**سادهساز جدول کارنو**

Karnaugh( با هدف سادهسازی عبارات بولی با استفاده از تکنیک جدول کارنو++C این برنامه متغیر ورودی است.۵ تا۲ ) طراحی شده است. این برنامه قادر به پردازش عبارات باMap ) مربوط به تابع بولی را به عنوان ورودی ارائه میدهند و برنامهMinterms( کاربران مینترمهای) راSum of Products - SOP( در خروجی، عبارت بولی سادهشده در فرم حاصلضرب مجموعنمایش میدهد.

**ویژگیهای کلیدی برنامه:**

|  |  |  |
| --- | --- | --- |
| ،۲ برنامه قادر است جداول کارنو را برای تعداد متغیرهای ورودی | **متغیر:۵ تا۲ پشتیبانی از** | •  • |
| متغیر بسازد و سادهسازی کند.۵ و۴ ،۳**:ورودی مینترم**۱ کاربران میتوانند مستقیما ً مینترمهایی را که مقدار تابع خروجی آنها | |

است، وارد کنند.

|  |  |  |
| --- | --- | --- |
| برای درک بهتر فرآیند سادهسازی، برنامه جدول کارنو را به | **نمایش بصری جدول کارنو:** | • |

همراه مقادیر ورودی (مینترمها) نمایش میدهد.

|  |  |  |
| --- | --- | --- |
| پس از اعمال الگوریتم سادهسازی، برنامه عبارت بولی | **سادهشده:SOP خروجی عبارت** | • |

) تولید و نمایش میدهد.SOP( سادهشده را به صورت حاصلضرب مجموع

**نحوه کامپایل برنامه:**

مانند ++C برای کامپایل این برنامه، شما نیاز به یک کامپایلر++g دارید که به طور معمول بر) موجود است. اگر از ویندوز استفادهmacOS ،روی سیستمهای یونیکسمانند (لینوکس استفاده کنید.++g برای نصب کامپایلرCygwin یاMinGW میکنید، میتوانید از

|  |  |  |  |
| --- | --- | --- | --- |
| . ذخیره کنید | کد برنامه را در فایلی با نام main.cpp | **ذخیره کد منبع:** | .1 |
| به دایرکتوری که فایل main.cpp را در آن ذخیره**باز کردن ترمینال یا خط فرمان:** | | | .2 |

کردهاید، بروید.

دستور زیر را در ترمینال وارد کنید:**اجرای دستور کامپایل:** .3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | bash g++ main.cpp -o kmap | | | | | |  |
| .++C : نام کامپایلر | | | | |  | | --- | | ++g | | ◦ |
| .: نام فایل حاوی کد منبع | main.cpp | | | | ◦ |
| : این بخش به کامپایلر میگوید که فایل اجرایی خروجی را با نام | | o kmap- | | | ◦ |
| . ایجاد کند. شما میتوانید نام دلخواه دیگری را انتخاب کنید | | | kmap | |  |

پس از اجرای موفقیتآمیز این دستور، یک فایل اجرایی به نام kmap (یا نامی که انتخابکردهاید) در همان دایرکتوری ایجاد خواهد شد.

**نحوه اجرای برنامه:**

پس از کامپایل موفقیتآمیز، برنامه را میتوانید از طریق خط فرمان اجرا کنید.

|  |  |  |
| --- | --- | --- |
| دستور زیر را در ترمینال وارد کنید: | **اجرای فایل اجرایی:** | .1 |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | bash ./kmap | | |  |
| .: نشاندهنده دایرکتوری فعلی است | |  | | --- | | /. | | ◦ |

پس از اجرای برنامه، مطابق با گردش کار زیر از شما ورودی خواسته خواهد شد:

|  |  |  |
| --- | --- | --- |
| ابتدا برنامه از شما میخواهد که تعداد متغیرهای ورودی را وارد کنید. شما | **تعداد متغیرها:** | .1 |

را وارد کنید.۵ تا۲ میتوانید عددی بین

|  |  |  |
| --- | --- | --- |
| پس از تعیین تعداد متغیرها، برنامه شروع به درخواست مینترمها | **ورود مینترمها:** | .2 |

میکند. مینترمها اعدادی هستند که نشاندهنده اندیسهایی در جدول کارنو هستند که است.۱ مقدار خروجی تابع در آنها برابر با

|  |  |  |
| --- | --- | --- |
| برای هر مینترم، برنامه از شما میخواهد که آن را وارد کنید. | | ◦ |
|  | | ◦ |
| . را وارد کنید | برای پایان دادن به فرآیند ورود مینترمها، مقدار 1- |

