دانشکده زبان‌ها و زبان‌شناسی

سوالات پایان‌ترم

درس هوش مصنوعی

استاد: دکتر محمد بحرانی

وحید مواجی
89702855

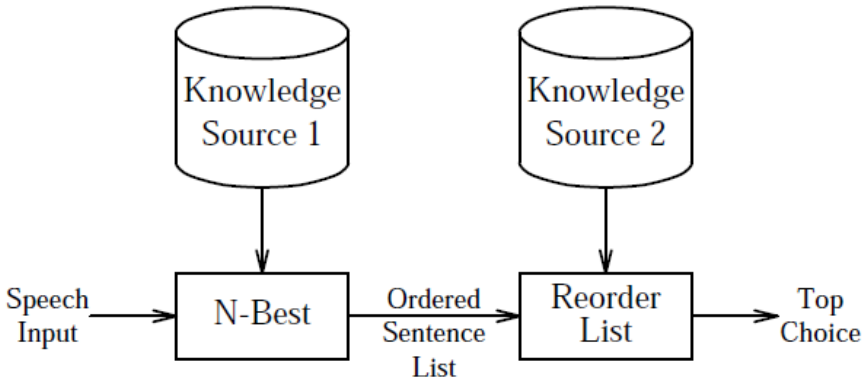
تابستان 1391

1- در بازشناسی گفتار مبتنی بر HMM، الگوریتم‌های جستجوی مختلفی به منظور دیکدینگ سیگنال گفتار (تبدیل دنبالهٔ گفتاری به دنبالهٔ واحدهای آوایی معادل) به کار می‌روند. روش‌های مختلف جستجو که در بحث بازشناسی گفتار به کار می‌روند را بنویسید و نحوهٔ به‌کارگیری آنها را در بازشناسی گفتار با جزئیات شرح دهید.

یک سیستم بازشناسی گفتار باید همه منابع دانشی که در دسترس دارد را موقع بازشناسی یک پاره‌گفتار در نظر بگیرد. علاوه بر سیگنال گفتار و مدل‌های واحدهای بازشناسی، دانش لازم درباره نحو، معنی و دیگر ویژگی‌های زبان طبیعی نیز باید هنگام جستجو برای محتمل‌ترین دنباله کلمات در نظر گرفته شوند. یک راه قرار دادن این منابع دانش در فرایند جستجو این است که از آنها به طور همزمان برای محدودسازی یک جستجوی واحد استفاده شود. از آنجا که بسیاری از منابع دانش زبان طبیعی دارای تأثیرات «فاصله-زیاد» (long-distance) هستند (یعنی وابستگی بین کلماتی در ورودی که خیلی از هم دور می‌باشند)، فرایند جستجو می‌تواند نسبتاً پیچیده گردد. ضمناً استراژی رایج جستجوی چپ به راست مستلزم این است که همه منابع دانش به صورت قابل پیش‌بینی و چپ به راست فرمول‌بندی شوند که همین موضوع نوع دانشی را که می‌توان مورد استفاده قرار داد محدود می‌سازد.

یک راه حل برای این مسأله این است که منابع دانش را نه به طور همزمان که به طور ترتیبی اِعمال نمود بگونه‌ای که جستجو برای محتمل‌ترین فرضیه (hypothesis) مرتباً محدود شود. لذا مزایای بدست آمده از یک منبع دانش را می‌توان با هزینه‌های بکارگیری آن مصالحه نمود (trade off). ابتدا قدرتمندترین و کم‌هزینه‌ترین منابع دانش اِعمال می‌گردند تا لیستی از N فرضیه برتر تولید نمایند. سپس این فرضیه‌ها توسط دیگر منابع دانش پرهزینه‌تر مورد ارزیابی قرار می‌گیرند به نحوی که لیست فرضیه‌ها را می‌توان بنا بر یک معیار مشابهت (likelihood score) پیشرفته‌تر دوباره مرتب‌سازی نمود. کل فرایند جستجوی N-best در شکل 1 ترسیم شده است. تا زمانی که فرضیه صحیح در بین فرضیه‌های N-best باشد، فرایند جستجوی N-best نهایتاً همان فرضیه صحیح را بعنوان جستجویی که شامل همه منابع دانش به طور همزمان است، می‌یابد. «فرضیه صحیح»، محتمل‌ترین فرضیه بنا بر مدل‌های موجود و منابع دانش می‌باشد – این که این تشخیص واقعاً همان گفتاری باشد که صحبت شده است، به کیفیت مدل‌ها و منابع زبانی بستگی دارد **نه** به الگوریتم جستجو.

الگوریتم‌های متفاوتی برای یافتن فرضیه‌های N-best ارائه شده‌اند. برخی از این الگوریتم‌ها دقیق می‌باشند در حالی که بقیه از تخمین‌های مختلفی برای کاهش نیازمندی‌های محاسباتی استفاده می‌کنند.



Knowledge Source 1	Knowledge Source 2
statistical grammar	full natural language model
syntax	semantics, etc.
bi-grammar	higher-order grammar
...	...

شکل 1: فرایند جستجوی N-best. ترکیب منابع دانش با استفاده از الگوریتم N-best

علاوه بر فرایند جستجوی N-best، استفاده‌های دیگری هم برای این الگوریتم‌های N-best وجود دارد:

- لیست‌های فرضیه‌های N-best که در طی آزمون‌های بازشناسی تولید شده‌اند می‌توانند برای بررسی منابع دانش جدید استفاده شوند. از آنجا که لازم نیست تا همه فرایند بازشناسی دوباره اجرا گردد، ارزیابی عملی اطلاعات اضافی یک منبع دانش جدید آسان‌تر خواهد بود.
- روش‌های آموزش متمایز HMM (discriminative) ها معمولاً به لیستی از خطاها نیاز دارد تا جواب صحیح محتمل‌تر گردد و خطاها کم‌تر شوند. چنین لیستی را می‌توان به راحتی توسط یک الگوریتم N-best آماده نمود.
- در یک سیستم بازشناسی گفتار، برخی پارامترها مانند وزن منابع دانش مختلف به راحتی قابل ارزیابی نیست. برای تنظیم دقیق این پارامترها معمولاً به آزمون مکرر بازشناسی نیاز می‌باشد. با استفاده از لیست‌های فرضیه‌های N-best که در طی یک بار اجرای بازشناسی تولید شده‌اند، بهینه‌سازی این پارامترها راحت‌تر خواهد بود.
- نسخه اصلاح‌شده‌ای از الگوریتم N-best درخت-شبکه (tree-trellis) را می‌توان برای تولید واژگان (lexicon) مورد نیاز تشخیص‌دهنده‌های مبتنی بر زیر-کلمه به کار برد.

الگوریتم جستجوی A*

برخی از الگوریتم‌های N-best از توی الگوریتم جستجوی درختی به نام *A استفاده می‌کنند. این الگوریتم غالباً در مسائل مرتبط با هوش مصنوعی به کار می‌رود. بسیاری از مسائل هوش مصنوعی را می‌توان به صورت یک فضای حالت مدل کرد. این امر بدین معنی است که با شروع از یک حالت اولیه، عملگرهایی به توصیف حالت اِعمال می‌شوند تا زمانی که به حالت هدف برسیم. جستجو برای دنباله‌ای از عملیات که حالت اولیه را به حالت هدف تبدیل می‌کند را می‌توان به صورت یک گراف مدل کرد. در این صورت، یک درخت بعنوان نوع خاصی از گراف کفایت می‌کند. هر گره این درخت به یک حالت نسبت داده می‌ود. با اِعمال همه عملگرهای ممکن روی حالت متناظر با یک گره، همه فرزندان آن گره تولید می‌شوند و لذا گره گسترش می‌یابد (expanded). ابتدا گره آغازین که حالت اولیه را در خود دارد گسترش می‌یابد. سپس فرایند گسترش فرزندان آنقدر ادامه می‌یابد تا زمانیکه به یک گره هدف برسیم. نشانگرهای (pointers) از سمت گره‌های مابعد به سمت گره‌های والد خود هنگام گسترش یک گره تنظیم می‌شوند. این نشانگرها این امکان را فراهم می‌آورند تا رد مسیر از گره اولیه به گره هدف را حفظ کنیم.

برای کمینه‌سازی تعداد گره‌های گسترش‌یافته و لذا بهینه‌سازی درخت جستجو، یک روش جستجو ترتیبی را که طبق آن گره‌ها گسترش می‌یابند مشخص می‌کند. با استفاده از اطلاعات هیوریستیک درباره ماهیت کلی درخت جستجو و مسیر کلی هدف، می‌توان با گسترش امیدبخش‌ترین گره‌ها، جستجو را به سمت هدف «کشاند»(pull). این امر با استفاده از یک تابع ارزیابی f و انتخاب گره n برای گسترش بعدی که کمترین f(n) را دارد، میسر می‌شود. این روش جستجوی مرتب از دو لیست OPEN و CLOSE استفاده می‌کند و در شکل 2 شرح داده شده است:

```
put start node s on OPEN
compute f̂(s) while OPEN is not empty
    remove from OPEN the node n whose f̂(n) value is smallest
    put n on CLOSED if n is a goal node
        obtain the solution path by tracing back through the pointers
        exit
    if node n has any successors
        expand node n by generating all of its successors n_i
        compute all f̂(n_i)
        put these successors on OPEN
        provide pointers back to n
exit with failure
```

شکل 2: الگوریتم جستجوی *A

عمق d(n) یک گره طول مسیری است که از گره آغازین s به گره n ادامه دارد. لذا عمق گره آغازین برابر d(s)=0 است، عمق فرزندان آن یعنی s_i برابر $d(s_i)=1$ است و غیره. با داشتن محدودیت عمق d_{max}، تابع ارزیابی $f(n) = -\min\{d(n), d_{max}\}$ به یک جستجوی اول-عمق منجر می‌شود که در آن گره‌های عمیق‌تر اول گسترش می‌یابند. از طرفی دیگر، تابع ارزیابی f(n)=d(n) به یک جستجوی اول-سطح منجر می‌شود. این تضمین می‌کند که کوتاه‌ترین مسیر به سمت گره هدف یافته خواهد شد. هر دو روش جستجویی که ذکر شد، روش‌های کور (blind) است چرا که از هیچ اطلاعات هیوریستیکی استفاده نمی‌کنند.

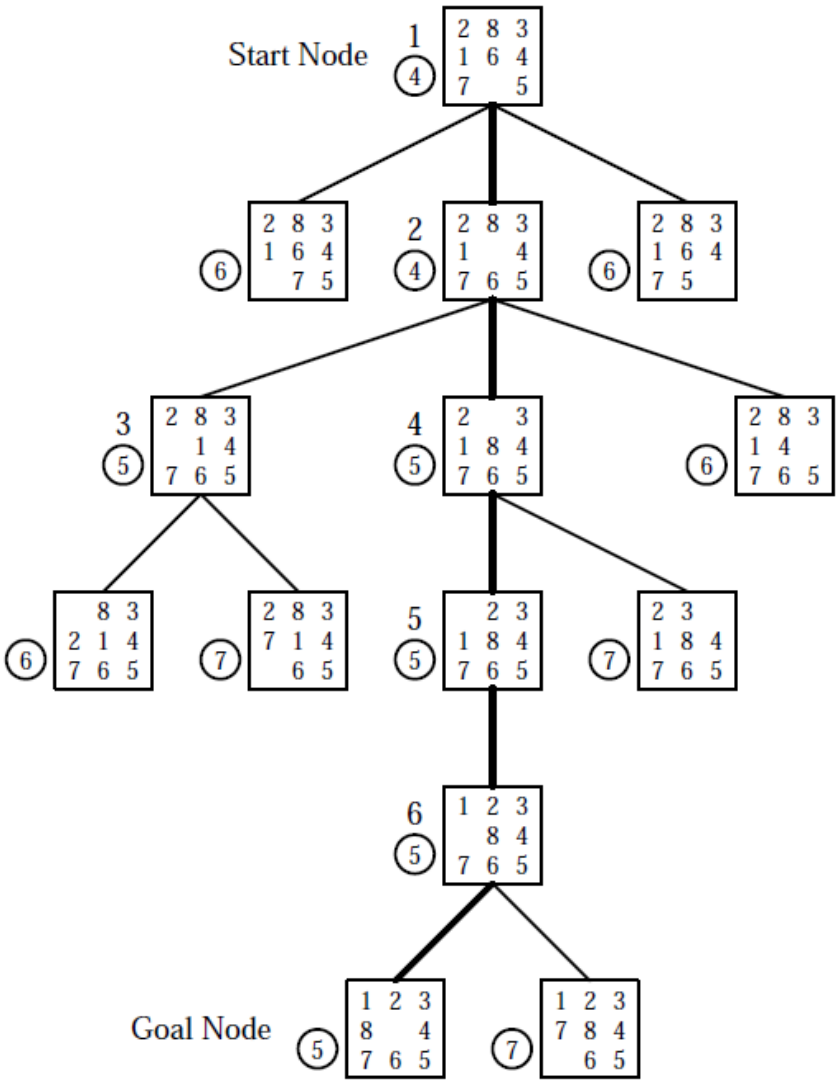
برای بدست آوردن توابع ارزیابی قوی‌تر، باید به یال‌های (arcs) درخت هم هزینه اختصاص دهیم. با فرض تابع هزینه $c(n_i, n_j)$ که هزینه رفتن از گره n_i به فرزندش n_j را می‌دهد، هزینه g(n) مسیر از گره آغازین s به گره n به صورت زیر محاسبه می‌شود:

$$g(s)=0$$

$$g(n_i)=g(n)+c(n, n_i)$$

تابع ارزیابی f(n)=g(n) به یک جستجوی هزینه-یکنواخت منجر می‌شود که هزینه کمینه مسیر گره هدف را پیدا می‌کند. وقتی از یال‌های با هزینه ثابت $c(n_i, n_j)=1$ استفاده می‌کنیم، این روش معادل روش اول-سطح می‌شود.

الگوریتم جستجوی A* یک روش جستجوی بهینه است که کارایی جستجو را بیشینه کرده و همزمان تضمین می‌کند که مسیری با کمترین هزینه به سمت گره هدف می‌یابد. این ویژگی‌ها با استفاده از تابع ارزیابی f(n)=g(n)+h(n) بدست می‌آید. تابع f(n) یک تقریب (estimate) از هزینهٔ کم‌هزینه‌ترین مسیر از گره آغازین s به گره هدف است که از گره n می‌گذرد. این مسیر از دو مسیر تشکیل شده است: مسیری از گره آغازین به گره n و مسیری از گره n به سمت گره هدف. g(n) هرینه اولین مسیر است، کم هزینه‌ترین مسیر از s به n. h(n) تقریبی از هزینه مسیر دوم است، کم هزینه‌ترین مسیر از گره n به گره هدف. تابع h(n) تابع هیوریستیک نام دارد. اگر h(n)=0 باشد هیچ اطلاعات هیوریستیکی بکار نرفته است و الگوریتم A* معادل روش هزینه-یکنواخت می‌شود. در شکل 3 مثالی از الگوریتم A* برای مسأله هشت وزیر آمده است.



الگوریتم *A برای مسأله هشت وزیر

الگوریتم‌های مختلف N-best

الگوریتم ویتربی که معمولاً در بازشناسی گفتار مبتنی بر HMM استفاده می‌شود فقط بهترین دنباله کلمات را می‌یابد (فرضیه متناظر با دنباله حالت با بیشترین معیار مشابهت). برای بدست آوردن نه تنها اولین بهترین فرضیه بلکه لیست بهترین N فرضیه، اصلاحات مختلفی از الگوریتم ویتربی ضروری می‌باشد. الگوریتم‌های مختلفی که قادرند فرضیه‌های N-best را بیابند در ادامه شرح داده می‌شوند.

الگوریتم N-best دقیق (exact N-best algorithm)

این الگوریتم مشابه الگوریتم ویتربی زمان-سنکرون است ولی به جای معیارهای مشابهت برای دنباله حالات، معیارهای مشابهت دنباله‌های کلمات محاسبه می‌شوند. برای اینکه بتوان

فرضیه‌های N-best را پیدا کرد، لازم است تا پیشینه‌های جدایی (records) برای تئوری‌های(مسیرها) با تاریخچه دنباله کلمات متفاوت داشته باشیم. وقتی دو یا چند مسیر در یک زمان به یک حالت منتهی می‌شوند و همچنین تاریخچه (دنباله کلمات) یکسانی دارند، احتمالات آنها جمع می‌شود. وقتی همه مسیرهای یک حالت محاسبه شدند، فقط تعداد خاصی از این تئوری‌های محلی

(local theories) نگه داشته می‌شوند. احتمال آنها باید درون یک آستانه احتمال محتمل‌ترین تئوری(دنباله کلمه) در آن حالت باشد. بنابراین هر فرضیه دنباله کلمه‌ای که به انتهای پاره‌گفتار می‌رسد، دارای یک معیار مشابهت دقیق است. این معیار، احتمال شرطی گفتار مشاهده‌شده به شرط فرضیه دنباله کلمات است. لذا لیست فرضیه‌های N-best تولید می‌شود. برای کاهش رشد نمایی تعداد دنباله کلمات ممکن، از هرس کردن (pruning) استفاده می‌شود. می‌توان نشان داد که این الگوریتم همه فرضیه‌های ممکنی را که در یک پرتو (beam) جستجوی مشخص‌شده با آستانه‌های هرس قرار دارند، پیدا می‌کند.

از آنجا که احتمالات مسیرهایی با تاریخچه دنباله کلمات یکسان، جمع می‌گردد، معیار مشابهت کلی (total likelihood score) محاسبه می‌شود، در مقابل معیار مشابهت بیشینه که توسط الگوریتم ویتربی محاسبه می‌شود. این امر همچنین به نرخ بازشناسی نسبتاً بالاتری می‌انجامد. برای کاهش نیازمندی‌های محاسباتی مرتبط با الگوریتم N-best دقیق، این امکان وجود دارد تا الگوریتم N-best را با الگوریتم جستجوی جلو-عقب (forward-backward) ادغام نمود. الگوریتم جستجوی جلو-عقب در دو مرحله انجام می‌شود. در مرحله اول، یک جستجوی سریع زمان-سنکرون از پاره‌گفتار در جهت جلو انجام می‌شود. در مرحله دوم، یک جستجوی پرهزینه‌تری انجام می‌شود و پاره‌گفتار را در جهت عکس و با استفاده از اطلاعات بدست آمده در جستجوی جلورو پردازش می‌کند. مثلاً یک جستجوی ویتربی در مرحله جلورو انجام می‌شود و سپس جستجوی پرهزینه‌تر N-best دقیق در مرحله عقب‌رو انجام می‌گردد. اطلاعات جستجوی جلورو برای پرهیز از گسترش درخت جستجوی عقب‌رو به فرضیات بدردنخور بکار می‌رود و لذا هزینه محاسبات را کاهش می‌دهد. این مفهوم ترکیب جستجوی جلورو و عقب‌رو مشابه مفهوم الگوریتم درخت-شبکه است.

الگوریتم درخت-شبکه (Tree-Trellis Algorithm)

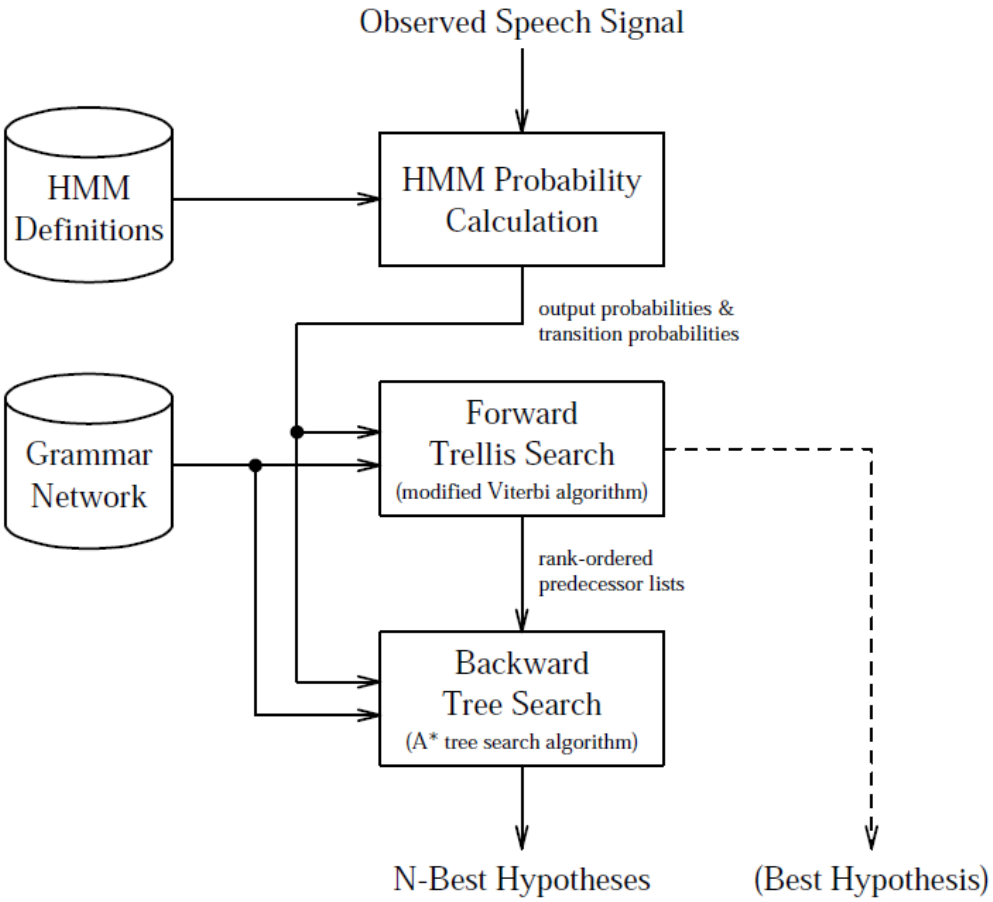
این الگوریتم یک جستجوی شبکهٔ جلوروی فریم-سنکرون را با یک جستجوی درختی عقب‌روی فریم-آسنکرون ترکیب می‌کند. در جستجوی جلوروی شبکه، از یک الگوریتم ویتربی اصلاح‌شده استفاده می‌شود. در الگوریتم ویتربی نرمال، فقط آرایه‌های پس-نشانگر

(backpointer) برای ردیابی بهترین فرضیه ضروی می‌باشند. الگوریتم اصلاح‌شده همچنین لیست‌های مرتب شده بر اساس امتیاز (rank-ordered) والدها را برای هر گره گرامر و فریم زمان ذخیره می‌سازد. برای هر گره گرامر و فریم زمان، چنین لیستی یک مدخل برای هر والد آن گره گرامر دارد. این مدخل شامل معیار مشابهت بهترین مسیر جزئی (partial) از آن والد به گره گرامر می‌باشد. مدخل‌ها قبل از این که ذخیره شوند، بر اساس معیارهای مشابهت خود امتیازبندی و مرتب می‌شوند. وقتی جستجوی ویتربی اصلاح‌شده به انتهای پاره‌گفتار برسد، بهترین فرضیه را می‌توان به راحتی با ردیابی (backtrack) بدست آورد.

