VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

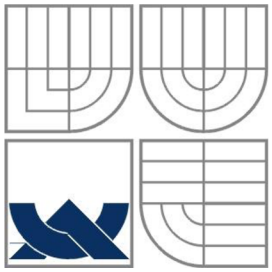
UMĚLÉ IMUNITNÍ VÝPOČETNÍ SYSTÉMY

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

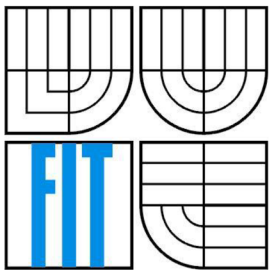
AUTOR PRÁCE
AUTHOR

DAVID NEUWIRTH

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

UMĚLÉ IMUNITNÍ VÝPOČETNÍ SYSTÉMY

ARTIFICIAL IMMUNE COMPUTATIONAL SYSTEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

DAVID NEUWIRTH

VEDOUCÍ PRÁCE
SUPERVISOR

DOC. ING. LUKÁŠ SEKANINA, PH.D.

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Neuwirth David**
Obor: Informační technologie
Téma: **Umělé imunitní výpočetní systémy**
Kategorie: Umělá inteligence

Pokyny:

1. Seznamte se s problematikou umělých imunitních výpočetních systémů. Prostudujte typické aplikace.
2. Vytvořte sadu appletů demonstrující principy umělých imunitních systémů.
3. Vytvořte výukový materiál na toto téma.
4. Zhodnoťte dosažené výsledky.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

1. Bod 1.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Sekanina Lukáš, doc. Ing., Ph.D.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. **Pan**

Jméno a příjmení: **David Neuwirth**
Id studenta: 84228
Bytem: Václava Jiříkovského 177/58, 700 30 Ostrava
Narozen: 03. 08. 1985, Vítkovice
(dále jen "autor")

a

2. **Vysoké učení technické v Brně**

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Umělé imunitní výpočetní systémy
Vedoucí/školitel VŠKP: Sekanina Lukáš, doc. Ing., Ph.D.
Ústav: Ústav počítačových systémů
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☒ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....	.....
Nabyvatel	Autor

Abstrakt

Umělé imunitní výpočetní systémy patří mezi relativně nová odvětví v oblasti počítačových systémů. Tato bakalářská práce demonstruje, jak se inspirovat v biologických imunitních systémech a jak díky těmto poznatkům vybrat nejdůležitější vlastnosti a principy, které se dají aplikovat v umělých imunitních systémech. Dále nabízí přehled o tom, kde a jakým způsobem se tyto myšlenky již dříve použily. Poslední částí této práce je popis implementace výukového materiálu, který by měl toto téma srozumitelně vysvětlit jeho budoucím studentům.

Klíčová slova

umělý imunitní systém, optimalizace, algoritmus pozitivní selekce, algoritmus negativní selekce, klonální selekční algoritmus

Abstract

Artificial immune computational system is a relatively new branch in the area of computer systems. This bachelor's thesis presents the fact that we can be inspired by biological immune systems and that we can use their attributes and principles in the artificial immune computational systems. On the following pages you can meet several examples, where and how the principles had been already used. The last part of this work is devoted to implementation of educational material into the lesson.

Keywords

artificial immune system, optimization, positive selection, negative selection, clonal selection

Citace

David Neuwirth: Umělé imunitní výpočetní systémy, bakalářská práce, Brno, FIT VUT v Brně, 2007

Umělé imunitní výpočetní systémy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením doc. Ing. Lukáše Sekaniny, PhD.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jméno Příjmení
Datum

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce doc. Ing. Lukášovi Sekaninovi, Ph.D. za cenné rady a trpělivost při konzultacích.
Dále děkuji Ing. Šárce Neuwirthové a Tereze Malcharové za připomínky a odbornou pomoc při psaní tohoto textu.
Speciální poděkování patří mé rodině, která mne všestranně podporuje ve studiu na vysoké škole a která mi byla oporou i při psaní této práce.

Obsah

1 Úvod

Umělé imunitní výpočetní systémy jsou relativně nová oblast v informačních systémech a ve výpočetní technice obecně. Imunitní systémy jsou zajímavé svými vlastnostmi nejen z medicínského hlediska, ale obsahují spoustu vlastností, mechanismů a prvků, kterými se můžeme inspirovat také v technice. Mezi nejzajímavější vlastnosti biologických imunitních systémů patří například přizpůsobivost neznámým situacím, robustnost nebo také schopnost učit se.

K tomu, abychom mohli vytvořit umělý imunitní výpočetní systém, musíme pochopit, jak funguje ten biologický. Z tohoto důvodu se budeme v první kapitole věnovat fungování biologických imunitních systémů. Vysvětlíme si základní vlastnosti a principy, které se následně pokusíme zobecnit a aplikovat v technice (kap. 2.2 a kap. 3). Ukážeme si základní algoritmy (mechanismy) používané v imunitních systémech (kap. 2.3) a jejich uplatnění v umělých imunitních systémech. Jako praktická část této bakalářské práce byla zadána tvorba učebního materiálu na téma „Umělé imunitní výpočetní systémy“ a tvorba tří Java apletů pro prezentaci principů důležitých algoritmů imunitních systémů. Ve čtvrté kapitole se proto podíváme na způsob realizace tohoto učebního materiálu.

2 Základní principy imunitních systémů

V této kapitole si vysvětlíme principy imunitních systémů. Podíváme se, jak fungují imunitní systémy v biologii, a pokusíme se aplikovat určité poznatky na umělé imunitní výpočetní systémy. K tomu, abychom vytvořili umělý imunitní systém, potřebujeme důkladně pochopit ten biologický a musíme z něj vybrat důležité principy a vlastnosti.

Poznatky o biologických imunitních systémech byly nastudovány převážně z [1], [2], [21] a [22], dále pak z [3] a [5].

2.1 Imunitní systém v biologii

Proč se zabývat právě biologickým imunitním systémem? V přírodě můžeme pozorovat spoustu případů, kdy zvířata přežívají různá zranění, léčí se z vážných i méně závažných nemocí a jsou pod neustálým útokem nejrůznějších bakterií a virů. To nás přivádí k myšlence a nápadu prozkoumat, proč jsou zvířata (a člověk) vůči těmto útokům a nehodám imunní a relativně snadno se s nimi vypořádávají, proč přežívají. Pokusíme se tedy zjistit, co činí imunitní systém tak výjimečným a co z něj dělá tak mocnou zbraň v boji proti infekcím, nemocím a zraněním.

Imunitní systém je soubor mechanismů v těle jedince, který zajišťuje identifikaci a eliminaci nepřátelských prvků. Imunitní systém je schopen detekovat útočníky od virů, přes nejrůznější bakterie až po různé plísně. Tito útočníci se obecně nazývají *patogeny* [1]. Samotná detekce je velice komplikovaná, neboť se patogeny v průběhu existence svého druhu vyvíjejí a různě mutují. Imunitní systém musí být tedy schopný rozpoznat a eliminovat útočníky, se kterými se již dříve setkal, ale také útočníky nové, které ještě nezná.

2.1.1 Úkol imunitního systému

Imunitní systém je velice vyspělý a sofistikovaný systém. Denně bojuje proti různým známým i neznámým bakteriím a virům - a vyhrává.[1]

Hlavním úkolem imunitního systému je ochrana těla jedince před neustálými útoky patogenů. Imunitní systém disponuje řadou prostředků k identifikaci a eliminaci těchto útočníků. Od fyzických bariér, buněk pro všeobecnou ochranu, přes buňky se specifickým zaměřením na určité útočníky až po paměťové buňky, které si daný patogen pamatují.

[1] Nejnebezpečnější dnes známý virus (Ebola – Zair) usmrtí až 90% nakažených [27], ale těch 10% imunitních systémů člověka si i s tímto nejsmrtelnějším virem poradí.

Úkoly imunitního systému bychom tedy mohli shrnout do dvou hlavních bodů. Prvním je identifikace (rozpoznání) cizí částice, kterou budeme nazývat antigen, a druhým důležitým bodem je reakce na tento cizorodý prvek, nejčastěji jeho eliminace.

2.1.2 Popis základních částí

Imunitní systém pracuje na několika vrstvách. První linií obrany je fyzická bariéra, která tělo chrání před vniknutím cizích látek. Pokud se patogenu podaří skrz tuto fyzickou bariéru proniknout (například drobnou kožní trhlinkou nebo díky zranění), pokusí se ho zastavit *vrozená imunita*. Tato vrozená imunita reaguje okamžitě, ale její reakce není specifická. Pokud se patogenu podaří proniknout i přes tuto část imunitního systému, čeká ho třetí vrstva - *adaptivní imunita*. Tato část imunitního systému se časem mění, přizpůsobuje se a učí se. Pokud tedy nějaký patogen zaútočí na naše tělo, adaptivní imunita si ho zapamatuje a při budoucím útoku bude reagovat daleko rychleji.

V této kapitole se tedy budeme zabývat jednotlivými částmi imunitního systému a podíváme se na důležité prvky a jakou úlohu v imunitním systému hrají.

2.1.2.1 Vrozená imunita

V této vrstvě hrají důležitou roli buňky, které hlídkují ve tkáních těla jedince. Na svém povrchu obsahují řadu čidel, která reagují na určité specifické vlastnosti a struktury patogenů. Pokud zaznamenají útočníka, okamžitě vyšlou řadu signálů, které přilákají další buňky z krevního řečiště a které aktivují systém adaptivní imunity.

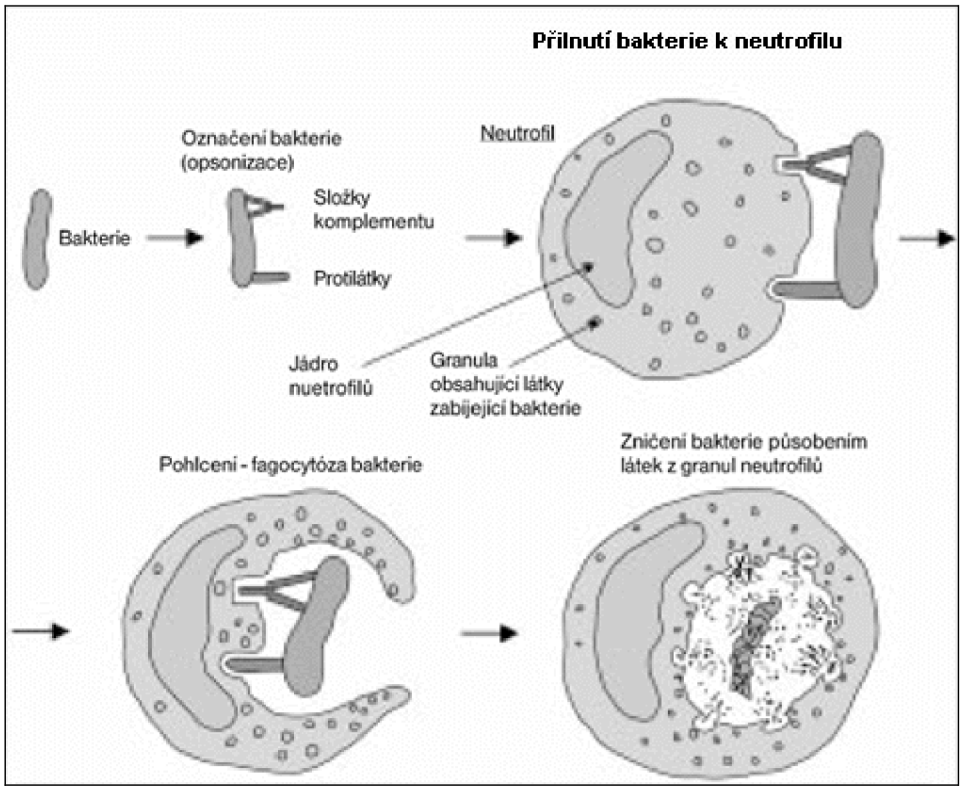
Důležitá vlastnost vrozené imunity je, že není specifická. Funguje na nejnižší biologicko-chemické úrovni a její chování by se dalo popsat jako „předprogramované“. Na rozdíl od adaptivní imunity reaguje na patogen jen obecnou reakcí a z dlouhodobého hlediska nemá vliv na vývoj schopností imunitního systému jako celku. Tato část imunitního systému hraje ale významnou roli.

