

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Výuka reprezentace znalostí

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 23. dubna 2015

Josef Kazák

Abstract

Teaching of knowledge representation

This thesis focuses on the knowledge representation issue. The theoretical part contains characteristics of this field from the basic terminology to the rule-based systems. In addition, the fundamental specifics of the development environment CLISP are described because the final program of this thesis is implemented in CLIPS.

The result of the practical part of this thesis are the study materials usable for learning the knowledge representation that consist of a set of presentations and a diagnostical knowledge system and therefore, the presentations and the system analysis can be found in practical part of this text. The testing of program is also described as well as the placement of the system and the study materials on a web site. Lastly, the results assessment is discussed.

Abstrakt

Výuka reprezentace znalostí

Tato práce se zabývá problematikou reprezentací znalostí. V teoretické části textu se nachází charakteristika již zmíněné problematiky, a to od základních pojmů k produkčním systémům. Dále jsou popsána základní specifika vývojového prostředí CLIPS, ve kterém je implementován výsledný program této práce.

Výsledkem praktické části této bakalářské práce jsou materiály pro výuku reprezentace znalostí, skládající se z prezentací a diagnostického znalostního systému. Tudíž v praktické části textu je uveden rozbor prezentací a znalostního systému. Dále je popsáno testování programu, umístění znalostního systému a prezentací na webových stránkách a neposlední v řadě je uvedeno zhodnocení dosažených výsledků.

Poděkování

Chtěl bych poděkovat především svému vedoucímu práce prof. Ing. Václavu Matouškovi, CSc. za jeho ochotu, trpělivost a cenné rady. Dále bych chtěl poděkovat mé rodině za veškerou podporu při studiu.

Obsah

1	Úvod	1
2	Úvod do reprezentace znalostí	2
2.1	Definice znalostí a jejich reprezentace	2
2.2	Znalostní systémy	3
2.2.1	Struktura ZS	3
2.2.2	Typologie ZS	6
2.3	Produkční ZS	6
2.3.1	Produkční pravidla	6
2.3.2	Základní způsoby implementace produkčních systémů .	7
2.3.3	Inferenční přístupy v produkčních systémech	8
2.3.4	Metoda zpětného řetězení	8
2.3.5	Metoda dopředného řetězení	9
2.3.6	RETE algoritmus	10
3	Prostředí CLIPS	13
3.1	Popis prostředí	13
3.2	Práce s fakty	13
3.3	Pravidla v CLIPS	14
3.4	Inferenční mechanismus	15
3.5	Další možnosti v CLIPS	16
4	ZS pro frézku Jeřábek SP410PRO	17
4.1	Srovnávací a tloušťkovací frézka Jeřábek SP410PRO	17
4.2	Implementace ZS	18
4.2.1	Soubor příkazů „zs-start.bat“	18
4.2.2	Soubor pravidel „operace-s-fakty.clp“	19
4.2.3	Soubor pravidel „univerzální-poznatky.clp“	19
4.2.4	Soubor „informativní-pravidla.clp“	19
4.2.5	Soubor pravidel „problematika-zapnutí.clp“	20
4.2.6	Soubor pravidel „problematika-vypnutí.clp“	20

4.2.7	Soubor pravidel „problematika-srovnávání.clp“	20
4.2.8	Soubor pravidel „problematika-tloušťkování.clp“	20
4.2.9	Soubor pravidel „problematika-stav-frézky.clp“	20
4.2.10	Soubor pravidel „počáteční-fakta.fct“	21
4.3	Průběh identifikace problémů	21
4.4	Uživatelská rozhodnutí	22
4.5	Výpis řešení problémů	23
4.6	Výhody a nevýhody implementace	23
5	Návrh prezentací o reprezentaci znalostí	24
5.1	První prezentace	24
5.2	Druhá prezentace	25
5.3	Třetí prezentace	25
6	Prezentace o reprezentaci znalostí	26
6.1	Způsob vytvoření	26
7	Umístění praktické části BP na webu	27
7.1	Popis zakomponování prezentací	27
7.2	Popis včlenění ZS	27
8	Testování	28
9	Zhodnocení	31
9.1	Prezentace o reprezentaci znalostí	31
9.2	ZS pro frézku Jeřábek SP410PRO	31
10	Závěr	33
A	Uživatelská dokumentace	36
A.1	Požadavky aplikace	36
A.2	Instalace CLIPS a spuštění ZS	36
A.2.1	GNU/Linux	36
A.2.2	MS Windows	37
A.3	Ovládání programu	38
B	Rozhodovací diagram ZS	39
B.1	Rozhodovací diagram po částech	39

1 Úvod

V dnešní době, která je označovaná jako „informační“, jsou pro téměř každý obor stěžejní především data a informace. Tyto údaje musejí být získávány a nejrůznějšími způsoby zpracovávány. K tomuto účelu existuje několik metodik, kam patří i zpracování znalostí znalostními systémy, které jsou středem zájmu této práce.

Cílem bakalářské práce je vytvoření výukových materiálů pro předmět „Úvod do znalostního inženýrství“, vyučovaného na FAV ZČU v Plzni (zkratkou předmětu je KIV/UZI). Výukové materiály se skládají ze dvou segmentů. První částí jsou prezentace, pojednávající o disciplíně z oboru umělé inteligence, a to o reprezentaci znalostí¹. Druhou částí je pak reálná implementace vybraných metod, která bude sloužit studentům předmětu KIV/UZI jako inspirace při řešení vlastních úkolů.

V úvodu textu je rozbor problematiky reprezentace znalostí. Ovšem tato problematika je velmi rozsáhlá, tudíž je již zmíněný rozbor koncipován od obecných základů k produkčním znalostním systémům. Takto je popis rozvržen, jelikož velký díl praktické části této BP je směřován právě na již zmíněné produkční systémy.

Další kapitola pojednává o „Diagnosticém znalostním systému pro srovnávací a tlušťkovací frézku Jeřábek SP410PRO“, který vznikl jako součást praktické části BP (je výsledkem úkolu implementace vybraných metod). V kapitole je uveden krátký popis frézky, rozbor jednotlivých komponent systému atd.

Ve zbývajících kapitolách je uvedeno testování výše zmíněného systému, rozbor vytvoření prezentací pro výuku reprezentace znalostí a popis umístění celé praktické části BP na webu předmětu KIV/UZI. A neposlední v řadě se v textu nachází zhodnocení dosažených výsledků (včetně nástinu dalšího rozvoje).

¹V původním zadání této BP je uvedena modifikace či návrh webových stránek. Ovšem od tohoto bylo upuštěno a byly vytvořeny standardní prezentace, a to z důvodu zachování kontinuity s již zpracovanými materiály.

2 Úvod do reprezentace znalostí

2.1 Definice znalostí a jejich reprezentace

Nejprve je nutné definovat, co to vlastně znalosti jsou. Pod tímto termínem se skrývají nejen teoretické vědomosti o dané problematice, ale také dlouhodobé zkušenosti dané osoby (mající ony znalosti) v této oblasti. Z toho plyne, že osoba disponující rozsáhlými znalostmi v dané problematice je zároveň v této oblasti i expertem. A právě znalosti takového člověka (experta) jsou jedním ze základních důvodů vzniku teorie pojednávající o reprezentaci znalostí.

V obecné rovině je možné reprezentovat znalosti mnoha způsoby, které jsou však závislé na metodě poznání. Pokud jsou určité znalosti získány přirozenou cestou během života, pak jsou zatíženy mírou vágnosti (neboli mírou neurčitosti). Takové znalosti lze reprezentovat pouze neformálním jazykem, např. českým jazykem. A pokud jsou dané znalosti získány metodami exaktních věd, pak je lze reprezentovat jak neformálním, tak jazykem formálním, např. matematickým zápisem. Znalosti, používané v oblasti zaobírající se umělou inteligencí, se nejčastěji vyjadřují prostřednictvím:

- *Predikátové logiky* – v umělé inteligenci se velmi často užívá predikátové logiky prvního řádu, jelikož (podle [4]) umožňuje odvozovat pravdivé formule z axiomů nebo jiných pravdivých formulí.
- *Produkčních pravidel* – každé pravidlo obsahuje podmínkovou část (tzv. „antecedent“) a důsledkovou část (tzv. „konsekvent“). Tato pravidla mají obvykle formát $a_1 \wedge a_2 \dots \rightarrow b$, což značí, že pokud platí podmínka, pak platí i důsledek.
- *Sémantických sítí* – pro popis se zde užívá orientovaný graf, kde uzly znázorňují jednotlivé objekty a hrany reprezentují vztahy mezi nimi.
- *Rámce* – rámce jsou struktury, do kterých lze uložit charakteristiku libovolného objektu. S rámci se lze setkat i v objektově orientovaném programování, kde se rámec nazývá „objekt“. Každý rámec (či objekt) obvykle obsahuje název, atributy a metody.

2.2 Znalostní systémy

Obecná definice všech znalostních systémů (dále jen ZS) v podstatě vůbec neexistuje, což je zapříčiněno rozmanitostí prostředí, ve kterých lze nejrůznější mutace těchto systémů využít. Jedna z možných definic (převzata z [1]) je od E. Feigenbauma: „ZS (původně „expertní systémy“) jsou počítačové programy, simulující rozhodovací činnost experta při řešení složitých úloh a využívající vhodně zakódovaných, explicitně vyjádřených znalostí, převzatých od experta, s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni experta.“

První ZS se začaly objevovat (obdobný výpis historie je např. v [1, 3, 9]) v první polovině 70. let 20. století. Za první ZS je označován systém „Dendral“, který pomáhal k identifikaci chemických sloučenin. Dalšími důležitými milníky v této vědní oblasti jsou systémy „PROSPECTOR“ a „MYCIN“. První ze jmenovaných dokázal na základě geografických údajů rozhodnout, zda je v dané oblasti vhodné provádět zemní vrty. A druhý jmenovaný byl využit ve zdravotnictví pro rychlé určení bakteriálních infekcí. Různé mechanismy a zpracování neurčitosti, vyskytující se v těchto systémech, jsou dodnes používány v různých mutacích (jako součásti) novějších ZS.

V mnohé literatuře je možné se setkat také s pojmem „expertní systém“ (dále jen ES). Pojem ES lze v dnešní době považovat za ekvivalent k termínu ZS. Ovšem dříve byl ES brán jako specifitější pojem, jelikož ES (podle dřívějšího pojetí) by měl využívat výlučně znalostí získaných od experta a několik dalších specializovaných mechanismů (např. vysvětlovací mechanismus). Ovšem tato specifika ES postupem času a rozvojem technologií přestala platit a dnes lze všechny systémy (spadající do dané kategorie) nazvat ZS.

2.2.1 Struktura ZS

Základními částmi každého ZS jsou „báze znalostí“, „báze faktů“ a „inferenční mechanismus“. Bez těchto částí nemůže existovat žádný ZS (popis jednotlivých částí – viz níže). Ovšem postupem času jsou vyžadovány i různé další rozšiřující funkce systému, tudíž jsou navíc obvykle používány následující komponenty:

- *I/O rozhraní* – zajišťuje komunikaci s uživatelem, získává data z měřicích přístrojů atd.

- *Vysvětlovací modul* – slouží pro znázornění postupu ZS při odvozování, dále může poskytnout popis, jak by postupoval, pokud by vstupy byly jiné.
- *Modul pro akvizici znalostí* – zjišťuje ze zdrojů (např. z odborných textů či od experta) znalosti, které následně reprezentuje v dané bázi znalostí.

Báze znalostí

Jedná se o datovou strukturu, do které jsou ukládány znalosti (a řešení problémů) z určité specifické oblasti. Ukládají se nejen poznatky exaktně dokázané, ale i znalosti, které získal expert během své praxe (ovšem tyto poznatky jsou zatíženy neurčitostí). Znalosti experta, získané praxí (nejsou exaktně dokázané, jedná se spíše o „dobré mínění“), se označují jako tzv. „heuristiky“. Podle [6] je báze znalostí kvůli již zmíněným „heuristicám“ zatížena tzv. „nejistotou v bázi znalostí“.

Při vytváření báze znalostí je kladen důraz na „modularitu“, která zajišťuje přehlednost, snadné úpravy a jednoduché rozšiřování stávající báze (uvedeno v [6, 9] atd.).

Dále (podle [9]) je kladen důraz na tzv. „strukturalizaci“, která slouží k rychlému vyhledávání pojmů, tudíž i k rychlému přechodu od obecných znalostí k úzce specializovaným či naopak.

Báze faktů

Báze se využívá pro ukládání konkrétních dat k danému problému. Tyto údaje jsou inferenčním mechanismem (využívající bázi znalostí) různě modifikovány tak, aby ZS mohl poskytnout odpovídající závěr.