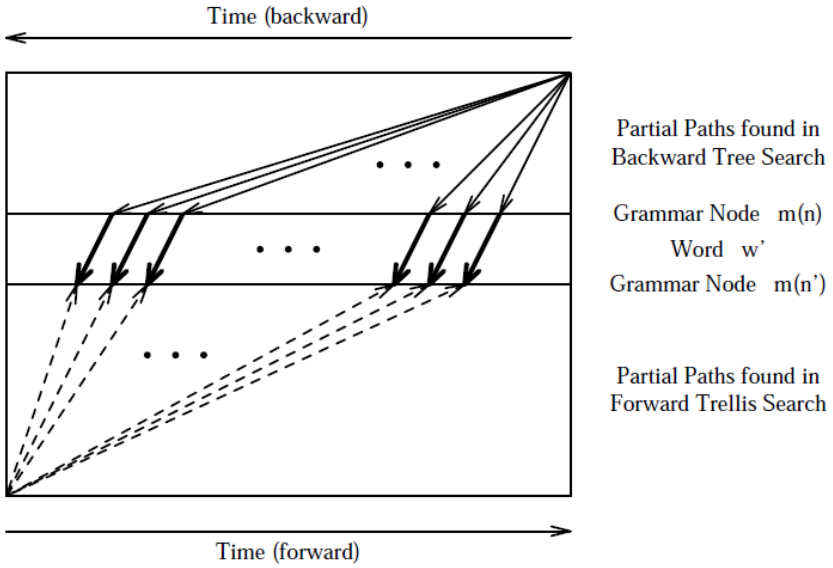
در جستجوی عقب‌رو از یک الگوریتم A* برای یافتن فرضیه‌های N-best استفاده می‌شود. این جستجوی درختی از انتهای پاره‌گفتار در آخرین گره گرامر آغاز می‌شود. در هر مرحله، مسیر عقب‌روی جزئی به سمت ابتدای پاره‌گفتار گسترش می‌یابد، با استفاده از یک جستجوی ویتربی زمان-معکوس برای گسترش بهترین تک کلمه (single word). گسترش بهترین تک کلمه با استفاده از لیست‌های والدین مرتب‌شده بر اساس امتیاز و تولید شده در طی جستجوی جلورو، یافته می‌شود. وقتی مسیر عقب‌روی جزئی به ابتدای پاره‌گفتار می‌رسد، بهترین فرضیه یافته می‌شود. با ادامه دادن جستجوی درختی A*، فرضیه‌های N-best را به ترتیب می‌توان یافت. نمودار کل الگوریتم درخت-شبکه در شکل 4 آمده است.

جستجوی درختی عقب‌رویی که در این جا انجام می‌شود پیچیده‌تر از جستجوی A* معمولی است چرا که جنبه‌های مرتبط با زمان در این جا متفاوت و متغیر است. یعنی معمولاً مسیرهای بسیاری برای یک دنباله کلمه یکسان با منحنی‌ها (trajectories) و معیارهای متفاوت وجود دارد. از آنجا که فقط مسیرهای کلمات متفاوت در نظر گرفته می‌شود، مسیرهایی با محتوای کلمه یکسان ولی منحنی‌های زمانی متفاوت ابتدا مقایسه می‌شوند. امتیاز بهترین مسیر ممکن سپس با بقیه مسیرهای با محتوای کلمه متفاوت سنجیده می‌شود. در شکل 5 جنبه‌های متغیر با زمان نشان داده شده است. در این حالت، دنباله کلمات تا گره گرامر m(n) که در جستجوی درختی عقب‌رو یافت شده است ابتدا با بهترین کلمه w’ تا گره گرامر m(n’) گسترش داده می‌شود. این امر توسط یک الگوریتم ویتربی زمان-معکوس که بهترین مسیرهای جزئی به گره گرامر m(n) را در فریم‌های زمانی متفاوت گسترش می‌دهد و لذا بهترین مسیرهای منتهی به گره گرامر m(n’) را برای فریم‌های زمانی متفاوت پیدا می‌کند، انجام می‌شود. سپس این مسیرهای جزئی عقب‌روی گسترش‌یافته به گره گرامر m(n’) با مسیرهای جزئی جلوروی منتهی به گره گرامر m(n’) که در جستجوی ویتربی اصلاح‌شده یافت شده بودند، ادغام می‌شوند. یعنی برای همه فریم‌های زمانی، امتیاز مسیر جزئی عقب‌رو با امتیازات مسیر جزئی

جلورو برای همه والدین گره گرامر m(n') که در طی جستجوی ویتربی اصلاح‌شده ذخیره شده بودند، ترکیب می‌شود. نهایتاً بالاترین امتیاز مشابهتِ ترکیب‌شده برای هر والد ممکن پیدا می‌شود و بنابراین بهترین گسترش کلمه برای مرحله بعدی یافت می‌گردد.



شکل 4: نمودار کل الگوریتم درخت-شبکه



شکل 5: الگوریتم درخت-شبکه

الگوریتم کامل جستجوی درختی عقب‌روی *A در شکل 6 آمده است. یک مدخل پشته (گره) n که در این جستجوی *A بکار می‌رود از یک لیست مرتب‌شده بر اساس امتیاز I(n) گسترش کلمه ممکن w(n,i) و امتیازات مسیر کلی آنها f(n,i) (که $f(n,i)\geq f(n,j)$ for $1\leq i<j\leq I(n)$)، اندیس $i_{next}(n)$ برای بهترین گسترش کلمه‌ای که تا کنون انجام نشده، آرایه‌ای شامل امتیازهای مشابهت g(n,t) از مسیرهای جزئی عقب‌رو برای هر فریم زمان t و یک پس-نشانگر به مدخل پشته قبلی $n_{pre}(n)$، تشکیل شده است. لیست‌های مرتب‌شده بر اساس امتیاز J(m(n)) والد ممکن p(m(n),j) گره گرامر m(n) و امتیازات مسیر جزئی جلوروی آنها h(m(n),t,j) در فریم زمان t در جستجوی جلورو تولید شده‌اند. m(n) گره گرامری است که مسیرهای جزئی عقب‌روی n به آن منتهی شده‌اند. مدخل‌های n پشته OPEN بر حسب $f(n, i_{next}(n))$ مرتب شده‌اند. از آنجا که الگوریتم ویتربی اصلاح‌شدهٔ بکاررفته در جستجوی جلورو با الگوریتم ویتربی معمولی فقط در تولید اضافی لیست‌های والدها تفاوت دارد، این مسأله این جا نشان داده نشده است. گره آغازین s برای جستجوی درختی عقب‌رو، گره خروجی m(s) EXIT شبکه گرامر است. پاره‌گفتار از فریم‌های زمانی $t=1,2,\ldots,T$ تشکیل شده است. در جستجوی ویتربی زمان-معکوس، c(n,n',t,t') نمایانگر امتیازهای مشابهت بهترین مسیر جزئی برای کلمه w' از گره گرامر m(n) به گره گرامر m(n') می‌باشند که در فریم زمانی t آغاز شده‌اند و در فریم زمانی t' پایان یافته‌اند. تمام امتیازهای مشابهت بکاررفته، لگاریتمی می‌باشند. لذا وقتی مسیرهای جزئی ترکیب می‌شوند، اینها با هم **جمع** می‌شوند.

```
/***** initialise start node s *****/
w(s,i) = p(m(s),i),    i = 1,2,...,J(m(s))
f(s,i) = h(m(s),T,i),    i = 1,2,...,J(m(s))
I(s) = J(m(s))
i_next(s) = 1
g(s,t) = -∞,    t = 0,1,...,T-1
g(s,T) = 0
n_pre(s) = NIL
put start node s on OPEN
N_found = 0
while OPEN is not empty
    remove from OPEN the top entry n (having maximum f(n,i_next(n)))
    take the single word extension w' = w(n,i_next(n)) to the best ungrown successor n'
    increment i_next(n)
    if i_next(n) > I(n)
        put n on CLOSED
    else
        reinsert n in OPEN according to f(n,i_next(n))
    if n' is a goal node (i.e. m(n') is network ENTER node)
        increment N_found
        obtain the N_found'th hypothesis by tracing back through the pointers n_pre(n)
        if N_found = N
            exit
    else
        /***** grow successor node n' by expanding n by the single word w' *****/
        /* do time-reverse Viterbi search for word w' from m(n) to m(n') */
        g(n',t') = max_{t=t'+1,t'+2,...,T}(g(n,t) + c(n,n',t,t')),    t' = 0,1,...,T-1
        g(n',T) = -∞
        /* merge backward partial paths to m(n') with all predecessors */
        w(n',i) = p(m(n'),i),    i = 1,2,...,J(m(n'))
        f(n',i) = max_{t=0,1,...,T}(g(n',t) + h(m(n'),t,i)),    i = 1,2,...,J(m(n'))
        I(n') = J(m(n'))
        sort f(n',i) and w(n',i) so that f(n',i) ≥ f(n',j) for 1 ≤ i < j ≤ I(n')
        i_next(n') = 1
        provide backpointer n_pre(n') = n
        insert n' in OPEN according to f(n',i_next(n'))
exit with failure
```

شکل 6: الگوریتم جستجوی درختی عقب‌رو برای یافتن فرضیه‌های N-best

برخلاف جستجوی درختی A* معمولی که در آن امتیاز قسمت ناتمام یک مسیر توسط یک تابع هیوریستیک h(n) تخمین زده می‌شود، جستجوی درختی مورد بحث در اینجا از امتیاز دقیق h(n) که در لیست والدهای مرتب‌شده بر اساس امتیاز ذخیره شده است استفاده می‌کند. این لیست در جستجوی ویتربی جلورو تولید می‌شود. بنابراین بیشترین بهینگی جستجوی درختی A* بدست می‌آید که بدین معنی است که فقط گسترش‌های ضروری در مسیر انجام می‌شود. علاوه بر این، تابع ارزیابی f(n)=g(n)+h(n) امتیاز دقیق مسیر کامل را در طی کل جستجوی درختی A* برای یک فرضیه بدست می‌دهد. لذا این امکان وجود دارد که اندازه پشته OPEN

را به اندازه N فرضیه‌ای که قرار است یافته شوند، محدود نمود. این واقعیت که در اینجا به جای *هزینه‌ها* از *امتیازها* استفاده می‌شود، بدین معنی است که گره با بالاترین امتیاز، نه گره با کمترین هزینه، ابتدا توسط الگوریتم A* گسترش داده می‌شود. تفاوت دیگری که با جستجوی درختی A* وجود دارد این است که یک گره فقط به بهترین گره فرزند خود گسترش داده می‌شود نه به همه گره‌های فرزند.

الگوریتم‌های تقریبی N-best (Approximate N-best Algorithms)

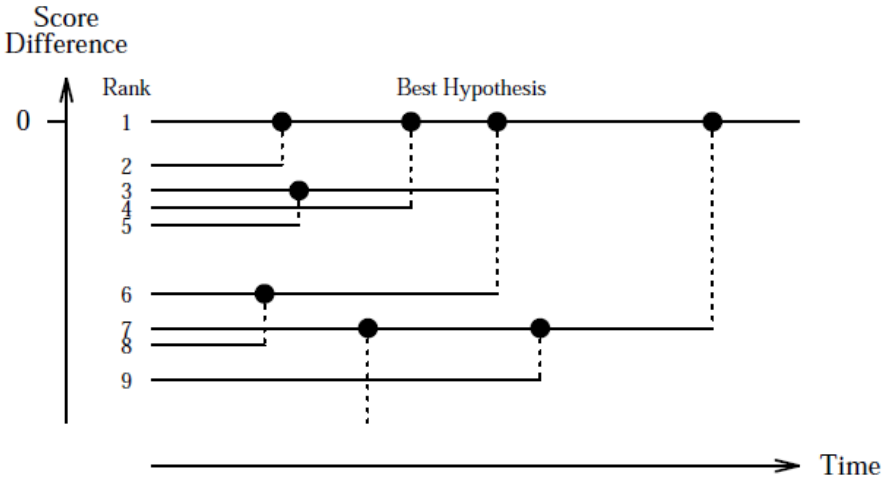
هم الگوریتم N-best و هم الگوریتم درخت-شبکه، هر دو به مقدار قابل توجهی از محاسبات علاوه بر میزان محاسبات معمولی در جستجوی ویتربی عادی نیاز دارند. به همین دلیل، الگوریتم‌های N-best سریعتری که مبتنی بر تقریب می‌باشند، ارائه شده‌اند. این الگوریتم‌ها تضمین نمی‌کنند که لیست دقیق فرضیه‌های N-best را بیابند. ممکن است امتیاز مشابهت یک مدخل کمتر از میزان واقعی‌اش تخمین زده شود یا حتی کلاً نادیده گرفته شود. ولی لیست تقریبی باز هم برای بسیاری از کاربردها کافی می‌باشد. دو نوع الگوریتم N-best با تقریب‌ها متفاوت در ادامه معرفی می‌شوند.

الگوریتم مشبک (Lattice Algorithm)

این الگوریتم مبتنی بر یک جستجوی ویتربی جلوروی زمان-سنکرون می‌باشد ولی در اطلاعات پس-اشاره‌گر ذخیره شده در طی جستجو با هم فرق دارند. در هر گره گرامر برای هر فریم زمانی، نه تنها کلمهٔ با بهترین امتیاز بلکه همه کلماتی که به آن گره منتهی می‌شوند همراه با امتیازاتشان و زمانی آغاز کلمه، در یک لیست ردیابی (trackback) ذخیره می‌شوند. به جای همه کلمات رسیده شده (arriving) این امکان وجود دارد که فقط بهترین N_{local} کلمه را ذخیره نمود. مانند جستجوی ویتربی معمولی، فقط امتیاز بهترین کلمه همراه با نشانگری به لیست ردیابی، بعنوان معیاری برای امتیازدهی بیشتر، در نظر گرفته می‌شود. در انتهای پاره‌گفتار، یک جستجوی درختی ساده بکار می‌رود برای واکاوی در لیست‌های ردیابی ذخیره شده و بدست آوردن ترتیبی جملات کامل N-best. این جستجوی درختی تقریباً به محاسباتی نیاز ندارد و می‌تواند خیلی سریع اجرا شود. تئوری ردیابی N-best که با جستجوی درختی انجام می‌شود در شکل 7 به تصویر کشیده شده است. در این شکل هر نقطه نشانگر آغاز یک کلمه و هر خط نقطه‌چین عمودی نشانگر یک لیست ردیابی ذخیره‌شده است.

الگوریتم مشبک را می‌توان به سادگی توسط یک مفهوم ردکنندهٔ واحد (token passing) تعمیم‌یافته پیاده‌سازی نمود که در آن نه تنها یک بهترین بلکه همه بهترین والدها ذخیره می‌شوند. الگوریتم مشبک را همچنین می‌توان با امتیازدهی مشابهت کلی (total likelihood scoring) یا مانند یک جستجوی ویتربی عادی با امتیازدهی مشابه بیشینه (maximum likelihood scoring) بکار برد.

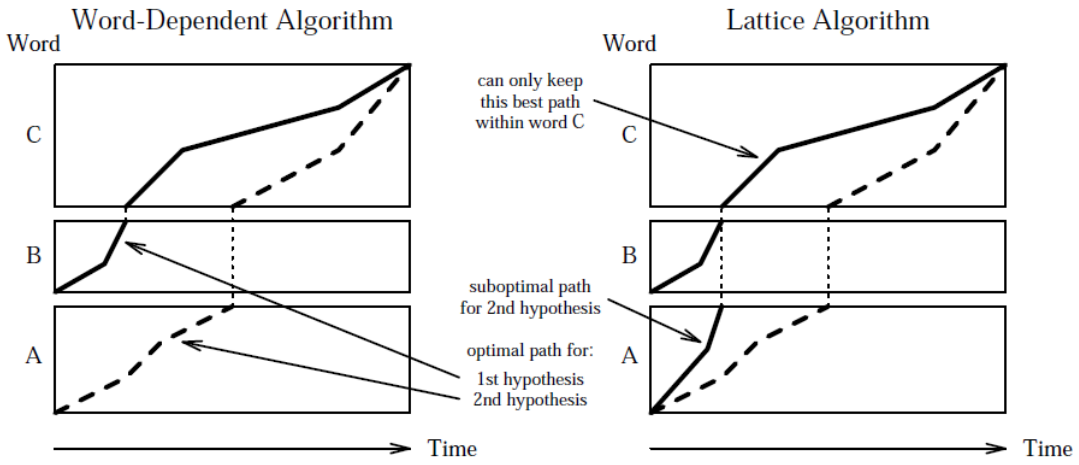
یک عیب عمده الگوریتم مشبک این است که فرضیه‌های با امتیاز با بالا را دست کم یا به کلی نادیده می‌انگارد، به خاطر این واقعیت که همه فرضیه‌ها (بجز بهترین) از تقطیع (segmentation) موجود برای دیگر فرضیه‌های با امتیاز بالا استخراج می‌شوند. این مسأله به دلیل این فرض است که زمان آغاز کلمه به کلمه قبلی بستگی ندارد – فرضی که در ماهیت و ذات الگوریتم مشبک وجود دارد. این مشکل را می‌توان با الگوریتم وابسته به کلمه حل نمود.



شکل 7: تئوری ردیابی N-best در الگوریتم مشبک

الگوریتم وابسته به کلمه (Word-Dependent Algorithm)

این الگوریتم مصالحه‌ای بین الگوریتم N-best دقیق (که الگوریتم وابسته به جمله است) و الگوریتم مشبک می‌باشد. در این حالت فرض می‌شود که زمان آغاز یک کلمه به کلمه قبلی وابسته است ولی به کلمات قبل از آن وابسته نیست. لذا تئوری‌ها وقتی متمایز می‌شوند که در کلمه قبلی با هم فرض داشته باشند. مزایای این روش در مقایسه با الگوریتم مشبک در شکل 8 نشان داده شده است.



شکل 8: مقایسه الگوریتم وابسته به کلمه و الگوریتم مشبک

درون یک کلمه، امتیازات مشابهت هر کدام از تئوری‌های محلی (کلمات قبلی) نگه داشته می‌شوند. در پایان هر کلمه، امتیاز مشابهت هر کلمه قبلی همراه با نام آن کلمه قبلی ذخیره می‌شود. سپس یک تئوری ساده همراه با اسم کلمه‌ای که تازه به پایان رسیده برای پیشرفت کار بکار می‌رود. در انتهای پاره‌گفتار، یک جستجوی درختی (مشابه آن چه که در الگوریتم مشبک بکار رفت) انجام می‌شود تا لیست N فرضیه محتمل بدست آید. برای کاهش نیازمندی‌های محاسباتی، تعداد N_{local} تئوری که به طور محلی نگهداری می‌شوند (یعنی درون یک کلمه) باید محدود شود. مقادیر رایج برای N_{local} از 3 تا 6 متغیر است. همچنین الگوریتم وابسته به کلمه را می‌توان همراه با *امتیازدهی مشابهت کلی* یا *امتیازدهی مشابهت بیشینه* بکار برد.

مقایسه الگوریتم‌های مختلف N-best

تمام الگوریتم‌های N-best که مورد بحث قرار گرفتند ویژگی‌ها و معیبی دارند. لذا بهترین الگوریتم برای یک کار خاص را می‌توان بسته به نیازمندی‌های آن کار انتخاب نمود. ویژگی‌های اصلی الگوریتم‌های مختلف به شرح ذیل می‌باشد:

- الگوریتم N-best دقیق: این الگوریتم به میزان قابل ملاحظه‌ای محاسبات بیشتر نسبت به جستجوی ویتربی عادی نیاز دارد. این الگوریتم همچنین این امکان را فراهم می‌آورد تا به جای *امتیازدهی مشابهت بیشینه* از *امتیازدهی مشابهت کلی* استفاده شود.
- الگوریتم درخت-شبکه: این الگوریتم فقط تا حدی به محاسبات بیشتری نسبت به یک جستجوی ویتربی معمولی نیاز دارد. این الگوریتم لیست دقیق فرضیه‌های N-best را می‌یابد و اگر N بزرگ نباشد روش مناسبی است چرا که محاسبات جستجوی درختی عقب‌رو نسب مستقیم با N دارد.

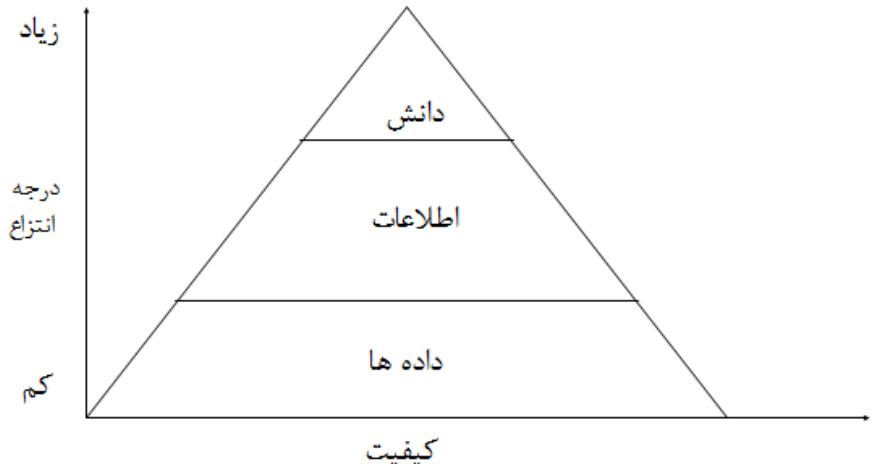
- الگوریتم مشبک: این الگوریتم سریعترین است و فقط به میزان بسیار اندکی محاسبات بیشتر از جستجوی ویتربی عادی نیاز داد. این الگوریتم ممکن است چندین فرضیه را دست کم یا نادیده انگارد ولی می‌تواند در مقادیر بزرگ N به راحتی استفاده شود.
- الگوریتم وابسته به کلمه: این الگوریتم مصالحه‌ای بین الگوریتم دقیق و الگوریتم مشبک است. این الگوریتم به محاسبات کمتری نسبت به الگوریتم دقیق نیاز دارد و در عین حال تضمین می‌کند که لیست نسبتاً دقیقی از فرضیه‌های N-best بدست دهد. این الگوریتم را هم می‌توان در مقادیر بزرگ N بکار گرفت.

2- برای بازنمایی دانش (knowledge representation) روش‌های مختلفی وجود دارد. روش‌های مبتنی بر منطق گزاره‌ای و منطق مرتبهٔ اول در طول درس ذکر شد. سایر روش‌های بازنمایی دانش را شرح داده و آنها را از جنبه‌های مختلف بررسی کنید.

از منابع دانش **می‌توان به** مستندات (کتاب،راهنماها،...)،غیر مستندات (در ذهن مردم،از ماشین ها،...)، کسب دانش از پایگاه داده هاي موجود **و** کسب دانش از اینترنت **اشاره نمود.** سطوح دانش **عبارتند از** دانش سطحي(Shallow Knowledge) **و** دانش عمقي Deep)

(Knowledge**.** برخي روش هاي ارائه دانش از قبيل شبكه هاي معنايي و فريم‌ها امكان پيادەسازي استدلال در سطح عميق(انتزاع و قياس) را ميسر مي كنند.

از انواع دانش **می‌توان به** دانش اعلاني(Descriptive knowledge)، دانش رويه اي(Procedural Knowledge) **و** فرا دانش(Meta Knowledge) **اشاره نمود كه در ادامه به شرح آن‌ها خواهيم پرداخت.**



دانش اعلاني

يک ارائه توصيفي از دانش است كه در اظهارات علمي بيان مي شود، سطحي است، با اهميت است **و** دانش توصيفي در ارتباط با يک شئ خاص **می‌باشد. دانش اعلانی** مرحله ابتدايي در يادگيري دانش مي باشد**.**

دانش رويه‌اي

دانش رويه‌اي به رويه‌هايي كه در فرايند حل مسئله به كار گرفته شده‌اند مرتبط است **و** ترتيب قدم به قدم و چگونگي انجام را شامل مي شود **و نیز** ممكن است شامل توضيحات باشد**. دانش**

رویه‌ای همچنین پاسخ هاي خودكار را شامل مي شود **و** ممكن است چگونگي استفاده از دانش سطحي و چگونگي استنباط كردن را بگويد**.**

فرا دانش

دانشي در رابطه با ساختار، جايگاه و مشخصات دانش موجود **می‌باشد.**

باز نمايي دانش

دانش اكتساب شده جهت محاسباتي شدن سازمان دهي مجدد ميگردد **و** هدف از بازنمايي دانش، تغيير syntax با كامل نمودن معني است. يک باز نمايي خوب، دانش را به صورت طبيعي در دامنه مسأله ارائه مي‌دهد. **از انواع بازنمایی‌ها به موارد زیر می‌توان اشاره نمود:**

- بازنمايي منطقي(Logical Representation) - در مرتبه اول به جبر،prolog و دانش اعلاني اطلاق مي‌شود**.**
- بازنمايي رويه‌اي(Procedural Representation)- يک مجموعه از دستورالعمل‌ها براي حل يک مسأله مثل يک سيستم فروش **می‌باشد.**
- بازنمايي شبكه‌اي(**N**etwork Representation)- دانش در يک ساختار گرافي قرار دارد مثل وابستگي مفهومي و گراف هاي مفهومي**.**
- بازنمايي ساختاري(structural Representation)- يک بسط از شبكه مثل اسكريپت‌ها و فريم‌ها**.**

باز نمايي منطقي

عبارت است از فرم كلي از هر پردازش منطقي **و** ورودي‌**های آن** فرض‌هاي منطقي **هستند. این** فرضيات منطقي براي توليد خروجي و نتيجه‌گيري به وسيله پردازش منطقي استفاده **می‌شوند.**(استنباط)**.** حقايق مي توانند براي مشخص كردن حقايق جديد كه صحيح‌اند استفاده شوند.