Neutrofily a makrofágy

Neutrofily a makrofágy spolu s dendritickými buňkami patří do řádu fagocytů. Jsou to buňky, které dokážou rozpoznat a určitým způsobem reagovat na patogen. Tyto buňky hlídají tkáně lidského těla a pátrají po známkách infekce.

Patogen, který vnikl do těla jedince, je označen neboli opsonizován pomocí protilátek nazývaných *Imunoglobuliny* (zmíníme se o nich později v této kapitole). Potom, co se neutrofil naváže na patogen, zničí ho pomocí procesu, který se nazývá fagocytóza. Jde o proces, kdy je patogen zničen granulemi, které má neutrofil uvnitř své buňky (viz obrázek č. 1).

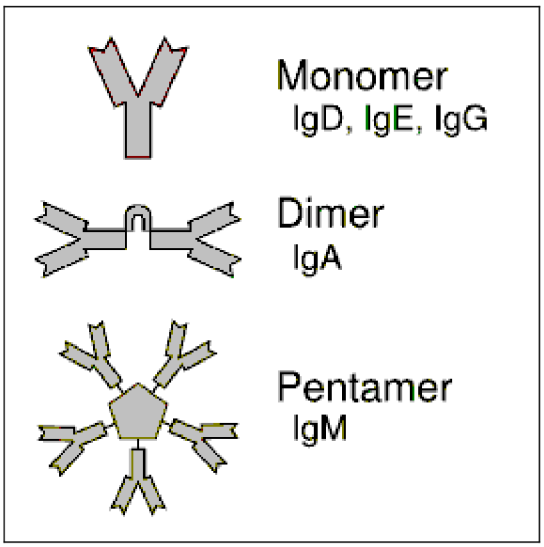
Pokud makrofágy najdou narušitele nesoucí cizorodý protein, obklopí jej a zničí. Zároveň vyplaví řadu *cytokinů*, z nichž některé spustí alarm nutící další buňky cestovat do místa zánětu a obecně řečeno uvádí imunitní systém do plné pohotovosti.



obrázek č. 1 – Fagocytóza (převzato z [21])

Imunoglobuliny

K tomu, aby makrofágy a neutrofily rozpoznaly patogen, je zapotřebí protilátek. Tyto protilátky se nazývají imunoglobuliny a dělí se do pěti tříd podle funkcí a schopností (viz obrázek č. 2). Jsou to třídy Imunoglobulin G (IgG), Imunoglobulin A (IgA), Imunoglobulin M (IgM), Imunoglobulin D (IgD) a Imunoglobulin E (IgE). Nejrozšířenější je IgG, který dokáže vnikat do různých tkání a jako jediný dokáže projít skrz placentu vyvíjejícího se plodu.



obrázek č. 2 – Imunoglobuliny (převzato z [22])

Jak patogen, tak i neutrofil nebo makrofág mají nejčastěji záporný elektrický potenciál. Pokud tedy dvě částice mají stejný náboj, je pro ně velice těžké přiblížit se k sobě natolik, aby se mohly navázat. Zde nastupují imunoglobuliny, které určitým způsobem patogeny zneutralizují, navážou se na ně a usnadní tím práci neutrofilům a makrofágům (viz obrázek č. 1).

Dendritické buňky

Dendritické buňky naproti tomu stráví narušitelské mikroby a pak putují do mízních uzlin, kde předloží fragmenty cizorodých proteinů armádě T-lymfocytů a také vyplaví cytokiny, což napomůže zahájit odpověď adaptivní imunity.

Další funkcí cytokinů je spuštění zánětlivé reakce. Ta se projevuje zarudnutím a otokem místa vniknutí patogenu, zvýšenou teplotou a příznaky podobnými chřipce. Pro některé patogeny zde jejich útok končí, jelikož nesnesou právě zvýšenou teplotu.

Toll-Like Receptory

Jak dendritické buňky rozpoznají patogen? Na tuto otázku nám přináší odpověď Toll-Like receptory [1]. Toll-Like receptory (TLR) jsou vlastně čidla, která reagují na určité bílkoviny, proteiny a specifické cizorodé molekuly. TLR hrají klíčovou roli ve vrozené imunitě. Existuje několik typů TLR - u člověka jich známe doposud 10. Například TLR 2 se dokáže navázat na kyselinu lipoteichoovou, která je součástí bakteriální stěny, TLR 3 rozpoznává genetický materiál virů, TLR 5 dokáže identifikovat flagellin, což je základní bílkovina pro tvorbu bičíků, díky němuž se bakterie pohybují, a například TLR 9 dokáže rozeznat chemický rozdíl ve vazbě mezi cytosinem a guaninem v DNA u bakterií a u savců.

Pokud některý TLR rozpozná cizí molekulu, okamžitě stimuluje dendritickou buňku k produkci cytokinů.

2.1.2.2 Adaptivní imunita

Adaptivní imunita je oproti vrozené imunitě specifická. Její obranné buňky cíleně útočí na určitý typ patogenu. K reakci adaptivní imunity dochází pouze tehdy, je-li stimulována imunitou vrozenou. Adaptivní imunita se v průběhu životního cyklu jedince neustále vyvíjí a zdokonaluje.

Díky adaptivní imunitě je imunitní systém jedince obdařen *pamětí*. Jakmile je infekce potlačena, trénované lymfocyty B a T obklopí zbytky patogenů a posléze se přemění na paměťové buňky. Díky této schopnosti máme možnost chránit se před chorobami očkováním. Pokud už je nepřítel známý (např. z dřívějšího útoku nebo právě díky očkování), je tvorba protilátek daleko rychlejší. Obranná reakce adaptivní imunity je tedy daleko rychlejší a efektivnější a jedinec často ani nepozná, že byl znovu nakažen.

Lymfocyty B a T

Lymfocyty typu B a lymfocyty typu T jsou nejdůležitější buňky adaptivního imunitního systému. Jsou to buňky, které dokážou detekovat patogen a určitým způsobem ho zajistit. Každý typ bakterie, viru nebo obecně jakékoli buňky je charakterizován svým *antigenem*, což je vlastně jeho charakteristický vzor nebo otisk. K tomu, aby bílé krvinky mohly rozpoznat patogen, mají na svém povrchu čidla, tzv. *receptory*, které dokážou daný antigen rozpoznat.

Lymfocyty typu B vznikají v kostní dřeni (B podle anglického názvu bone marrow) a lymfocyty typu T vznikají také v kostní dřeni, ale potom (zatím z neznámých důvodů) putují do brzlíku (T od anglického názvu thymus) a tam dozrávají. Každá bílá krvinka, když vstupuje do krevního řečiště, obsahuje právě jeden specifický antigenový receptor. Tato specifičnost je dána speciálními mechanismy, podle kterých se lymfocyty typu B a T utvářejí. Tyto mechanizmy mohou generovat milióny různých kombinací antigenových receptorů. V krvi jedince (v případě dospělého člověka) kolují milióny lymfocytů B a T a tudíž i milióny různých antigenových receptorů. Zde také nastává problém s tzv. *autoimunitou*, což je porucha imunitního systému, při které jsou buňky vlastního tělo označeny jako cizí a jsou postupně ničeny. Mezi nejznámější autoimunitní choroby patří cukrovka (kdy je eliminován inzulín) a revmatická artritida. Jak zabránit efektu autoimunity si řekneme v kapitole o algoritmu negativní selekce.

Pokud se bílá krvinka naváže svým receptorem na nějaký antigen patogenu, její metabolismus se prudce zrychlí a začne se množit. Tím se vytvoří stovky dalších lymfocytů, které jsou schopny detekovat tento určitý patogen. Tomuto množení se říká *klonální selekční expanze* (dále v textu si vysvětlíme a ukážeme, jak v umělých imunitních systémech funguje klonální selekční algoritmus).

Některé lymfocyty, potom, co se navážou na patogen, ho úplně zničí. Jiné lymfocyty patogeny jen rozlámou na malé kousky a s těmito fragmenty patogenu na svém povrchu odcestují do mízních uzlin. Podle nich se vytvoří během několika dní armáda lymfocytů B a T namířena přímo proti tomuto druhu patogenu.

2.1.3 Činnost systému

Pokud se tedy patogen dostane do těla poprvé, je zaznamenán jedním z Toll-Like receptorů, které mají na svém povrchu hlídkující dendritické buňky. V jiném případě se na něj navážou imunoglobuliny a je rozpoznán neutrofily a makrofágy. Nepřítel je zničen nebo se naváže na určitý TLR a ten pak stimuluje buňku k produkci cytokinů. Tito „poslové“ aktivují další makrofágy a dendritické buňky, aby se odpoutaly od svých stanovišť a nespecificky napadly nepřátelské bakterie. Současně cytokiny způsobují zánětlivou reakci, horečku a další příznaky podobné chřipce. Makrofágy a dendritické buňky pak pohltí nepřátelský patogen a rozlámou ho na malé kousky, které vystaví na svém povrchu. Poté spolu s uvolněnými cytokiny definitivně aktivují systém adaptivní imunity (lymfocyty T a B).

Pokud je naše tělo vystaveno patogenu, se kterým se v minulosti již setkalo, bude si armáda trénovaných buněk adaptivního imunitního systému narušitele pamatovat a vypořádá se s ním během několika málo hodin. Pokud je tento patogen pro naše tělo nový a neznámý, navážou se na něj lymfocyty typu B a T a dojde ke klonální selekční expanzi. Potom, co si nově vytvořená armáda lymfocytů s infekcí poradí, některé lymfocyty se promění na paměťové buňky pro případ, kdyby se nákaza vrátila.

2.2 Důležité principy imunitních systémů

Po pochopení biologických imunitních systémů, můžeme aplikovat tyto poznatky při návrhu a tvorbě umělých imunitních výpočetních systémů. V biologickém imunitním systému je spousta prvků, které mají nějakou souvislost s detekcí a eliminací útočníků (T a B lymfocyty, makrofágy, dendritické buňky, cytokiny, TLR, …). Ačkoli jsou dobře popsány a prozkoumány, většinou stoprocentně nevíme, jaké přesně úlohy v imunitním systému hrají. Při aplikaci principů z biologických imunitních systémů do umělých imunitních systémů si proto některé prvky zjednodušíme.