Data do této datové struktury mohou být získávána např. z měřicích zařízení, od uživatele (na základě procesu „konzultace“) atd.

Inferenční mechanismus

Jedná se o programový modul (složený z několika komponent), který umožňuje ZS napodobovat uvažování lidského experta. Tento mechanismus funguje především na základě interakce s bázemi znalostí a faktů.

Obvykle je založen na určité strategii prohledávání báze znalostí či na inferenčním pravidle pro odvozování nových údajů ze stávajících znalostí. Pro odvozování nových údajů se ve většině případů využívají následující inferenční metody (popis vybraných metod je získán z [1, 3, 9]):

- *Indukce* – inferenční mechanismus postupuje od specifitějších poznatků k obecnějším.
- *Dedukce* – konečný závěr je odvozen z předpokladů.
- *Abdukce* – odvozování od již známého závěru k možným předpokladům.
- *Generování a testování* – poznatky jsou získávány metodou „pokusů a omylů“.
- *Heuristiky* – odvozování na základě „zkušeností“.
- *Analogie* – vytváření odpovědí na základě podoby s jiným (již známým) stavem.
- *Absence* – pokud při odvozování chybí daná specifická znalost, pak je nahrazena obecnější.
- *Nemonotónní inference* – při zjištění nových poznatků přestávají platit původní a expertiza se provádí znovu.

Jak již bylo zmíněno v části „2.2.1 Báze znalostí“, do báze znalostí se vkládají (mimo jiné) tzv. „heuristiky“. Tyto heuristiky (a nejen ony) zatěžují bázi znalostí neurčitostí, kterou musí umět inferenční mechanismus zpracovat. K tomu jsou využívány různé metody a strategie založené např. na fuzzy množinách, na teorii pravděpodobnosti apod.

2.2.2 Typologie ZS

Zařadit jednotlivé typy ZS do určité kategorie je možné podle mnoha kritérií. Mezi základní klasifikační kritérium patří dělení podle charakteru řešené úlohy. Sem spadají následující systémy (podle [9]):

- *Diagnostické* – systémy, spadající do této kategorie, vždy disponují znalostmi z určité problematiky a snaží se získat data o aktuální situaci např. od uživatele, z měřicích přístrojů atd. Na základě těchto dat diagnostické ZS zjišťují, která z konečné množiny hypotéz nejvíce odpovídá aktuálnímu problému.
- *Plánovací* – tyto ZS řeší takové problémy, u kterých je dopředu znám počáteční a koncový stav. A jejich úkolem je, na základě získaných dat (od uživatele, z měřicích přístrojů, . . .), nalézt konkrétní cestu, jak dosáhnout požadovaného cílového stavu.
- *Hybridní* – tyto systémy kombinují postupy a metody využívané v diagnostických a plánovacích systémech s dalšími metodami, jako jsou evoluční strategie (a algoritmy) či neuronové sítě.

Dalším klasifikačním kritériem je dělení ZS na základě báze znalostí na (podle [3]):

- *Prázdné* – jedná se o systémy, které disponují inferenčním mechanismem (a dalšími funkcemi pro správný chod ZS), ovšem mají prázdné báze znalostí a faktů.
- *Problémově orientované* – tyto systémy jsou již orientovány na konkrétní problematiku a k tomu mají uzpůsobené báze znalostí a faktů.

2.3 Produkční ZS

2.3.1 Produkční pravidla

Produkční systémy jsou kategorií ZS, která se od logických systémů liší „nemonotónním přemítáním“ a možností uvažování za neurčitosti. Název je

odvozen od tzv. „produkčních pravidel“, prostřednictvím nichž jsou obvykle vytvářeny báze znalostí těchto systémů.

Produkční pravidla mají jednotný tvar, skládající se z levé a pravé části. Obecný tvar pravidla je:

```
IF levá_část THEN pravá_část
```

Levá část každého pravidla obsahuje tzv. „vzory“. Jedná se o množinu podmínek, která je porovnávána (inferenčním mechanismem) s bází faktů. Tato část je nazývána „antecedent“, „situace“, „podmínková část“ anebo (jak již bylo naznačeno) „část vzorů“. V tomto segmentu se často vyskytují logické operátory (např. AND, OR, ...). Ovšem negace (běžně značená jako NOT) zde má odlišný význam, a to takový, že daný vzor (jemuž předchází negace) nesmí mít odpovídající protějšek v bázi faktů.

Pravá část (tedy segment za THEN) je tzv. „konsekvent“, který je obvykle závěrem či akcí, provádějící úpravy v bázi faktů apod. Může se zde nacházet logický symbol AND či klasické větvení (známé z jiných programovacích jazyků, jako jsou C, JAVA či C#), které je znázorněno níže:

```
IF levá_část THEN pravá_část_1 ELSE pravá_část_2
```

2.3.2 Základní způsoby implementace produkčních systémů

Systémy porovnávání se vzorem

Tyto systémy jsou založeny na jednoduchém principu porovnávání levých částí (částí vzorů) pravidel s bázemi faktů. A v případě kompletní shody je možné pravidlo aplikovat. Systémy porovnávání se vzorem jsou vhodné pro plánovací a návrhové úlohy.

Systémy implementované jako inferenční síť

Inferenční síť je další způsob implementace produkčních systémů. Touto sítí je graf, jehož uzly odpovídají jednotlivým faktům a hrany odpovídají pravi-

dlům. Síť se obvykle používá u klasifikačních a diagnostických úloh.

2.3.3 Inferenční přístupy v produkčních systémech

Modus ponens

Základním prvkem inferenčních metod v produkčních systémech je klasické pravidlo modus ponens. Pravidlo říká: „Jestliže platí předpoklad E a pravidlo $E \rightarrow H$, pak platí i závěr H .“ Zápis pravidla viz níže (definice a zápis převzat z [8]):

$$\frac{E, E \rightarrow H}{H} \quad (2.1)$$

Modus tollens

Pravidlo také patří k základním kamenům inferenčních metod. Modus tollens je opakem pravidla modus ponens a říká: „Jestliže platí pravidlo $E \rightarrow H$ a neplatí závěr H , neplatí ani předpoklad E .“ Podstata pravidla viz níže (definice a zápis převzat z [8]):

$$\frac{\neg H, E \rightarrow H}{\neg E} \quad (2.2)$$

2.3.4 Metoda zpětného řetězení

ZS, postupující metodou zpětného řetězení, nejprve vybere možný závěr a poté se snaží, na základě získaných faktů (od uživatele, z měřicích přístrojů, ...) a znalostní báze, potvrdit vybranou hypotézu. Z toho vyplývá, že se prochází pravidla v bázi znalostí od závěrů k předpokladům. Pořadí vyhodnocování pravidel je dáno posloupností zápisu pravidel a ručně zadanými prioritami (jednotlivých pravidel).

Systém, využívající metodu zpětného řetězení, je vhodný zejména pro diagnostické úlohy, kde uživatel odpovídá na pokládané otázky a ZS na jejich základě vyhodnotí hypotézu řešení daného problému. Dále se tento postup často využívá pro řešení klasifikačních problémů.

Algoritmus zpětného řetězení (převzatý z [3]):

1. Utvoř zásobník a naplň jej všemi koncovými cíli.
2. Shromáždí všechna pravidla schopná splnit cíl na vrcholu zásobníku. Je-li zásobník prázdný, pak konec.
3. Zkoumej postupně všechna pravidla z předchozího kroku.
 - (a) Jsou-li všechny předpoklady splněny, pak odvod' závěr (proved' pravidlo). Jestliže zkoumaný cíl byl koncový, pak jej odstraň ze zásobníku a vrat' se na krok 2. Jestliže to byl podcíl (dílčí cíl), odstraň jej ze zásobníku a vrat' se ke zpracování předchozího pravidla, které bylo dočasně odloženo.
 - (b) Jestliže fakty nalezené v bázi faktů nesplňují předpoklady pravidla, je zkoumání pravidla ukončeno.
 - (c) Jestliže pro některý parametr předpokladu chybí hodnota v bázi faktů, zjišť'uje se, zda existuje pravidlo, z něhož by mohla být tato hodnota odvozena. Pokud ano, parametr se vloží do zásobníku jako podcíl, zkoumané pravidlo se dočasně odloží a přejde se na krok 2. V opačném případě se tato hodnota zjistí od uživatele a pokračuje se v kroku 3.a) zkoumáním dalšího předpokladu.
4. Jestliže prostřednictvím žádného ze zkoumaných pravidel nebylo možné odvodit hodnotu důsledku, pak daný cíl zůstává neurčen. Odstraní se ze zásobníku a pokračuje se krokem 2.

2.3.5 Metoda dopředného řetězení

Systém, využívající metodu dopředného řetězení (někdy také „přímé řetězení“), porovnává aktuální bázi faktů s levými stranami pravidel (tj. „části vzorů“). Takto vybírá použitelná pravidla, prostřednictvím nichž postupuje k závěrům. Tento přístup je zejména vhodný pro řešení plánovacích, rozvrhových a konfiguračních problémů.

Algoritmus metody se skládá z následujících kroků (podle [8]):

1. *Porovnání* – prosté porovnávání levých stran produkčních pravidel s bázi faktů a získání množiny odpovídajících pravidel.

2. *Řešení konfliktů* – při nalezení více než jednoho vyhovujícího pravidla se řeší, které pravidlo má přednost. K tomu se využívá několik postupů, z nichž základní jsou popsány níže (v části „Přístupy pro řešení konfliktů“).
3. *Exekuce pravidla* – u pravidla, vybraného v předchozím kroku, je zpracována pravá část.

Přístupy pro řešení konfliktů

Mezi základní strategie řešení konfliktů patří (podle [3]):

- *Prohledávání do hloubky* – přednostně jsou vybírána produkční pravidla, jejichž vzory odpovídají novějším položkám v bázi faktů.
- *Prohledávání do šířky* – přednost je dávána pravidlům, která mají vzory odpovídající starším položkám v bázi faktů.
- *Preference složitosti* – vybírána jsou pravidla s největším počtem odpovídajících vzorů.
- *Preference jednoduchosti* – preferována jsou produkční pravidla s obecnější vzorovou částí.

2.3.6 RETE algoritmus

Reálně používaný produkční systém se obvykle skládá ze stovek (až tisíců) pravidel. Kvůli použitelnosti takového systému je nezbytný efektivní algoritmus, zajišťující rychlý výběr pravidla. Pro tyto účely existuje několik řešení, např. RETE algoritmus, dále pak LEAPS či TREAT. Ovšem RETE algoritmus je nejpoužívanější a v nástroji CLIPS je také využíván (v tomto nástroji je vytvořena praktická část tohoto projektu), tudíž je následující text věnován právě jemu.

RETE algoritmus slouží k rychlému výběru pravidla v produkčním systému. Jedná se o propracovanější postup, než je výběr pravidla přístupem dopředného řetězení. V případě dopředného řetězení se po každém provedení pravidla porovnává báze faktů se všemi vzory („vzor“ je podmínka umístěná v levé části pravidla) každého produkčního pravidla (daného systému).

Ovšem vykonáním již zmíněného pravidla se obvykle báze faktů upraví jen minimálně, tudíž dochází k mnoha redundantním operacím.

Pro zamezení právě zmíněného jevu RETE algoritmus využívá síťovou architekturu (RETE je latinský výraz pro „sít“), skládající se z hran a uzlů. Každá hrana spojuje dva uzly (počáteční a koncový uzel) a reprezentuje vztahy proměnných počátečního uzlu. Uzly jsou tří typů: startovní uzel, podmínkový uzel (existuje vždy jeden uzel pro jednu podmínku) a uzel pro konjunkci podmínek (pro každou konjunkci je jeden uzel). Každý uzel má určitý počet výstupních hran vedoucích do dalších uzlů. Pokud však daný uzel výstupní hrany nemá, pak odpovídá pravidlu, které se (za předpokladu splnění všech podmínek) vykoná.

Nově přidaný fakt do báze faktů je opatřen příznakem (tzv. „token“). Tento příznak se vkládá do startovního uzlu, odkud se šíří dále do sítě, ale jen těmi hranami, ve kterých jeho argumenty splňují vztah. A pokud je některý z faktů odstraněn, pak se tento stav šíří po síti stejným způsobem, jen jsou jednotlivé kopie mazány.

