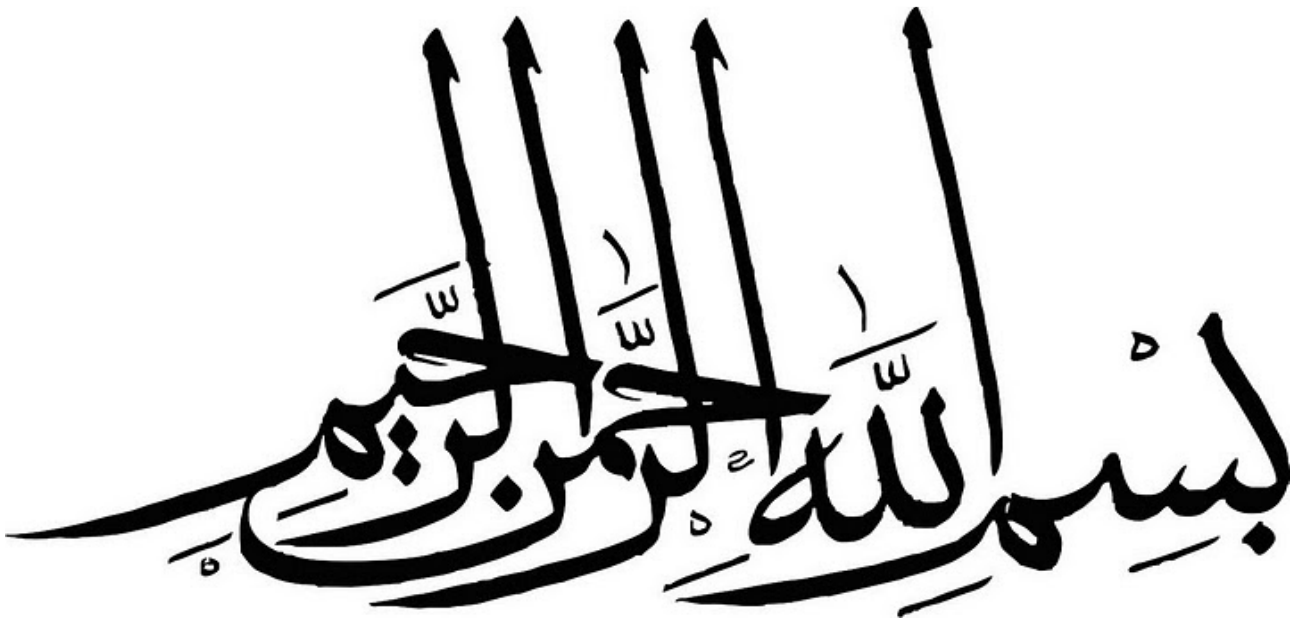

بسم الله الرحمن الرحيم

دانشکده فنی و مهندسی

پایان نامه دوره کارشناسی ارشد رشته مهندسی برق الکترونیک

عنوان پایان نامه:

به دست آوردن مرتبه مؤثر متغییرهای گراف BDD با استفاده از مفهوم انتروپی و ساختارهای ID۳ و C۴.۵

استاد راهنما:

دکتر محمد باقر غزنوی قوشچی

نگارنده:

میثاق تکاپو

پاییز ۱۳۸۷

کلیه حقوق این پایان نامه متعلق به دانشگاه شاهد می باشد.

تقدیم به پدر و مادر عزیزم که همواره مشوق و پشتیبان من در سخت ترین شرایط زندگیم بوده اند.

از استاد عزیزم جناب آقای دکتر محمد باقر غزنوی قوشچی که همواره با ممارست و پیگیری هایشان دلگرمی و با اخلاق خوب و ممتازشان به ما درس فروتنی داده اند تشکر و قدردانی می کنم. از ایشان همچنین به علت آشنا کردن من با شیوه های جستجو در اینترنت که همواره در کل زندگی من تاثیر بسیار مثبتی داشته است تشکر و سپاسگذاری می کنم. همچنین از جناب آقایان دکتر عبدالرضا نبوی و دکتر حامد ساجدی به خاطر اعتمادی که به اینجانب و استاد عزیزم داشته اند قدردانی و تشکر میکنم.

