Kapitola 4

Znalostní systémy

Největších úspěchů ve využívání metod umělé inteligence je v současné době dosahováno realizací tzv. *expertních systémů*. Jsou to počítačové programy pro řešení takových úloh, které jsou všeobecně obtížné a jejichž uspokojivé řešení může provést pouze specialista (expert) v daném oboru. Jsou založeny na myšlence vhodně reprezentovat znalosti experta tak, aby mohl být vytvořen program, který je bude využívat obdobným způsobem jako expert při řešení nějaké úlohy.

Expertní systém obsahuje *bázi znalostí*, což jsou obecné poznatky (přírodní zákony a zkušenosti experta) použitelné k řešené úloze, *bázi dat* (fakta potřebná k řešení dané úlohy), *odvozovací* a *vysvětlovací mechanismus*. Odvozovací mechanismus je představován souborem procedur, které podle vložené *řídicí strategie* vybírají z báze znalostí potřebné poznatky k řešení daného příkladu za použití báze dat. Vysvětlovací mechanismus umožňuje práci v konzultačním režimu. Charakteristickým rysem architektury expertního systému je oddělení báze znalostí a odvozovacího mechanismu. To umožňuje doplňovat expertní systém novými znalostmi, poskytovat vysvětlení postupu řešení problému a vytvořit tzv. *prázdné expertní systémy* pro řešení podobných úloh.

Další charakteristickou vlastností expertního systému je schopnost používat i ne zcela exaktní znalosti a používat k řešení údaje o úloze, které jsou zatíženy určitými chybami nebo je lze přijímat jen s určitým stupněm důvěry. Velký počet úspěšně využívaných expertních systémů ve světě vede k tomu, že dnes se ve vyspělých zemích věnují na jejich další výzkum a vývoj nemalé finanční částky. Jejich praktické nasazování vedlo také ke vzniku tzv. *znalostního inženýrství*, které zahrnuje činnosti spojené s přijímáním znalostí od expertů a jejich modifikaci do formy využitelné pro zpracování a využití v expertních systémech.

Určitým zobecněním expertních systémů pak jsou tzv. *znalostní systémy*, které se od výše uvedených odlišují tím, že jejich báze znalostí obsahují znalosti obecnějšího charakteru a jsou využívány pro řešení takových úloh, kde účast experta zpravidla nevyžadujeme nebo nepotřebujeme. Lze tedy říci, že *expertní systémy jsou speciální podmnožinou znalostních systémů.*

4.1 Reprezentace znalostí, tvorba báze znalostí

Efektivnost reprezentace znalostí v počítači je považována za centrální problém; znalosti musí být formulovány tak, aby jejich reprezentace

- byla pro danou oblast dostatečně přirozenou a přitom expresivní,
- umožnila aplikaci efektivních deduktivních prostředků,
- zabezpečovala rychlý přístup k položkám v bázi znalostí i bázi dat.

Při návrhu reprezentace znalostí je důležitým požadavkem *požadavek modularity báze znalostí*. Je třeba, aby bylo možno přidávat znalosti do již existující báze. Tím je usnadněno vytváření báze znalostí postupným zjemňováním (např. rozkladem na podproblémy) i umožněna trvalá "údržba" báze. Díky tomu je báze jednak použitelná pro znalostní systém jako návod k řešení, jednak srozumitelná pro člověka k jeho poučení.

Modulární reprezentace znalostí je výhodná při provádění změn v bázi znalostí; změny jsou lokální a nepromítají se do ostatních částí báze. Při využívání báze se tato výhoda stává nevýhodou. Údaje týkající se jediného objektu mohou být roztroušeny a nelze zaručit, že se stejná informace nebude v bázi znalostí vícekrát opakovat. Chceme-li získat všechny údaje o daném objektu, je třeba prohledat celou bázi znalostí.

Požadavek sémantického sdružování znalostí vyplývá jak z potřeby rychlého vybavování znalostí, tak i z potřeby vytvářet hierarchii pojmů. Tato hierarchie umožňuje vyvozovat ve směru od obecného ke speciálnímu a naopak, odvozovat na základě analogií apod. Požadavek modularity splňují např. jazyky logických kalkulů, produkční systémy apod., požadavek sdružování zdůrazňují např. formalismy sémantických sítí nebo rámců.

V souvislosti s problematikou reprezentace znalostí v systémech umělé inteligence se často znalosti dělí na:

- znalosti reprezentované *deklarativně* (poznatky), které vyjadřují, co je nebo má být poznáno, resp. dokázáno (např. *"železo je kov"*);
- znalosti reprezentované *procedurálně* (v podobě pravidel), které říkají, jak poznávat nebo odvozovat (např. *"je-li X železo, potom je X kov"*).

Reálné systémy v sobě nutně zahrnují oba typy reprezentace, přičemž je v mnoha realizacích obtížné přesně stanovit hranici a jasně oddělit procedurálně a deklarativně reprezentované znalosti.

V různých realizacích znalostních systémů se setkáváme s nejrozmanitější reprezentací znalostí. Konstruktérům znalostních systémů je povoleno vše, jakákoliv reprezentace, která je dostatečně efektivní.

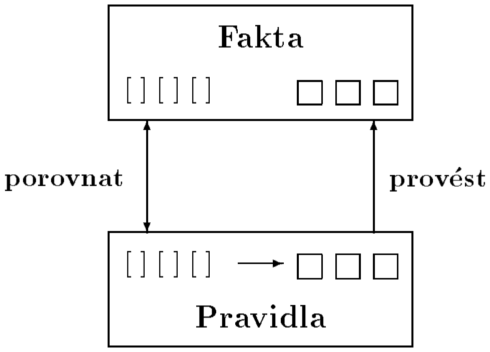
4.1.1 Reprezentace znalostí pravidly

Nejrozšířenějším způsobem reprezentace znalostí je reprezentace založená na *pravidlech*. Pravidla zajišťující formální způsob reprezentace doporučení, instrukcí nebo strategií, jsou vhodná zvláště v případech, kdy předmětové znalosti vznikají z empirických asociací, získaných za léta práce při řešení úloh v dané oblasti. Pravidla se vyjadřují v podobě tvrzení typu *IF–THEN:*

1: IF byla rozlita hořlavá kapalina,
THEN zavolejte požárníky.
2: IF pH faktor kapaliny je menší než 6,
THEN rozlitá kapalina je kyselina.
3: IF rozlitý materiál je kyselina AND je cítit octem,
THEN rozlitý materiál je kyselina octová.

Tato pravidla znalostního systému pro řešení krizové situace pomáhají určit, co bylo rozlito. Pravidla se někdy zapisují šipkou ($\rightarrow$), aby bylo vidět, kde je část *IF* a kde část *THEN* daného pravidla. Pravidlo 2 bude tedy vypadat takto:

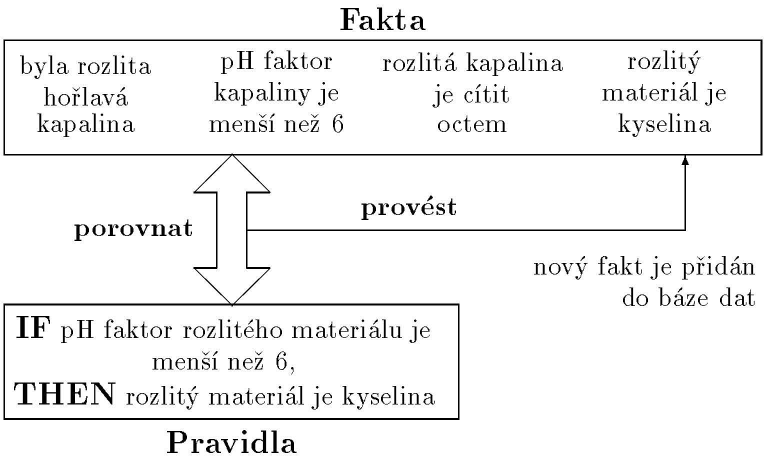
$$2:\ \textit{pH faktor rozlité tekutiny} < 6 \ \rightarrow\ \textit{rozlitou kapalinou je kyselina.}$$



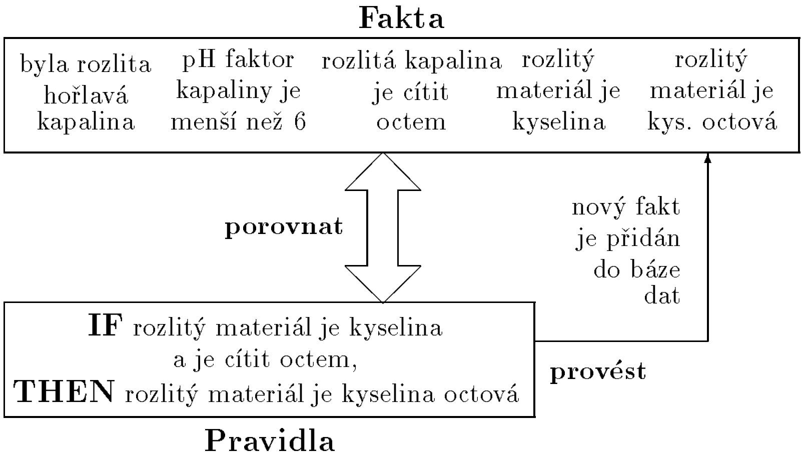
Obr. 4. 1: Interpret pravidel pracuje v cyklu

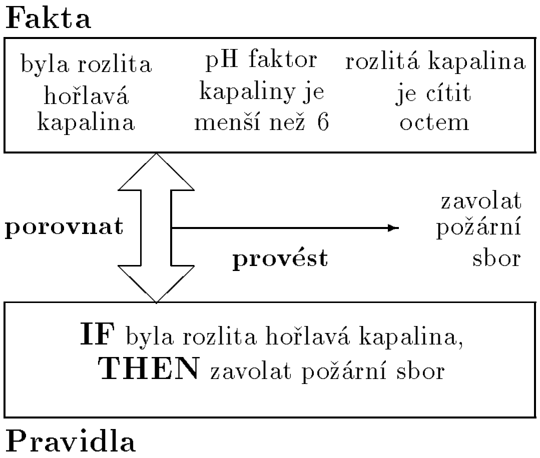
Ve znalostních systémech, založených na pravidlech, jsou předmětové znalosti reprezentovány souborem pravidel, která se prověřují na skupině faktů nebo znalostí o právě probíhající situaci. Když část *IF* pravidla vyhovuje faktům, pak se provede akce popsaná v části *THEN* a říkáme, že pravidlo je *splněno.* Interpret pravidel porovnává části *IF* pravidel s fakty a provede to pravidlo, jehož část *IF* souhlasí s fakty (viz obr. 4. 1).