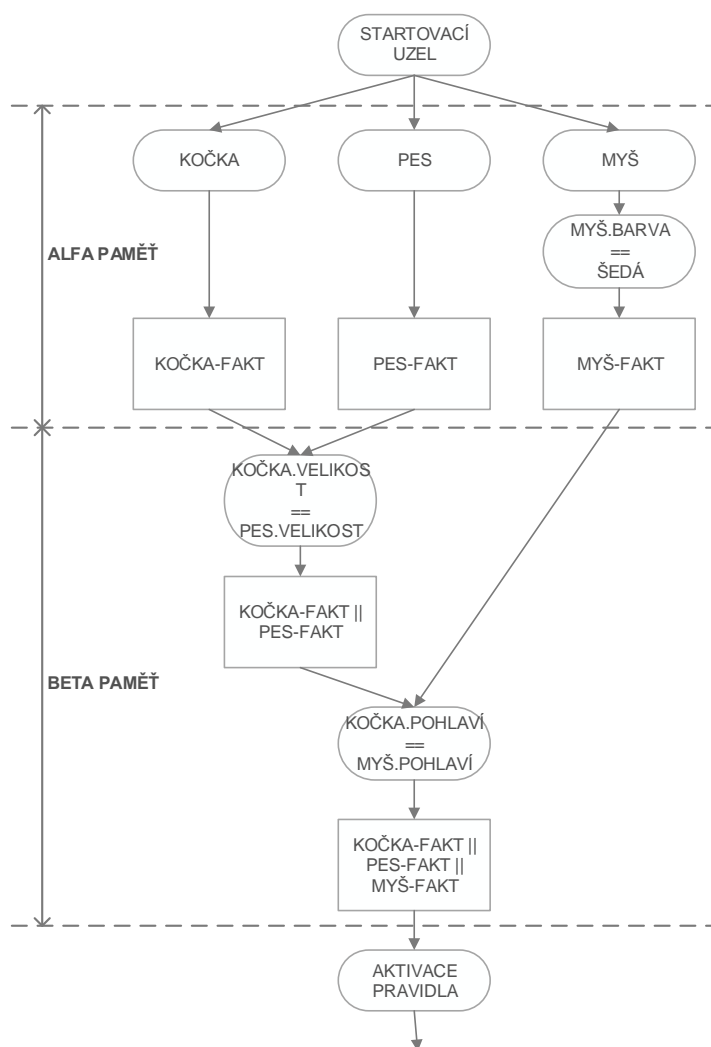
Jednotlivé uzly RETE sítě se ukládají do tzv. „alfa“ a „beta“ paměti. V „alfa“ paměti jsou uzly (označované $A_{i,1}; A_{i,2} \dots A_{i,n}$) vytvářené pro každý vzor každého pravidla. A v „beta“ paměti jsou „propojovací“ uzly (značené $B_{i,1}; B_{i,2} \dots B_{i,n}$) vytvořené pro „alfa“ uzly každého pravidla. Indexy „i“ představují dané pravidlo. Jednotlivé uzly v „beta“ paměti jsou vytvářeny následujícím způsobem:

- pro $B_{i,1}$ – levý vstup je z $A_{i,1}$ a pravý vstup je z $A_{i,2}$
- pro $B_{i,j}$, kde $j > 1$ – levý vstup je z $B_{i,j-1}$ a pravý vstup je z $A_{i,j+1}$

Příklad RETE algoritmu

Pravidlo (uvedené níže) má v síťové struktuře RETE algoritmu podobu – viz obrázek 2.1 (na obrázku se v hranatých obdélnících nacházejí fakty, daným uzlem akceptovatelné. Příklad je inspirován z [11]).

```
(defrule příklad-pravidla
  (KOČKA (VELIKOST ?x) (POHLAVÍ ?y))
  (PES (VELIKOST ?x))
  (MYŠ (BARVA ŠEDÁ) (POHLAVÍ ?y))
  => ...
)
```



Obrázek 2.1: Příklad RETE algoritmu

3 Prostředí CLIPS

3.1 Popis prostředí

Tato kapitola pojednává o speciálním vývojovém prostředí CLIPS, které je určeno pro tvorbu ZS. Kapitola se zde nachází, jelikož praktická část této BP (kromě prezentací) je vytvořena právě v tomto vývojovém prostředí. Kapitola je pouze krátkým úvodem do problematiky (již zmíněného nástroje) a je pojata tak, aby i laik mohl rychle porozumět zdrojovému kódu, který je napsaný v CLIPS.

CLIPS je vývojové prostředí, které je vytvářeno již od roku 1985 (původně společností NASA). Tento nástroj je šířen pod tzv. „public domain licenci“, která umožňuje volné využití jak pro soukromé, tak komerční účely.

Toto prostředí je neustále vyvíjeno a postupně je rozšiřováno i jeho pole působnosti. V dnešní době již existuje mnoho mutací tohoto prostředí, které (na rozdíl od původní verze napsané v programovacím jazyce C) jsou implementované např. v jazyce JAVA, čímž se CLIPS výrazně přiblížil skupině vývojářů, využívající právě zmíněný programovací jazyk. Konkrétní mutací (vytvořená v jazyce JAVA) je kupř. JESS.

Jazyk, používaný v tomto prostředí, je (až na několik rozdílů, mezi nimiž je např. rozlišování velkých a malých písmen) velmi podobný jazyku LISP.

CLIPS využívá integrovaný inferenční mechanismus, který je založen na dopředném řetězení a RETE algoritmu (oba přístupy jsou popsány výše). To zaručuje velmi rychlé prohledávání stavového prostoru a následné usuzování relevantních závěrů.

3.2 Práce s fakty

Ukládání faktů do báze je v CLIPS velmi všestranné, jelikož se jednotlivé fakty ukládají do tzv. „polí“. Jeden fakt může zabírat jedno nebo více polí. Díky této vlastnosti se mohou fakty pojmenovávat (ale není to nutné) a různě s nimi manipulovat.

Datové typy jednotlivých polí mohou být „float“, „integer“, „string“, „symbol“ atd. Kde „string“ (neboli znakový řetězec) musí být uveden v uvozovkách a „symbol“ se musí skládat z ASCII znaků, a to pouze tisknutelných.

V bázi faktů jsou jednotlivé záznamy přístupné pod indexy (jedná se o kladné celé číslo, kde hodnota 0 je vyhrazena pro interní fakt – tzv. „initial-fact“). Jednotlivé fakty se dají zobrazit příkazem „(facts)“, přidání nového faktu se provádí funkcí „assert“ a odstranění prostřednictvím „retract“. Níže je uveden příklad přidání faktu, který zaujímá více „polí“.

```
(assert (nazev-faktu hodnota-1 ... hodnota-N))
```

3.3 Pravidla v CLIPS

Jednotlivá pravidla se skládají (jak již bylo zmíněno výše) z levých segmentů (tedy částí vzorů) a pravých segmentů (tedy částí akcí, které obvykle pozměňují bázi faktů). Jednotlivá pravidla mohou být doplněna komentáři a prioritou. Priorita se přiřazuje funkcí „declare“ a může být v intervalu <-10 000; 10 000). V případě neuvedení funkce „declare“ je hodnota priority implicitně nastavena na 0. Obecný zápis pravidla (včetně nepovinného komentáře):

```
(defrule nazev-pravidla "Komentář nemusí být uveden."
  (declare (salience 30))
  (vzor-1)
  ...
  (vzor-N)
=>
  (akce-1)
  ...
  (akce-N)
)
```

Levé strany pravidel se mohou skládat z jednoho nebo více vzorů (vzor je „podmínkový fakt“). Se vzory lze pracovat prostřednictvím logických operátorů, jako jsou AND, OR atd. Ovšem negace (NOT) je zde chápána ve významu, že daný vzor (jemuž předchází negace) nesmí mít protějšek v bázi faktů.

Pro lepší zpracování položek pravidel se využívají tzv. „wildcard“ (lze je považovat za zástupné symboly). Zapisují se prostřednictvím jednoho otazníku a symbolického názvu (v případě, že mají zastupovat právě jedno pole ve vzorech). A nebo zápisem „\$?“ a názvu, pokud mají zastupovat 0 nebo více polí jednotlivých faktů.

3.4 Inferenční mechanismus

Mechanismus funguje (jak již bylo zmíněno) na základě dopředného zřetězení (a RETE algoritmu). Tudíž postupně porovnává podmínkové části jednotlivých pravidel (tedy části vzorů) s bázi faktů, a pokud jsou nalezeny shody, pak jsou daná pravidla zařazena (s určitými prioritami) do tzv. „agendy“.

Agenda je zásobníková datová struktura, do které jsou ukládána pravidla (společně s jejich prioritami), která je nutné co nejdříve zpracovat. CLIPS nahlédne vždy po dokončení aktuální úlohy do agendy. A pokud je v něm uložen záznam, pak je přednostně proveden. Pokud je záznamů více, pak je vybíráno pravidlo s nejvyšší prioritou (pokud existuje více pravidel se shodnými prioritami, pak se aplikuje standardní algoritmus zásobníkové datové struktury, tedy LIFO – poslední vložený záznam do agendy je odebrán jako první).

Po vybrání a provedení pravidla je také tento záznam z agendy odebrán. A pokud právě aplikované pravidlo změní bázi faktů tak, že vzorové části pravidel (uložených v agendě) již neodpovídají, pak jsou tato nepoužitá pravidla z agendy odstraněna.

Jak již bylo zmíněno, základem inference v CLIPS je dopředné řetězení, jehož popis je v 2.3.5. Tento přístup vždy používá určitou strategii, prostřednictvím níž určuje pořadí aplikace použitelných pravidel, která mají shodnou prioritu. V CLIPS existují následující strategie:

- *Prohledávání do hloubky* – preference pravidel, jejichž vzory odpovídají novějším položkám v bázi faktů.
- *Prohledávání do šířky* – přednost mají pravidla, která mají vzory odpovídající starším položkám v bázi faktů.
- *Jednoduchosti* – preferována jsou produkční pravidla s obecnější vzorovou částí.

- *Složitosti* – pravidla s největším počtem odpovídajících vzorů mají přednost.
- *LEX* – jedná se o kombinaci strategií prohledávání do hloubky a strategie složitosti. Tudíž jsou preferována pravidla, jejichž části vzorů odpovídají novějším faktům (v bázi faktů). A pokud nastane shoda „aktuálnosti“ i podle tohoto řazení, pak je dáována přednost pravidlům s větším počtem vzorů.
- *MEA* – pravidla jsou řazena podle aktuálnosti prvních vzorů. V případě konfliktu je aplikována strategie LEX.
- *Náhodnosti* – produkční pravidla jsou řazena náhodně.

3.5 Další možnosti v CLIPS

Při vytváření ZS je možné využívat interních funkcí, jako jsou I/O metody či matematické funkce. Dále je možné definovat funkce nové, k tomu se využívá deklarace „*deffunction*“. Nově definované funkce mohou, kromě používání již zmíněných interních metod, využívat i kód napsaný v jazyce C. Jednotlivé metody mohou mít vstupní parametry (stejně jako v jiných programovacích jazycích) a vracet hodnoty. Návrátovým hodnotám funkce musí předcházet „*return*“ a výstup bývá standardního datového typu (např. „*integer*“).

Prostředí CLIPS disponuje proměnnými, a to lokálními i globálními. Proměnné jsou zde chápány obdobně, jako v jiných programovacích jazycích. Tudíž globální proměnné jsou přístupné v celém ZS a lokální jsou dostupné v rámci daného pravidla. K přiřazení se nejčastěji používá funkce „*bind*“. Globální proměnné je navíc možné definovat funkcí „*defglobal*“. Příklady přiřazení viz níže (v příkladu je uvedena funkce „*read*“, jedná se o interní funkci, která zajišťuje získání vstupu od uživatele).

```
(bind ?lokalni-promenna (read))
```

```
(defglobal ?*glob-promenna-a* = 7.6)
```

```
(bind ?*glob-promenna-b* (/ 81 9))
```

4 ZS pro frézku Jeřábek SP410PRO

Jedním z úkolů této BP je implementace vybraných metod reprezentace znalostí. Během zpracovávání této části byl vytvořen „Diagnostický ZS pro srovnávací a tloušťkovací frézku Jeřábek SP410PRO“. Jedná se o produkční systém, jehož úkolem je identifikace závad a poskytování řešení problémů frézky Jeřábek SP410PRO (popř. podobných frézek ze stejné kategorie).

V této kapitole se nachází rozbor implementace již zmíněného ZS, popis jednotlivých částí, způsob fungování systému atd. Ovšem, aby byl popis úplný, je nejprve nutné objasnit, co to vlastně již zmíněná frézka je.