2.2.1 Prvky a principy

Mezi principy, které hrají v imunitních systémech důležitou roli, patří rozpoznávání a paměť. Na tyto dva principy se podíváme v následujících podkapitolách.

2.2.1.1 Rozpoznávání

Ve vrozené imunitě byly přítomny buňky jako neutrofily, makrofágy a dendritické buňky. V adaptivní imunitě hráli hlavní roli T a B lymfocyty. Z pohledu rozpoznávání je pro nás důležitá přítomnost receptorů, které mají schopnost rozpoznat a zachytit určitý vzor. Této části buňky říkáme *komplement* (anglicky antibody). Každý komplement je schopný rozeznat určitý vzor. Tomuto vzoru se říká *antigen*. Princip rozpoznání vzoru antigenem by se dal přirovnat ke klíči a zámku (viz obrázek č. 3).

Komplementy budeme modelovat jako *řetězec bitů délky l*. Jako navázání komplementu na neznámý prvek budeme považovat shodu řetězce bitů na komplementu (detektoru) a řetězce bitů na antigenu (neznámém prvku). Komplement je tedy schopný rozeznat a navázat se jen na jeden určitý antigen. Jako příklad uveďme, že v učebním materiálu v Java apletu pro demonstraci algoritmu pozitivní selekce je detektor (komplement) reprezentován řetězcem délky 63 bitů, což je vlastně matice 7x9 bodů reprezentující jedno písmeno.

K tomu, aby se komplement navázal na antigen, potřebujeme, aby se řetězce sobě navzájem rovnaly – aby byly shodné. To je ale v praxi těžko splnitelný požadavek, proto se využívá podobnosti řetězců. Existuje několik metod, pomocí kterých se dá určit, jak moc si jsou dva prvky podobné. Patří

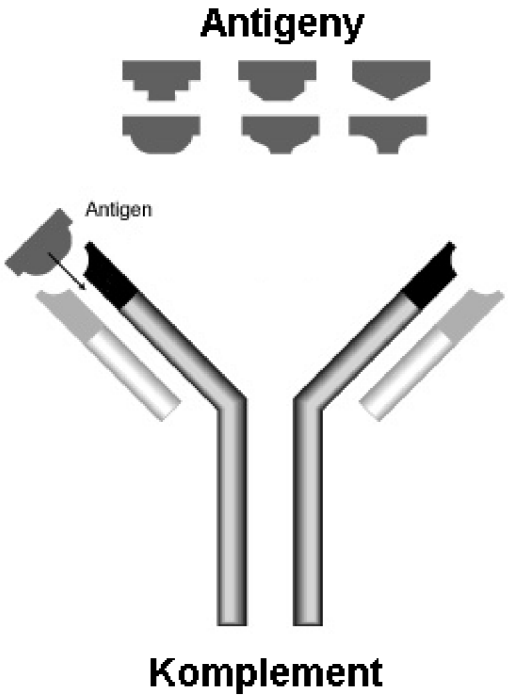
mezi ně například Hammingova vzdálenost, Euklidova vzdálenost nebo tzv. Manhattanská vzdálenost. Euklidova vzdálenost se vypočítá:

$$D = \sqrt{\sum_i (A_i - B_i)^2} \quad \text{a} \tag{1}$$

Manhattanská vzdálenost se vypočítá:

$$D = \sum_i |A_i - B_i|, \tag{2}$$

kde *A1* až *An* a *B1* až *Bn* jsou souřadnice dvou prvků (*A* a *B*). V případě dvou porovnávaných řetězců mohou být hodnoty *A1* až *An* a *B1* až *Bn* hodnoty jednotlivých prvků z daného řetězce. Například pokud uvažujeme dva řetězce písmen anglické abecedy: „abc“ a „eec“, kde hodnota písmene *a* je 1, písmene *b* 2, písmene *c* 3 a písmene *e* 5, potom Euklidova vzdálenost těchto řetězců bude 5 a Manhattanská vzdálenost bude 7.



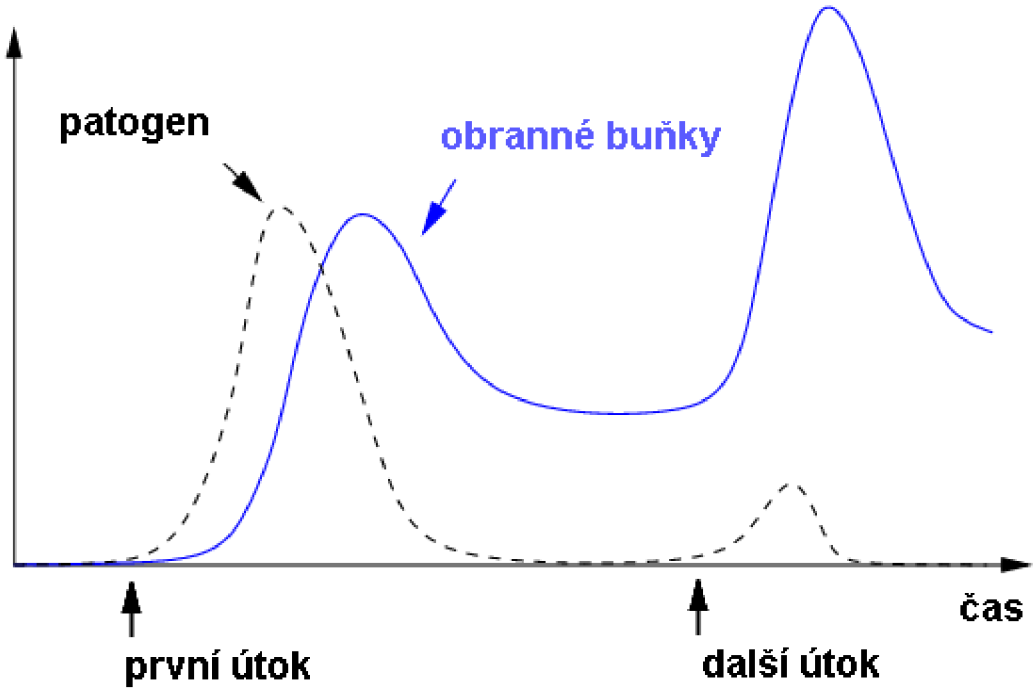
obrázek č. 3 – Komplement (převzato z [22])

Zabývat se budeme nejpoužívanější metodou, kterou je Hammingova vzdálenost. Hammingova vzdálenost má několik verzí, ale nejčastěji určuje, kolik odlišných prvků je ve dvou řetězcích nebo kolik změn musíme provést, abychom z jednoho řetězce dostali ten druhý. Například Hammingova vzdálenost pro řetězce „00110111“ a „00100110“ je 2, pro řetězce „kouzlo“ a „housle“ je 3. V Java apletu pro demonstraci algoritmu negativní selekce se podobnost (afinita) určuje právě pomocí Hammingovy vzdálenosti, která určuje počet různých bitů v řetězci.

2.2.1.2 Paměť

Další důležitý princip, který můžeme najít v imunitních systémech, je schopnost adaptability. Adaptivní část imunitního systému disponuje pamětí.

Pokud se do systému (těla) jedince dostane nový neznámý patogen, který imunitní systém ještě nezná, dendritické buňky ho zachytí a dopraví do mízních uzlin. Zde je podle fragmentů přivezeného patogenu vytvořena a upravena armáda lymfocytů, které opustí mízní uzlinu a specificky vyrazí do boje proti tomuto patogenu. Po úspěšném zvládnutí nákazy se některé lymfocyty přemění v paměťové buňky imunitního systému (prodlouží se jejich životnost) a kolují po systému (těle) jedince pro případ, že by se nákaza vrátila. Pokud se do těla jedince dostane známý patogen, zachytí ho jedna z paměťových lymfocytů, která okamžitě spustí imunitní odpověď – sám lymfocyt se začne množit a vytvoří armádu lymfocytů namířenou proti tomuto patogenu (viz kap. 1.1.2.2). Fragmenty patogenu už tedy nemusí být dopraveny do mízních uzlin, aby tam pomohly vytvořit armádu lymfocytů, což výrazně urychlí imunitní odpověď a jedinec vůbec nemusí poznat, že byl znovu nakažen.



obrázek č. 4 - Útok známého patogenu

V biologických imunitních systémech nemůže být v těle jedince neomezený počet paměťových buněk imunitního systému. Životnost paměťových buněk je tedy určitým způsobem omezena, systém zapomíná. V technických imunitních systémech budeme řešit kompromis mezi paměťovými nároky a rychlosti imunitní odpovědi.

2.2.2 Vlastnosti

Z modelu biologického imunitního systému bychom mohli abstrahovat několik významných vlastností, které najdou své uplatnění při návrhu umělého imunitního systému.

2.2.2.1 Paralelní činnost

Biologický systém dokáže fungovat a reagovat na více místech současně. Pokud se například patogen dostane do systému jedince na určitém místě, imunitní systém bude okamžitě reagovat. Během této reakce může jiný patogen zaútočit z jiného místa a imunitní systém musí být schopen reagovat i na tento a každý další souběžný útok. Tato vlastnost biologického imunitního systému by určitě měla být aplikována například v oblasti bezpečnostních systémů.

2.2.2.2 Komplexnost

Imunitní systém je stavěn tak, aby byl schopný zareagovat na jakýkoli typ vetřelce (vir, bakterie, plíseň, …). Jakákoli buňka může být imunitním systémem napadena (včetně prvků samotného imunitního systému). Zde musíme mít na paměti nebezpečí autoimunity, kdy jsou i vlastní buňky označeny jako cizí a je proti nim vedena imunitní odpověď. Riziko autoimunity snižují algoritmy, které si popíšeme v následující kapitole.

2.2.2.3 Decentralizované řízení

Imunitní systém neobsahuje žádný centrální bod, který by fungování imunitního systému řídil či kontroloval. Absence tohoto prvku má velkou výhodu, protože dysfunkce centrálního řídícího bodu by mohla ovlivnit fungování celého systému. V biologickém imunitním systému je každý prvek zodpovědný sám za svou vlastní funkci.

2.2.2.4 Adaptibilita

Imunitní systém je schopen se přizpůsobovat a v průběhu životního cyklu jedince se učit efektivněji rozpoznávat nové nepřátele. IS není schopen udržet si všechny paměťové buňky, protože musí udržet maximální koncentraci lymfocytů v krevním řečišti. Funguje určitý kompromis mezi počtem prvků a dobou jejich existence v systému.

2.2.2.5 Distribuovaná struktura

Poslední, ale ne nejméně cenná vlastnost, která má svůj význam při uplatnění poznatků z imunitních systémů například v bezpečnostních systémech, je distribuovaná struktura. Jsou dvě části imunitního systému, u kterých je tato vlastnost důležitá. První z nich jsou aktivní prvky, které se podílejí na eliminaci patogenu (nepřítele). V případě počítačové sítě to znamená, že nestačí mít například antivirový systém a firewall jen na prvku, který nás spojuje s internetem, ale také na jednotlivých pracovních stanicích.

Druhá část imunitního systému, která by měla mít distribuovanou strukturu, je paměť. To znamená, že není jeden centrální bod, kde by byla uložena paměť systému, ale je rozložená po celé struktuře. Zde existují dva koncepty, buď mít identické kopie paměti na všech místech, nebo paměť rozložit a mít na každém místě jen určitou část (samozřejmě se zachováním určité redundance, abychom zajistili ještě větší bezpečnost).