Akce pravidel mohou spočívat v modifikaci souboru faktů v bázi dat, např. v doplnění nového faktu (obr. 4. 2). Nová fakta, přidaná k bázi, mohou být použita ke srovnávání s částmi *IF* pravidel (obr. 4. 3). Činnost, prováděná při splnění pravidla, může bezprostředně působit na vnější prostředí (obr. 4. 4).



Obr. 4. 2: Provedení pravidla může vést ke změně v bázi dat

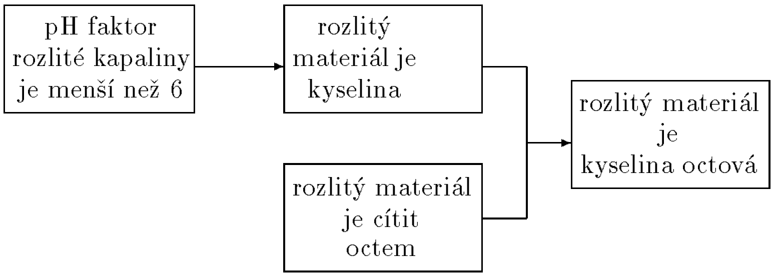




△ Obr. 4. 3: Fakta, přidaná pravidly, mohou být opět porovnávána s pravidly

◁ Obr. 4. 4: Provedení pravidel může mít vliv na reálný svět

▽ Obr. 4. 5: Řetězec odvození závěru o podstatě rozlité tekutiny



Proces porovnání s fakty částí *IF* pravidel může vést ke vzniku tzv. *řetězce odvození*. Řetězec, který vznikne v důsledku postupného plnění pravidel 2 a 3, je zobrazen na obr. 4. 5. Tento řetězec odvození dokumentuje, jak systém při použití pravidel odvozuje závěr o podstatě rozlité tekutiny. Řetězce odvození znalostního systému mohou být předvedeny uživateli, což pomáhá k pochopení postupu systému při získávání závěrů.

Existují dva způsoby použití pravidel v systému založeném na pravidlech. Jedním je *odvozování dopředným řetězením*, druhým *odvozování zpětným řetězením*. Ve výše uvedeném příkladu s rozlitou tekutinou bylo použito odvozování dopředným řetězením. Na obr. 4. 6 je podrobně uvedeno, jak tato metoda pracuje v případě jednoduché množiny pravidel.

Pravidla v tomto příkladu používají písmena pro označení situací a koncepcí:

$F \,\&\, B \rightarrow Z$

znamená:

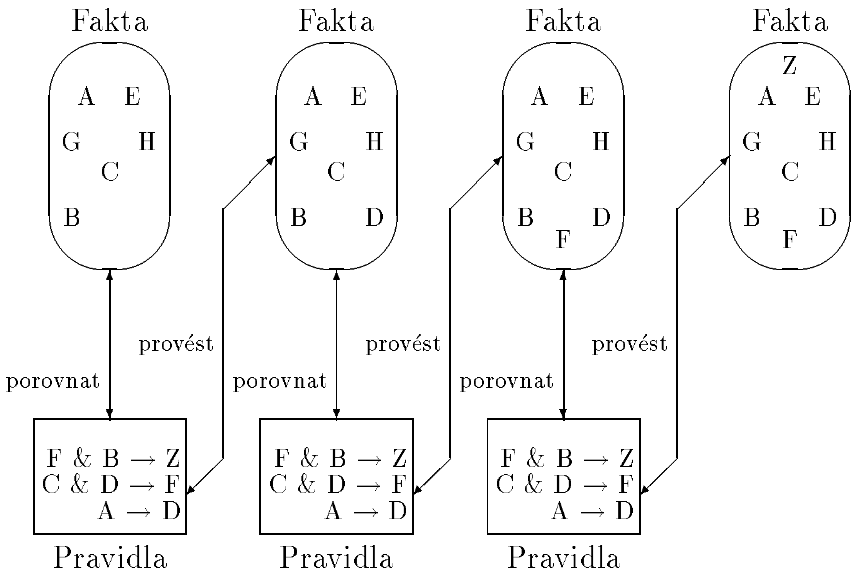
IF *existuje jak situace F, tak i situace B,*
THEN *existuje také situace Z.*

Množinu známých faktů budeme nazývat *bází dat*.

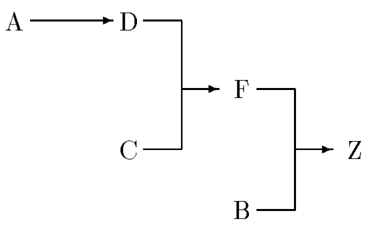
Podíváme se nyní, jak tato pravidla fungují. Dejme tomu, že pokaždé, když se množina pravidel prověřuje vzhledem k bázi dat, provede se pouze první (nejvyšší) pravidlo, které odpovídá datům. Proto se na obr. 4. 6 provádí pravidlo $A \rightarrow D$ pouze jednou, i když vždy souhlasí s bází dat.

První pravidlo, které se provádí, je $A \rightarrow D$, protože A se již nachází v bázi dat. Podle tohoto pravidla je odvozen fakt existence D a D je umístěno do báze dat. Nyní je možné vyplnit druhé pravidlo $C \,\&\, D \rightarrow F$. V důsledku tohoto pravidla je odvozeno F a umístěno do báze dat. V dalším postupu je provedeno třetí pravidlo $F \,\&\, B \rightarrow Z$, které vede k umístění Z do báze dat.

Tato metoda se nazývá *odvozování dopředným řetězením*, protože hledání nové informace se provádí ve směru šipek, které oddělují levé a pravé části pravidel. Systém využívá informace z levých částí pravidel, aby odvodil informaci obsaženou v jejich pravých částech. Odvozovací řetězec, získaný v příkladu z obr. 4. 6, je zobrazen na obr. 4. 7. Bylo odvozeno, že existuje situace Z v závislosti na F a D.

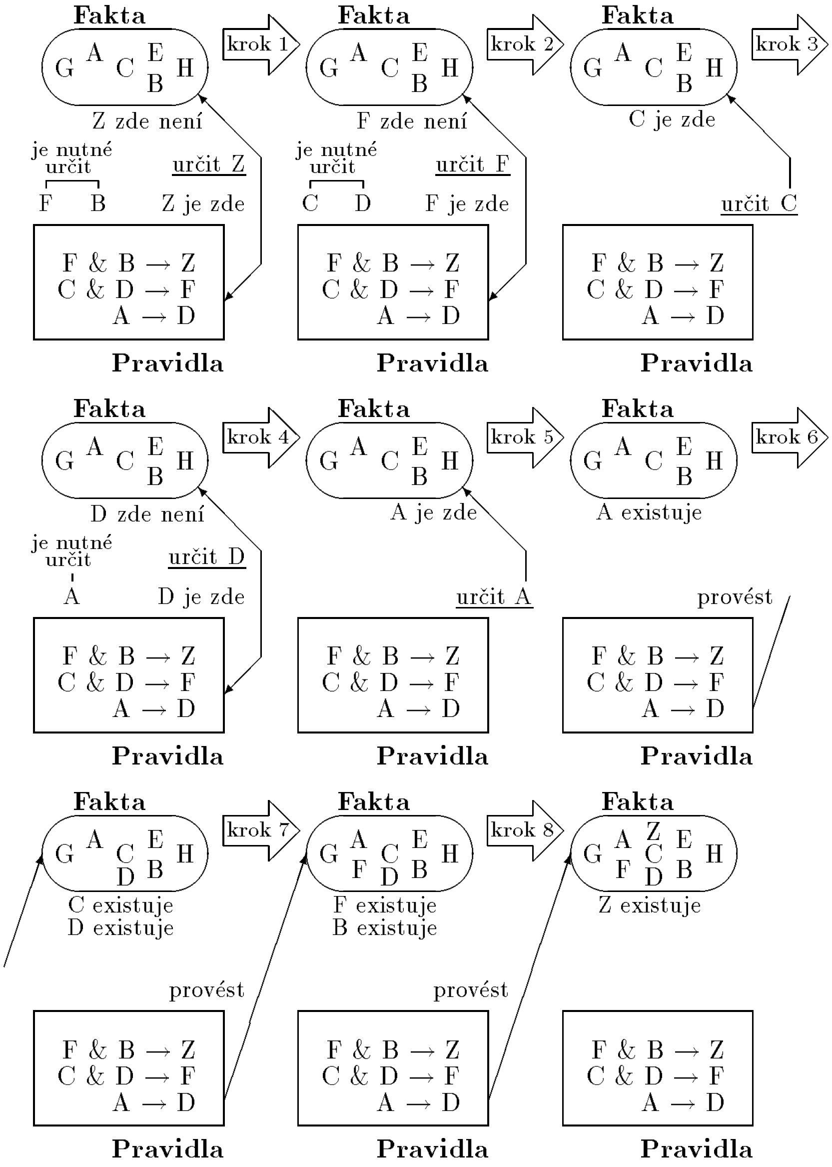


Obr. 4. 6: Příklad odvozování dopředným řetězením



Obr. 4. 7: Řetězec odvození (viz příklad 4. 6)

Předpokládejme, že jsme použili tento systém, abychom určili, zda existuje situace Z. Můžeme dojít k závěru, že systém pracuje dobře, rychle rozhoduje, že situace Z skutečně existuje. Bohužel v reálně existujících znalostních systémech nejsou jen tři pravidla, ale stovky, nebo tisíce. Pokud použijeme tak velký systém, abychom našli informaci, související se Z, pak bude provedeno i mnoho pravidel, která nemají k Z žádný vztah. Vznikne mnoho pravidel, která nemají k Z žádný vztah. Vznikne mnoho odvozovacích řetězců a situací, které jsou sice správné, ale nijak se nevztahují k Z. Proto, je-li naším cílem určení dílčího faktu (např.Z), pak je odvozování přímým řetězením jen zbytečná ztráta času a peněz.



Obr. 4. 8: Příklad odvození zpětným řetězením

V takových případech je metoda *odvozování zpětným řetězením* mnohem výhodnější. Systém začíná odvozování od toho, co je třeba dokázat (např. existuje situace Z). Je tedy nutné provést jen pravidla, která se týkají určení tohoto faktu. Na obr. 4.8 je uvedeno, jakým způsobem by mělo pracovat odvozování zpětným řetězením při použití pravidel z příkladu odvozování přímým řetězením.

Krok 1: Systém dostane za úkol určit (pokud to dokáže), že existuje situace Z. Při hledání Z nejprve prověřuje bázi dat a v případě nezdaru bude hledat mezi pravidly to, u kterého se Z nachází vpravo od šipky. Systém nalezne pravidlo $F \mathbin{\&} B \rightarrow Z$ a zjišťuje, že musí určit fakta F a B, aby mohl odvodit Z.

