

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA

Provozně ekonomická fakulta

Katedra informačního inženýrství



**PROBLEMATIKA ZPĚTNÉ VAZBY V PODMÍNKÁCH
E-LEARNINGU**

Disertační práce

Autor: Ing. Dana VYNIKAROVÁ

Školitel: Prof. RNDr. Jiří Vaníček, CSc.

Obor: Informační management

Praha 2010

Touto cestou bych chtěla poděkovat svému školiteli Prof. RNDr. Jiřímu Vaničkovi, CSc. za poskytnutí cenných rad a názorů a pomoc při tvorbě mé disertační práce. Poděkovat bych rovněž chtěla Katedře informačního inženýrství Provozně ekonomické fakulty ČZU v Praze za zázemí poskytnuté v průběhu psaní mé disertační práce.

Praha, Prosinec 2010

Dana Vyníkarová

Obsah

1. ÚVOD	- 4 -
2. CÍLE PRÁCE, MOTIVACE A POUŽITÁ METODIKA	- 6 -
2.1 CÍLE PRÁCE	- 6 -
2.2 MOTIVACE PRÁCE	- 8 -
2.3 POUŽITÁ METODIKA.....	- 8 -
2.3.1 Terminologie	- 9 -
3. E-LEARNING A JEHO VÝZNAM.....	- 10 -
3.1 E-LEARNING.....	- 10 -
3.1.1 Vznik a vývoj e-learningu	- 10 -
3.1.2 Vymezení pojmu e-learning	- 10 -
3.1.3 Pojetí elektronické výuky	- 11 -
3.1.4 Výhody a omezení elektronické výuky	- 12 -
3.2 ZPĚTNÁ VAZBA V E-LEARNINGU.....	- 13 -
3.2.1 Možnosti zpětné vazby.....	- 14 -
3.3 PROBLEMATIKA POČÍTAČOVÉHO TESTOVÁNÍ.....	- 18 -
3.3.1 Návrh hodnocení odpovědí studentů při počítačovém testování.....	- 21 -
4. MODEL Y VÝPOČTU A PROCESU	- 26 -
4.1 VÝUKOVÝ SYSTÉM	- 26 -
4.2 FORMÁLNÍ MODELOVÁNÍ VÝUKOVÉHO PROCESU.....	- 31 -
4.2.1 Konečný automat.....	- 31 -
4.2.1.1 Popis činnosti konečného automatu.....	- 32 -
4.2.1.2 Ukázka činnosti konečného automatu	- 33 -
4.2.1.3 Modelování výukového systému konečným automatem	- 35 -
4.2.2 Zásobníkový automat	- 36 -
4.2.2.1 Popis činnosti zásobníkového automatu	- 37 -
4.2.2.2 Síla modelu zásobníkového automatu	- 38 -
4.2.2.3 Modelování výukového systému zásobníkovým automatem	- 39 -
4.2.3 Turingův stroj.....	- 39 -
4.2.3.1 Modelování výukového systému Turingovým strojem	- 42 -
4.2.4 Petriho síť.....	- 42 -
4.2.4.1 Změna stavu systému.....	- 44 -
4.2.4.2 Vybrané vlastnosti a typy Petriho sítí	- 46 -
4.2.4.3 Modelování inteligentních systémů pomocí Petriho sítí.....	- 48 -
4.3 MOŽNOSTI MODELOVÁNÍ VÝUKOVÉHO SYSTÉMU	- 51 -
4.3.1 Modelování výukového systému pomocí barevné Petriho sítě	- 51 -
4.3.1.1 Interpretace sítě.....	- 52 -
4.3.1.2 Systém přechodů.....	- 53 -
4.3.1.3 Struktura systému	- 55 -
4.3.2 Možnosti řízení výukových procesů s použitím Petriho sítí a fuzzy logiky.....	- 56 -
4.3.3 Technologie Petriho sítí při návrhu e-learningových kurzů	- 57 -
4.3.4 GLIF model průchodu studenta výukovým systémem.....	- 57 -
4.3.4.1 Základní typy vrcholů.....	- 58 -
4.3.4.2 Rozhodovací kritéria.....	- 59 -
4.3.4.3 Model průchodu studenta výukou	- 59 -
4.3.5 Využití Petriho sítí k analýze chování studentů v e-learningovém prostředí ...	- 62 -
4.3.6 Výzkum zatížení e-learningového systému založeného na Petriho sítích.....	- 64 -
4.3.7 Návrh e-learningového systému pomocí hierarchického modelu Petriho sítí ...	- 65 -

4.3.8	Návrh výukového procesu pomocí zrcadlicí Petriho sítě	- 67 -
5.	NÁVRH ZOBECNĚNÉ PETRIHO SÍTĚ.....	- 69 -
5.1	VYUŽITÍ ZOBECNĚNÉ PETRIHO SÍTĚ PRO E-LEARNING.....	- 69 -
5.2	POZNATKY A INFORMACE	- 70 -
5.2.1	Sémantická teorie informace	- 70 -
5.3	INTELIGENTNÍ SYSTÉM.....	- 72 -
5.4	ZOBECNĚNÁ PETRIHO SÍŤ	- 73 -
5.4.1	Vývoj zobecněné Petriho sítě.....	- 77 -
5.5	PŘÍKLADY UŽITÍ ZOBECNĚNÉ PETRIHO SÍTĚ NA KONKRÉTNÍ VÝUKOVÁ TÉMATA .-	83 -
5.5.1	Příklad konstrukce kombinačního logického obvodu pomocí tranzistorů	- 83 -
5.5.2	Příklad vyjádření reálného čísla ve formátu IEEE 754	- 100 -
6.	ZÁVĚRY PRÁCE	- 113 -
6.1	SHRnutí A DISKUSE	- 113 -
6.2	REKAPITULACE VÝSLEDKŮ PRÁCE	- 115 -
6.3	APLIKACE VÝSLEDKŮ PRÁCE.....	- 117 -
6.4	MOŽNOSTI DALŠÍHO ROZVOJE TÉMATU	- 117 -
7.	SEZNAM POUŽITÉ LITERATURY	- 119 -
8.	PŘEHLED PUBLIKOVANÝCH PRACÍ UCHAZEČKY	- 123 -
8.1	PŘÍSPĚVKY VE SBORNÍKU	- 123 -
8.2	PUBLIKACE V RECENZOVANÉM ČASOPISE	- 125 -
8.3	KAPITOLA V MONOGRAFII.....	- 125 -
8.4	SKRIPTA	- 125 -
9.	SEZNAM POUŽITÝCH OBRÁZKŮ A TABULEK	- 126 -
10.	REJSTRÍK POUŽITÝCH POJMŮ A ZKRATEK.....	- 126 -

1. Úvod

Současná moderní doba se vyznačuje dynamickým rozvojem informačních a komunikačních technologií, což se odráží téměř ve všech oblastech lidského života. Toto se promítá také do situace ve výuce a vzdělávání. Školy i ostatní vzdělávací instituce investují stále větší částky do pořizování výpočetní techniky a programového vybavení, které je využíváno jako prostředek či podpora výuky. Také stále více firem specializovaných na vývoj softwaru obrací svou pozornost k produktům určených prvotně pro výuku a testování studentů. Proto je tedy nyní již možné si výukový software nechat vyrobit přímo „na míru“ podle specifických požadavků, což může určitým způsobem přispět ke zkvalitnění procesu výuky.

I přes výše jmenované možnosti ale nelze výuku prostřednictvím elektronického systému užít ve všech tématických oblastech a prostředích výuky. Existují i oblasti (např. formování velmi náročných praktických dovedností a návyků), ve kterých není vhodné využít elektronickou výuku, tj. kde výukovým programem nelze nahradit fyzický kontakt studenta s učitelem. Využití e-learningu má větší opodstatnění při distanční formě výuky, než-li při prezenční výuce, kde se využívá pouze jako pomůcka či doplněk.

Je zřejmé, že je velmi náročné vyvinout kvalitní a v praxi použitelný elektronický výukový systém, a to jak z hlediska technického provedení, tak i s ohledem na hledisko pedagogické. Je nutné, aby systém zahrnoval obě tyto roviny. K tomu, aby daný výukový systém byl kvalitní a přínosný, a to jak pro studenta, tak i pro pedagoga, musí také kromě základních funkcí (tj. výuky, testování, komunikace, konzultace, atd.) obsahovat účinné zpětnovazební prvky, pomocí kterých je možné studenta hodnotit a určitým způsobem korigovat jeho průchod výukovým systémem. Je proto nezbytné analyzovat, jak vhodným způsobem konstruovat a aplikovat do elektronického výukového systému funkční a kvalitní zpětnovazební prvky. Z výše uvedených důvodů je tato disertační práce věnována tomu, jakým způsobem vhodně formalizovat zpětnou vazbu a vytěžit z ní požadované informace.

Tato předkládaná disertační práce je tématicky rozčleněna na tři hlavní části – kapitoly s názvem „E-learning a jeho význam“, „Modely výpočtu a procesu“ a dále „Návrh zobecněné Petriho sítě“.

První kapitola práce „**E-learning a jeho význam**“ obsahuje teoretickou část i část pokrývající vlastní práci. Teoretická část se zabývá problematikou vymezení a pojetí pojmu e-learning, výhodami a úskalími jeho využití pro výuku a v neposlední řadě problematice zpětné vazby v prostředí e-learningu. Kromě teoretického základu, týkajícího se elektronické výuky daná kapitola obsahuje podkapitulu s názvem „**Problematika počítačového testování**“, která obsahuje nové náměty, jakým způsobem vhodněji hodnotit odpovědi v testech s otázkami s možností výběru více alternativ.

Druhá teoretická kapitola práce „**Modely výpočtu a procesu**“ je věnována možnostem modelování elektronického výukového systému pomocí vybraných modelů výpočtu, konkrétně konečného a zásobníkového automatu, Turignova stroje a Petriho sítí a shrnutí jejich výhod, nevýhod a vhodnosti použití pro modelování e-learningového systému. Podkapitola práce „**Možnosti modelování výukového systému**“ si klade za cíl shrnout známé případy modelování průchodu studenta elektronickým výukovým kurzem za pomoci Petriho sítí i dalších formálních matematických modelů. V kapitole jsou obsaženy vybrané modely e-learningových systémů českých i zahraničních autorů.

Žádný z výše uvedených formálních modelů výpočtu a procesu není plně vhodný pro modelování elektronického výukového systému a navigaci studenta ve výukovém procesu. Proto autorka ve třetí praktické části disertační práce s názvem „**Návrh zobecněné Petriho sítě**“ navrhuje nový model – tzv. zobecněná Petriho síť, která se jeví jako vhodnější pro navigaci studenta v elektronickém výukovém systému než výše uvedené nástroje.

2. Cíle práce, motivace a použitá metodika

2.1 Cíle práce

Předmětem výzkumu předkládané disertační práce je vystižení společných zásad, kterými se má řídit kvalitní elektronický výukový systém, tedy konkrétně jde o:

1. **Navigaci studenta**, tj. v jakém pořadí má student studovat jednotlivé výukové bloky. To vše na základě výsledků testů.
2. **Získávání zpětné vazby** pro lektora elektronického kurzu s cílem:
 - kontroly výuky a studijních výkonů studenta,
 - jeho vstupu do kurzu, a to buď přímo „on-line“ nebo pozváním si studenta na konzultaci.

Student by měl mít při průchodu výukovým systémem poskytnutou určitou omezenou úroveň „soukromí“. Například ne o všech jeho dílčích neúspěších by se měl lektor ihned dovědět. Jednou z možností je například tolerovat neznalost studenta v kontrolní otázce na konci tématického celku, poskytnout studentovi možnost ověřit si své znalosti látky ve zkušebním testu, či umožnit studentovi více možností pro splnění daného testu. Při přílišné kontrole ze strany lektora by student mohl být demotivován a je možné, že student ve snaze v kurzu uspět, nechá testy za sebe vyplňovat jinou osobou.

Cíle disertační práce jsou tedy tyto následující:

- **Vytvoření obecného modelu datové struktury**, popisující stavový prostor používání elektronického výukového kurzu, tak aby umožňoval navigaci studenta, generování rad a doporučení studentovi a získávání zpětné vazby pro lektora i studenta.
- **Navržení omezení a konkretizace modelu** při navigaci studenta výukovým systémem, zabezpečující základní požadavky, tj. navigaci studenta, generování rad a doporučení studentovi a získávání zpětné vazby.

Vzniká otázka, který z uvedených možných modelů procesu (konečné automaty, zásobníkové automaty, Turingovy stroje a Petriho sítě) bude shledán nejvhodnější pro daný úkol. Předběžně se zdá, že Petriho sítě poskytují nejširší repertoár možností, nejméně omezení a jsou pro znázornění výukových tahů poměrně názorné. Jejich výhodou je i skutečnost, že existují pokusy pro modelování e-learningu tento nástroj použít, například dle [6], [7], [25] až [30].