چکیده

امروزه با پیشرفت علوم الکترونیک و کامپیوتر و همچنین پیچیده تر شدن این علوم نیاز بیش از پیش به استفاده از نرم افزار ها و ابزارهای کمک طراحی که در اصطلاح CAD گفته میشوند، احساس می شود.

گراف های تصمیم باینری(BDD)، ساختار داده ای برای توابع منطقی سویچینگ می باشند که برای پیاده سازی بسیاری از توابع منطقی، مورد استفاده قرار می گیرند. مهمترین مسئله در این نوع از گراف ها و نوع خاصی از آنها که بیشترین کاربرد را دار هستند، یعنی گراف های تصمیم مرتبه ای(OBDD)، تعداد گره های این گراف ها می باشد. در گراف های OBDD، با تغییر مرتبه گراف ابعاد گراف به ازاء یک تابع ثابت، ممکن است تغییر کند. پیدا کردن مرتبه موثر در این نوع از گراف ها که منجر به پیدایش کوچکترین گراف از نظر ابعاد می شود با توجه به امکان پیاده سازی مستقیم این گراف ها به صورت ساختارهای ترانزیستوریو مباحث مصرف توان، به یکی از داغ ترین مباحث در این حوزه از علم تبدیل شده است.

الگوریتم ID۳ یکی از الگوریتم های تصمیم گیری کلاسیک در حوزه حل مسائل تصمیم گیری هوشمند می باشد. الگوریتم C۴.۵ ، در واقع همان الگوریتم ID۳ است که برای داده ای پیوسته تغیراتی در آن داده شده است. در این پایان نامه برای کاهش تعدا گره ها و بدست آوردن مرتبه موثر گراف های OBDD از این دو الگوریتم استفاده کردیم.

در ادامه تحقیقات، به اثر الگوریتم ID۳ برای بهینه سازی الگوریتم ژنتیک، که خود به عنوان الگوریتمی برای کاهش گره ها استفاده شده است[۱]، به عنوان یک عملگر پیش پردازش پی میبریم.

نتایج حاصل از این تحقیقات در مقاله ای جمع آوری شد که در کنفرانس ECTI-۲۰۱۰ مورد پذیرش قرار گرفت.

فهرست مطالب

عنوان | صفحه

فصل اول : نظریه گراف ها ۱

۱ - ۱مقدمه ۱

۲ - ۱ انواع گرافها ۳

۱- ۲ - ۱ گرافهای بدون جهت ۳

۲ - ۲ - ۱ گرافهای جهت دار ۴

۳ - ۲ - ۱ گرافهای بسته ۵

۴ - ۲ - ۱ گرافهای باز ۵

۵ - ۲ - ۱ گرافهای جهت دار باز ۶

فصل دوّم : گراف های تصمیم باینری ۹

۱ - ۲ مقدمه ..... ۹.