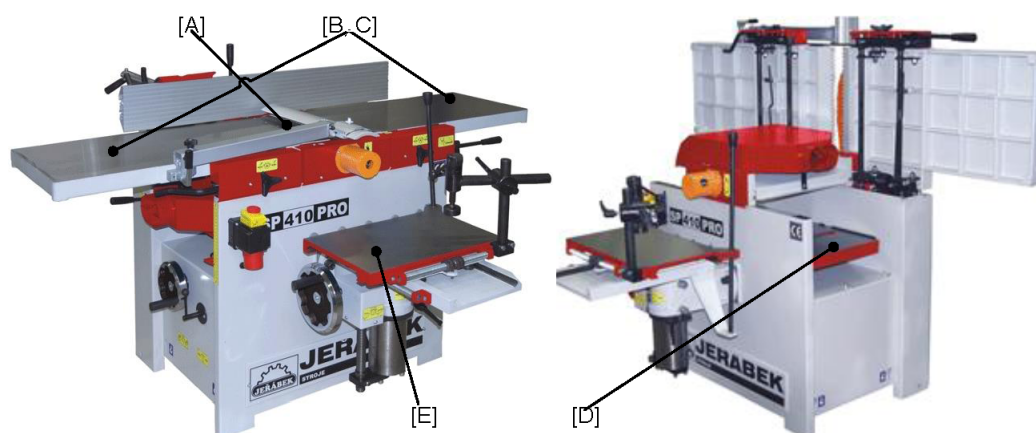
4.1 Srovnávací a tloušťkovací frézka Jeřábek SP410PRO

Frézka (znázorněná na obr. 4.1) je kombinovaný truhlářský stroj určený k vyrábění hranolů ze dřeva. Frézka plní dvě funkce, a to funkci „srovnávací“ a funkci „tloušťkovací“ (obě jsou nastíněny níže). Stroj je vybaven hnacím motorem, litinovými stoly a neposlední v řadě nožovou hřídelí (na obr. 4.1 značená jako [A]) atd. Nožová hřídel je válec osazený čtyřmi hoblovacími noži, sloužící k úběru materiálu.

Srovnávací funkce obvykle slouží pro dosažení rovné plochy na obrobku (popř. několika rovných ploch, které svírají určitý úhel). Tato funkce se provádí na dvou litinových stolech (na obr. 4.1 to jsou [B, C]).

Tloušťkovací funkce slouží, jak již název napovídá, pro získání požadované tloušťky hranolu. Během tloušťkování je opracováván materiál posouván pod nožovou hřídelí, tudíž se tato funkce provádí na přítlačném litinovém stole (na obr. 4.1 značený jako [D]).

Na obr. 4.1 je společně s frézku vyobrazeno přídatné zařízení (označené jako [E]). Jedná se o tzv. „dlabačku“, která není předmětem zájmu ZS (vytvořeného v rámci této BP). Obr. 4.1 je získán z www.stroje-atol.cz.



Obrázek 4.1: Srovnávací a tloušťkovací frézka Jeřábek SP410PRO

4.2 Implementace ZS

Systém je vytvořen ve speciálním vývojovém prostředí CLIPS (stručný popis prostředí se nachází v kapitole 3). Tento nástroj disponuje mnoha komponentami pro vývoj ZS, z nichž je důležité zmínit, že inferenční mechanismus funguje na základě dopředného řetězení a RETE algoritmu (popis obou pojmů viz sekce 2.3.5). CLIPS dále disponuje mnoha interními funkcemi, jako je načítání vstupů z klávesnice apod. A neposlední v řadě CLIPS umožňuje implementaci funkcí v programovacím jazyce C.

ZS (vytvořený v rámci praktické části této BP) je produkční systém, pracující pouze s určitými informacemi (tzn., že uživatel musí vždy exaktně odpovědět). Systém je rozdělen do deseti souborů, které obsahují celkem 183 produkčních pravidel, tři funkce a čtyři šablony. Souhrnný popis pravidel, nacházejících se v jednotlivých souborech, je uveden níže.

4.2.1 Soubor příkazů „zs-start.bat“

Při spuštění ZS (návod na spuštění systému – viz příloha A) je nejprve načten soubor „zs-start.bat“. V souboru se nachází sada příkazů zajišťujících úpravu způsobu prohledávání stavového prostoru systému, specifikaci přístupu nakládání s fakty apod. Z toho vyplývá, že při spuštění ZS je nejprve upraveno nastavení prostředí CLIPS do požadovaného stavu (důležité příkazy jsou zmíněny níže) a poté jsou dalšími instrukcemi postupně načítány

zbylé části ZS, obsahující pravidla báze znalostí, definice vícepolevých faktů atd. Nakonec je načtena počáteční báze faktů a poté je ZS spuštěn.

Důležitá nastavení nástroje CLIPS:

- (*set-strategy depth*) – při výběru pravidla užívat strategii „prohledávání do hloubky“.
- (*set-salience-evaluation when-defined*) – zohledňovat při výběru pravidla ručně zadané priority.
- (*set-fact-duplication FALSE*) – v bázi faktů se nesmí nacházet duplikát žádného záznamu.

4.2.2 Soubor pravidel „operace-s-fakty.clp“

Produkční pravidla, nacházející se v tomto souboru, zajišťují pohyb v hierarchii otázek identifikace problémů (hierarchie otázek – viz příloha B.1). Dále se zde nacházejí funkce, zajišťující např. získání korektního vstupu od uživatele, nebo definice šablon, sloužící pro správný chod ZS.

4.2.3 Soubor pravidel „univerzální-poznatky.clp“

V tomto souboru jsou uloženy tzv. „univerzální“ řešení (či znalosti). Jedná se o řešení, která jsou využita u více než jedné hypotézy.

4.2.4 Soubor „informativní-pravidla.clp“

V souboru se nachází komunikace s uživatelem, která slouží ke zjištění informací o dané problémové frézce. Mezi zjišťované informace patří, zda se jedná o frézku Jeřábek SP410PRO nebo alespoň jestli spadá do stejné kategorie atd.

4.2.5 Soubor pravidel „problematika-zapnutí.clp“

Pravidla v tomto souboru slouží pro komunikaci s uživatelem a vygenerování řešení daných problémů. Problematika (zde řešená) se zabývá zapnutím frézky a záležitostmi spojenými se zapnutím stroje.

4.2.6 Soubor pravidel „problematika-vypnutí.clp“

V souboru se vyskytují produkční pravidla, zajišťující komunikaci s uživatelem (pro identifikaci problémů) a obsahující řešení daných problémů. Řešenou problematikou je vypínání frézky a problémy s vypínáním spojené.

4.2.7 Soubor pravidel „problematika-srovnávání.clp“

V tomto souboru se nacházejí pravidla, která slouží pro komunikaci s uživatelem a výpis adekvátních řešení potíží se srovnávací funkcí frézky. Mezi řešenou problematiku patří nejen problémy se srovnáváním, ale také s přidruženými komponentami již zmíněné funkce.

4.2.8 Soubor pravidel „problematika-tloušťkování.clp“

Zde se nacházejí pravidla, sloužící pro komunikaci a získání řešení z problematiky, týkající se tloušťkovací funkce frézky. Mezi tyto problémy patří např. nekvalitní obrobení či nefunkční posuvné válce.

4.2.9 Soubor pravidel „problematika-stav-frézky.clp“

Produkční pravidla, nacházející se v tomto souboru, zajišťují komunikaci s uživatelem a poskytují řešení potíží z problémové domény týkající se fyzického stavu frézky. Do této oblasti spadají např. mechanické poruchy.

4.2.10 Soubor pravidel „počáteční-fakta.fct“

V tomto souboru jsou obsažena fakta načítaná vždy při spuštění či restartu ZS. Získávána jsou funkcí „load-facts“, která je volána v souborech „zs-start.bat“ a „operace-s-fakty.clp“.

4.3 Průběh identifikace problémů

Jak již bylo zmíněno, inferenční mechanismus nástroje CLIPS funguje na základě dopředného řetězení. To znamená, že jsou nejprve nalezena všechna pravidla, jejichž podmínková část je splněna, a poté je na základě určité strategie vybráno pravidlo, které je dále aplikováno. A pokud žádné pravidlo nemá splněnou podmínkovou část, pak inferenční mechanismus výběr ukončí. Podrobnější popis dopředného řetězení – viz sekce 2.3.5.

V případě ZS, vytvořeného v rámci této BP, je již při zavádění (úprava nastavení CLIPS atd.) upravena báze faktů tak, že ihned po spuštění je nejprve zjištěno, o jakou frézku se jedná. A pokud je problematický stroj Jeřábek SP410PRO (nebo je alespoň ze stejné kategorie), pak je báze faktů upravena takovým způsobem, aby bylo inferenčním mechanismem vybráno pravidlo zajišťující posun ZS k identifikaci problémů.

Identifikace problémů se skládá z hierarchie otázek. Ta je znázorněná (včetně všech řešení problémů) v rozhodovacím diagramu, který je uveden v příloze B.1. Výběr pravidel při identifikaci je ovlivněn dvěma fakty. Prvním faktem je „(aktuální-úroveň (hodnota N))“, kde N znázorňuje nezáporné celé číslo, které vyjadřuje úroveň zanoření ve výše zmíněném rozhodovacím diagramu. Hodnota 0 (čísla N) odpovídá uzlu, který vyjadřuje inicializaci ZS. Druhým faktem je „(hypotéza (popis A) B)“, kde A znázorňuje odpovědi na již položené otázky a B zastupuje další pole šablony „hypotéza“, která slouží pro výpis řešení problémů – viz sekce 4.5.

Konkrétní příklad pravidla identifikace problémů (ovlivněný výše zmíněnými fakty):

```
(defrule iden-uroven-2-opacny-smer
  (aktualni-uroven (hodnota 2))
  (hypoteza (popis zapnuti))
  (not (nezadouci opacny-smer))
=>
  (printout t crlf crlf crlf crlf
    "      Nozova hridel se otaci opacnym smerem?")
  (assert (odpoved-identifikace
    (ziskej-odpoved ano ne zpet ukoncit) opacny-smer))
)
```

V podmínkové části příkladu (uvedeného výše) se nachází podmínka „(not (nežádoucí opačný-směr))“, tento vzor je uveden u pravidel s identifikačními otázkami, které jsou pouze jednou možností z aktuální množiny otázek. Z toho vyplývá, že se jedná o pomocný fakt, zamezující nechtěnému chování ZS při změně stavu báze faktů.

4.4 Uživatelská rozhodnutí

Pokud uživatel odpoví na některou z otázek identifikace problémů, pak dojde k inkrementaci hodnoty N u faktu „(aktuální-úroveň (hodnota N))“ a přidání identifikační značky otázky do šablony „hypotéza“. V některých případech je přidána i uživatelská odpověď.

Identifikační značka otázky je vždy posledním polem faktu pojmenovaného „odpověď-identifikace“ nebo „odpověď-rozhodnutí“. Ve výše uvedeném příkladu pravidla je značkou výraz „opačný-směr“. Fakt „odpověď-identifikace“ je využíván v případě, kdy daná otázka je pouze jednou možností z aktuální množiny otázek. Fakt „odpověď-rozhodnutí“ je používán, pokud lze na danou otázku odpovědět (kromě ukončení ZS či vrácení posledního uživatelského rozhodnutí) pouze „ano“ nebo „ne“.

Získání korektní odpovědi od uživatele zajišťuje funkce „získej-odpověď“, jejíž parametry tvoří množina přijatelných odpovědí, z nichž musí uživatel jednu položku zadat.

Pokud chce uživatel vrátit předešlé rozhodnutí (či několik odpovědí), musí zadat výraz „zpět“. Zadáním výrazu je k dekrementována hodnota N u faktu

„(aktuální-úroveň (hodnota N))“ a poté je ještě odebrána poslední identifikační značka pravidla, popř. včetně přidružené uživatelské odpovědi.

4.5 Výpis řešení problémů

V případě, že uživatel zodpoví všechny otázky dané hypotézy (hierarchie otázek, včetně všech řešení problémů, je znázorněna v příloze B.1), dojde k úpravě báze faktů a následnému výpisu řešení problému.

Během úpravy báze faktů je především modifikováno pole „poznatek“ faktů „hypotéza“. Do tohoto pole je uloženo řešení problému, které je posléze zpracováno pravidlem (a nejen ním) „výpis-poznatku“. Pravidlo se nachází v souboru „operace-s-fakty“.