I když jsou Petriho sítě z daných modelů procesů relativně nejvhodnější k modelování průchodu studenta výukovým kurzem, mají jistá omezení. Petriho sítě je vhodné zobecnit tak, aby byly schopny zachytit více procesů, které mají vzájemnou vazbu. Z těchto důvodů je v disertační práci navržena modifikace Petriho sítě, tzv. „zobecněná“ Petriho síť, která bude lépe modelovat studentův postup skrz elektronický výukový kurz.

Lze stanovit některé zásady, které bude vyvíjený model zobecněné Petriho sítě muset respektovat:

- model bude reprezentován orientovaným grafem,
- model bude obsahovat jeden výchozí (počáteční) uzel, který reprezentuje vstup studenta do výukového systému,
- model bude obsahovat jeden nebo více koncových uzlů, reprezentujících výsledek studenta ve výukovém kurzu (např. prospěl, neprospěl), případně rozeznávat několik možností úspěšného průchodu uzlem v závislosti na kvalitě podle nějaké stupnice ordinálního typu,
- uzly, které se nacházejí mezi uzlem výchozím a koncovým(i), reprezentují jednotlivé výukové či testovací lekce, kterými student pro splnění výukového cíle musí projít.

Výuka v elektronickém výukovém systému bude reprezentována sledem průchodů jednotlivými uzly modelu. Úspěšné absolvování kurzu končí v koncovém uzlu (koncových uzlech) modelu.

V neposlední řadě je cílem předkládané disertační práce **úvaha o možnostech hodnocení odpovědí studentů v testech**. Součástí této úvahy bude návrh, zda připouštět a jakým způsobem hodnotit u testů s výběrem odpovědi studentovo přiznání, že správnou odpověď na danou otázku nezná. Úvaha spočívá v tom, zda připouštět na různé typy testových otázek odpověď „nevím“ a jakým způsobem tuto odpověď hodnotit. Snahou existence takovéto alternativy odpovědi by mělo být upřednostnit studenty, kteří jsou kritičtí ke svým znalostem a odpověď „nevím“ zvolí a nevědomost přiznají, oproti studentům, kteří typují odpověď náhodným způsobem a této možnosti odpovědi nevyužijí.

2.2 Motivace práce

V současné moderní době je stále více využívána distanční forma výuky, fungující za pomoci informačních a komunikačních technologií. Proto je v procesu vzdělávání využívána celá řada elektronických výukových systémů. Ne všechny tyto systémy umožňují vhodnou navigaci studenta ve výukovém procesu a kontrolu úspěšnosti jeho studia.

Z výše uvedených důvodů je motivací autorky disertační práce snaha o zlepšení kvality navigace studenta skrz elektronický výukový systém. Proto bude součástí práce vytvoření vlastního modelu průchodu studenta výukovým kurzem pomocí „zobecněné“ Petriho sítě a ověření vhodnosti daného modelu.

2.3 Použitá metodika

Metodika této disertační práce je založena na studiu odborné literatury a dále, vzhledem k tomu, že je autorka jedním ze správců elektronického výukového systému Moodle na České zemědělské univerzitě, také z vlastních zkušeností a znalostí z oblasti e-learningu.

Práce autorky se skládala z níže uvedených kroků:

- analýza problému,
- studium odborné literatury,

- analýza současného stavu problematiky modelování výukového procesu,
- zobecnění načerpaných expertních poznatků,
- zobecnění vlastních zkušeností a znalostí,
- užití formálních modelů sledujících průběh výukového procesu,
- vytvoření vlastního modelu průchodu studenta výukovým kurzem pomocí „zobecněné“ Petriho sítě a ověření vhodnosti daného modelu.

2.4 Terminologie

Odborné termíny, které jsou v předkládané disertační práci užívány, jsou vysvětleny v kapitole 10 s názvem „Rejstřík použitých pojmů a zkratk“. Pokud v této kapitole některý termín obsažen není, či mohou vzniknout pochybnosti o tom, v jakém smyslu je termín využíván, je tento použit ve smyslu uvedeném v [32].

3. E-learning a jeho význam

3.1 E-learning

3.1.1 Vznik a vývoj e-learningu

První pokusy o využití počítačů ve výuce byly uskutečněny v šedesátých letech na sálových počítačích a vzdělávací software byl založen na teorii programovaného učení. Základní myšlenkou této metody bylo úplné řízení práce studenta. Celý postup byl rozdělen do dílčích kroků. Každý z těchto kroků obsahoval výklad určité látky, kontrolní otázku a reakci ze strany stroje, která informovala studenta o správnosti jeho odpovědi. Podle této správnosti se pak vybral další krok. Program mohl být i větvený, takže každý student jím mohl procházet jiným způsobem podle svých znalostí. Studenti sice mohli postupovat vlastním tempem podle svých znalostí, ale jediné co se od nich vyžadovalo, bylo číst text a vybírat jednu z možných variant odpovědí. Studenti často ani text nečetli a snažili se odpovídat na otázky náhodnou volbou.

Koncem sedmdesátých let se začalo od programovaného učení upouštět. Snahou vývojářů začalo být dát studentovi větší volnost při práci a tím zvýšit jeho motivaci k učení. Základním principem však stále zůstávalo řízení celého procesu programem. Bylo proto potřeba, aby se tento program choval co nejinteligentněji, tj. aby byl schopen reagovat různě na odlišné situace a reakce ze strany studenta. Proto začaly vznikat celé výukové systémy založené na umělé inteligenci.

Dle [5] se vzdělávání na osobních počítačích začalo rozvíjet v období 1984 až 1993, přičemž elektronická výuka v dnešním slova smyslu se začala prudce rozšiřovat až s rozvojem Internetu po roce 1993. Samotný pojem „E-learning“ byl zaveden až po roce 1999. Před tím se používaly pojmy WBL (Web-based training) nebo online learning (výuka prostřednictvím webu).

3.1.2 Vymezení pojmu e-learning

Dle [1] výraz e-learning vznikl propojením slovního základu „learning“ (učení) a předpony „e“ (elektronické), tedy e-learning můžeme chápat jako elektronické učení

či výuku, přesněji jako elektronickou podporu výuky. Je zřejmé, že e-learning vznikl propojením vzdělávání s moderními informačními a komunikačními technikami.

V širším slova smyslu lze tedy e-learning chápat jako technologii výuky, která zahrnuje tvorbu, distribuci, řízení výuky a zpětnou vazbu ve výukových kurzech podporovaných elektronickými prostředky.

V užším slova smyslu si lze e-learning představit jako proces výuky vedený prostřednictvím výpočetní techniky (počítač), dalších moderních komunikačních technologií (intranet, Internet) a speciálního software (multimediálních programů).

Základní součástí elektronické výuky je studijní kurz, který zahrnuje učební text, doplněný o zvuk, obrázky, animace, audio/video sekvence, simulace (tzv. hypertext), dále cvičení, testy, diskusní skupiny, on-line semináře a konzultace v synchronním režimu, odkazy na další internetové zdroje a zpětnovazební prvky.

3.1.3 Pojetí elektronické výuky

Nejširším pojmem z oblasti elektronické výuky je e-learning, neboli Technology-Based Learning, který zahrnuje širokou škálu aplikací a procesů, včetně CBL (Computer-based learning), WBL (Web-based learning), synchronní formy on-line výuky (virtuální třídy) a spolupráci a komunikaci po síti.

Dle [2] On-line Learning (Web-based Learning, Internet-based Learning) tvoří jen část e-learningu a popisuje studium prostřednictvím Internetu nebo intranetu, téměř výhradně na základě web technologie.

Užším pojmem je Computer-based Learning, jež se vztahuje ke kurzům distribuovaným např. prostřednictvím CD-ROM a prezentovaným na počítači, který není připojen k síti. Učební materiály kurzu nejsou vybaveny odkazy na studijní zdroje mimo kurz.

Specifičtější formou e-learningu je Corporate e-learning, který je často využíván pro podnikové vzdělávání. Tato forma e-learningu popisuje podnikové vzdělávání (trénink) poskytovaný prostřednictvím e-learningu. Podnikové vzdělávání bývá více zaměřeno na osvojení si znalostí a dovedností požadovaných v daném podniku.

On-line forma studia je realizována synchronní nebo asynchronní formou. Synchronní forma představuje učitelem vedenou výuku, kdy jsou všichni účastníci výuky připojeni k počítačové síti a v reálném čase mezi sebou komunikují. Jsou využívány prostředky pro video konference, online rozhovory a sdílení aplikací, společně označované jako virtual classroom (virtuální třída).

Asynchronní forma představuje studium, ve kterém nejsou účastníci připojeni k síti v reálném čase. Studenti si sami volí v určitém rozsahu vlastní tempo studia. Pracují s učebními texty, prezentacemi, plní zadané úkoly, komunikují prostřednictvím elektronické pošty a diskusních skupin.

3.1.4 Výhody a omezení elektronické výuky

Dle [2] je jednou ze základních myšlenek e-learningu jeho integrace do procesu vzdělávání, která přinese zvýšení kvality studia a snazší dostupnost studijních materiálů za přijatelnou cenu. Zejména při využívání WBT není nutná fyzická přítomnost studentů a lektora ve společné učebně, materiály jsou distribuovány pomocí sítě Internet, což také přináší časovou úsporu a flexibilitu a úsporu nákladů. Každý student má možnost individuálního přístupu ke studiu, výukové materiály může studovat dle svých potřeb a například se navracet k obtížnějším pasážím, či nestudovat jemu dobře známou problematiku, což také zefektivňuje proces vzdělávání. Z hlediska lektora je výhodou elektronických kurzů zpětná vazba od účastníků kurzů a možnost snadné aktualizace kurzů o nové informace a poznatky, což zabrání studentům, aby studovali již neplatné materiály.

Dle [2] nejsou s využívání e-learningu však spojena pouze pozitiva, ale také jistá úskalí. Při zavádění e-learningu jde například o zvýšené počáteční náklady do potřebných technologií, dále nízká propustnost tuzemských sítí a vysoká cena za rychlý

přenos. Je třeba také zohlednit psychologické aspekty. Lidskou intuici lze jen částečně naprogramovat a dobrý lektor ji obvykle má a nemůže ji při zpětné vazbě v dostatečné míře uplatnit. Dalšími aspekty jsou chybějící počítačová gramotnost, psychologická bariéra a odpor k využívání IT. Asi největší slabinou e-learningu je absence přímého kontaktu studenta s lektorem a minimální rozvoj verbální komunikace studentů.

3.2 Zpětná vazba v e-learningu

Jedním z hledisek hodnocení kvality elektronického výukového kurzu je i kvalita obsažených zpětnovazebních prvků. Zpětná vazba je soubor kontrolních prvků v elektronickém kurzu, pomocí kterých lektor získává informace o tom, jak si který student vede v konkrétním úseku kurzu, jak dlouho se jednotlivými částmi zabývá, která vyučovaná látka je pro něj obtížná či které úseky vyučované problematiky student ovládá dobře.

Dle [4] zvláště při distančním typu výuky, tj. samostudiu, musí učební materiály obsahovat ve zvýšené míře zpětnovazebné prvky. Ty slouží nejen k aktivizaci a motivaci studenta, ale rovněž k hodnocení jeho práce. Zpětná vazba může být cenná pro studenta i v případě, že do ní nevstupuje lektor. Dosažené výsledky jsou využitelné také samotnými studenty jako zpětná vazba jejich studia. Student, který má také přístup ke zpětnovazebným prvkům si může pomocí jejich výstupů (výsledky testů, kontrolní otázky, atd.) ověřovat, jak úspěšně nastudoval danou učební látku, do kterého místa kurzu se má vrátit, případně jak studium dále modifikovat.

Z kvalitní zpětné vazby je dále možné získat přehled o vstupních znalostech studenta, statistiku výsledků průběžných testů a výstupy ze závěrečných testů, které vyhodnocují, zda student vyučovanou problematiku:

- **nastudoval** – tzn. prokázal znalost textu, který měl za úkol prostudovat,
- **porozuměl jejímu obsahu** – tzn. dokázal z učebního textu odvodit souvislosti mezi nastudovanými informacemi, jejich význam a využití,
- **a dokázal aplikovat na konkrétní příklady** – tj. pomocí znalostí nabytých z teoretického výukového textu dokázal řešit konkrétní úlohy z praxe.

3.2.1 Možnosti zpětné vazby

Prvky zpětné vazby, obsažené v elektronickém výukovém kurzu do jisté míry musí nahradit roli učitele, který v prezenční výuce může studenty sledovat a hodnotit studenty pomocí ústního zkoušení, otázek či písemných testů.

Dle [4] jsou zpětnovazební prvky v elektronické výuce většinou řešeny pomocí aktivizačních úkolů k zamyšlení, drobných úkolů, praktických příkladů, kontrolních otázek a úkolů, didaktických testů a hodnocených testů.

Kontrolní otázky lze obdobným způsobem jako dle [19] rozdělit podle úrovně znalostí a dovedností, které jsou od studenta vyžadovány:

- **znalosti** (*knowledge*) – ověříme otázkami, které vyžadují od studenta pouze vybavit si z paměti detaily z učebnice či jiného informačního zdroje,
- **porozumění** (*understanding*) – ověříme otázkami, které vyžadují provedení nové představy a její zabudování do již existujícího konceptu (např. systému),
- **vyhodnocení** (*evaluation*) – ověříme otázkami, které vyžadují využití kritického myšlení a vyhodnocení jejího dopadu.