V každém případě bychom se měli ujistit, že není žádná nekrytá vstupní cesta pro vetřelce.

2.3 Algoritmy používané v imunitních systémech

V této kapitole si vysvětlíme tři základní algoritmy, které se týkají imunitních systémů [2]. Patří mezi ně algoritmus pozitivní selekce, algoritmus negativní selekce a klonální selekční algoritmus. K tomu, abychom si mohli popsat, jak tyto algoritmy fungují, musíme si vysvětlit několik pojmů.

Jako *množinu S* nebo *SELF* budeme považovat množinu vlastních prvků. V biologických imunitních systémech patří do této množiny buňky vlastního těla. Je to množina, u které se budeme snažit zajistit konzistenci. Pokud se například do těla dostane cizorodá buňka, množina SELF je pozměněna a to musíme být schopni co nejrychleji detekovat.

Jako *řešení* (nebo také detektor) budeme označovat takový prvek, který je schopný rozpoznat určitý problém (např. cizí prvek, nepřátelskou buňku, vir nebo bakterii). V biologických imunitních systémech to byly imunoglobuliny, TLR receptory a lymfocyty typu T a B. Množinu všech možných řešení budeme značit jako *množinu P*. Množina P je tedy množina všech potenciálních detektorů. Zde nastává problém s autoimunitou, což znamená, že prvky množiny P jsou schopny označit za cizí i prvky z množiny SELF. Proto si zavedeme *množinu A*, do které za použití následujících algoritmů vložíme jen žádoucí prvky z P. To znamená, že v množině A budou pouze ty detektory, které rozpoznají a označí jen prvky, které neleží v množině SELF (tzv. non-SELF prvky). V opačném případě můžeme vytvářet takovou množinu A, ve které budou pouze ty detektory, které jsou schopny označit jen prvky z množiny SELF.

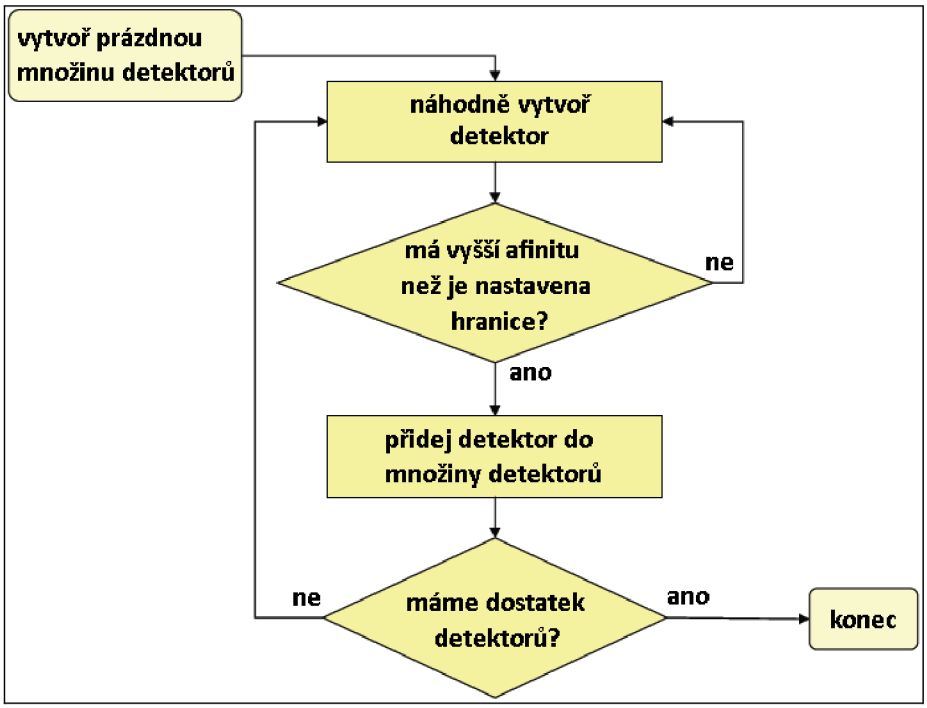
Afinita určuje podobnost, v našem případě úspěšnost (schopnost) rozeznání určitého prvku. Afinitu můžeme určit pomocí několika metod, např. Hammingovou vzdáleností (viz kap. 2.2.1.1).

2.3.1 Algoritmus pozitivní selekce

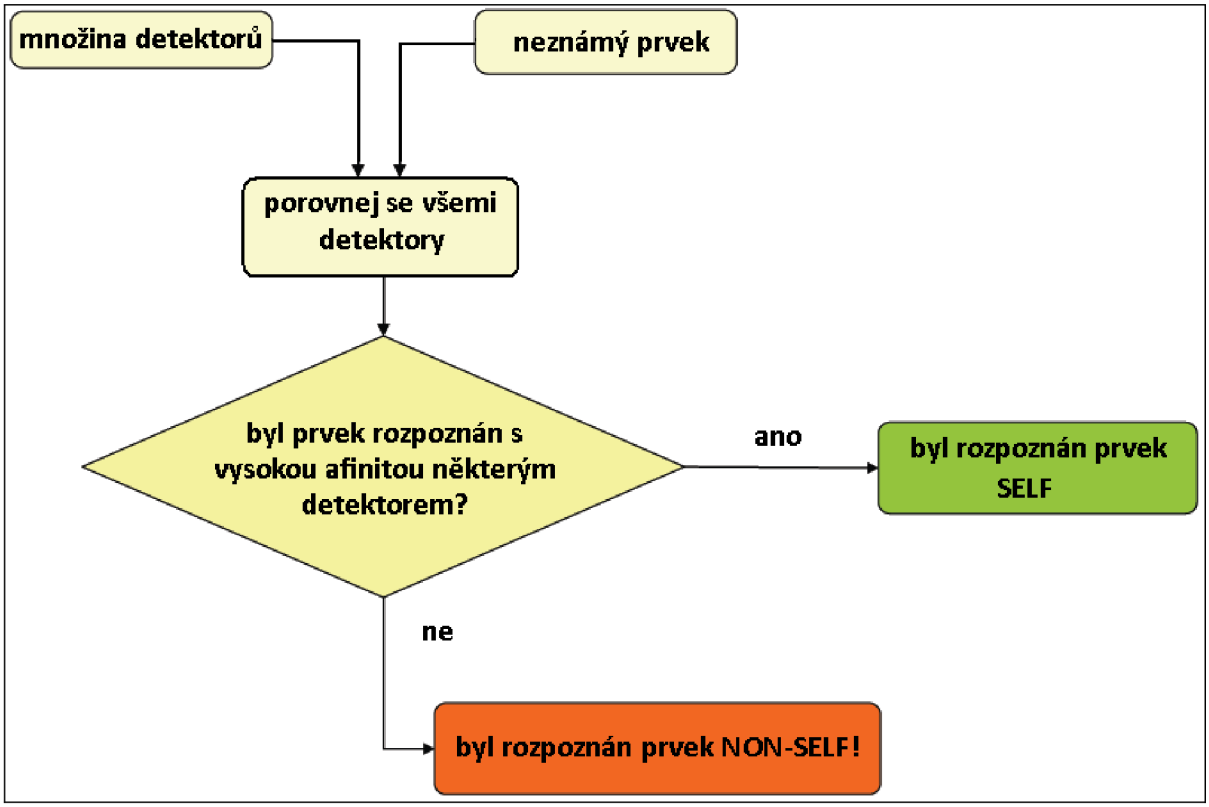
Princip pozitivní selekce se v imunitním systému využíval k odstranění zbytečných a neužitečných lymfocytů, které neměly žádné receptory nebo je měly nějakým způsobem poškozeny. Ve výsledku lymfocyty, které prošly pozitivní selekcí, byly ušetřeny zániku a mohly být použity v efektivní obraně systému jedince.

Algoritmus pozitivní selekce slouží k odstranění prvků z množiny P, které nedokážou rozpoznat žádný z vlastních SELF prvků. Tento algoritmus uplatníme tam, kde vyžadujeme, aby

množina řešení (detektorů) obsahovala jen detektory, které dokážou poznat prvky z množiny SELF. Například v případech, kdy množina P je mnohonásobně větší než množina SELF.



obrázek č. 5 - Pozitivní selekce, algoritmus



obrázek č. 6 - Pozitivní selekce, rozpoznávání

Kroky algoritmu:

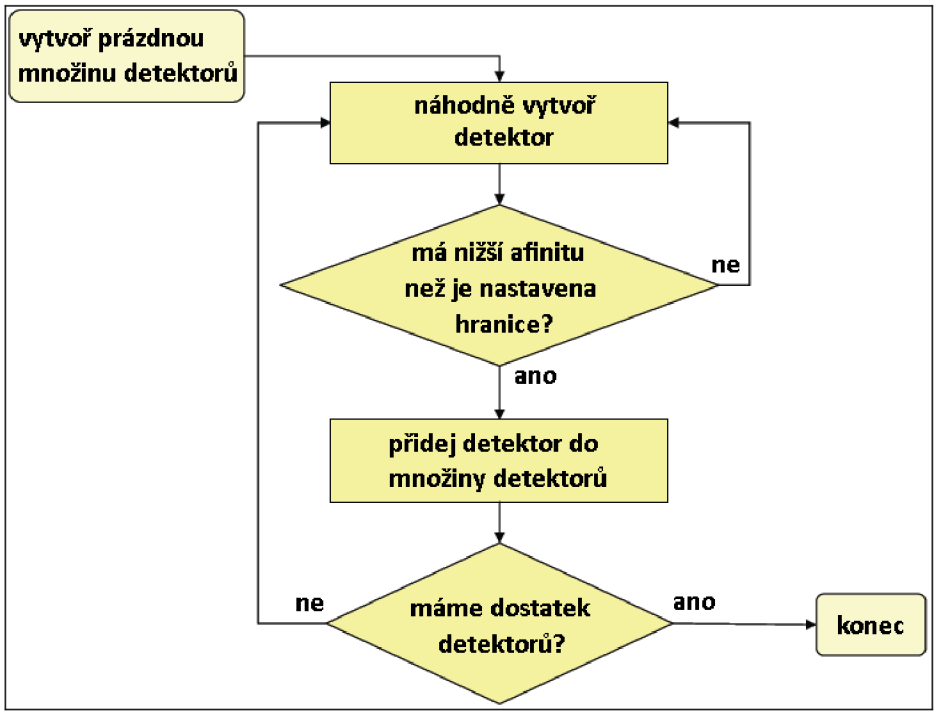
1. *Inicializace prázdné množiny P a určení hranice afinity:* Množina P je množina detektorů, která bude obsahovat jen ty detektory, které jsou schopny navázat se na některý prvek z množiny SELF. Hranice afinity určuje, jaká je minimální podobnost mezi detektorem a některým prvkem z množiny SELF.
2. *Vytvoření náhodného řešení:* Vytvoří se řešení (detektor), který bude úplně náhodný. Zde můžeme algoritmus rozšířit (např. o podmínku, že se nevytvoří detektor, který už někdy vytvořen byl, atd.).
3. *Určíme afinitu tohoto řešení:* Určíme afinitu detektoru postupně aplikací na všechny prvky z množiny SELF a jako afinitu detektoru budeme považovat tu největší z nich.
4. *Pokud má řešení větší afinitu než je určená hranice afinity, přidej ho do množiny P.*
5. *Pokud je množina P dostatečně velká, ukonči algoritmus, jinak pokračuj krokem 2.*

Pomocí algoritmu pozitivní selekce jsme vygenerovali množinu P, ve které jsou detektory (řešení), které dokážou rozpoznat prvky z množiny SELF. Použití této množiny na kontrolu množiny SELF ilustruje diagram na obrázku č. 6. Testovaný prvek je vystaven všem detektorům z množiny P. Během rozpoznávání se určuje stupeň podobnosti (detektoru s neznámým prvkem) – afinita. Pokud je tato afinita alespoň u jednoho detektoru větší než předem nastavená hranice afinity, je tento prvek označen jako prvek množiny SELF. Hranice afinity určuje toleranci, ve které se mohou pohybovat prvky množiny SELF.

