

Case Based Reasoning System

حسن شجاعی مند

دانشجوی کارشناسی ارشد نرم افزار

shojaee@ce.sharif.edu

خلاصه

CBR¹، یکی از تکنیک‌های تصمیم‌گیری در خصوص راه‌حل مشکلات می‌باشد. این سیستم‌ها، با استفاده از راه‌حل مشکلات قبلی پیش‌آمده در یک دامنه خاص، سعی می‌کنند تا برای مشکل موجود راه‌حلی ارائه کنند. ایده اصلی آن مبتنی بر این اصل است که مشکلات مشابه راه‌حل‌های مشابه نیز خواهند داشت. در این مقاله، مروری کلی بر سیستم‌های CBR، اجزاء تشکیل دهنده آن و فرایند ارائه راه‌حل برای مشکلات توسط این سیستم‌ها خواهیم داشت.

۱- مقدمه

ایده اصلی در CBR، مبتنی بر این است که مشکلات مشابه راه‌حل‌های مشابه نیز دارند. بنابراین یک مشکل جدید می‌تواند با یافتن راه‌حل‌های مشکلات قبلی مشابه این مشکل، و سازگار نمودن راه‌حل آنها برای مشکل فعلی حل شوند. هر مورد²، شامل تعریفی از مشکل به همراه راه‌حل مربوطه می‌باشد. در حقیقت بسیاری از افراد در زندگی روزمره نیز از همین تکنیک، یعنی استفاده از مشکلات و تجربیان گذشته و نتایجی که به همراه داشته است، استفاده می‌کنند. دانش و فرایند استدلالی که توسط یک خبره برای رسیدن به راه‌حل مورد استفاده قرار می‌گیرد، در همین مورد‌های گذشته ذخیره شده است [4]. با بروز یک مشکل جدید یک فرد خبره ابتدا مورد‌های قبلی مشابه با این مورد جدید را که قبلاً برایش اتفاق افتاده است و راه‌حل ارائه شده برای آنها با موفقیت همراه بوده است و شاید نتایج خوبی دربرنداشته است را در نظر می‌گیرد. ملاک و معیاری که استفاده می‌کند تا به این شباهت پی ببرد، ویژگی‌های مهمی هستند که برای مسئله مطرح می‌باشد. فرد خبره همچنین باید تفاوت‌های مسئله فعلی با مسائل مشابهی که از حافظه خود پیدا کرده است، تشخیص دهد که برای اینکار نیاز به یکسری دانش و تجربیاتی دارد. سپس با در نظر گرفتن تفاوت‌های موجود راه‌حلی که برای مسائل و مشکلات قبلی ارائه شده و موفقیت آمیز بوده است را برای مسئله فعلی و شرایط موجود سازگار می‌کند. مثلاً یک پزشک با مشاهده علائم مشابه در بیمار خود با بیماری که یک ماه قبل معاینه کرده است، اگر داروهای تجویز شده به آن بیمار مفید بوده باشد از همان تجویز استفاده می‌کند و اگر مناسب نبوده آنرا تغییر می‌دهد. یک مشاور مالی یک شرکت می‌تواند بر اساس موقعیت فعلی شرکت و مشکلات پیش‌آمده قبلی، روش مناسبی را پیشنهاد دهد.

یک سیستم CBR، سعی می‌کند تا از این روش برخورد انسان‌ها با مشکلات برای تصمیم‌گیری تقلید کند. آنگونه که در [1] آمده است، CBR به دلایل زیر مورد توجه قرار گرفته است:

- نیاز به یک مدل برای دامنه مورد نظر ندارد و مجموعه دانش آن تنها از یکسری مورد‌هایی تشکیل شده است که قبلاً در آن دامنه بوقوع پیوسته است. این مورد‌ها تنها شامل توصیفی از مشکل و راه‌حل ارائه شده برای آن مشکل خواهد بود و علاوه بر آن نیاز به کسب دانش خاصی از آنها نداریم.