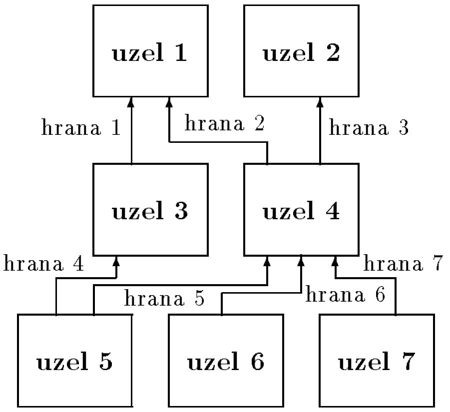
Krok 2: Systém se pokouší určit fakt F. Nejprve prověřuje bázi dat, pak hledá pravidlo, kde se F nachází v jeho pravé části. Z pravidla $C \mathbin{\&} D \rightarrow F$ systém určí, že je nutno zjistit existenci situací C a D, aby mohl získat závěr o existenci F.

Kroky 3–5: Systém nachází C v bázi dat a zjišťuje, že musí určit fakt A před tím, než může získat závěr o existenci D. Fakt A je nalezen v bázi dat.

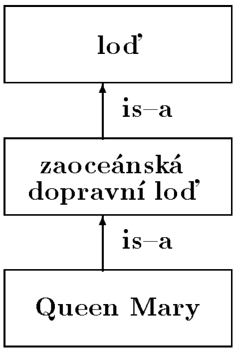
Kroky 6–8: Systém provede třetí pravidlo, aby určil D, pak provede druhé pravidlo, aby určil F, a nakonec provede první pravidlo, aby určil základní cíl – fakt existence Z. Získaný odvozovací řetězec je identický s tím, který byl vytvořen v důsledku odvozování přímým řetězením. Rozdílnost těchto přístupů spočívá ve způsobu hledání pravidel a dat.

4.1.2 Reprezentace znalostí sémantickými sítěmi

Pojem *sémantická síť* se používá pro popsání metody reprezentace znalostí, založené na síťové struktuře. Sémantické sítě byly původně vypracovány pro psychologické modely lidské paměti, ale nyní je to standardní metoda reprezentace znalostí v umělé inteligenci a ve znalostních systémech. Sémantické sítě se skládají z *uzlů* a *hran*, které spojují uzly a vyjadřují vztahy mezi nimi. Uzly v sémantické síti odpovídají objektům, koncepcím nebo událostem. Hrany mohou být určeny různými metodami, které závisí na druhu reprezentovaných znalostí. Obvykle hrany, které jsou používány pro reprezentaci hierarchie, zahrnují hrany typu *is–a* (je) a *has–part* (má část).



Obr. 4. 9: Struktura sémantické sítě



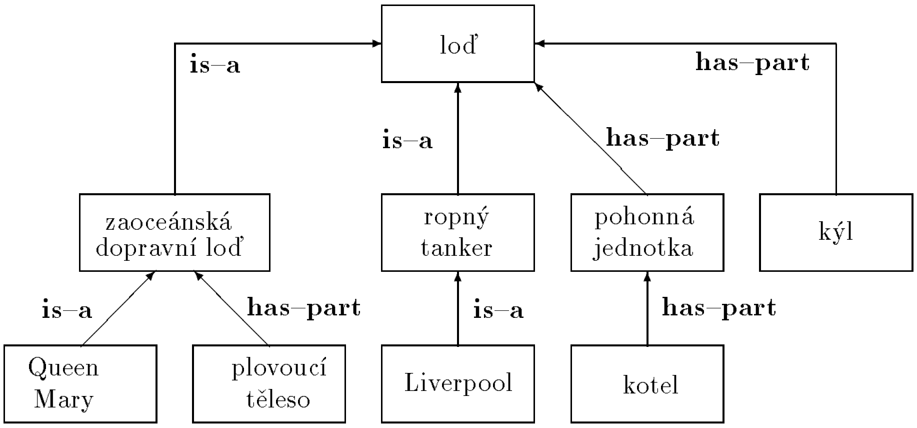
Obr. 4. 10: Jednoduchá sémantická síť s využitím vztahu *is–a*

Sémantické sítě pro popis přirozených jazyků používají hrany typu *agent, objekt, recipient.* Na obr. 4.9 je znázorněna struktura takové sémantické sítě.

Jako jednoduchý příklad prozkoumáme tvrzení *"Queen Mary je zaoceánská dopravní loď."* a *"Každá zaoceánská dopravní loď je loď."*. Tato tvrzení mohou být reprezentována sémantickou sítí, jaká je vidět na obr. 4. 10. V tomto příkladu jsou použity hrany typu *is–a*.

Protože známe vlastnosti hran spojujících uzly (např. vztah *is–a* je tranzitivní), můžeme ze sítě odvodit třetí tvrzení: *"Queen Mary je loď."*, i když nebylo zřejmým způsobem formulováno. Vztah *is–a* a jiné (na způsob vztahu *has–part*) určují vlastnost *hierarchie dědění* v síti. To znamená, že prvky nižší úrovně sítě mohou dědit vlastnosti prvků vyšší úrovně v síti. Dochází k úspoře paměti, protože není třeba opakovat informaci o podobných uzlech v každém uzlu sítě. Místo toho se může umístit v jednom ústředním uzlu sítě, jak je vidět na obr. 4. 11.

Například v sémantické síti představující loď, jsou některé její části (*pohonná jednotka, trup, kotelna*) zahrnuty najednou na úrovni lodi, místo toho, aby tyto uzly byly opakovány na nižších úrovních hierarchie, na úrovni typu lodi nebo konkrétní lodi. Tím se může ušetřit velký objem paměti. Dokonce i tehdy, budeme-li pracovat jen se stovkami lodí a jejich částí. V síti je možné uskutečnit vyhledávání při využití znalostí o smyslu vztahů, vyjádřených hranami sítě, abychom určili fakta druhu *"Queen Mary má kotelnu"*. Sémantické sítě jsou



Obr. 4. 11: Jednoduchá sémantická síť pro pojem "loď"

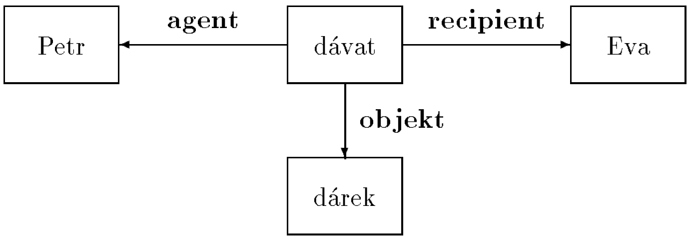
jedním z nejúspěšnějších způsobů reprezentace znalostí o předmětové oblasti s dobře stanovenou taxonomií s cílem zjednodušit hledání řešení úloh.

Sémantické sítě se také úspěšně využívají ve vědeckých pracích, které se týkají přirozeného jazyka, pro reprezentaci složitých gramatických vět. Jednoduchý příklad je uveden na obr. 4. 12.

Poznamenejme, že hrany zde určují vztahy mezi predikátem (*dávat*) a pojmy (např. *Eva, dárek*) souvisejícími s tímto predikátem. Tuto metodu lze použít i pro reprezentaci složitější věty (viz obr. 4. 13).

Věta: Petr dává Evě dárek.

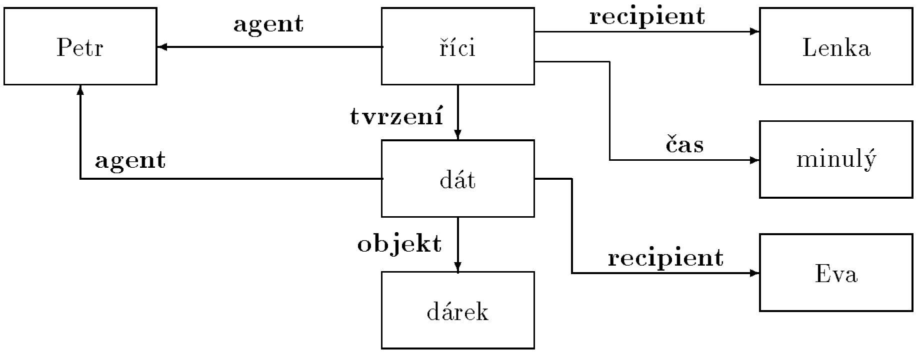
Sémantická síť:



Obr. 4. 12: Reprezentace jednoduché věty sémantickou sítí

Věta: Petr řekl Lence, že dal Evě dárek.

Sémantická síť:



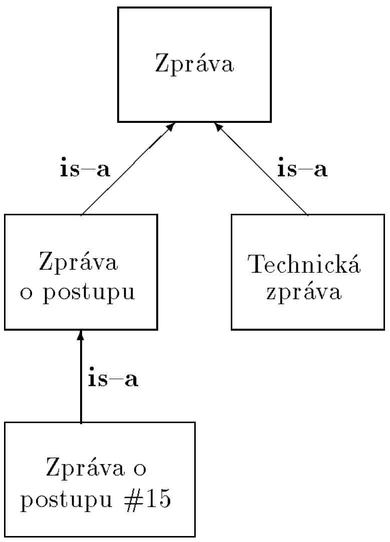
Obr. 4. 13: Reprezentace souvětí sémantickou sítí

Reprezentace sémantickou sítí je výhodná, protože zajišťuje standardní postup analýzy smyslu věty. Kromě toho, ukazuje podobnost ve smyslu úzce souvisejících vět, ale majících různou strukturu. I když věty na obr. 4. 12 a 4. 13 vypadají velmi rozdílně, sémantické sítě, které reprezentují smysl těchto vět, jsou si velmi podobné. Ve skutečnosti je celá sémantická síť z obr. 4. 12 obsažena v síti na obr. 4. 13.

4.1.3 Reprezentace znalostí rámci

V oblasti umělé inteligence se termín *rámec* vztahuje ke speciální metodě reprezentace společných koncepcí a situací. Význam rámce je možné vyjádřit následovně:

Rámec je struktura dat, reprezentující stereotypní situaci (např. pozvání na večírek). Ke každému rámci se přidružuje informace. Část této informace je o tom, jak používat rámec. Část je o tom, co je možné očekávat dále. Část je o tom, co je třeba udělat, pokud se očekávání nesplní.