Kontrolní otázky a úkoly by měly být krátké a jednoduché, a měly by představovat součást lekce nového učiva. Jejich účelem je aktivizovat studenta, aby se lépe soustředil na probírané učivo. Dále by měly napomáhat lepšímu pochopení látky pomocí praktických příkladů a zároveň studentovi slouží jako zpětná vazba, zda pochopil dobře probíranou problematiku.

Nejjednodušší úroveň představují kontrolní úkoly, které se týkají přímo textu výukové lekce a v textu rovněž student nalezne správnou odpověď. Náročnější formou jsou úkoly, které po studentovi vyžadují vytvoření vlastního řešení či odpovědi. Zde je nezbytné studentovi poskytnout správnou odpověď či řešení, což je snadné. Na druhou stranu je také třeba, aby výukový software studentovu odpověď analyzoval a opravil, což je naopak velmi náročné. Problém proto nastává, pokud je třeba „očekávanou“ odpověď porovnat se slučitelnou odpovědí. Obě tyto odpovědi mohou tématicky

vyjadřovat totéž, ale přitom se syntakticky a sémanticky velmi lišit. V opačném případě se odpovědi mohou lišit například pouze v jednom rozdílném, vynechaném či nadbytečném slově (například „ne“, „nebo“, „a“, „alespoň“, apod.) a mohou vyjadřovat každá něco podstatně jiného. Problémem je, jak algoritmicky zjistit „vzdálenost“ mezi odpovědí studentovou a „očekávanou“ odpovědí. Čím je předmět, jehož znalosti jsou pomocí výukového programu ověřovány, intelektuálně náročnější, a tudíž zde velmi záleží na funkčních vazbách a logických souvislostech, tím je takovéto „měření vzdálenosti“ složitější a v mnoha případech i nemožné. Naopak v případech, kde jde například o paměťové výpočty, je toto „měření“ poměrně snadné.

Dle [4] jsou dalším zpětnovazebním prvkem didaktické testy. Jejich prvotním cílem je sdělit studentům, ale i pedagogovi, zda studenti pochopili probírané učivo. Testy mohou být vytvářené jako otevřené nebo uzavřené:

- **úlohy s výběrem odpovědi** (jedna správná odpověď, jediná nejpřesnější odpověď, více správných odpovědí a více nesprávných nabízených odpovědí),
- **úlohy dichotomické** (odpověď ano/ne),
- **přiřazovací úlohy**,
- **uspořádací úlohy**,
- **otevřené úlohy** (je požadována jednoduchá nebo rozsáhlá odpověď).

U všech těchto typů odpovědí, kromě otevřených, je nutné zajistit, aby odpovědi byly vyhodnoceny automaticky, bez zásahu lektora. Ve většině případů (tj. u úloh s výběrem odpovědi, úloh dichotomických, přiřazovacích a uspořádacích úloh) je vyhodnocení snadné. Problém ale může nastat u otevřených úloh, kdy v limitním případě lze tento problém srovnat s již zmiňovaným problémem vyhodnocení odpovědí (tj. srovnání odpovědi studenta a odpovědi „očekávanou“) u úkolů, které po studentovi vyžadují vytvoření vlastního řešení či odpovědi. Jak již bylo řečeno, vzhledem k odlišnostem ve vyjadřování každého jedince a množství slov a výrazů v našem jazyce musí být otázky formulovány co nejpřesněji. Například je možné po studentovi požadovat doslovné uvedení definice. To je ale často nemožné a také nežádoucí, neboť

to může vést k tomu, že je vlastně vyžadováno pouhé doslovné „odřikávání“ definice. Je otázkou, zda je v tomto případě při hodnocení užitečné využít softwarové hodnocení.

Dle [3], [4] se nejčastěji využívají úlohy typu úlohy „výběr z několika možností“ (multiple choice). Tyto úlohy je vhodné zařazovat jako vstupní testy každé lekce i jako testy, které mají na konci lekce ověřit studentovi znalosti. U náročnějších úloh je třeba doplnit odpovědi o rozsáhlejší výklad problematiky.

U úloh typu multiple choice a uspořádacích úloh je z hlediska zpětné vazby vhodné věnovat se problému, jakým způsobem, hodnotit „částečně správné a částečně špatné“ odpovědi, v případě, že je více než jedna odpověď z dané nabídky správná.

Mějme například tuto situaci: Skupině studentů je předložena otázka s výběrem odpovědi, přičemž dvě odpovědi (b, c) z pěti v nabídce jsou správné, tři (a, d, e) odpovědi špatné. Student počet správných odpovědí nezná. Jednotliví studenti odpoví například takto:

- **student 1:** odpověď jen b,
- **student 2:** odpověď a, b, c, d,
- **student 3:** odpověď b, c, e,
- **student 4:** odpověď c, d,
- **student 5:** řešení vynechá (neví).

Jakým způsobem tedy hodnotit tyto odpovědi? Je žádoucí lépe ohodnotit studenta, který zvolí více odpovědí, přičemž ne všechny jsou správné, nebo studenta, který zvolí například pouze jednu odpověď z více správných, protože si není jistý svými znalostmi?

Dalším problémem u multiple choice a uspořádacích úloh je, zda poskytnout studentovi možnost na otázku odpovědět „nevím“ a jak opět tuto odpověď ohodnotit. Zda odpověď ohodnotit lépe, než odpověď špatnou a brát v potaz určité studentovo „přiznání se“ a uvědomění si vlastní neznalosti, či obě odpovědi postavit na stejnou úroveň. V případě, že je žádoucí vychovat kriticky myslícího a odpovědného člověka, je

nejspíše vhodnější přiřadit jeho odpovědi „nevím“ lepší hodnocení, nežli špatné odpovědi, a tedy snaze studenta odpovědět za každou cenu, nehledě na to, že špatně. Jeden z užívaných a patrně vhodných algoritmů pro test s volbou mezi N možnostmi z nichž právě jedna je správná je následující:

- za otázku správně zodpovězenou přičíst jeden bod,
- za otázku bez odpovědi nebo s více označenými odpověďmi než jedinou nepřipočítávat nic,
- za otázku s nesprávnou odpovědí odečítat $1/(N - 1)$ bodů.

Toto hodnocení je neutrální v tom smyslu, že odpovídá pravděpodobnosti „uhádnutí“ správné odpovědi při náhodné volbě jedné z možností. Nepreferuje tedy přiznání se k nevědomosti před náhodnou volbou. Pokud bychom chtěli přiznání neznalosti před „hádáním“ zvýhodnit, musela by být srážka za nesprávnou odpověď vyšší. Této otázce se bude autorka disertační práce věnovat blíže v kapitole 3.3 Problematika počítačového testování.

Dalším zpětnovazebním prvkem, pomocí něhož lze hodnotit práci studentů, jsou testy. Jejich výsledky ukazují lektorovi, zda student dobře nastudoval vyučovanou látku. Tyto testy mohou mít formu vzdálenou, tj. student zpracovává test vzdáleně, u svého počítače a po odeslání jsou výsledky uloženy do databáze. Lektor poté výsledky shlédne a popřípadě přiřadí komentář. Nevýhodou této podoby testu je malá možnost kontroly, zda student pracoval samostatně. Objektivnější formou jsou takové testy, které musí student vyřešit na daném místě a v daném čase, nevýhodou však je, že se student musí dostavit do vzdělávací instituce. Přesto jsou tyto testy vhodným typem zjišťování znalostí například na konci celého kurzu.

Díky rozvoji informačních a komunikačních technologií mohou učební materiály obsahovat zpětnovazebné prvky nejen ve statické podobě. Je možno plně rozvinout jejich dynamičnost, interaktivnost a zároveň do jisté míry nezávislost na práci tutora kurzu. Dle [3] a [4] nemusí student čekat na odpověď a komentáře tutora, test může být vyhodnocen okamžitě po jeho dokončení. Co se týče nástrojů pro tvorbu testů, ty by měly zajistit náhodnou generaci otázek i nabídnutých odpovědí. Výhodou je, že

například dva shodné testy nemusí mít stejný vzhled, testové otázky i odpovědi je možné zobrazit v odlišném pořadí, či v daném okamžiku zobrazit pouze jednu z otázek testu. Po jejím zodpovězení je opět náhodně zobrazena otázka další. Student je pak více nucen zamýšlet se nad otázkou i odpovědí a neřešit úlohu pouze na základě zapamatování si vizuální podoby testu.

3.3 Problematika počítačového testování

Jak již bylo zmíněno v předchozí kapitole, velmi důležitou součástí elektronických výukových systémů jsou prvky zpětné vazby. Tento soubor kontrolních prvků slouží nejen jako podklad pro pedagoga k hodnocení práce studenta, ale v neposlední řadě slouží také jako motivační prvek studenta, tj. jako informace, zda student látku úspěšně nastudoval a ovládnul.

Vzhledem k tomu, že by účelem zabudovaných zpětnovazebních prvků mělo být studenta spíše motivovat, nežli jej od studia odradit, je potřeba nastavit vhodně váhy odpovědí, tak aby bylo možné rozlišit poctivé studenty, kteří v testu přiznají neznalost správné odpovědi, od studentů, kteří zvolí odpověď dle náhodného výběru.

Z tohoto důvodu se v této podkapitole autorka pokusí navrhnout hodnocení odpovědí studentů v elektronických testech a odlišit váhu odpovědi dle náhodného výběru od odpovědi studenta, který přiznává neznalost testované látky a vybere odpověď typu „nevím“.

Při elektronickém testování je třeba zásadně odlišit výše zmíněné cíle, které může takovéto testování mít snahu plnit:

- **Prvním cílem** je poskytnout studentovi možnost si sám ověřit své znalosti, dovednosti a porozumění látce a na základě výsledku mu doporučit další navigaci ve výukovém systému. Tento cíl lze plnit bez účasti učitele a ten ani nemusí být informován o výsledku testu. Není zde třeba řešit otázku, zda na testovací otázku student odpověděl samostatně, ani nehrozí, že neúplně či nejasně formulovaná otázka studenta poškodí. Využití výsledku testu je plně v rukou a odpovědnosti studenta.

- **Druhým cílem** programových testů může být získání informací pro pedagoga o znalostech a dovednostech studenta a jeho píli, případně podkladů pro klasifikaci studenta, získávání příslušné kvalifikace či rozhodnutí o jeho přijetí na školu. Tedy jde o náhradu klasické zkoušky. Výsledek takového testu pak může mít pro studenta velký význam a může i ovlivnit jeho životní dráhu. Proto je třeba validitu výsledků takového testu posuzovat velmi obezřetně. Především je třeba zabezpečit autenticitu objektu testu (zda ve skutečnosti odpovídá příslušná osoba) i zda odpovídá samostatně, bez nápovědy a nepovolených pomůcek. Při distančním typu studia to může být problém. Dále je třeba zabezpečit jednoznačnost odpovědi, vyloučit nedorozumění a odpovědi spravedlivě hodnotit a výsledky spravedlivě agregovat. O výše uvedených problémech se podrobněji zmíníme v následujících odstavcích.

Nehledě na tyto problémy je programové testování velmi oblíbené a to nejen pro úsporu času pedagoga. Poskytuje totiž (často jen zdánlivě) záruku objektivnosti a vyloučí zmanipulování výsledků. Je typické, že se využívá nejčastěji v přijímacím řízení, kdy uchazeč ještě není studentem, neexistuje tedy vztah učitel – žák a výsledek zkoušky může podléhat přezkoumání podle správního řádu. Počítačový test tak může být užit pro prokázání nestrannosti.

Jednoznačnost odpovědi může být problémem z mnoha důvodů. Typicky problematickou otázkou jsou otázky typu:

„Co je nejvhodnější pro ...?“

„Co je nejvíce podobné (něčemu)?“

„Co je k A ve stejném vztahu jako B k C?“

Zde může být odpověď závislá na oprávněném subjektivním pohledu, který nemusí být totožný u toho, kdo test sestavil a u studenta. Vhodnost je relativní. Každá akce mívá klady i zápory. Co z toho je důležitější? Mezi objekty může být řada různých vztahů. Který je ten, podle kterého máme hledat odpověď?

Nebezpečí může být i v zdánlivě bezproblémových otázkách typu ano/ne a to i v exaktních vědách. Často můžeme zapomenout na nějaký „samozřejmý“ předpoklad. Je platná například tato tzv. základní věta algebry: „Každý polynom má nad tělesem komplexních čísel alespoň jeden kořen“? Není. Chybí předpoklad, že stupeň polynomu je alespoň jedna. Nenulové konstanty jsou polynomy nultého stupně a rovnice $a = 0$ při $a \neq 0$ nemá řešení pro žádné x .

