاضی حاصل استدلال و تفکر است. از مدل‌های مختلف منطق ریاضی، برای صوری‌سازی تفکر قیاسی استفاده می‌شود. منطق گزاره‌ای یکی از مدل‌های ریاضی برای تفکر و استنتاج‌های ریاضی است که در بسیاری از شاخه‌های مختلف علوم استفاده می‌شود. در این مقاله ضمن اشاره به جنبه‌های نظری و کامپیوتری منطق گزاره‌ای، کاربردهایی از آن در سایر علوم بیان می‌شوند.

۱. مقدمه

منطق گزاره‌ای، یکی از مدل‌های ریاضی بازنمایی استدلال و استنتاج قیاسی است. در این نوع مدل، گزاره‌ها و ارتباط بین آن‌ها بررسی و مطالعه می‌شوند. گزاره‌ها به دونوع گزاره‌های ساده (جملات خبری ساده) و گزاره‌های پیچیده تقسیم می‌شوند. در منطق گزاره‌ای از عملگرهای منطقی $\vee$، $\wedge$، $\neg$، $\rightarrow$ و $\leftrightarrow$ ، برای به‌دست آوردن گزاره‌ها استفاده می‌شود.

با وجود سادگی منطق گزاره‌ای، از آن در مدل‌سازی بسیاری از شاخه‌های علوم کامپیوتر و علوم مهندسی استفاده می‌شود. به‌عنوان مثال، شاخه عامل‌های منطقی در هوش مصنوعی، به استدلال‌های مبتنی بر انواع منطق از جمله منطق گزاره‌ای مربوط است. هم‌چنین منطق گزاره‌ای و منطق مرتبه اول در طراحی مدارهای منطقی و مدارهای دیجیتالی نقش اساسی دارند.

از دیدگاه ریاضی نیز تا حدودی انتقال مسائل به قالب صوری منطق گزاره‌ای مفید و کارگشا است؛ ولی این مدل ریاضی در صوری‌سازی تمام استدلال‌ها و استنتاج‌های ریاضی موفق نمی‌باشد. مثال معروف زیر در مورد استدلال، در منطق گزاره‌ای قابل بیان نیست:

سقراط یک انسان است.
هر انسانی خواهد مرد.
پس سقراط خواهد مرد.

در این مقاله منطق گزاره‌ای به‌صورت دقیق و ریاضی توصیف می‌شود. در بخش ۲، ابزار اصلی منطق گزاره‌ای یعنی فرمول‌های صوری بررسی و مطالعه می‌شوند. فرمول‌ها یا گزاره‌ها در منطق گزاره‌ای به دونوع اتمی و غیر اتمی تقسیم‌بندی می‌شوند. هر گزاره اتمی ترجمه‌های مختلفی را در زبان طبیعی دارد. در بخش ۳، مفهوم ترجمه گزاره‌های اتمی با استفاده از جدول ارزش و تابع ارزش به‌صورت ریاضی صورت‌بندی می‌شود و ارزش گزاره‌های پیچیده‌تر با استفاده از ارزش گزاره‌های اتمی محاسبه می‌شود. یکی از مفاهیم بسیار مهم در علوم کامپیوتر بررسی این است که آیا برای یک مدل ریاضی پیشنهادی، الگوریتمی موجود است که در مورد

عبارات و کلمات کلیدی: عامل منطقی، تابع بولی، ارزش درستی، منطق گزاره‌ای.
دبیرتخصصی رابط: صغری نوبختیان
نوع مقاله: مروری
تاریخ دریافت: ۱۴۰۰/۱۱/۱۰ تاریخ پذیرش: ۱۴۰۱/۰۶/۰۶
http://dx.doi.org/10.22108/MSCI.2022.132540.1492 .

عضویت دنباله‌ی متناهی از نمادهای مورد بحث در مدل پیشنهادی تصمیم بگیرد؟ در بخش ۴، منطق گزاره‌ای از این منظر مطالعه می‌شود. در بخش ۵، کاربردهایی از منطق گزاره‌ای آورده می‌شود.

۲. ساخت فرمول‌ها در منطق گزاره‌ای

در این بخش ساخت فرمول‌ها در منطق گزاره‌ای بررسی می‌شود. هر فرمولی در منطق گزاره‌ای، یک دنباله متناهی از نمادهای ریاضی است که طبق قواعد خاصی ایجاد شده‌اند.

در منطق گزاره‌ای، ترکیبات جملات خبری ساده مطالعه و بررسی می‌شوند. «π یک عدد گنگ است.» و «تهران پایتخت ایران است.» دو جمله خبری ساده هستند که در منطق گزاره‌ای، به‌ترتیب فرمول‌های اتمی $p_۱$ و $p_۲$ در نظر گرفته می‌شوند. از این جمله‌های ساده، جملات پیچیده‌تری به‌صورت زیر به‌دست می‌آیند:

- π یک عدد گنگ است و تهران پایتخت ایران است.
- π یک عدد گنگ است یا تهران پایتخت ایران است.
- اگر π یک عدد گنگ است، آنگاه تهران پایتخت ایران است.
- π یک عدد گنگ است اگر و تنها اگر تهران پایتخت ایران است.
- π یک عدد گنگ نیست.

در منطق گزاره‌ای، هر یک از جملات به‌دست آمده متناظر با یک فرمول است. به‌عنوان نمونه جمله اول، متناظر با فرمول $(p_۱ \wedge p_۲)$ است. در این بخش، تولید چنین فرمول‌هایی بدون در نظر گرفتن معنای واقعی آن‌ها مد نظر است و تمام مباحث با استفاده از ابزار و نمادهای ریاضی بیان می‌شوند.

تعریف ۱.۲. *یک فرمول اتمی، نماد p_i برای $i = ۱, ۲, \ldots$ است. نماد $\top$ یک فرمول اتمی در منطق گزاره‌ای است که همواره درست است. نماد $\bot$ یک فرمول اتمی در منطق گزاره‌ای است که همواره نادرست است. در واقع فرمول‌های اتمی همان جملات خبری و ساده زبان طبیعی هستند* ([۱] و [۷]).

مجموعه فرمول‌ها در منطق گزاره‌ای با روند استقرایی زیر تعریف می‌شوند:

۱- *هر فرمول اتمی، یک فرمول است.*

۲- *برای هر فرمول F، $(\neg F)$ نیز یک فرمول است.*

۳- *اگر F و G دو فرمول باشند، آنگاه $(F \wedge G)$، $(F \vee G)$، $(F \to G)$ و $(F \leftrightarrow G)$ نیز فرمول هستند.*

نمادهای ریاضی $\wedge$، $\vee$، $\to$، $\leftrightarrow$ و $\neg$ عملگرهای منطقی هستند که دارای نام‌های سنتی به شرح زیر هستند [۱]:

$\wedge$: عطف– وصل– «و»،

$\vee$: فصل– «یا»،

$\to$: شرطی– شرطیه– استلزام– «اگر ... آنگاه ...»،

$\leftrightarrow$: دو شرطی– «... اگر و تنها اگر ...»،

$\neg$: نفی– چنین نیست که ...– «نقیض».

مثال ۲.۲. *$F = (p_۱ \wedge p_۲)$ یک فرمول در منطق گزاره‌ای است.*

هر فرمولی مانند G که در فرمول دیگری مانند F ظاهر شده باشد، یک زیرفرمول F نامیده می‌شود. زیرفرمول‌های مثال ۲.۲، فرمول‌های $(p_۱ \wedge p_۲)$، $p_۱$ و $p_۲$ هستند.

تبصره ۳.۲. *تعریف استقرایی فرمول‌ها در منطق گزاره‌ای، ابزار قدرتمندی است که در دو جهت عمده از آن استفاده می‌شود:*

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

۱- اثبات وجود ویژگی‌های خاصی در مورد فرمول‌ها، با استفاده از استقرا انجام می‌شود که به اصل استقرا[1] معروف است. به‌عنوان مثال تساوی تعداد پرانتزهای راست و چپ در فرمول‌های منطق گزاره‌ای با استفاده از اصل استقرا ثابت می‌شود (به‌مثال ۲. ۴ توجه کنید).

۲- برخی مفاهیم جدید در مورد فرمول‌ها، با استفاده از استقرا تعریف می‌شوند که به تعریف به‌وسیله بازگشت[2] معروف است. به‌عنوان مثال درخت هر فرمول در منطق گزاره‌ای را می‌توان با استفاده از تعریف به‌وسیله بازگشت تعریف کرد (به‌مثال ۲. ۵ توجه کنید).

مثال ۲.۴. *در هر فرمولی در منطق گزاره‌ای، تعداد پرانتزهای چپ و راست برابر است.*

فرض می‌کنیم F یک فرمول باشد. در این‌صورت تعداد پرانتزهای راست فرمول F با نماد $R(F)$ و تعداد پرانتزهای چپ آن با نماد $L(F)$ نشان داده می‌شود. با استفاده از اصل استقرا نشان داده می‌شود که $R(F) = L(F)$.