¹ Case-based reasoning

² Case

توصیف کلی چرخه فوق، آنگونه که در [1] آمده است به این صورت می‌باشد که یک مشکل جدید با موردهای موجود در پایگاه مورد¹ مقایسه می‌شود و یک یا چند مورد مشابه بازیابی می‌شود. راه‌حل پیشنهاد شده توسط موردهای مشابه بازیابی شده، مورد استفاده مجدد قرار می‌گیرد و مناسب بودن آن برای مشکل پیش‌آمده تست می‌شود. صرف‌نظر از مواردی که مشکل موجود کاملاً مطابق با مورد بازیابی شده باشد، احتمالاً نیاز به بازبینی راه‌حل پیشنهاد شده خواهیم داشت و در نهایت پس از موفقیت راه‌حل ارائه شده، به عنوان یک مورد جدید باید به سیستم اضافه شود و به عبارتی باید آنرا برای استفاده‌های بعدی حفظ کنیم.

۳- نمایش مورد

یک مورد به صورت مفهومی دانشی است که ارائه دهنده یک تجربه می‌باشد. یک توافق کلی بر روی اینکه یک مورد باید شامل چه اطلاعاتی باشد، وجود ندارد. در [3] تعریفی از یک مورد و مواردی که شامل می‌شود ارائه شده است. مورد، یک توصیف به زبان طبیعی از یک وضعیت یا مشکل، یکسری سوالهایی که باید پرسیده شود و جوابهای ممکن برای آنها و همچنین به ازای هر کدام از آنها، عملیاتی که باید انجام شود را مشخص می‌کند. به عنوان مثال در یک کاربرد پشتیبانی مشتری یک مورد به صورت زیر تعریف می‌شود:

توصیف: مشتری درخواست قیمت کالایی را می‌کند

سوال: از چه روشی برای پرداخت هزینه آن استفاده می‌کنید؟

جواب: کم شدن از شارژ

عملیات: از قیمت کالا تخفیف بابت داشتن شارژ قبلی را کم کن و قیمت را اعلام کن.

در [1] به طور کلی دو فاکتور را مطرح کرده است که می‌توان آنها را در خصوص اطلاعاتی که باید در یک مورد لحاظ شود، در نظر گرفت. یکی مفید بودن² اطلاعات است و دیگری دست‌یابی آسان به آنها می‌باشد. موارد دیگری نیز در منابع مختلف عنوان شده است که بر نحوه نمایش یک مورد تاثیر می‌گذارند. به عنوان مثال می‌توان به نیازمندی‌های یک دامنه و مسئله خاص و همچنین ساختار موردهای قبلی که قبلاً ذخیره شده‌اند و در سیستم می‌خواهیم از آنها استفاده کنیم را ذکر کرد.

در [5] که کاربرد سیستم CBR در فرایند مهندسی مجدد فرایندهای یک سازمان را مورد توجه قرار داده است، ساختاری که برای نمایش یک مورد در سیستم در نظر گرفته است یک ساختار سلسله مراتبی می‌باشد. به عبارت دیگر پایگاه مورد سیستم استفاده شده یک درخت است که در بالاترین لایه سطح کلی یک تجارت را نشان می‌دهد و در لایه‌های پایین تر موارد جزئی تر مطرح می‌شوند تا به فرایندها می‌رسیم و سپس به ازای هر فرایند جدیدی که به پایگاه مورد اضافه می‌شود، یک گره در درخت اضافه خواهد شد.

همانگونه که گفته شد، نحوه نمایش یک مورد بسته به کاربرد مورد نظر متفاوت خواهد بود. ولی دو روش مطرح در نحوه نمایش موردها در سیستم‌های CBR وجود دارد.

اولین نحوه نمایش مورد، استفاده از یک لیست شامل دوتایی‌هایی از ویژگی و مقدار ویژگی می‌باشد. به عنوان مثال در نظر بگیرید که می‌خواهیم یک سیستم تشخیص دلیل یک ایراد در ماشین تهیه کنیم. مشکلاتی که می‌تواند، بروز کند شامل یک توصیفی از مشکل خواهد بود که می‌توان آنرا به صورت ویژگی مشکل و سپس مقداری برای آن که یک رشته خواهد بود شامل توصیفی از نشانه‌ای که بروز کرده است مانند: موتور کار نمی‌کند. همچنین برای تشخیص ایراد ماشین به ویژگی‌هایی مانند ولتاژ باطری ماشین نیز ممکن است نیاز داشته باشیم که آنها را هم در مورد در نظر گرفته و مقادیر

¹ Case base

² functionality

متناظر آنها را حفظ خواهیم کرد. به عبارت دیگر در این روش کلیه ویژگی‌های تاثیرگذار در مسئله باید تشخیص داده شده و مقادیر متناظر آنها در مورد ذخیره شود. این نحوه نمایش مورد، برای مسائلی کاربرد دارد که بتوان ویژگی‌های مسئله را از ابتدا مشخص کرد و همچنین از مزایای آن استفاده از بانک‌های اطلاعاتی برای نگهداری موردها به صورت یک رکورد می‌باشد.