Obr. 4.14 Pojem "Písemná zpráva"

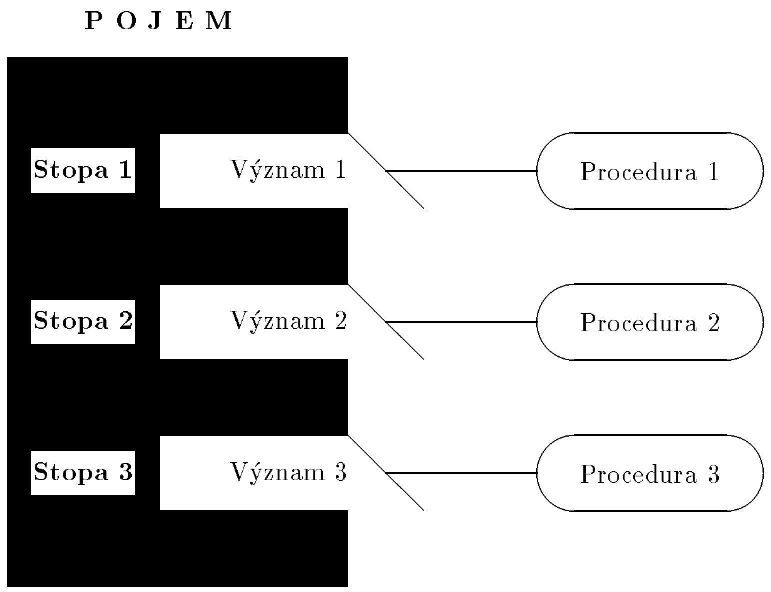
Rámec je svou organizací velmi podobný sémantické síti (ve skutečnosti považujeme jak sémantické sítě, tak i rámce za systémy založené na rámcích). Reprezentaci rámci si můžeme představit jako graf s uzly a hranami organizovanými hierarchicky, kde vrchní uzly představují všeobecné pojmy a nižší uzly více ojedinělé případy těchto pojmů, tzv. *instance*. V systému založeném na rámcích může být pojem *písemná zpráva* organizován tak, jak je uvedeno na obr. 4.14.

V systému založeném na rámcích je pojem v každém uzlu určen souborem *atributů* (např. *jméno, barva, rozměr*) a významy (*hodnotami*) těchto atributů (např. *nový, modrý, malý*). Atributy jsou nazývány *stopami*. Každá stopa může být spojena s procedurami, které se provádějí, pokud se mění informace ve stopách (hodnoty atributů). Příklad takového uzlu je uveden na obr. 4.15.

S každou stopou je možné spojit libovolný počet procedur. Níže jsou uvedeny tři typy procedur, které jsou nejčastěji spojovány se stopami:

1. Procedura IF_ADDED — provede se, pokud se přidává nová informace do stopy;
2. Procedura IF_REMOVAL — provede se, pokud je ze stopy odebrána (zrušena) informace;
3. Procedura IF_NEEDED — provede se, pokud je žádána informace ze stopy, která je prázdná.

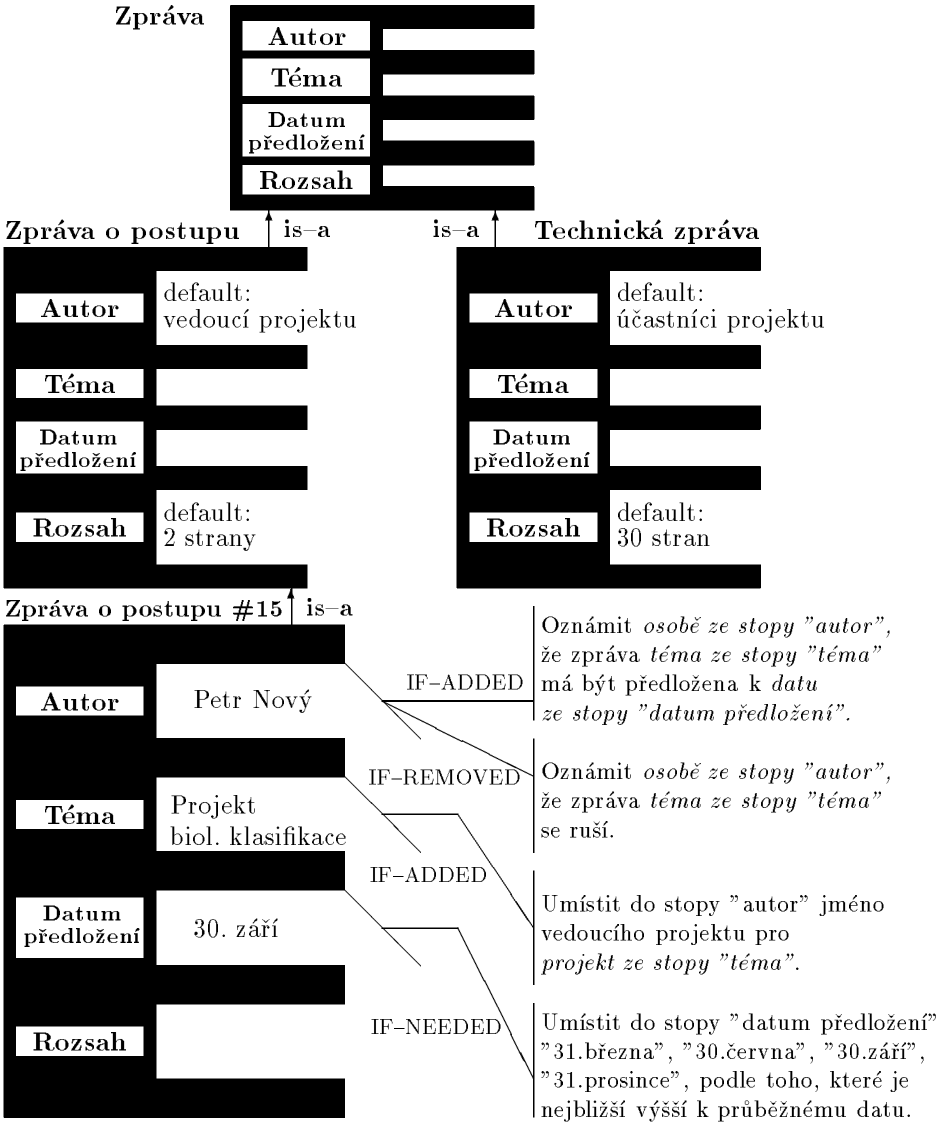
Tyto procedury mohou "dávat pozor" na připisování informace k danému uzlu a kontrolovat, zda se při změně významu provádějí odpovídající činnosti. Jak je vidět z jejich struktury, systémy založené na rámcích jsou vhodné v těch předmětových oblastech, kde hrají velkou roli očekávání vztažená na formu a obsah dat (např. interpretace vizuální informace nebo porozumění řeči).



Obr. 4. 15: Uzel v rámcovém systému

Abychom ilustrovali, jak pracuje systém založený na rámcích, je na obr. 4. 16 uvedena hierarchie zprávy, jejíž struktura je znázorněna na obr. 4. 14 se všemi stopami, jejich hodnotami a procedurami. Připusťme pro zjednodušení, že některé stopy mají *default* hodnotu (např. není-li informace o opaku, pak je autor zprávy o postupu práce považován za vedoucího projektu).

Jak lze použít takovým způsobem organizované znalosti? Předpokládejme, že vedoucí specialista pro programové zabezpečení v nějakém podniku má k systému (založenému na rámcích) přístup přes terminál. Připusťme, že uvedený specialista komunikuje se systémem prostřednictvím programového prostředí, které dovoluje komunikaci v jazyce blízkém přirozenému. Specialista zadává: *"Potřebuji zprávu o postupu projektu biologické klasifikace"*. Programové prostředí analyzuje tuto žádost a ukládá *Projekt biologické klasifikace* do stopy ≪*téma*≫ následujícího prázdného uzlu ≪*zpráva o postupu*≫, v našem případě uzlu *#15*. Dále vše probíhá automaticky.



Obr. 4.16: Reprezentace pojmu "Zpráva" rámci

1. Procedura IF_ADDED, která je spojena se stopou ≪*téma*≫, se provede, protože do stopy byla uložena hodnota. Tato procedura uskutečňuje hledání (v bázi dat systému) vedoucího projektu biologické klasifikace. Připusťme, že jméno vedoucího je *Petr Nový*. Procedura zapíše toto jméno do stopy ≪*autor*≫ zprávy o postupu prací *#15*.

2. Procedura IF_ADDED, která je spojena se stopou ≪*autor*≫, se provede, protože do stopy byla právě vepsána hodnota. Tato procedura začíná sestavovat oznámení, které má být doručeno *Petrovi Novému*, ale zjišťuje, že nemá potřebnou hodnotu ≪*datum předložení*≫.

3. Procedura IF_ADDED prohlíží stopu ≪*datum předložení*≫ a najde ji prázdnou. Aktivuje proceduru IF_NEEDED, která je spojena s touto stopou. Procedura IF_NEEDED s použitím kalendáře v bázi dat určí průběžné datum a rozhoduje, že datum ≪*30. září*≫ je k němu nejblíže. Procedura pak vepíše ≪*30. září*≫ do stopy s datem předložení.

4. Nyní procedura IF_ADDED, která je spojena se stopou ≪*autor*≫, zjistí, že chybí ještě jedna hodnota, kterou je třeba zahrnout do oznámení. Tou je ≪*rozsah zprávy*≫. Tato stopa však není spojena s žádnou procedurou a nemůže tedy pomoci při hledání hodnoty. Ale nad uzlem *#15* existuje uzel *obecné koncepce zprávy o postupu prací*, který obsahuje hodnotu rozsahu. Procedura využije tuto hodnotu a sestaví následující zprávu:

"Petře Nový, ukončete zprávu o postupu práce na projektu biologické klasifikace k 30. září. Předpokládaný rozsah zprávy je roven 2 stranám."

Jestliže je kdykoliv jméno *Petr Nový* odstraněno ze stopy ≪*autor*≫, pak systém automaticky odešle oznámení, že se jeho zpráva již nepotřebuje (provede procedura IF_REMOVAL).

4.1.4 Tvorba báze znalostí

Kvalita báze znalostí ovlivňuje rozhodujícím způsobem efektivitu celého znalostního systému. Proto musí být tvorbě báze znalostí věnována mimořádná pozornost.

Tvorba báze znalostí je vždy dlouhodobým procesem získávání znalostí od experta a jejich kódování do tvaru, vhodného pro příslušný znalostní systém. Účastní se jej jak expert ve zvolené problémové oblasti, tak specialista pro tvorbu báze znalostí (*znalostní inženýr*), protože expert sám obvykle není schopen své znalosti nejen kódovat, ale i formulovat způsobem vhodným pro počítačovou reprezentaci. Na znalostního inženýra jsou kladeny tyto požadavky: musí být dostatečně seznámen s problematikou znalostních systémů, s možnostmi reprezentace znalostí a s řídicími mechanismy znalostních systémů, které má k dispozici. Jeho úkolem je vniknout do terminologie a základů problémové oblasti, získávat znalosti od experta a vhodným způsobem je zakódovat.