- *اگر F یک فرمول اتمی باشد، آنگاه $R(F) = L(F) = ۰$.*
- *اگر $F = (G \circ H)$ که در آن $\circ \in \{\wedge, \vee, \to, \leftrightarrow\}$، $R(G) = L(G)$ و $R(H) = L(H)$، آنگاه*

$$R(F) = ۱ + R(H) + R(G)$$

$$L(F) = ۱ + L(H) + L(G)$$

بنابراین $L(F) = R(F)$.

- *اگر $F = (\neg G)$ که $R(G) = L(G)$، آنگاه $R(F) = ۱ + R(G) = ۱ + L(G) = L(F)$.*

مثال ۲.۵. *به هر فرمول در منطق گزاره‌ای یک درخت نسبت داده می‌شود. به ریشه این درخت، خود فرمول نسبت داده می‌شود و به گره‌های پایین‌تر، زیر فرمول‌های آن، نسبت داده می‌شود [۱، ۳].*

فرض کنیم درخت منسوب به‌فرمول F با نماد $T(F)$ نشان داده شود. تعریف استقرایی $T(F)$ به‌صورت زیر است:

- *اگر F یک فرمول اتمی باشد، آنگاه $T(F)$ درختی با یک گره است.*
- *اگر $F = (G \circ H)$ که در آن $\circ \in \{\wedge, \vee, \to, \leftrightarrow\}$، آنگاه $T(F)$ درخت زیر است:*

$$\begin{array}{c} F \\ \diagup \quad \diagdown \\ T(G) \quad T(H) \end{array}$$

- *اگر $F = (\neg G)$، آنگاه $T(F)$ درخت زیر است:*

$$\begin{array}{c} F \\ | \\ T(G) \end{array}$$

درخت مثال ۲. ۲، به‌صورت زیر است:

$$\begin{array}{c} p_۱ \wedge p_۲ \\ \diagup \quad \diagdown \\ p_۱ \qquad p_۲ \end{array}$$

[1] Induction Principle [2] Definition by recursion

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

در پایان گرامر مستقل از متن در فرم بکوس_نائور[3] برای ساخت فرمول‌ها در منطق گزاره‌ای آورده می‌شود [۶]:

$$G:\begin{cases} Sentence \to AtomicSentence \mid ComplexSentence \\ AtomicSentence \to \top \mid \bot \mid p_۱ \mid p_۲ \mid \cdots \\ ComplexSentence \to (Sentence) \mid \neg Sentence \mid Sentence \wedge Sentence \\ \qquad\qquad\qquad\qquad \mid Sentence \vee Sentence \\ \qquad\qquad\qquad\qquad \mid Sentence \to Sentence \\ \qquad\qquad\qquad\qquad \mid Sentence \leftrightarrow Sentence \\ OPERATOR\ PRECEDENCE: \ \neg, \wedge, \vee, \to, \leftrightarrow \end{cases}$$

گرامر فوق یک گرامر مستقل از متن است که با یک لیست تقدم استفاده از عملگرهای منطقی تکمیل شده است. این کار ابهام در استفاده از چند عملگر منطقی در یک فرمول را از بین می‌برد.

۳. معنای فرمول‌ها در منطق گزاره‌ای

یک مدل یا یک تعبیر برای فرمول‌ها، چیزی جز یک وضعیت ممکن برای راست یا غلط بودن آن‌ها نیست. منظور از معناشناسی منطق گزاره‌ای، چگونگی به‌دست آوردن درستی یا نادرستی یک فرمول از روی درستی یا نادرستی فرمول‌های اتمی موجود در آن فرمول است. در این بخش با استفاده از توابع ریاضی، مفهوم مدل یا تعبیر فرمول‌ها و معناشناسی منطق گزاره‌ای به‌صورت ساده جمع‌بندی می‌شود.

برای هر نماد فرمول اتمی مانند p_i، ترجمه‌های مختلفی در زبان طبیعی وجود دارد. در واقع هر جمله خبری ساده، یک ترجمه برای فرمول اتمی p_i است. براساس واقعیت امور در جهان مادی، هر کدام از این ترجمه‌ها درست یا نادرست هستند. بنابراین با در نظر گرفتن مجموعه $\{۰, ۱\}$ به‌عنوان مجموعه ارزش‌ها، به‌فرمول اتمی p_i، ارزش ۱ (به معنای درست بودن p_i) و یا ارزش ۰ (به معنای نادرست بودن p_i) نسبت داده می‌شود (مراجع [۱] و [۳] را ببینید).

تعریف ۱.۳. *فرض می‌کنیم D یک زیرمجموعه از مجموعه فرمول‌های اتمی باشد. هر تابع مانند $v : D \to \{۰, ۱\}$ یک تابع ارزش گفته می‌شود.*

لازم به‌توضیح است که اگر D یک مجموعه متناهی با n عضو باشد، آنگاه ۲^n تابع ارزش متفاوت با دامنه D وجود دارد. هر تابع ارزش مانند $v : D \to \{۰, ۱\}$، به هر گزاره اتمی $p_i \in D$، ارزش $v(p_i)$ را نسبت می‌دهد. با استفاده از ارزش هر فرمول اتمی ظاهر شده در یک فرمول، ارزش خود فرمول نیز محاسبه می‌شود.

تعریف ۲.۳. *فرض می‌کنیم D یک زیر مجموعه از فرمول‌های اتمی و $v : D \to \{۰, ۱\}$ یک تابع ارزش باشد. در این‌صورت تابع یکتا $V : E \to \{۰, ۱\}$ موجود است که در آن:*

- *E مجموعه فرمول‌های تولید شده از مجموعه D است.*
- *به‌ازای هر $p_i \in D$، $V(p_i) = v(p_i)$.*

فرض می‌کنیم برای $G \in E$ و $H \in E$، $V(G)$ و $V(H)$ تعریف شده باشد.

[3] Backus–Naur form

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

$$V(G \wedge H) = \begin{cases} ۱ & V(G) = V(H) = ۱ \\ ۰ & \text{در غیر این صورت} \end{cases}$$

$$V(G \vee H) = \begin{cases} ۰ & V(G) = V(H) = ۰ \\ ۱ & \text{در غیر این صورت} \end{cases}$$

$$V(G \to H) = \begin{cases} ۰ & V(G) = ۱, V(H) = ۰ \\ ۱ & \text{در غیر این صورت} \end{cases}$$

$$V(G \leftrightarrow H) = \begin{cases} ۱ & V(G) = V(H) \\ ۰ & \text{در غیر این صورت} \end{cases}$$

$$V(\neg G) = \begin{cases} ۱ & V(G) = ۰ \\ ۰ & \text{در غیر این صورت} \end{cases}$$

لازم به توضیح است که تعریف ۲.۳ نمونه‌ای از تعریف به‌وسیله بازگشت در منطق گزاره‌ای است. تأثیر معنایی عملگرهای منطقی، توسط جدول‌های زیر نیز بیان می‌شوند:

جدول ۱: تأثیر معنایی $\wedge$

G	H	$(G \wedge H)$
۰	۰	۰
۰	۱	۰
۱	۰	۰
۱	۱	۱

جدول ۲: تأثیر معنایی $\vee$

G	H	$(G \vee H)$
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۱

جدول ۳: تأثیر معنایی $\to$

G	H	$(G \to H)$
۰	۰	۱
۰	۱	۱
۱	۰	۰
۱	۱	۱

جدول ۴: تأثیر معنایی $\leftrightarrow$

G	H	$(G \leftrightarrow H)$
۰	۰	۱
۰	۱	۰
۱	۰	۰
۱	۱	۱

جدول ۵: تأثیر معنایی $\neg$

G	$(\neg G)$
۰	۱
۱	۰

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

مثال ۳.۳. *فرض می‌کنیم $v : \{p_1, p_2, p_3\} \to \{0, 1\}$ یک تابع ارزش باشد که $v(p_1) = v(p_2) = 0$ و $v(p_3) = 1$. در این‌صورت $V(((p_1 \vee p_2) \to p_3)) = 1$.*

تعریف ۴.۳. *فرض می‌کنیم F یک فرمول در منطق گزاره‌ای و v یک تابع ارزش باشد که $V(F) = 1$. در این‌صورت v یک مدل برای فرمول F است و با نماد $v \models F$ نشان داده می‌شود.*

تعریف ۵.۳. *فرض می‌کنیم F یک فرمول در منطق گزاره‌ای باشد.*

- *فرمول F یک فرمول صدق‌پذیر[4] است هرگاه حداقل یک مدل داشته باشد.*
- *فرمول F یک فرمول صدق ناپذیر[5] است هرگاه هیچ مدلی نداشته باشد.*
- *فرمول F یک فرمول همانگو[6] است هرگاه هر تابع ارزش، یک مدل آن باشد.*

برای به‌دست آوردن ارزش هر فرمول در منطق گزاره‌ای، دانستن ارزش فرمول‌های اتمی ظاهر شده در آن فرمول کافی است. بنابراین برای تشخیص صدق‌پذیر بودن یا همانگو بودن فرمول F، کافی است تمام توابع ارزش روی مجموعه فرمول‌های اتمی ظاهر شده در فرمول F مشخص شود. ارزش فرمول F برای هریک از توابع ارزش فوق محاسبه شود. همه توابع ارزش ممکن برای فرمول F با n فرمول اتمی در جدول ارزشی با 2^n سطر نمایش داده می‌شود (در بخش ۴ یک الگوریتم برای محاسبه جدول ارزش آورده شده است).