U některých řešení je nutné zmínit, že by daný úkon měl provést odborník (např. elektrikář či servis). Tyto poznámky jsou automaticky generovány, pokud je v pravidlu s řešením problému modifikován některý ze slotů faktů „odborná-pomoc“. V případě modifikace některého slotu je nejprve vypsáno řešení problému (způsobem popsaným výše) a poté je příslušným pravidlem vypsána poznámka k již zmíněnému řešení. Kupř. když je modifikován slot „elektrikář“ faktů „odborná-pomoc“, pak je aplikováno pravidlo „potřeba-pomoci-elektrikáře“, které se nachází v souboru „operace-s-fakty“.

4.6 Výhody a nevýhody implementace

Součástí zadání této BP je také zmínění hlavních výhod a nevýhod implementace programového řešení. Celkový souhrn kladů a záporů je umístěný v sekci 9.2.

5 Návrh prezentací o reprezentaci znalostí

Jak již bylo zmíněno v úvodu, v původním zadání BP je uveden návrh či modifikace webových stránek, které mají pojednávat o podoboru umělé inteligence – o reprezentaci znalostí. Ovšem od tohoto bylo během vytváření BP upuštěno a budou vytvořeny standardní prezentace, a to z důvodu zachování kontinuity s již zpracovanými materiály předmětu KIV/UZI.

Obsah jednotlivých prezentací bude rozvržen tak, aby na sebe jednotlivá témata co nejlépe navazovala a zároveň, aby vystihovala sylabus předmětu KIV/UZI. Tudíž zpracovaná problematika bude rozdělena celkem do tří prezentačních souborů. Každý soubor bude tematicky zaměřen na konkrétní oblast reprezentace znalostí a bude vystihovat vždy nejdůležitější termíny z dané domény.

5.1 První prezentace

Ve struktuře první prezentace se bude nacházet definice základních pojmů reprezentace znalostí. Obsah bude rozdělen na „obecný úvod“, „produkční (pravidlové) systémy“ a „nepravidlové systémy“. Souhrnný název souboru fólií bude „Reprezentace znalostí – úvod“. Struktura prezentace bude:

- definice znalostí a jejich získávání, prostředky reprezentace znalostí. . .
- definice ZS, typy a architektura ZS, popis jednotlivých komponent ZS (báze znalostí, báze faktů, inferenční mechanismus). . .
- pravidlové (produkční) systémy:
 - definice pravidlových systémů, produkční pravidla, popis struktury, popis inference v produkčních systémech, dopředné a zpětné řetězení, rezoluce, RETE algoritmus. . .
- nepravidlové systémy:
 - reprezentace znalostí v nepravidlových systémech, sémantické sítě, rámce, objekty, Petriho sítě, rozhodovací stromy, zpětná indukce. . .

5.2 Druhá prezentace

Druhý soubor výukových materiálů bude úvodem do zpracování neurčitosti ve znalostech a faktech. Bude zde uveden popis základních termínů a přístupů ke zpracování neurčitosti. Souhrnný název prezentace bude „Zpracování neurčitosti“. Prezentace bude mít následující strukturu:

- definice neurčitosti a usuzování za neurčitosti
- úvod do základních přístupů pro zpracování neurčitosti v ZS
- popis Bayesovského přístupu
- definice inferenční sítě a šíření pravděpodobnosti v inferenčních sítích

5.3 Třetí prezentace

Poslední prezentace bude pojednávat o vybraných přístupech nakládání s neurčitostí ve znalostech (popř. ve faktech). Souhrnný název fólií bude „Vybrané přístupy řešení neurčitosti“. Struktura prezentace bude:

- faktory jistoty, míry důvěry a nedůvěry...
- Dempster-Shaferova teorie, míry domnění a plauzibility...
- definice přibližného usuzování, fuzzy logiky, fuzzy množiny, fuzzifikace, fuzzy inference...

6 Prezentace o reprezentaci znalostí

6.1 Způsob vytvoření

Obsah prezentací je vytvořen podle návrhu struktury, který se nachází v kapitole 5. V prezentacích jsou vždy uvedeny nejdůležitější pojmy tak, aby se s pomocí těchto fólií dalo jednoduše přednášet.

Při vytváření prezentací pro výuku reprezentace znalostí byly využity (jako šablony pro vzhled a způsob zápisu informací) již existující práce. Jako výchozí byla vzata práce pana Bc. T. Ciglera, pojednávající o programování v Prologu.

Všechny nové stránky pro výuku reprezentace znalostí jsou vytvořeny v textovém procesoru „MS Office Word 2007“ (v tomto nástroji jsou udělány i předešlé prezentace předmětu KIV/UZI). Obrázky (v prezentacích uvedené) jsou vytvořeny v nástrojích „Adobe Illustrator CS3“ a „MS Office Visio 2013“. Pokud je některý z obrázků inspirován z cizí práce, pak je tato skutečnost u daného objektu vždy uvedena.

1. Reprezentace znalostí - úvod

Co je to znalost?

Pojem **znalost** zahrnuje nejen teoretické vědomosti člověka z dané domény, ale také jeho dlouhodobé zkušenosti z praxe.

V počítačové terminologii je **znalost** považována za formu organizace strukturovaných dat.



Schéma inspirováno z [2].

© Josef Kazák, Václav Matoušek / KIV
Úvod do znalostního inženýrství, ZS 2015/16 P-2

Obrázek 6.1: Ukázka stránky z prezentací

7 Umístění praktické části BP na webu

Kvůli udržení stávajícího konceptu organizace datových souborů je praktická část této BP (tj. fólie a programové řešení) umístěna na oficiálních stránkách předmětu KIV/UZI. Web je dostupný na následující adrese: <http://www.kiv.zcu.cz/studies/predmety/uzi/>.

7.1 Popis zakomponování prezentací

Prezentace jsou (ve formátu PDF) umístěny v podsekcí „Fólie pro přednášky“, která se nachází v sekci „Organizační informace“. Fólie jsou dostupné pod odkazy:

- „Fólie „Reprezentace znalostí – úvod“ Josefa Kazáka (2015)“
- „Fólie „Zpracování neurčitosti“ Josefa Kazáka (2015)“
- „Fólie „Vybrané přístupy řešení neurčitosti“ Josefa Kazáka (2015)“

7.2 Popis včlenění ZS

„Diagnostický znalostní systém pro srovnávací a tloušťkovací frézku Jeřábek SP410PRO“ je umístěn v archívu, který je ke stažení na již zmíněných stránkách předmětu pod odkazem „Příklad znalostního systému v CLIPS“, který se nachází v sekci „Zajímavosti“. Společně se ZS je v archívu také nástroj CLIPS (ve verzích pro MS Windows a GNU/Linux) a uživatelská dokumentace.

8 Testování

Předmětem testování je pouze programový segment praktické části BP, a to „Diagnostický znalostní systém pro srovnávací a tloušťkovací frézku Jeřábek SP410PRO“. Kvůli zajištění objektivnosti experimentování a ověření použitelnosti cílovými uživateli proběhlo testování prostřednictvím tří profesionálních truhlářů.

Experimentování bylo provedeno následujícím způsobem. Nejprve bylo na jednom ukázkovém příkladu každému truhláři vysvětleno, jak se ZS ovládá. A poté již byl měřen čas každé identifikace problému, přičemž byla započítávána chybná rozhodnutí (jednalo se o rozhodnutí uživatele, která vedla k nesprávné identifikaci problému).

Měření času identifikace problému započalo vždy od stavu systému, ve kterém se nachází ihned po spuštění, a skončilo při nalezení daného řešení. Případně by měření času bylo ukončeno v případě, že by se na hledání řešení rezignovalo (což se ale při experimentování nestalo).

Pro ověření použitelnosti ZS bylo připraveno následujících šest modelových případů:

1. Frézka nejde zapnout, přičemž elektrický proud jde, motor je chladný a přepět'ová pojistka je v pořádku.
2. Jeden z litinových stolů je nerovný.
3. Po zapnutí frézky se nožová hřídel otáčí opačným směrem.
4. Frézka při tloušťkování vrhá obrobek zpět k podávací osobě, přičemž jsou hoblovací nože správně uloženy a nožová hřídel je v pořádku.
5. Při tloušťkování se vytvářejí na obrobku podélné rýhy.
6. Při srovnávání není možné dosáhnout pravého úhlu, pomocné pravítko je v pořádku (je správně nastaveno a styčná plocha je nepoškozená).

Správné odpovědi na výše uvedené modelové příklady jsou (samozřejmě během testování tyto odpovědi uživatelé k dispozici neměli a zde jsou ve zkrácených verzích uvedeny jen pro informativní účely):

1. Zkontrolujte následující (časté) zdroje těchto problémů. Zda jsou sepnuté všechny ochranné systémy. Jestli nejsou zoxidované či uvolněné vodiče v hlavním vypínači. . .
2. Nechte stůl/stoly zbrousit po celé ploše, případně je nahraďte novými.
3. Vyměňte dvě fáze na přívodním elektrickém kabelu.
4. Zkontrolujte klíny, zabraňující zpětnému vrhu obrobku, zda nejsou poškozené, jsou ostré a svévolně se vrací do výchozí polohy. Případně je namažte, nabruste či vyměňte.
5. Odstraňte všechny nečistoty z litinového stolu, možné nedokonalosti litiny vybruste.
6. Zkuste obrobit jiný materiál, patrně dochází k postupnému kroucení obrobku.

V tabulkách níže (tj. tab. 8.1 – tab. 8.3) jsou k nahlédnutí dosažené výsledky. Pro úplnost, sloupec s hlavičkou „Počet nesprávných rozhodnutí“ vyjadřuje stav, kdy uživatel musel (jak již bylo uvedeno výše) vrátit své rozhodnutí (tzn., že musel zadat „zpět“). Tento sloupec slouží pro ověření, jestli jsou otázky správně formulovány.

Prvním testujícím truhlářem byl Josef Kazák (jedná se o otce autora této BP, který disponuje frézku Jeřábek SP410PRO), který dosáhl výsledků zaznamenaných v tab. 8.1. Druhým byl Miloš Novák (ze společnosti zabývající se výrobou nábytku a vlastní frézku ze stejné kategorie, jako je Jeřábek SP410PRO), který dosáhl výsledků znázorněných v tab. 8.2. A posledním byl Daniel Závorka (ze stejné firmy jako Miloš Novák), jehož výsledky jsou v tab. 8.3.

Tabulka 8.1: Výsledky Josefa Kazáka

Úkol	Řešení nalezeno	Počet nesprávných rozhodnutí	Doba nalezení od-povědi [m:ss]
1	ano	0	1:47
2	ano	1	3:59
3	ano	0	1:56
4	ano	2	3:24
5	ano	0	1:01
6	ano	0	1:50

Tabulka 8.2: Výsledky Miloše Nováka

Úkol	Řešení nalezeno	Počet nesprávných rozhodnutí	Doba nalezení odpovědi [m:ss]
1	ano	2	2:03
2	ano	0	3:31
3	ano	1	2:06
4	ano	1	2:56
5	ano	0	0:58
6	ano	0	1:37

Tabulka 8.3: Výsledky Daniela Závorky

Úkol	Řešení nalezeno	Počet nesprávných rozhodnutí	Doba nalezení odpovědi [m:ss]
1	ano	1	1:53
2	ano	1	4:02
3	ano	0	1:45
4	ano	1	2:57
5	ano	0	0:49
6	ano	0	1:26

Všem výše zmíněným truhlářům patří poděkování za jejich projevenou ochotu při testování.

Z tabulek vyplývá, že systém je pro běžného truhláře použitelný, jelikož každý z nich zvládl nalézt správná řešení všech problémů. Celkový počet nesprávných odpovědí (rozhodnutí uživatele, vedoucí k chybné identifikaci problému) je u každého uživatele obdobný s ostatními, avšak tyto chyby vznikaly spíše z nepozornosti než z chybné formulace otázky. Dobu, potřebnou k nalezení odpovědi, lze považovat za přijatelnou, navíc tento čas je velmi ovlivněný rychlostí čtení uživatele než samotným usuzováním ZS.

Ovšem během testování bylo zjištěno, že cíloví uživatelé mají problémy s absencí diakritiky ve ZS, což je ale kvůli multiplatformnosti nutné.

9 Zhodnocení

Jelikož se praktická část této BP skládá ze dvou segmentů, a to z fólií pro výuku reprezentace znalostí a „Diagnostického ZS pro srovnávací a tloušťkovací frézku Jeřábek SP410PRO“, je následující kapitola rozdělena pro každou z právě zmíněných částí zvlášť. U obou částí je zhodnocena výsledná práce a nastíněn další možný rozvoj.