**مثال نحوه استفاده:**

) را ساده کنیم و مینترمهای مربوط به آنA, B, C( متغیر۳ فرض کنید میخواهیم یک تابع با.3 ،1 ،0 عبارتند از

|  |
| --- |
| 3 :)۵-۲( تعداد متغیرها را وارد کنید |

|  |
| --- |
| . را وارد کنید1- ). برای پایان۷ تا۰( مینترمها را وارد کنید |

|  |
| --- |
| 0 :مینترم |

|  |
| --- |
| 1 :مینترم |

|  |
| --- |
| 3 :مینترم |

|  |  |
| --- | --- |
|  | 1- :مینترم |
| SOP ، برنامه جدول کارنو را پردازش کرده و عبارت سادهشده را به صورت | پس از وارد کردن 1- |

نمایش خواهد داد.

**توضیح بخشهای مختلف کد (توابع):**

کد برنامه شامل توابع متعددی است که هر کدام مسئول بخشی از فرآیند سادهسازی یا نمایشهستند:

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **binaryToGray(int n)** | | • |
| تبدیل یک عدد صحیح (که نشاندهنده یک مینترم یا اندیس باینری است) | | **:هدف** | ◦ |

) معادل آن.Gray Code( به کد گری

|  |  |  |
| --- | --- | --- |
| جداول کارنو از کد گری برای چیدمان خانههای خود استفاده میکنند تا | **کاربرد:** | ◦ |

تضمین شود که خانههای مجاور فقط در یک بیت با هم تفاوت دارند. این تابع برایتولید اندیسهای صحیح برای چیدمان جدول کارنو استفاده میشود.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| و | بینXOR به کد گری، عملیاتn | | برای تبدیل یک عدد باینری n | | **الگوریتم:** | | | ◦ |
|  |  |
| .) انجام میشود | | ( شیفت داده شده به سمت راستn >> 1 | | | | | n |  |
| **:** | | | | |  | | --- | | **grayToBinary(int gray)** | | | | | • |
| تبدیل یک عدد کد گری به معادل باینری آن. | | | | | | **:هدف** | | ◦  ◦ |
| این تابع برای تفسیر مقادیر نمایش داده شده در جدول کارنو (که بر اساس | | | | | | **کاربرد:** | |

کد گری مرتب شدهاند) به اندیسهای استاندارد مینترم (باینری) استفاده میشود.

|  |  |  |
| --- | --- | --- |
| تبدیل کد گری به باینری به صورت تکراری انجام میشود. بیت هشتم | **الگوریتم:** | ◦ |

کد گری برابر است. بیت بعدیMSB ) باینری باMost Significant Bit - MSB( . کردن بیت قبلی باینری با بیت متناظر کد گری بدست میآیدXOR باینری از

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **getKMapIndex(int row, int col, int rows, int cols)** | | • |
|  | محاسبه اندیس خطی (مسطح) یک خانه در جدول کارنو بر اساس شماره | **:هدف** | ◦ |

سطر و ستون آن.

|  |  |  |
| --- | --- | --- |
| این تابع برای دسترسی به مقادیر در آرایه دوبعدی که جدول کارنو را نمایش | **کاربرد:** | ◦ |

میدهد، مورد استفاده قرار میگیرد.

**فرمول:**index = row \* cols + col ◦

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **varsForIndex(int index, int bits, int startVar)** | | • |
| , ...) برای یک اندیس خاص در جدول کارنو،A, B, C ً تولید نام متغیرها (مثال | | **:هدف** | ◦  ◦ |
| با توجه به تعداد کل بیتها (متغیرها) و متغیر شروع (مثال ً برای سطر یا ستون). | | |
| این تابع برای برچسبگذاری سطرها و ستونهای جدول کارنو و همچنین در | | **کاربرد:** |

زمان استخراج عبارت سادهشده، برای تخصیص متغیرها به هر گروه استفادهمیشود.

|  |  |  |
| --- | --- | --- |
| با بررسی بیتهای اندیس (که باینری شدهاند) و مقایسه آنها با | **نحوه کار:** | ◦ |
| متغیرهای اصلی، نام متغیر متناسب با آن بیت را تولید میکند (مثال ً اگر بیت برای | |