جدول ۶: جدول ارزش F

p_1	p_2	$\cdots$	p_{n-1}	p_n	F
۰	۰	$\cdots$	۰	۰	$V_1(F)$
۰	۰	$\cdots$	۰	۱	$V_2(F)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
۱	۱	$\cdots$	۱	۱	$V_{2^n}(F)$

اگر ستون مربوط به F، فقط ۰ باشد، آن‌گاه F یک فرمول صدق ناپذیر است. اگر ستون مربوط به F، فقط ۱ باشد، آن‌گاه F یک فرمول همانگو است. اگر ستون مربوط به F، هم شامل ۰ و هم شامل ۱ باشد، آن‌گاه F یک فرمول صدق‌پذیر است.

مثال ۶.۳. *فرمول F مثال ۳.۳ را در نظر بگیرید. جدول ۶.۳، جدول ارزش فرمول F است. بنابراین F یک فرمول صدق‌پذیر است؛ در حالی‌که یک فرمول همانگو نیست.*

در منطق گزاره‌ای مفهوم نتیجه منطقی یا استنتاج منطقی یکی از موضوعات کاربردی در بسیاری از زمینه‌ها است. فرض می‌کنیم Σ یک مجموعه از فرمول‌ها و v یک تابع ارزش باشد که روی تمام فرمول‌های اتمی موجود در فرمول‌های Σ تعریف شده باشد. اگر به‌ازای هر فرمول $F \in \Sigma$، $V(F) = 1$، آنگاه v را یک مدل Σ گویند.

تعریف ۷.۳. *فرض می‌کنیم Σ یک مجموعه از فرمول‌ها و F یک فرمول در منطق گزاره‌ای باشد. در این‌صورت F یک نتیجه منطقی Σ است و با نماد $\Sigma \models F$ نشان داده می‌شود هرگاه هر مدل Σ، یک مدل برای F نیز باشد.*

[4]Satisfiable [5]Unsatisfiable [6]Tautology

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

جدول ۷: جدول ارزش F

p_1	p_2	p_3	$(p_1 \vee p_2) \to p_3$
۰	۰	۰	۱
۰	۰	۱	۱
۰	۱	۰	۰
۰	۱	۱	۱
۱	۰	۰	۰
۱	۰	۱	۱
۱	۱	۰	۰
۱	۱	۱	۱

اگر Σ یک مجموعه متناهی از فرمول‌ها باشد، برای بررسی $\Sigma \models F$ کافی است ابتدا مدل‌های مجموعه Σ به‌دست آورده شود. سپس درست بودن فرمول F در مدل‌های Σ بررسی شود. این شیوه به روش وارسی مدل[7] معروف است که در عملکرد عامل‌های منطقی کاربرد فراوانی دارد و در بخش ۴ آورده خواهد شد.

۴. تصمیم‌پذیری منطق گزاره‌ای

در دو بخش قبلی جنبه نظری و ریاضی منطق گزاره‌ای مطرح شده است. در این بخش به‌مباحث پیرامون علوم کامپیوتر پرداخته می‌شود.

در بررسی صدق‌پذیری یک فرمول در منطق گزاره‌ای، می‌توان با به‌دست آوردن جدول ارزش مربوط به آن فرمول به‌صورت نظری و ریاضی مشخص کرد که آیا فرمول صدق‌پذیر است؟ سؤالی که در اینجا مطرح می‌شود این است که «آیا به‌صورت الگوریتمیک می‌توان به این سؤال جواب داد؟» طرح این نوع سؤالات در زمینه‌های مشابه در منطق ریاضی منجر به شاخه نظریه محاسبه‌پذیری شد که یکی از جنبه‌های مهم آن موضوع تصمیم‌پذیری است. در بررسی تصمیم‌پذیری هر مسئله ریاضی، دو نوع از اطلاعات باید مشخص باشد.

۱- ورودی صحیح مسئله،

۲- پرسشی که باید در مورد مسئله جواب داده شود (به عبارت دیگر هدف اصلی مسئله با این پرسش مشخص می‌شود).

یک مسئله ریاضی تصمیم‌پذیر[8] است هرگاه الگوریتمی موجود باشد که به‌ازای هر ورودی مسئله، جواب پرسش مسئله را در زمان محدود به‌صورت «بله» یا «خیر» مشخص کند.

مسئله صدق‌پذیری منطق گزاره‌ای (مسئله SAT) به‌صورت زیر است:

- ورودی مسئله: یک فرمول مانند F در منطق گزاره‌ای،
- پرسش مسئله: آیا فرمول F صدق‌پذیر است؟

ایده وجود الگوریتمی برای بررسی صدق‌پذیری فرمول‌ها در منطق گزاره‌ای به ساختن جدول ارزش برای هر فرمول برمی‌گردد. الگوریتم SAT برای این منظور استفاده می‌شود، که برای ورودی F، در پایان جواب true (یعنی صدق‌پذیری F) یا false (یعنی صدق‌ناپذیری F) را تولید می‌کند:

[7] Model-checking approach [8] Decidable

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

function SAT? (F) **return** true or false
  inputs: A sentence in propositional logic F
  symbols $\leftarrow$ a list of the proposition symbols in F
  return TT-CHECK- ALL (F, symbols, {})

function TT-CHECK- ALL (F, symbols, model) **return** true or false

  if EMPTY? (symbols) **then return** PL-TRUE? (F, model)
  else
    P $\leftarrow$ FIRST(symbols)
    rest $\leftarrow$REST(symbols)
    return(TT-CHECK-ALL(F, rest, model$\cup$ {P=true})
      and
     TT-CHECK-ALL(F, rest, model$\cup$ {P=false}))

الگوریتم SAT، برای ورودی F از الگوریتم TT-ENTAILS? استفاده می‌کند که مشابه همان الگوریتم از [۶] با تغییرات جزئی است. در الگوریتم فوق، تابع SAT، صدق‌پذیری فرمول F را مشخص می‌کند. تابع TT-CHECK- ALL، سطرهای مربوط به جدول ارزش فرمول F را مشخص می‌کند. تابع PL-TRUE، نیز درست بودن فرمول در هر سطر از جدول ارزش را مشخص می‌کند. برای جزئیات بیشتر در مورد توابع TT-CHECK- ALL و PL-TRUE به کدهای آنها [۴] مراجعه شود.

قضیه ۱.۴. *مسئله صدق‌پذیری فرمول‌ها در منطق گزاره‌ای تصمیم‌پذیر است.*

اثبات. برای صدق‌پذیری هر فرمول F در منطق گزاره‌ای جدول ارزش آن بررسی می‌شود. بنابراین با استفاده از الگوریتم SAT، صدق‌پذیری F به‌صورت الگوریتمیک مشخص می‌شود. $\square$

فرض کنید Σ یک مجموعه متناهی از فرمول‌ها و F یک فرمول در منطق گزاره‌ای باشد. روش وارسی مدل برای استنتاج منطقی $\Sigma \models F$ استفاده می‌شود. در واقع الگوریتم TT-ENTAILS? (همان روش وارسی مدل) به‌پرسش «آیا F نتیجه منطقی از Σ است؟» جواب true (به معنای بله) یا false (به معنای خیر) را می‌دهد. در الگوریتم فوق مشابه الگوریتم SAT، از تابع TT-CHECK- ALL استفاده شده است (برای جزئیات بیشتر در مورد الگوریتم فوق به [۶] مراجعه شود).

لازم به‌توضیح است که اجرای الگوریتم SAT، برای فرمولی با n فرمول اتمی نیازمند انجام ۲^n محاسبه مجزا است که مدت زمان حصول جواب را کنترل ناپذیر می‌کند. به‌عنوان مثال اگر در فرمول F، ۶۴ فرمول اتمی ظاهر شده باشد و مدت زمان محاسبات هر سطر از جدول ارزش یک ثانیه باشد، آنگاه مدت زمان مورد نیاز برای اتمام محاسبات جدول ارزش فرمول F، $\frac{۲^{۶۴}}{۳۱۶۰۰۰۰۰}$ سال است. پس زمان اجرای الگوریتم نمایی است. وجود یک الگوریتم کارآمد برای مسئله صدق‌پذیری فرمول‌ها در منطق گزاره‌ای، به‌صورت یک سؤال باز مطرح است که از نوع مسئله معروف $P = NP$ است.