9.1 Prezentace o reprezentaci znalostí

Prezentace plně respektují syllabus předmětu KIV/UZI, tudíž jsou rozděleny do tří tematických celků a obsahují všechny vytyčené pojmy. Jednotlivé stránky fólií vystihují pouze nejdůležitější informace tak, aby se s jejich pomocí dalo snadno přednášet. Z toho vyplývá, že nejsou koncipovány pro samostudium, k tomu slouží různé webové stránky, skripta a knihy. Všechny fólie mají dohromady 94 stran (včetně titulní strany, literatury atd.), což je na tři přednáškové bloky odpovídající.

Jako další rozšíření je možné doplnění dalších (detailnějších) pojmů, pokročilejších přístupů pro zpracování neurčitosti atd. Dále je možné dodat různé příklady zdrojových kódů, ukázky z praxe atd.

9.2 ZS pro frézku Jeřábek SP410PRO

Výše zmíněný ZS je tzv. „produkčním systémem“, který se skládá z deseti souborů, obsahujících 183 produkčních pravidel a další pomocné metody. Systém je implementován v nástroji CLIPS. Nástroj umožňuje vytváření pravidlových systémů a disponuje inferenčním mechanismem, který využívá „dopředné řetězení“ a „RETE algoritmus“. Z toho vyplývá, že sice tento nástroj není nejlepší pro tvorbu ZS provádějícího diagnostické úlohy (vhodnější by byl nástroj se „zpětným řetězením“), ale díky neustálému vývoji, volné licenci užívání, multiplatformnosti atd., je vhodnou volbou pro takový ZS, jako je systém vytvořený v rámci této BP.

Samotná implementace ZS je navržena takovým způsobem, aby i člověk, neznalý dané realizace, rychle porozuměl a dokázal rozšířit či upravit stávající bázi znalostí (popř. bázi faktů). Jednotlivé soubory programu (obsahující produkční pravidla apod.) jsou rozděleny podle charakteru řešené úlohy. Pravidla v některých souborech řeší správný chod systému a v jiných souborech zajišťují identifikaci a poskytování řešení daných problémů. Soubory, jejichž obsah zajišťuje identifikaci a řešení problémů, jsou rozděleny podle problémové domény. Např. pravidla jednoho souboru jsou určena pro identifikaci problémů ohledně zapnutí frézky a pravidla druhého souboru zajišťují řešení potíží týkajících se srovnávací funkce frézky.

Nevýhodou systému je hlavně absence diakritiky (což bylo vytknuto i testujícími uživateli – viz kapitola 8). Tato skutečnost je zapříčiněna různým kódováním písma v příkazových řádkách napříč platformami. Ovšem tohoto handicapu je možné se zbavit v případě specifikace ZS pro určitou platformu.

Mezi možnosti dalšího rozvoje patří např. doplnění stávající báze znalostí o nové poznatky. Dále pak je možná integrace ZS do komplexního softwarového řešení pro truhláře, kde by diagnostika problémů frézky byla pouze jednou z mnoha komponent atd.

10 Závěr

Cílem bakalářské práce bylo vytvoření výukových materiálů, zabírajících se reprezentací znalostí. Výsledkem jsou tři prezentace, pokrývající teorii reprezentace znalostí od definic základních pojmů až po vybrané přístupy zpracování neurčitosti, a diagnostický znalostní systém, využívající vybrané metody teorie reprezentace znalostí.

Prvním cílem práce bylo seznámení se s již zmíněnou problematikou reprezentace znalostí a metodami vytváření výukových materiálů. Zpracováním této části byl získán teoretický rozhled, na jehož základě byl zpracován prvotní návrh prezentací. Jednou z věcí, která z tohoto návrhu vyplynula je, že zpracované prezentace jsou standardními fóliemi, nikoli webovými stránkami, jak je uvedeno v původním zadání projektu. Změna byla provedena z důvodu zachování kontinuity s již zpracovanými materiály předmětu KIV/UZI.

Dalšími dvěma cíli projektu byl návrh struktury prezentací a jejich samotné vytvoření. Výsledkem jsou již zmíněné tři prezentace, které byly vedoucím práce označeny za kvalitní.

Do čtvrtého cíle spadá implementace vybraných metod reprezentace znalostí. Zpracováním této části vznikl produkční systém nazvaný „Diagnostický znalostní systém pro srovnávací a tloušťkovací frézku Jeřábek SP410PRO“. Systém byl testován cílovými uživateli, kteří jej shledali správným.

Předposledním cílem bylo umístění zpracovaných výukových materiálů (prezentací a znalostního systému) na samostatný web, který měl být dále začleněn do stávajících stránek předmětu KIV/UZI. Ovšem opět z důvodu zachování kontinuity s již zpracovanými materiály předmětu, byly prezentace vloženy přímo na oficiální stránky již zmíněného předmětu.

Posledním cílem bylo provedení zhodnocení dosažených výsledků. Ve zhodnocení je uveden nástin dalšího možného rozvoje a výčet kladů a záporů zpracování výukových materiálů.

Splněním všech výše zmíněných cílů vznikly komplexní výukové materiály, a to včetně reálného znalostního systému, které je možné dále rozšiřovat, jak již bylo nastíněno v kapitole 9.

Literatura

- [1] BERKA, P. a kol. *Expertní systémy*. Skriptum. Praha, VŠE 1998.
- [2] KELEMEN, J. *Tvorba expertních systémů v prostředí CLIPS: Podrobný průvodce*. 1. vyd. Praha: Grada Publishing, 1999, ISBN 80-716-9501-7.
- [3] DVOŘÁK, J. *Expertní systémy*. Skriptum. Brno, VUT 2004.
- [4] EXPERTNÍ SYSTÉMY I [online]. Jaroslav Teda. [cit. 12. ledna 2015]. Dostupné z: <http://programujte.com/clanek/2008020100-expertni-systemy-i/>
- [5] JIROUŠEK, R. *Metody reprezentace a zpracování znalostí v umělé inteligenci*. 1. vyd. Praha: Vysoká škola ekonomická, 1995, ISBN 80-7079-701-0.
- [6] CELBOVÁ, I. *Uvod do problematiky expertních systémů* [online]. [cit. 11. března 2015]. Dostupné z: <http://ikaros.cz/uvod-do-problematiky-expertnich-systemu>.
- [7] VOLNÁ, E. a KOTYRBA M. *Umělá inteligence*. Skriptum. Ostrava, Ostravská univerzita, 2013.
- [8] WEBOVÝ PRŮVODCE SVĚTEM EXPERTNÍCH SYSTÉMŮ [online]. Petr Fazurel. [cit. 13. ledna 2015]. Dostupné z: <http://www.milost.wz.cz/umi/referat/index.html>.
- [9] EXPERTNÍ SYSTÉMY [online]. Miloslav Thon. [cit. 13. ledna 2015]. Dostupné z: <http://www.milost.wz.cz/umi/referat/index.html>.
- [10] *Reprezentace znalostí* [online], poslední aktualizace 23. 10. 2014 ve 14:04 [cit. 13. ledna 2015], Wikipedie. Dostupné z: http://cs.wikipedia.org/wiki/Reprezentace_znalosti
- [11] LAGUN, E. *Evaluation and Implementation of Match Algorithms for Rule-based Multi-Agent Systems using the Example of Jadex*. Hamburk, 2009. Diplomová práce. Universität Hamburg.

-
- [12] ALGORITMY NA EFEKTÍVNU REPREZENTÁCIU PRAVIDIEL [online]. Jozef Valo. [cit. 7. března 2015]. Dostupné z: <http://www2.fiit.stuba.sk/~kapustik/ZS/Clanky0809/valo/index.html>
- [13] CLIPS REFERENCE MANUAL: VOLUME I BASIC PROGRAMMING GUIDE [online]. Culbert, C. a kol. [cit. 3. března 2015]. Dostupné z: <http://clipsrules.sourceforge.net/documentation/v630/bpg.htm>
- [14] DVOŘÁK, P. *Uvod do inference*. Brno, 2002. Bakalářská práce. Masarykova univerzita.
- [15] MÜLLER, L. *Znalostní systémy*. Skriptum. Plzeň, ZČU 2002.

A Uživatelská dokumentace

V této části jsou popsány základní požadavky, instalace prostředí CLIPS, některé možnosti spuštění „Diagnostického znalostního systému pro srovnávací a tloušťkovací frézku Jeřábek SP410PRO“ (dále jen ZS) a ovládání ZS.

A.1 Požadavky aplikace

Pro správný běh ZS je nutné prostředí CLIPS. Toto prostředí je možné instalovat na následující operační systémy: Unix, Mac OS X a MS Windows (doporučovaným minimem je MS Windows XP). CLIPS je zdarma k dispozici např. na oficiálních stránkách: <http://clipsrules.sourceforge.net>.

ZS byl vyvíjen a testován na MS Windows 8.1 (s nástrojem CLIPS ve verzích 6.24 a 6.3 Beta) a na Xubuntu 14.04 (s nainstalovaným CLIPS 6.24).

A.2 Instalace CLIPS a spuštění ZS

Jak již bylo zmíněno výše, ZS byl vyvíjen a testován na MS Windows a Xubuntu. Tudíž jsou dále popsány některé možnosti instalace prostředí a spuštění ZS pouze pro tyto dvě platformy.

A.2.1 GNU/Linux

Instalace prostředí

Instalace CLIPS na GNU/Linux lze provést např. stažením nejnovější verze z repositářů. Pro tento způsob musí být uživatel v příkazové řádce přihlášen jako administrátor systému a poté musí zadat příkaz:

```
apt-get install clips
```

Dalším způsobem je kompilace zdrojových kódů. Zdrojové kódy je možné stáhnout např. z výše zmíněných stránek.

Spuštění ZS

Způsobů, jak spustit ZS na GNU/Linux, existuje několik. Tím nejjednodušším je přesun příkazovou řádkou do adresáře se zdrojovými soubory ZS a poté zadat následující příkaz:

```
clips -f "zs-start.bat"
```

A.2.2 MS Windows

Instalace CLIPS

Po stažení verze pro MS Windows (z výše zmíněných stránek) stačí spustit průvodce instalací a postupovat podle pokynů. Po instalaci je prostředí již plně k dispozici.

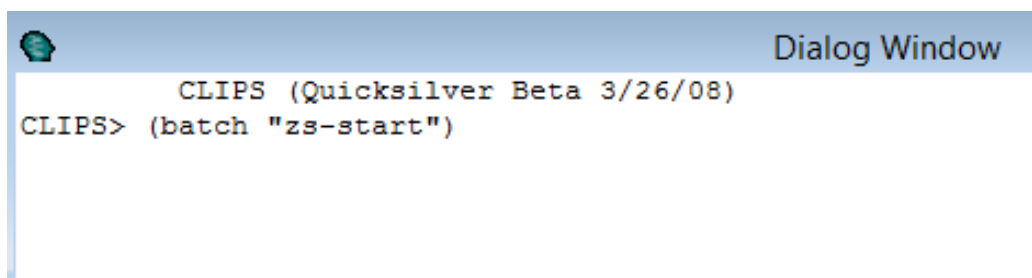
Spuštění ZS

V MS Windows lze spustit ZS příkazovou řádkou. Stačí se přesunout do adresáře se zdrojovými soubory a zde zadat (za předpokladu, že jsou správně nastaveny cesty v proměnném prostředí systému) příkaz:

```
clipsdos -f "zs-start.bat"
```

Další variantou je spuštění ZS v prostředí CLIPSWin, které je součástí instalace CLIPS. V tomto prostředí je nejprve nutné otevřít soubor „zs-start.bat“ a poté do dialogového okna zadat příkaz uvedený níže. Ukázka spuštění ZS v prostředí CLIPSWin viz obrázek A.1.

```
(batch "zs-start.bat")
```



Obrázek A.1: Ukázka spuštění ZS v prostředí CLIPSWin

A.3 Ovládání programu

Po úspěšném spuštění aplikace je uživatel tázán a vždy je mu nabídnuta množina možných odpovědí. Uživatel zadá jednu položku z již zmíněné množiny a potvrdí ji klávesou Enter. Takto se pokračuje, dokud není vypsán příslušný poznatek.

Uživatel může při jakékoliv otázce ZS ukončit. Během identifikace problému rušit předešlá rozhodnutí. A neposlední v řadě je možné (podle odpovědi uživatele na příslušnou otázku) celou exekuci ZS zaznamenat do textového souboru, který je automaticky vytvořen v adresáři ZS.

B Rozhodovací diagram ZS

V této kapitole (tedy přesněji v sekci B.1) je znázorněna kompletní identifikace problémů (i se závěrečnými doporučeními/poznatky) ZS. V obdélnících s ostrými rohy jsou otázky (s výjimkou nejvyššího obdélníku v diagramu B.1, ten popisuje inicializaci ZS a báze faktů) a v obdélnících se zaoblenými rohy jsou vypsány poznatky.