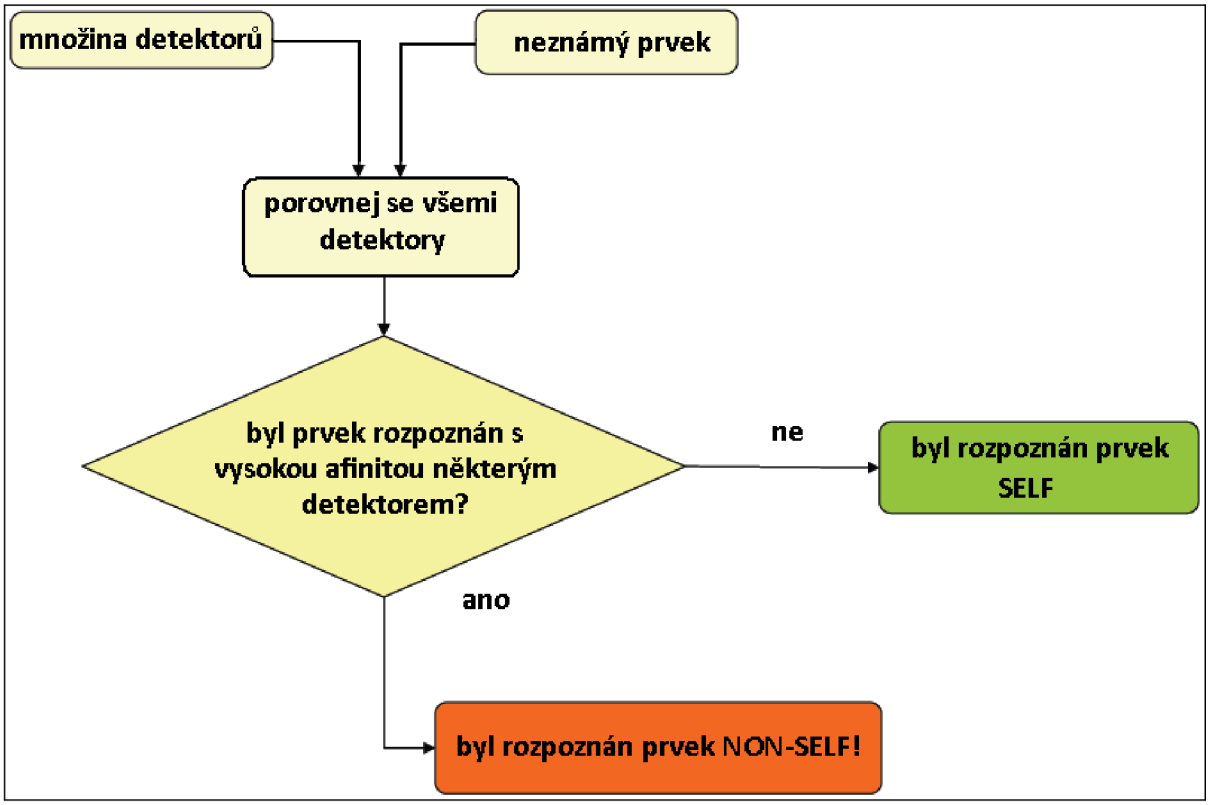
2.3.2 Algoritmus negativní selekce

Princip negativní selekce se v biologických imunitních systémech aplikuje na lymfocyty T. Potom, co jsou lymfocyty T vytvořeny v kostní dřeni, cestují krevním řečištěm do brzlíku. Cestou je na ně aplikován algoritmus pozitivní selekce (jsou zachovány jen ty lymfocyty T, které mají v pořádku receptory a budou schopny dále fungovat – zabíjet buňky, na které se naváží). Když dorazí do brzlíku, dozrávají a je na ně aplikován algoritmus negativní selekce. Z celé populace lymfocytů T zůstanou pouze ty, které jsou schopny navázat se na cizí buňku (odstraní se takové lymfocyty T, které se mohou navázat na prvky množiny SELF). Lymfocyty T se budou vázat jen na buňky, které nepatří do množiny SELF, a budou je zabíjet.

Algoritmus negativní selekce slouží k vybrání těch řešení (detektorů), která jsou schopna rozpoznat pouze cizí prvky (non-SELF prvky). Tento algoritmus tedy odstraňuje ty prvky (ta řešení), které poznají SELF prvek. Používají se tam, kde je množina SELF daleko větší než její doplněk.



obrázek č. 7 - Negativní selekce, algoritmus



obrázek č. 8 - Negativní selekce, rozpoznávání

Kroky algoritmu:

1. *Inicializace prázdné množiny P a určení hranice afinity:* Množina P je množina detektorů, která nebude obsahovat ani jeden detektor, který by byl schopen navázat se

na některý prvek z množiny SELF. Hranice afinity určuje, jaká je maximální podobnost mezi detektorem a některým prvkem z množiny SELF.

2. *Vytvoření náhodného řešení:* Vytvoří se řešení (detektor), které bude úplně náhodné. Zde můžeme algoritmus rozšířit (např. o podmínku, že se nevytvoří detektor, který už někdy byl vytvořen, …).
3. *Určíme afinitu tohoto řešení:* Určíme afinitu detektoru postupně pro všechny prvky z množiny SELF a jako afinitu detektoru budeme považovat tu největší.
4. *Pokud má řešení menší afinitu než je určená hranice afinity, přidej ho do množiny P.*
5. *Pokud je množina P dostatečně velká, ukonči algoritmus, jinak pokračuj krokem 2.*

Pomocí algoritmu negativní selekce jsme vytvořili množinu P, jejíž prvky jsou schopny detekovat prvky, které neleží v množině SELF. Jak tato detekce funguje, si ukážeme na diagramu na obrázku č. 8. Neznámý prvek je vystaven množině P, a pokud je některým detektorem rozpoznán s afinitou vyšší než je předem určená hranice afinity (tolerance), je označen jako prvek non-SELF.

2.3.3 Klonální selekční algoritmus

Klonální selekční princip je způsob, jakým se lymfocyty T a B vypořádají s patogenem, pokud se na nějaký navážou. V případě, že se lymfocyty navážou na patogen, nastartuje se jejich velice aktivní metabolismus a začnou se rychle množit. Na nových lymfocytech se tvoří komplementy právě pomocí klonálního selekčního principu. V konečném důsledku, potom, co se lymfocyt naváže na patogen a rozmnoží se, je vytvořena armáda lymfocytů, které mají schopnost rozpoznat tento určitý patogen a jeho nejbližší příbuzné (podobné) patogeny.

Klonální selekční algoritmus (nebo klonální selekční princip) je teorie, která popisuje, jakým způsobem vznikají řešení (detektory), která jsou schopna efektivně reagovat na určitý problém nebo množinu problémů. Prezentuje myšlenku, ve které jsou vytvářeny a generovány vlastně jen ta řešení, která eliminují a rozpoznají daný problém (problémy), a řešení, jejichž rozpoznávací schopnosti jsou menší, jsou eliminovány. Tento algoritmus nám pomáhá udržet množinu možných řešení co nejefektivnější a nejpřesnější. V praxi můžeme jako problém považovat např. hledání optimální cesty pro obchodního cestujícího. Jako množinu problémů můžeme definovat rozpoznávání písmen, kde rozpoznávání jednoho určitého písmene je jeden problém z této množiny problémů.

Kroky algoritmu:

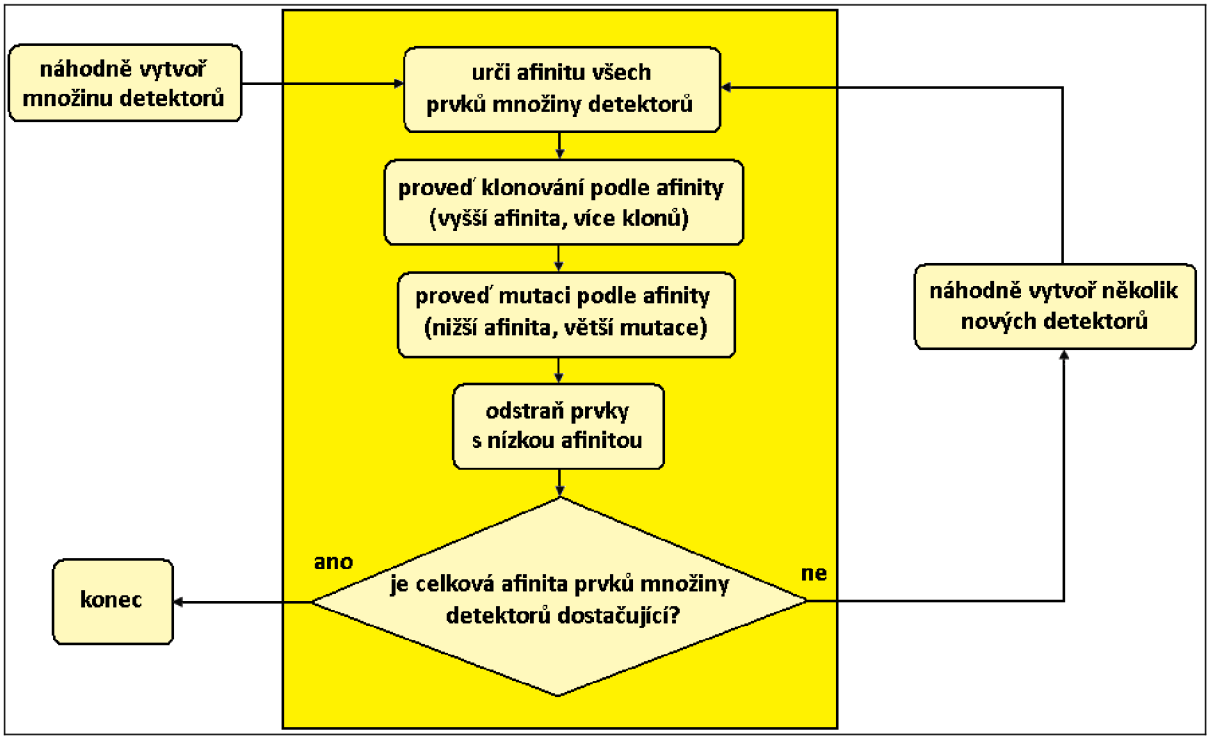
1. *Náhodná inicializace množiny P:* Množina P je množina detektorů (řešení), která po inicializaci obsahuje určitý počet náhodně vytvořených detektorů (řešení).
2. *Test řešení daného problému.* Pro každý problém z množiny problémů udělej:
    a. *Pro každý prvek z množiny P urči jeho afinitu:* Na daný problém aplikujeme postupně všechna řešení a určíme jejich afinitu.

b. *Klonální expanze:* Vytvoříme kopie (klony) prvků v množině P podle jejich afinity. Čím větší je afinita (lepší řešení), tím více klonů tohoto prvku vytvoříme.

c. *Mutace prvků:* Na všechny prvky z množiny P aplikujeme mutaci podle jejich afinity. Čím větší afinita, tím menší stupeň mutace. Jinými slovy dobrá řešení pozměníme málo a špatná se změní více.

d. *Nahraď určitý počet prvků s nízkou afinitou novými náhodnými prvky.*

3. *Pokud je dosaženo určitého kritéria afinity, ukonči algoritmus, jinak pokračuj krokem číslo 2.*



obrázek č. 9 - Klonální selekční princip, algoritmus

3 Přehled základních aplikací

V této kapitole se zaměříme na konkrétní použití principů biologických imunitních systémů v technické oblasti. Podíváme se, které prvky imunitních systému se nejčastěji využívají a kde už byly použity.