باشد ۱ ' نمایش داده میشود، اگرA' ، باشد۰ ' باشد و مقدار بیتA' متغیر'A .)نمایش داده میشود

|  |  |  |  |
| --- | --- | --- | --- |
| **getMintermFromRowCol(int row, int col, int rowBits, int** | | | • |
| **:** | **(colBits** | |  |
| محاسبه شماره مینترم (اندیس باینری استاندارد) یک خانه در جدول کارنو، با | | **:هدف** | ◦ |

دانستن موقعیت سطر و ستون آن و تعداد بیتهای اختصاص داده شده به سطر وستون.

|  |  |  |
| --- | --- | --- |
| این تابع برای نگاشت خانههای جدول کارنو به مینترمهای اصلی تابع بولی | **کاربرد:** | ◦ |

استفاده میشود.

ابتدا، با استفاده از grayToBinary، نمایش کدهای گری سطر و**نحوه کار:** ◦

ستون به باینری تبدیل میشود. سپس این دو عدد باینری با هم ترکیب شده و عددباینری نهایی برای مینترم ساخته میشود.

|  |  |
| --- | --- |
| **isValidGroup(const vector<int>& kmap, int rows, int** | • |

**&<<cols, int r, int c, int h, int w, vector<vector<bool**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **:** | | | | | **(visited** | |  |
| سطر وw ستون) که | | بررسی این که آیا یک گروه مستطیلی (با ابعاد h | | | | **:هدف** | ◦ |
| . شروع میشود، یک گروه معتبر برای سادهسازی است یا خیر | | | | از خانه )r, c( | | |  |
| این تابع هسته الگوریتم یافتن گروهها در جدول کارنو است. یک گروه | | | | | | **کاربرد:** | ◦ |
| است و قبال ً در۱ معتبر باید شامل خانههایی باشد که مقدار آنها در جدول کارنو | | | | | | |
| .) است | آنهاfalse | | گروه دیگری عالمتگذاری نشدهاند (یعنی visited | | | |

و ...) و۱x۱، ۱x۲، ۲x۱، ۲x۲، ۴x۴ ً باشد (مثال۲ همچنین، ابعاد گروه باید توانی از

|  |  |  |  |
| --- | --- | --- | --- |
| . است۱ آنها | کل گروه باید شامل خانههایی باشد که مقدار kmap | |  |
| این تابع باید پیچیدگیهای چیدمان جدول کارنو (دورانی بودن) را نیز در | | **مالحظات:** | ◦ |

نظر بگیرد.

|  |  |  |  |
| --- | --- | --- | --- |
| **,markGroup(vector<vector<bool>>& visited, int r, int c** | | | • |
| **:** | **(int h, int w, int rows, int cols** | |  |
| عالمتگذاری تمام خانههایی که در یک گروه معتبر یافت شدهاند، به عنوان | | **:هدف** | ◦ |

"بازدید شده".

|  |  |  |
| --- | --- | --- |
| پس از اینکه یک گروه معتبر شناسایی شد، این تابع از آن برای جلوگیری از | **کاربرد:** | ◦ |

استفاده مجدد از آن خانهها در گروههای بعدی استفاده میکند. این امر تضمین حداکثر در یک گروه نهایی قرار میگیرد.۱ میکند که هر خانه

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| و | با ابعادh | با پیمایش خانههای موجود در گروه (از )r, c( | | | **نحوه کار:** | | ◦ |
| . تغییر میدهد | | | را بهtrue | )، مقدار متناظر در ماتریسvisited | | w |  |
| |  | | --- | | **extractTerm(int r, int c, int h, int w, int rowBits, int** | | | | | | | | • |

**colBits, int rows, int cols, int startVarRow, int**

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **(startVarCol** | |  |
| استخراج عبارت بولی مربوط به یک گروه خاص از خانهها در جدول کارنو. | | **:هدف** | ◦  ◦ |
| این تابع خروجی نهایی فرآیند سادهسازی را تولید میکند. برای هر گروه، | | **کاربرد:** |

) ساخته میشود.Product Term( تابعی به صورت حاصلضرب

|  |  |  |
| --- | --- | --- |
| **نحوه کار:**ابتدا، متغیرهای مربوط به سطر و ستون شروع گروه )r, c( و همچنین | | ◦  .1 |
|  |
| . مشخص میشوند | ابعاد گروه )h, w( |
| برای هر متغیر، بررسی میشود که آیا در کل گروه مقدار آن ثابت میماند یاخیر.  بود یا۰ اگر متغیری در تمام خانههای گروه مقدارش ثابت بود (مثال ً همیشه | | .2 .3 |