روش دیگری که برای نمایش مورد رایج می‌باشد، استفاده از روش شیء‌گرا^۱ می‌باشد. در این روش باید بتوان برای موردها یک ساختار سلسله مراتبی مانند آنچه در [5] انجام شده است و در بالا توضیح آن آمد تعریف کرد. اشیاء مختلف با یکدیگر در ارتباط هستند و هر شیء می‌تواند زوج‌های ویژگی و مقدار را شامل شود.

یک پایگاه مورد را بسته به نوع ویژگی‌هایی که برای موردها در نظر می‌گیریم می‌توان به دو دسته کلی تقسیم بندی کرد: (۱) یکنواخت^۲: در این پایگاه مورد، تمام موردها دارای یکسری ویژگی‌های یکسانی هستند. این پایگاه برای مسائلی مناسب می‌باشد که بتوان از ابتدا کلیه ویژگی‌هایی که در مشکلات آن دامنه خاص اتفاق می‌افتد را در نظر گرفت.

(۲) غیر یکنواخت^۳: در برخی از مسائل نمی‌توان از ابتدا کلیه ویژگی‌هایی که در مشکلات آن دامنه دخالت دارند را در نظر گرفت. به عنوان مثال در یک سیستم تشخیص دلیل بیماری شخص ممکن است فاکتور زیادی مانند انواع مرض‌های مختلف، آزمایشات مختلف و رفتارهای متنوعی داشته باشیم و بنابراین موردهایی که در پایگاه مورد خواهیم داشت دارای ویژگی‌های یکنواختی نخواهند بود.

۴- شاخص گذاری^۴

یکی از موارد مهم در یک سیستم CBR، شاخص گذاری موردها می‌باشد تا بتوان به سرعت در مرحله بازیابی موردهای مشابه، موردهای مشابه را پیدا کرده و بازیابی انجام گیرد. شاخص گذاری موردها عبارت است از نسبت دادن یک شاخص به موردها، به گونه‌ای که سرعت دسترسی به آنها افزایش یابد. در [1] دو مدل برای ساختار پایگاه مورد ذکر شده است که در ادامه آنها را مورد بررسی قرار می‌دهیم.

- مدل حافظه پویا^۵

شاید بتوان اولین سیستمی که از CBR استفاده کرد را سیستم CYRUS عنوان کرد که از همین مدل برای پایگاه مورد خود استفاده کرده بود. این مدل در سال ۱۹۸۲ در مقاله [6] مطرح شد. در این مدل، یک ساختار سلسله مراتبی برای پایگاه مورد در نظر گرفته می‌شود. موردهایی که یکسری ویژگی‌های مشترکی دارند در زیر یک گره بالاتر با عنوان مورد کلی^۶ قرار می‌گیرند. این ساختار شامل سه نوع گره اصلی می‌باشد: نرم^۷، مورد و شاخص. هر شاخص یک ویژگی را مشخص می‌کند که نتوانسته است با ویژگی‌های دیگر مقایره مشابه داشته باشد و بهمین دلیل به ازای هر گره شاخص یک گره به عنوان فرزند این گره حاوی مقدار این ویژگی به درخت اضافه می‌شود و در نهایت به گره مورد خواهیم رسید که موردها را در پایگاه مورد مشخص می‌کنند. ساختار این مدل آنگونه که در [2] آمده است در شکل ۲ دیده می‌شود.