3.1 Klasifikace

Po prozkoumání biologického imunitního systémy můžeme vidět, že imunitní systém je velice efektivní a úspěšný v detekci a eliminaci nepřátelských útočníků. Je to systém, který má schopnost velké adaptability a dokáže rozpoznat a označit jako cizí (non-SELF) spoustu nových a ještě nebezpečnějších útočících prvků.

3.1.1 Počítačová bezpečnost

V oblasti počítačové bezpečnosti jsou dvě hlavní oblasti, kde se imunitní systém nasazuje. Jsou to detekce a eliminace virů, červů a trojských koní a detekce průniků do počítačových sítí (neautorizované přístupy, útoky hackerů, DoS útoky, atd.). V obou těchto oblastech se využívá algoritmů (např. algoritmus negativní selekce), ale také vlastností imunitních systémů jako např. decentralizované řízení nebo adaptibilita.

3.1.1.1 Detekce a eliminace virů

Aktuální antivirové programy hledají viry na základě detekce určitého řetězce, který daný vir obsahuje. Další techniky využívají určitá pravidla, podle kterých se specifické viry chovají. Ačkoli jsou tyto metody více či méně spolehlivé, využívají pouze statickou databázi znalostí, kterou musí v pravidelných intervalech obnovovat (nové definice virů, …).

V aplikaci imunitních systémů do antivirového software by mapování mezi imunitním systémem a antivirovým softwarem vypadalo asi následovně. Prvky množiny SELF by obsahovaly informace charakterizující soubory (kontrolní součty, velikost, datum změny, atd.). Komplementy by sloužily jako detektory, které kontrolují soubory a neutralizují napadené soubory [2]. Pokud bychom uvažovali počítače připojené do sítě, mohli bychom vytvořit populaci počítačů, které si budou vyměňovat znalosti (paměťové lymfocyty). Buňky systému by byly běžící procesy na jednotlivých strojích, jež budou pomocí komplementů kontrolovány. Kůže a vrozená imunita by byla reprezentována bezpečnostními mechanismy jako např. hesla, oprávnění přístupu nebo bezpečnostní politiky. Adaptivní imunita by byla tvořena skupinou detektorů, které by kontrolovaly poruchy v normálním chování daného počítače. Jako množinu SELF by tedy reprezentovalo normální chování

a jakákoliv změna v tomto chování by vedla k poplachu. Falešný poplach reprezentuje autoimunitní odpověď.

V publikaci [6] autoři přišli s myšlenkou, jak vytvářet komplementy pro nové a neznáme viry. Tyto komplementy jsou schopny extrahovat z virů jejich otisky a zapamatovat si je. Autoři zároveň uvádí myšlenku, jak co nejvíce omezit autoimunitní odpověď, tedy jak předejít falešným poplachům.

Jak jsme si již uvedli, množina SELF může mít několik podob. Jednotlivé prvky množiny SELF mohou tedy být kontrolní součty jednotlivých souborů na disku, běžící procesy operačního systému, posloupnost volání API funkcí nebo procesů v Unixu [8]. Množinu SELF můžeme také vytvořit na základě způsobu ovládání myši uživatelem (rychlosti pohybu, četnosti stisků tlačítek).

3.1.1.2 Detekce průniků

Jako způsob, který identifikuje spojení (k určení zda, jde o korektní komunikaci nebo útok), se často využívá adresace spojení. Adresace spojení se skládá z trojice hodnot, mezi něž patří zdrojová IP adresa, cílová IP adresa a cílový port. Někdy se do adresace také přidává zdrojový port. Při použití protokolu IPv4 (dnes nejrozšířenější v prostředí internetu) tato trojice tvoří bitový řetězec o délce 78 bitů, ale často se používá komprese a délka tohoto řetězce se zkrátí na 49 bitů [16].

Podle [9] jsou dva hlavní přístupy k implementaci detektorů průniku. První přístup k této problematice tvoří kontrolní mechanizmy k *detekci zneužití přístupu*. Předpoklad k úspěšnému odhalení takovéhoto útoku je znalost otisku nebo vzoru tohoto útoku, například specifická adresace spojení (zdrojová IP adresa je v seznamu „nebezpečných“ IP adres, atd.). Otisk spojení (nebo parametry, vlastnosti) jsou uživatelem přidány do systému a ten pak bude podle těchto pravidel reagovat. Tento postup je často používán v komerčních detektorech průniku [10]. Takový typ detekce průniků je tedy převážně statický, založený na zadaných pravidlech a jeho pravidla a postupy se v průběhu nasazení mění jen minimálně. Jeden z nejsilnějších argumentů proti tomuto přístupu je právě jeho statičnost. Příští generace počítačových útoků bude pocházet od útočníků, kteří se dokážou přizpůsobovat a v čase měnit své chování. Stejně jak se tomu děje v případě biologických imunitních systémů, kde bakterie a viry mutují a vznikají noví a neznámí útočníci.

Druhý přístup k implementaci detektorů průniku tvoří *detekce anomálií*. Před zahájením provozu detektoru průniku je vytvořen profil normální síťové aktivity a jakákoli jiná aktivita (jiné spojení) je považováno za útok. Velká výhoda oproti prvnímu přístupu je, že detektor anomálií v síťovém provozu nemusí znát útok předtím, než je proveden. Nicméně detektor průniku musí znát normální provoz na síti. Pokud odhalí útok, uloží si o něm informace do své databáze znalostí nebo zároveň informuje a upraví databázi znalostí v první části systému (v detektoru zneužití přístupů). Tento přístup je tedy dynamický, protože v průběhu svého působení se zdokonaluje a učí se.

Mapování mezi biologickým imunitním systémem a detektorem průniku s využitím principů imunitních systémů by vypadalo následovně: vrozená imunita je reprezentována detektory známých vzorů a pokusy o zneužití a adaptivní imunita pak bude reprezentována detektory anomálií.

Na Univerzitě v Novém Mexiku tým vedený profesorkou Stephanií Forrestovou vytvořil unixový detektor průniku založený na principech imunitních systémů [8]. Tento software je tzv. Host-based software, což znamená, že je instalován na jednotlivé klientské počítače na síti. Software se po instalaci spustí v režimu učení, kdy si zapamatuje, jak vypadá běžný provoz na síti. Po vytvoření své databáze znalostí se přepne do normálního režimu a pomocí principu imunitních systému a již vytvořené databáze znalostí kontroluje provoz na síti. Program kontroluje TCP spojení, normální spojení klasifikuje jako SELF prvky a cokoli jiného jako non-SELF. Detektory jsou řetězce bitů vytvořené algoritmem negativní selekce. Každý detektor se skládá z trojice zdrojové a cílové IP adresy a cílového portu. Pokud detektor rozpozná prvek s vyšší afinitou než je nastavena hranice, spustí poplach. Z detektorů, které často spouští poplachy, se časem stanou paměťové buňky s nižší hranicí afinity (aby detekce těchto útoků byla příště rychlejší). V tomto systému rozdělili práci na několik počítačů, kde každý kontroluje určitou část komunikace na síti.

Další případ vytvoření detektorů průniku je systém LISYS [11]. Tento systém distribuuje množinu detektorů mezi několik počítačů, kde každý počítač pracuje nezávisle – sám určí, zda se jedná o TCP spojení z množiny SELF nebo jde o útok. Množinu detektorů vytváří pomocí algoritmu negativní selekce v režimu neboli v čase učení. Pokud je skupinou detektorů rozpoznána nekorektní komunikace (více než *r* nekorektních spojení, kde *r* je nastavená hranice), vyvolá poplach.

Dasgupta a Gonzales [12] vytvořili detektor průniku s aplikací klonálního selekčního algoritmu a pozitivní selekce. Množina SELF prvků (trojice IP adres a portu) je generována v učebním období pomocí algoritmu pozitivní selekce.

3.1.1.3 Další oblasti

Jako další příklad využití imunitních systémů v oblasti počítačové bezpečnosti bychom mohli zmínit anti-spamové filtry. S růstem rychlosti připojení k internetu firem rostou větší možnosti pro spamery. Organizace MAAWG provedla průzkum 100 miliónů e-mailových schránek a došla k závěru, že na konci roku 2005 80-85% e-mailové komunikace tvořila nevyžádaná pošta [17]. Potřeba lepších a efektivnějších anti-spamových filtrů je proto nutností. Oda a White ve své práci uvedli, jak efektivně implementovat imunitní systémy do anti-spamových filtrů [18].

3.1.2 Detekce anomálií a chyb

Distributivní řízení imunitního systému inspirovalo například Ishida [13], který ve své práci navrhl systém agentů pro kontrolu chyb systému. Hlavní charakteristikou jeho systému byla přizpůsobivost na měnící se prostředí a schopnost přizpůsobovat množinu SELF.

Další přístup k problému detekce anomálií a chyb je vytvořit systém, který bude předvídat budoucí chování systému nebo procesu podle již známých informací z minulosti. Tento přístup popsal Lane [14], který zkoumá interakci uživatele s počítačem. Model vytváří sekvence akcí, které by

uživatel mohl s určitou pravděpodobností provést. Pokud uživatel provede akci, která má malou pravděpodobnost, systém spustí poplach.

3.2 Optimalizace

Optimalizace je proces, jak přeměnit určitý systém tak, aby pracoval co nejefektivněji. Tento proces často upravuje podmínky a vstupní hodnoty daného systému a sleduje jejich výsledek. Umělé imunitní systémy nabízí několik vlastností a principů, které nám v určitých oblastech optimalizaci usnadní.

3.2.1 Problém obchodního cestujícího

Problém obchodního cestujícího (traveling salesman problem) je problém (úloha), ve které máme zadáno několik měst a jednoho nebo několik obchodních cestujících. Úkolem je, aby tito obchodní cestující navštívili každé město právě jednou a aby délka jejich cestování mezi městy byla co nejkratší. Tímto problémem za využití umělých imunitních systémů se zabývají ve své publikaci Endoh, Toma a Yamada [20]. Antigen reprezentuje informace o městech a jednotlivé komplementy jsou trasy obchodních cestujících. Za použití umělých imunitních systémů tedy hledá optimální cestu mezi městy.

3.2.2 Doporučování filmů

Cayzer [20] navrhl systém pro doporučování filmů uživatelům. Tento systém využívá prvky umělých imunitních systému, díky kterým vytvoří každému uživateli seznam „top ten“ filmů, které by se danému uživateli mohly líbit.