Další zdroj problémů může spočívat v nedorozumění stran modelu, který je užíván. Naprostá většina výuky se totiž netýká poznatků přímo o empirickém světě, ale o tvrzení platných v nějakém modelu tohoto světa, který realitě odpovídá „docela dobře“ z nějakého pohledu. Typický případ nedorozumění v modelu může být jakýkoliv příklad na skládání rychlostí. Střela vystřelená z letadla, či raketa z rakety. Výsledek samozřejmě závisí na tom, zda použijeme Newtonův model, což je asi předpokládáno, nebo model relativistický.

Nedorozumění tohoto druhu lze ilustrovat na následujícím příkladu: „Máme dvě dvoulitrové nádoby. V první je přesně jeden litr destilované vody, v druhé přesně jeden litr čistého ethylalkoholu. Z druhé nádoby odebereme půl litru lihu, vlijeme do první nádoby a dokonale promícháme. Poté odlijeme z první nádoby půl litru směsi a přelijeme zpět do druhé nádoby.“

Zvolte jednu z následujících možností:

- a) zbyde více lihu v první nádobě než vody ve druhé,
- b) zbyde více vody v druhé nádobě než lihu v první,
- c) zbyde stejné množství lihu ve vodě jako vody v lihu,
- d) není dost informací k rozhodnutí mezi odpověďmi (a), (b), a (c).

Autor zřejmě předpokládá, že správná je odpověď (c). Pilný ale nepřiliš důvtipný student k ní dojde výpočtem. Inteligentní student úvahou, že jde o uzavřený systém, ve kterém je jeden litr vody a jeden litr lihu, nic se neztratilo, ani nedoplňovalo, takže mají-li obě nádoby v závěru objem 1 litr, musí platit odpověď (c).

Kdo však má znalosti z fyziky či chemie, bude mít problém s tím, že v pozemských podmínkách je existence 100% C_2H_5OH vyloučená a především s tím, že slijeme-li 0,5 litru lihu a 1 litr vody, nevznikne 1,5 litru roztoku.

Matematicky založený student zvolí jistě odpověď (d), protože nebude mít dostatek informací o tom, zda je počet molekul vody obsažených v jednom litru dělitelný třemi a počet molekul lihu v jednom litru dělitelný šesti.

3.3.1 Návrh hodnocení odpovědí studentů při počítačovém testování

K problému hodnocení výsledků programových testů lze říci toto. Pokud test slouží pouze studentovi pro jeho informaci o tom, jak si látku osvojil, postačí jej upozornit na všechny jeho chybné odpovědi s případným uvedením správné, nejlépe se stručným zdůvodněním. Ještě důležitější než správná odpověď je však navigace studenta na tu komponentu výukového kurzu, kde je vysvětlena látka na kterou odpověděl nesprávně. Bodování výsledků a jejich agregace je nevýznamná a může i odpadnout. Pokud však výsledek testu slouží pro informaci učitele, případně jako hodnocení studenta a podklad pro přiznání kvalifikace, je bodování odpovědí a agregace získaných bodů podstatné.

Zde vzniká poměrně zásadní otázka, zda připouštět odpověď „NEVÍM“ a jak ji hodnotit. Autorka práce a její pedagogičtí spolupracovníci se na základě svých zkušeností výrazně kloní k názoru, že taková možnost by měla být připuštěna. Důvodem je jednak snaha podpořit kritický přístup studentů ke svým znalostem a poctivost jejich uvažování, za druhé pak skutečnost, že přiznání nekompetence v řadě případů vede k nesrovnatelně nižším škodám než neuvážené chybné rozhodnutí. Typický případ je chování lékaře, který v případě své nejistoty o diagnóze pošle pacienta raději ke specialistovi, než aby riskoval, že jej v důsledku chybné diagnózy poškodí.

V každém případě však není možné hodnotit odpověď „NEVÍM“ stejně jako odpověď nesprávnou. To by nezaslouženě zvýhodňovalo studenty, kteří odpověď typují náhodně, před těmi kritickými ke svým znalostem. Například u binárních otázek

s odpověďmi ANO/NE je při náhodné volbě 50% pravděpodobnost „uhádnutí“ správné odpovědi. Bude-li správná odpověď hodnocena jedním bodem a odpověď „NEVÍM“ nula body, musí být za chybnou odpověď udělován alespoň jeden minusový bod. I to ale vede ke stejnému hodnocení studenta, který nevědomost přizná jako studenta, který slepě hádá. Spravedlivější a výchovnější se tedy jeví trestat chybu odečtením více než jednoho bodu. Například přidělením $-(1 + \alpha)$ bodů, kde $\alpha > 0$. Zde autorka a její spolupracovníci testovali volby $\alpha \in (0, 1)$ s výsledkem, že hodnoty v okolí $\alpha = 0.5$ se jeví jako poměrně vhodné.

V následujících doporučeních ponecháme konstantu α jako volnou pro výběr z intervalu $(0, 1)$ s doporučením volby $\alpha = 0.5$. Návrh hodnocení vysvětlíme podle typu otázek:

1. Otázky na volbu ze dvou vzájemně se vylučujících možností, z nichž jedna jistě nastane. Typ odpovědi ANO/NE/NEVÍM. Hodnocení tímto způsobem:
 - správná odpověď: 1 bod,
 - chybná odpověď: $-(1 + \alpha)$ bodů,
 - odpověď NEVÍM: 0 bodů.
2. Otázky na výběr z několika $N > 2$ odpovědí, které se nevylučují ani není jisté, zda některá nastane. Student má zvolit libovolný počet možností, které se mu nabízejí, případně i žádnou z nich, či zvolit možnost NEVÍM u kterékoliv možnosti. Tento typ otázek se redukuje na typ ANO/NE/NEVÍM tak, že se otázka s N volbami převede na N samostatných binárních otázek, které hodnotíme podle bodu 1. Pokud chceme, aby odpovědi měly stejnou váhu nezávisle na počtu nabídnutých odpovědí,

přidělíme $\frac{\sum_{j=1}^N h_j}{N}$ bodů, kde

- $h_j = 1$ je-li jistá možnost zvolena správně,
- $h_j = -(1 + \alpha)$ je-li jistá možnost zvolena chybně,
- $h_j = 0$ je-li zvolena odpověď NEVÍM.

Je třeba upozornit na to, že u takovýchto otázek nesmí být na prvý pohled jasné, že některé dvě možnosti nemohou nastat současně, ani že nabízené možnosti pokrývají všechny možnosti, které mohou nastat.

3. Otázky na výběr jedné a pouze jedné z několika možností, z nichž jedna nastane. Zadavatel v tomto případě přijímá na sebe odpovědnost a deklaruje, že se možnosti vzájemně vylučují a některá je správná. V tom případě je při volbě „naslepo“ mezi $N > 2$ možnostmi pravděpodobnost správné volby $1/N$ a nesprávné volby $\frac{N-1}{N}$. Zachováme-li hodnocení správné volby jako 1 bod a odpovědi NEVÍM jako 0 bodů, vedly by náhodné odpovědi k matematické naději na výsledek 0 v případě odečítání x bodů pokud by bylo $\frac{1}{N} + x \cdot \frac{N-1}{N} = 0$. Odtud $x = \frac{1}{N-1}$. Pokud chceme náhodné odpovědi oproti přiznání nevědomosti trestat adekvátně jako v předchozích případech, hodnoty bodů jsou:
- výběr správné možnosti: 1 bod,
 - výběr chybné možnosti: $-\frac{(1+\alpha)}{N-1}$ bodů,
 - výběr odpovědi NEVÍM: 0 bodů.
4. Dalším typem otázek může být úkol seřadit prvky nějaké množiny o N prvcích do lineárního pořadí podle hodnoty nějakého jejich atributu (např. země podle počtu obyvatel, hor podle nadmořské výšky, historické události podle data, apod.).

Míru správnosti odpovědi pak lze nejlépe stanovit podle počtu inverzí oproti správnému pořadí prvků množiny. Je-li $<$ ostré lineární uspořádání, které považujeme za správné, je inverze v uspořádání $<$ určeného studentem taková dvojice prvků $\{a, b\}$, pro kterou je $a < b$ ale $b < a$. Každá nesprávně uspořádaná dvojice se tedy započítává jen jednou. Chceme-li zachovat princip bodování v souladu s případy 1., 2., a 3., je vhodné za zcela správnou odpověď, ve které není žádná inverze, přidělit 1 bod a za vzdání se odpovědi

(přiznání nevědomosti) udělit 0 bodů. Zcela nesprávná odpověď, při které student nabídl opačné pořadí, bude mít $\frac{N \cdot (N-1)}{2}$ inverzí. V souladu s principy uplatněnými v předchozích případech by měla být hodnocena $-(1 + \alpha)$ body, tedy o $2 + \alpha$ bodů níže než odpověď správná. Na jednu inverzi tedy připadá srážka $\frac{2 \cdot (2 + \alpha)}{N \cdot (N-1)}$ bodů a je proto přirozené za pořadí s I inverzemi přidělit $1 - I \cdot \frac{4 + 2\alpha}{N \cdot (N-1)}$ bodů. Takové hodnocení odpovídá transformaci úlohy stanovit pořadí na $\frac{N \cdot (N-1)}{2}$ binárních úloh vyslovit se k pořadí každé dvojice prvků z dané množiny.

Ostatní typy úloh uplatňovaných v programovaných testech lze obvykle převést na binární problémy analogickým postupem. Například problém párování na problémy, zda daná dvojice je přiřazena správně či nikoliv, problém třídění do zadaných kategorií na problém, zda byl prvek přiřazen správně či nikoliv, apod.

Úlohy, kde je vyžadována slovní odpověď nebo odpověď smíšená je strojově hodnotit velmi obtížné. Sémantický rozbor českého jazyka není jednoduchá záležitost a autorka a její pedagogičtí spolupracovníci jsou v tomto ohledu velmi skeptičtí.

Pro test, který obsahuje více otázek, je nutné stanovit vhodný algoritmus pro agregaci výsledků jednotlivých složek testu. Zde je vhodné každé otázce přiřadit váhu odpovídající jejímu významu pro požadovaný profil studenta a touto váhou násobit přidělené (kladné i záporné) body.

Pokud test pokrývá více témat nebo prověřuje různé znalosti a dovednosti studenta (např. některé otázky jsou zaměřeny na pochopení teorie, jiné na uplatnění teoreticky zdůvodněných postupů v praxi, apod.), je vhodné agregovat výsledky pro každé téma či každý pohled zvlášť. To může být důležité jak pro navigaci studenta ve výukovém systému, tak pro rozhodnutí, zda student splnil či nesplnil kvalifikační kritéria. Podmínka splnění kritéria pro „uznání“ zkoušky pak může například být

alespoň vyhovující znalost všech témat nebo alespoň vyhovující orientace v teorii i praxi současně. Dokonalé znalosti jednoho tématu pak nemohou nahradit úplné neznalosti jiného a dokonalá znalost teorie nestačí pro uznání zkoušky, nedovede-li ji student využít, právě tak jako nestačí schopnost realizovat dané postupy v praxi bez znalosti teoretického zázemí.

Při respektování těchto zásad se autorce a jejím spolupracovníkům osvědčuje v každé skupině vyžadovat pro uznání testu kritérium

$$V = \frac{\sum_{j=1}^N w_j b_j}{\sum_{j=1}^N w_j} > 0.5, \text{ kde:}$$

- N je počet otázek testu,
- w_j ($j = 1, \dots, N$) je váha příslušné otázky a
- $b_j \in \langle -1 - \alpha, 1 \rangle$ je počet dosažených bodů za odpověď na j -tou otázku.

Pokud má být výsledkem zkoušky známka v ordinální klasifikační stupnici, například na České zemědělské univerzitě v Praze užívané hodnocení známkami 1, 2, 3 a 4, je vhodné v každé skupině připravit otázky různého stupně obtížnosti, aby bylo možné odlišit vynikající znalosti od dobrých a dobré od ještě vyhovujících.

Nastavení mezníků pro jednotlivé klasifikační stupně je pak vhodné provést s ohledem na obtížnost otázek, například:

$V \leq 0.5$	hodnocení „Nevyhověl“,
$0.5 < V \leq 0.65$	hodnocení „Dobře“,
$0.65 < V \leq 0.8$	hodnocení „Velmi dobře“,
$V > 0.8$	hodnocení „Výborně“.

U látky, kde je požadována vyšší míra jistoty znalostí je potřeba limit 0.5 pro úspěšné hodnocení adekvátně zvýšit a příslušně upravit i meze pro jednotlivé klasifikační stupně.

4. Modely výpočtu a procesu

4.1 Výukový systém

Jedním z nejdůležitějších problémů v elektronické výuce je, jakým způsobem zajistit vhodné kontrolování práce a přezkušování studentů a jakým způsobem z těchto informací získávat vypovídající zpětnou vazbu.

Dle [6] většina stávajících řešení (realizovaných většinou na bázi LMS – Learning Management System) obsahuje pouze možnost vytvářet sadu otázek, na které existuje žádná, jedna či více správných odpovědí ze zadané nabídky (testy s volenou odpovědí). Výsledkem těchto testů je pouze zjištění označující, kolik procent dotazů testovaný správně zodpověděl a není rozlišeno, z jaké oblasti znalosti či neznalosti jsou. Jistým vylepšením je přiřazení vah jednotlivým otázkám a tudíž vhodným nastavením vah donutit studenta vypracovat alespoň část úloh z každého tématu. Problémem je správné nastavení vah.