¹ object-oriented

² Homogenous

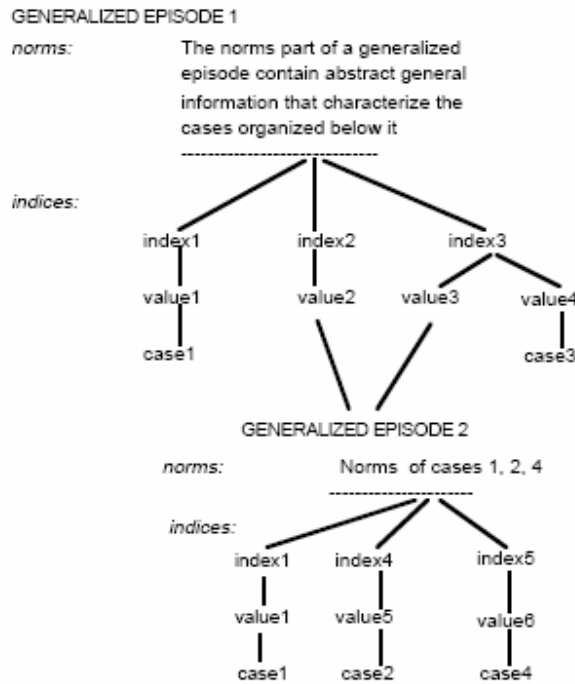
³ Heterogenous

⁴ indexing

⁵ dynamic memory model

⁶ generalized episode

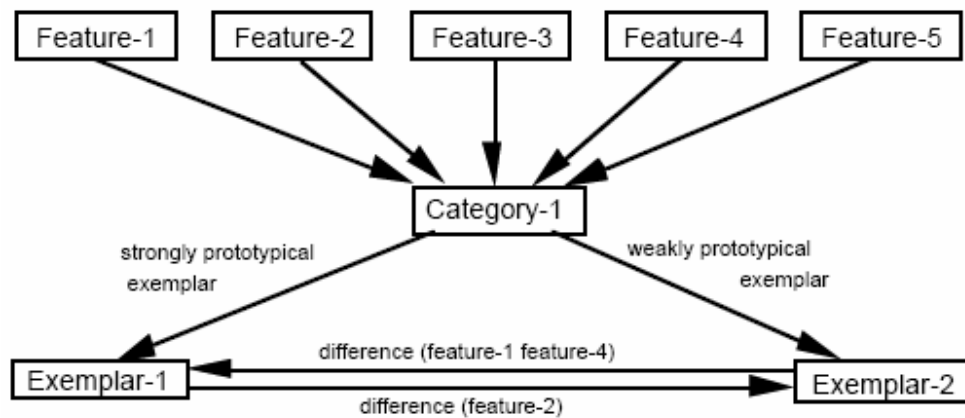
⁷ norm



شکل ۲: مدل حافظه پویا

– مدل طبقه‌بندی^۱

این مدل اولین بار در سیستم PROTOS مطرح شد [7]. در این مدل، پایگاه مورد یک ساختار شبکه‌ای دارد که از طبقه^۲، مورد و نشانه‌گر شاخص تشکیل شده است. هر مورد به یک طبقه نسبت داده می‌شود. نشانه‌گر می‌تواند به یک مورد یا یک طبقه اشاره کند. یک ویژگی به همراه یک نام و مقدار در ساختار مشخص می‌شود. ساختار این مدل در شکل ۳ دیده می‌شود.



شکل ۳: ساختار مدل طبقه‌بندی

¹ Category & Exemplar Model

² category

۵- بازیابی

این مرحله شامل بازیابی مورد های مشابه موجود در کتابخانه موردها می باشد که شبیه به مشکل فعلی پیش آمده می باشند. الگوریتم انتخاب شده برای این مرحله اهمیت خاصی دارد چون یک سیستم CBR در صورتی برای کاربردهای بزرگ کارایی خواهد داشت که قدرت بازیابی مورد های مشابه را از میان تعداد زیادی از مورد های موجود داشته باشد. جستجو در یک سیستم CBR با جستجو های معمولی متفاوت است چون به دنبال یک مورد خاص و مشخص نیستیم، بلکه باید موردی را پیدا کنیم که شبیه به مشکل فعلی باشد و بنابراین باید از روش های اکتشافی^۱ برای آن استفاده کرد. یکی از مشهورترین الگوریتم هایی که در این مرحله مورد استفاده قرار می گیرند الگوریتم نزدیکترین همسایه^۲ می باشد. در این الگوریتم، ارزیابی شباهت بین موردها از طریق یک تابعی انجام می شود که این تابع فاصله بین دو مورد را با استفاده از ویژگی های آن و وزنی که به هر ویژگی داده می شود بدست می آورد. همانگونه که قبلا گفته شد، یکسری ویژگی برای موردها تعیین می کنیم و مقادیری به آنها نسبت می دهیم. با استفاده از تابعی مانند تابع زیر می توان، فاصله بین دو مورد را پیدا کرد:

$$\frac{\sum_{i=1}^n w_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^n w_i}$$

در این فرمول، w_i وزن ویژگی نام، $sim(f_i^I, f_i^R)$ میزان شباهت بین ویژگی نام مشکل موجود و مورد بازیابی شده می باشد. برای اندازه گیری میزان شباهت می توان از توابع مختلفی استفاده کرد که یکی از آنها فاصله اقلیدسی می باشد.