Proces tvorby báze znalostí lze obecně rozložit do těchto etap:

identifikace problému,
návrh koncepce báze znalostí,
volba reprezentace znalostí,
implementace,
ladění báze znalostí.

Ladění je časově nejdelší etapou. V průběhu této etapy se trvale opakuje cyklus:

1. testování báze znalostí na reálných případech,
2. konzultace výsledků s experty,
3. úprava báze znalostí.

Je užitečné testovat jak typické, tak zejména hraniční, extrémní, netypické případy, neboť tak lze nejrychleji odhalit nedostatky v bázi znalostí. Není řídký ani případ, kdy se v průběhu ladění ukáže nezbytnou podstatná modifikace báze znalostí či její části.

Každá z uvedených etap je časově náročná (řádově týdny a měsíce práce) a vyžaduje opakovaná pracovní setkání znalostního inženýra a experta.

4.2 Struktura a funkce znalostního systému

Základním problémem umělé inteligence je otázka reprezentace velkého množství znalostí ve tvaru, který by dovoloval jejich efektivní využívání a interakci. V umělé inteligenci je tedy kladen větší důraz na znalosti, než na mechanismy jejich využívání. Vznikly systémy, u nichž je oceňována kvalita, rozsah a reprezentace znalostí – expertní systémy.

Expertní systémy jsou programy pro řešení úloh, které jsou všeobecně považovány za obtížné a jejichž uspokojivé řešení může provést pouze specialista v daném oboru (expert).

Expert se při řešení opírá o znalosti, které získal jednak studiem známých zákonitostí v dané problematice, jednak vlastní zkušeností z řešení podobných úloh. Jen malá část znalostí experta má tvar formálních matematizovaných teorií, které poskytují jednoznačné výpočetní postupy vedoucí k řešení. Většina znalostí má charakter heuristik. Soubor znalostí experta je tedy tvořen dvěma částmi: formální a heuristickou. Protože formální část znalostí je podstatně lépe sdělitelná, lze očekávat, že schopnosti experta budou úměrné právě heuristické složce jeho znalostí (více závislé na zkušenostech). Heuristiky nezaručují optimalitu řešení, dokonce ani nalezení řešení, i když existuje. Protože však nelze většinu problémů reálného světa řešit matematicky exaktními metodami, jsou heuristické znalosti významným pomocníkem a často i jediným dostupným prostředkem.

Expertní systémy jsou založeny na myšlence převzetí znalostí od experta a jejich vhodné reprezentaci tak, aby je mohl využívat program podobným způsobem jako expert, a to i s podobným výsledkem. Cílem expertních systémů není co nejvěrněji modelovat mentální procesy při rozhodování, ale dosáhnout co nejlepších odezev systému na reálná data.

4.2.1 Charakteristické rysy expertních systémů

Konzultační činnost experta probíhá obvykle formou dialogu; expert klade dotazy, jejichž zodpovězení považuje v daném okamžiku za nejdůležitější. Expert musí být schopen kdykoliv v průběhu konzultace vysvětlit postup svého uvažování, event. sdělit dílčí závěry.

Požadavky kladené na činnost expertních systémů se odvíjejí z představ o činnosti experta, resp. skupiny expertů s tím rozdílem, že dnešní expertní systémy zatím většinou nevyužívají znalostí nabytých vlastní činností, nýbrž znalostí převzatých od člověka – experta.

Přes různorodost realizací expertních systémů lze vytipovat některé jejich charakteristické rysy:

1. Znalosti experta jsou vyjádřeny explicitně, v podobě samostatného počítačového souboru v tzv. *bázi znalostí*, a předem je dána pouze strategie využívání znalostí. To zabezpečuje vyšší čitelnost znalostí, jednodušší přístup expertů ke znalostem a snadnou modifikovatelnost obsahu těchto bází. Striktní oddělení báze znalostí a programového modulu pro jejich využívání, tzv. *řídicího mechanismu*, umožňuje využívat stejného řídicího mechanismu pro efektivní rozhodovací činnost v různých aplikačních oblastech.

2. Báze znalostí obsahuje veškeré znalosti (kterých expert využívá při řešení úloh v příslušné problémové oblasti): od nejobecnější zákonitosti až po znalosti specializované, od exaktně prokázaných znalostí až k nejistým heuristikám, od jednoduchých faktů až po *metaznalosti* (tj. znalosti o znalostech) atd.

3. Expertní systémy jsou určeny k řešení konkrétních případů. Data k danému konkrétnímu případu poskytuje obvykle uživatel sekvenčně, *v dialogovém režimu*. Dodávání dat lze chápat jako "dosazování" konkrétních údajů do obecných rozhodovacích struktur, zachycených v bázi znalostí. Dialog uživatele s počítačem má charakter dialogu laika či méně zkušeného odborníka s expertem. Otázky jsou expertním systémem voleny dynamicky: systém předkládá v každém kroku konzultace tu otázku, od jejíhož zodpovězení očekává co nejrychlejší upřesnění vnitřního, strojového modelu řešeného případu. Množinu všech údajů k danému případu nazýváme – na rozdíl od báze znalostí – *bází dat*.

4. Expertní systémy bývají vybaveny schopností *kombinovat nejisté znalosti s nejistými, či nepřesnými daty* v bázi dat. Nejisté znalosti bývají zpravidla vyjádřeny jako znalosti s přidělenou mírou důvěry v jejich platnost, nejistá data jsou obsažena v nejistých odpovědích uživatele (odpovědi typu "*nevím*", "*asi ano*", "*spíše ne*" apod.).

5. Expertní systémy musí být schopny poskytovat radu i v situacích, kdy část vyžadovaných dat není dostupná. Je tedy kladen důraz na to, aby v bázi znalostí byly zahrnuty násobné či alternativní cesty odvozování.

6. Expertní systém musí být nejen schopen vést s uživatelem dialog, ale i dílčí závěry a kterýkoliv dotaz *vysvětlit a zdůvodnit*.

4.2.2 Struktura a funkce expertních systémů

Z hlediska charakteru řešených úloh lze expertní systémy v podstatě rozdělit na diagnostické a plánovací.

Diagnostický expertní systém provádí efektivní interpretaci dat s cílem určit, která z hypotéz nejlépe koresponduje s reálnými daty týkajícími se daného konkrétního případu. Řešení probíhá formou postupného ohodnocování a přehodnocování dílčích a cílových hypotéz v rámci pevně daného vnitřního (strojového) modelu řešeného problému.

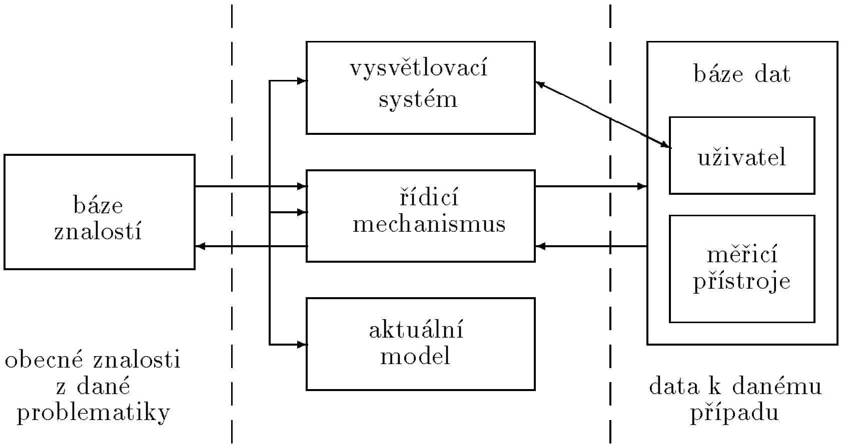
Struktura nejrozšířenější třídy diagnostických expertních systémů je znázorněna na obr. 4. 17. Jádrem takového systému je řídicí (odvozovací) mechanismus, který s využitím báze znalostí a báze dat po každé odpovědi z báze dat upřesňuje *aktuální model* konzultovaného případu. Řídicí mechanismus je odpovědný za výběr dotazu, od jehož zodpovězení očekává největší přínos k upřesnění aktuálního modelu, a za úpravu aktuálního modelu po obdržení odpovědi. Báze dat může být obecně tvořena jak přímými odpověďmi uživatele, tak hodnotami automaticky odečtenými z měřicích přístrojů.

Aktuální model je reprezentací současného stavu řešení úlohy, tj. množinou všech momentálně platných poznatků a faktů (tj. bází dat a faktografickými znalostmi z báze znalostí), a může se měnit dvojím způsobem:

a) přidáním dalšího údaje do báze dat,
b) odvozením nového poznatku na základě aktuálního modelu.

Často je aktuální model vyjádřen aktualizovanými hodnotami jistot (vah, pravděpodobností apod.) poznatků zahrnutých v bázi znalostí. Na počátku mají jednotlivé poznatky z báze přiřazeny apriorní hodnoty jistot a v průběhu programu jsou tyto hodnoty měněny.

Plánovacími expertními systémy jsou obvykle řešeny úlohy, kdy je znám cíl řešení a počáteční stav a systém má s využitím dat o konkrétně řešeném případu nalézt (pokud možno optimální) posloupnost kroků, kterými lze cíle dosáhnout. Podstatnou částí těchto expertních systémů je generátor možných řešení, který automaticky kombinuje posloupnost kroků. S rostoucím počtem kroků rostou velmi rychle kombinatorické možnosti při vytváření jejich posloupností (*kombinatorická exploze*) . Znalosti experta i data o reálném případu jsou pak užívány k výraznému omezení kombinatorické exploze zkoumaných řešení. Výsledkem činnosti plánovacího systému je *seznam přípustných řešení*.



Obr. 4. 17: Struktura diagnostického expertního systému

Na obr. 4. 18 je blokově zachycena struktura jedné třídy plánovacích expertních systémů, opírajících se o princip generování a testování přípustných řešení. Řídicí mechanismus s využitím báze znalostí a báze dat ovlivňuje výběr přípustných operátorů, omezuje generativní schopnost generátoru.