منطق نمادين

به سيستم قوانين و رويه‌**هایی** كه **امكان** استنباط از فرضيات منطقي را **فراهم می‌آورند اطلاق می‌گردد**. فرم‌هاي اساسي منطق محاسباتي **عبارتند از**:

- منطق گزاره اي (Propositional Logic)

- منطق گزاره نما(Predicate Knowledge)

در منطق گزاره‌اي گزاره عبارتي است كه صحيح يا غلط **است و از** قوانيني براي مشخص كردن صحيح يا غلط بودن يک گزاره جديد استفاده مي‌شود. جبر گزاره‌نما يک عبارت را به قسمت‌هاي جزئي مي‌شكند **و** از متغيرها و توابع متغير در يک عبارت نمادين منطقي استفاده مي‌كند. جبر گزاره نما پايه prolog است (برنامه نويسي در منطق). مثال‌هاي عبارات prolog :

Likes(jay,chocolate)

لیست‌ها

مجموعه‌اي از عناصر مرتبط **هستند که** به صورت طبيعي براي ارائه سلسله مراتبي دانش استفاده مي‌شوند. **در ليست‌ها** اشياء بر اساس درجه يا ارتباط گروه‌بندي يا مدرج مي‌شوند.

درخت‌هاي تصميم‌گيري

به درخت‌هاي تصميم‌گيري در تئوري تصميم‌گيري شبيه هستند **و** مي‌توانند فرايند يادگيري دانش را ساده کنند.

در اين درخت‌ها، ارائه دانش به صورت دياگرام **است.**

سه‌تايي O-A-V

شامل اشياء، صفات و مقادير **هستند.** اشياء مي توانند فيزيکي يا مفهومي باشند، صفات مشخصه اشياءاند **و**مقادير اندازه خاص صفات **مي‌باشند.**

نمايش آيتم‌هاي O-A-V، مثال:

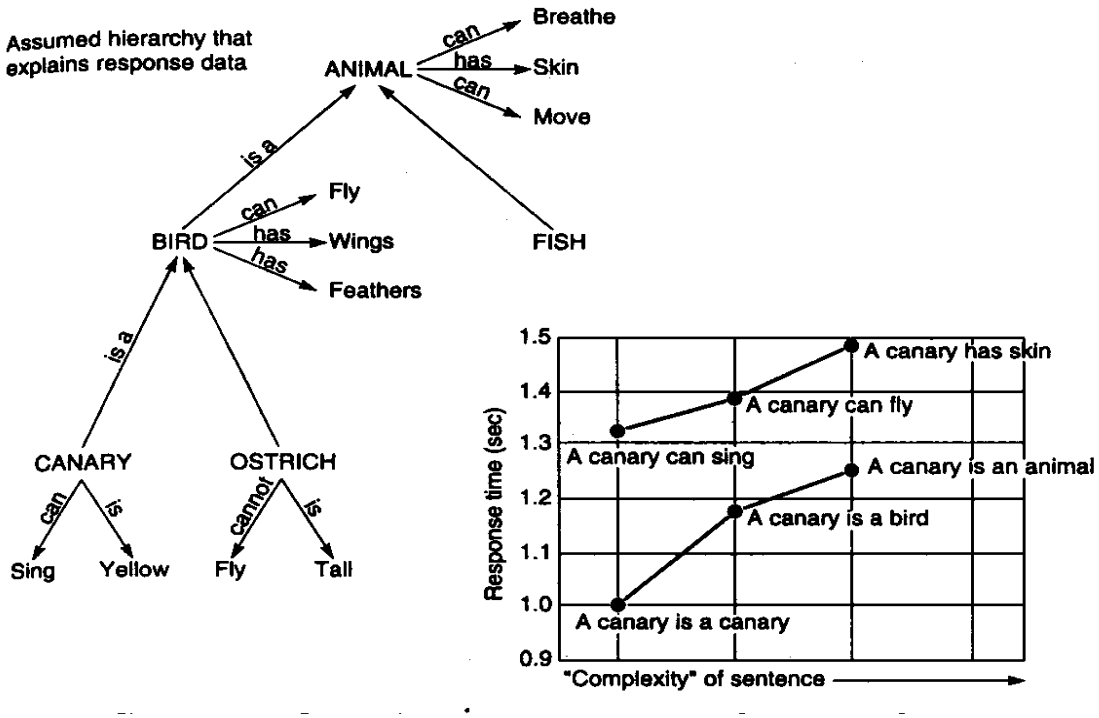
Representative O-A-V Items

Object	Attributes	Values
House	Bedrooms	2, 3, 4, etc.
House	Color	Green, white, brown, etc.
Admission to a university	Grade-point average	3.0, 3.5, 3.7, etc.
Inventory control	Level of inventory	14, 20, 30, etc.
Bedroom	Size	9 X 10, 10 X 12, etc.

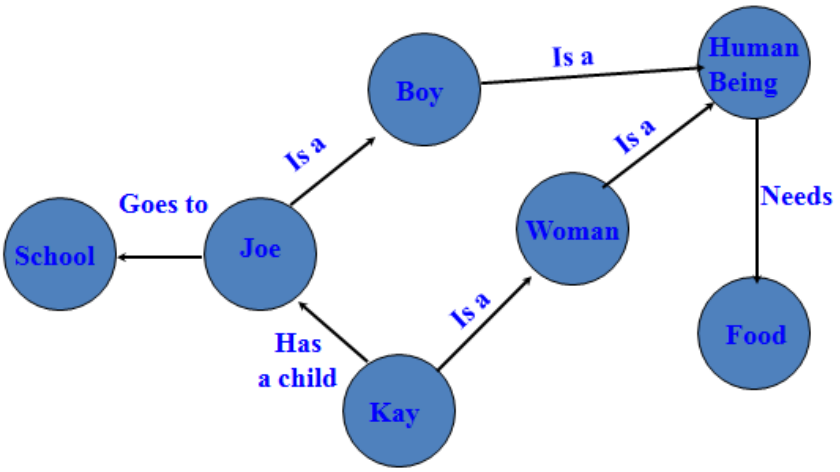
شبکه‌هاي معنايي

شبکه‌هاي معنايي به وسيله Quilian در قبل از 1960 براي ارائه دانش به عنوان يک شبکه از تخصيصات ارائه شده‌اند. **اين شبکه‌ها** نمايش گرافيکي از **دانش را فرام آورده و در آنها** گره‌ها و لينک‌ها ارتباطات سلسله مراتبي بين اشياء را نشان مي‌دهند. گرهها **نمايانگر** اشياء **و** لينک‌ها **نمايانگر** ارتباطات **مي‌باشند. از روابط مي‌توان به** Is-a **و** Has-a **اشاره نمود.**

شبکه هاي معنايي مي توانند وراثت را نشان دهند، ارائه بصري از ارتباطات **را نشان دهند و نيز** مي‌توانند با ساير روش‌هاي نمايش ترکيب شوند **مثال در شکل زير:**



مثالي از شبکه هاي معنايي **در شکل زير آمده است:**

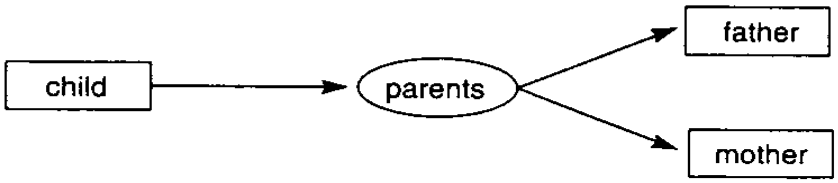


گراف هاي مفهومي(Conceptual Graphs)

ساختار گرافی **دارند و در آن‌ها** لینک‌ها برچسب ندارند. گره‌ها ارتباطي مفهومي بين مفاهيم ارائه شده‌اند **و** در ترسيم، مفاهيم در جعبه‌ها **(مستطيل‌ها)** و ارتباطات مفهومي در بيضي‌ها نمايش داده مي‌شوند. مفاهيم ممكن است حقيقي باشند (**مثل** بچه،سگ،...) يا مجازي **باشند (مثل** عشق،زيبايي،...). مثالي از ارتباطات يک، دو و سه تايي **در شكل زير نشان داده شده است:**

Flies is a 1-ary relation.

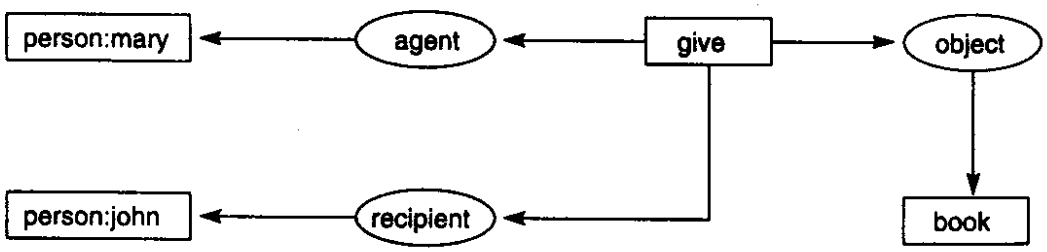
Color is a 2-ary relation.



Parents is a 3-ary relation.

گراف يک جمله

در يک وابستگي مفهومي فعل يک نقش مركزي در ساختار را نمايش مي‌دهد. **مثلاً** فعل "give" در اين جمله يک عامل، يک شئ و يک گيرنده دارد**:**



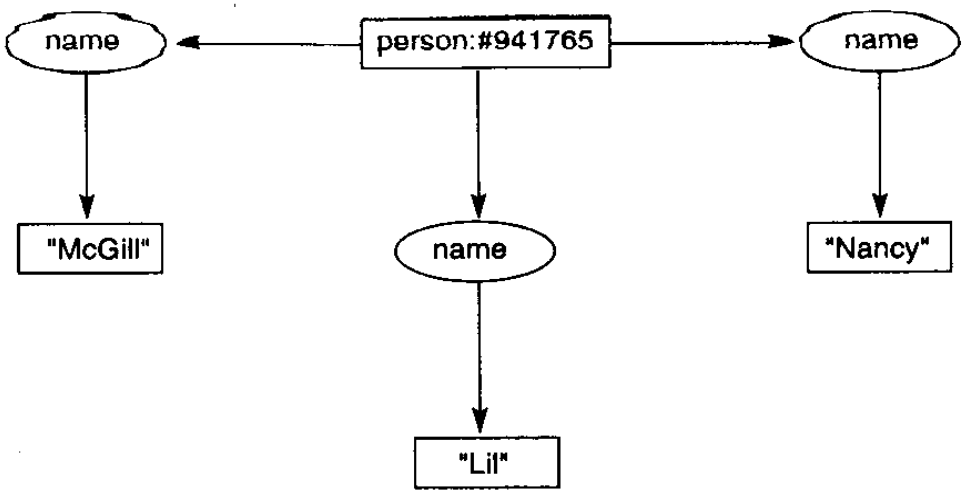
انواع و انحصارات فردي

در **شکل زیر** نوع سگ است و انحصار فردي آن emma :

یک سگ مشخص شده اما بدون نام، با یک شماره:

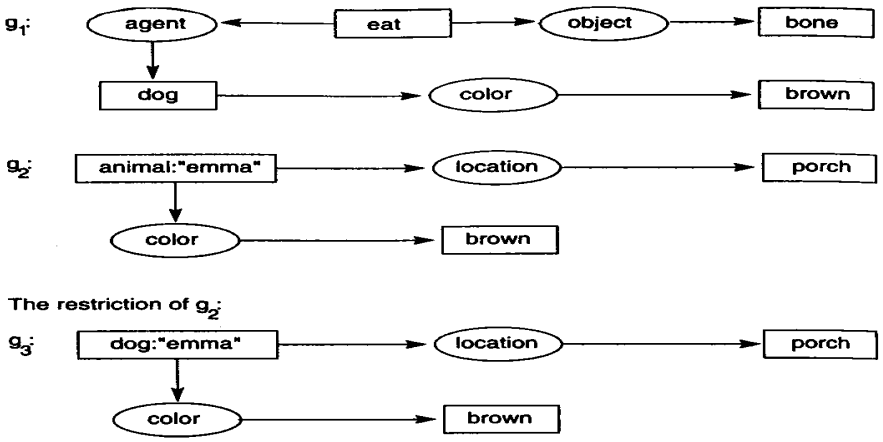
نمايشي از يک سگ مشخص شده به وسيله # و اضافه کردن يک ارتباط مفهومي براي نام:

مثال دیگر: نامش McGill بود و خودش را Lil صدا مي زد اما همه او را با Nancy مي شناختند. هنرمند کي بود؟ نام آواز چه بود؟



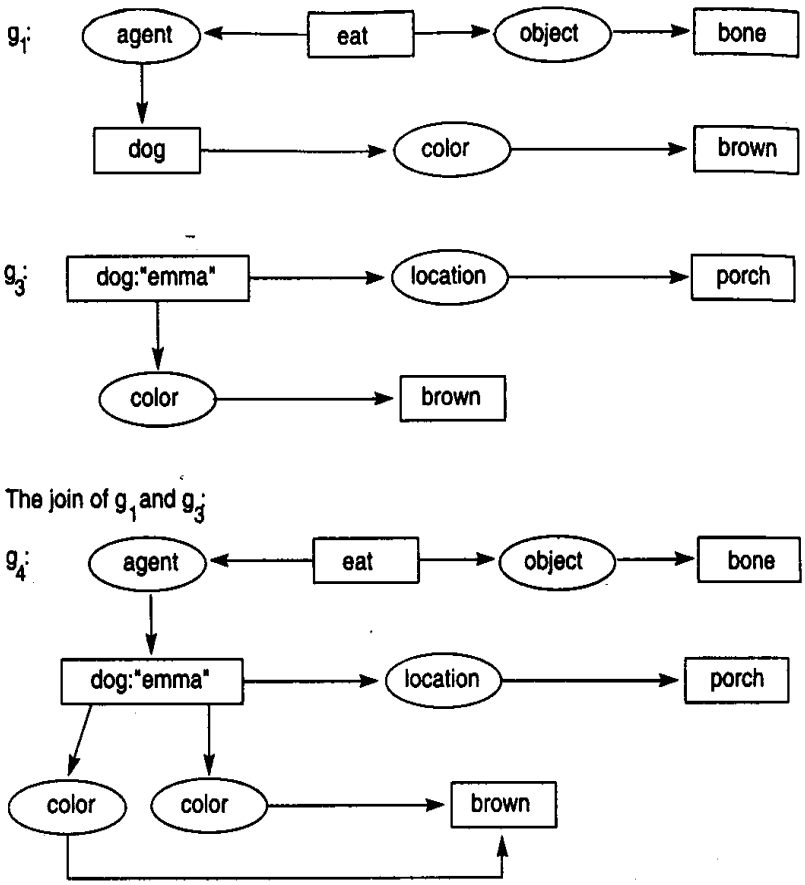
عمومیت و تخصیص (Generalization and Specialization)

به معنی کلی‌سازی و تعمیم مفاهیم و یا بالعکس محدودسازی آنها می‌باشد. مثال:



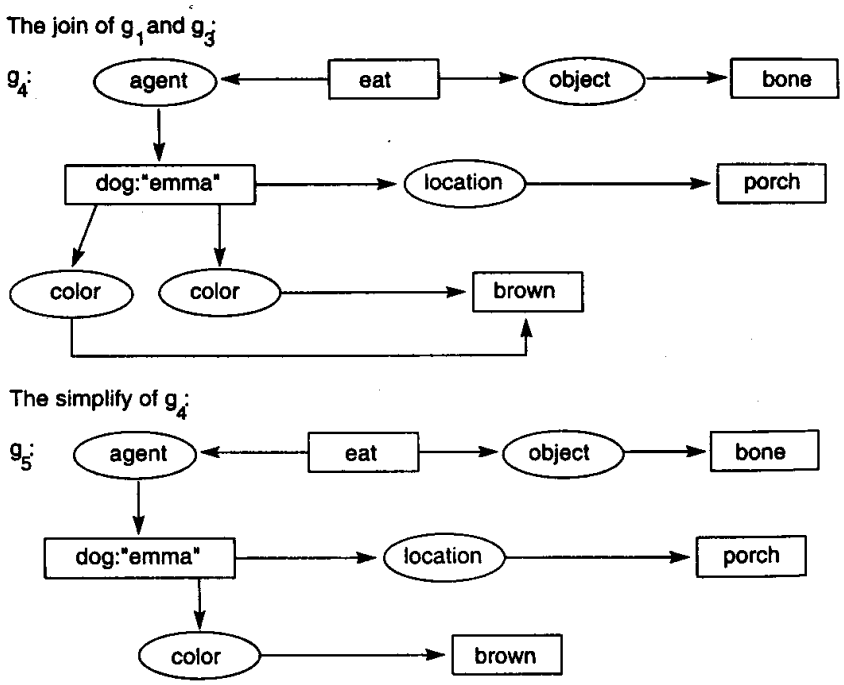
اتصال مفاهیم

اگر دو گراف شامل یک نود یکسان باشند مي توانند به هم متصل شوند. **وقتی** که گراف برآیند ازگراف اصلي خاص‌تر است، اتصال به هم نوعي محدوديت **می‌باشد**. **مثال:**



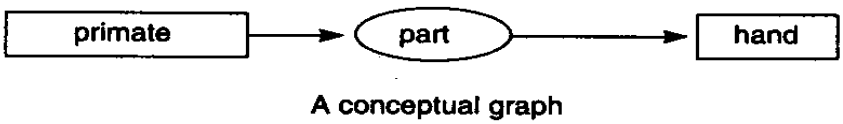
سادەسازي

اتصال دو گراف ممكن است منتج به نتايج تكراري شود **به همين دليل** عملگر سادەسازي باعث حذف اطلاعات تكراري مي‌شود **و از آن** براي رفع ابهام **استفاده مي‌گردد**. **مثال:**

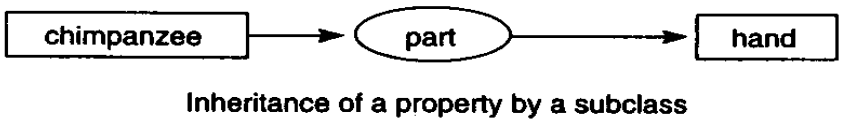


وراثت

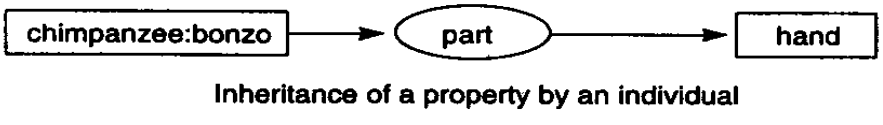
وراثت يک فرم از عموميت‌دادن **يا تعميم** است. عموميت‌دادن تضمين نمي‌كند كه گراف برآيند صحيح **باشد** حتي اگر گراف‌هاي اصلي صحيح باشند. **مثال:**



A conceptual graph



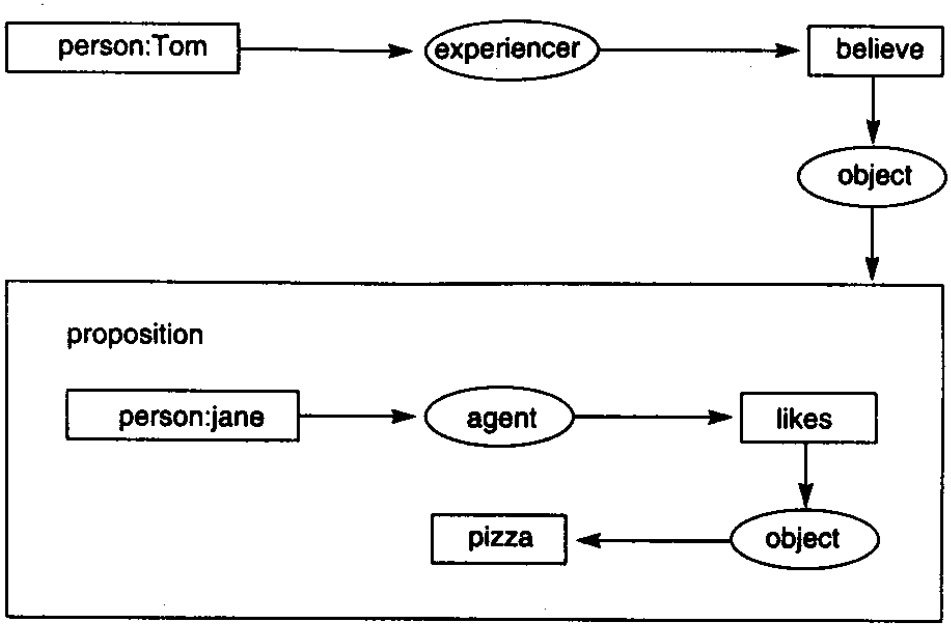
Inheritance of a property by a subclass



Inheritance of a property by an individual

گره‌هاي گزاره‌اي

براى مثال فعل "believes" يک گره گزاره‌اي مي‌گيرد:



گراف‌هاي مفهومي و منطق

گراف‌هاي مفهومي در قابليت نمايش دانش با جبر گزاره‌نما معادلند. **مثال:**

$$\forall X \forall Y (\neg (dog(X) \wedge color(X,Y) \wedge pink(Y))).$$

الگوريتمي براي تبديل يک گراف مفهومي به جبر گزاره‌نما **به شرح ذيل مي‌باشد**:

- هر متغيير يکتا ي $x_1,x_2,\ldots,x_n$ به هر يک از n مفهوم عمومي در g نسبت دهيد .
- يک ثابت يکتا به هر مفهوم فردي در g نسبت دهيد. اين مفهوم ممکن است به سادگي نام يا علامت‌دهنده‌اي که براي نمايش استنباط مفهومي استفاده شده است، باشد .
- هر نود**(گره)** مفهومي را به وسيله يک گزاره‌نماي يکتا با همين نام نشان دهيد.
- هر ارتباط مفهومي n تايي در g را به عنوان يک گزاره‌نماي n تايي که نام آن نام همين ارتباط است، نشان دهيد.
- اتصال همه جملات اتمي که تحت **دو مرحله قبل** تشکيل شده‌اند را برقرار کنيد. اين بدنه عبارت جبر گزاره‌نمايي است.

قوانین(قواعد)

به معنی جفت‌هاي شرط – عمل **می‌باشند.** اگر اين شرايط اتفاق بيافتد**آنگاه** بعضي از اعمال، اتفاق خواهد افتاد يا بايد بيافتد. **مثلاً** اگر چراغ توقف قرمز باشد و شما ايستاده باشيد آنگاه گردش به راست صحيح است. **قوانين می‌توانند به شکل‌های زير باشند:**

If condition **then** conclusion

Conclusion, **if** premise

قوانين دانش و استنتاج

انواع متداول قوانين **عبارتند از:**

- قوانين دانش يا قوانين اعلاني **که** همه وقايع و ارتباطات درباره يک مسئله را شرح مي‌دهند.
- قوانين استنتاج يا قوانين رويه‌اي که چگونگي حل يک مسأله را توصيه مي‌کنند
- قوانيني درباره قوانين (meta rules).
- قوانين دانش در پايگاه دانش ذخيره شده‌اند.
- قوانين استنتاج قسمتي از موتور استنتاج مي‌شوند.

مزاياي قوانين

- فهم ساده اي دارند (فرم طبيعي دانش).
- استنتاج و توضيح آن ها ساده است.
- تغيير و حفظ آنها ساده است.
- ترکيب آن ها با چيز ها ي نامعلوم ساده است.
- قوانين غالبا مستقل اند.