۲ - ۲ دیاگرام های تصمیم باینری BDDS ۱۰

۳ - ۲ انواع گراف های تصمیم باینری ۱۳

۱ - ۳ - ۲ OBDD ۱۳

۲ - ۳ - ۲ ROBDD ۱۴

۳ - ۳ - ۲ SBDD ۱۶

۴ - ۳ - ۲ ZBDD ۱۸

۵ - ۳ - ۲ FBDD ۱۹

۶ - ۳ - ۲ FDD ۲۱

۴ - ۲ جا به جایی متغییرها ۲۳
۵ - ۲ روش تخصیص دادن وزن دینامیک ۲۶
فصل سوّم : کاربرد گرافهای BDD ۳۰
۱ - ۳ نگاشت یک گراف BDD بر روی ساختار PTL ۳۰
۱ - ۱ - ۳ منطق SPL ۳۱
۲ - ۱ - ۳ منطق CPL ۳۲
۲ - ۳ BDD ها با لبه های مکمل ۳۳
۳ - ۳ قانون المور ۳۴
۴ - ۳ یک الگوریتم ساده برای قرار دادن بافر ۳۵
۵ - ۳ تجزیه منطق برپایه BDD ۳۷
۱ - ۵ - ۳ تجزیه بر پایه AMD و OR بولی ۳۷
۲ - ۵ - ۳ تجزیه بر پایه XOR ۳۸
۳ - ۵ - ۳ تجزیه بر پایه مالتی پلکسر ۳۹
۴ - ۵ - ۳ تجزیه توابع با چند خروجی ۴۱
فصل چهارم : ID۳ ..... ۴۴
۱ - ۴ مقدمه ۴۵
۲ - ۴ تاریخچه ۴۵
۱ – ۲ – ۴ ظهور ID۳ ..... ۴۶
۲ – ۲ – ۴ کاربرد ID۳ در بازی شطرنج ۴۶
۳ - ۴ یادگیری به وسیله درخت های تصمیم ۴۷

۴ - ۴ عمل استنتاج ۴۹
۵ - ۴ تعاریف پایه ID۳ ..... ۵۳
۶ - ۴ الگوریتم ID۳ ..... ۵۸
۷ - ۴ نرخ تقویت ۶۰
۸ - ۴ الگوریتم C۴.۵ ..... ۶۱
۹ - ۴ مزایا و معایب ID۳ ..... ۶۲
فصل پنجم : معرفی نرم افزارها ۶۴
۱ - ۵ معرفی نرم افزارهای به کار رفته به عنوان تحلیلگر گراف های BDD ۶۵
۱ – ۱ – ۵ نرم افزار BeDD ۶۵
۱ - ۱ - ۱ - ۵ مقدمه..... ۶۵
۲ - ۱ - ۱ - ۵ خصوصیات اصلی ۶۵
۳ - ۱ - ۱ - ۵ محیط اجرایی..... ۶۶
۴ - ۱ - ۱ - ۵ پیاده سازی یک SBDD ۶۸
۵ - ۱ - ۱ - ۵ جابه جایی متغییرها با استفاده از الگوریتم ژنتیک..... ۷۰
۲ - ۱ - ۵ اپلت مبدّل BDD به مدار سنتز شده با منطق PTL ۷۱
۱ - ۲ - ۱ - ۵ ایجاد گراف ها ۷۱
۲ - ۲ - ۱ - ۵ ایجاد مدار سنتز شده..... ۷۳
۳ - ۱ - ۵ نرم افزار TT۲BDD ۷۶
۴ - ۱ - ۵ کتابخانه BuDDY ۷۹
۵ - ۱ - ۵ C۴.۵ ..... ۸۰