برای صدق‌پذیری برخی از فرمول‌ها مانند فرمول‌های هورن[۹] و روش وارسی مدل در استنتاج‌های منطقی مربوط به‌فرمول‌های هورن، الگوریتم‌های کارآمدی موجود است که در ادامه مطرح و بررسی می‌شوند.

[9] Horn Formulas

تعریف ۲.۴. *دو فرمول F و G هم‌ارز منطقی هستند هرگاه به‌ازای هر تابع ارزش v، $V(F) = V(G)$.*

هم‌ارزی منطقی F و G با نماد $F \equiv G$ نشان داده می‌شود.

تعریف ۳.۴. *هر فرمول اتمی یا نقیض فرمول اتمی یک لفظ[10] نامیده می‌شود.*

تعریف ۴.۴. *فرم نرمال عطفی و فرم نرمال فصلی دو استاندارد برای فرمول‌ها در منطق گزاره‌ای هستند. فرض کنید F یک فرمول در منطق گزاره‌ای باشد.*

- *فرمول F در فرم نرمال عطفی[11] (CNF) است هرگاه*

$$F = (\bigwedge_{i=1}^{n}(\bigvee_{j=1}^{m_i} L_{i,j}))$$

که به‌ازای هر i و j، $L_{i,j}$ یک لفظ است. به‌ازای هر i، $\bigvee_{j=1}^{m_i} L_{i,j}$ یک بندفصلی گفته می‌شود.

- *فرمول F در فرم نرمال فصلی[12] (DNF) است هرگاه*

$$F = (\bigvee_{i=1}^{n}(\bigwedge_{j=1}^{m_i} L_{i,j}))$$

که به‌ازای هر i و j، $L_{i,j}$ یک لفظ است. به‌ازای هر i، $\bigwedge_{j=1}^{m_i} L_{i,j}$ یک بندعطفی گفته می‌شود.

مثال ۵.۴. *فرمول $F = ((p_1 \wedge \neg p_2) \vee (p_{100} \wedge p_2 \wedge \neg p_4))$ در فرم DNF است.*

فرمول $G = ((p_{100} \vee \neg p_2 \vee \neg p_1) \wedge (p_2) \wedge (p_3 \vee \neg p_6))$ در فرم CNF است.

قضیه ۶.۴. *برای هر فرمول F در منطق گزاره‌ای، فرمول هم‌ارز منطقی F_1 در فرم CNF و فرمول هم‌ارز منطقی F_2 در فرم DNF موجود هستند.*

اثبات. مراجعه کنید به [۷، بخش ۱.۲]. □

تعریف ۷.۴.
- *یک بند فصلی را بند معین[13] گویند هرگاه دقیقاً یک لفظ مثبت (فرمول اتمی) در آن موجود باشد.*
- *یک بند فصلی را بند هورن[14] گویند هرگاه حداکثر یک لفظ مثبت (فرمول اتمی) در آن موجود باشد.*
- *فرمول F در فرم CNF یک فرمول هورن است هرگاه هر بندفصلی آن یک بند هورن باشد.*

گرامری برای تولید فرم‌های CNF، بندهای هورن و بندهای معین از [۶، بخش ۲.۵.۷] با تغییرات جزئی به‌صورت زیر است:

$$G : \begin{cases} CNFSentence \rightarrow Clause_1 \wedge \cdots \wedge Clause_n \\ Clause \rightarrow Literal_1 \vee \cdots \vee Literal_m \\ Literal \rightarrow Symbol \mid \neg Symbol \\ Symbol \rightarrow \top \mid \bot \mid p_1 \mid p_2 \mid \cdots \\ HornClauseForm \rightarrow DefiniteClauseForm \mid \neg Symbol \\ DefiniteClauseForm \rightarrow Symbol \mid (\neg Symbol_1 \vee \cdots \vee \neg Symbol_l \vee Symbol) \end{cases}$$

با توجه به هم‌ارزی منطقی $\neg A \vee \neg B \vee C \equiv (A \wedge B) \rightarrow C$، هر بند فصلی در فرمول هورن، با استفاده از عملگرهای منطقی $\wedge$ و $\rightarrow$ بازنویسی می‌شود:

[10] Literal [11] Conjunctive Normal Form [12] Disjunctive Normal Form [13] Definite clause [14] Horn clause

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

- بند معین $\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n \vee p_{n+1}$ به‌صورت $(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \to p_{n+1}$ بازنویسی می‌شود.
- بند معین p_i به‌صورت $\top \to p_i$ بازنویسی می‌شود.
- بند $\neg p_i$ به‌صورت $p_i \to \bot$ بازنویسی می‌شود.

بنابراین هر فرمول هورن به‌صورت ترکیب عطفی بندهای فصلی بازنویسی شده، برای استفاده در الگوریتم صدق‌پذیری هورن[15] نوشته می‌شود.

مسئله صدق‌پذیری (SAT) برای فرمول‌های هورن به‌صورت زیر است:

- ورودی مسئله: فرمول هورن F،
- پرسش مسئله: آیا فرمول F صدق‌پذیر است؟

الگوریتم صدق‌پذیری هورن (SAT) برای فرمول‌های هورن [۷]:

۱- فرمول هورن F را دریافت و به‌صورت CNF بازنویسی کن.

۲- اگر بندی به‌صورت $\top \to p_i$ موجود باشد، آنگاه همه p_iهای فرمول F را به p_i^* تبدیل کن.

۳- حلقه زیر را تکرار کن:

۳-۱ اگر بندی به‌صورت $p_1^* \wedge \cdots \wedge p_n^* \to p_{n+1}$ موجود باشد، آنگاه همه p_{n+1}های فرمول F را به p_{n+1}^* تبدیل کن.

۳-۲ اگر بندی به‌صورت $p_1^* \wedge \cdots \wedge p_n^* \to \bot$ موجود باشد، آنگاه کلمه «صدق ناپذیر» را چاپ کن و متوقف شو.

۴- کلمه «صدق‌پذیر» را چاپ کن و تابع v را روی اتمهای موجود در F به‌صورت زیر تعریف کن:

اگر p_i به p_i^* تبدیل شده باشد، آنگاه $v(p_i) = 1$. در غیراین صورت $v(p_i) = 0$.

تبصره ۸.۴. *الگوریتم صدق‌پذیری هورن برای فرمول‌های هورن با n فرمول اتمی، حداکثر در n مرحله تمام می‌شود. بنابراین زمان اجرای آن به‌صورت خطی است که در مقایسه با الگوریتم حاصل از جدول ارزش قابل کنترل است.*

تبصره ۹.۴. *اگر فرمول هورن F، یک فرمول صدق‌پذیر در منطق گزاره‌ای باشد، آنگاه الگوریتم صدق‌پذیری هورن برای فرمول F به‌عنوان ورودی، خروجی «صدق‌پذیر» را تولید می‌کند. هم‌چنین اگر الگوریتم فوق برای فرمول هورن F به‌عنوان ورودی، خروجی «صدق‌پذیر» را تولید کند، آنگاه فرمول F یک فرمول صدق‌پذیر است.*

مثال ۱۰.۴. *فرمول $F = \{(p_1 \wedge p_2 \wedge p_3 \to \bot) \wedge (p_4 \to \bot) \wedge (p_5 \to p_1) \wedge (\top \to p_5)\}$ را درنظربگیرید. الگوریتم صدق‌پذیری هورن برای فرمول F به‌صورت زیر اجرا می‌شود:*

۱- چون $\top \to p_5$ موجود است. پس با اجرای گام ۱ از الگوریتم صدق‌پذیری هورن، p_5 به p_5^* تبدیل می‌شود.

$$((p_1 \wedge p_2 \wedge p_3 \to \bot) \wedge (p_4 \to \bot) \wedge (p_5^* \to p_1) \wedge (\top \to p_5^*))$$

۲- چون $(p_5^* \to p_1)$ موجود است. پس با اجرای گام ۳-۱ از الگوریتم صدق‌پذیری هورن، p_1 به p_1^* تبدیل می‌شود.

$$((p_1^* \wedge p_2 \wedge p_3 \to \bot) \wedge (p_4 \to \bot) \wedge (p_5^* \to p_1^*) \wedge (\top \to p_5^*))$$

۳- با اجرای گام ۴ از الگوریتم صدق‌پذیری هورن، کلمه «صدق‌پذیر» به‌عنوان خروجی صادر می‌شود.

تابع ارزش $v : \{p_1, \ldots, p_5\} \to \{0, 1\}$ که $v(p_1) = v(p_5) = 1$ و $v(p_2) = v(p_3) = v(p_4) = 0$ یک مدل برای F است.