V [6] autor tvrdí, že hledání řešení problému se může odvíjet ve dvou základních rovinách:

- posouzení, zda je správná forma testu s volenou odpovědí,
- problematika, zda jsou znalosti testovány rovnoměrně.

Co se týče první otázky, je nasnadě, že jiné druhy testů, než s volenou odpovědí, nelze pomocí dnešních počítačů exaktně vyhodnotit. Doplnující otázky vyžadují na straně počítače provést lexikální a sémantickou kontrolu odpovědi. I pokud bude otázka definována jednoznačně a bude se ptát na konkrétní výraz, není vzhledem k rozmanitosti jazyka a počtu synonym zaručeno její správné vyhodnocení.

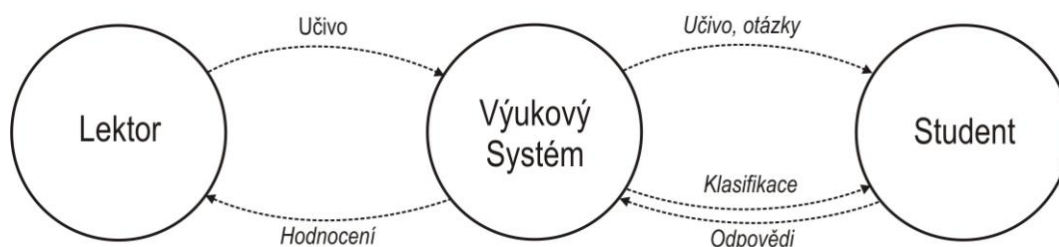
Odpovědi na druhou otázku se zabývá řada prací, zejména [6] a [7] vycházející také z [8] a [9]. Tyto práce se zabývají modelováním průchodu studenta výukovým systémem a definují průběh testování jako pohyb v Petriho síti, kde jednotlivými uzly jsou testovací otázky. Počet možností, jak reagovat na odpověď (ať již správnou či špatnou) je značná.

Vždy půjde o jednu ze dvou základních možností, pohyb v rámci testovací sítě (k jiné sadě otázek, např. zaměření se na důkladnější prostestování uvedené oblasti) nebo pohyb mimo síť (ukončení testu, např. přesun v hypertextu na vysvětlení nepochopeného jevu). Dle [6] je pohyb ve vyhodnocovací síti základem pro konstrukci LMS a tvorba testování je pouze podsítí sítě pro celý kurz, který může být podsítí celého distančního studia.

Dle [7] je potřeba pro možnost modelování průchodu jednotlivého studenta studiem zavést vhodný komunikační systém, který obsahuje základní komponenty příslušného studia a vazby mezi nimi. Takovýto komunikační systém by měl obsahovat minimálně dvě základní vazby:

- přenos informací od lektora ke studentovi a
- přenos informací od studenta k lektorovi.

Tyto typy komunikace ve výukovém systému je znázorněn na obrázku 1.



Obr. 1 – Struktura výukového systému (převzato z [6], [7])

Z tohoto důvodu jsou v [6] a [7] definovány tyto dva základní typy komunikace mezi výukovým systémem, lektorem a studentem:

- **P** (prezentace učiva) – přenos informací od lektora ke studentovi, formou poskytnutí výukových materiálů, odpovědí na dotazy, apod.
- **T** (examinace, testování) – přenos informací od studenta k lektorovi, například formou testů.

Ve výukovém systému mohou bez účasti lektora probíhat také další typy komunikace. Může jít o komunikaci mezi studenty navzájem, formou diskusního fóra, chatu či vzájemné výměny studijních materiálů, zkušeností, znalostí, apod. Také může jít o komunikaci pouze mezi studentem a systémem, například prostřednictvím cvičných testů, pomocí kterých má student možnost ověřit si své znalosti. Test je systémem vyhodnocen a studentovi je poskytnuto správné řešení. Lektor by měl mít možnost účastnit se této komunikace alespoň pasivně, tj. shlédnout komunikaci studentů či výsledky testů.

Na základě [6] a [7] lze takovýto systém považovat za jistý typ expertního systému. Jeho bázi dat představují obsahy prezentačních nebo testovacích lekcí, bázi znalostí vazby mezi jednotlivými uzly (lekcemi) a inferenční mechanismus řídicí systém, který řídí výukový proces.

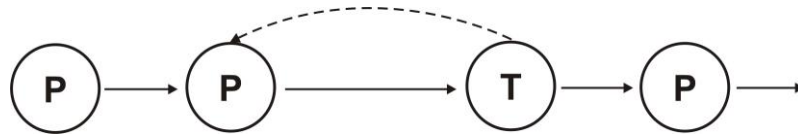
Dle [6] a [7] lze z hlediska typů vazeb mezi prezentační a examinační částí rozlišit několik struktur řízení výukových systémů:

- **Struktura výukového systému s lineárním základem** – prezentace učiva je ve formě lekcí (modulů, kapitol, sekcí), sestavených do lineární posloupnosti (Skinnerův nebo Presseyho lineární program), která po určitých logických celcích obsahuje testovací lekci. Testovací lekce pak při neúspěchu vrací studenta na některou z předchozích výukových lekcí

Začátek vývoje programovaného učení je spojován se jmény B. F. Skinnera a S. L. Presseyho. Profesor psychologie S. L. Pressey v roce 1920 sestrojil testovací stroj na objektivní testování inteligence a zjistil, že jeho stroj nejen testuje, ale i učí. Za vznik programovaného učení je označován rok 1954. Tehdy na vědecké konferenci v Pittsburgu vystoupil profesor psychologie Harvardské univerzity B. F. Skinner s přednáškou na téma Věda o učení a umění učit se. V 60. letech Skinner Presseyho objev znovupoužil, jako vyučovací stroj k programovanému učení.

Dle [14] je u programovaného učení každý krok předem určen a uživatel nemá žádný prostor pro vlastní iniciativu. Program určuje, co má uživatel nastudovat a pak

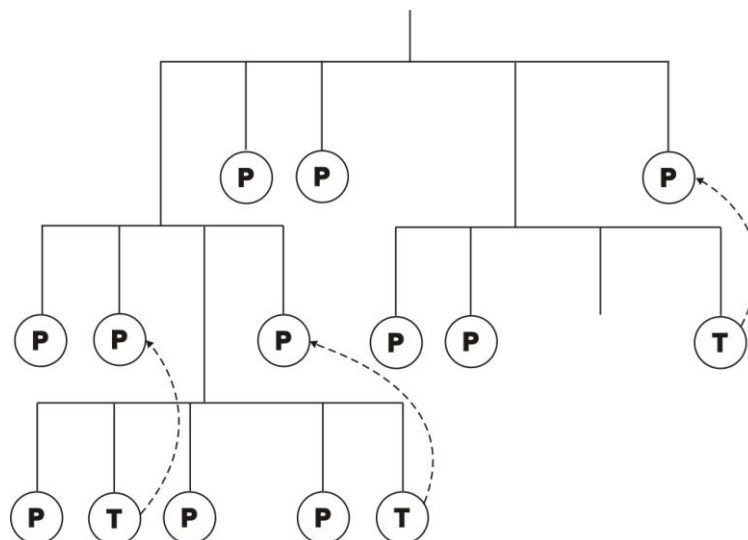
mu předloží kontrolní otázku. Podle odpovědi se program větví – buď pokračuje dál nebo se vrací na výklad probírané látky.



Obr. 2 – Struktura systému s lineárním základem (převzato z [6], [7])

Obloučkovitá čárkovaná šipka se v systému vyskytuje pouze v případě neúspěchu studenta v testovací lekci, kdy je student navrácen zpět do některé z předchozích prezentačních lekcí, k podrobnějšímu prostudování učební látky. Pokud je student v testu úspěšný, postupuje „lineárně“, tj. do prezentační lekce následující.

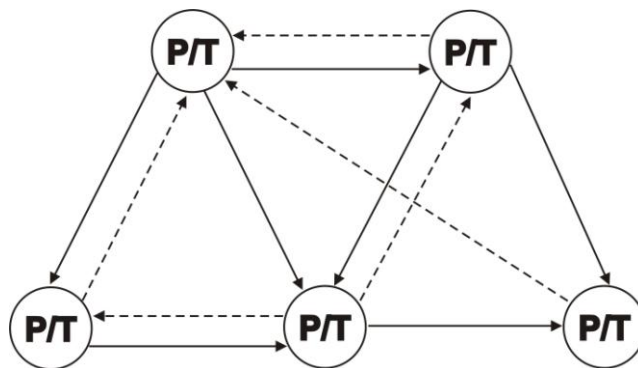
- **Struktura výukového systému se stromovým základem** – prezentační a examinační lekce jsou uspořádány do stromové struktury podle logických vazeb mezi lekcemi. Testovací lekce v případě neúspěchu studenta provede návrat do prezentační lekce v nejbližším nadřazeném uzlu stromu.



Obr. 3 – Struktura systému se stromovým základem (převzato z [6], [7])

Dle [6] obdobně jako v případě systému s lineárním základem, i zde vývoj procesu proběhne po čárkovaných šipkách pouze v případě neúspěchu studenta v testu, kdy je systémem navrácen do některé z předchozích výukových lekcí. Pokud student postupuje skrz testovací lekce úspěšně, systém ho zpět nevrací a šipky se v systému nevyskytují.

- **Struktura výukového systému se síťovým základem** – prezentační a examinační lekce jsou uzly sítě a mezi nimi jsou vazby související s logickou návazností učiva. Uzly mají i testovací obsah, proto existuje také vazba na předchozí uzel, jež je využita v případě neúspěšného provedení testu (na obrázku čárkovaný spoj). Síťová struktura svou obecností poskytuje nejlepší možnost skladby lekcí a testů se vzájemnými vazbami. Uzly P/T představují jiný (například lineární) systém prezentačních a examinačních lekcí (obvykle sekvence prezentačních lekcí zakončená lekcí examinační. Na základě celkového výsledku uzlu je vyhodnocen další postup sítě.



Obr. 4 – Struktura systému se síťovým základem (převzato z [6], [7])

Za přínosy struktury výukového systému se síťovým základem lze dle [6] a [7] pak považovat tyto:

- student může při výuce v systému postupovat více možnými cestami a může vynechat jednotlivé úrovně systému,
- při neúspěchu studenta v testovací lekci je možné vrátit jej do výukové lekce v libovolné úrovni sítě, nejen do úrovně bezprostředně nadřazené.

4.2 Formální modelování výukového procesu

V matematické teorii výpočtu se pro popis průběhu výpočetního procesu využívá řada formálních modelů. Vzniká přirozená otázka, zda by nebylo možné tyto modely, po jejich případné modifikaci, využít i v elektronické výuce, a to pro modelování procesu průchodu studenta výukovým kurzem a využít je jak pro jeho navigaci, tak i pro zprostředkování zpráv uživatelům (tj. student, lektor, tvůrce kurzu, apod.).

V následující rešeršní části práce bude stručně zmíněno o nejčastěji užívaných modelech tohoto typu a pokusíme se o jejich stručné zhodnocení z hlediska cílů této disertační práce.

4.2.1 Konečný automat

Konečný automat (též *FSM z anglického finite state machine*) je teoretický výpočetní model používaný pro studium vyčíslitelnosti a obecně formálních jazyků. Popisuje velice jednoduchý model stroje, který může být v jednom z konečně mnoha stavů, mezi kterými přechází na základě symbolů, které čte ze vstupu. Množina stavů je konečná (odtud název), konečný automat nemá žádnou další paměť kromě informace o aktuálním stavu. Konečný automat je velice jednoduchý výpočetní model, který dokáže rozpoznávat pouze regulární jazyky. Konečné automaty se používají pro zpracování regulárních výrazů, např. jako součást lexikálního analyzátoru v překladačích.

Konečný automat je dle [17] definován jako uspořádaná pětice prvků $(Q, \Sigma, \sigma, q_0, F)$, kde:

- Q je neprázdná konečná množina stavů (*uzlů* - v grafické reprezentaci),
- Σ je neprázdná konečná množina vstupních symbolů, nazývaná *abeceda*,
- σ je tzv. *přechodová funkce* (též *přechodová tabulka*), popisující pravidla přechodů mezi stavy. Je to zobrazení $\sigma : Q \times \Sigma \rightarrow Q$. Přechod je určen stavem, ve kterém se automat nachází a symbolem, který přichází na vstup automatu nebo který je čten na vstupu automatu,

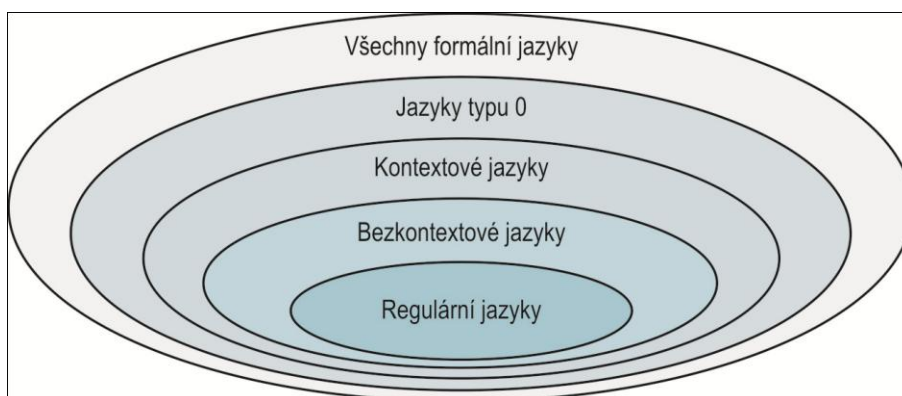
- q_0 je počáteční (iniciální) stav, $q_0 \in Q$,
- F je neprázdná množina koncových stavů, $F \subseteq Q$.