۲ - ۵ نرم افزار Espresso ۸۳

فصل ششم : نتایج پیاده سازی ۸۶

۱ - ۶ مقایسه مرتبه پیشنهادی ID۳ با روش های باز آرایی گراف BDD ۸۶

۲ - ۶ استفاده از الگوریتم ID۳ به عنوان یک پیش پردازشگر در الگوریتم ژنتیک ۹۱

فصل اول

نظریه گراف ها

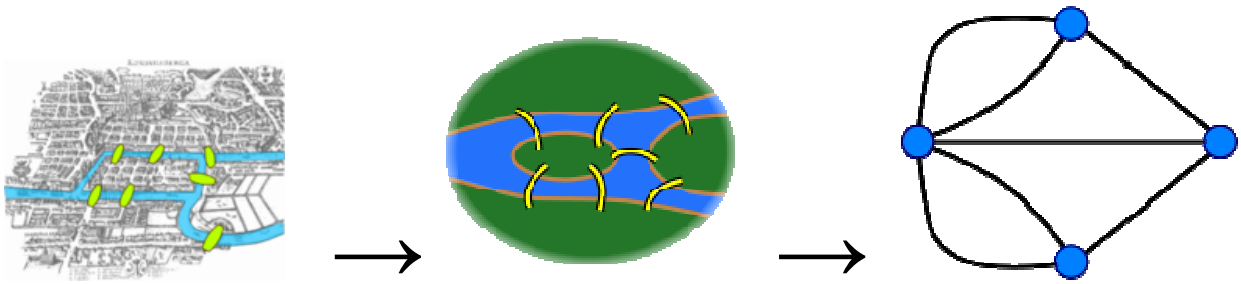
۱-۱ مقدمه

یکی از شاخه های مهم علم ریاضی که در تمامی علوم نظیر علوم مهندسی، علوم زیست شناسی، علوم فیزیک، ژنتیک وغیره کاربرد دارد، استفاده از نظریه گراف می باشد. نظریه گراف اولین بار در سال ۱۷۳۶ توسط دانشمندی به نام لئونارد اویلر[۱] درمقاله ای که براساس مسئله هفت پل کنیزبرگ که در شکل ۱-۱ نشان داده است، منتشر شد، پا به عرصه وجود گذاشت [۱]. فرمولهایی که اویلر در مقاله اصلی خود ارائه داده بود، شامل معرفی مفاهیمی از قبیل تعداد لبه ها، رئوس و همچنین اشکال یک چند وجهی محدب[۲] بود. ارائه این مفاهیم توسط اویلر و همچنین مطالعه و بررسی اشکال گراف، توسط دو نفر به نامهای آگوستین چآو[۳] [۲] و سیمون آنتونیو هیولیر[۴] [۳]، پایه و اساس مفهوم توپولوژی[۱] قرار گرفت. بیش از

۱-Leonhard Euler ۲-Faces of a convex polyhedron ۳-Augustin Cauchy ۴-Simon Antoine Jean L'Huillier ۵- Topology

یک قرن پس از معرفی مفاهیم گراف و توپولوژی، دانشمندی به نام آرتور کایلی[۱] با مطالعه بر روی روش تحلیلی خاصی که ناشی از معادلات دیفرانسیل بود، به رده خاصی از گراف ها، به نام درخت ها، رسید. مطالعات این شخص با مباحث نظریه شیمی بسیار مطابقت داشت. کایلی سپس نتایج حاصله از مفاهیم درخت را با مفاهیم ساختار های شیمیایی زمان خود مرتبط کرد. اتصال این دو مفهوم که یکی از علم ریاضی و دیگری از علم شیمی بود، به پایه گذاری اصطلاحات استاندارد نظریه گراف منجر گردید. در نهایت مفاهیم گراف به صورت یک استاندارد کامل در سال ۱۸۷۸ توسط جیمز جوزفه سیلوستر[۲] [۴] در قالب یک مقاله معرفی و به چاپ رسید [۵].

ازمهمترین مثال های موجود در استفاده از گراف در تحلیل های جبری مدرن می توان به قوانین کریشوف که در سال ۱۸۴۵ توسط این فیزیکدان ارائه شد و یا دیاگرامهای جریان سیگنال[۳] و موارد دیگر اشاره نمود. امروزه گراف یکی از مناسبترین ابزارهای ریاضی برای حل کردن مسائل مختلف شبکه ای، از قبیل (شبکه های کامپیوتری، شبکه های حمل و نقل و....) می باشد. در واقع از گراف ها می توان در تحلیلگرهای شبکه[۴] استفاده کرد. از مهم ترین موارد استفاده از گراف ها



شکل۱-۱- مسئله هفت پل کنیزبرگ که باعث ابداع نظریه گراف ها شد

استفاده در علوم مهندسی برق و کامپیوتر می باشد که در فصل های بعد به موارد استفاده از آنها بخصوص در مدارهای دیجیتال پرداخته می شود.