به‌طور مشابه الگوریتمی برای بررسی نتایج منطقی وجود دارد. فرض کنید Σ یک مجموعه متناهی از فرمول‌ها در فرم هورن باشد که هر بند آن یک بند معین است و p یک فرمول اتمی باشد، بررسی $\Sigma \models p$، با استفاده از روش وارسی مدل روش کار آمدی نیست (با توجه به‌توضیحات بخش ۲). الگوریتم زنجیره‌سازی پیشرو[16]، روش بسیار کارآمدی برای بررسی $\Sigma \models p$ است [۶]. مطابق این

[15] Horn-SAT algorithm [16] Forward-chaining algorithm

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

الگوریتم (PL-FC-ENTAILS)، در مجموعه Σ حقایقی به‌صورت فرمول‌های اتمی و فرمول‌های دیگری به فرم هورن وجود دارند. اگر در یک عبارت شرطی موجود در Σ، مقدم‌ها همگی عضو Σ باشند، تالی آن نیز به مجموعه حقایق اضافه می‌شود. همین روند تا یافتن p به‌عنوان یک حقیقت ادامه می‌یابد. الگوریتم PL-FC-ENTAILS و شبه کد آن در ادامه آورده می‌شود ([۶، بخش ۴.۵.۷]):

function PL-FC-ENTAILS? (Σ, α) **return** true or false

inputs: Σ, A sentence in propositional logic α

count $\leftarrow$ a table, where count[c] is initially the number of symbols in clause c's premise

inferred $\leftarrow$ a table, where inferred[s] is initially false for all symbols

a queue of symbols, initially symbols known to be true in Σ

while queue is not empty **do**

p $\leftarrow$Pop (queue)

if $p = \alpha$ **then return** true

if inferred[p]=false **then**

inferred[p]$\leftarrow$true

for each clause c in Σ where p is in c.PREMISE **do**

decrement count [c]

if count[c]=0 **then** add c.CONCLUSION to queque

return false

مثال ۱۱.۴. *فرض کنید* $\Sigma = \{p_1, p_2, \neg p_1 \vee \neg p_2 \vee p_3, \neg p_3 \vee \neg p_1 \vee p_4\} = \{p_1, p_2, p_1 \wedge p_2 \to p_3, p_1 \wedge p_3 \to p_4\}$. *الگوریتم زنجیره‌سازی پیشرو برای استنتاج* $\Sigma \models p_4$ *به‌صورت زیر است:*

مشخصات جدول count، جدول inferred و queue در ابتدا به‌صورت زیر است:

count$[p_1 \wedge p_2 \to p_3] = |\ \{p_1, p_2\}\ | = 2$

count$[p_1 \wedge p_3 \to p_4] = |\ \{p_1, p_3\}\ | = 2$

inferred$[p_1]$*=false*

inferred$[p_2]$*=false*

inferred$[p_3]$*=false*

inferred$[p_4]$*=false*

queue$=\{p_1, p_2\}$

در مرحله اول inferred$[p_1]$ *به true تبدیل می‌شود. بنابراین فرمول اتمی* p_1 *در مقدم فرمول* $p_1 \wedge p_2 \to p_3$ *و مقدم فرمول* $p_1 \wedge p_3 \to p_4$ *به‌حقیقت تبدیل می‌شود. لذا در این مرحله* count$[p_1 \wedge p_2 \to p_3]$ *به* count$[\top \wedge p_2 \to p_3]$*،* count$[p_1 \wedge p_3 \to p_4]$ *به* count$[\top \wedge p_3 \to p_4]$ *و queue به‌مجموعه* $\{p_2\}$ *تبدیل می‌شود. مشخصات جدول count، جدول inferred و queue در این مرحله به‌صورت زیر است:*

count$[\top \wedge p_2 \to p_3] = |\ \{p_2\}\ | = 1$

count$[\top \wedge p_3 \to p_4] = |\ \{p_3\}\ | = 1$

inferred$[p_1]$*=true*

inferred$[p_2]$*=false*

inferred$[p_3]$*=false*

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

inferred[p_4]=false

queue=$\{p_2\}$

در مرحله دوم inferred[p_2] به true تبدیل می‌شود. بنابراین فرمول اتمی p_2 در مقدم فرمول $\top \wedge p_2 \to p_3$ به‌حقیقت تبدیل می‌شود. لذا در این مرحله count[$\top \wedge p_2 \to p_3$] به count[$\top \to p_3$] و queue به‌مجموعه $\{p_3\}$ تبدیل می‌شود. مشخصات جدول count، جدول inferred و queue در این مرحله به‌صورت زیر است:

count[$\top \to p_3$]=$|\emptyset| = 0$

count[$\top \wedge p_3 \to p_4$]=$|\{p_3\}| = 1$

inferred[p_1]=true

inferred[p_2]=true

inferred[p_3]=false

inferred[p_4]=false

queue=$\{p_3\}$

در مرحله سوم inferred[p_3] به true تبدیل می‌شود. بنابراین فرمول اتمی p_3 در تالی فرمول $\top \to p_3$ و مقدم فرمول $\top \wedge p_3 \to p_4$ به‌حقیقت تبدیل می‌شود. لذا در این مرحله count[$\top \to p_3$] به count[$\top \to \top$]، count[$\top \wedge p_3 \to p_4$] به count[$\top \to p_4$] و queue به‌مجموعه $\{p_4\}$ تبدیل می‌شود. مشخصات جدول count، جدول inferred و queue در این مرحله به‌صورت زیر است:

count[$\top \to \top$]=$|\emptyset| = 0$

count[$\top \to p_4$]=$|\emptyset| = 0$

inferred[p_1]=true

inferred[p_2]=true

inferred[p_3]=true

inferred[p_4]=false

queue=$\{p_4\}$

چون queue=$\{p_4\} \neq \emptyset$ و α همان p_4 است پس جواب true رو تولید می‌کند که به معنای درست بودن استنتاج $\Sigma \models p_4$ است.

۵. کاربردهایی از منطق گزاره‌ای

بسیاری از مسائل در زندگی روزانه، به‌مجموعه‌ای از فرمول‌ها در منطق گزاره‌ای تبدیل می‌شوند و با بررسی صدق‌پذیری این مجموعه از فرمول‌ها، راه حل‌های مناسب برای مسائل پیدا می‌شود. در ادامه سه نمونه از کاربردهای منطق گزاره‌ای در ۱.۵، ۲.۵ و ۳.۵ توصیف می‌شوند.

۱.۵. معمای دختر زیبا و ببر گرسنه. داستان دختر زیبا و ببر گرسنه نوشته استاکتن[17] برای خیلی از افراد آشنا است [۲]. در این داستان زندانی باید از میان دو اتاق یکی را انتخاب کند. در یکی از اتاق‌ها دختری زیبا و در اتاق دیگر ببری گرسنه وجود دارد. اگر زندانی اتاقی را که دختر در آن است، انتخاب کند، آن‌گاه با دختر ازدواج می‌کند. اگر اتاق دیگر را انتخاب کند، آن‌گاه ببر او را خواهد خورد. پادشاهی این داستان را به گونه‌ای دیگر برای زندانیان مطرح می‌کند. «در هر یک از دو اتاق دختری زیبا و ببری گرسنه وجود دارد؛ اما ممکن است ببر و دختر هر دو در یک اتاق باشند یا در یک اتاق دختر و در دیگری ببر باشد.» همچنین پادشاه بر در

[17] Stockton

اتاق‌ها جملاتی را نصب می‌کند و اشاره می‌کند که جملات یکی از درها درست و دیگری نادرست است:

(جملات اتاق ۱)
در این اتاق دختر و در اتاق دیگر ببر وجود دارد.

(جملات اتاق ۲)
در یکی از دو اتاق دختر و در دیگری ببر وجود دارد.

شکل ۱: معمای دختر و ببر

اگر شما به‌جای زندانی بودید کدام در را باز می‌کردید؟

فرض کنید فرمول‌های اتمی $p_۱$ و $p_۲$ به‌ترتیب جملات خبری ساده «دختر در اتاق ۱ است.» و «ببر در اتاق ۲ است.» باشند. معما به دو مسئله زیر تبدیل می‌شود:

۱- برچسب اتاق ۱ درست و برچسب اتاق ۲ نادرست است.

۲- برچسب اتاق ۲ درست و برچسب اتاق ۱ نادرست است.

بنابراین مسئله ۱، مجموعه فرمول‌های $\{p_۱ \wedge p_۲, \neg((p_۱ \wedge p_۲) \vee (\neg p_۱ \wedge \neg p_۲))\}$ است که یک مجموعه صدق ناپذیر است. بنابراین برچسب اتاق ۱ نادرست و برچسب اتاق ۲ درست است. پس دختر در اتاق ۲ است و ببر در اتاق ۱ است. بنابراین زندانی باید اتاق ۲ را انتخاب کند.