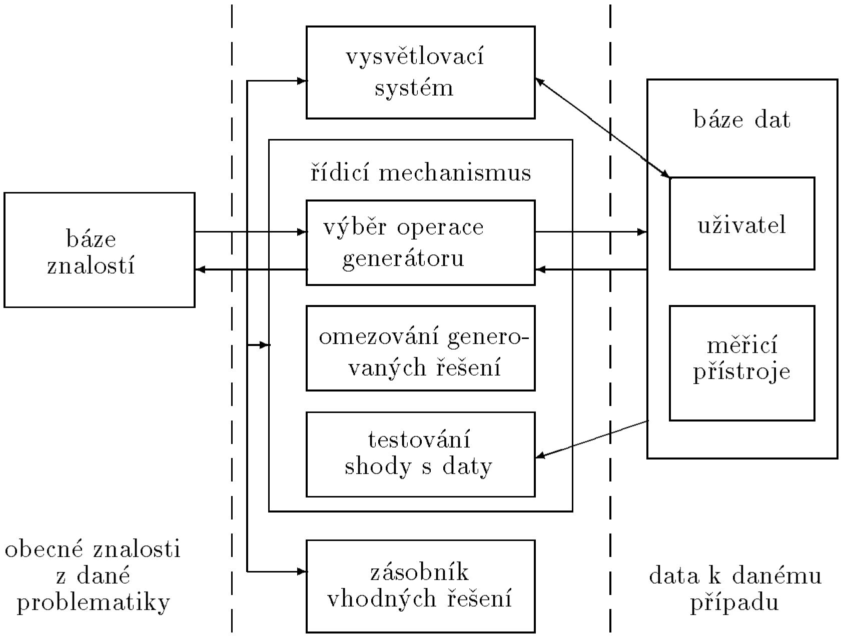
Expertní systémy bez báze znalostí a bez báze dat se nazývají *"prázdnými" expertními systémy*. Doplněním báze znalostí k prázdnému systému se systém orientuje na řešení příslušné problematiky (problémově orientovaný expertní systém). Dodáním báze dat je pak řešen konkrétní případ.

Odvozovací (řídicí) mechanismus zabezpečuje využívání znalostí v expertním systému. Ucelená teorie odvozovacích mechanismů neexistuje. Při návrhu těchto mechanismů se obvykle vychází z pojmů i výsledků obecné *teorie řešení úloh* (problem solving), především z představy *prohledávání stavového prostoru* (viz druhá kapitola). U reálných systémů tato teorie nestačí a musí být využívány:

a) další principy řízení (princip agendy, démonů, nemonotónní inference aj.),

b) techniky tzv. aproximativní inference, tj. zpracování nejisté informace.

Reálný odvozovací mechanismus je obvykle tvořen více či méně složitým komplexem inferenčních metod.



Obr. 4. 18: Struktura plánovacího expertního systému

4.2.3 Stavový prostor a jeho prohledávání

V každém okamžiku se prostředí nachází v některém ze svých *stavů*. Počáteční model popíšeme počátečním stavem, koncový model koncovým stavem. Množinu všech stavů nazýváme *stavový prostor*. Přechody mezi modely odpovídají přechodům mezi stavy. Stavový prostor lze reprezentovat orientovaným grafem; uzel grafu reprezentuje stav, orientovaná hrana přechod mezi stavy. *Řešení úloh* lze formulovat jako hledání přípustné cesty mezi uzlem počátečního stavu a uzlem cílového stavu ve stavovém prostoru. Cílový stav nemusí být popsán explicitně, mohou být udány pouze podmínky, které musí splňovat.

Nyní můžeme úlohu popsat obecným formalismem produkčních systémů, přičemž každému přechodu mezi stavy odpovídá právě jedno produkční pravidlo.

Při hledání cesty z počátečního do některého z cílových stavů můžeme postupovat od počátečního stavu k cíli, pak hovoříme o *přímém řetězení pravidel* (forward rule chaining), nebo od cíle směrem k počátečnímu stavu, pak hovoříme o *zpětném řetězení pravidel* (backward rule chaining), nebo oba režimy vhodně kombinovat.

V případě přímého řetězení se nejprve generuje a expanduje (*expanzí uzlu* rozumíme nalezení množiny všech možných bezprostředně následujících uzlů) počáteční uzel, v průběhu procesu prohledávání se pak expandují některé z dříve expandovaných uzlů. Je-li vygenerován některý uzel z množiny cílů, prohledávání končí, neboť ve stromu řešení existuje orientovaná cesta z počátečního do koncového uzlu.

Řešení úloh má obvykle nedeterministický charakter. Není pevně definováno pořadí aplikace pravidel v případě, že na daný stav lze aplikovat pravidel více (hovoříme o tzv. *konfliktní množině pravidel*).

Vzhledem k rozsahu běžných úloh může být náhodné i systematické prohledávání stavového prostoru velice neefektivní. Podobně, jako jsme v druhé kapitole definovali heuristickou ohodnocující funkci, je i zde možné podle charakteru úlohy definovat funkci, která pro každý uzel dosud vygenerovaného stromu ve stavovém prostoru určí jeho ohodnocení. Hodnoty hodnotící funkce se využívají k dalšímu výběru uzlu pro expanzi. Podle toho, zda se v ohodnocující funkci využívá znalostí o dané úloze či nikoliv, se rozlišují *algoritmy informované* a *algoritmy neinformované* (viz druhá kapitola).

Stejně jako v teorii řešení úloh i zde dělíme neinformované metody na *slepé prohledávání do hloubky* a *slepé prohledávání do šířky*. Metody slepého prohledávání jsou vždy *úplné*, tj. existuje-li cesta k cíli, pak bude vždy nalezena, ale bude expandován neúměrně větší počet uzlů, než je skutečně třeba k řešení.

Jednou z možností *formalizace informovaných algoritmů* je vyjádření heuristické znalosti ve tvaru již známé *heuristické funkce*. Efektivní metoda prohledávání stavového prostoru, využívající tuto ohodnocující funkci, se i zde nazývá *A^* algoritmus* (viz druhá kapitola). Další možností je zahrnout znalosti do předpokladů jednotlivých produkčních pravidel, čímž zmenšíme počet stavů, v nichž je pravidlo aplikovatelné. Prohledáváme tedy menší část stavového prostoru.

Velmi efektivní způsob využívání znalostí o úloze je vyjadřování znalostí ve formě samostatných pravidel (tzv. *metapravidel*), kterými se pak řídí používání pravidel definujících úlohu. Tento způsob dovoluje oddělit pravidla, která definují úlohu, od pravidel, která ji pomáhají řešit, tj. od řídicí strategie.

V souvislosti s obecným přístupem k prohledávání stavového prostoru lze využívat dalších velmi efektivních technik; např.:

a) *Dekompozice úlohy na podúlohy*: rozklad úlohy na dílčí, jednodušší podúlohy. Dekompozici úlohy zpravidla vyjadřujeme součinově-součtovým grafem (viz odst. 2.6.1).

b) *Hledání v hierarchicky uspořádaném stavovém prostoru*: danou úlohu lze popsat v různých stavových reprezentacích, které se liší obecností popisu a jsou hierarchicky uspořádány. Hledání výsledného řešení je uspořádáno hierarchicky, počínaje nejobecnějším podstromem. Každý krok je rozložen na posloupnost kroků v méně obecném prostoru atd. až je nalezeno výsledné řešení v nejméně obecném, a tedy i "nejjemnějším" prostoru.

c) *Metoda generování a testování*: generátor nabízí řešení, ta jsou testována a odmítána, nesplňují-li zadané omezení. Znalosti problému lze vhodně rozdělit mezi generátor a testér. Velmi efektivní je metoda hierarchického uspořádání generování a testování, která umožňuje odmítnout řešení na základě jenom jeho částečného popisu.

4.2.4 Další techniky pro konstrukci řídicích mechanismů

Agenda: v průběhu řešení úlohy se jako vedlejší produkt řešení asociativně vytváří a přetváří zásobník dalších úkolů, které by měly být prioritně řešeny. Po dokončení dílčího úkolu se přistupuje k řešení dalšího úkolu, který je právě na vrcholu zásobníku.

Démoni: vychází se z představy o činnosti jakýchsi skřítků (démonů), kteří tiše sedí a pozorují inferenční proces. Do procesu zasahují jen za předem specifikované situace. Využívá se programů vyvolávaných za určité situace (*pattern – invoked programs*). Po každém kroku řešení je nutno prověřovat, zda jsou splněny podmínky pro vyvolání démonů.

Nemonotónní inference: uvažování na základě předpokladů, které se mohou ukázat jako nesplnitelné nebo nepřijatelné. Pracuje se např. s pravidly typu *"Jestliže je splněno A a jestliže je možné předpokládat B, pak platí C"*. Řídicí mechanismy připouštějící nemonotónní inferenci musí být vybaveny efektivními algoritmy pro korektní úpravu aktuálního modelu po "zhroucení" původních předpokladů v průběhu řešení úlohy.

Tabule: k řešení velmi složitých problémů se využívá větší počet samostatných bází znalostí (*zdrojů znalostí*). Dílčí závěry jsou ukládány do tzv. *tabule*, tj. do datové a řídicí struktury, přístupné všem zdrojům znalostí. Podstatné mezivýsledky jsou zapisovány do tabule. Údaji v tabuli může být "evokována" aktivita některých dalších zdrojů znalostí, které opět svými závěry přispějí do tabule atd. Zavedení a využívání datové struktury typu tabule vychází z představy, že daná úloha je řešena panelem expertů různých specializací. Princip tabule umožňuje zaostřovat pozornost na řešení dílčích problémů bez ztráty globálního pohledu.

4.2.5 Práce s neurčitostí

Nejistota bývá vyjadřována vahami, měrami, stupni důvěry, faktory věření, subjektivními pravděpodobnostmi ap. Často se používá hodnot z intervalu $< 0, 1 >$ nebo $<$ -1, 1 $>$.

Jak se využívá pravidla, pokud splnění předpokladu není jisté? Jak se určuje míra důvěry závěru v případě, že existuje více pravidel se stejným závěrem? Většina modelů počítá výslednou důvěru v závěr vážením vlivu všech podporujících a vyvracejících pravidel a dat. Každé tvrzení by mělo mít přiřazeno *apriorní míru důvěry* (váhu).

Začínají se postupně prosazovat přístupy, v nichž je *neurčitost vyjadřována intervalem hodnot*. Např. M. Ginsberg navrhl charakterizovat neurčitost dvojicí čísel, z nichž jedno vyjadřuje mez, při které jsme ještě ochotni na základě evidencí věřit v závěr, a druhé mez, kdy už závěr zamítáme.

V případě *kvalitativní metody aproximativní inference* se nepracuje s nejistotou, vyjádřenou číslem ze spojitého intervalu hodnot, nýbrž s diskrétními pravdivostními hodnotami z nespojitého intervalu (*"Téměř jistě"*, *"Velmi přesvědčivě"*, *"Přesvědčivě"*, *"Málo přesvědčivě"*, *"Možná"*, *"Spíše ne"*). V určitém zjednodušení lze říci, že pro každou pravdivostní hodnotu určitého tvrzení lze v bázi znalostí najít vhodné produkční pravidlo.