Poznatky, začínající znakem „*“, se v diagramech (kvůli přehlednosti) nacházejí vícekrát, ovšem v reálné implementaci existují jen jednou. A příslušná pravidla, na která takto označené poznatky navazují, pouze upravují bázi faktů tak, aby inferenční mechanismus vybral tyto poznatky přednostně.

Všechny nákresy, nacházející se níže, tvoří jeden rozsáhlý diagram¹, jehož hlavní spojovací článek je znázorněn v diagramu B.1 (množina hypotéz, vyobrazených v nákresu B.1, je „vstupním bodem“ do zbylých diagramů).

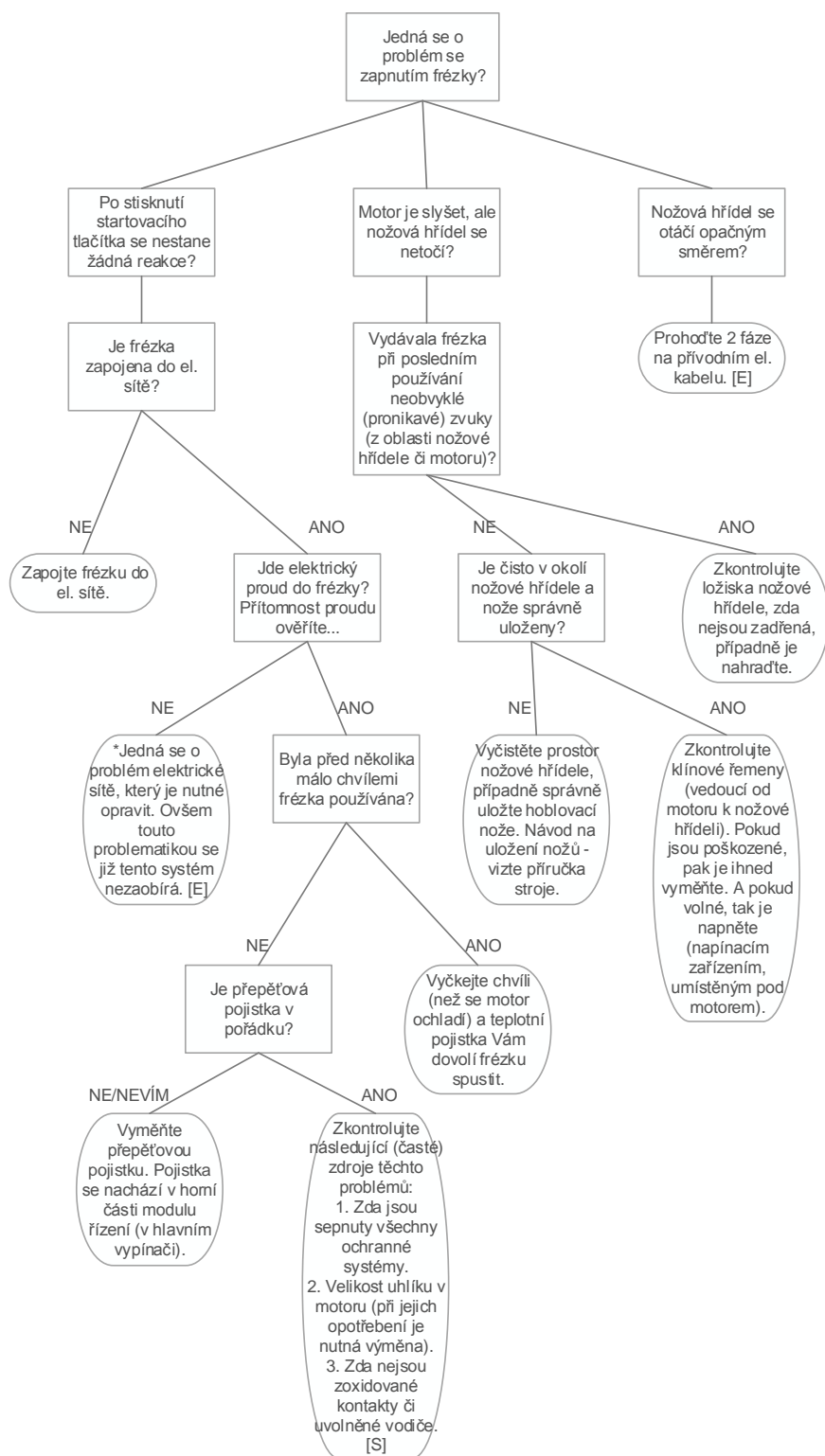
Některé znalosti v diagramech jsou ukončeny značkou „[E]“ nebo „[S]“. „[E]“ značí, že k danému výpisu znalosti přidává ZS dovětek, který upozorňuje, že daný úkon by měla provést osoba s patřičným oprávněním. A „[S]“ signalizuje, že k příslušnému výpisu znalosti přidává ZS upozornění, které říká, že na provedení daného úkonu by měl být přivolán odborný servis.

B.1 Rozhodovací diagram po částech

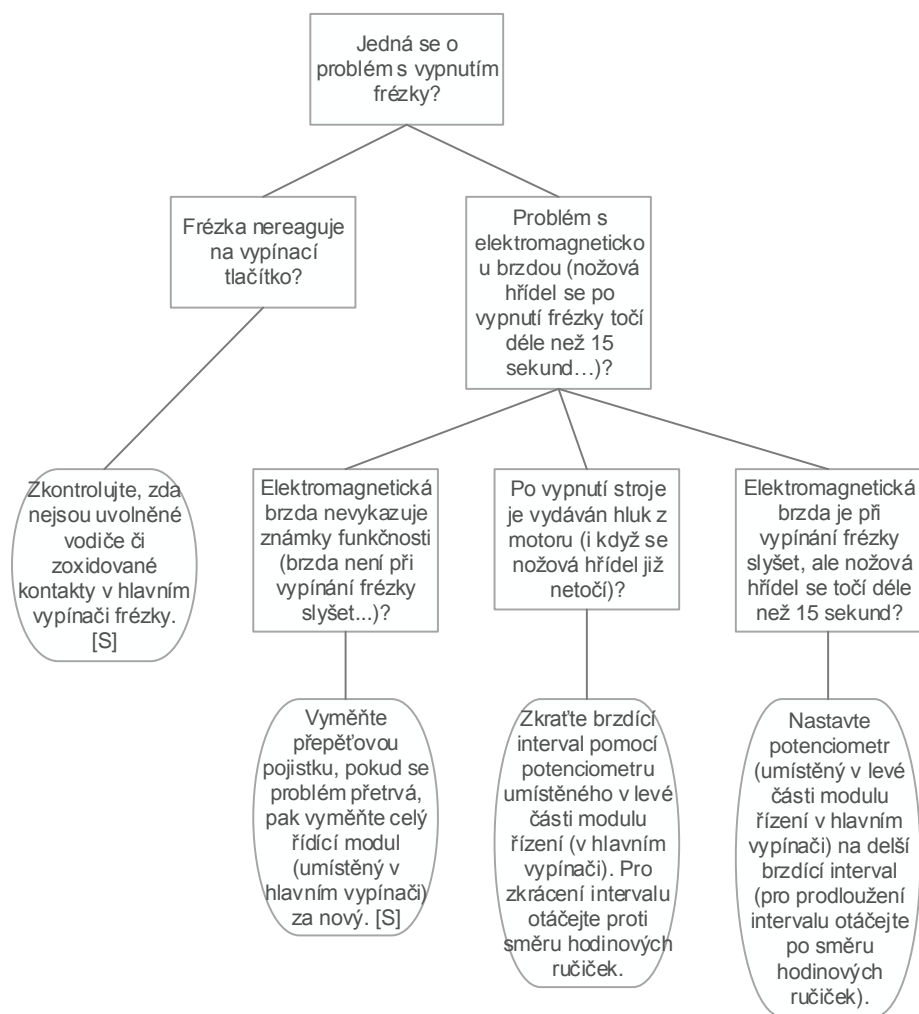


Obrázek B.1: Rozhodovací diagram počátku identifikace problémů

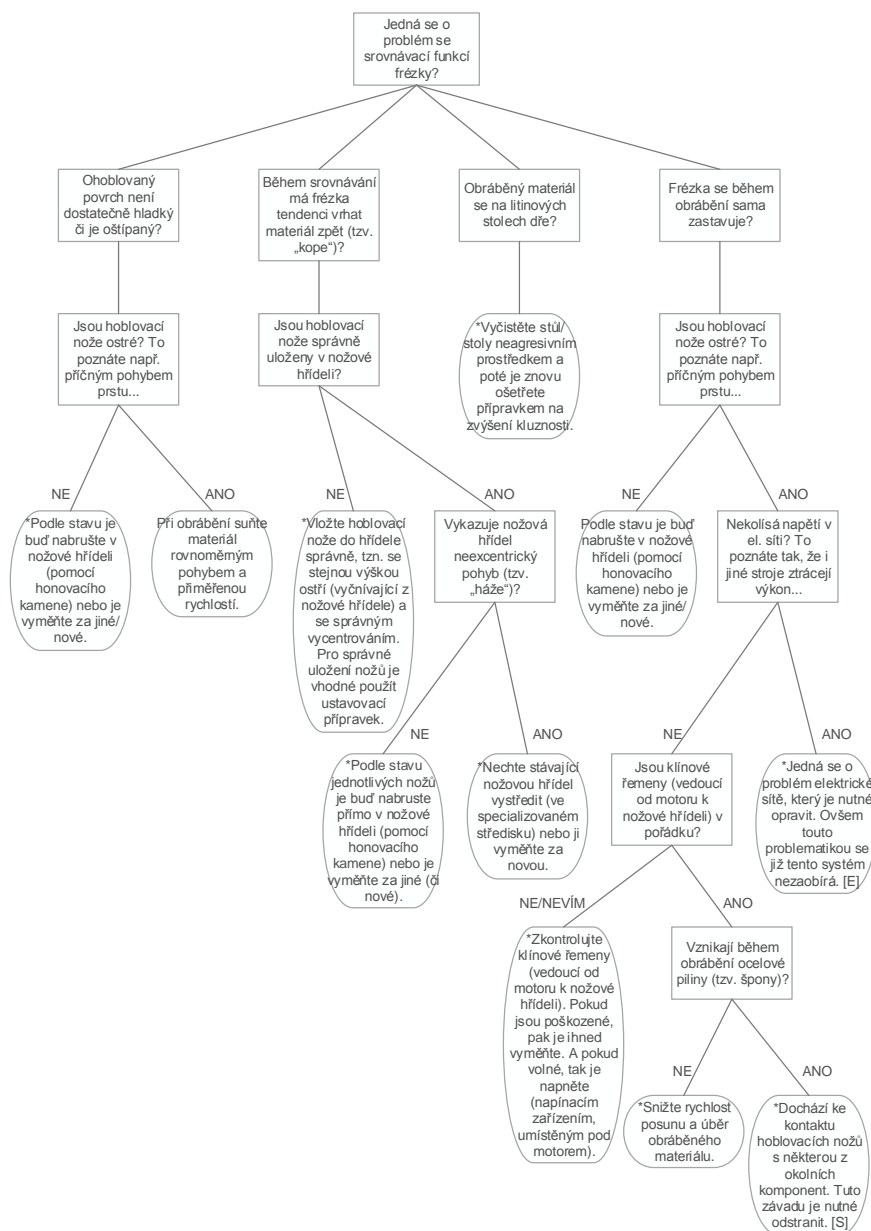
¹Diagram identifikace problémů je v této sekci rozdělen z důvodu velkého rozsahu, který není možné umístit na papír formátu A4 tak, aby bylo vše čitelné.