لازم به‌توضیح است که مسئله ۲، مجموعه فرمول‌های $\{(\neg p_۱ \vee \neg p_۲), ((p_۱ \wedge p_۲) \vee (\neg p_۱ \wedge \neg p_۲))\}$ است که یک مجموعه صدق‌پذیر است. اگر $v : \{p_۱, p_۲\} \to \{۰, ۱\}$ یک تابع ارزش باشد که $v(p_۱) = v(p_۲) = ۰$، آنگاه v یک مدل برای مجموعه‌ی فرمول‌های بالا است. یعنی $\neg p_۱$ و $\neg p_۲$ درست هستند. پس دختر در اتاق ۲ است.

۲.۵. طراحی مدارهای منطقی. کاربرد دیگری از منطق گزاره‌ای، استفاده از آن در طراحی مدارهای منطقی مورد استفاده در سیستم‌های محاسباتی مدرن است که در ادامه با آوردن تعاریف و مفاهیم مورد نیاز بیان می‌شود [۳].

تعریف ۱.۵. *تابع $f : \{۰, ۱\}^k \to \{۰, ۱\}$ را یک تابع بولی kـ موضعی[۱۸] نامند.*

مثال ۲.۵. *تابع $C : \{۰, ۱\}^۲ \to \{۰, ۱\}$ با ضابطه $C(۰, ۱) = C(۰, ۰) = C(۱, ۱) = ۱$ و $C(۱, ۰) = ۰$، یک تابع بولی 2ـ موضعی است.*

تبصره ۳.۵. *فرض کنید $f : \{۰, ۱\}^n \to \{۰, ۱\}$ یک تابع بولی nـ موضعی باشد. جدول ارزش متناظر با تابع بولی f، ۲^n سطر و $n + ۱$ ستون دارد. هر سطر از جدول ارزش به‌صورت $(x_۱, \dots, x_n, x_{n+۱})$ است که $(x_۱, \dots, x_n) \in \{۰, ۱\}^n$ و $x_{n+۱} = f(x_۱, \dots, x_n)$. به‌عنوان مثال جدول ۸، جدول ارزش تابع بولی مثال ۲.۵ است.*

[18] k-arity Boolean function

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

جدول ۹: جدول ارزش مثال ۴.۵

شماره سطر	x_1	x_2	x_3	x_4
۱	۱	۱	۱	۰
۲	۱	۱	۰	۱
۳	۱	۰	۱	۱
۴	۱	۰	۰	۰
۵	۰	۱	۱	۱
۶	۰	۱	۰	۱
۷	۰	۰	۱	۰
۸	۰	۰	۰	۱

جدول ۸: جدول ارزش تابع بولی مثال ۲.۵

شماره سطر	x_1	x_2	x_3
۱	۰	۰	۱
۲	۰	۱	۱
۳	۱	۰	۰
۴	۱	۱	۱

بنابراین هرتابع بولی متناظر با یک جدول ارزش و هر جدول ارزش متناظر با یک تابع بولی است. از طرف دیگر برای هر فرمول یک جدول ارزش موجود است. می‌توان برای هر جدول ارزش یک فرمول در منطق گزاره‌ای به‌دست آورد (به‌مثال ۴.۵ توجه کنید). پس هر فرمول در منطق گزاره‌ای متناظر با یک تابع بولی و هر تابع بولی متناظر با یک فرمول در منطق گزاره‌ای است.

مثال ۴.۵. *جدول ۴.۵ را در نظربگیرید.*

جدول ۴.۵ متناظر با تابع بولی ۳ـ موضعی $f : \{0,1\}^3 \to \{0,1\}$ است که در آن:

$$f(1,1,1) = f(1,0,0) = f(0,0,1) = 0$$
$$f(1,1,1) = f(1,0,1) = f(0,1,1) = f(0,1,0) = f(0,0,0) = 1$$

لازم به‌توضیح است که برای به‌دست آوردن یک فرمول در فرم نرمال عطفی متناظر با تابع بولی، از هر سطری که ارزش صفر دارد، یک بند فصلی دارای ارزش صفر به‌ازای ارزش‌دهی آن سطر ایجاد می‌شود. ترکیب عطفی همه بندهای فصلی متناظر با سطرهای دارای ارزش صفر، فرمولی در فرم نرمال عطفی را ایجاد می‌کند. به‌ازای هر سطری که دارای ارزش صفر است، یکی از بندهای فصلی آن دارای ارزش صفر است و لذا فرمول دارای ارزش صفر است. به‌ازای بقیه سطرها چون همه بندهای فصلی موجود در فرمول دارای ارزش یک هستند، پس فرمول دارای ارزش یک است.

برای به‌دست آوردن یک فرمول در فرم نرمال فصلی متناظر با تابع بولی، از هر سطری که ارزش یک دارد، یک بند عطفی دارای ارزش یک به‌ازای ارزش‌دهی آن سطرایجاد می شود. ترکیب فصلی همه بندهای عطفی متناظر با سطرهای دارای ارزش یک، فرمولی در فرم نرمال فصلی را ایجاد می‌کند. به‌ازای هر سطری که دارای ارزش یک است، یکی از بندهای عطفی آن دارای ارزش یک است و

لذا فرمول دارای ارزش یک است. به‌ازای بقیه سطرها چون همه بندهای عطفی دارای ارزش صفر هستند، فرمول دارای ارزش صفر است.

در ادامه چگونگی به‌دست آوردن فرمولی در فرم نرمال عطفی متناظر با تابع بولی فوق توصیف می‌شود (فرم نرمال فصلی به‌طور مشابه با سطرهای یک به‌دست می‌آید):

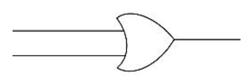
به‌ازای هر سطری که دارای ارزش صفر است یک فرمول با استفاده از فرمول‌های اتمی و عملگر منطقی $\vee$ ایجاد می‌شود.

- *سطر یک:* $(\neg p_1 \vee \neg p_2 \vee \neg p_3)$
- *سطر چهار:* $(\neg p_1 \vee p_2 \vee p_3)$
- *سطر هفت:* $(\neg p_1 \vee \neg p_2 \vee p_3)$

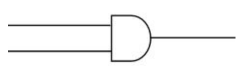
عطف سه بندفصلی به‌دست آمده فرمول موردنظر است.

$$((\neg p_1 \vee \neg p_2 \vee \neg p_3) \wedge (\neg p_1 \vee p_2 \vee p_3) \wedge (\neg p_1 \vee \neg p_2 \vee p_3))$$

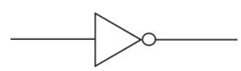
تابع‌های بولی و فرمول‌های متناظر با آنها، از جمله در صوری‌سازی مدارهای دیجیتالی استفاده می‌شوند. عمل هر دستگاه الکتریکی به‌وسیله تابع بولی $F(x_1, x_2, \ldots, x_n) = y$ توصیف می‌شود که در آن y خروجی دستگاه الکتریکی به‌ازای ورودی $(x_1, x_2, \ldots, x_n)$ است. گیت‌های منطقی بیشترین کاربرد را در طراحی مدارهای الکتریکی دارند که متناظر با توابع بولی ۲ـ موضعی و ۱ ـ موضعی هستند.



شکل ۲: گیت منطقی OR



شکل ۳: گیت منطقی AND



شکل ۴: گیت منطقی NOT

در گیت منطقی OR، ولتاژ خروجی بیشینه ولتاژ ورودی است. تابع بولی گیت OR همان تابع بولی مربوط به‌عملگر منطقی $\vee$ است (شکل ۲). در گیت منطقی AND، ولتاژ خروجی کمینه ولتاژ ورودی است. تابع بولی گیت AND همان تابع بولی مربوط به‌عملگر منطقی $\wedge$ است (شکل ۳). در گیت منطقی NOT، ولتاژ خروجی مخالف ولتاژ ورودی است. تابع بولی گیت NOT همان تابع بولی مربوط به‌عملگر منطقی $\neg$ است (شکل ۴).

از گیت‌های منطقی در طراحی مدارهای الکتریکی به‌این صورت استفاده می‌شود که ابتدا مسئله به یک تابع بولی تبدیل می‌شود. از روی جدول ارزش تابع بولی، فرمول متناظر با آن جدول ارزش به‌دست می‌آید. در پایان از روی فرمول به‌دست آمده، مدار مربوط به‌مسئله طراحی می‌شود.

مثال ۵.۵. *مداری طراحی کنید که یک قفل را با سه کلید A، B و C کنترل کند. رمز قفل در حالتی باز می‌شود که کلید C بسته باشد یا کلیدهای A و B هر دو بسته باشند (بسته بودن به‌مفهوم ۱ منطقی است).*

تابع بولی f متناظر با این مسئله یک تابع بولی سه موضعی است که:

$$f(0,0,0) = f(0,1,0) = f(1,0,0) = 0$$
$$f(1,1,1) = f(1,1,0) = f(1,0,1) = f(0,1,1) = f(0,0,1) = 1$$

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

بنابراین جدول ۱۰ ، جدول ارزش تابع بولی است.