Dle [17] lze přechod definovat také tak, že v každém bodě tabulky nemusí být jeden následný stav, ale celá množina stavů. Takový automat se nazývá *nedeterministický konečný automat* (Oproti *deterministickému konečnému automatu*, který v každém místě tabulky obsahuje právě jeden cílový stav). V přechodové tabulce nedeterministického automatu je také navíc sloupeček pro *prázdný vstup*, označovaný ε (ε obecně v celé teorii formálních jazyků označuje prázdné slovo; musí platit, že $\varepsilon \notin \Sigma$). Lze dokázat, že výpočetní síla deterministických a nedeterministických automatů je tatáž.

4.2.1.1 Popis činnosti konečného automatu

Na počátku se automat nachází v definovaném počátečním stavu. Dále v každém kroku přečte jeden symbol ze vstupu a přejde do stavu, který je dán hodnotou, která v přechodové tabulce odpovídá aktuálnímu stavu a přečtenému symbolu. Poté pokračuje čtením dalšího symbolu ze vstupu, dalším přechodem podle přechodové tabulky atd., až do přečtení posledního znaku zpracovávaného slova. Skončí-li automat v některém z cílových stavů, je slovo přijato. Pokud skončí mimo množinu cílových stavů, je slovo odmítnuto. Znak ε v tabulce nedeterministického automatu má význam možné změny stavu bez odebrání symbolu ze vstupního slova.

Množina všech řetězců, které konečný automat přijme, tvoří *regulární jazyk*. Dle [16] a [21] lze dokázat, že regulární jazyky jsou právě ty jazyky, které lze generovat pomocí tak zvaných regulárních gramatik, to je gramatik Chomského třídy 3. Právě tak lze dokázat, že regulární jazyky jsou právě ty jazyky, jejichž slova lze popsat pomocí tak zvaných regulárních výrazů.



Obr. 5 – Chomského hierarchie tříd jazyků

4.2.1.2 Ukázka činnosti konečného automatu

Jako příklad lze uvést tento deterministický konečný automat:

$$S = \{S_0, S_1, S_2\}$$

$$\Sigma = \{0, 1\}$$

σ = viz tabulka 1

$$q_0 = S_0$$

$$F = \{S_0\}$$

Stav	0	1
S_0	S_0	S_1
S_1	S_2	S_0
S_2	S_1	S_2

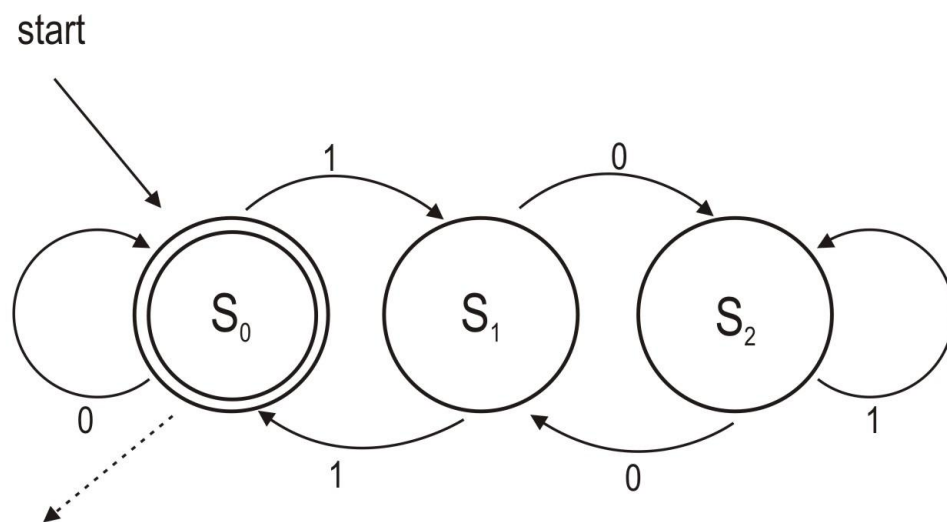
Tab. 1 – stavy deterministického konečného automatu

Pokud má daný automat zpracovat vstup 1011, bude to probíhat takto: Na počátku je automat ve stavu S_0 . Na vstup přijde první symbol, jednička. Z tabulky vyplývá, že na příchod jedničky ve stavu S_0 automat reaguje přechodem do stavu S_1 . Dále přichází nula, ze stavu S_1 se příchodem nuly přechází do stavu S_2 . Poté přichází jednička, ze stavu S_2 se příchodem jedničky přechází do stavu S_2 (zůstává se ve stejném stavu). Nakonec přichází další jednička, takže automat opět zůstává ve stavu S_2 . Stav S_2

nepatří do množiny F , tudíž tento automat vstup 1011 *nepřijal*, řetězec 1011 *nepatří* do jazyka přijímaného tímto automatem.

Tento konečný automat přijímá regulární jazyk slov, která vyjadřují binární číslo dělitelné beze zbytku třemi, přičemž číslo $1011_2 = 11_{10}$ není dělitelné třemi (má zbytek 2, odpovídající výslednému stavu S_2).

Místo relativně nepřehledného (zvláště pro větší automaty) popisu konečného automatu přímo tabulkou se obvykle používá grafické znázornění, na kterém kolečka znázorňují jednotlivé *stavy* a šipky (s přidruženým vstupním symbolem) mezi těmito kolečky popisují jednotlivé *přechody*. Příklad takového znázornění pro předchozí ukázkový automat je na obrázku 6. Dvojitě kolečko označuje přijímané neboli koncové stavy (v našem případě pouze jeden, S_0), počáteční stav je označen šipkou, někdy s připsaným textem, např. *START*. (Tato notace není jediná, jindy se např. koncové stavy označují tlustším orámováním a dvojitě kolečko označuje počáteční stav, případně koncový stav se označuje šipkou směřující „ven“).



Obr. 6 – Znázornění konečného automatu se stavy S_0 , S_1 a S_2

Důležitou charakteristikou konečného automatu je, že se snaží modelovat takové systémy, jejichž reakce v daném okamžiku závisí i na historii, resp. na vstupech v dřívějších časových okamžicích. Konečný automat si svou historii „pamatuje“ díky

tomu, že je vybaven určitým (a to konečným) počtem stavů a v každém okamžiku se nachází právě v jednom z nich. Jeho reakce na vnější vstupy je pak definována nejen v závislosti na hodnotě samotného vstupu, ale i na stavu, ve kterém se právě nachází (čímž je dána závislost jeho chování na historii). Konkrétní "činnost" konečného automatu pak probíhá v postupných krocích, přičemž v každém z nich podle aktuálního stavu a podle momentální hodnoty svého vstupu přechází do nového stavu.

4.2.1.3 Modelování výukového systému konečným automatem

Konečný automat (resp. sekvenční stroj), je dosti silným prostředkem. Postačuje například pro formální popis většiny přenosových protokolů používaných v počítačových sítích. Je ovšem jen jedním z mnoha nástrojů, které si vytvořily teoreticky orientované vědy o počítačích, a není zdaleka prostředkem nejsilnějším. Existují totiž i takové „činnosti“ (výpočty, algoritmy), které pomocí konečného automatu namodelovat nelze. Důvodem je právě omezení dané konečným počtem stavů automatu, který je díky tomu schopen si pamatovat například jen konečný počet mezivýsledků (resp. konečný počet kroků ze své „historie“). Konečný automat například nelze použít pro namodelování tak banálního výpočtu, jakým je násobení dvou libovolně velkých čísel, neboť při určité velikosti obou činitelů by si již nedokázal zapamatovat potřebné mezivýsledky v tom počtu stavů, které má k dispozici. Obecně lze říci, že každou jednotlivou úlohu, kterou lze řešit algoritmicky, tj. pomocí výpočetního systému, lze řešit i konečným automatem. Neplatí to však o algoritmu jako takovém, který by měl být hromadný, tj. pracovat pro celou, potenciálně nekonečnou množinu daných úloh. Jinými slovy, každý konkrétní algoritmický výpočet prováděný počítačem lze modelovat nějakým konečným automatem, avšak neexistuje konečný automat, kterým by bylo možné modelovat každý algoritmický (na nějakém počítači proveditelný) výpočet.

Jedna z vlastností konečného automatu, který je nastíněný v předešlém odstavci, tj. že konečný automat je schopen zapamatovat si jen konečný počet stavů, omezují jeho využití při modelování výukového systému. Je-li jádro výukového systému tvořené pomocí konečného automatu, nedokáže si systém zapamatovat přirozené číslo a je

schopen rozlišovat pouze konečný počet stavů, omezený předem daným přirozeným číslem N . Pokud by tedy student pro úspěšný průchod výukovým systémem potřeboval více jak těchto N stavů, konečný automat by nebyl schopen všechny tyto stavy uchovat. Při užití konečného automatu pro modelování procesu výuky musí být tedy brána v potaz výše uvedená omezení.

Přesný matematický důkaz omezení možnosti konečného automatu vyplývá z Nerodovy věty. Tato věta tvrdí, že jazyk je rozpoznatelný konečným automatem tehdy a jen tehdy, jestliže existuje ekvivalence na množině všech slov nad danou abecedou, která je invariantní vůči rozšíření zprava a jazyk, který má konečný automat rozpoznat je sjednocením konečně mnoha tříd rozkladu podle této ekvivalence. Pokud bychom tedy chtěli použít konečný automat pro modelování výukového systému (kurzu), bylo by potřeba nejprve stanovit apriorní omezení na počty průchodů kurzem a pokud by student „vyčerpal“ tento počet průchodů, kurz by byl ukončen.

4.2.2 Zásobníkový automat

Zásobníkový automat (*PDA* z anglického *pushdown automaton*) je teoretický výpočetní model používaný v informatice při studiu vyčíslitelnosti a formálních jazyků. Dle [15] je zásobníkový automat jednosměrný nedeterministický automat, který má pomocnou, potenciálně neomezenou paměť organizovanou jako zásobník (tedy s přístupem LIFO, pouze na vrchol zásobníku).

Deterministický zásobníkový automat může být dle [17] definován jako uspořádaná sedmice $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$, kde:

- Q je konečná neprázdná množina *vnitřních stavů*,
- Σ je konečná neprázdná množina *vstupních symbolů* – tzv. vstupní abeceda,
- Γ je konečná neprázdná množina *zásobníkových symbolů* – zásobníková abeceda,
- δ je tzv. *přechodová funkce*, popisující pravidla činnosti automatu (jeho program), je definována jako zobrazení $z Q \times (\Sigma \cup \{ \varepsilon \}) \times \Gamma \rightarrow (Q \times \Gamma^*)$, kde Γ^* je řetězec znaků z množiny Γ ,
- q_0 je počáteční stav, $q \in Q$,

- z_0 popisuje symbol uložený na počátku v zásobníku, $z_0 \in \Gamma$,
- F je množina *koncových stavů*, $F \subseteq Q$,
- ε značí prázdný řetězec. Je tedy možné, aby zásobníkový automat provedl přechod, aniž by odebral znak ze zkoumaného slova a odebral ze zásobníku znak, aniž by jej nahradil nějakým znakem či řetězcem znaků.

Zásobníkový automat je stroj, který stejně jako konečný automat obsahuje řídicí jednotku, která je vždy v nějakém ze svých stavů, a která čte ze vstupní pásky slovo (nad nějakou abecedou) a po jeho přečtení rozhodne, zda slovo patří či nepatří do jazyka, který zásobníkový automat rozpoznává. Avšak narušil od konečných automatů, zásobníkový automat využívá navíc zásobníku, neboli jakési paměti typu LIFO. Tedy může ukládat a vybírat symboly pouze z vrcholu zásobníku.

Dle [15] se zásobníkový automat v podstatě skládá z konečného automatu, který má navíc k dispozici potenciálně neomezenou paměť ve formě zásobníku. Obsah tohoto zásobníku ovlivňuje činnost automatu tím, že vstupuje jako jeden z parametrů do přechodové funkce.