Systém pro doporučování filmů funguje tak, že každému uživateli nejdříve vytvoří profil. Jednotliví uživatelé si ve svém profilu nastaví filmy, které již viděli, a každému filmu přidělí hodnocení (kladné nebo také záporné). V biologickém imunitním systému je uživatel, jemuž chceme zobrazit doporučení, reprezentován jako antigen a ostatní uživatelé jsou reprezentováni jako komplementy. Systém tedy hledá skupinu komplementů, které jsou s určitou afinitou podobné antigenu, a na základě těchto komplementů (uživatelů se stejným vkusem na filmy) vytvoří doporučení filmů, které by se uživateli mohly líbit.

Čím více uživatelů bude tento systém využívat, tím přesnější bude produkovat doporučení.

3.2.3 CLONALG

De Castro a von Zuben upravili klonální selekční algoritmus, který pojmenovali CLONALG [15]. Hlavní využití tohoto algoritmu je rozpoznání vzoru a optimalizace funkcí. Rozdíl mezi použitím pro rozpoznání vzoru a použitím pro optimalizaci je v definici množiny SELF. V případě rozpoznávání

vzoru budou jako prvky množiny SELF vzory, které hledáme, a v případě optimalizace bude prvek množiny SELF funkce f(), kterou optimalizujeme. Potom prvky množiny P budou jednotlivé hodnoty funkce f().

4 Popis implementace výukového materiálu

Druhá část bakalářské práce na téma „Umělé imunitní výpočetní systémy“ spočívala ve vytvoření výukového materiálu pro studenty. Výukový materiál byl vytvořen jako webová stránka se třemi Java aplety, které demonstrují základní algoritmy používané v imunitních systémech. V této kapitole se podíváme na popis implementace tohoto výukového materiálu.

Výukový materiál je umístěn na *http://www.stud.fit.vutbr.cz/~xneuwi00/AIS/*.

4.1 Popis webových stránek

Výukový materiál se skládá z pěti HTML stránek. Každá stránka je ve formátu XHTML 1.0 Transitional a kódování češtiny je ISO-8859-2. Validita stránky a jejích součástí byla zkontrolována validátorem [23]. Pro úpravu grafické podoby stránky byly využity CSS kaskádové styly.

Výukový materiál je rozdělen do pěti základních souborů. Tyto soubory jsou *index.html*, *bis.html*, *tis.html*, *praxe.html* a *zaver.html*. Každá z těchto stránek obsahuje obsah formou odkazů na ostatní části.

První část výukového materiálu je výchozí webová stránka s názvem *index.html*. Na této stránce je úvod k danému tématu. Název *index.html* je použit z důvodu jednoduššího publikování na webových serverech, protože tento název je nejčastěji nastaven jako výchozí webová stránka. Na stránce *tis.html* jsou tři Java aplety. Každý z těchto apletů demonstruje jeden z algoritmů používaných v imunitních systémech.

Jednotlivé webové stránky mají následující strukturu. Jako první je uvedena deklarace typu dokumentu (*<!DOCTYPE html public ...*). Dále se zde nachází hlavička, ve které je uveden text titulku prohlížeče a odkaz na soubor s CSS styly. Za hlavičkou následuje tělo dokumentu. Tato část je rozdělena na další čtyři části. Nejdříve je uveden nadpis, který tvoří název bakalářské práce. Dále následuje obsah výukového materiálu formou nečíslovaného seznamu v podobě menu, který odkazuje na jednotlivé stránky materiálu. Třetí část tvoří vlastní text stránky. Pro nadpisy jsou použity standardní značky *<H1>* až *<H5>*, pro text *<P>*. Vzhled a forma těchto položek je definována různými CSS styly. Na konci těla dokumentu je pro lepší orientaci znovu obsah.

CSS kaskádové styly jsou uloženy v souboru *styly.css*. Pro tvorbu kaskádových stylů byl použit standard CSS 2.1. Soubor *styly.css* byl zkontrolován validátorem [23], který nenalezl žádné odchylky od použitého standardu.

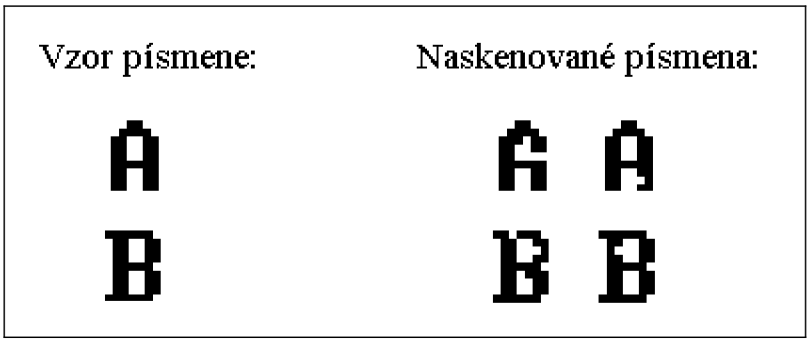
4.2 Java aplety

Pro tvorbu Java apletů bylo použito vývojové prostředí *NetBeans IDE 5.5* a virtuální stroj *JavaVM verze 1.6.0*. Informace o programovacím jazyce Java byly čerpány z [24] a [25].

Java aplety jsou uloženy v oddělených složkách (na CD v příloze). Jednotlivé Java aplety jsou členěny na několik tříd, aby se zajistila přehlednost, určitá abstrakce a dodržovala dobrá kultura programového kódu [26].

4.2.1 Algoritmus pozitivní selekce

První z Java apletů je aplet demonstrující algoritmus pozitivní selekce. Jako příklad byl vybrán model skenování tištěného textu. V porovnání s vytištěným textem, text naskenovaný může obsahovat určité odchylky a nedostatky. Při rozlišování jednotlivých písmenek se tedy vzor nemusí stoprocentně shodovat s naskenovaným písmenkem (viz obrázek č. 10). Pro rozlišení naskenovaného písmene použijeme tedy algoritmus pozitivní selekce.



obrázek č. 10 - Skenovaný text

4.2.1.1 Implementace

Tento aplet obsahuje celkem čtyři třídy. Třída *MainPositive* obsahuje rozložení formuláře s jednotlivými prvky (tlačítka, posuvníky, popisky, atd.) a základní funkce programu (obsluha stisku tlačítek, inicializaci apletu, obsluha myši, atd.). Tato třída dále obsahuje instance množiny P a množiny S. Tyto množiny jsou implementovány pomocí třídy *Mnozina*, která tvoří rozšíření třídy *ArrayList* pomocí jednoduché dědičnosti. Funkce a metody, které třídu *ArrayList* rozšiřují, jsou například metody pro generování a tisk jednotlivých prvků nebo funkce pro výpočet afinity daného prvku. Množina obsahuje několik prvků třídy *Cell*. *Cell* je další implementovaná třída, která obsahuje informace o jednotlivých písmenech (a jejich variantách). Například bitové pole, které určuje vzhled daného obrazce, hodnotu typu boolean určující zda je prvek přeškrtnutý, atd. Poslední třída v Java apletu je třída *Konstanty*. V této třídě jsou důležité konstanty, používané v apletu (výška a šířka písmene, atd.) a je zde implementován generátor náhodných čísel.

4.2.1.2 Návod k použití

Než začneme tento Java aplet používat musíme nastavit některé parametry. *Velikost množiny SELF* udává počet prvků v množině SELF. Možné hodnoty jsou v uzavřeném intervalu 1 až 4. *Hranice afinity* určuje, kolik procent pixelů naskenovaného písmene se musí rovnat pixelům některého písmene v množině SELF. Čím je tedy hranice afinity nižší, tím větší odchylky jsou tolerovány, ale zvětšuje se riziko chybného rozpoznání písmene. Doporučená hodnota je 90%. Poslední parametr určuje rychlost animace apletu – doporučená hodnota pro demonstraci je 200 ms.

Po nastavení těchto parametrů (nebo zachováním výchozích hodnot) se Java aplet spustí pomocí tlačítka *start*. Po stisku tohoto tlačítka proběhne první krok algoritmu (vytvoření množiny SELF). K druhému kroku algoritmu pozitivní selekce se dostaneme po opětovném stisku tlačítka, kdy proběhne generování množiny P. Pro lepší demonstraci tohoto algoritmu je zobrazena jen část množiny P. Dalším krokem je výběr prvků z množiny P, které vyhovují námi zadané afinitě (odstranění nevyhovujících prvků).

Poslední funkcí demonstračního Java apletu je možnost vložit vlastní prvek (písmeno) a otestovat, jestli ho aplet rozpozná a s jakou afinitou. Po stisku tlačítka *další krok* se zobrazí pole, do kterého můžeme nakreslit vlastní prvek. Po nakreslení se otestuje pomocí tlačítka *otestuj*.

Algoritmus můžeme ukončit pomocí tlačítka *ukonči* a tím ho resetovat do výchozího stavu.

4.2.2 Algoritmus negativní selekce

Další Java aplet demonstruje algoritmus negativní selekce. Jako prvky jsou považovány sedmi-bitové řetězce, kde *1* je reprezentována červenou barvou a *0* zelenou. Vytvoříme si tedy množinu SELF a vygenerujeme množinu P. Podle nastavené afinity a pomocí algoritmu negativní selekce vytvoříme množinu A.

4.2.2.1 Implementace

Implementace tohoto Java apletu byla rozdělena do pěti tříd. Hlavní třída, která inicializuje a spouští aplet, se jmenuje *MainNegative*. Tato třída dědí od třídy *Java.applet.Applet* a rozšiřuje ji o několik funkcí a metod. Například o instance množin (SELF, P a A), o metody vykreslování nebo o obsluhu jednotlivých kroků algoritmu. Jednotlivé množiny jsou implementovány pomocí třídy *Mnozina*, která dědí od třídy *ArrayList*. Rozšíření třídy *ArrayList* spočívá v implementaci metod pro generování prvků, funkce pro test, zda prvek už množina obsahuje nebo například metody pro tisk jednotlivých prvků. Každý prvek je třídy *Cell*. Data prvku jsou uloženy v sedmi-bitovém poli. Třída *Cell* dále obsahuje funkce pro spočítání afinity mezi dvěma prvky a metody pro tisk prvku na formulář apletu.

Důležitá třída pro korektní vykreslení animace apletu je třída *Semafor*. Samotné otestování, zda prvek patří do množiny A nebo ne, je v animaci rozdělen do několika podkroků (zobrazení černé šipky, přesun prvku do středu formuláře, zobrazení zelené nebo červené šipky a případně zařazení do

množiny A). K tomu, aby aplet správně vykreslil tuto animaci, je zapotřebí třídy *Semafor*, která určuje, ve kterém podkroku se právě nacházíme.

Poslední třída, která byla v tomto apletu vytvořena, je třída *Konstanty*. Tato třída obsahuje několik důležitých konstant, které aplet využívá při demonstraci algoritmu negativní selekce. Například velikost buňky, generátor náhodných čísel nebo je zde implementována mocnina (pro výpočet využívá rekurzi).