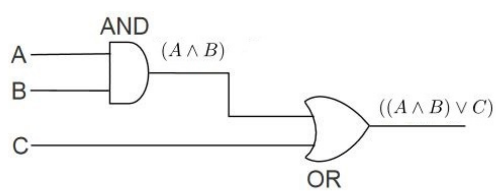
جدول ۱۰: جدول ارزش تابع بولی f

شماره سطر	$x_۱$	$x_۲$	$x_۳$	$x_۴$
۱	۱	۱	۱	۱
۲	۱	۱	۰	۱
۳	۱	۰	۱	۱
۴	۱	۰	۰	۰
۵	۰	۱	۱	۱
۶	۰	۱	۰	۰
۷	۰	۰	۱	۱
۸	۰	۰	۰	۰

گزاره منطقی $((\neg A \vee B \vee C) \wedge (A \vee \neg B \vee C) \wedge (A \vee B \vee C))$ ، فرمول متناظر با تابع بولی f است.

$$\begin{aligned}((\neg A \vee B \vee C) \wedge (A \vee \neg B \vee C) \wedge (A \vee B \vee C)) &\equiv ((\neg A \vee B \vee C) \wedge ([(A \vee \neg B) \wedge (A \vee B)] \vee C)) \\ &\equiv ((\neg A \vee B \vee C) \wedge ([A \vee (\neg B \wedge B)] \vee C)) \\ &\equiv ((\neg A \vee B \vee C) \wedge (A \vee C)) \\ &\equiv ([(\neg A \vee B) \wedge A] \vee C) \\ &\equiv ((A \wedge B) \vee C)\end{aligned}$$

بنابراین گزاره منطقی متناظر با مسئله فرمول $((A \wedge B) \vee C)$ می‌باشد که مدار منطقی آن شکل ۵ است.



شکل ۵: مدار منطقی مثال ۵.۵

۳.۵. دنیای ومپوز. در این بخش محیط و عملکرد یک عامل منطقی[19] مبتنی بر دانش بیان می‌شود و نقش منطق گزاره‌ای در عملکرد آن توصیف می‌شود. بخش اصلی از یک عامل منطقی مبتنی بر دانش، پایگاه دانش[20] (KB) است. در واقع پایگاه دانش جملاتی در مورد واقعیت‌هایی از دنیای عامل منطقی هستند که یا به‌صورت اصول تعریف نشده هستند یا جملاتی هستند که از سایر جملات پایگاه دانش به‌دست آمده‌اند. جملات موجود در پایگاه دانش به‌صورت فرمول‌هایی در زبان‌های صوری مختلف بیان می‌شوند. در اینجا جملات پایگاه دانش در منطق گزاره‌ای بیان می‌شود.

[19]Logical Agent [20]Knowledge Base

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

ومپوز[۲۱] نام یک بازی کامپیوتری است که در آن عامل به‌یک غار با اتاق‌های همجوار وارد می‌شود. در جای برخی اتاق‌ها گودال یا تله‌ای وجود دارد که در اتاق‌های مجاور آن‌ها ورزیدن نسیم ادراک می شود، هم‌چنین در هر اتاق امکان وجود ومپوز و طلا وجود دارد. هدف عامل پیدا کردن طلا و با سلامت از غار خارج شدن است. اگر عامل وارد اتاقی شود که در آن تله وجود دارد، آنگاه در تله گیر کرده، کشته می‌شود و امتیاز –۱۰۰۰ را کسب می‌کند. اگر عامل وارد اتاقی شود که ومپوز وجود دارد، آنگاه یا توسط ومپوز خورده خواهد شد و امتیاز –۱۰۰۰ را کسب می‌کند یا به ومپوز شلیک می‌کند و به‌ازای هر شلیک امتیاز –۱۰ را کسب می‌کند. لازم به‌توضیح است که در طول بازی عامل فقط در یک اتاق حق استفاده از تیر برای کشتن ومپوز را دارد. در ادامه تابع عامل مربوط به ومپوز توصیف می‌شود.

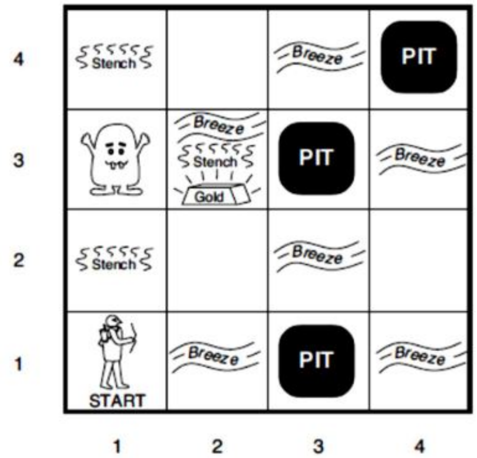
- **معیار کارآیی**
  برای برداشتن طلا +۱۰۰۰ امتیاز،
  برای افتادن در تله یا کشته شدن –۱۰۰۰ امتیاز و
  برای شلیک تیر –۱۰ امتیاز.
- **محیط**
  محیط یک مربع ۴×۴ است با اتاق‌های هم‌جوار و در ارتباط با هم است.
  در شروع عامل در اتاق [۱, ۱] قرار دارد.
  در همه اتاق‌ها به جز اتاق [۱, ۱] امکان وجود تله، ومپوز و طلا وجود دارد.
  بوی نامطبوع در اتاق‌های همجوار ومپوز وجود دارد.
  نسیم در اتاق‌های همجوار تله وجود دارد.
  درخشش در اتاق حاوی طلا وجود دارد.
- **حسگرها** عامل دارای ۵ حسگر است که اطلاعات جزئی را به‌عامل می‌دهد.
  بوی نامطبوع [۲۲]، امکان وجود ومپوز در اتاق‌های همجوار را می‌دهد.
  نسیم[۲۳]، امکان وجود تله را در اتاق‌های همجوار می‌دهد.
  درخشش[۲۴]، وجود طلا را در اتاقی که عامل در آن قرار دارد را مشخص می‌کند.
  جیغ زدن[۲۵]، کشته شدن ومپوز را مشخص می‌کند.
  ضربه[۲۶]، برخورد عامل با دیوارهای اتاق را مشخص می‌کند.
- **محرک‌ها**
  گردش به‌چپ، گردش به‌راست، جلو رفتن، برداشتن، انداختن، شلیک کردن

شکل ۶ یک نمونه از دنیای ومپوز است. ادراک دریافتی عامل در هر لحظه توسط یک پنج تایی [Stench, Breeze, Glitter, Screem, Bumb] مشخص می‌شود که مولفه اول در مورد وجود بوی نامطبوع، مولفه دوم در مورد وجود نسیم، مولفه سوم در مورد وجود درخشش، مؤلفه چهارم در مورد وجود جیغ و مؤلفه پنجم در مورد وجود ضربه اطلاعات می‌دهد. پنج‌تایی [None, None, None, None, None] نشانگر این است که در اتاقهای همجوار اتاقی که عامل قرار دارد، تله و ومپوز وجود ندارد، در اتاقی که عامل قرار دارد طلایی وجود ندارد، ومپوز کشته نشده است و برخوردی به دیوار هم صورت نگرفته است.

مجموعه استنتاج‌های انجام گرفته توسط عامل منطقی برای فهمیدن وجود ومپوز در اتاق [۱, ۳]، تله در اتاق [۳, ۱] و امن بودن اتاق [۲, ۲] ابتدا به‌صورت شهودی در ادامه بیان می‌شود:

[21] Wumpus [22] Stench [23] Breeze [24] Glitter [25] Screem [26] Bumb

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492



شکل ۶: نمونه‌ای از دنیای ومپوز

[1,4]	[2,4]	[3,4]	[4,4]
[1,3]	[2,3]	[3,3]	[4,3]
[1,2] ok	[2,2]	[3,2]	[4,2]
[1,1] ok	[2,1] ok	[3,1]	[4,1]

شکل ۷: وضعیت عامل در شروع بازی

[1,4]	[2,4]	[3,4]	[4,4]
[1,3] W?	[2,3]	[3,3]	[4,3]
[1,2] Ok S	[2,2] W?	[3,2]	[4,2]
[1,1] ok	[2,1] ok	[3,1]	[4,1]

شکل ۸: وضعیت عامل در اتاق [۱, ۲]

[1,4]	[2,4]	[3,4]	[4,4]
[1,3] W?	[2,3]	[3,3]	[4,3]
[1,2] Ok S	[2,2] W? P?	[3,2]	[4,2]
[1,1] ok	[2,1] Ok B	[3,1] P?	[4,1]

شکل ۹: وضعیت عامل در اتاق [۲, ۱]