Inference s využitím neurčitých znalostí a neurčitých dat se nazývá *aproximativní inference*. Techniky aproximativní inference tvoří podstatnou či dokonce dominantní část řídicích mechanismů. Proto dělíme běžně používané expertní systémy podle prioritně využívané reprezentace znalostí na expertní systémy založené na pravidlech a na rámcích.

4.2.6 Expertní systémy založené na pravidlech

U plánovacích systémů se – až na vzácné výjimky – nepracuje s neurčitostí; budeme se tedy věnovat pouze *diagnostickým expertním systémům*. V nich bývají znalosti vyjádřeny ve zjednodušeném tvaru produkčních pravidel, tzv. pravidel $E \rightarrow H$:

IF < předpoklad E > THEN < závěr H > WITH < váha V >,

kde E (*evidence*) a H (*hypotéza*) jsou tvrzení výchozí a závěrné (předpoklad a závěr) a váha V je subjektivním stupněm důvěry experta v platnost pravidla. Pravidlo $E \rightarrow H$ chápeme takto:

"Je-li splnění předpokladu E naprosto jisté, akceptuj závěr H s vahou V".

Bázi znalostí, vyjádřenou pravidly typu $E \rightarrow H$, lze s výhodou reprezentovat orientovaným grafem: každému tvrzení přiřadíme uzel grafu, každému pravidlu orientovanou hranu. Získaný *acyklický orientovaný graf* zachycuje strukturu inferenčního procesu a nazývá se *inferenční síť*. Tvrzení, která vždy vystupují jen jako předpoklady, jsou reprezentována listy v grafu inferenční sítě. Tvrzení, která vždy figurují v pravidlech jen jako závěr, jsou reprezentována kořeny grafu. Ostatní tvrzení jsou tzv. mezilehlá tvrzení. Stupně důvěry experta v "sílu" jednotlivých pravidel (váhy V) tvoří ohodnocení hran inferenční sítě.

Veškerá aktuální informace je zachycena ve stupních důvěry v platnost jednotlivých tvrzení; tyto parametry se mění šířením informace podél orientovaných cest v grafu inferenční sítě s využitím expertovy znalosti o síle pravidel. *V režimu výběru dotazu* se v síti vychází od právě vyšetřované vrcholové hypotézy (kořen grafu). Vyhledává se ve směru od kořene k listům v příslušném podgrafu takový uzel, jemuž příslušející tvrzení nebylo dosud vyšetřeno, přičemž od zodpovězení tohoto tvrzení lze očekávat velmi podstatný přínos k upřesnění aktuálního modelu. Výběr dotazu je realizován prohledáváním (obvykle do hloubky) grafu od kořene grafu (cíle) k listům (požadavkům na data). Hovoříme o strategii výběru řízené cílem (*goal–driven*), resp. o zpětném řetězení pravidel.

V režimu zpracování údajů z báze dat nastává aktualizace parametrů (stupňů důvěry) všech uzlů, které leží na některé z orientovaných cest v grafu inferenční sítě vedoucích ze zodpovězeného uzlu k některým cílovým hypotézám. Aktualizace modelu je pak řízena strategií *data–driven*, tj. strategií s přímým nebo lépe dopředným řetězením pravidel.

Při vedení konzultace se prověřuje jedna z cílových hypotéz (cílovými hypotézami jsou obvykle všechny vrcholové hypotézy i některá tvrzení mezilehlá), přičemž se trvale střídá režim výběru dotazu s režimem zpracování údajů z báze dat, a to do okamžiku, kdy je

- daný cíl zcela vyšetřen (systém k němu nemá dalších otázek)

nebo

- uživatel převezme iniciativu.

Výsledkem konzultace je konečné ohodnocení cílových hypotéz, jejich uspořádání ve smyslu tohoto ohodnocení.

4.2.7 Expertní systémy založené na rámcích

V expertních systémech založených na rámcích probíhá postupně vyplňování systému struktur rámce. Algoritmy pro manipulaci s neurčitou informací a pro "výpočet" neurčitosti mají charakter heuristik a ad hoc technik.

Postupy spojené s položkami rámců lze považovat za určitá produkční pravidla vyvolávaná strategií vyplňování rámců. Rámcová reprezentace je tedy speciálním případem produkčních systémů s pravidly soustředěnými kolem sémanticky významných pojmů (*entit*) a vyvolávaných podle schématu prověřování vlastností pojmů (*entit*).

4.3 Inferenční síť

Základem báze znalostí pro znalostní systém je reprezentace znalostí pravidly typu $E \rightarrow H$, která lze interpretovat takto: je-li pravdivé tvrzení E (anglicky evidence), je pravdivé i tvrzení H (anglicky hypothesis). Je patrné, že jde o speciální případ produkčních pravidel s omezeními kladenými na situační a akční část pravidel. Situační část pravidla bude splněna, pokud je v bázi dat nalezeno nějaké tvrzení E, akční část potom znamená zápis jiného tvrzení H.

Další zvláštnost reprezentace znalostí vyplývá ze skutečnosti, že situační a akční část pravidel mají stejný tvar (jedná se o tvrzení) a že jedno tvrzení může vystupovat v některých pravidlech jako hypotéza H, v jiných jako předpoklad E. Pak lze bázi znalostí definovanou pravidly typu $E \rightarrow H$ reprezentovat orientovaným grafem, a to tak, že každému tvrzení odpovídá uzel a každému pravidlu orientovaná hrana. Takový orientovaný graf nazýváme *inferenční sítí*. Řetězení pravidel je v takové bázi znalostí (na rozdíl od klasického produkčního systému) explicitně určeno.

V inferenční síti lze obvykle nalézt uzly tří typů:

- *Vrcholové uzly*, z nichž nevede žádná orientovaná hrana. Tyto uzly reprezentují tzv. vrcholové hypotézy, v terminologii teorie grafů *norny*.
- *Listové uzly*, do nichž nevede žádná orientovaná hrana. Listové uzly reprezentují tzv. listová tvrzení, jejichž platnost je vždy nutno ověřovat pozorováním reálného světa (v terminologii teorie grafů je nazýváme *zdroji*).
- *Mezilehlé uzly*, které nejsou ani vrcholovými, ani listovými. Tyto uzly reprezentují mezilehlá tvrzení, která mohou být buď dokazována z listových nebo hierarchicky níže umístěných tvrzení, nebo někdy též ověřována přímým pozorováním reálného světa.

Ta tvrzení, která mají být v průběhu konzultace potvrzena či vyvrácena, se nazývají *cílovými hypotézami*. Cílovými hypotézami jsou obvykle všechny vrcholové hypotézy a mohou jimi být i vybrané mezilehlé hypotézy.

Každý uzel může být buď *dotazovatelný* (dotaz na platnost příslušného tvrzení nebo na kvantifikaci tvrzení je možné položit uživateli), nebo *nedotazovatelný* (nemá smysl ptát se uživatele na platnost).

Inferenční síť představuje jistou fixní, orientovanou strukturu znalostí, kostru sloužící jak k výběru vhodných otázek k zodpovězení uživatelem (výběr probíhá metodou zpětného řetězení pravidel od vrcholových uzlů směrem k listům), tak i k šíření informace dodané od uživatele (přímé řetězení pravidel od listů k vrcholovým uzlům).

Znalostní systém pracuje jak s *neurčitostí ve znalostech* (tj. s neurčitými pravidly), tak s *neurčitostí v datech* (tj. s nejistými údaji od uživatele). Neurčitost každého tvrzení v inferenční síti je zachycována stupněm neurčitosti, který je přiřazen každému tvrzení a který nazýváme pravděpodobností (i když se vlastně jedná o pseudopravděpodobnost). Tato pravděpodobnost může nabývat hodnoty od 0 (kategorická neplatnost daného tvrzení) až po 1 (kategorická platnost tvrzení).

Na počátku konzultace nabývají pravděpodobnosti *apriorních hodnot* (zadává je expert), v průběhu konzultace se tyto hodnoty podle informací mění – hovoříme pak o *aposteriorních hodnotách* pravděpodobností.

Neurčitost pravidel $E \rightarrow H$ bývá vyjadřována dvěma stupni, a to

- *stupněm postačitelnosti* pravidla, reprezentovaným subjektivní "pravděpodobností" $P(H/E)$, tj. pravděpodobností, že platí závěr H, platí-li kategoricky E, a
- *stupněm nezbytnosti* pravidla, reprezentovaným subjektivní "pravděpodobností" $P(H/\neg E)$, tj. pravděpodobností platnosti závěru H, jestliže předpoklad E kategoricky neplatí.

Jako příklad neurčitého pravidla uveďme následující pravidlo $E \rightarrow H$:

JESTLIŽE	*PAK (jedná se o angínu)*
(pacient má červeno v krku)	*se stupněm P(H/E) = 0.8*
	JINAK (jedná se o angínu)
	se stupněm P(H/¬E) = 0.001.

Tedy tvrzení $E \equiv$ *pacient má červeno v krku,*
$H \equiv$ *jedná se o angínu,*
stupeň postačitelnosti P(H/E) = 0.8,
stupeň nezbytnosti P(H/¬E) = 0.001.

Pravidlo lze interpretovat takto:
"Jestliže má pacient červeno v krku, pak bude mít angínu se subjektivní (expertem odhadnutou) pravděpodobností 80%; pokud pacient v krku červeno nemá, je subjektivní pravděpodobnost angíny 0.1%."

V inferenční síti je tedy každému uzlu (tvrzení) přiřazena pravděpodobnost, která se v průběhu konzultace (v procesu dodávání konkrétních faktů) postupně mění. Každému pravidlu jsou přiřazeny (expertem) dva pevné stupně, a to stupeň postačitelnosti a stupeň nezbytnosti. Tyto stupně vyjadřují "sílu" pravidla a slouží především k přepočtu pravděpodobností závěru při změně pravděpodobnosti předpokladu v etapě šíření informace od listů směrem k vrcholovým hypotézám.

Dosud byla popsána jenom nejjednodušší, základní varianta inferenční sítě. Znalostní systém však umožňuje práci i se složitější inferenční sítí, např. umožňuje zahrnovat logické vazby mezi tvrzeními, vyjadřovat metaznalosti ve formě *řídících* (kontextových a prioritních) *vazeb* i zpracovávat nejen kvalitativně ("ano", "ne", "asi ano", "nejspíš ne" atd.) vyjádřené informace od uživatele, nýbrž i kvantitativní, numerická data a znalosti strukturalizovat a efektivně využívat taxonomickými strukturami.