۱- Arthur Cayley ۲- James Joseph Sylvester ۳- Signal Flow Diagrams ۴- Network Analyzers

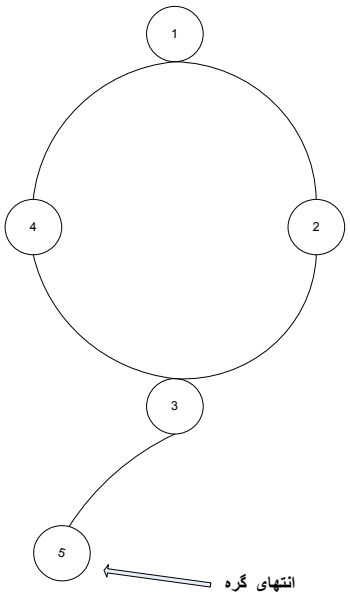
۲-۱ انواع گراف ها

گراف ها از لحاظ جهت دار بودن و بدون جهت بودن به ۲ دسته تقسیم می شوند:

۱-۲-۱ گراف های بدون جهت[۱]

یک گراف بدون جهت G با دو عامل V و G به صورت مقابل تعریف می شود: $G=(V,E)$.

که در آن V مجموعه رئوس یا گره ها[۲] می باشد و E مجموعه خطوط ساده مرتبط به گره ها می باشد که به آن لبه[۳] می گویند. گره هایی که تنها به یک لبه یا خط وصل باشند را در این تعریف گره انتهایی می گویند. تعریف ،مرتبه گراف عبارت است از تعداد گره های آن، اندازه گراف عبارت است از تعداد لبه ها و در نهایت درجه هر گره عبارت است از تعداد



شکل۲-۱- نمونه ای از گراف های بدون جهت

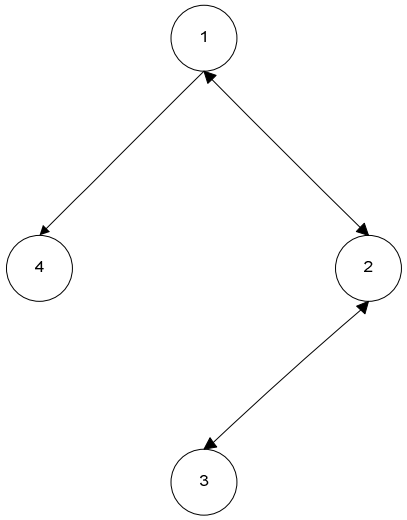
۱-Undirected ۲-Vertices ۳-Edges

گره های دیگر که به این گره متصل می باشند. شکل۲-۱ نمونه یک گراف بدون جهت را نشان می دهد.

۲-۲-۱ گراف های جهت دار[۱]

یک گراف جهت دار در مقابل یک گراف بدون جهت به صورت مقابل تعریف می شود: $G = (V, A)$.

که در آن V مجموعه گره ها و A مجموعه خطوط واصل می باشد. در تعریف گراف های جهت دار، این خطوط با توجه به اینکه یک طرفه یا دو طرفه باشند، دارای درجه هستند و به خطوط واصل بین گره ها پیکان[۲] می‌گویند (که با توجه به جهت و مرتبه هر پیکان به آنها پیکان‌های دو طرفه یا یک طرفه می گویند). در این تعریف گره‌ای که جهت پیکان بسوی آن است را پذیرنده[۳] جهت و گره ای که پیکان از آن خارج می‌شود را جهت ماقبل[۴] می گویند. شکل ۳-۱ نمونه یک گراف جهت دار را نشان می دهد.