۶- استفاده مجدد

بعد از آنکه مورد های مشابه برای یک مشکل بازیابی شدند، در مرحله بعدی باید راه حل استفاده شده در آن موردها را با مشکل موجود تطبیق دهد. برای انجام عمل تطبیق، اختلاف مهم بین مورد فعلی و مورد های ذخیره شده را مشخص کرده و سپس با در نظر گرفتن این اختلافات از یک فرمول یا قاعده استفاده می کند تا راه حل های مورد های ذخیره شده به گونه تغییر داده می شود که مناسب برای مشکل پیش آمده باشد.

چند روش برای تطابق راه حل مورد بازیابی شده برای رسیدن به یک راه حل برای مشکل موجود می باشد:

Null Adaptation: در این حالت تطابقی بر روی راه حل بازیابی شده صورت نمی گیرد. همان راه حل بازیابی شده به عنوان راه حل مشکل موجود ارائه می شود. این حالت برای مواردی مناسب می باشد که تعداد مورد های موجود در پایگاه موردها زیاد باشد و تقریباً تمام مواردی که ممکن است در آن دامنه اتفاق افتد را شامل شود. به عنوان مثال یک سیستم واگذاری وام در یک بانک را در نظر بگیرید. در این سیستم، اگر تنها دو فاکتور درآمد ماهیانه شخص و میزان اقساط ماهیانه شخص مهم باشند. می توان مورد های قبلی را در پایگاه مورد، ذخیره کرد و آنها را به دو دسته مورد های موفقیت آمیز و مورد های که بازپرداخت وام با مشکل مواجه شده است، تقسیم کرد. هنگام تصمیم گیری در خصوص یک واگذاری وام جدید، سیستم می تواند با استفاده از الگوریتم نزدیکترین همسایه، نزدیکترین همسایه را به مورد فعلی

¹ heuristic

² nearest neighbor

مشخص کرده و اگر نزدیکترین همسایه یک مورد موفقیت آمیز بود با پرداخت وام موافقت شود در غیر اینصورت وام پرداخت نشود. بعبارت دیگر، سیستم تنها از راه حل مشابه ترین مورد استفاده می کند.

User Adaptation: این روش برای سیستم های نسبتاً پیچیده مناسب می باشد. در این روش کاربر با دریافت راه حل مورد یا مورد های مشابه پیدا شده و در نظر گرفتن تفاوت های مشکل پیش آمده با آن مورد ها راه حل را تغییر می دهد. این روش به جای آنکه از یک پایگاه دانش^۱ استفاده کند متکی بر اطلاعات کاربر است.

Structural Adaptation: در این روش از یکسری پارامترها، معادلات و قواعد که پایگاه دانش سیستم را تشکیل می دهند استفاده می شود تا راه حل باز یابی شده منطبق بر مشکل موجود شود. در این روش با استفاده از دانشی که داریم استفاده می کنیم تا راه حل باز یابی شده را تغییر و سازگار با مشکل موجود نماییم.

Derivational Adaptation: در این روش نیز، همانند روش قبلی متکی بر یک پایگاه دانش می باشد. تفاوت آن با روش قبلی در این است که در این روش از راه حل باز یابی شده استفاده نمی شود، بلکه از قواعد استفاده شده برای رسیدن به راه حل استفاده خواهد شد تا راه حلی برای مشکل موجود پیدا کنیم. بعبارت دیگر، در این روش باید سیر منطقی ارائه یک راه حل برای مشکلات قبلی در پایگاه مورد ها نگهداری شود تا سیستم پس از باز یابی مورد های مشابه بر اساس آن سیر منطقی استفاده شده برای ارائه راه حل مشکلات قبلی، راه حلی برای مشکل موجود بدست آورد.