4.2.2.2 Návod k použití

Před spuštěním algoritmu můžeme nastavit dvě hodnoty. *Rychlost* animace a *afinitu*. Afinita může nabývat hodnot 0, 1 a 2. Tyto hodnoty představují maximální počet rozdílných bitů ve vyřazených prvcích (v porovnání s některým ze SELF). V případě nastavení na hodnotu 0 budou z množiny P vyřazeny jen ty prvky, které jsou shodné s některým prvkem z množiny SELF. Rychlost animace se dá měnit posuvníkem i během průběhu kroku algoritmu.

Pomocí tlačítka *spustit* se algoritmus spustí. Prvním krokem je vytvoření množiny SELF. Po opětovném stisku tlačítka se přejde k dalšímu kroku algoritmu. Tím je generování množiny P, popř. generování množiny A. Po stisku tlačítka *konec* se aplet resetuje do původního stavu.

4.2.3 Klonální selekční algoritmus

Jako téma pro demonstraci klonálního selekčního algoritmu byl vybrán problém obchodního cestujícího. Obchodní cestující musí projít určitý počet měst, z nichž každé právě jednou, a délka jeho cesty musí být co nejkratší. Pro hledání optimální cesty (řešení) použijeme právě klonální selekční algoritmus.

4.2.3.1 Implementace

Aplet pro demonstraci klonálního selekčního algoritmu je rozdělen do šesti tříd. Ve třídě *MainClonal* jsou implementovány základní funkce apletu (inicializace, obsluha klávesnice a myši, …) a rozložení formuláře. Třída *Konstanty* obsahuje používané konstanty jako např. výška a šířka mapy nebo generátor náhodných čísel. Třída *Mapa* dědí od třídy *ArrayList* a rozšiřuje ji o funkce pro generování určitého počtu náhodně umístěných měst. Každé město je reprezentováno třídou *Souradnice*, která obsahuje souřadnice a funkci pro výpočet vzdálenosti od jiného města.

Další třídou je třída *Cesta*. Tato třída reprezentuje cestu obchodního cestujícího (řešení našeho problému). Obsahuje tedy posloupnost měst v pořadí, ve kterém je obchodní cestující navštíví. Třída *Cesta* dále obsahuje funkci, která vrátí délku cesty. Tato hodnota je v průběhu životního cyklu tohoto objektu vyžadována několikrát, proto je třída *Cesta*, resp. funkce pro výpočet délky cesty, navržena tak, aby tuto hodnotu spočítala jen jednou a zapamatovala si ji, případně po změně v pořadí měst, přepočítala znovu. Množina P je implementována jako třída *MnozinaP*, což je vlastně *ArrayList*

obsahující jako prvky jednotlivé cesty. Dále obsahuje například funkci pro mutování (změnu v pořadí měst u jednotlivých cest).

4.2.3.2 Návod k použití

Na formuláři Java apletu se nacházejí dva posuvníky, pomocí kterých se nastavuje *rychlost animace* a *počet měst* na mapě. Počet měst může nabývat hodnot od 5 do 15 měst, výchozí hodnota je 8. Čím méně měst bude na mapě, tím rychleji algoritmus dojde k nejoptimálnějšímu řešení. Dále se na formuláři nacházejí dvě tlačítka. Tlačítko *nová mapa* generuje podle počtu měst novou mapu a druhé tlačítko řídí kroky algoritmu.

Po stisku tlačítka *start* se vygenerují náhodná řešení. Pomocí tlačítka další krok se postupně prochází mezi kroky algoritmu. Mezi kroky algoritmu se tlačítkem *reset* může algoritmus resetovat a vygeneruje se nová mapa.

4.3 Poznámky k implementaci

Učební materiál a Java aplety byly otestovány na počítači s touto HW konfigurací: procesor Intel Pentium IV centrino 1.8 GHz, operační paměť 1024 MB, grafická karta ATI Radeon 9700 128 MB, LCD displej s rozlišením 1280x800 pixelů. Softwarová konfigurace: operační systém MS Windows Vista Business v anglické verzi s posledními aktualizacemi. Správné kódování češtiny bylo ověřeno na studentském serveru eva.fit.vutbr.cz.

Java aplety (jejich programové kódy) byly implementovány s ohledem na dobrou kulturu kódu a srozumitelnost [26], které napomáhají často se vyskytující komentáře. Formuláře Java apletů byly navrženy intuitivně, aby byly snadno ovladatelné i bez nutnosti předchozího pročtení návodů k použití. Programový kód byl optimalizován, aby co nejefektivněji využíval paměť a aby byl co nejméně náročný na čas procesoru.

4.4 Závěr

Tento učební materiál, který je umístěn na: *http://www.stud.fit.vutbr.cz/~xneuwi00/AIS/,* podrobně vysvětluje problematiku imunitních systémů a prezentuje možnosti nasazení prvků a principů imunitních systémů v oblasti techniky. Pomocí tří Java apletů demonstruje algoritmy používané v imunitních systémech.

Vytvořený učební materiál by se dal zařadit do předmětů *Biologií inspirované počítače* nebo částečně do *Aplikované evoluční algoritmy*.

5 Závěr

V této bakalářské práci na téma *Umělé imunitní výpočetní systémy* jsme se podívali na problematiku umělých imunitních systémů. Vysvětlili jsme si, jak fungují biologické imunitní systémy, jaké jsou jejich důležité vlastnosti a principy, a jak tyto poznatky aplikovat v technických imunitních systémech. V další kapitole jsme shrnuli a nastínili přehled o tom, kde už byly prvky a principy imunitních systému použity v technice a ve kterých oblastech ještě můžeme imunitní systém použít (nebo jeho část).

Další část této bakalářské práce tvoří tvorba výukového materiálu pro studenty na toto téma. Tvorba výukového materiálu spočívala ve vytvoření webových stránek, které budou srozumitelně a přehledně vysvětlovat imunitní systémy a které budou snadnou pomůckou pro vysvětlení této problematiky. Součásti výukového materiálu jsou tři Java aplety, které demonstrují algoritmy používané v imunitních systémech. Tyto aplety byly tvořeny s ohledem na jednoduchou ovladatelnost a snadnou pochopitelnost daného problému.

Umělé imunitní systémy jsou velice rozsáhlou oblastí a existuje ještě spousta odvětví na poli techniky, kde by se tyto poznatky daly využít. Další studium této oblasti by určitě bylo zajímavé z pohledu témata na diplomovou práci. Z obsahu bakalářské práce, jako teorie k umělým imunitním systémům, by se dalo vycházet při tvorbě systému, který by využíval a byl postaven na principech imunitních systémů (např. při tvorbě systému pro detekci průniků na síti, antivirového nebo antispamového filtru), a porovnání výsledku chování tohoto systému s výsledky chování obdobného, již existujícího (např. komerčního) systému, který neobsahuje prvky imunitních systémů.

Literatura

[1] O'NEILL, L. A. J. *Imunitní systém včasné výstrahy. Scientific american*. únor 2006, s. 44-51.

[2] CASTRO, L. N. de, TIMMIS, J. *Artificial Immune Systems : A New Computational Intelligence Approach*. [s.l.] : Springer, 2002. 380 s.

[3] Immune System. *Science Aid* [online]. 2006 [cit. 2007-04-23]. Dostupný z WWW: <http://www.scienceaid.co.uk/biology/humans/immunesystem.html>.

[4] *University of New Mexico : Computer Science* [online]. 2006 [cit. 2007-04-23]. Dostupný z WWW: <http://www.cs.unm.edu/>.

[5] *Patient and Family Handbook of the Immune Deficiency Foundation.* USA : The Immune Deficiency Foundation. 1993.

[6] KEPHART, J. O., SORKIN, G. B., SWIMMER, M., WHITE, S. R. *Blueprint for a Computer Immune System*. 1999.

[7] HOAR, R. *Applications of Immune System Computing*. University of Calgary : Department of Computer Science. 2003.

[8] SOMAYAJI, A., FORREST, S., HOFMEYR, S. A., LONGSTAFF, T. *A Sense of Self for Unix Processes.* IEEE Symposium on Security and Privacy. 1996, s. 120-128.

[9] MIDDLEMISS, M. *Framework for Intrusion Detection Inspired by the Immune System.* University of Otago : Information Science Department. 2005.

[10] JACKSON, K. *Intrusion detection system product survey*. Los Alamos Nation Laboratory : Research report LA-UR-99-2882. 1999.

[11] BATHROP, J., FORREST, S., GLICKMAN, M. *Revisiting lisys : Parameters and normal behaviour.* Proceedings of the Congress on Evolutionary Computation. 2002, vol. 2, s. 1045-1050.

[12] DASGUPTA, D., GONZÁLEZ, F. *An immunity-based technique to characterize intrusions in computer networks.* IEEE Transaction on Evolutionary Computation. 2002, vol. 6, is. 3, s. 281-291.

[13] ISHIDA, Y. *An immune network approach to sensor-based diagnosis for self organization.* Complex Systems. 1996, vol. 10, is. 1, s. 73-90.

[14] LANE, T. *Hidden markov models for human/computer interface modeling.* IJCAI-99 Workshop on Learning About Users. 1999, s. 35-44.

[15] CASTRO, L. N. de, ZUBEN, F. J. von. *The clonal selection algorithm with engineering applications.* Proceedings of GECOO´00. 2000, s. 36-37.

[16] HOFMEYR, S. A., FORREST, S. *Immunity by Design : An Artificial Immune System.* University of New Mexico : Department of Computer Science. Albuquerque.

[17] MAAWG Issues First Global Email Spam Report. *News & Events - MAAWG Release* [online]. 2006 [cit. 2007-04-23]. Dostupný z WWW: <http://www.maawg.org/news/maawg060308>.

[18] ODA, T., WHITE, T. *Crossroads Magazine : Spam Detection using an Artificial Immune System* [online]. 2004, is. 4. Dostupný z WWW: <http://terri.zone12.com/doc/academic/crossroads/>.

[19] CAYZER, S. *Artificial Immune System.* HP Labs Bristol. 2003.

[20] ENDOH, S., TOMA, N., YAMADA, K. *Immune Algorithm for n-TSP*. IEEE International Conference on 1998 : Systems, Man and Cybernetics. 1998, vol. 4, s. 3844-3849.

[21] BARTŮŇKOVÁ, J., ŠEDIVÁ, A., HÖLZELOVÁ, E. *Primární imunodeficience : Příručka pro pacienty a jejich rodiny*. FN Motol, Praha, 2. LF UK : Ústav imunologie. 1999. Dostupný z WWW: <http://www.tigis.cz/Knihy/imuno/>.

[22] *Antibody* [online]. 2007 [cit. 2007-04-28]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Immunoglobulin>.

[23] *W3C : Markup Validation Service*. Dostupný z WWW: <http://validator.w3.org/>.

[24] HATINA, P., JELÍNEK, L. *Programování v jazyku Java*. Linuxsoft.cz. 2004-2007. Dostupný z WWW: <http://www.linuxsoft.cz/article.php?id_article=244>.

[25] SEMECKÝ, J. *Naučte se Javu.* Interval.cz. 2002-2003. Dostupný z WWW: <http://interval.cz/clanky/naucte-se-javu-uvod/>.

[26] MARTINEK, D. *Nedělejte zbytečné chyby!* Brno. CZ. 2004. s. 39. Dostupný z WWW: <http://www.fit.vutbr.cz/~martinek/papers/noerrors.pdf>.

[27] *Ebola* [online]. 2007 [cit. 2007-05-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Ebola/>.

Seznam příloh

Příloha 1. CD s učebním materiálem