[1,4]	[2,4]	[3,4]	[4,4]
[1,3] W	[2,3]	[3,3]	[4,3]
[1,2] Ok S	[2,2] Ok	[3,2]	[4,2]
[1,1] Ok	[2,1] Ok B	[3,1] P	[4,1]

شکل ۱۰: امن بودن اتاق [۲, ۲]

عامل در شروع بازی ادراک [None, None, None, None, None] را دریافت می‌کند. وضعیت لحظه‌ای عامل در شکل ۷ آورده شده است. عامل با حرکت به‌اتاق [۱, ۲]، ادراک [Stench, None, None, None, None] را دریافت می‌کند و با حرکت به‌اتاق [۲, ۱]، ادراک [None, Breeze, None, None, None] را دریافت می‌کند. بنابراین در اتاق [۱, ۲]، امکان وجود ومپوز در یکی از اتاق‌های

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

همجوار آن مشخص می‌شود (به‌شکل ۸ توجه کنید)، و در اتاق [۱ ,۲] امکان وجود تله در اتاق‌های هم‌جوار مشخص می‌شود (به شکل ۹ توجه کنید). با توجه به عدم وجود Stench در اتاق [۲ ,۱]، در اتاق [۲ ,۲] ومپوز وجود ندارد. با توجه به عدم وجود Breeze در اتاق [۱ ,۲]، در اتاق [۲ ,۲] تله وجود ندارد. پس ومپوز در اتاق [۱ ,۳] و تله در اتاق [۳ ,۱] وجود دارد. همچنین اتاق [۲ ,۲] یک اتاق امن است (به شکل ۱۰ توجه کنید).

در ادامه عملکرد عامل منطقی با استفاده از منطق گزاره‌ای بررسی می‌شود. فرمول‌های اتمی را به‌صورت زیر در نظر بگیرید:

- فرمول اتمی $B_{i,j}$ جمله «در اتاق $[i,j]$ نسیم دریافت می‌شود.» را بیان می‌کند.
- فرمول اتمی $S_{i,j}$ جمله «در اتاق $[i,j]$ بوی نامطبوع دریافت می‌شود.» را بیان می‌کند.
- فرمول اتمی $P_{i,j}$ جمله «در اتاق $[i,j]$ تله وجود دارد.» را بیان می‌کند.
- فرمول اتمی $W_{i,j}$ جمله «در اتاق $[i,j]$ ومپوز وجود دارد.» را بیان می‌کند.

با استفاده از فرمول‌های اتمی فوق، اصول مربوط به پایگاه دانش دنیای ومپوز با فرمول‌هایی در منطق گزاره‌ای بیان می‌شود.
در شروع حرکت اتاق [۱ ,۱] یک اتاق امن است و فرمول‌های $\neg B_{۱,۱}$ و $\neg S_{۱,۱}$ درست هستند. از طرفی طبق اصول پایگاه دانش فرمول‌های زیر درست هستند:

$$S_{۱,۱} \leftrightarrow W_{۱,۲} \vee W_{۲,۱}$$

$$B_{۱,۱} \leftrightarrow P_{۱,۲} \vee P_{۲,۱}$$

بنابراین مجموعه $KB_۱ = \{\neg B_{۱,۱}, \neg S_{۱,۱}, S_{۱,۱} \leftrightarrow W_{۱,۲} \vee W_{۲,۱}, B_{۱,۱} \leftrightarrow P_{۱,۲} \vee P_{۲,۱}\}$ اطلاعات پایگاه دانش در این لحظه است. می‌توان دید که $KB_۱ \models \neg P_{۱,۲} \wedge \neg W_{۱,۲} \wedge \neg P_{۲,۱} \wedge \neg W_{۲,۱}$. این هم به‌معنای امن بودن اتاق‌های [۱ ,۲] و [۲ ,۱] است.

بنابراین به‌طور مشابه استنتاج انجام گرفته توسط عامل منطقی در چند مرحله توصیف شده برای رسیدن به وجود ومپوز در اتاق [۱ ,۳]، تله در اتاق [۳ ,۱] و امن بودن اتاق [۲ ,۲]، به‌صورت استنتاج‌های منطقی زیر است:

- اگر $KB_۲ = \{S_{۱,۲}, S_{۱,۲} \leftrightarrow W_{۱,۳} \vee W_{۳,۱}, \neg S_{۲,۱}, S_{۲,۱} \leftrightarrow W_{۲,۲} \vee W_{۳,۱}, \neg B_{۱,۲}, B_{۱,۲} \leftrightarrow P_{۱,۳} \vee P_{۲,۲}\}$، آنگاه $KB_۲ \models W_{۱,۳} \wedge \neg P_{۲,۲}$.
- اگر $KB_۳ = \{B_{۲,۱}, B_{۲,۱} \leftrightarrow P_{۳,۱} \vee P_{۲,۲}, \neg B_{۱,۲}, B_{۱,۲} \leftrightarrow P_{۲,۲} \vee P_{۱,۳}, \neg S_{۲,۱}, S_{۲,۱} \leftrightarrow W_{۳,۱} \vee W_{۲,۲}\}$، آنگاه $KB_۳ \models P_{۳,۱} \wedge \neg W_{۲,۲}$.

پس

$$KB_۱ \cup KB_۲ \cup KB_۳ \models P_{۳,۱} \wedge \neg W_{۲,۲} \wedge W_{۱,۳} \wedge \neg P_{۲,۲}$$

بنابراین با مجموعه اطلاعات $KB_۱$، $KB_۲$ و $KB_۳$ وجود ومپوز در اتاق [۱ ,۳] ($W_{۱,۳}$)، وجود تله در اتاق [۳ ,۱] ($P_{۳,۱}$) و امن بودن اتاق [۲ ,۲] ($\neg W_{۲,۲} \wedge \neg P_{۲,۲}$) مشخص می‌شود.

۶. جمع‌بندی

در این مقاله منطق گزاره‌ای، به‌صورت ریاضی و کاربردی به‌طور مختصر توصیف شده است. با توجه به‌اینکه در اکثر علوم مورد استفاده توسط بشر استدلال‌هایی براساس اطلاعات وجود دارد، استفاده منطق در بسیاری از علوم اجتناب‌ناپذیراست. استفاده از عامل‌های منطقی مختلف در زمینه‌های مختلف به‌تنهایی اهمیت آن را روشن می‌سازد. هم‌چنین در علوم نظری و کاربردی مانند

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

شیمی و مدیریت و غیره می‌توان از منطق گزاره‌ای برای پرهیز از اشتباه در استنتاج استفاده کرد [۵]. پیدا کردن الگوریتم‌های کارآمدتر نسبت به الگوریتم‌های موجود یک زمینه برای بهبود نتایج موجود است.

مراجع

[۱] م. اردشیر، *منطق ریاضی*، مؤسسه‌ی انتشارات هرمس، تهران، ۱۳۸۹.

[۲] ر. اسمولیان، *معماهایی در منطق ریاضی*، مترجم م. شریف‌زاده، انتشارات فاطمی، تهران، ۱۳۶۶.

[۳] ه. ب. اندرتون، *آشنایی با منطق ریاضی*، مترجمان غ. برادران خسروشاهی و م. رجبی طرخورانی، مرکز نشر دانشگاهی، تهران، ۱۳۹۰.

[4] T. Dohmke, https://github.com/aimacode/aima-python/blob/master/logic.py.

[5] A. Mosley and E. Baltazar, *An introduction to logic: From every day life to formal systems*, Independently published, 2019.

[6] S. Russell and P. Norvig , *Artificial Intelligence: A Modern Approach*, Pearson; 4th edition, 2021.

[7] U. Schőning, *Logic for computer scientists*, Birkhäuser Boston, 2008.

سمیه تاری

گروه ریاضی، دانشگاه شهید مدنی آذربایجان، تبریز

s_tari @azaruniv.ac.ir

سمیه تاری، عضو هیات علمی دانشگاه شهید مدنی آذربایجان و دانش آموخته دکترای ریاضی محض، گرایش منطق ریاضی از دانشگاه تبریز است. علائق پژوهشی وی در منطق ریاضی و کاربردهای آن، نظریه مدل و ساختارهای ترتیب کمینه است.

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492

Some Applications of Propositional Logic

Somayyeh Tari

Abstract: The appearance of most of mathematical theories is the consequence of thinking and reasoning. Different mathematical models are used for modeling the deductive reasoning. Propositional logic is one of the mathematical models of deductions that has been used in many fields of applied science. In this paper, propositional logic will be investigated and some applications of it will be described.

Keywords: Propositional Logic, Boolean Function, Logical Agent, Truth Value.

Somayyeh Tari

Department of Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran.

Email: s_tari@azaruniv.ac.ir

Communicated by Soghra Nobakhtian.

Article Type: Review Paper.

Received: 30/01/2022, Accepted: 28/08/2022.

DOI: http://dx.doi.org/10.22108/MSCI.2022.132540.1492