همانگونه که دیده شد، دو روش آخر، روش های اتوماتیک برای رسیدن به راه حلی برای مشکل موجود می باشند که در هر دو روش به یک پایگاه دانش نیاز داریم تا بتوانیم، راه حلی برای مشکل ارائه کنیم.

۷- یادگیری و نگهداری^۲ سیستم

یکی از مزایای اصلی این سیستم، روشی است که برای یادگیری استفاده می کند. در این سیستم یادگیری به صورت بسیار ساده و تنها با افزودن مورد های جدید به سیستم امکان پذیر می شود. پس از ارائه یک راه حل برای سیستم، راه حل باید ارزیابی شود و در صورت مناسب بودن راه حل، مشکل جدید به صورت یک مورد به پایگاه مورد سیستم اضافه خواهد شد.

به مرور زمان تعداد مورد های موجود در پایگاه مورد، افزایش یافته و برخی از مورد های موجود اعتبار خود را از دست می دهند. به همین دلیل برای یک سیستم CBR، باید فاکتور هایی را برای نگهداری آن در نظر گرفت. از لحاظ حذف مورد های غیر معتبر می توان از دو روش استفاده کرد:

(۱) استفاده از یک فاکتور اعتبار^۳ برای مورد های موجود در پایگاه مورد: از هر موردی که استفاده شود اعتباری به آن نسبت داده می شود و به این ترتیب مورد های پر کاربرد مشخص شده و مورد هایی را که کمتر مورد استفاده قرار گرفته اند از پایگاه مورد حذف می کنیم.

(۲) استفاده از یک فاکتور طول عمر برای مورد: زمانی برای طول عمر هر مورد در نظر گرفته می شود و پس از طی شدن زمان مورد نظر، مورد از پایگاه مورد ها حذف خواهد شد.

به طور کلی، کیفیت پایگاه مورد ها باید از دو نظر کیفی^۴ و کمی^۵ توسط سیستم تضمین شود. از لحاظ کیفی می توان فاکتور هایی زیر را در نظر گرفت:

¹ knowledge base
² maintenance
³ credit
⁴ Qualitative
⁵ Quantitative

- سازگاری^۱: عدم وجود موردهای اضافه، عدم وجود موردهای متناقض و همچنین موردهای نامعتبر
- کامل بودن^۲: پایگاه مورد باید حاوی تمام موردهای حیاتی و ضروری برای پاسخدهی به مشکلات مطرح شده در آن دامنه مورد نظر باشد.

از لحاظ کمی باید حجم اطلاعات موجود در پایگاه داده را کنترل کرد. باید موردهایی که ارزش زیادی ندارند از پایگاه داده حذف شوند تا مرحله بازبینی با سرعت بهتری انجام شود. ساختار شاخص و مدل استفاده شده برای پایگاه موردها باید در طول زمان مورد بازبینی قرار بگیرد تا در صورت نیاز از مدل‌های مناسب‌تر برای سیستم استفاده شود و همچنین موردهای خطا دار باید در پایگاه تشخیص داده شوند و برای جلوگیری از بروز اشتباه از پایگاه حذف شوند.

مراجع

- [1] case based reasoning: A Review ,Cambridge University Press, 1994. The Knowledge Engineering Review, Vol. 9, No. 4: pp355-381
- [2] A. Aamodt, E. Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press, Vol. 7: 1, pp. 39-59, 1994.
- [3] E. Turban, J. E. Aronson, Decision support systems and Intelligent systems, 6th Ed, 2000.
- [4] Carl G. Looney, Case-Based Reasoning Logic, <http://www.cse.unr.edu/~looney/cs773a/unit3.htm>, Sep. 2000
- [5] Selma Limam Mansar, Hajo A. Reijers, Farhi Marir, Case-Based Reasoning as a Technique for Knowledge Management in Business Process Redesign, *Electronic Journal on Knowledge Management, Volume 1 Issue 2 (2003) 113-124*
- [6] R. Schank, *Dynamic memory; a theory of reminding and learning in computers and people*. Cambridge University Press. 1982.
- [7] B. Porter, R. Bareiss, Robert Holte, Concept learning and heuristic classification in weak theory domains. *Artificial Intelligence*, vol. 45, no. 1-2, September 1990. pp 229-263.

¹ consistency

² Completeness