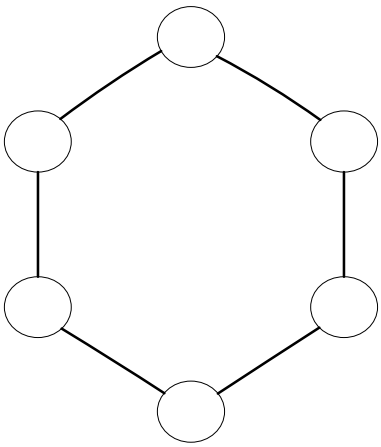
شکل۳-۱- نمونه ای از گراف جهت دار

۱-Directed ۲-Arrows ۳-Successor ۴-Predecessor

گراف ها از لحاظ بسته و باز بودن به ۲ قسمت تقسیم می شوند:

۳-۲-۱گراف های بسته[۱]

گراف های بسته به گراف هایی گفته می شوند که دارای یک زنجیره اتصالات بسته از گره ها هستند. این گراف ها را با C_n نشان می دهند. n در اینجا تعداد رئوس می باشد. هر راس در این گراف ها از درجه ۲ می باشد، یعنی اینکه در این نوع از گراف ها هر راس تنها با دو راس مجاور خود در تماس می باشد. شکل۴-۱ نمونه یک گراف بسته را نشان می دهد.

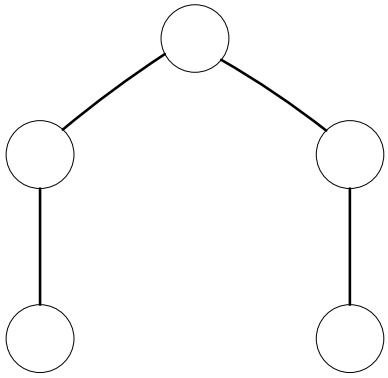


شکل۴-۱- نمونه یک گراف بسته

۴-۲-۱گراف های باز[۲]

گراف های باز دارای هیچ گونه اتصالات زنجیره ای به شکل بسته نیستند. در واقع گراف های باز دارای هیچ گونه مسیر برگشت پذیر به یک راس که شامل تمامی رئوس باشد، نمی باشند. شکل۵-۱ نمونه یک گراف باز را نشان می دهد[۶].

۱-Acyclic ۲-Cyclic

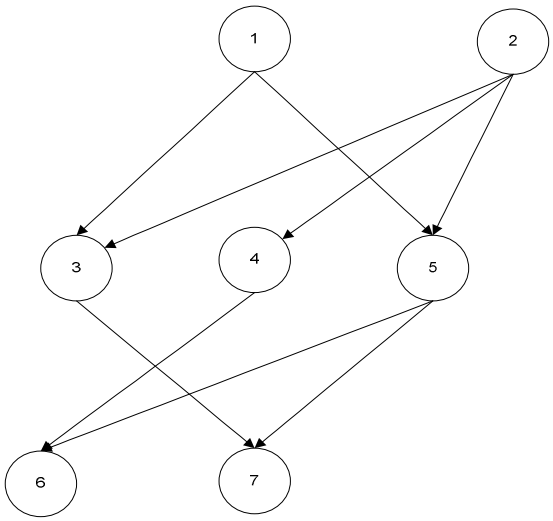


شکل۵-۱- نمونه ای از یک گراف باز

۵-۲-۱گراف های جهت دار باز[۱]

گراف های جهت دار باز، گراف هایی هستند که دارای دو مشخصه باز بودن و جهت دار بودن می باشند. از خصوصیات ویژه این دسته از گراف ها آن است که نمی توانند یک لبه جهت دار از یک گره به همان گره رادارا باشند. در این نوع از گراف ها حداقل یک گره مبداء (که به آن ورودی وارد نمی شود و تنها خروجی دارد) و یک گره مقصد (که هیچ گره ای از آن خارج نمی شود و فقط ورودی دارد) وجود دارد. به این نوع از گراف ها در اصطلاح DAG می گویند. در شکل۶-۱ یک نمونه از انواع گراف های جهت دار باز نشان داده شده است. درگراف های جهت‌دار باز با تعداد رئوس محدود، طول گراف برابر است با طولانی ترین راه مستقیم از گره مبداء تا گره انتهایی[۶]. علت استفاده از این نوع گراف ها، بخصوص در محاسبات و طراحی های کامپیوتری در آن است که، با استفاده از این نوع گراف ها می توان در موارد متعددی به ساده سازی و همچنین مدیریت گراف ها و طراحی‌هایی که بر مبنای اصول گراف ها می باشند پرداخت. با دقت در ساختار گراف های جهت‌دار باز می‌توان ملاحظه کرد که این گروه را می توان در ساختارهای درختی عمومیت داد و در این حالت آنها را به زیر درختانی که قابلیت اشتراک با زیر درختان دیگر را داشته باشند، تبدیل کرد. در این حالت ما با سیستمی که دارای یکسری ساختار