محدوديت هاي قوانين

- دانش پيچيده نيازمند قوانين زيادي است.
- سازندگان قوانين را دوست دارند. (hammer syndrome)
- در سيستم هاي با قوانين زياد، محدوديت هاي جستجو **وجود دارد.**

ویژگی‌های ارائه قوانین

Characteristics of Rule Representation

	First Part	Second Part
Names	Premise ⟶ Antecedent ⟶ Situation ⟶ IF ⟶	Conclusion Consequence Action THEN
Nature	Conditions, similar to declarative knowledge	Resolutions, similar to procedural knowledge
Size	Can have many IFs	Usually one conclusion
Statements	AND statements	All conditions must be true for a conclusion to be true
	OR statements	If any of the OR statement is true, the conclusion is true

فریم‌ها

یک فريم ساختمان داده‌اي است كه كليه دانش‌ها درباره يک شئ خاص را شامل مي شود. **اين** دانش در يک ساختار سلسله‌مراتبي سازمان يافته است **و** قالبي براي برنامه نويسي شئ‌گرا **فراهم مي‌آورد.** اصطلاحات فريم **در شكل زير نشان داده شده است:**

Frame Terminology

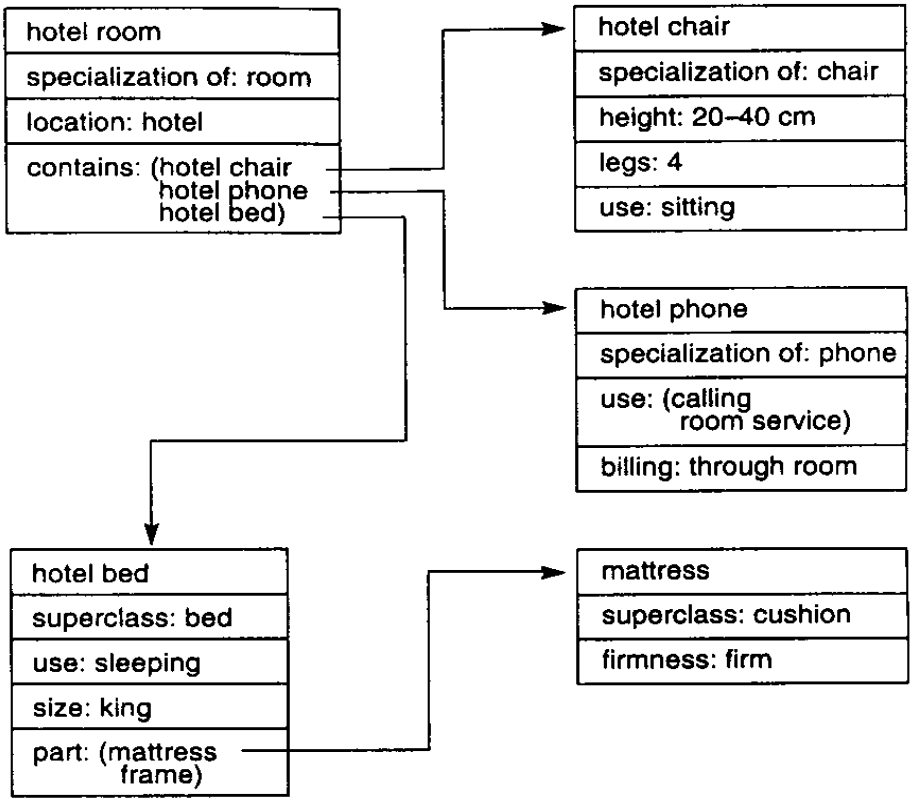
Default	Instantiation
Demon	Master frame
Facet	Object
Hierarchy of frames	Range
If added	Slot
If needed	Value (entry)
Instance of	

اجزاء يک فريم:

- اطلاعاتي كه فريم مشخص مي‌كند.
- ارتباط اين فريم با ساير فريم‌ها.

- توصيف‌كننده نيازمنديها براي هماهنگي فريم: به عنوان مثال يک صندلي ارتفاعي بين 20 تا 40 سانتي‌متر از كف دارد، پشت آن از 60 سانتي‌متر بلندتر است و ...
- اطلاعات رويه‌اي: فريمها توانايي اتصال كد رويه‌اي به Slot را دارند.
- اطلاعات پيش‌فرض فريم
- اطلاعات نمونه جديد: Slot هاي فريم ممكن است به صورت نامشخص جا گذاشته شوند تا وقتي كه مقاديري براي يک نمونه خاص داده شوند يا تا **زمانی که** آنها براي بعضي جنبه‌هاي حل مسأله لازم باشند.

مثال:



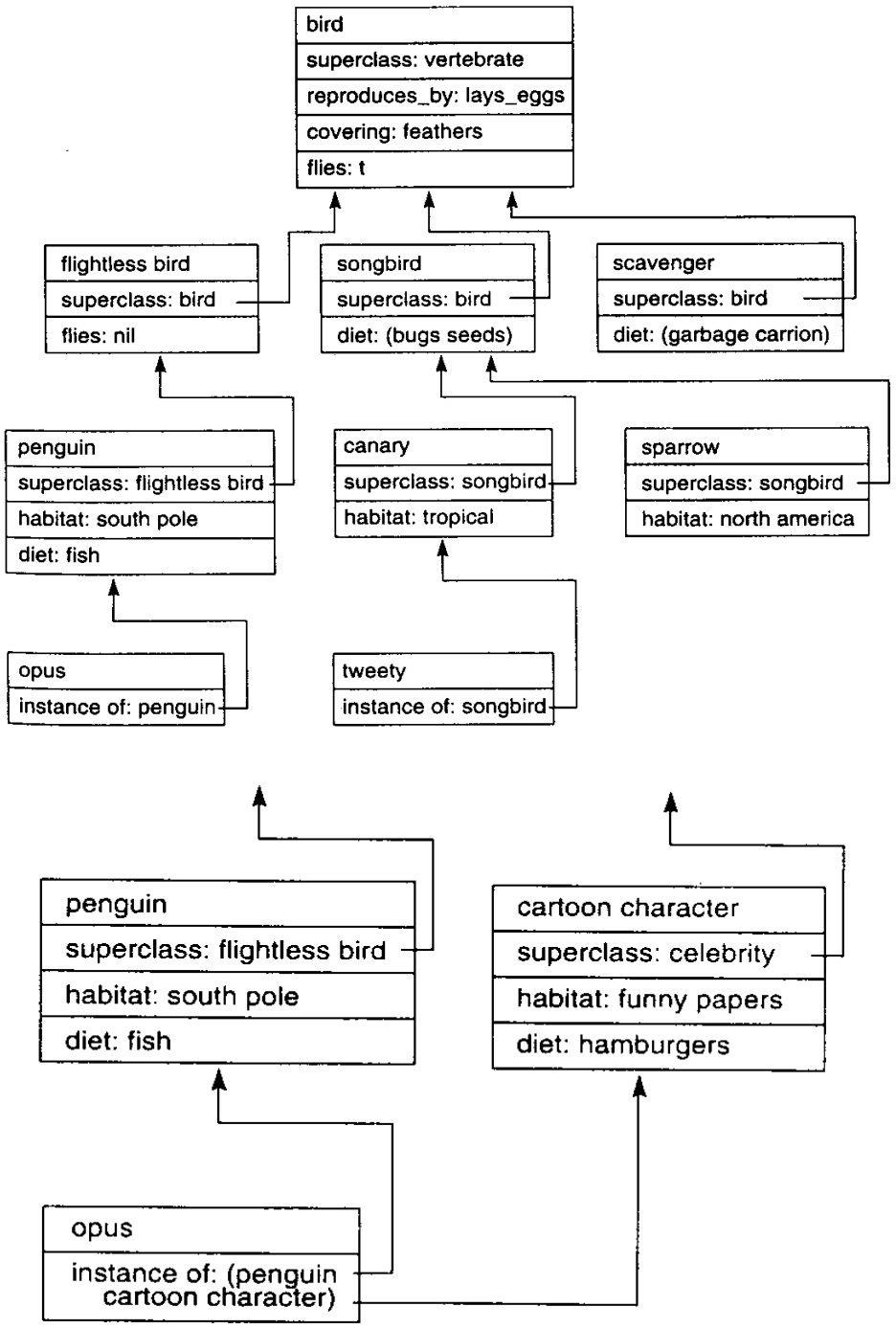
<u>قابليت هاي فريم:</u>

- توانايي آشكار كردن اطلاعات مستند درباره يک مدل دامنه‌اي. به عنوان يک مثال قطعات ماشين و صفات تخصيصي آنها.
- توانايي محدود كردن مقادير مجاز كه يک صفت مي تواند بگيرد.
- پيمانه‌اي بودن اطلاعات، تسهيل اجازه توسعه و نگهداري سيستم.
- مكانيزمي كه امكان محدود كردن حقايق در طول زنجيره «رو به عقب» يا «رو به

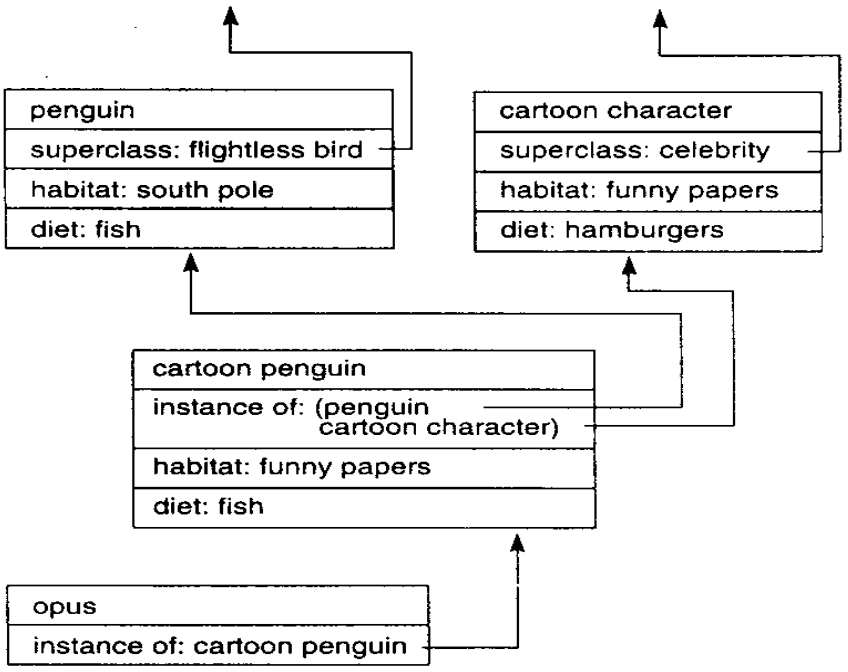
جلو» را میسر می‌کند.

- دستیابی به مکانیزمی که وراثت اطلاعات را در یک سلسله مراتب کلاس پشتیبانی می‌کند.

نمونه‌هایی از وراثت در شکل‌های زیر نشان داده شده است:



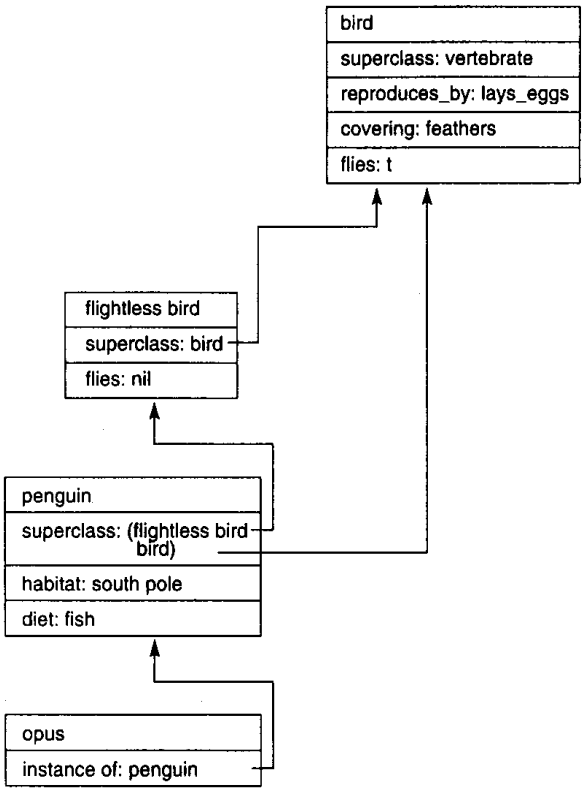
یک کلاس جدید براي حل مجدد ابهام:



انتقال‌پذیری زیرکلاس‌ها

این مسأله را در نظر بگیرید: پنگوئن ها پرواز نمي كنند؛ يک كلاس از پرنده اي كه پرواز نمي‌كند توليد مي‌شود.

نتايج در ساير زير مسئله ها: اگر زيركلاس‌ها انتقال‌پذير باشند ما استنباط مي‌كنيم كه يک پنگوئن، يک پرنده است. **في‌الواقع** يک لينک اضافي مسائلي با وراثت چندگانه را توليد مي‌كند:



به طور خلاصه می‌توان گفت که: فريم‌ها دانش را در داخل ساختارهايي سازمان مي‌دهند؛ رويه‌ها مي‌توانند به فريم‌ها متصل شوند؛ فريم‌ها وراثت كلاس را پشتيباني مي‌كنند؛ فريم‌ها مي‌توانند دانش پيش‌فرض را اعمال كنند؛ **و نهايتاً** فريم‌ها شبكه‌هاي معنايي را به وسيله مهيا كردن «سازمان و ساختار» توسعه مي‌دهند.

اسکریپت‌ها

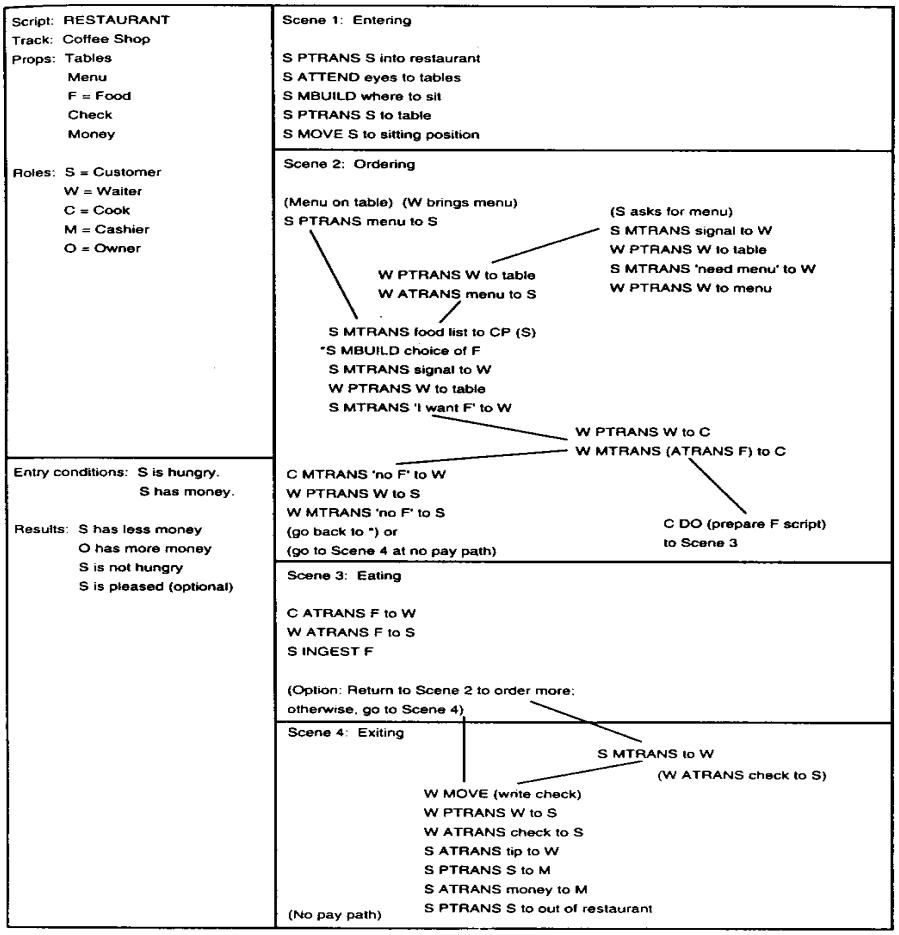
اسکریپت‌ها شامل عناصر **ذیل هستند**:

- شرايط ورود: اين شرايط بايد قبل از **رويدادهايی** كه در اسكريپت مي‌تواند اتفاق بيافتد

برآورده شده باشد.

- وسائل صحنه نمایش: نمایش اشیاء اسلات‌ها که شامل **رویدادها** می‌باشند.
- نقش‌ها: اشخاصي که در **رویدادها** هستند.
- دنبال‌کردن: تغییر در اسکریپت.
- صحنه نمایش: ترتیبي از **رویدادهایی** که ممکن است اتفاق بیافتند. **رویدادهای** در فریم‌هاي وابستگي مفهومي ارائه شده‌اند.

مثال: اسکریپت رستوران

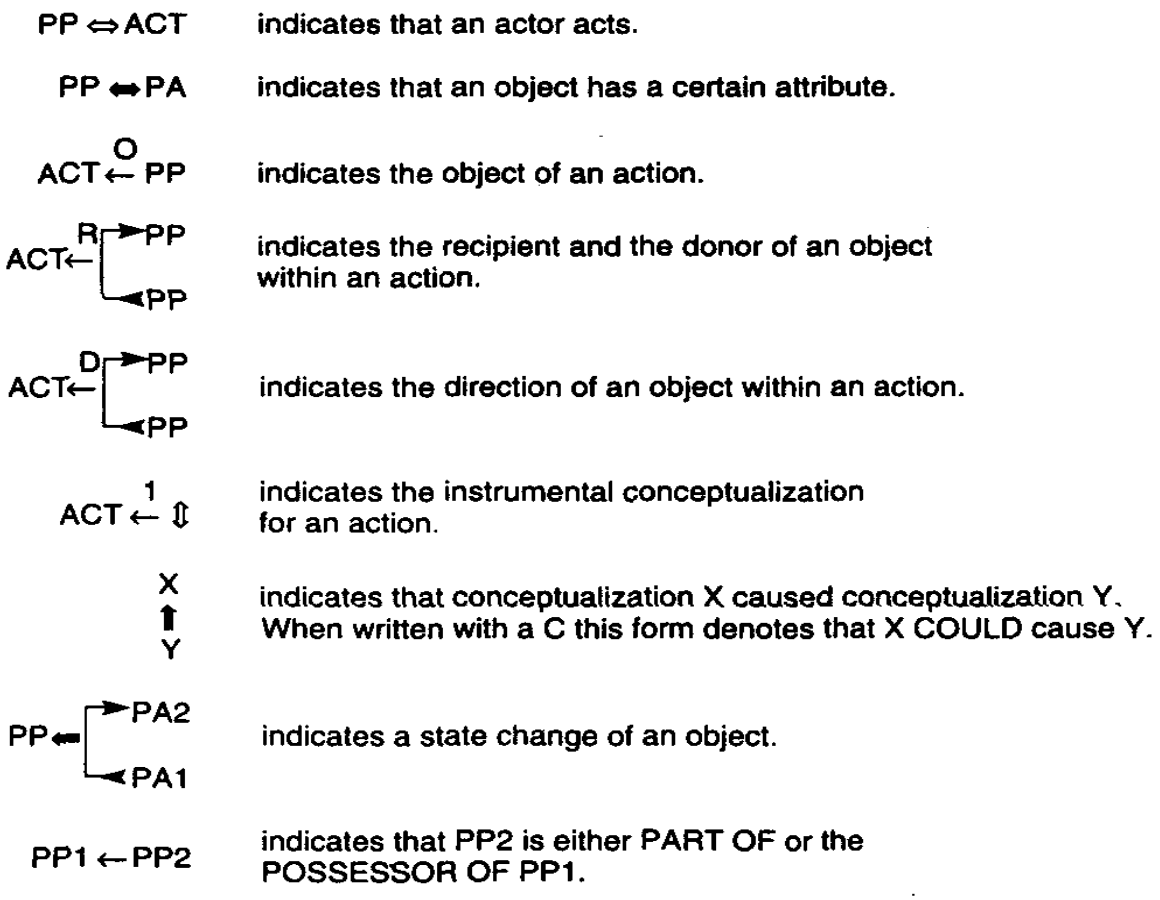


مثال: اسکریپت دزدی

Script: ROBBERY	*Track: Successful Snatch*
Props: G = Gun, L = Loot, B= Bag, C = Get away car.	*Roles*: R = Robber, M = Cashier, O = Bank Manager, P = Policeman.
Entry Conditions: R is poor. R is destitute.	*Results*: R has more money. O is angry. M is in a state of shock. P is shot.
Scene 1: Getting a gun R PTRANS R into Gun Shop R MBUILD R choice of G R MTRANS choice. R ATRANS buys G (go to scene 2)	
Scene 2 Holding up the bank R PTRANS R into bank R ATTEND eyes M, O and P R MOVE R to M position R GRASP G R MOVE G to point to M R MTRANS "Give me the money or ELSE" to M P MTRANS "Hold it Hands Up" to R R PROPEL shoots G P INGEST bullet from G M ATRANS L to M M ATRANS L puts in bag, B M PTRANS exit O ATRANS raises the alarm (go to scene 3)	
Scene 3: The getaway M PTRANS C	

مزایای **این روش** توانايي پيشگويي **رويدادها** و معايب **آن عبارت است از این که:** نسبت به فريم‌ها عموميت كمتري دارند **و** ممكن است براي ارائه همه انواع دانش‌ها مناسب نباشند.

ارتباطات اساسي وابستگي مفهومي **در شكل زير نمايش داده شده است:**



توضیحات:

←: مسیر وابستگي را نشان مي‌دهد.

⇔: ارتباط فعل عامل را نشان مي‌دهد.

P : زمان گذشته را نشان مي‌دهد.

INGEST :یک عمل اولیه از تئوري است.

O : ارتباط اشیاء.

D : مسیر اشیاء در فعالیت را نشان مي‌دهد.

F :آینده.

T : گذر.

K : در حال ادامه دادن.

Ts : شروع گذر.

? : پرسشي.

Tf : اتمام گذر.

C : شرطي.

/: منفي.

Nil : حاضر.

Delta? : نداشتن زمان.