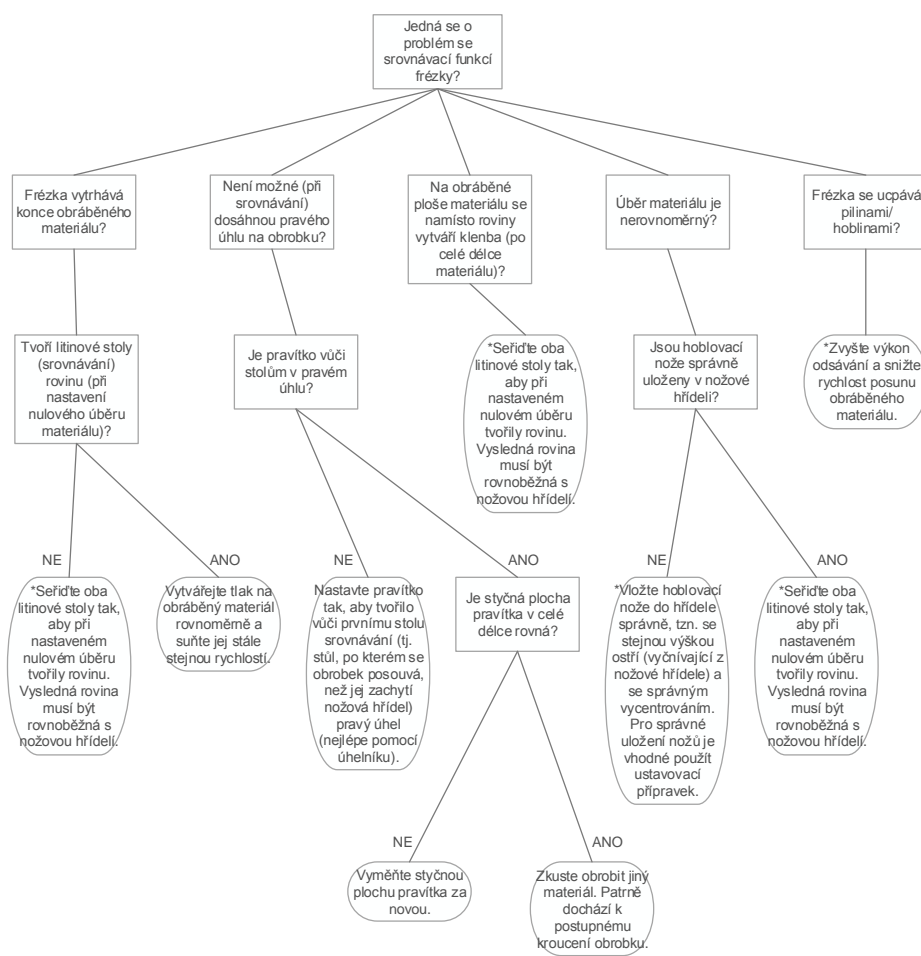
Obrázek B.2: Rozhodovací diagram pro problematiku zapnutí frézky



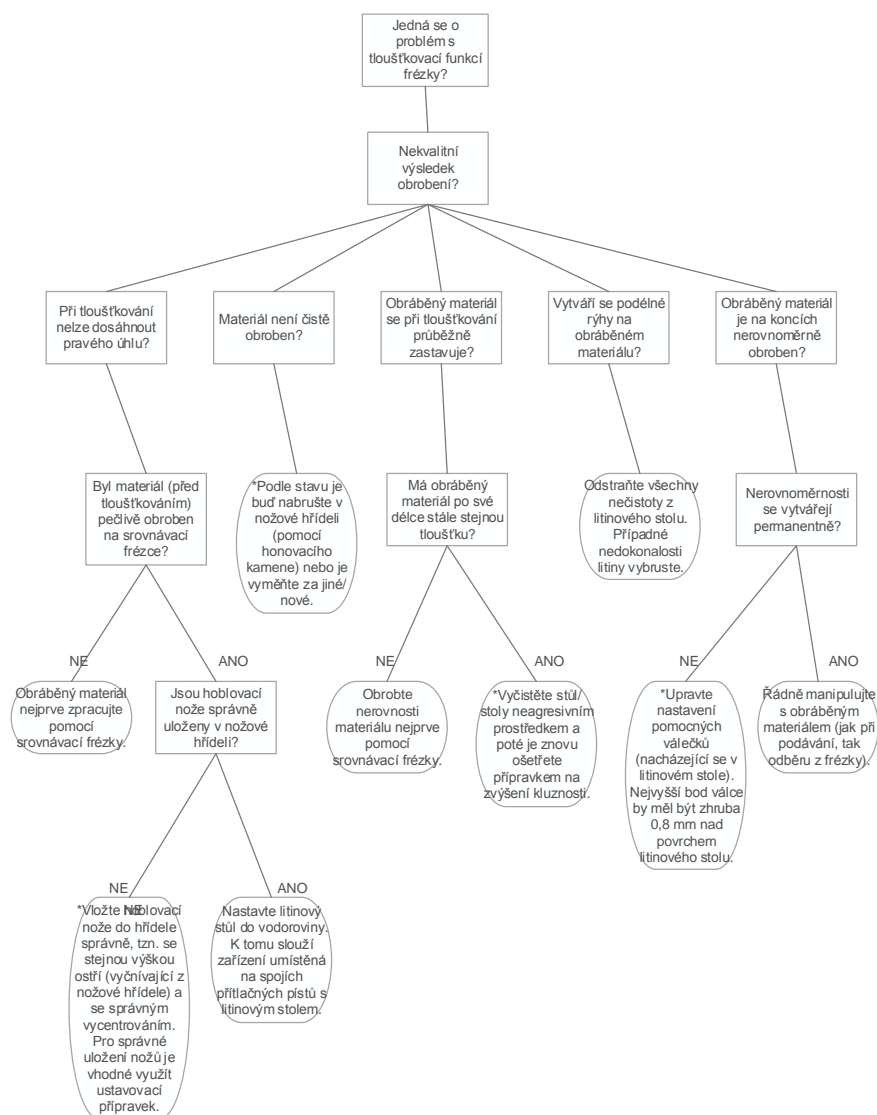
Obrázek B.3: Rozhodovací diagram pro problematiku vypnutí frézky

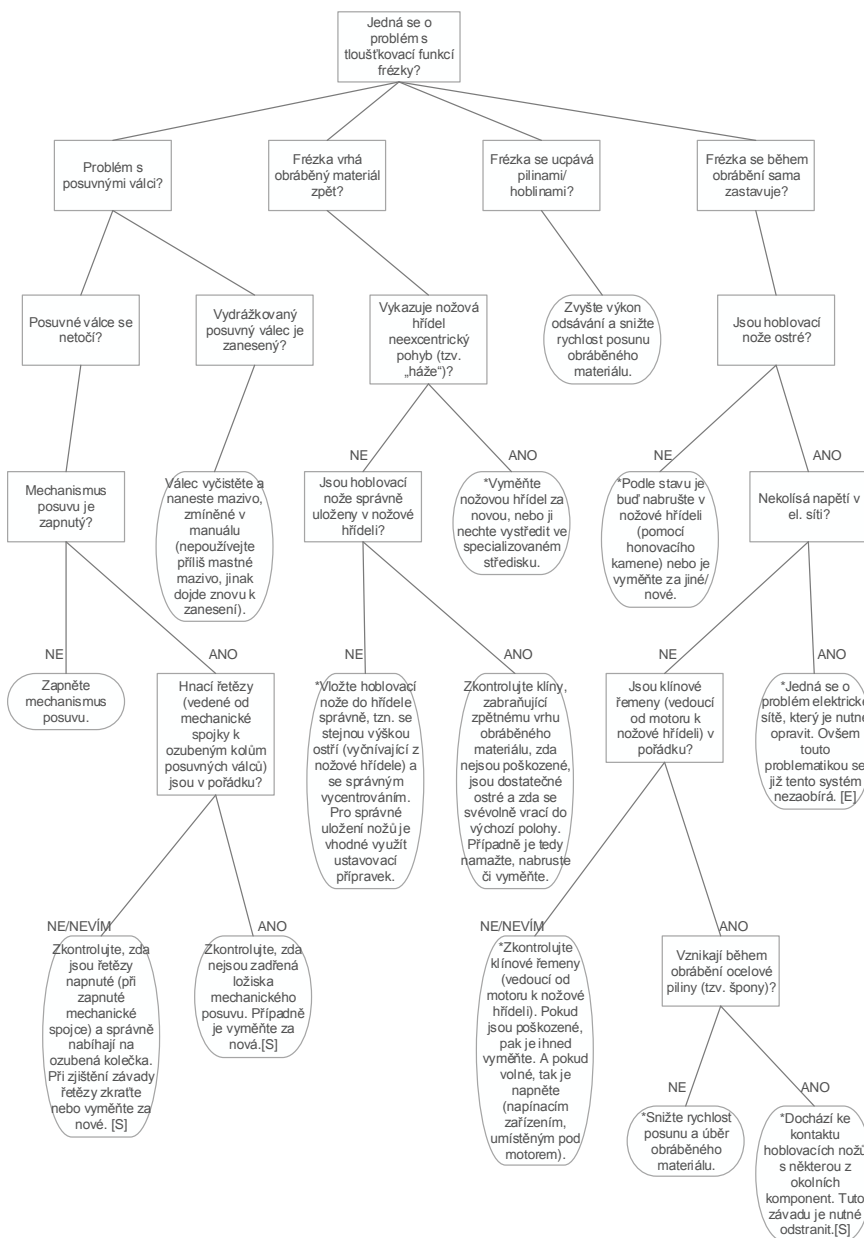


Obrázek B.4: Rozhodovací diagram srovnávací funkce frézky – část 1/2

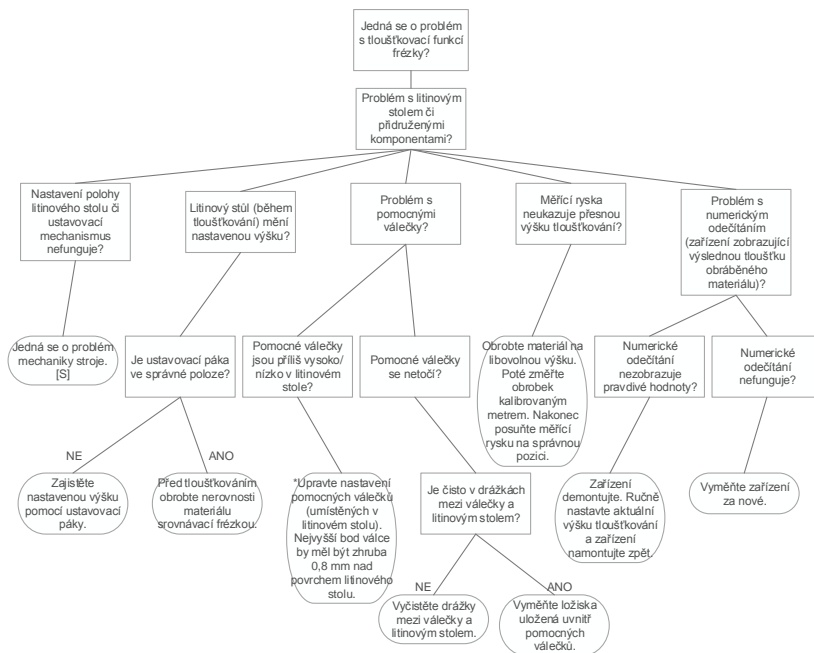


Obrázek B.5: Rozhodovací diagram srovnávací funkce frézky – část 2/2

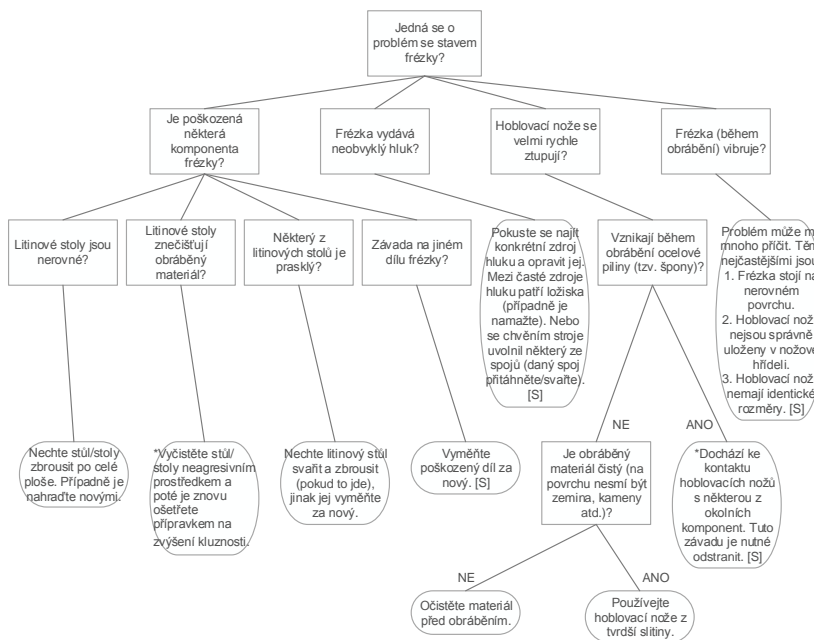




Obrázek B.7: Rozhodovací diagram tloušťkovací funkce frézky – část 2/3



Obrázek B.8: Rozhodovací diagram tloušťkovací funkce frézky – část 3/3



Obrázek B.9: Rozhodovací diagram pro problematiku celkového stavu frézky