۱-Directed Acyclic Graph (DAG)



شکل۶-۱- نمونه ای از گراف های جهت دار باز

زیر درخت ایده ال است رو به رو می شویم. با تبدیل هر قسمت به این زیر درختان، امکان ذخیره سازی ساختارها و کاهش فضای ذخیره سازی بشدت افزایش می یابد [۶]. یکی از انواع گراف های جهت‌دار باز که امروزه کاربرد آنها برای طراحی سیستمهای دیجیتال بسیار گسترش یافته است، گراف های دیاگرام تصمیم باینری[۱] می باشند.

ایده اولیه دیاگرام های تصمیم باینری (BDDs) اولین بار توسط دانشمندی به نام لی[۲] [۷] و پس از آن توسط آکرز[۳] ارائه شد [۸]، اما روشهای موثر برای برای به کار گیری گراف های (BDD) اولین بار توسط برایانت[۴] [۹] توسعه و بسط داده شدند. به خاطر مناسب و موثر بودن نمایش و تحلیل توابع منطقی در قالب BDD، امروزه سهم قابل توجهی در زمینه تحلیل و نمایش توابع منطقی به دیاگرام های تصمیم باینری تعلق دارد. همچنین کوششهای بسیاری برای افزایش بهره وری این گراف ها در زمینه کاهش استفاده از فضای حافظه و زمان اجرای آنها انجام شده که این تلاشها منجر به ارائه خانواده دیگری از بین گراف ها به نام دیاگرام های تصمیم باینری مشترک[۵] یا (SBDD) شد که با اتصال گره های

۱-Binary Decision Diagram ۲-C. Y. Lee ۳-S.B Akers ۴- R. E. Bryant ۵-Shared BDD

مشترک گروهی از BDD ها به یکدیگر در یک گراف واحد، آنها رامرتب می کرد. با گسترش کاربردهای این گراف ها، بسته-های نرم افزاری مختلفی برای استفاده این گراف ها در کاربردهای مختلف، بخصوص در سیستمهای CAD[۲] وVLSI [۱] ارائه شده است.

با وجود گسترش بسیار زیادی که BDD ها در به کاربری توابع منطقی یافته است، چند نکته در مورد این گراف ها وجود دارد که در کاربردهای عملی باید به آنها توجه کرد. یکی از این نکات مرتبه[۳] متغیر ها می‌باشد. BDD های رایج نیاز دارند که مرتبه متغیرهای ورودی آنها ثابت باشند. از طرفی هم ابعاد BDD ها وابستگی زیادی به این مرتبه دارد. پیدا کردن بهترین مرتبه که ابعاد را کاهش دهد، بسیار مشکل می باشد. در عمل، استفاده از الگوریتم‌های کاهش مرتبه متغیرها، یکی از گسترده ترین نواحی استفاده در کاربردی کردن BDD ها می باشد. در باره مفهوم مرتبه متغیرها و جابه‌جائی[۴] آنها در فصل دوم توضیح داده شده است. نکته دیگری که در رابطه با این گراف‌ها باید در نظر گرفت وجود حالت های بی‌اهمیت[۵] در طراحی ها می باشد. چون BDD های معمول فقط با منطق های دوتایی سر و کار دارند این موضوع را می توان به صورت نمایش منطق های چند مقداره یا با تابع های صحیح، تعمیم داد[۱۰].