اعمال اولیه در اسکریپت:

ATRANS : گذر یک ارتباط.(give)

PTRANS : گذر محل فيزيكي از يک شي.(go)

PROPEL : اعمال نيروي فيزيكي به يک شئ.(push)

MOVE : حرکت قسمتي از بدن به وسيله خودش.(kick)

GRASP : گرفتن يک شئ به وسيله يک اکتور.(grasp)

INGEST : قورت دادن يک شئ به وسيله يک حيوان.(eat)

EXPEL : بيرون انداختن از بدن يک حيوان.(cry)

MTRANS : گذر اطلاعات ذهني.(tell)

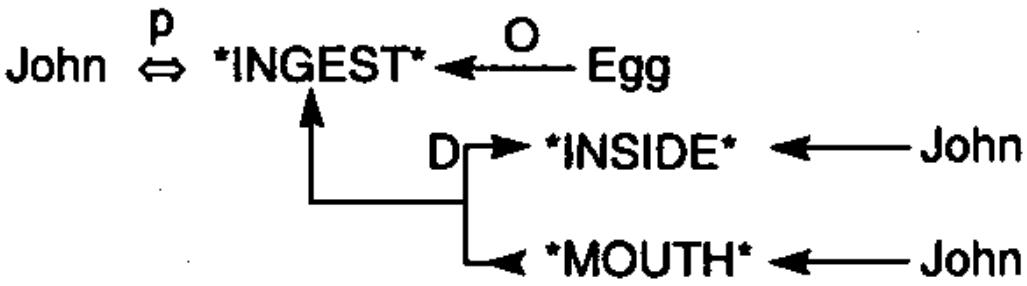
MBUILD : ساخت اطلاعات جديد به صورت ذهني.(decide)

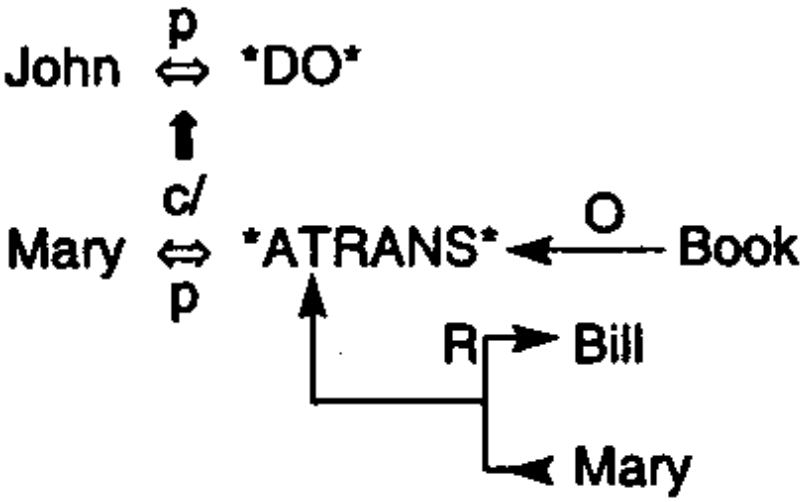
CONC : به صورت مفهوم درآوردن يا فکر کردن درباره يک نظر.(think)

SPEAK : توليد صدا.(say)

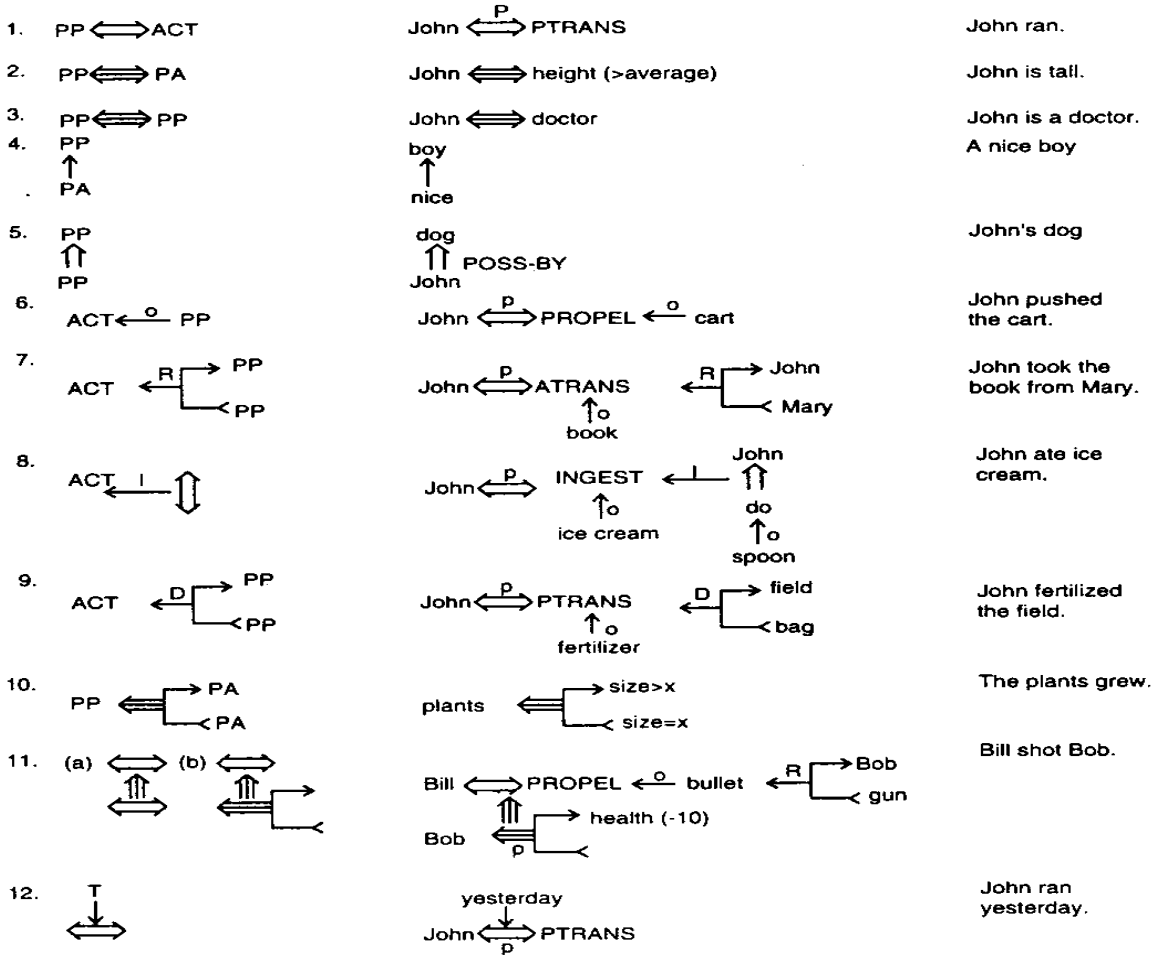
ATTEND : تمرکز دادن عضو حسي.(listen)

مثال:


John
p
⇔
INGEST
O
Egg
D
INSIDE
John
MOUTH
John


John
p
⇔
DO
c/
Mary
⇔
ATRANS
O
Book
p
R
Bill
Mary

مثال هاي بيشتر:



<u>ملاحظات براي ارزيابي يک بازنمايي دانش:</u>

- طبيعي بودن ،يکنواخت بودن و قابليت فهم.
- درجه‌گذاري براي اينکه کدام دانش صريح‌تر است يا اينکه در کدام رويه جاسازي شده است.
- پيمانه‌اي بودن و ثابت‌بودن يک پايگاه دانش.
- راندمان بازيابي دانش و قدرت اکتشافي رويه استنتاج کننده.

باید توجه داشت که هيچ روش باز نمايي دانش‌ی به تنهايي براي همه کارها مورد استفاده قرار نمي گيرد.

بازنمايي دانش چند گانه: هر چيزي براي يک زير وظيفه متفاوت مناسب است.

بازنمايي دانش چند گانه

از قوانين + فريم‌ها **تشکيل می‌شود.** بازنمايي دانش بايد **از کسب دانش، بازیابی دانش و استنتاج** پشتيباني کند**.**

3- یکی از روش‌های تولید هستان‌شناسی‌های مختلف که امروزه بیشتر مورد توجه است، ساخت خودکار هستان‌شناسی و استفاده از روش‌های یادگیری برای اکتساب دانش از منابع مختلف می‌باشد. روش‌های مختلف تولید خودکار هستان‌شناسی را به تفصیل شرح دهید.

ساختن دستی هستان‌شناسی کار سخت و طاقت‌فرسایی می‌باشد. این امر مستلزم دانش وسیع از حوزه مربوطه بوده و در اغلب حالات، نتیجه حاصل ناقص یا نادقیق خواهد بود. هستان‌شناسی‌هایی که به صورت دستی ساخته شده‌اند، گران، خسته‌کننده، خطاپذیر، و متمایل به نظر سازندگان آن (غیربیطرف) می‌باشند. همچنین غیرقابل انعطاف و خاص-منظوره هستند.

برای غلبه بر کاستی‌ها و نواقص ساختن دستی هستان‌شناسی از روش‌های نیمه‌خودکار یا خودکار برای ساختن آنها استفاده می‌شود. ساختن خودکار هستان‌شناسی نه تنها هزینه‌ها را کاهش می‌دهد، بلکه به هستان‌شناسی‌ای می‌انجامد که با نیازهای کاربردی سازگاری بیشتری دارد. روش‌ها و سیستم‌های گوناگونی برای یادگیری هستان‌شناسی ارائه شده است. این روش‌ها به دو طریق هستان‌شناسی را می‌سازند. یک روش، توسعه دادن ابزارهایی است که توسط مهندسان دانش یا متخصصین حوزه مربوطه بکار می‌روند مانند Protege و ontoEdit. روش دیگر ساختن نیمه‌خودکار یا خودکار هستان‌شناسی از طریق یادگیری آن از منابع دانش گوناگون می‌باشد.

یادگیری هستان‌شناسی به معنی استخراج عناصر هستان‌شناسی (دانش مفهومی) از ورودی و ساختن هستان‌شناسی از روی آنهاست. هدف ساختن نیمه‌خودکار یا خودکار هستان‌شناسی از پیکره متنی ورودی با دخالت محدود انسانی است. یادگیری هستان‌شناسی را می‌توان به صورت مجموعه‌ای از روش‌ها و تکنیک‌های بکار رفته برای ساختن هستان‌شناسی از صفر (from scratch)، غنی‌سازی (enriching) یا وفق‌دادن (adapting) یک هستان‌شناسی موجود، به روشی نیمه‌خودکار از روی منابع مختلف، تعریف نمود. یادگیری هستان‌شناسی از روش‌های متعددی از قبیل یادگیری ماشین، اکتساب دانش، پردازش زبان طبیعی، استخراج اطلاعات، هوش مصنوعی، استنتاج (reasoning) و مدیریت پایگاه داده استفاده می‌کند. سیستم‌های یادگیری هستان‌شناسی را می‌توان برحسب نوع داده‌ای که یاد می‌گیرند، طبقه‌بندی نمود. این انواع داده، ساخت‌نیافته، نیمه‌ساخت‌یافته یا ساخت‌یافته می‌باشند. داده‌های ساخت‌نیافته معادل متن طبیعی هستند مثل کتاب‌ها و مجلات. داده نیمه‌ساخت‌یافته متنی مثل فایل‌های HTML و XML است. داده ساخت‌یافته معادل پایگاه‌های داده و دیکشنری‌ها می‌باشد. در ادامه به یادگیری هستان‌شناسی از روی داده‌های ساخت‌نیافته و نیمه‌ساخت‌یافته می‌پردازیم.

یادگیری از روی داده ساخت‌نیافته

داده ساخت‌نیافته، سخت‌ترین نوع داده برای یادگیری می‌باشد. این کار مستلزم پردازش بیشتری نسبت به داده نیمه‌ساخت‌یافته است. سیستم‌هایی که برای یادگیری از روی متن آزاد ارائه شده‌اند، معمولاً به پردازشگرهای زبان طبیعی متکی هستند. برخی سیستم‌ها از پردازش متن کم‌عمق با تحلیل آماری و برخی دیگر از پارسرهای مبتنی بر قاعده برای تعیین روابط وابستگی بین کلمات موجود در زبان طبیعی استفاده می‌کنند. پردازش زبان طبیعی روشی مشترک بین همه

تکنیک‌ها می‌باشد، لذا راهکارهای مختلف بر اساس تکنیکی که علاوه بر پردازش زبان طبیعی بکار برده‌اند طبقه‌بندی می‌شوند.

راهکار آماری

یکی از راه‌ها، ساختن هستان‌شناسی از روی کلمات کلیدی‌ای است که به مفاهیم هستان‌شناسی مشابه و نزدیکند. در این روش کلمات کلیدی اولیه به موتور جستجو داده می‌شود تا صفحات مرتبط با آن بازیابی شوند. سپس این صفحات وب تحلیل می‌شوند تا مفاهیم کاندید مهم برای حوزه‌ای خاص پیدا شوند. این کلمه کلیدی برای یادگیری مفاهیم فرزند از صفحاتِ بدست آمده و از طریق استخراج بایگرم‌هایی که شامل این کلمه کلیدی بعنوان دومین عنصر هستند، استفاده می‌شود. مثلاً اگر کلمه کلیدی biosensor باشد و کلمه بلافاصله قبل از آن optical باشد (مثلاً optical biosensor) آنگاه گروه optical biosensor که مفهوم فرزند کاندید برای biosensor است اگر اندازه کمینه‌ای داشته باشد و یک stop word نباشد. انتخاب مفاهیم نمونه از روی مفاهیم کاندید بر حسب ویژگی‌های زیر صورت می‌گیرد:

- کل تعداد ظهور (appearance) در کلِ صفحات وبِ تحلیل شده.
- تعداد صفحات وبِ مختلفی که شامل این مفهوم هستند.
- تعداد تقریبی نتایجی که توسط موتور جستجو برای فقط کلمه قبلی برگردانده می‌شود (مثلاً optical).
- تعداد تقریبی نتایجی که توسط موتور جستجو برای ترکیب مفهوم انتخاب‌شده با کلمه کلیدی اولیه، برگردانده می‌شود.
- نسبت بین دو معیار قبلی.

فقط مفاهیم کاندیدی که ویژگی‌های آنها درون مجموعه‌ای از محدودیت‌های خاص می‌گنجد (که معادل بازه مقادیر برای هر پارامتر است) انتخاب می‌شوند. این سیستم از عباراتِ ریشه‌یابی‌شده استفاده می‌کند و در عین حال تعداد دفعات وقوع عبارات را برای بهبود کارایی‌اش در فرایند یافتن مفاهیم، می‌شمارد. این مفاهیم یافته‌شده بعنوان کلمات کلیدی جدید استفاده می‌شوند و سیستم دوباره اجرا می‌شود تا مفاهیم فرزند یافت شود. این فرایند به طور بازگشتی آنقدر تکرار می‌شود تا با یک عمق خاص برسیم یا نتیجه جدیدی یافتن نشود. نتیجه بدست آمده، سلسله‌مراتبی است که به صورت یک هستان‌شناسی ذخیره می‌شود.

راهکار پردازش زبان طبیعی

یک روش استفاده از الگوهای نحوی برای یافتن روابط وابستگی بین کلمات است. در این روش از قاعده‌های نحوی که در طبیعت زیرزبانی (sublanguage) مستندات وب‌سرویس مستتر هستند استفاده می‌شود. این طبیعت زیرزبانی، نوع خاصی از زبان طبیعی می‌باشد. مراحل استخراج هستان‌شناسی عبارتند از: پارس وابستگی، الگوهای نحوی، ساختن هستان‌شناسی و هرس هستان‌شناسی. از یک پارسر وابستگی برای تشخیص روابط وابستگی بین کلمات در زبان طبیعی استفاده می‌شود. رابطه وابستگی یک رابطه دودویی نامتقارن است بین یک کلمه که هسته (head) نامیده می‌شود و یک کلمه که تغییردهنده یا وابسته (modifier) نامیده می‌شود. مثلاً در جمله

"find antigenic sites in proteins" کلمه "antigenic" صفتی است که اسم "site" را تغییر می‌دهد و "sites" مفعول فعل "find" است. سپس مجموعه‌ای الگوهای نحوی بکار می‌رود تا اطلاعات مفید به منظور ساختن هستان‌شناسی از روی پیکره برچسب‌خورده مشخص و استخراج شوند.

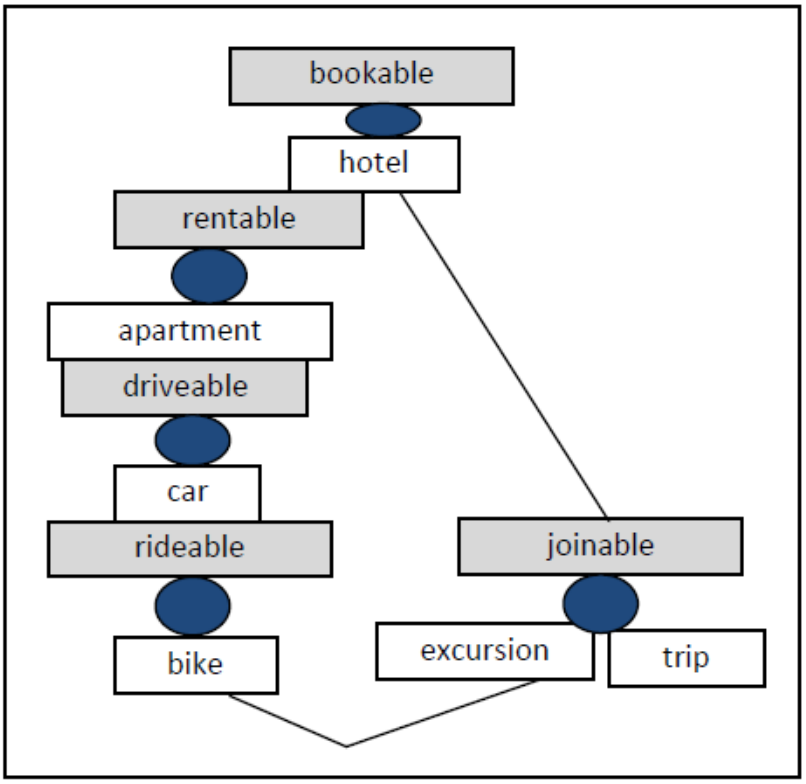
سه گروه/طبقه عمده برای الگوهایی تعریف می‌شود که برای استخراج انواع مختلف اطلاعات بکار می‌روند. اولین گروه برای تشخیص مفاهیم حوزه‌ای بکار می‌روند. در این حالت الگوهای اسم و گروه اسمی ("NN", "NMod") برای یافتن مفاهیم و روابط وابستگی بین آنها بکار می‌رود (مثل <site>, <antigenic site>). گروه دوم برای تشخیص عملکردهایی بکار می‌رود که غالباً در آن حوزه رخ می‌دهد. از افعال و اسامی بسیار مرتبط با این افعال برای شناسایی عملکرد یک رویه استفاده می‌شود (مثلاً <find>, <antigenic site>). گروه آخر برای یافتن روابط از طریق گروه حرف اضافه‌ای (PP) بکار می‌رود تا یک رابطه جزء به کل (meronymy) بین عبارات را بیابد (مثلاً در جمله قبل "in protein" یک PP است و لذا <antigenic sites> جزئی از <protein> هستند).

روش دیگر استفاده از یک راهکار خودکار برای ساختن رده‌بندی‌ها یا سلسله‌مراتب مفاهیم از روی پیکری متنی می‌باشد. در این روش از تحلیل صوری مفهوم (Formal Concept Analysis) استفاده می‌شود که روابط ذاتی و ماهوی بین اشیا را از طریق مجموعه‌ای از ویژگی‌ها و خود ویژگی‌ها استخراج می‌کند.

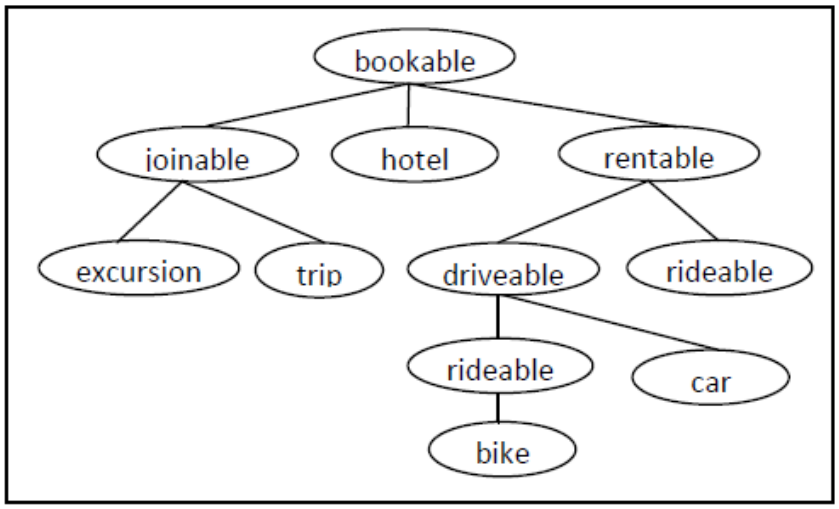
ابتدا پیکره پارس می‌شود تا کلمات آن برحسب POS متناظر برچسب‌گذاری شوند و درخت‌های پارس برای هر جمله بدست آید. وابستگی‌های فعل/فاعل، فعل/مفعول و فعل/گروه حرف اضافه از روی این درخت‌های پارس استخراج می‌شوند. سپس فعل و هسته‌ها lemmatize می‌شوند. از آنجا که فرض کامل‌بودن اطلاعات هیچ وقت محقق نمی‌شود، مجموعهٔ زوج‌ها هموار

(smooth) می‌شوند. هموارسازی با خوشه‌بندی همه عباراتی که دوبدو مشابه هستند برحسب معیار مشابهت مورد نظر، انجام می‌شود. شمارش زوج ویژگی/شیء‌های بیشتر از آنچه که واقعاً در متن موجود است به فرکانس غیرصفری برای برخی زوج ویژگی/شیء‌ها می‌انجامد که عملاً در پیکره ظاهر نشده‌اند. لذا نتیجه کلی، «هموارسازی» فرکانس نسبی از طریق نسبت دادن برخی فرکانس‌های نسبی غیرصفر به ترکیب‌های افعال و مفعول‌هایی که واقعاً در پیکره نیستند، می‌باشد. مثلاً «ماشین» و «دوچرخه» دوبدو مشابه‌اند، و متعاقباً زوج‌هایی که هرکدام از اینها را با ویژگی‌های فعل خود داشته باشند، با هم خوشه‌بندی می‌شوند. زوج‌های شیء/ویژگی با استفاده از احتمال شرطی، اطلاعات دوبدوی point wise و آنتروپی نسبی توزیع prior و posterior مجموعه‌ای از زوج‌ها وزن‌دهی می‌شوند تا «قدرت انتخابی» (selectional strength) فعل در یک مکان وابسته مفروض تعیین شود.

فقط افعالی که بیشتر از یک آستانه خاص باشند به یک بافت صوری (formal context) تبدیل می‌شوند که انوقت تحلیل صوری مفهوم روی آنها اِعمال شده تا هستان‌شناسی به شکل یک مشبک تولید شود (شکل زیر). تحلیل صوری مفهوم روشی است که مبتنی بر نظریه ترتیب (order theory) بوده و برای تحلیل داده‌ها مخصوصاً برای یافتن روابط ذاتی بین اشیاء بکار می‌رود. این اشیاء از یک طرف توسط مجموعه‌ای از ویژگی‌ها و از طرف دیگر توسط خود ویژگی‌ها، توصیف شده‌اند. سپس نتیجه حاصل از شکل مشبک به شکل ترتیب جزئی تبدیل می‌شود که به سلسله مراتب مفاهیم نزدیک‌تر است.



مشبک مفاهیم صوری برای مثال توریسم



سلسله‌مراتب متناظر مفاهیم هستان‌شناسی برای مثال توریسم

راهکار یکپارچه (integrated)

Text2Onto به کاربرانش این امکان را می‌دهد تا یک الگوریتم یادگیری مناسب برای نوع هستان‌شناسی مورد نظرشان انتخاب کنند. ابتدا پیکره پارس می‌شود تا کلماتش برچسب POS خورده و ریشه‌یابی شوند. Text2Onto کتابخانه‌ای از الگوریتم‌ها برای یادگیری عناصر مختلف هستان‌شناسی دارد. این عناصر عبارتند از: مفاهیم، وراثت مفاهیم، نمونه‌های مفاهیم، روابط کلی، روابط جزء به کل (metrological) و هم‌ارزی.

الگوریتم‌های یادگیری مفاهیم موجود در این راهکار بر این فرض بنا شده‌اند که یک عبارت با فرکانس بالا در مجموعه‌ای از متن حوزه‌ای خاص، نشانگر وقوع یک مفهوم مرتبط است. لذا این الگوریتم‌ها مفاهیم را از طریق فرکانس نسبی عبارت (Relative Term Frequency=RTF)، TFIDF(Term Frequency Inverted Document Frequnect)، آنتروپی و روش C-value/NC-value یاد می‌گیرند. برای استخراج روابط وراثت مفاهیم، Text2Onto الگوریتم‌های گوناگونی را بسته به استفاده از ساختار شمول معنایی WordNet، تطبیق الگوهای Hearst و اِعمال قواعد هیوریستیک زبان‌شناختی، پیاده‌سازی کرده است. برای یادگیری روابط کلی، Text2Onto از یک استراتژی پارس کم‌عمق استفاده می‌کند تا فریم‌های زیرمقوله‌ایِ غنی‌سازی شده با اطلاعاتِ دربارهٔ فرکانس عباراتی که به صورت وابسته ظاهر می‌شوند را استخراج کند. این برنامه فریم‌های نحوی مثل love(subj, obj) را استخراج می‌کند و این فریم‌های زیرمقوله‌ای را به روابط هستان‌شناسی نگاشت می‌کند. روابط جزء به کل با استفاده از تکنیک‌های تطبیق الگو یاد گرفته می‌شوند. یادگیری روابط نمونه‌های مفاهیم (concept instance) متکی بر یک روشِ مبتنی بر مشابهت است که بردارهای بافت (context vector) نمونه‌ها و مفاهیم را از روی مجموعه متون استخراج کرده و بر اساس برداری که بیشترین مشابهت را دارد، نمونه‌ها را به مفاهیم نسبت می‌دهد. روابط هم‌ارزی طبق این فرض یاد گرفته می‌شوند که مفاهیم، به همان اندازه که بافت‌های نحوی مشابه مشترکی دارند، هم‌ارز می‌باشند. بعد از این که پردازش استخراج هستان‌شناسی تمام شد، به کاربر ارائه می‌شود تا اصلاحات آن را انجام دهد. نهایتاً کاربر می‌تواند از بین روش‌هایی که برای ترجمه هستان‌شناسیِ یادگرفته شده به زبان‌های مختلفِ نمایش هستان‌شناسی بکار می‌روند، یکی را انتخاب کند.