Podobně jako u konečného automatu je účelné i u zásobníkových automatů uvažovat jeho nedeterministickou variantu. U této varianty není přechodová funkce zobrazením do $Q \times \Gamma^*$, ale zobrazením do množiny $\exp(Q \times \Gamma^*)$ všech podmnožin $Q \times \Gamma^*$. Do kterého stavu se zásobníkový automat v daném kroku přesune a co zapíše na vrchol zásobníku nemusí tedy vždy být určeno jednoznačně. Může zde být „výběr“ z více možností. Přijetí slova je u nedeterministických zásobníkových automatů definováno analogicky jako u konečných automatů. Slovo je přijato, když lze požadovaného stavu dosáhnout alespoň při jedné z možných voleb během práce automatu.

4.2.2.1 Popis činnosti zásobníkového automatu

Na počátku se automat nachází v definovaném počátečním stavu a zásobník obsahuje pouze počáteční symbol. Dále v každém kroku podle aktuálního stavu symbolů na vrcholu zásobníku a symbolu na vstupu provede přechod, při kterém může

vyjmout ze zásobníku jeden symbol, vložit místo něj jeden nebo více nových symbolů a na vstupu přečíst další symbol nebo ponechat symbol na vstupu i pro další krok. Toto se opakuje.

Po dokončení činnosti (po přečtení celého vstupu, pokud do té doby nedojde k chybě) je rozhodnuto, jestli automat vstupní řetězec přijal. Pokud nebylo slovo na vstupu přečteno celé, protože se automat zastavil, protože neměl definovaný přechod nebo došlo k vyprázdnění zásobníku, automat slovo nepřijímá. Pokud slovo do svého konce bylo přečteno celé, vyhovuje se alternativně podle následujících dvou kritérií:

- **přijímání koncovým stavem** – slovo je zásobníkovým automatem přijato, jestliže existuje možnost, že po zpracování (přečtení) celého slova se automat ocitne v koncovém stavu,
- **přijímání prázdným zásobníkem** – slovo je zásobníkovým automatem přijato, jestliže existuje možnost, že po zpracování celého slova se automat ocitne v situaci s prázdným zásobníkem (narozdíl od konečného automatu může zásobníkový automat rozpoznávat slova nejen tím, že končí v koncovém stavu, ale také tím, že vyprázdní celý svůj zásobník v okamžiku, kdy je slovo přečteno celé).

U nedeterministických zásobníkových automatů jsou obě výše uvedené možnosti ekvivalentní. U deterministických zásobníkových automatů tomu tak není, lze toho ale dosáhnout úpravou jazyka (pomocí doplnění koncových značek).

4.2.2.2 Síla modelu zásobníkového automatu

Nejdůležitější částí zásobníkového automatu je jeho paměť (zásobník). Konečný automat bez zásobníku dokáže rozpoznávat pouze regulární jazyky. „Vnitřní“ konečný automat může být velice jednoduchý, dokonce s jediným stavem – důležitější částí je zásobník, který umožňuje automatu rozpoznávat bezkontextové jazyky, tedy jazyky Chomského třídy 2.

Jelikož se tedy přidáním zásobníku rozšíří třída jazyků, které automat umí rozpoznat, nabízí se otázka, zda by se téhož nedosáhlo přidáním *dalšího* zásobníku.

A skutečně, zásobníkový automat se dvěma zásobníky má výpočetní sílu ekvivalentní Turingovu stroji, neboť jedním zásobníkem může emulovat část pásky vlevo od polohy hlavy, druhým zásobníkem pak část pásky vpravo od hlavy. O Turingovu stroji se zmíníme v dalším odstavci. Dalším přidáváním zásobníků již výpočetní síla neroste.

4.2.2.3 Modelování výukového systému zásobníkovým automatem

Nedeterministický zásobníkový automat by v elektronickém výukovém kurzu byl vhodný pro navigaci studenta při průchodu kurzem (neboť je obecně nedeterministický) i pro poskytování zpráv lektorovi (pomocí zásobníku). Patrně ale neposkytuje studentovi při průchodu kurzem vždy potřebnou volnost, neboť neumožňuje řídit průchod studenta kurzem na základě historie jeho předchozích kroků. Zdá se, že proto, aby bylo možné studenta navigovat v souladu s jeho individuální povahou je účelné brát v úvahu celý kontext jeho dosavadní práce a zohlednit jeho studijní typ. To zásobníkový automat neumožňuje, protože při rozhodování o dalším kroku v navigaci studenta je přístupný pouze údaj obsažený ve vrcholu zásobníku.

4.2.3 Turingův stroj

Turingův stroj (*Turing machine*) je teoretický model počítače, popsáný matematikem Alanem Turingem. Skládá se z procesorové jednotky, tvořené konečným automatem, programu ve tvaru pravidel přechodové funkce a potenciálně nekonečné pásky pro zápis mezivýsledků. Využívá se pro modelování algoritmů v teorii vyčíslitelnosti.

Turingův stroj je dle [22] sedmice $M = (Q, \Gamma, \Sigma, s, \square, F, \delta)$, kde:

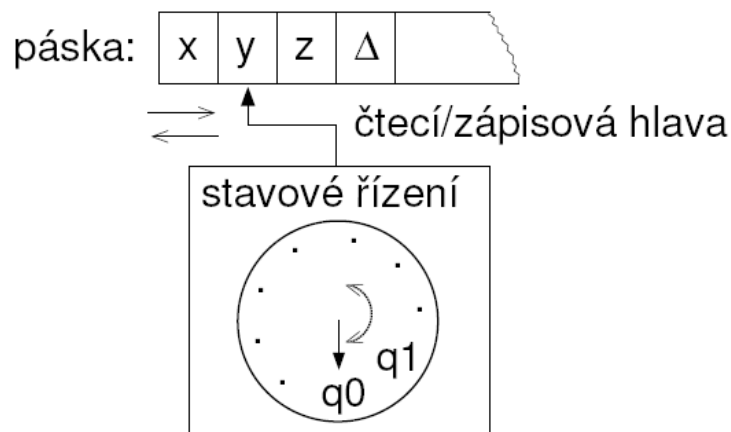
- Q je konečná množina stavů,
- Γ je konečná množina abecedy na pásce, $\Gamma = \Sigma \cup \{\square\}$
- Σ je konečná množina vstupní abecedy,
- $s \in Q$ je počáteční stav,
- $\square \in \Gamma$ je symbol reprezentující mezeru (prázdný symbol),
- $F \subseteq Q$ je množina koncových stavů,

- $\delta \subseteq Q \times \Gamma \rightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ přechodová funkce, kde:
 - \leftarrow značí posun čtecí hlavy vlevo,
 - \rightarrow značí posun čtecí hlavy vpravo.

Turingův stroj má několik základních vlastností:

- Nahradil složitou symboliku matematických kroků. V takovém případě šlo každou konečnou množinu symbolů nahradit množinou obsahující pouze dva symboly (jako je 0 a 1) a prázdnou mezerou, která by kódy původních symbolů oddělovala,
- Turingův stroj má k zápisu nekonečnou pásku skládající se z buněk, do/ze kterých se symbol zapisuje/čte,
- nad touto páskou je možné provádět za pomoci čtecí hlavy operace čtení, zápisu a posunu pásky,
- protože je možné symboly číst, zapisovat nebo se posunovat po pásce, je pro Turingův stroj důležitý vnitřní stav, ve kterém operaci čtení provádíme (čtený symbol a stav tak určují další akci a přechod do dalšího stavu).

Protože se dle [24] chování tohoto stroje vyvíjí podle tabulky přechodů, můžeme říci, že každý následující stav lze jednoznačně určit na základě čteného symbolu a aktuálního stavu. Jeho chování je proto *deterministické*.



Obr. 7 – Deterministický Turingův stroj (převzato z [23])

Výpočet začíná tak, že jsou na pásce uložena počáteční data a vlastní kód programu. Hlava je pak uvedena do stavu, který odpovídá načtení kódu programu a stroj tak započne výpočet, přechází mezi stavy a po skončení většinou zapíše výsledek. Je zřejmé, že takto se chovající model by se dal velmi dobře přirovnat k funkci dnešních počítačů. Přestože dle [22] moderní technologie velmi od třicátých let pokročily, Turingův model lze k popisu chování počítačů použít bez úprav i dnes.

Turingův stroj je v mnoha směrech podobný konečnému automatu, má konečný počet stavů, ve kterých se může nacházet, postupně přijímá jednotlivé vstupy, které dostává ze svého okolí, a reaguje na ně přechodem do nového stavu. Na rozdíl od konečného automatu je ale vybaven navíc ještě i potenciálně nekonečně dlouhou páskou, na kterou si může zapisovat znaky z určité pevně dané abecedy. V každém okamžiku je však na této pásce zapsán jen konečný počet symbolů. Díky nekonečnosti pásky, kterou si Turingův stroj může podle potřeb posouvat, tak má k dispozici nekonečně velkou paměť (obdobně jako zásobníkový automat). Právě díky tomu, a na rozdíl od konečného automatu, je pak schopen namodelovat jakýkoli výpočet, kterého je v principu schopen kterýkoli počítač, třebaže dnes reálně neexistující.

Alan Turing ještě v témže roce, kdy Turingův stroj sestavil, spolu s dalším anglickým matematikem, Alonzem Churchem, vyslovili tézi o tom, že počítače (s nekonečně velkou pamětí) a Turingovy stroje jsou vzájemně rovníčné - tedy že ke každému výpočtu, který je schopen provést počítač, existuje takový Turingův stroj, který jej provede také, a opačně že ke každému Turingovu stroji existuje takový program, který jeho chování bude modelovat. Na rozdíl od konečného automatu lze tedy říci, že existuje Turingův stroj, který je schopen realizovat každý algoritmický výpočet, tj. simulovat všechny existující počítače, i počítače, které budou realizovány v budoucnosti. Přinejmenším všechny, které budou sestrojeny na klasickém von Neumannově principu.

Dle [22] existují i modely nedeterministického výpočtu, nedeterministické Turingovy stroje. Jejich rozpoznávací schopnost je stejná jako u deterministických, počet potřebných kroků však může být diametrálně odlišný. Stejnou výpočetní

schopnost, jako Turingovy stroje mají tzv. RASP (stroje s pamětí s přímým přístupem), jejichž formální definice je bližší reálným počítačům. Jde v podstatě o formalizaci stroje s velmi jednoduchým operačním kódem, který čte a ukládá data do potenciálně nekonečné paměti s přímým přístupem.

4.2.3.1 Modelování výukového systému Turingovým strojem

Pomocí Turingova stroje lze podle Churcheho téze implementovat každý algoritmus. Proto by bylo možné pomocí Turingova stroje implementovat také algoritmus průchodu studenta elektronickým výukovým kurzem. Modelování výukového procesu pomocí Turingova stroje by bylo poměrně obtížné, je proto vhodnější použít jiné, jednodušší modely (viz další podkapitoly).

4.2.4 Petriho síť

Petriho sítěmi (*Petri nets*) je v [10] označována široká třída diskretních matematických modelů (strojů), které umožňují popisovat specifickými prostředky řídicí toky a informační závislosti uvnitř modelovaných systémů. Jejich historie je datována od roku 1962, kdy německý matematik C. A. Petri zavedl ve své disertační práci „Kommunikation mit Automaten“ nové koncepty popisu vzájemné závislosti mezi podmínkami a událostmi modelovaného systému. Petriho sítě vznikly za účelem rozšíření modelovacích možností konečných automatů.

V odborné literatuře existuje řada definic Petriho sítě, v této disertační práci je zvolena ta definice, která je využita jako základ pro modelování průchodu studenta elektronickým výukovým systémem.

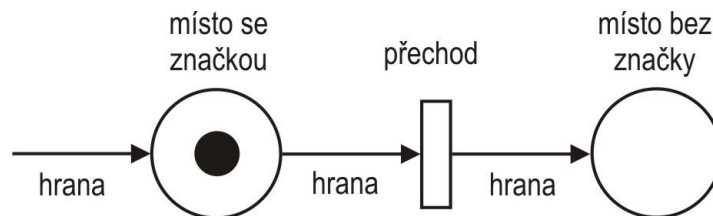
Petriho síť je definována jako orientovaný multigraf $G = (P, T, F)$, kde:

- $P = \{p_1, p_2, \dots, p_m\}$ je neprázdňá konečňá množina uzlů nazývaných místa (pozice),
- $T = \{t_1, t_2, \dots, t_n\}$ je neprázdňá konečňá množina přechodů,
- F je množina orientovaných hran a toků, tzv. přechodová funkce (float),
- $P \cap T = \emptyset$; každému prvku F odpovídá uspořádaná dvojice vrcholů, z nichž jeden patří do P a druhý do T .

Dle [11] je neoznačená Petriho síť orientovaný ohodnocený bipartitní graf. Petriho síť se tedy skládá z uzlů, které jsou navzájem propojeny orientovanými hranami. Označená Petriho síť vznikne umístěním značek (tokenů) do míst neoznačené Petriho sítě:

- **místo** (*place*) – může obsahovat nezáporný celý počet značek,
- **přechod** (*transition*) – v okamžiku aktivace (přeskoku) přechodu jsou odebrány značky ze vstupních míst a přidány značky do výstupních míst přechodu,
- **hrana** (*arc*) – propojují vždy místo s přechodem či přechod s místem.