Ke zpracování kvantitativních odpovědí uživatele (hodnota této odpovědi je dále označována symbolem x) je možné používat dvou typů kvantitativních uzlů, které jsou určeny ke zpracování dvou typů kvantitativních odpovědí:

- *Typ Q (intervalový uzel)*: Odpověď uživatele se skládá z udání číselného intervalu $< x_1, x_2 >$ a stupně jistoty, že skutečná hodnota x leží v tomto intervalu (např. obsahuje–li uzel dotaz *"Jak staré je auto?"*, uživatel odpovídá např. *"8 – 9, 4"*, což znamená, že udává rozmezí 8 až 9 let se stupněm jistoty 4).
- *Typ S (jednohodnotový uzel)*: Odpovědí uživatele je přesná hodnota x, a to bez udání stupně jistoty, neboť při exaktní kvantitativní odpovědi lze očekávat naprostou jistotu (např. na dotaz *"Jakou má pacient teplotu?"* uživatel odpoví *"37,5"*).

Reprezentace kvantitativních uzlů je řešena takovým způsobem, že každý uzel obsahuje rozdělení stupnice číselných hodnot x na disjunktní intervaly.

Báze znalostí pro znalostní systém je vytvořena podle následujících zásad:

- Každé elementární tvrzení je reprezentováno uzlem sítě.
- Pravidla typu $E \rightarrow H$ se znázorňují orientovanou hranou v grafu inferenční sítě.
- Podle způsobu výpočtu aposteriorní pravděpodobnosti uzlu dělíme uzly na:
  - *uzly logické (AND, OR a NOT)*, které vyjadřují logickou pravdivost dílčích tvrzení a jejichž aposteriorní pravděpodobnost se počítá podle vztahů, převzatých z oblasti fuzzy logiky,
  - *uzly Bayesovského typu*, jejichž aposteriorní pravděpodobnost je vyčíslována podle Bayesova vztahu.

  Zvláštním typem Bayesovského uzlu je EXE–uzel.
- Uzly mohou být:
  - *nekvantitativní* (v průběhu konzultace se ověřuje platnost příslušného tvrzení),
  - *kvantitativní* (v průběhu konzultace se od uživatele vyžaduje kvantifikace příslušného tvrzení).

  Logické uzly jsou vždy nekvantitativní.
- Kvantitativní uzly dělíme na:
  - *Q–uzly*, které vyžadují odpověď uživatele formou číselného intervalu stupně jistoty, resp. více disjunktních intervalů a stupňů jistoty,
  - *S–uzly*, které vyžadují odpověď uživatele v podobě jedné přesné číselné hodnoty).

- K jednotlivým uzlům inferenční sítě příslušejí doplňkové informace, a to
  - *ke každému uzlu:*
    - typ uzlu,
    - text příslušného tvrzení,
  - *k nekvantitativnímu uzlu Bayesovskému:*
    - apriorní pravděpodobnost $P(E)$ tvrzení,
    - údaj o tom, zda je uzel dotazovatelný, nedotazovatelný nebo reprezentuje cílovou hypotézu,
  - *k nekvantitativnímu uzlu logickému:*
    - údaj o tom, zda je uzel dotazovatelný, nedotazovatelný nebo reprezentuje cílovou hypotézu,
  - *ke Q–uzlu:*
    - údaje o rozdělení stupnice číselných hodnot x na disjunktní intervaly (počet intervalů, meze),
    - distribuční funkce apriorní pravděpodobnosti $P(H/E, x)$ ve tvaru stupňovité ("schodovité") funkce,
  - *k S–uzlu:*
    - údaje o rozdělení stupnice číselných hodnot x na disjunktní intervaly (počet intervalů, meze),
  - *k EXE–uzlu:*
    - jméno volaného programu a seznam parametrů,
  - *k dotazovatelnému uzlu:*
    - identifikátor odpovědi v datovém souboru, resp. název položky odpovědi v databázovém souboru (je–li požadováno),
    - komentář (je–li požadován),
  - *k cílovému uzlu:*
    - název položky ve výstupním databázovém souboru (je–li požadováno),
- Užití kvantitativních uzlů je vázáno následujícími omezeními:
  - *Q–uzly* a *S–uzly* mohou být pouze listovými, dotazovatelnými a necílovými uzly a v pravidlech, kde E je kvantitativní uzel, může být uzlem H jen nekvantitativní Bayesovský uzel.

- Z hlediska reprezentace neurčitosti se v inferenční síti vyskytují pravidla typu $E \rightarrow H$ ve variantách
  - *nekvantitativní uzel → Bayesovký nekvantitativní uzel*
    K hraně reprezentující dané pravidlo jsou přiřazeny subjektivní, expertem určené stupně, a to stupeň postačitelnosti pravidla, vyjádřený podmíně-

nou pravděpodobností $P(H/E)$, a stupeň nezbytnosti pravidla, vyjádřený podmíněnou pravděpodobností $P(H/\neg E)$;

- *nekvantitativní uzel → logický uzel*
  K hraně reprezentující dané pravidlo není přiřazen žádný stupeň neurčitosti;
- *Q–uzel → nekvantitativní Bayesovský uzel*
  Pravidlo se vyjadřuje po částech konstantní, nespojitou funkcí $P(H/E,x)$, rozdělení intervalů je specifikováno v zápisu příslušného Q–uzlu.
  Příklad:
  Uvažujme pravidlo E → H:
  E: "Stáří vozu je x let"
  Nechť se jedná o Q–uzel s intervaly <0;2), <2;6), <6;8), <8;10), <10;12> let.
  H: "Velké roční investice do oprav"
  Expert např. specifikuje závislost P(H/E,x) následující stupňovitou (schodovitou) funkcí s hodnotami:

interval	$P(H/E,x)$
0 – 2	*1.14*
2 – 6	*1.10*
6 – 8	*1.14*
8 – 10	*1.34*
10 – 12	*1.73*

  Do zápisu vazby E → H je tedy nutno zaznamenat posloupnost stupňů P(H/E, x): 1.14, 1.10, 1.14, 1.34, 1.73. Intervaly, k nimž stupně příslušejí, jsou specifikovány v zápisu Q–uzlu. Pochopitelně, že přesnější aproximaci závislosti P(H/E, x) by bylo možné získat rozdělením reálné osy na větší počet intervalů.
- *S–uzel → nekvantitativní Bayesovský uzel*
  Pravidlo se vyjadřuje po částech lineární, spojitou funkcí $P(H/E,x)$. Tato funkce je v zápisu vazby specifikována hodnotami $P(H/E)$ v hraničních bodech intervalů; uvnitř intervalů se hodnoty *P(H/E)* linárně aproximují.

  Příklad:
  Uvažujme pravidlo E → H:
  E: "Stáří vozu je x let"
  Nechť se jedná o S–uzel s intervaly <0;2), <2;6), <6;8), <8;10), <10;12> let.
  H: "Velké roční investice do oprav"
  Expert např. specifikuje závislost $P(H/E,x)$ po částech lineární funkcí.

Do zápisu vazby $E \rightarrow H$ je tedy nutno zaznamenat hodnoty funkce $P(H/E, x)$ v celkem 6 hraničních bodech 5 intervalů (intervaly jsou specifikovány v S–uzlu), např. 0.2, 0.11, 0.1, 0.21, 0.53, 0.95.

- V orientovaném grafu reprezentujícím inferenční síť nesmějí vzniknout uzavřené cykly.
- V bázi znalostí lze používat *řídicí kontextové vazby* k podmiňování vyšetřování jednotlivých uzlů inferenční sítě, tj. tam, kde vyšetření některého uzlu je vázáno podmínkou na výsledek vyšetření jiného uzlu (např. nemá smysl se ptát, zda bratr pacienta je geneticky postižen, pokud dostatečně neprokážeme, že pacient má bratra).
- Vedle kontextových vazeb lze používat i tzv. *řídicí prioritní vazby*. Jejich použití je obdobné jako u kontextových vazeb s tím rozdílem, že řídicí prioritní vazba určuje pořadí vyšetřování uzlů bez ohledu na velikost jejich aktuální pravděpodobnosti.

4.4 Příklady technických aplikací

Znalostní systémy jsou užívány k řešení nejrůznějších úloh, ať již charakteru analytického (klasifikace, interpretace dat, porozumění složitým signálům, identifikace, lékařská a technická diagnostika), syntetizujícího (plánování, technické návrhy, návrhy terapie v medicíně, automatické programování) či smíšeného (aplikace při vyučování, monitorování).

V posledních letech se objevily stovky systémů v nejrozmanitějších aplikačních oblastech. Nejméně polovina aplikací je zaměřena na oblast medicíny. Zde jsou pro ilustraci uvedeny některé známější systémy spolu s oblastí jejich expertizy.

Aplikace v medicíně a příbuzných oborech:

1. MYCIN – provádí diagnostiku při infekčních onemocněních;
2. PUF – interpretuje výsledky plicních vyšetření;
3. VM – monitoruje data na jednotce intenzivní péče a řídí "umělé plíce" v reálném čase;
4. ONCOCIN – řídí léčení pacienta na jednotce onkologické péče;

5. HEADMED – využívá se v klinické psychofarmakologii;
6. CLOT – diagnostikuje poruchy srážlivosti krve;
7. EMYCIN – "prázdný" systém, který se předáním báze znalostí orientuje na danou problematiku; na jeho základě byly vytvořeny systémy 1, 4, 5, 6 tohoto výčtu a 8, 9 v následující skupině;
8. MOLGEN – plánuje experimenty s DNA kyselinou;
9. CADUCEUS – údajně obsahuje 85% veškerých znalostí z interního lékařství;
10. ATTENDING – provádí výuku mediků v oblasti anesteziologie;
11. EEG ANALYSIS SYSTEM – analyzuje EEG záznamy.

Nemedicínské aplikace:

1. DENDRAL – identifikuje organické sloučeniny na základě hmotnostního spektrogramu;
2. PROSPECTOR – pomáhá geologům při hledání rudných nalezišť;
3. AL/X – identifikuje závady na technologickém zařízení;
4. R1/XCON – pomáhá při konfigurování počítačů VAX/11 na základě požadavků zákazníka;
5. ACE – identifikuje závady na telefonních kabelech;
6. HEARSAY II – systém pro porozumění řeči, rozpoznává mluvený požadavek z databáze;
7. DIPMETER ADVISOR – interpretuje záznamy z naftových vrtů;
8. SACON – radí uživatelům, jak používat rozsáhlý program MARC pro strukturální analýzu;
9. DART – provádí diagnózu závad počítačů;
10. CRIB – pomáhá lokalizovat chyby v technických prostředcích a programovém vybavení počítačů;
11. EDAAS – radí odborníkům, které informace je možno zveřejnit.