یادگیری از روی داده‌های نیمه‌ساخت‌یافته

ساختن هستان‌شناسی از روی داده‌های نیمه‌ساخت‌یافته از روش‌های داده‌کاوی (data mining) و واکاوی محتوای وب (web content mining) استفاده می‌کند. می‌توان از ساختار صفحات وب برای ساختن جدولی از پایگاه داده استفاده نمود و سپس از روش‌های خوشه‌بندی برای ساختن هستان‌شناسی متناظر آنها بهره برد. از ساختار فایل‌های HTML همراه با دانش زبان‌شناسی می‌توان به عنوان ویژگی‌هایی استفاده کرد که مفاهیم کاندید مشخص شوند. در روشی دیگر می‌توان صفحات HTML را به ساختارهای سلسله‌مراتبی معنایی مانند XML

تبدیل کرد تا از واکاوی آن، رده‌بندی (taxonomoy) متناظر استخراج گردد. در روشی دیگر می‌توان هستان‌شناسی را با استفاده از دو راهکار مکمل ایجاد نمود. راهکار اول از ساختار عباراتی که در قسمت heading مستندات HTML وجود دارد استفاده می‌کند و راهکار دوم از ساختار سلسله‌مراتبی قسمت heading فایل‌های html برای تشخیص مفاهیم جدید و روابط رده‌شناختی آنها بین مفاهیم seed و بین یکدیگر استفاده می‌کند.

راهکار داده‌کاوی

از روش‌های خوشه‌بندی برای گروه‌بندی کلمات مشابه به منظور تعریف یک سلسله‌مراتب مفاهیم استفاده می‌شود. ابتدا متن و ساختار صفحات HTML بررسی می‌شوند تا از روی آنها مفاهیم ساخته شوند. قسمت‌های و تَگ‌های عنوان (title)، زیرعنوان (sub title)، bold، italic، underlined، big character، کلمات کلیدی، hyperlinkها، لیست‌ها و پاراگراف‌ها و کل متن، پردازش می‌شوند. سپس جدول داده‌ای ساخته می‌شود که فیلدهای آن شامل کلمات، کلمات برچسب‌خورده (محتوایی که کلمه به آن متعلق است)، سبک کلمه (یعنی title یا bold یا غیره)، عددی که نشانگر تعداد دفعات ظهور آن کلمه در آن سبک در HTML و تعداد مستنداتی است که آن کلمه در آن قرار دارد. سپس کلماتی که به یک معنی اشاره می‌کنند از طریق تعامل کاربر گروه‌بندی می‌شوند. سپس از روش‌های بی‌سرپرست که نوعی خوشه‌بندی تقسیم‌کننده (divisive) است، برای ساختن سلسله‌مراتب خوشه‌های مفاهیم استفاده می‌شود.

در روشی دیگر می‌توان صفحات وب را به داده‌های ساخت‌یافته‌ای که توسط یک جدول (پایگاه داده) رابطه‌ای نمایش داده می‌شوند، تبدیل نمود. سپس این نمایش رابطه‌ای با توصیف ویژگی‌های ساختاری و زبان‌شناختی غنی‌سازی می‌شود تا بافت یک عبارت و همسایگی آن (vicinity) به طور دقیق تعیین شود. صفحات وب مورد پردازش قرار می‌گیرند تا فقط متنی که متناظر مجموعه‌ای از markupهاست (مثل <h1>, <b>, <i>, <li>) باقی بماند چرا که اینها برای استخراج مهمترین عبارات، در درجه اول اهمیت قرار دارند. برای تأکید روی عبارات مهم از تَگ <TITLE_URL> برای هایپرلینک، <CHOICE> برای checkbox و <KEYWORDS> برای همه عناصر metadata درون مستند استفاده می‌شود. خروجی این مرحله در جدول پایگاه داده قرار می‌گیرد. ویژگی‌های جدول عبارتند از عبارت، markup آن (تَگ متناظر)، و تَگ مرتبط قبلی (مثلاً <h1> یک تَگ قبلی برای <h2> است) و رتبه‌بندی آن در مستند منبع (مثلا درجه اهمیت 1 برای تَگ <title> و <h1> و درجه اهمیت 2 برای <li>) در این قسمت پر می‌شوند. سپس از سه نوع تحلیل برای ارزیابی و توصیف ویژگی‌های ساختاری، طبیعی و زبان‌شناختی پیکره استفاده می‌شود. تحلیل ساختاری، ویژگی‌های ساختاری

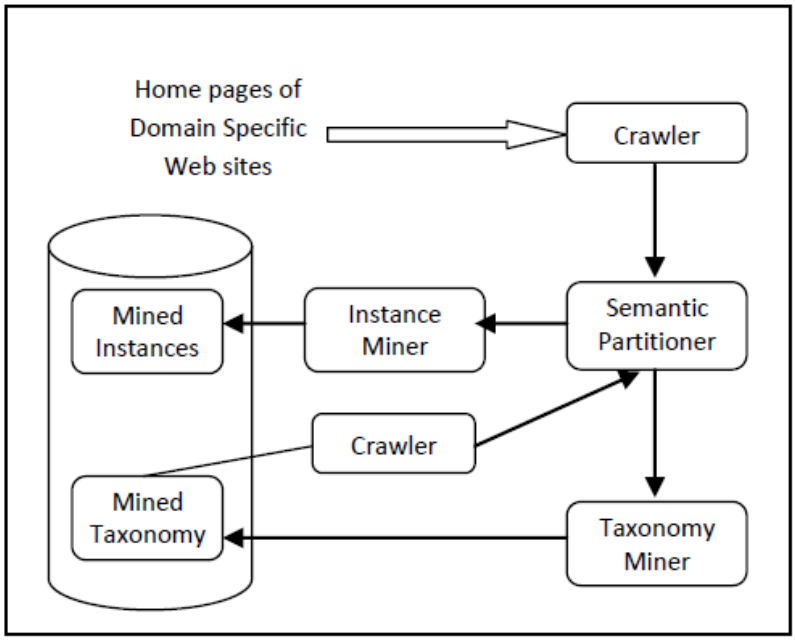
پیکره مورد نظر را از طریق محاسبه فرکانس markup برای هر مقوله markup و درصد عبارت متناظر با آن ارزیابی می‌کند. همچنین الگوهای ساختاری را برای markupهایی که با هم ظاهر می‌شوند استخراج می‌کند (<h1>-><p>). این الگوهای ساختاری به کاربر این امکان را می‌دهند تا تعریف بافت عبارت را با محدودکردن همسایگی آن، ویرایش کند.

تحلیل طبیعی (nature analysis)، پیکره صفحات HTML را آنقدر با حذف یا اضافه کردن مستندات HTML تغییر می‌دهد تا به یک همگونی در حوزه مورد نظر برسد. تحلیل و توصیف زبان‌شناختی، ریشه و مقوله نحوی (فعل، اسم، صفت، قید، ...) عبارت را مشخص می‌سازد. مثلاً از ابزار TreeTagger برای تعیین مقوله نحوی و ریشه عبارات درون پیکره می‌شود استفاده کرد. این اطلاعات، پایگاه داده رابطه‌ای را با پرکردن ویژگی‌های مرتبط با خصوصیات زبان‌شناختی تقویت می‌کند. همچنین می‌شود الگوهایی را استخراج نمود که برای پالایش تعریف بافت عبارت و رابطه معنایی آن بکار می‌روند.

برای خوشه‌بندی، از معیار مشابهت یا فاصله برای محاسبه مشابهت یا فاصله دوبدو بین بردارهای متناظر دو عبارت استفاده می‌شود تا مشخص شود که می‌توانند در یک خوشه باشند یا نه. کاربر می‌تواند نتایج بدست آمده از طریق اِعمال معیارهای مشابهت مختلف (مانند کسینوسی، اقلیدسی، فاصله، jaccard) را با هم مقایسه نماید. می‌توان هم-وقوعی در یک بافت ساختاری (با استفاده از الگوهای ساختاری) و هم-وقوعی در یک بافت نحوی (با استفاده از الگوهای نحوی) را با هم ترکیب نمود تا اهمیت یک زوج عبارت را سنجید. اگر دو عبارت در یک تَگ هم‌سطح (<h1></h1>) واقع شوند، بافت، توسط تَگ، محدود شده است و هم-وقوعی در این بافت محاسبه می‌شود. اگر دو عبارت در دو تَگ مختلف واقع شده‌اند که از لحاظ ساختاری مرتبط می‌باشند (<h1>, <p>)، آنگاه هم-وقوعی آنها از روی این لینک در این بافت محاسبه می‌شود. خوشه اولیه سلسله‌مراتب از روی تَگ‌های کلمات کلیدی که متناظر مهمترین عبارت می‌باشند بدست می‌آید. سپس خوشه‌های برگ با در نظر گرفتن عبارات هم-وقوع در بافت‌های ساختاری و نحوی، پالایش می‌شوند. سپس درختی ساخته می‌شود که نمایانگر سلسله‌مراتب markup است و به رویه خوشه‌بندی کمک می‌کند تا بطور بازگشتی دو عبارت را متعلق به سطح سلسله‌مراتب مورد نظر، در نظر بگیرد. این خوشه‌بندی بازگشتی به کاربر این امکان را می‌دهد تا خوشه را در هر مرحله ارزیابی کند. بعد از هر مرحلهٔ تکرار، کاربر خوشه‌ها را سنجیده و ارزیابی می‌کند.

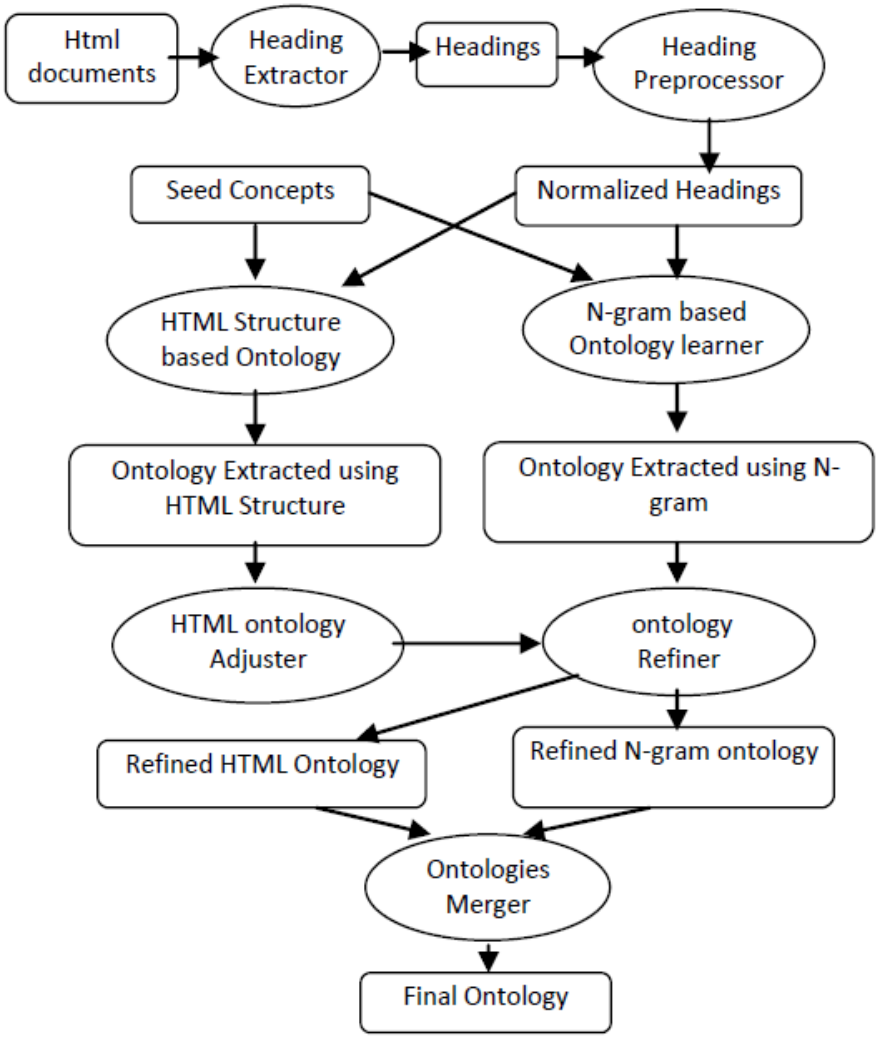
<u>راهکار واکاوی محتوای وب</u>

OntoMiner ابزاری است که با یادگیری از صفحات html، رده‌بندی را ایجاد می‌کند. OntoMiner یک تکنیک خودکار برای bootstrapping و جمع‌آوری هستان‌شناسی‌های خاص-حوزه از طریق سازمان‌دهی و واکاوی مجموعه‌ای از سایت‌های وبِ مرتبط، همپوشان، خاص-حوزه و مبتنی بر رده‌بندی می‌باشد. این سایت‌ها توسط کاربر فراهم می‌شوند و حوزه علائق او را مشخص می‌سازند. یک وب‌سایت مبتنی بر رده‌بندی (taxonomy-directed)، وب‌سایتی است که حداقل یک رده‌بندی برای سازمان‌دهی محتوایش دارد و موارد متعلق به یک مفهوم را با روشی استاندارد نگهداری می‌کند (مثلاً علمی، خبری، تفریحی، مسافرتی). آنگونه که در شکل زیر نشان داده شده است، صفحات وب خزیده (crawl) می‌شوند و به پیمانه معنی‌شناسی داده می‌شوند که صفحهٔ وب را به قسمت‌های منطقی تقسیم می‌کند و درخت DOM(=Document Object Model) را می‌سازد. نهایتاً از قواعد ارتقاء (promotion) که مبتنی بر نمایش و فرم صفحات وب هستند، برای ارتقاء برچسب‌های مورد تأکید (مثلاً گروهی از کلمات که در heading یا در bullet ظاهر می‌شوند) استفاده می‌کند؛ (همراه با تَگ‌هایی مثل <h1> ,<u> ,<b> در بالاترین قسمت گروه‌هایی خاص بعنوان گره والد xml). پیمانه واکاویِ رده‌بندی، ابتدا برچسب‌های با فرکانس بالا در مستندات xml را واکاوی می‌کند. برچسب‌هایی که فرکانسی بیشتر از حد آستانه دارند بعنوان برچسب‌های مهم از بقیه مستند جدا می‌شوند (مثلاً اقتصادی، ورزشی، سیاسی، فناوری، سلامتی و تفریحی بعنوان مفاهیم مهم در حوزه خبری). برای برچسب‌های نادیده گرفته شده که مرتبط ولی کم‌فرکانس هستند، مسیرهای تَگِ برچسب‌های پرفرکانس یاد گرفته می‌شوند و سپس روی بخش‌های منطقی متناظر، اِعمال می‌شوند تا برچسب‌های بیشتری بدست آیند. مثلاً «تفریحی» یک برچسب پرفرکانس است و مسیر تَگِ مشابهی با «فرهنگی» که برچسب کم‌فرکانسی است دارد. اگر برچسبی hyperlink نداشته باشد، نادیده گرفته می‌شود. این برچسب‌های مهم ریشه‌یابی می‌شوند و به گروه‌های برچسب‌های معادل سازماندهی می‌شوند (مثلاً «ورزش» و «ورزشی»). هر مجموعه از برچسب‌ها بعنوان یک مفهوم در نظر گرفته می‌شود. یک مفاهیم، مسطح (flat) هستند. سازماندهی این مفاهیم به شکل رده‌بندی نیازمند واکاوی روابط is-a از صفحات وبی است که از لحاظ معنی‌شناختی قسمت شده‌اند. برای گسترش رده‌بندی حوزه، hyperlinkهای متناظر با هر مفهوم دنبال می‌شوند. مثلاً «ورزش» یک مفهوم است و صفحاتی که با کلمات متناظر با مفهوم «ورزش» مرتبط شده‌اند، به منظور ساختن زیررده‌بندی «ورزش» و گسترش عمقی رده‌بندی بکار می‌روند. نهایتاً نمونه‌های مفهوم و مقادیر ویژگی‌های نمونه به همان طریق واکاوی زیررده‌بندی، واکاوی می‌شوند.



معماری OntoMiner

در روشی دیگر، هم ساختار عباراتی که در مستندات HTML ظاهر می‌شوند و هم ساختار سلسله‌مراتبی headingهای html به منظور یافتن مفاهیم جدید و روابط رده‌شناختی بین مفاهیم seed و یکدیگر، بکار می‌روند. معماری چنین سیستمی در شکل زیر آمده است. ابتدا استخراج‌کنندهٔ heading، قسمت‌های heading مستندات HTML ورودی را به منظور استخراج مفاهیم، استخراج می‌کند. headingهای بدست آمده توسط پیش‌پردازنده heading نرمال‌سازی می‌شوند. این قسمت، متن heading را با حذف اعداد یا stop-wordهای موجود در درون آنها و ریشه‌یابی، نرمال‌سازی می‌کند. اولین راهکار یادگیری، مبتنی بر N-gram است. مفاهیم و روابط رده‌شناختی آنها با استفاده از دنباله کلمات (عبارات N-gram) در heading متن‌ها استخراج می‌شوند. فرزندان مفاهیم seed در متن heading با استخراج همه عبارات ممکنی (کلمات n-gram) که یکی از مفاهیم seed را بعنوان headword خود دارند، یافته می‌شوند. مثلاً اگر مفهوم seed، "disease" باشد، و عنوان heading برابر "powdery mildew disease" باشد، آنگاه یادگیرنده n-gram باید عبارت "powdery mildew disease" را به همراه عبارت "powdery mildew" بعنوان عبارات کاندید در نظر بگیرد.



فرایند یادگیری هستان‌شناسی

هستان‌شناسی بدست آمده ممکن است شامل مفاهیم جعلی باشد، به همین دلیل از مجموعه‌ای فیلترها استفاده می‌شود که مفاهیم نویزی یا جعلی را حذف کنند. برخی اوقات، مفاهیم seed بعنوان headword مفاهیم فرزند خود، عمل نمی‌کنند. لذا از ساختار heading مستندات ورودیِ وب برای یادگیری هستان‌شناسی از طریق راهکاری دیگر استفاده می‌شود. در این راهکار، ساختار مستند HTML (سطوح heading) برای یادگیری هستان‌شناسی رده‌شناختی بکار می‌رود. مفاهیم seed در heading های بالاترین سطح مستند یافته می‌شوند و مفاهیم موجود در سطح دوم به عنوان فرزندان مفاهیم سطح اول در نظر گرفته می‌شوند و مفاهیم موجود در سطح سوم بعنوان فرزندان سطح دوم و الخ. از HTML Ontology Refiner برای گسترش هستان‌شناسیِ بدست آمده از طریق این راهکار، استفاده می‌شود. در این روش مفاهیم جدیدی که روابط برادر-خواهری با مفاهیمِ قبلاً یادگرفته شده دارند، یافت می‌شوند. ادغام هستان‌شناسی‌های ساخته شده از طریق Ontology Merger انجام می‌شود. این پیمانه،

یادگیرندهٔ هستان‌شناسیِ مبتنی بر N-gram را با یادگیرندهٔ هستان‌شناسیِ مبتنی بر ساختار HTML، ادغام می‌کند.

4- در الگوریتم‌های تکاملی مختلف، روش‌های گوناگونی برای پیاده‌سازی اپراتورهای الگوریتم (ترکیب، جهش، انتخاب والدها، انتخاب نسل بعد) به کار می‌رود. انواع روش‌های پیاده‌سازی اپراتورها را در الگوریتم‌های تکاملی مختلف بنویسید. نمایش کروموزوم‌ها می‌تواند باینری، عدد صحیح، عدد حقیقی و ... باشد.

انتخاب

در مرحله انتخاب، یک جفت از کروموزوم‌ها برگزیده می‌شوند تا با هم ترکیب شوند. عملگر انتخاب رابط بین دو نسل است و بعضی از اعضای نسل کنونی را به نسل آینده منتقل می‌کند. بعد از انتخاب، عملگرهای ژنتیک روی دو عضو برگزیده اعمال می‌شوند. معیار در انتخاب اعضا، ارزش تطابق آنها می‌باشد، اما روند انتخاب حالتی تصادفی دارد.

شاید انتخاب مستقیم و ترتیبی به این شکل که بهترین اعضا دو به دو انتخاب شوند در نگاه اول روش مناسبی به نظر برسد اما باید به نکته‌ای توجه داشت. در الگوریتم ژنتیک ما با ژن‌ها روبرو هستیم. یک عضو با تطابق پایین اگر چه در نسل خودش عضو مناسبی نمی‌باشد اما ممکن است شامل ژن‌هایی خوب باشد و اگر شانس انتخاب شدنش صفر باشد، این ژن‌های خوب نمی‌توانند به نسل‌های بعد منتقل شوند. پس روش انتخاب باید به گونه‌ای باشد که به این عضو نیز شانس انتخاب شدن بدهد. راه حل مناسب، طراحی روش انتخاب به گونه‌ای است که احتمال انتخاب شدن اعضای با تطابق بالاتر بیشتر باشد. انتخاب باید به گونه‌ای صورت گیرد که تا جایی که ممکن است هر نسل جدید نسبت به نسل قبلی‌اش تطابق میانگین بهتری داشته باشد. روش‌های متداول انتخاب عبارتند از:

- انتخاب چرخ رولت

- انتخاب ترتیبی
- انتخاب بولتزمن
- انتخاب حالت پایدار
- نخبه‌سالاری
- انتخاب رقابتی
- انتخاب قطع سر
- انتخاب قطعی بریندل
- انتخاب جایگزینی نسلی اصلاح شده
- انتخاب مسابقه
- انتخاب مسابقه تصادفی

انتخاب چرخ رولت

انتخاب چرخ رولت که اولین بار توسط هالند پیشنهاد شد یکی از مناسب‌ترین انتخاب‌های تصادفی بوده که ایده آن، احتمال انتخاب می‌باشد. احتمال انتخاب متناظر با هر کروموزوم، بر اساس برازندگی آن محاسبه شده که اگر f_k مقدار برازندگی کروموزوم kام باشد، احتمال بقای متناظر با آن کروموزوم عبارت است از:

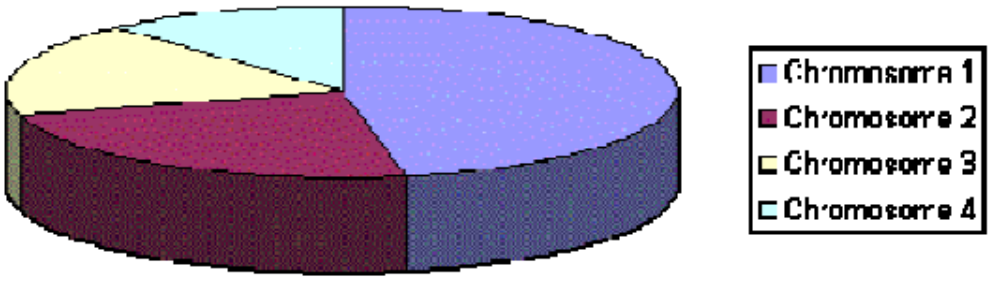
$$P_K = f_k / \sum_{i=1}^{n} f_i$$

حال کروموزوم‌ها را بر اسال Pk مرتب کرده و qk که همان مقادیر تجمعی Pk می‌باشد به صورت زیر بدست می‌آید:

$$q_k = \sum_{i}^{k} P_i$$

چرخ رولت به این صورت عمل می‌کند که برای انتخاب هر کروموزوم یک عدد تصادفی بین یک و صفر تولید کرده و عدد مذکور در هر بازه‌ای که قرار گرفت، کروموزوم متناظر با آن انتخاب می‌شود. البته روش پیاده‌سازی چرخ رولت به این شکل است که ما یک دایره در نظر گرفته و آن را به تعداد کروموزوم‌ها طوری تقسیم می‌کنیم که هر بخش متناظر با مقدار

برازندگی کروموزوم مربوطه باشد. حال چرخ را چرخانده و هر کجا که چرخ متوقف شد به شاخص چرخ نگاه کرده، کروموزوم مربوط به آن بخش انتخاب می‌گردد.