۱-Very Large Scale Integrated ۲-Computer Aided Design ۳-Order ۴-Reordering ۵-Don’t Care

فصل دوم

گراف های تصمیم باینری

۲-۱ مقدمه

به کار گیری توابع منطقی یکی از بخش‌های مهم در علوم کامپیوتر می باشد. بسیاری از مشکلات در طراحی و تست سیستمهای دیجیتال می‌توانند به صورت ترتیبی[۱] از عملکردها[۲] در توابع منطقی بیان شوند. پیشرفت های سریع در (VLSI) باعث افزایش این حجم و پیچیدگی طراحی و تحلیل سیستمها به صورت دستی شده است و در نتیجه نیاز به استفاده روز افزون از نرم افزارهای کمک طراحی توسط کامپیوتر یا (CAD) شده است. کارایی این سیستم ها، ابزارهای (CAD)، در حوزه طراحی، سنتز و بازبینی سیستم های دیجیتال، بخصوص در حوزه‌های (VLSI)و طراحی‌های کامپیوتری بستگی بسیار زیادی به استفاده مطلوب از تابع های منطقی که استفاده می کنیم دارد [۱۰]. روشهای کلاسیک مختلفی برای ارائه و به کار بردن توابع منطقی توسعه داده شده اند مانند جدول صحت[۳] و جدول کارنو[۴]. اما بزرگترین مشکل این روشهای کلاسیک اشغال فضای حافظه (2^n بیتی) به ازاء n ورودی می‌باشد و در نتیجه برای پیاده سازی ساده ترین تابع در این روشها، نیاز به حافظه های بسیار بزرگ (متناسب با توان نمایی

۱-Sequence ۲-Operations ۳-Truth Table ۴-Karno Map

ورودی ها) و در نتیجه رفتن به سمت زمانهای شبیه سازی نمایی می باشیم که باعث می شود که ما به تامل دراستفاده از این توابع کلاسیک بپردازیم. بنابراین به خاطر وجود این مسائل، و عدم استفاده و کارایی روشهای کلاسیک در کاربردهای مقیاس بالا، ما مجبور به توسعه روشهای کارآمدی برای استفاده در کاربردهای عملی می شویم.

پس از معرفی که توسط برایانت از دیاگرام تصمیم باینری شد، تحقیقات گسترده ای حول این موضوع انجام شد و امروزه انواع مختلفی از این گراف ها به وجود آمده اند که در ادامه به بعضی از آنها خواهیم پرداخت.

۲-۲ دیاگرام های تصمیم باینری[۱] (BDDs)

یک BDD در واقع یک گراف جهت دار مستقیم بدون چرخش با گره هایی دو بعدی است که از یک نقطه کل گراف شروع می شود و در این حالت گفته می شود که BDD یک گراف جهت دار بدون چرخش با یک ریشه می-باشد[۲]. به این گره ها، گره با منطق "۰" و گره با منطق "۱" می گوییم. هر گره دارای دو لبه خروجی نیز می باشد که به آنها لبه "۰" و لبه "۱" می گوییم. شکل ۱-۲ نمونه یک گراف BDD را نشان می دهد.

یک (OBDD)[۳] در واقع یک گراف BDD است که متغیر های ورودی آن در تمام شاخه های گراف دارای مر-تبه ثابتی هستند و هیچ متغیری بیش از یک بار در یک مسیر ظاهر نمی شود[۱۰]. در این گراف های باینری، ترمینالهای خروجی ۰ و ۱ نمایانگر منطق "۰" و"۱" می باشند و هر گره بیانگر توصیف شانون[۴] از تابع منطقی می باشد. در واقع می توان گفت که اساس تفکری که موجب ایجاد این شاخه از گراف ها شده است، همان بسط شانون است.

عبارت بسط شانون را به صورت زیر بیان می کنند:

$$f = v.f_{v} + \overline{v}.f_{\overline{v}} \qquad (۲-۱)$$

۱-Binary Decision Diagrams ۲-Rooted Directed Acyclic Graph (R-DAG) ۳-Order BDD ۴-Shannon