۰ بود)، آن متغیر در عبارت نهایی لحاظ میشود. اگر مقدار ثابت۱ همیشه

بود، متغیر به صورت متمم ('A بود، متغیر به صورت۱ ) و اگر مقدار ثابت

|  |  |  |
| --- | --- | --- |
| .) ظاهر میشود | اصلی (A | .4 |
| را بگیرد)،۱ و هم۰ اگر مقدار یک متغیر در طول گروه تغییر کند (مثال ً هم | |

آن متغیر از عبارت نهایی حذف میشود، زیرا باعث سادهسازی میشود.

این تابع برای هر گروه یک "ترم" تولید میکند و تمام این ترمها با هم در .5 نهایی را تشکیل دهند.SOP " ترکیب میشوند تا عبارتOR" نهایت با عملگر

|  |  |
| --- | --- |
| **simplifyKMap(const vector<int>& kmap, int rows, int** | • |

**cols, int rowBits, int colBits, int startVarRow, int**

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **(startVarCol** | |  |
| الگوریتم اصلی برای سادهسازی جدول کارنو. | | **:هدف** | ◦ |

|  |  |  |
| --- | --- | --- |
| این تابع مسئول پیدا کردن تمام گروههای ممکن (مستطیلی، با ابعاد توان | **:کاربرد** | ◦ |

" در جدول کارنو است و سپس با استفاده از روشی (مانند۱" ) از خانههای۲ سادهشده)، ترمهای الزم برایQuine-McCluskey یا الگوریتمPetrick الگوریتم " ها را انتخاب میکند.۱" پوشش تمام **نحوه کار:** ◦

یک ماتریس visited برای پیگیری خانههایی که قبال ً در گروهی قرار .1

گرفتهاند، ایجاد میکند.

) و پیمایش در کل جدول، به دنبالr=0, c=0( با شروع از اولین خانه .2 " میگردد که هنوز بازدید نشدهاند.۱" خانههای  
" بازدید نشده، سعی میکند بزرگترین گروه ممکن را از آن۱" برای هر خانه .3، ...،۱x۱، ۱x۲، ۱x۴ :خانه شروع کند (با امتحان کردن ابعاد مختلف گروهها  
 ، ...).۲x۱، ۴x۱، ...، ۲x۲

بزرگترین گروه ممکن را پیدا کرده و با استفاده ازmarkGroup خانههای .4

آن را عالمتگذاری میکند.

با استفاده از extractTerm، عبارت بولی مربوط به این گروه را .5

استخراج میکند.

نهایی راSOP تمام عبارتهای استخراج شده با هم ترکیب شده و عبارت .6تشکیل میدهند.

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **simplify5Var(const vector<int>& kmap)** | | • |
| متغیره.۵ پردازش سادهسازی برای جداول کارنو | | **:هدف** | ◦  ◦ |
| متغیره نمایش۴ متغیره به طور سنتی به صورت دو جدول۵ جدول کارنو | | **کاربرد:** |

وA=0 " ها روی الیه دیگر قرار میگیرد (یکی برای۱" داده میشود که یک الیه از متغیره را به طور جداگانه پردازش۴ ). این تابع این دو جدولA=1 دیگری برایمیکند.

**نحوه کار:** ◦

متغیره تقسیم میکند (اولین جدول برای۴ متغیره را به دو جدول۵ جدول .1۱ است و دومین جدول برای زمانی که متغیر اول۰ زمانی است که متغیر اول است).

|  |  |  |
| --- | --- | --- |
| ) (یا مشابه آن | متغیره، تابع ۴ برای هر جدولprintSingleKMap | .2 |

فراخوانی میشود تا سادهسازی انجام شود.

متغیره با هم ترکیب میشوند.۴ عبارات سادهشده از هر دو جدول .3

|  |  |
| --- | --- |
| **,printSingleKMap(const vector<int>& kMapValues, int rows** | • |

**int cols, int rowBits, int colBits, int startVarRow, int**

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **(startVarCol** | |  |
| نمایش و پردازش یک الیه از جدول کارنو (که معموال ً مربوط به یک یا دو | | **:هدف** | ◦ |

متغیر است).

|  |  |  |
| --- | --- | --- |
| این تابع هم برای نمایش بصری جدول کارنو (با برچسبگذاری مناسب | **کاربرد:** | ◦ |

سطرها و ستونها) و هم برای انجام فرآیند سادهسازی بر روی آن الیه استفادهمیشود.