انتخاب چرخ رولت، روشی است که به نسبت مقدار تطابق، اعضا را انتخاب می‌کند. این روش یک چرخ رولت را شبیه‌سازی می‌کند تا تعیین کند کدام اعضا شانس بازتولید را دارند. هر عضو به نسبت تطابقش، تعدادی از بخش‌های چرخ رولت را به خود اختصاص می‌دهد. سپس در هر مرحله انتخاب یک عضو برگزیده می‌شود و این روند آنقدر تکرار می‌شود تا به اندازه کافی، جفت، برای تشکیل نسل بعد انتخاب گردد. این روش انتخاب را می‌توان به صورت زیر بیان کرد:

برداری مانند v در نظر می‌گیریم: $v=[1,\ldots,M]$

M تعداد عناصر بردار است و اگر تعداد اعضای مجموعه N باشد، هر عضو $i\ 1,\ldots,N$ دارای تطابقی مانند f_i می‌باشد. هر عضو i به نسبت f_i، P_i بار در v تکرار می‌شود. هر چه f_i بیشتر باشد، عضو مکان‌های بیشتری را به خود اختصاص می‌دهد. عدد از تشکیل بردار v یک مقدار تصادفی $1\le r\le M$ انتخاب می‌شود. این مقدار به مکانی در بردار اشاره می‌کند که آن مکان خود معرق عضوی از اعضای جمعیت است. به عنوان مثال اگر جمعیتی با N=4 داشته باشیم و تطابق اعضا عبارت باشد از: f1=10, f2=10, f3=15, f4=25

مقدار مجموع تطابق‌ها عبارت است از:

$$\sum_{i=1}^{N} f_i = ۶۰$$

بردار v را برداری با 60 عنصر در نظر می‌گیریم. این بردار به صورت زیر پر می‌شود. به عضو 1، 10 مکان، به عضو 2، 10 مکان، به عضو 3، 15 مکان و به عضو 4، 25 مکان اختصاص می‌یابد.

$$V=\{1,1,\ldots,1,2,2,\ldots,2,3,3,\ldots,3,4,4,\ldots,4\}$$

حالا r بین 1 تا 60 به تصادف انتخاب می‌شود. فرض کنید32=r نتیجه می‌شود:

V=[32]=3

پس عضو 3 انتخاب می‌شود.

انتخاب ترتیبی

در این انتخاب، اعضای جمعیت بر اساس تطابقشان مرتب می‌شوند. ارزش منتظره هر عضو به جای تطابقش، در این روش، به رتبه‌اش بستگی دارد. روش ترتیبی خطی که در سال 1985 توسط بیکر ارائه شده است، به صورت زیر می‌باشد: اعضا در جمعیت طبق تطابقشان به صورت صعودی از 1 تا M مرتب می‌شوند. M تعداد اعضای جمعیت است. کاربر، ارزش منتظره Max≤0 را برای عضوی که رتبه M را داراست، در نظر می‌گیرد. ارزش منتظره هر عضو i در جمعیت در زمان t عبارت است از:

$$ExpVal(i,t)=min(max-min)\ rank(i,t)/M-1$$

که min ارزش منتظره عضو با رتبه 1 است. با اعمال قیدهای

$ExpVal(i,t) = Max\ Max \geq 0$ لازم است تا $min=2-Max$ و $1 \leq Max \leq 2$ باشد. در هر نسل، اعضا مرتب می‌شوند و طبق رابطه بالا، ارزش منتظره آن‌ها تعیین می‌شود. مقدار پیشنهادی بیکر برای Max 1,1 است. این مقدار یعنی به طور میانگین انتظار می‌رود بهترین عضو 1,1 بار به عنوان والد انتخاب شود.

انتخاب بولتزمن

در چرخ رولت امکان انتخاب یک عضو به طور مستقیم به مقداری بستگی داشت که برای آن عضو از تابع تطابقش به دست می‌آمد. مشکلی که انتخاب مستقیم مقدار تطابق به عنوان تنها ملاک انتخاب به وجود می‌آورد این است که اگر در جمعیتی اختلاف بین مقدارهای تطابق در اعضا زیاد باشد، شانس انتخاب شدن اعضای بد بیشتر و بیشتر به صفر نزدیک می‌شود. به همین دلیل در روش قبل، روش ترتیبی و همچنین روش بولتزمن و برخی روش‌های دیگر، از پارامترهای دیگری علاوه بر مقدار تطابق، در انتخاب استفاده می‌شود.

در روش بولتزمن، به جای مقدار تطابق یک عضو از مقدار به نام ϕ_i برای هر عضو i استفاده می‌شود که این مقدار از رابطه بولتزمن به صورت زیر محاسبه می‌شود:

$$\varphi_i = \frac{e^{-BY_i}}{Z}$$

$$Z = \sum_{i=1}^{N} -BY_i$$

که در رابطه بالا N تعداد اعضای جمعیت و Y_i مقدار تطابق عضو i می‌باشد.

انتخاب حالت پایدار

در اکثر الگوریتم‌های ژنتیک که در مقالات ارائه شده‌اند، جمعیت جدید به طور کامل توسط فرزندان به وجود می‌آید و این فرزندان جایگزین والدین خود می‌شوند. در بعضی روش‌ها، به برخی از اعضای والد یا اعضای جمعیت قدیمی، اجازه حضور در جمعیت جدید داده می‌شود. انتخاب حالت پایدار یکی از این روش‌ها است.

در این روش، فقط تعداد اندکی از اعضای جمعیت کنونی با اعضای جدید جایگزین می‌شوند. به عبارت دیگر بدترین اعضا با فرزندانی که از بهترین اعضا به وجود آمده‌اند تعویض می‌شوند اما بافت کلی جمعیت چندان تغییر نمی‌کند.

نخبه‌سالاری

ایده نخبه‌سالاری، ویژگی‌های تازه‌ای به فرایند انتخاب اضافه می‌کند. در نخبه‌سالاری، بهترین عضو هر جمعیت زنده می‌ماند و در جمعیت بعد حضور دارد. به عبارت دیگر عضوی که بالاترین تطابق را دارد به طور خودکار به جمعیت جدید منتقل می‌شود. این روش ابتدا در سال 1975 توسط جونز معرفی شد. اعمال نخبه‌سالاری در الگوریتم ژنتیک معمولاً باعث بهبود کارایی آن می‌شود.

انتخاب رقابتی

این روش تعدادی از اعضای جمعیت را به تصادف انتخاب می‌کند و سپس اگر شرطی خاص برقرار باشد، بهترین یا تعدادی از بهترین‌های آن‌ها را به عنوان والد برمی‌گزیند. اگر شرط برقرار نشود، بدترین عضو یا تعدادی از بدترین‌ها در تشکیل جمعیت آینده به عنوان والد در نظر گرفته می‌شوند.

شکل استاندارد این روش، رقابت دوتایی یا باینری است و به شکل زیر می‌باشد:

- 2 عضو به تصادف انتخاب می‌شوند.

- مقدار r بین 0 و 1 به تصادف تعیین می‌شود.
- پارامتر $0 \leq k \leq 1$ توسط کاربر تعیین می‌شود مثلاً $k=0.75$
- اگر $r<k$ عضو برتر و اگر $r \geq k$ عضو بدتر بین این دو عضو به عنوان والد انتخاب می‌شود.
- دو عضو انتخاب شده برای رقابت به جمعیت برمی‌گردند و می‌توانند دوباره در رقابت شرکت کنند.

روش انتخاب رقابتی می‌تواند به صورت رقابت nتایی نیز انجام شود.

انتخاب قطع سر

در این روش که توسط گلدبرگ معرفی و ارائه شده، ابتدا یک عدد T که کوچکتر از صد می‌باشد، تعریف شده سپس کروموزوم‌ها را بر مبنای مقادیر برازندگی مرتب کرده و T درصد برتر را انتخاب می‌کنیم حال از هر یک از آنها 100/T کپی به نسل بعد انتقال می‌دهیم. مثلاً فرض کنید T=20 و تعداد کروموزوم‌ها نیز 20 عدد باشد. بنابراین 20% کروموزوم‌های اولیه یعنی 20/5=4 تای اول را از لیست مرتب‌شده انتخاب کرده و از هر کدام 5 کپی در نظر می‌گیریم.

انتخاب قطعی برینذل

این روش که توسط برینذل معرفی و ارائه شده، به این صورت است که احتمال انتخاب برای هر کروموزوم طبق

$$P_K = \frac{f_k}{\sum f_i}$$

محاسبه می‌شود و تعداد مورد انتظار برای هر کروموزوم نیز به صورت $e_k = P_k \times pop\text{-}size$ تعریف شده است.

حال هر کروموزوم بر طبق قسمت صحیح مقدار مورد انتظار به نمونه تخصیص داده می‌شود و سپس جمعیت بر حسب قسمت اعشار تعداد مورد انتظار مرتب شده و به تعداد مورد نیاز جهت تکمیل جمعیت، به ترتیب از بالای لیست برداشته می‌شود.

انتخاب جایگزینی نسلی اصلاح شده

در این روش که توسط وایتلی ارائه شده است، ابتدا کروموزوم‌ها بر اساس مقدار برازش منظم شده، سپس به تعداد نوزادان تولید شده از انتهای لیست، کروموزوم‌ها حذف می‌گردند، آنگاه نوزادان جایگزین کروموزوم‌های حذف شده می‌شوند. مثلاً اگر 10 کروموزوم وجود داشته باشد، ابتدا آنها را مرتب کرده و بعد اکر قرار باشد 4 نوزاد نیز تولید شوند پس از تولید نوزادان، آنها جایگزین 4 کروموزم آخر لیست می‌شوند.

انتخاب مسابقه

در این روش که توسط گلدبرگ ارائه شده، به تعداد pop-size مجموعه شامل چند عضو که از قبل مشخص شده، تولید کرده و در هر مجموعه بهترین عضو انتخاب می‌گردد. اندازه مجموعه فوق که به آن اندازه مسابقه گفته می‌شود معمولاً برابر 2 فرض می‌گردد.

انتخاب مسابقه تصادفی

این روش که توسط وزل ارائه شده، مانند حالت قبل بوده، با این تفاوت که به جای اینکه مجموعه به صورت تصادفی انتخاب شود با کمک چرخ رولت انتخاب شده و بهترین آن به عنوان یک عضو از نسل جدید در نظر گرفته می‌شود.

ترکیب

در طبیعت بقای نسل یکی از مهمترین فاکتورهاست و تنها عملگر ممکن برای این امر آمیزش است. در الگوریتم‌های ژنتیکی بطبع طبیعت آمیزش وجود دارد. آمیزش با تعویض ژن‌ها بین دو کروموزوم انجام می‌گیرد و هر کدام از کروموزوم‌ها خصوصیاتی از خود را به فرزندان انتقال می‌دهند. بدیهی است کروموزوم‌هایی که دارای برازندگی بیشتری هستند شانس بیشتری برای آمیزش دارند.

مهمترین عملگر در الگوریتم ژنتیک، عملگر ترکیب است. ترکیب، فرایندی است که در آن نسل قدیمی کروموزوم‌ها با یکدیگر مخلوط و ترکیب می‌شوند تا نسل تازه‌ای از کروموزوم‌ها به وجود آید. جفت‌هایی که در قسمت انتخاب به عنوان والد در نظر گرفته شدند، در این قسمت ژن‌هایشان را با هم مبادله می‌کنند و اعضایی جدید به وجود می‌آورند. بر اساس مباحث تئوری ترکیب اعضا با تطابق بالا باعث به وجود آمدن اعضایی می‌شود که از تظابق میانگین، تطابق

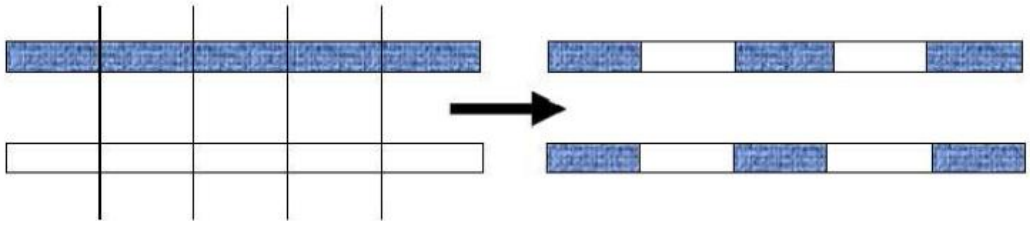
بیشتری دارند. ترکیب در الگوریتم ژنتیک باعث از بین رفتن پراکندگی یا تنوع ژنتیکی جمعیت می‌شود. زیرا اجازه می‌دهد ژن‌های خوب یکدیگر را بیابند.

جابجایی دودویی (binary crossover)

روش‌های معمول، جابجایی تک نقطه، دو نقطه، چند نقطه و جابجایی یکنواخت می‌باشد. ساده‌ترین حالت جابجایی، جابجایی تک‌نقطه‌ای (single point) است. در جابجایی تک‌نقطه‌ای، ابتدا جفت کروموزوم والد (رشته دودویی) در نقطه مناسبی در طول رشته بریده شده و سپس قسمت‌های از نقطه برش با هم عوض می‌شوند. بدین ترتیب دو کروموزوم جدید بدست می‌آید که هر نقطه از آنها ژنهایی را از کروموزوم‌های والد به ارث می‌برند.

برای جابجایی چندنقطه‌ای m موقعیت جابجا شدن و $k_i \in \{1,2,\ldots,l-1\}$ که k_i نقطه جابجایی و l طول کروموزوم است را به صورت تصادفی و بدون تکرار انتخاب می‌کنیم. سپس جهت ایجاد فرزندی جدید، بیت‌های بین نقاط مشخص شده در والدین با هم عوض می‌شوند.

به هر بیت از رشته کروموزوم‌ها یک آلل (allele) گفته می‌شود. بخش بین اولین آلل تا اولین نقطه قطع بین والدین جابجا نمی‌شود. این عملیات در شکل زیر نشان داده شده است. فلسفه انجام جابجایی در چند نقطه و البته حالت‌های مختلف دیگر عملگر جابجایی این است که قسمت‌هایی از کروموزوم که بیان کننده سهم بسزایی در عملکر بهتر یک عضو خاص هستند ممکن است در زیررشته‌های همسایه یافت نشوند.



جابجایی چندنقطه

به نظر می‌رسد نحوه عملکر عملگر جابجایی در چندنقطه (multi crossover) نسبت به روش همگرایی به مقادیر بالاتر برازندگی به پیشرفت و توسعه جستجو در فضای داده‌های مربوطه بیشتر کمک می‌کند، لذا جستجو در دامنه جواب قوی‌تر می‌شود. یانگ عملکرد جابجایی چندنقطه را مورد بررسی قرار داده و ثابت کرد که عملگر جابجایی بیشتر، عملکر الگوریتم ژنتیک را کاهش می‌دهد.

عملگرهای جابجایی یک نقطه‌ای و چند نقطه‌ای در جایی اثر می‌کنند که کروموزوم دقیقاً در آن نقاط فقط می‌تواند جابجا شود، اما عملگر جابجایی یکنواخت پتانسیل جابجا شوندگی را به

تمام نقاط یک کروموزوم به صورت یکنواخت نسبت می‌دهد. به این معنی که احتمال جابجا شدن کروموزوم در هر نقطه برابر خواهد بود. یک الگوی بیان کننده عمل جابجایی (به همان طولی که کروموزوم‌ها دارند) به صورت تصادفی ایجاد می‌شود و مقدار تعیین شده در هر بیت از این نمونه نشان می‌دهد که کدام یک از والیدن به عنوان مرجع مقداردهی برای آن بیت از فرزند خواهد بود. تعداد نقاط برش ثابت نیست ولی به طور متوسط برابر 1/2 است. دو کروموزوم اولیه (والدین) و الگوی عملگر جابجایی و فرزندان حاصل را در نظر بگیرید.

$$P_1 = 1011000111$$

$$P_2 = 0001111000$$

$$\text{Mask} = 0011001100$$

$$O_1 = 0011110100$$

$$O_2 = 1001001011$$

در اینجا مشاهده می‌شود که فرزند اول O_1 بدین صورت ایجاد شده است که اگر بیت مربوط در الگوی جابجایی (mask) برابر 1 باشد مقدار آن بیت در O_1 برابر با مقدار بیت متناظر در P_1 و همچنین اگر بیت مربوط در الگوی جابجایی 0 باشد مقدار آن بیت در O_1 برابر با مقدار بیت متناظر در P2 است.

رشته O_2 با جابجا کردن P_1, P_2 و در نظر گرفتن همین شیوه ایجاد شده است یعنی مقدار 1 در رشته mask به معنی مقدار بیت متناظر در P_2 برای همان بیت در O_2 است.

مشخصاً عملگر جابجایی یکنواخت همانند عملگر جابجایی چندنقطه‌ای باعث کاهش خطای همگرایی ناشی از طول باینری استفاده شده و نوع کدکردن سری پارامترهای داده شده می‌شود. این مسأله کمک می‌کند که بر خطای همگرایی موجود در حالت جابجایی تک نقطه‌ای در زیررشته‌های کوتاه غلبه کنیم بدون اینکه نیاز به دانستن مقادیر بیت‌های اعضا در کروموزوم‌های ارائه شده داشته باشیم.

اسپیرس و دژونگ نشان داده‌اند که چگونه جابجایی یکنواخت به وسیله احتمال عوض شدن و جابجا شدن بیت‌ها پارامتری می‌شود. این پارامتر فوق‌العاده می‌تواند بدون توصیف یک همگرایی مربوطه به طول رشته‌های استفاده شده در کنترل مقدار تغییریافته در طول ترکیب‌بندی مجدد استفاده شود؛ هنگامی که از عملگر جابجایی یکنواخت در مقادیر حقیقی استفاده شود به آن ترکیب‌بندی منفصل گفته می‌شود.

در مقایسه‌هایی که بین عملگرهای دودویی به هر دو صورت تئوری و تجربی انجام شده و نتایج بدست آمده نشان می‌دهد که هیچ یک از این عملگرها نمی‌تواند به طور مطلق بهترین بوده و اختلاف در سرعت این روش‌ها هم نمی‌تواند بیش ار 2% باشد.

عملگر جابجایی دیگری که مطرح می‌شود shuffle است. یک نقطه قطع مجزا انتخاب می‌شود اما قبل از اینکه بیت‌ها تعویض شوند، در هر دو والد بیت‌ها به صورت تصادفی جابجا می‌شوند. بعد از ترکیب‌بندی مجدد، بیت‌ها در رشته فرزند جایگذاری می‌شوند. این عملگر نیز خطای همگرایی رشته‌ها را با جابجایی تصادفی بیت‌ها در هر جایی که عملگر جابجایی انجام می‌شود حذف می‌کند.

عملگر دیگری نیز عمل جابجایی را مقید می‌کند که همیشه اعضای جدید ایجاد کند در هر جایی که ممکن باشد. معمولاً این عملگر بدین صورت عمل می‌کند که مکان نقاط قطع را محدود می‌کند به گونه‌ای که نقاط قطع تنها جایی اتفاق می‌افتند که مقادیر ژن در دو کروموزوم متفاوت است.

جابجایی حقیقی (real crossover)

در کدگذاری حقیقی که کروموزوم‌ها به صورت برداری از اعداد حقیقی می‌باشند روش‌های زیادی برای عملگر جابجایی حقیقی ارائه شده است که اکثر آنها در دو دسته زیر خلاصه می‌شوند:

- جابجایی عمومی
- جابجایی محاسباتی

عملگرهای جابجایی عمومی با توسعه روش‌های جابجایی دودویی برای کدگذاری حقیقی تهیه می‌شود که مثال ساده آن عملگر جابجایی ساده می‌باشد که شامل جابجایی تک نقطه، دو نقطه و چند نقطه است که مشابه همان حالت دودویی می‌باشند با این تفاوت که در این جابجایی یک بیت دودویی (0و1) یک عدد حقیقی در رشته است. با فرض اینکه:

$$C_1 = (C_1^1, \ldots, C_n^1) \quad \text{و} \quad C_2 = (C_1^2, \ldots, C_n^2)$$

دو کروموزومی می‌باشند که تحت عمل جابجایی قرار می‌گیرند و $i \in \{1,2,\ldots,n-1\}$ نقطه جابجایی باشد، دو کروموزوم جدید که از اعمال عملگر جابجایی ساده حاصل می‌شوند و به صورت روابط زیر خواهند بود:

$$H1 = \left(C_1^1, C_2^1, .., C_i^1, C_{i+1}^2, .., C_n^2\right)$$

$$H2 = \left(C_1^2, C_2^2, .., C_i^2, C_{i+1}^1, .., C_n^1\right)$$

$$Hk = \left(h_1^k, .., h_i^k, .., h_n^k\right)$$

که k=1,2 است. جابجایی محاسباتی بر اساس مفهوم ترکیب خطی بردارها تهیه شده است. با این فرض که دو فرزند بر اساس این عملگر تولید شده باشند، در این صورت تحت شرایط مختلف برای λ_1, λ_2 ضرایب در روابط مذکور، سه نوع مختلف از این عملگر ایجاد می‌شود:

- near crossover که در آن λ_1, λ_2 هر دو حقیقی هستند.
- affine crossover که در آن $\lambda_2+\lambda_1 =1$ است.
- convex crossover که در آن $\lambda_2+\lambda_1 =1$ بوده و λ_1, λ_2 هر دو حقیقی مثبت هستند.

اسامی linear, affine, convex از تئوری مجموعه‌های محدب وام گرفته شده است. عملگر convex معمولاً بیشتر از بقیه کاربرد دارد.

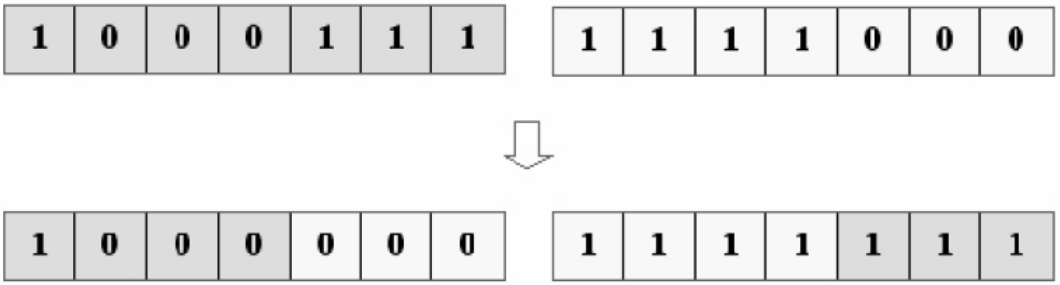
$$h_i^1 = \lambda_1 C_i^2 + \lambda_2 C_i^1 \qquad h_1^1 = \lambda_1 C_i^1 + \lambda_2 C_i^2$$

روش‌های پیاده‌سازی عملگر ترکیب

ترکیب تک نقطه‌ای

ترکیب تک نقطه‌ای، دو کروموزوم را با انتخاب تصادفی موقعیتی مانند P ترکیب می‌کند که P مقداری کمتر یا مساوی طول کروموزوم‌هاست. اگر N تعداد ژنها در کروموزوم‌ها باشد، از دو کروموزوم والد، دو فرزند به صورت زیر به وجود می‌آید:

یک فرزند با کپی کردن ژن‌های 1,..,P-1 از کروموزوم والد اول و ژن‌های P…N از کروموزوم والد دوم ساخته می‌شود و فرزند دیگر به طور مشابه، این بار با کپی کردن ژن‌های P-1…1 از والد دوم و ژن‌های P…N از والد اول به وجود می‌آید. در این نوع ترکیب از دو والد، دو فرزند به وجود می‌آید. به عنوان مثال، این نوع ترکیب در شکل زیر نشان داده شده است. در این مثال P=4 می‌باشد.



