

گزارش پروژه درس ریاضیات گسسته

عنوان پروژه: حل سودوکو با استفاده از SAT Solver

نام دانشجو: سید محمد سعید مظهری

شماره دانشجویی: ۴۰۳۲۵۳۲۳

نام استاد: دکتر سید حسین خواسته

تاریخ تحویل: ۳۱ مرداد ۱۴۰۴

مقدمه

در این پروژه، هدف حل جدول سودوکو با استفاده از SAT Solver می‌باشد.

مسئله SAT (Satisfiability Problem) یکی از مهم‌ترین مسائل پایه‌ای در منطق گزاره‌ای و علوم کامپیوتر است. این مسئله بررسی می‌کند که آیا یک فرمول منطقی می‌تواند با یک تخصیص مناسب از مقادیر درست و نادرست به متغیرهایش ارضا شود یا خیر.

یک SAT Solver برنامه‌ای است که چنین فرمول‌هایی را به طور کارآمد بررسی و در صورت وجود پاسخ، یک تخصیص معتبر ارائه می‌کند. از SAT Solver ها در حوزه‌های مختلفی مانند راستی‌آزمایی مدارهای دیجیتال، تحلیل نرم‌افزار، برنامه‌ریزی، زمان‌بندی و حتی حل پازل‌هایی مانند سودوکو استفاده می‌شود.

سودوکو به عنوان یک مسئله منطقی با قیدهای مشخص انتخاب شد تا نحوه تبدیل آن به فرمول CNF و حل آن با ابزارهای منطقی نمایش داده شود.

روش کار

ابتدا با شماره‌گذاری خانه‌ها و تعریف متغیرها، قیود سودوکو شامل موارد زیر به فرمول CNF تبدیل شدند:

هر خانه دقیقاً یک عدد داشته باشد.

هر عدد دقیقاً یک بار در هر سطر ظاهر شود.

هر عدد دقیقاً یک بار در هر ستون ظاهر شود.

هر عدد دقیقاً یک بار در هر بلوک ۳×۳ ظاهر شود.

برخی خانه‌ها دارای مقادیر اولیه باشند.

سپس از کتابخانه PySAT و حل‌کننده Glucose3 برای یافتن جواب استفاده شد. یک رابط کاربری گرافیکی با

tkinter برای ورود داده‌ها طراحی شد و جواب نهایی (در صورت وجود) در قالب تصویر ذخیره گردید.

پیاده‌سازی

کد با زبان Python نوشته شده و از کتابخانه‌های `numpy`، `matplotlib`، `pysat`، `tkinter` استفاده می‌کند. برنامه پس از اجرا، جدولی 9×9 برای ورود مقادیر نمایش می‌دهد و پس از فشردن دکمه Done فایل `sudoku.cnf` تولید شده و جواب در `sudoku_solution.png` ذخیره می‌شود.

توضیحات بخش‌های مختلف کد

کد برنامه به چند بخش اصلی تقسیم شده است:

۱- تبدیل جدول سودوکو به فرمول CNF

- تابع `varnum(i, j, v)`: هر خانه جدول سودوکو را به یک متغیر منطقی یکتا تبدیل می‌کند.
- تابع `exactly_one(literals)`: تضمین می‌کند که از بین مجموعه‌ای از متغیرها دقیقاً یکی برقرار باشد (قید اصلی سودوکو).
- تابع `sudoku_to_cnf(grid)`: قیود سودوکو شامل موارد زیر را تولید کرده و در قالب فایل DIMACS CNF ذخیره می‌کند:
 - هر خانه باید دقیقاً یک عدد داشته باشد.
 - هر عدد در هر سطر دقیقاً یک بار بیاید.
 - هر عدد در هر ستون دقیقاً یک بار بیاید.
 - هر عدد در هر بلوک 3×3 دقیقاً یک بار بیاید.
 - اعداد داده شده‌ی اولیه به‌عنوان قیود اضافه می‌شوند.

۲- حل سودوکو با SAT Solver

- تابع `solve_sudoku(grid)`: پس از ساخت قیود، آنها را به حل‌کننده‌ی منطقی `Glucose3` از کتابخانه‌ی `PySAT` می‌دهد.
- اگر جواب وجود داشته باشد، مدل استخراج می‌شود و جدول کامل سودوکو بازسازی می‌شود.

۳- نمایش و ذخیره‌ی جواب

- تابع `visualize_sudoku(grid)`: جدول حل‌شده را با استفاده از کتابخانه‌ی `matplotlib` رسم کرده و به‌صورت تصویر با فرمت PNG ذخیره می‌کند.

۴- رابط کاربری گرافیکی

- ماژول `tkinter` برای دریافت ورودی از کاربر استفاده شده است.
- کاربر می‌تواند با کلیک روی هر خانه مقدار آن را تغییر دهد. (۰ نشان‌دهنده خانه خالی است).
- با فشردن دکمه Done، برنامه ابتدا جدول ورودی را چاپ کرده، سپس فرمول CNF ساخته شده و حل‌کننده اجرا می‌شود و جواب به‌صورت تصویر ذخیره می‌گردد.

نتایج

با اجرای برنامه، سودوکوهای ورودی به طور کامل و صحیح حل شدند و جواب‌ها تقریباً به صورت آنی به دست آمدند. تصویر جواب نیز برای اطمینان از صحت حل ذخیره شد.

جمع‌بندی

این پروژه نشان داد که مسائل منطقی مانند سودوکو را می‌توان با استفاده از **SAT Solver** به صورت کارآمد حل کرد. از این روش می‌توان برای حل مسائل پیچیده‌تر در زمینه‌های مختلف علوم کامپیوتر و مهندسی استفاده نمود.