**نحوه کار:** ◦

برچسبهای سطر و ستون را با استفاده از varsForIndex تولید و .1

نمایش میدهد.

مقادیر مینترمها را در خانههای جدول قرار میدهد. .2

تابع اصلی سادهسازی (simplifyKMap) را برای این الیه فراخوانی .3

میکند.

|  |  |  |  |
| --- | --- | --- | --- |
| عبارت سادهشده حاصل را برای این الیه نمایش میدهد. | | | .4 |
| **printKarnaughMap(const vector<int>& kMapValues, int** | | | • |
| **:** | **(numVariables** | |  |
| نمایش کامل جدول کارنو برای تعداد متغیرهای ورودی مشخص شده. | | **:هدف** | ◦  ◦ |
| این تابع ابتدا مشخص میکند که جدول کارنو چند بعدی است و چگونه باید | | **کاربرد:** |

و ...) تقسیم کند.۳x۸ یا۴x۴ ًآن را به الیههای قابل نمایش (معموال

|  |  |
| --- | --- |
| **نحوه کار:** بر اساس numVariables، ابعاد و ساختار جدول کارنو را تعیین میکند متغیر، دو۵ ؛ برای۴x۴ ، متغیر۴ خواهد بود؛ برای۲x۴ ، متغیر۳ (مثال ً برای | ◦  .1 |

).۴x۴ جدول

برای هر الیه از جدول کارنو، تابع printSingleKMap را فراخوانی .2

میکند و عبارت سادهشده مربوط به آن الیه را نمایش میدهد.

|  |  |  |  |
| --- | --- | --- | --- |
| **:** | **()main** | | • |
| تابع اصلی اجرای برنامه. | | **:هدف** | ◦  ◦  ◦  .1  .2 |
| این تابع جریان کلی برنامه را کنترل میکند. | | **کاربرد:** |
| **نحوه کار:** پیام خوشآمدگویی و دستورالعملها را نمایش میدهد.از کاربر تعداد متغیرها را دریافت میکند. | | |

بر اساس تعداد متغیرها، اندازه جدول کارنو و محدوده مینترمها را محاسبه .3 میکند.

|  |  |
| --- | --- |
| حلقهای برای دریافت مینترمها از کاربر ایجاد میکند تا زمانی که کاربر 1- | .4 |

.را وارد کند

|  |  |  |
| --- | --- | --- |
| . ذخیره میکند | مینترمهای وارد شده را در یک vector | .5 |
| در نهایت، تابع printKarnaughMap را با مینترمهای جمعآوری شده | | .6 |

فراخوانی میکند تا جدول کارنو چاپ و سادهسازی انجام شود.

**وابستگیها:**

+C استفاده میکند که باید در اکثر محیطهای توسعه++C این برنامه از کتابخانههای استاندارد موجود باشند:+

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| برای ورودی و | برای عملیات ورودی/خروجی استاندارد (مانندcin **:** | **<iostream>** | | | • |
| .) برای خروجی | | | | cout |  |
| برای استفاده از ساختار داده داینامیکvector که برای ذخیره**:** | | | **<vector>** | | • |

مقادیر جدول کارنو و لیست مینترمها استفاده میشود.

|  |  |  |
| --- | --- | --- |
| برای استفاده از توابع ریاضی مانندpow (قدرت) که ممکن است در**:** | **<cmath>** | • |

محاسبات مربوط به ابعاد جدول کارنو یا تبدیلهای باینری/گری مفید باشد.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **:** برای قالببندی خروجی، مانند تنظیم عرض فیلدها برای نمایش بهتر | | | **<iomanip>** | | • |
| .) | وleft | جدول کارنو (مثال ً با استفاده از setw | | |  |
| **:** ممکن است برای مدیریت مجموعهای از عناصر منحصر به فرد (مانند | | | | **<set>** | • |

مینترمهای غیر تکراری) یا برای پیادهسازی برخی از الگوریتمهای یافتن گروه استفادهشده باشد.

|  |  |  |
| --- | --- | --- |
| **:** ممکن است برای نگاشت نام متغیرها به مقادیر یا برای ذخیره ترمهای | **<map>** | • |

سادهشده استفاده شده باشد.

هستند و نیازی به نصب جداگانه ندارند.++C این کتابخانهها جزو کتابخانههای استاندارد