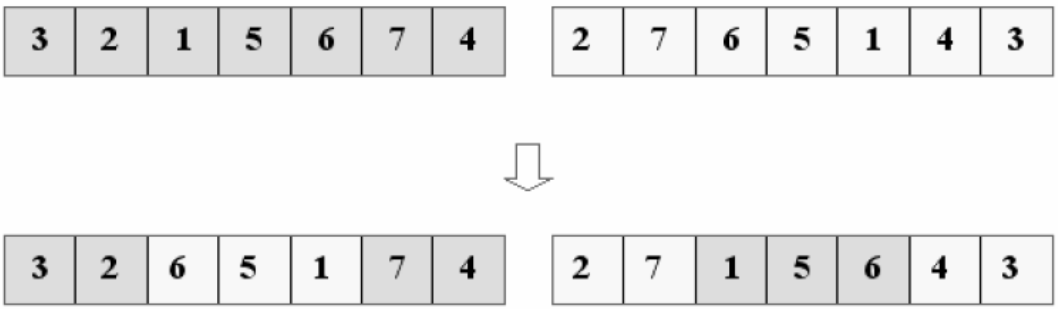
ترکیب تک نقطه‌ای

لازم به ذکر است اگر P برابر یک شود یا برابر طول کروموزوم‌ها، آنگاه دو والد بدون تغییر وارد جمعیت بعدی می‌شوند.

ترکیب دو نقطه‌ای

در ترکیب دو نقطه‌ای، دو موقعیت p1, p2 به عنوان موقعیت‌های ترکیب به طور تصادفی بین 1 و طول کروموزوم‌ها (N) انتخاب می‌شود. روش ایجاد فرزندان مانند ترکیب تک نقطه‌ای است. فرزند اول، ژنهای 1-P1…1 را از والد اول، ژنهای P1…P2-1 را از والد دوم و ژنهای P2…N را مجدداً از والد اول به ارث می‌برد. فرزند دوم، ژنهای 1-P1…1 را از والد دوم، ژنهای P1…P2-1 را از والد اول و ژنهای P2…N را مجدداً از والد دوم بدست می‌آورد.

در این روش ترکیب نیز، از یک جفت، دو فرزند به وجود می‌آید. در این روش، احتمال اینکه والهد بدون تغییر به جمعیت بعد منتقل شوند کمتر است. در شکل زیر نمونه‌ای از این ترکیب با موقعیت‌های ترکیب P1=2, P2=5 نشان داده شده است:



ترکیب دو نقطه‌ای

ترکیب n نقطه‌ای

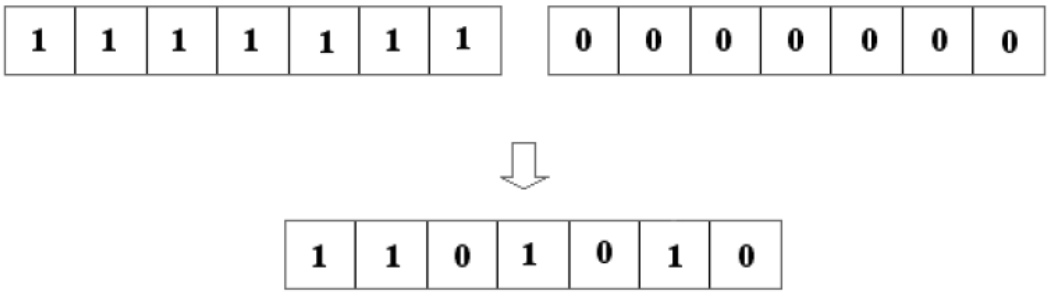
با انتخاب n موقعیت ترکیب و چیدن ژنها مشابه آنچه در ترکیب تک نقطه‌ای و دو نقطه‌ای گفته شد، ترکیب n نقطه‌ای خواهیم داشت.

ترکیب یکنواخت

در ترکیب یکنواخت، هر ژن کروموزوم جدید به صورت جداگانه انتخاب می‌شود. هر ژن وابسته به موقعیتش به صورت تصادفی از یکی از دو والد انتخاب می‌شود. مثلاً ژن اول از والد دوم، ژن دوم از والد دوم، ژن سوم از والد اول و تا ژن آخر. بر خلاف ترکیب‌هایی که قبلاً ذکر شد، این نوع ترکیب، یک فرزند به وجود می‌آورد. در واقع در این حالت از یک ماسک استفاده می‌شود.

جمعیت جدیدی که با ترکیب یکنواخت به وجود می‌آید دارای تنوع ژنتیکی بیشتری نسبت به ترکیب‌های تک نقطه‌ای و دو نقطه‌ای می‌باشد. به همین دلیل این نوع ترکیب در جمعیت‌هایی که اعضای کمی دارند اثر بهتری دارد تا جمعیت‌هایی که تعداد اعضای زیادی دارند.

در جمعیت‌های کوچک ممکن است به تنوع ژنتیکی نیاز باشد تا روش، سریعتر همگرا شود. اما در جمعیت‌های بزرگ معمولاً تنوع ژنتیکی لازم فراهم است. در شکل زیر نمونه‌ای از ترکیب یکنواخت مشاهده می‌شود.



ترکیب یکنواخت

ترکیب حسابی

ترکیب حسابی به صورت زیر تعریف می‌شود:

اگر A, B دو عضو از جمعیت فعلی باشند که به عنوان والد انتخاب شده‌اند، از آنها دو فرزند a,b به صورت زیر به وجود می‌آید:

$$a = \delta A + (1 - \delta) B$$

$$b = \delta B + (1 - \delta) A$$

پارامتر δ مقداری در بازه [0,1] می‌باشد که در هر ترکیب می‌تواند مقدار مختلفی داشته باشد.

ترتیب

این روش توسط دیویس معرفی گردیده و به روش OX معروف می‌باشد. در این روش دو عدد را به صورت تصادفی به عنوان نقاط برش به دست آورده و سپس قسمت مابین را در دو طرف کروموزوم ثابت نگه داشته ولی قسمت‌های دو طرف به این صورت به دست می‌آید که برای نوزاد اول در والد دوم از ابتدای کروموزوم شروع کرده و آنهایی که در قسمت مابین نوزاد وجود ندارد در جای خالی قرار می‌گیرند. با مثال زیر بحث روشن می‌شود:

والد اول: 476/3598/12

والد دوم: 835/7641/29

برای نوزاد اول قسمت مابین والد یک، بدون تغییر جابجا می‌شود.

نوزاد اول: --/3598/---

حال قسمت خالی از روی والد دوم پر می‌گردد.

نوزاد اول: 746/3598/12

برای نوزاد دوم نیز به همان صورت عمل می‌شود:

نوزاد دوم: 359/7641/82

چرخه

این روش توسط الیور، اسمیت و هالند معرفی شده و به نام عملگر CX معروف می‌باشد و به این گونه عمل می‌کند که ابتدا اولین ژن را عیناً از والد اول به نوزاد اول کپی کرده، سپس یک چرخه بین ژن‌هایی که در دو کروموزوم والد اول و دوم وجود دارد ایجاد می‌گردد، نقطه شروع همان ژن فوق می‌باشد. برای ایجاد چرخه باید ابتدا همان ژن را در کروموزوم دوم یافته و مکان آن را در نظر بگیریم. سپس ژن موجود در همان مکان از کروموزوم اول تثبیت می‌کنیم. این عمل تا جایی که چرخه کامل شود ادامه می‌یابد (تا جایی که به نقطه شروع برسد). در این لحظه برای تکمیل نوزاد اول باید ژنهای باقیمانده از کروموزوم دوم را به همان ترتیب نوزاد اول جایگذاری نمود. مثال:

والد اول: 346578291

والد دوم: 794356281

برای تولید نوزاد اول، ابتدا اولین ژن از کروموزوم والد اول را منتقل کرده و عمل ادامه می‌یابد:

نوزاد اول : ----57---3

حال که چرخه کامل شد، عددهای به دست آمده را در والد دوم حذف کرده باقیمانده به ترتیب عبارت است از 946281 که به همین حالت در جاهای خالی نوزاد اول قرار می‌گیرد:

نوزاد اول: 394576281

برای نوزاد دوم نیز به همان صورت عمل می‌شود:

نوزاد دوم: ----35---7

که با ادامه روند فوق رشته زیر حاصل می‌شود:

نوزاد دوم: 746358291

محدب

در این عملگر اگر والد اول P1 و والد دوم P2 باشد، نوزاد اول و دوم به صورت زیر حاصل می‌شود:

نوزاد اول : $c_1 = \lambda_1 p_1 + \lambda_2 p_2$

نوزاد دوم: $c_2 = \lambda_1 p_2 + \lambda_2 p_2$

اگر $\lambda_1=\lambda_2=0.5$ به آن عملگر تقاطعی متوسط می‌گویند. اگر $\lambda_1=1.5, \lambda_2=-0.5$ باشد به آن نسبت سلبی گفته و در صورتی که λ_1, λ_2 به صورت تصادفی از بازه $[-d, d+1]$ انتخاب شود به آن تقاطعی میانه توسعه یافته گویند.

تقاطعی متوسط توسط دیویس و تقاطعی میانه توسعه یافته توسط مولن بین و نسبت سلبی توسط رایت ارائه شده‌اند. البته چنگ وجن حالتی را که λ_1, λ_2 دو عدد تصادفی بوده و دارای شرط $\lambda_1>1$ و $\lambda_2>0$ و$\lambda_2+\lambda_1 \leq 2$ باشد را تحت عنوان عملگر خطی ارائه نمودند.

بخش-نگاشته

این روش که توسط گلدبرگ و لینگل معرفی شده به روش PMX معروف بوده و در حقیقت همان عملگر دو نقطه برش می‌باشد که برای حالت خاص ارائه شده است.

در این روش دو عدد به صورت تصادفی به عنوان نقاط برش به دست آورده، سپس قسمت مابین دو نقطه برش را در دو کروموزوم تعویض کرده و آنگاه قسمت‌های دو طرف طوری مقدارگذاری می‌شوند که در هیچ کدام از دو کروموزوم تکرار صورت نگیرد. مثال:

والد اول: 43/5628/791

والد دوم: 65/8349/217

حال برای تولید نوزاد به این صورت عمل می‌شود که قسمت مابین را عوض کرده بعد در والد اول از ابتدای کروموزوم شروع نموده هر عددی را که در قسمت مابین کروموزوم جدید نباشد عیناً نوشته و برای تکراری‌ها جای خالی قرار داده می‌شود. سپس در والد دوم از ابتدای کروموزوم شروع کرده و هر عددی که در نوزاد جدید نباشد به جای محل خالی گذاشته می‌شود تا کلیه جاهای خالی پر گردد. برای نوزاد دوم نیز به همین صورت عمل می‌شود. بنابراین ابتدا قسمت‌های میانی را جابجا کرده دو کروموزوم زیر حاصل می‌شود:

نوزاد اول: ---/8349/--

نوزاد دوم: ---/5628/--

سپس جاهای خالی نوزاد اول در صورت تکراری نبودن ژن مورد نظر پر می‌شود.

نوزاد ا ول: 1-7/8349/--

حال جاهای خالی باقیمانده نوزاد اول پر می‌شود.

نوزاد اول: 65/8349/721

با توجه به مطالب گفته شده، نوزاد دوم به شکل زیر بدست می‌آید.

نوزاد اول: 17-/5628/--

نوزاد دوم: 43/5628/917

<u>احتمال ترکیب</u>

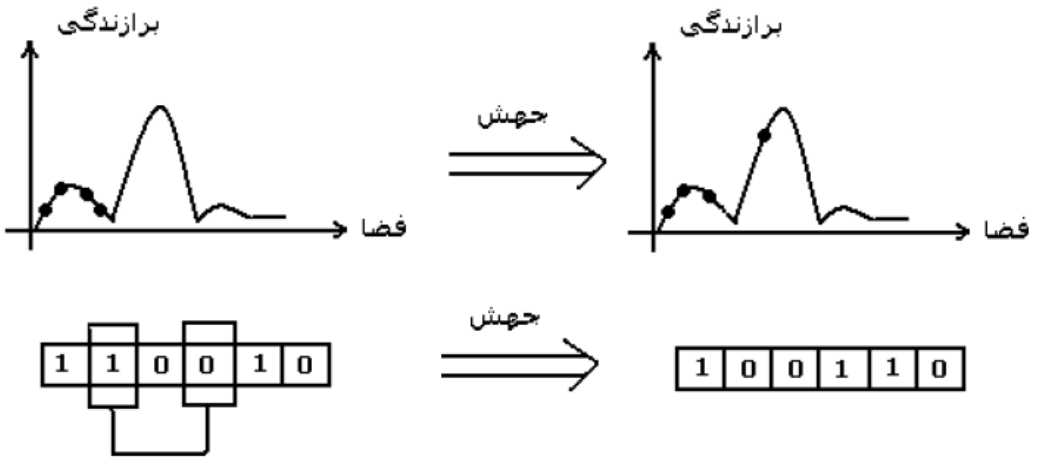
ترکیب لازم نیست در هر نسل اتفاق بیفتد. در واقع ممکن است نسل‌هایی بدون عملگر ترکیب به نسل‌های جدید تبدیل شوند. برای تعیین رخ دادن یا ندادن ترکیب از پارامتری به نام احتمال ترکیب Pc استفاده می‌شود که مقدار این پارامتر بین 0و 1 است. از آنجا که ترکیب نقشی اساسی در رشد مقدار میانگین تطابق جمعیت دارد، مقدار Pc بین 0.5 تا 0.8 و بیشتر بین 0.7 تا 0.8 درنظر گرفته می‌شود.

اگر ترکیبی صورت نگیرد، فرزندان دقیقاً همانند والدین خواهند بود. اگر ترکیب صورت بگیرد، فرزندان از بخش‌هایی از کروموزوم‌های والدین به وجود می‌آیند. اگر احتمال ترکیب 100% باشد، در این صورت، همه فرزندان در نتیجه ترکیب به وجود آمده‌اند. اگر این احتمال 0% باشد، کل نسل جدید در اثر نسخه‌برداری عینی کروموزوم‌های نسل قدیم به وجود آمده است (این بدان معنی نیست که نسل جدید همانند نسل قدیم است). ترکیب با این امید انجام می‌گیرد که کروموزوم‌های جدید حاوی بخش‌های مناسب کروموزوم‌های قدیمی است و در

نتیجه کروموزوم‌های جدید بهتر خواهند بود. با این حال خوب است که برخی از قسمت‌های نسل قدیم برای نسل بعدی باقی بماند.

جهش

در طبیعت برخی عوامل مانند تابش اشعه ماورای بنفش باعث به وجود آمدن تغییرات غیرقابل پیش‌بینی در کروموزوم‌ها می‌شوند. از آنجاییکه الگوریتم‌های ژنتیکی از قانون تکامل پیروی می‌کنند در این الگوریتم‌ها نیز عملگر جهش با احتمال کم اعمال می‌شود. جهش باعث جستجو در فضاهای دست نخورده مسأله می‌شود. می‌توان استنباط کرد که مهمترین وظیفه جهش اجتناب از همگرایی به بهینه محلی است. در اشکل زیر نحوه جهش و کارکرد آن نمایش داده شده است:



به تعبیری دیگر می‌توان جهش را مشابه شروع مجدد تصافی الگوریتم تپه‌نوردی به هنگام گیرافتادن در فلات دانست. در الگوریتم ژنتیک نیز بعد از اینکه یک عضو در جمعیت جدید به وجود آمد هر ژن آن با احتمال جهش، جهش می‌یابد. در جهش ممکن است ژنی از مجموعه ژنهای جمعیت حذف شود یا ژنی که تا بحال در جمعیت وجود نداشته است به آن اضافه شود. جهش یک ژن به معنای تغییر آن ژن است و وابسته به نوع کدگذاری، روش‌های متفاوت جهش استفاده می‌شود.

همانطور که گفته شد، هر عضو بسته به احتمال جهش، جهش می‌یابد. احتمال جهش Pm، مقداری است که توسط کاربر تعیین می‌شود. در الگوریتم استاندارد ژنتیک، مقدار این پارامتر بسیار کوچک، مثل Pm=0.01 یا حتی Pm=0.001 در نظر گرفته می‌شود؛ اما از آنجا که

کمتر از شکل استاندارد این الگوریتم استفاده می‌شود، به عنوان یک پیشنهاد می‌توان Pm را به صورت زیر تخمین زد:

$$Pm=1/N$$

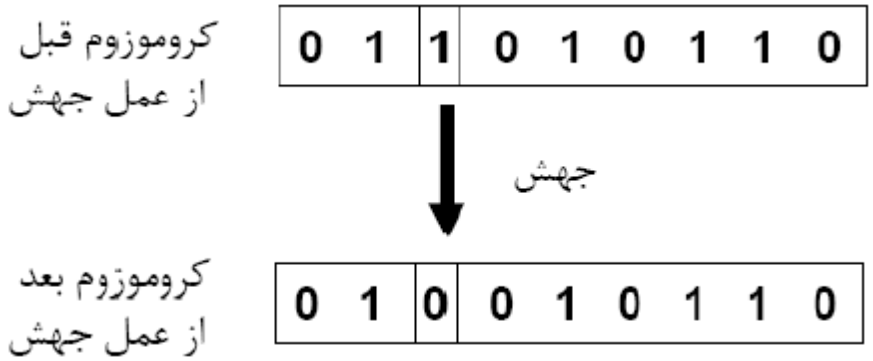
که N تعداد ژنهای کروموزوم است.

در هر حال Pm مقداری بین 0 و 1 است و معمولاً عددی کوچک انتخاب می‌شود. در فرزندی که به وجود آمده است (توسط ترکیب)، به ترتیب مقداری تصادفی بین 0 و 1 به هر ژن اختصاص می‌یابد. اگر این مقدار اختصاص داده شده از Pm کمتر باشد، ژن جهش می‌یابد و اگر بیشتر باشد، ژن تغییر نمی‌کند. نرخ بالای جهش باعث تنوع و پراکندگی ژنتیکی در جمعیت می‌شود که این پراکندگی ممکن است همگرایی را به تأخیر بیندازد. به همین دلیل برای جمعیت‌های بزرگ یا در نسل‌های آخر از Pmهای کوچکتر و برای جمعیت‌های کوچک یا در نسل‌های ابتدایی از Pmهای بزرگتر استفاده می‌شود.

تقسیم‌بندی روش‌های جهش

جهش باینری

در الگوریتم ژنتیک با کدگذاری باینری، این عملگر اغلب با تولید تصادفی یکی از اعداد 0 و 1 و جایگزینی آن به جای بیت مورد نظر صورت می‌گیرد. اما در برخی کاربردهای ژنتیک، عمل جهش دودویی در یک بیت با متمم ساختن آن بیت انجام می‌شود. بدین صورت که اگر بیت مورد نظر 0 بوده به بیت 1 و بالعکس تبدیل خواهد شد که آزمایش‌ها نشان داده‌اند که روش دوم مناسبتر است.



عملگر جهش دودویی

جهش حقیقی

در کدگذاری حقیقی، عملگر جهش باعث تولید تصادفی یک مقدار جدید در یک موقعیت خاص در کروموزوم می‌شود. در نتیجه این تغییرات تصادفی در جمعیت کروموزوم‌ها، نواحی بیشتری از فضای کاوش بررسی شده و از همگرایی بی‌موقع (ناگهانی محلی) الگوریتم جلوگیری می‌شود. یک مثال از عملگر جهش حقیقی، جهش تصادفی یا یکنواخت می‌باشد با این فرض که $C=(c_1,\ldots,c_i,\ldots,c_n)$ یک کروموزوم و c_i ژنی باشد که تحت عمل جهش قرار می‌گیرد، آنگاه c_i یک مقدار انتخابی تصادفی جدید از محدوده c_i می‌باشد که به جای ژن c_i در کروموزوم جدید جایگذاری خواهد شد. مثال دیگری برای این روش عملگر جهش مرزی است که در آن یکی از ژنهای کروموزوم به طور تصادفی با حد پایین یا بالای محدوده آن ژن جایگزین می‌شود

$(c_i=a_i \text{ or } c_i=b_i)$

چند روش برای پیاده‌سازی عملگر جهش

وارونه‌سازی بیت

از این نوع جهش هنگامی استفاده می‌شود که کدگذاری، باینری باشد. در این جا بیتی که شرایط جهش را دارد اگر 0 باشد به 1 و اگر 1 باشد به 0 تغییر مقدار می‌دهد. به عنوان نمونه اگر در شکل زیر ژن چهارم شرایط جهش را داشته باشد به صورت نشان داده شده، جهش می‌یابد.

وارونه سازی بیت

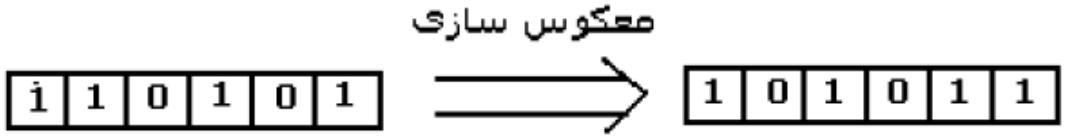
تغییر ترتیب قرارگیری

از این نوع جهش مخصوصاً در الگوریتم‌هایی استفاده می‌شود که کدگذاری بر اساس مقدار باشد البته در دیگر کدگذاری‌ها مثل کدگذاری باینری هم می‌توان این جهش را بکار برد. در این جهش، محل قرارگیری دو ژنی که می‌خواهند جهش بیایند در کروموزوم تعویض می‌شود. در شکل زیر نمونه‌ای از این جهش، نشان داده شده است.

تغییر ترتیب قرارگیری

وارون‌سازی

این عمل در طبیعت بسیار رخ می‌دهد ولی در الگوریتم‌های ژنتیکی به ندرت استفاده می‌شود و دلیل آن ایجاد تخریب زیاد است. این عملگر کروموزوم را معکوس می‌کند.



تغییرمقدار

این نوع جهش را نمی‌توان برای کدگذاری باینری یا کدگذاری‌های مشابه که امکان تغییر ژنها وجود ندارد به کار برد. در این جهش به ژنی که شرایط جهش را دارد مقداری اضافه یا کم می‌شود. اضافه شدن یا کم شدن می‌تواند به تصادف انتخاب شود یا الگوریتم مقید به استفاده از یکی از این دو عمل باشد. مقداری که به ژن افزوده یا از آن کاسته می‌شود، وابسته به محدوده مقدار ژن است و باز می‌تواند به تصادف انتخاب شود یا برای الگوریتم تعریف شود. بدیهی است مقدارهای بزرگ، پراکندگی ژنتیکی را افزایش می‌دهند. در شکل زیر نمونه‌ای از این جهش نشان داده شده است. این جهش خصوصاً برای کدگذاری‌هایی که در آنها ژنها به صورت اعداد حقیقی هستند مناسب می‌باشد.

(1.29,5.68,2.86, 4.11, 5.55) (1.29,5.68,2.73, 4.22, 5.55)

تغییر مقدار

همانطور که دیده می‌شود ژنهای سوم و چهارم کروموزوم جهش یافته‌اند. از شکل پیداست که مقدار انتخالی برای افزودن یا کستن می‌تواند از ژنی به ژن دیگر متفاوت باشد.

احتمال جهش

اگر مرحله جهش صورت نگیرد، فرزندان بلافاصله بعد از ترکیب و بدون هیچ تغییر به وجود می‌آیند (یا مستقیماً نسخه‌برداری می‌شوند که عمل ترکیب هم صورت نگرفته است). اگر تغییر صورت بگیرد، یک یا بیش از یک قسمت از کروموزوم تغییر می‌کند. اگر احتمال تغییر 100% باشد، یعنی همه کروموزوم‌ها تغییر کرده‌اند و اگر 0% باشد هیچ چیز تغییر نکرده است.

به طور کلی جهش از قرار گرفتن الگوریتم ژنتیک در اکسترمم‌های محلی جلوگیری می‌کند. جهش نباید زیاد صورت بگیرد زیرا در اینصورت الگوریتم ژنتیک به جستجوی کاملاً تصادفی تبدیل خواهد شد.