Místa a orientované hrany jsou ohodnoceny přirozenými čísly či symbolem ∞ . Je-li P množina všech míst v síti, je ohodnocení míst (tak zvaná kapacita míst) zobrazením $P \rightarrow \mathbf{N} \cup \{\infty\}$, kde \mathbf{N} je množina všech přirozených čísel. Je-li H množina všech hran, je ohodnocení zobrazení $H \rightarrow \mathbf{N} \cup \{\infty\}$. Váha hrany je implicitně nastavena na hodnotu 1 a kapacita místa na hodnotu ∞ .



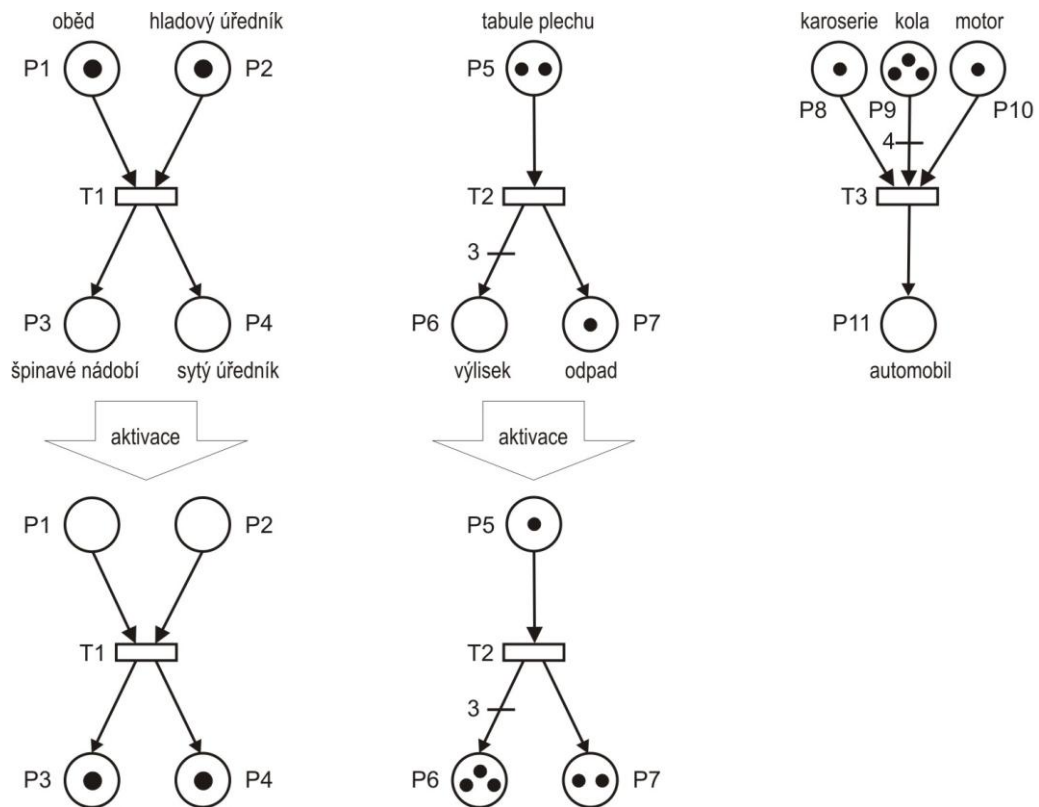
Obr. 8 – Prvky Petriho sítě

Umístění značek v místech Petriho sítě před první aktivací (přeskočením) některého přechodu se nazývá počáteční značení a popisuje počáteční stav systému. Vývoj systému je reprezentován přesunem značek v síti na základě aktivace přechodů. Každé nové značení reprezentuje nový stav systému.

4.2.4.1 Změna stavu systému

Jelikož přechod v Petriho síti může mít více vstupních i výstupních míst, je nutné určit, za jakých podmínek může dojít k přesunu značek po síti, tj. za jakých podmínek dochází k aktivaci přechodů.

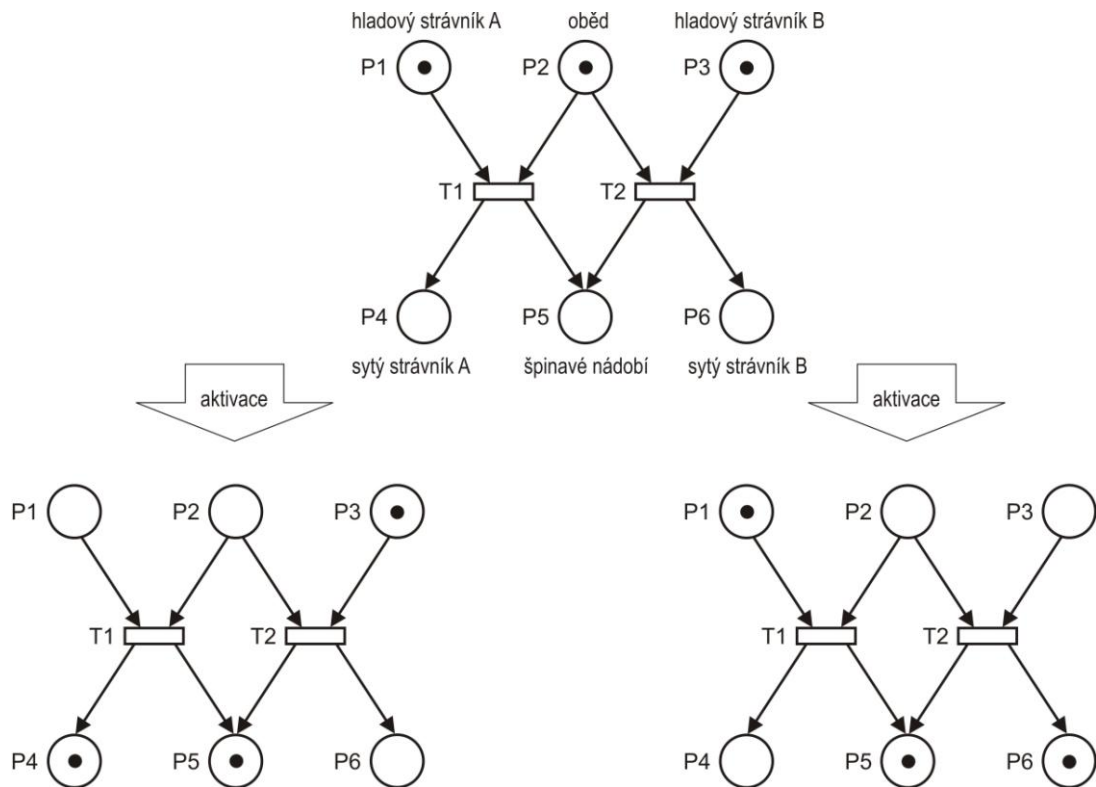
Dle [11] je přechod **uvolněn**, pokud každé jeho vstupní místo obsahuje počet značek větší nebo roven váze hrany, která spojuje vstupní místo a přechod. Při aktivaci přechodu jsou odebrány značky ze vstupních míst a nové značky jsou uloženy do výstupních míst. Počet odebraných (vložených) značek je roven váze vstupní (výstupní) hrany. Aktivace je nedělitelná operace. Pro zjednodušení si lze představit, že doba aktivace je rovna nule. Příklady aktivace autonomní Petriho sítě (autonomní = její vývoj je nezávislý na okolním prostředí) jsou zobrazeny na následujícím obrázku (čísla zapsaná u hran reprezentují jejich váhu, v případě, kdy je váha rovna jedné se číslo nepíše). Princip je ilustrován na příkladu převzatém z [11], kde je vyobrazen průběh oběda a některé kroky výroby automobilu.



Obr. 9 – Příklady aktivace Petriho sítě [11]

Z obrázku je patrné, že přechod T3 nemůže být aktivován, neboť k výrobě automobilu jsou nezbytná čtyři kola, pokud ovšem nechceme zkonstruovat Velorex se třemi koly.

Z předchozího textu je zřejmé, že uvolnění a aktivace přechodu nejsou totožné. To je také názorně vyobrazeno na obrázku 9, kde jsou oba přechody T1 a T2 uvolněny, ale pouze jeden z nich může být aktivován. Tento stav je označován jako **efektivní konflikt**. Úkolem Petriho sítí je tento konflikt modelovat, ale ne řešit.



Obr. 10 – Příklad konfliktu mezi přechody T1 a T2 [11]

Podle [11], lze z pravidel pro aktivaci přechodu odvodit některé typické konstrukce, vyskytující se při modelování systémů prostřednictvím Petriho sítí:

- **paralelismus** (*přechod s více výstupními hranami*) – případ, kdy z jedné sekvence vznikne více souběžně vykonávaných a na sobě nezávislých sekvencí (např. založení procesu v operačním systému),
- **synchronizace** (*přechod s více vstupními hranami*) – případ, kdy několik sekvencí na sebe navzájem čeká a až poté pokračuje jedinou sekvencí (např. schůzka dvou partnerů),
- **výběr** (*místo s více výstupními hranami*) – konstrukce, která umožňuje, aby se značka pohybovala různými cestami (sdílení zdrojů),
- **spojení** (*místo s více vstupními hranami*) – konstrukce, která spojuje několik možných cest vývoje systému do jediné (vrácení sdíleného zdroje).

4.2.4.2 Vybrané vlastnosti a typy Petriho sítí

Petriho sítě jsou vhodným nástrojem pro analýzu systému diskrétních událostí. Poté, co je systém namodelován, lze prokázat jeho vlastnosti, tj. prokázat jeho použitelnost a bezchybnost před jeho praktickou realizací. V následujícím textu jsou dle [11] a [18] uvedeny některé vlastnosti Petriho sítí, respektive typy Petriho sítí definované v závislosti na tom, zda Petriho síť příslušnou vlastnost má či nikoliv.

Před samotným započítím tvorby elektronického výukového systému je také nezbytné vybrat model Petriho sítí, který je z pedagogického hlediska pro modelování tohoto systému vhodný. Z tohoto důvodu bude v rešeršní části této disertační práce poskytnut přehled typů Petriho sítí, které lze využít pro modelování průchodu studenta skrz elektronický výukový systém.

Petriho sítě jsou vhodné pro popis a prevenci tzv. uváznutí procesu. Říkáme, že značení Petriho sítě je ve stavu uváznutí, pokud z daného stavu nelze uskutečnit již žádný přechod. **Petriho síť** se nazývá **síť prostá uváznutí**, pokud v ní neexistuje žádný výpočetní proces, který vede ke značení sítě, které je ve stavu uváznutí.

Dle [11] je **Petriho síť** se nazývá **živá**, pokud pro každé přirozené číslo K a každý její přechod platí, že existuje nějaká výpočetní posloupnost, při které se tento přechod provede alespoň k -krát.

Živá síť nemusí být samozřejmě prostá uváznutí. Tato připomínka je nutná, obecně však není postačující. Mohou existovat sítě, u kterých po konečném počtu využití některých přechodů již při žádném dalším možném vývoji procesu k těmto přechodům nedojde.

Petriho síť se nazývá **pseudoživá**, tehdy a jen tehdy, když pro každý přechod existuje výpočetní proces sítě, při kterém je tento přechod využit alespoň jednou.

Pro aplikaci Petriho sítí na modelování průchodu studenta elektronickým výukovým kurzem je zřejmě třeba se omezit na sítě prosté uváznutí. Jinak by docházelo k situacím, kdy systém přestane pracovat. Další logický požadavek je, aby síť modelující proces byla pseudoživá. Plánovat přechody, které nebudou využity, by bylo zbytečné. Na druhé straně však tyto sítě nemusí být živé. Naopak v typickém případě budou pro modelování průchodu studenta výukovým systémem vhodné sítě, které živé nebudou. To z toho důvodu, že k látce kterou student bezpečně zvládnul se již není nutné znovu vracet.

Dle [11] se **Petriho síť** nazývá **barevná (ohodnocená)**, pokud se jejím hranám a značkám přiřazuje nějaká hodnota. Touto hodnotou může být číslo, v tom případě se obvykle mluví o ohodnocených sítích, grafech apod. Může se také jednat o nějaký symbol, či to může být přiřazení do nějaké množiny. Pokud je těchto možností „málo“, bývá názorné je značit různými barvami. Proto se hovoří o barvené či ohodnocené Petriho síti, tj. tento termín se používá v případě, je-li možností co přiřadit hraně či značce jen přehledně málo.

V některých případech bývá žádoucí určit přechod, který bude aktivován (přeskočen) při současném povolení několika přechodů. **Petriho síť** se tedy nazývá

prioritní, pokud je ke každému jejímu přechodu přiřazena číselná hodnota (priorita), která udává pořadí aktivace přechodů při jejich současném povolení.

Petriho sítě, které umožňují popis systémů, jejichž funkce je časově závislá (pokud například určitá operace trvá nějaký čas), nazýváme **časované petriho sítě**. Existují dva modely popisu časovaných sítí. U prvního způsobu popisu je časování asociováno s místy (tzv. P-časované), u druhého je časování asociováno s přechody (tzv. T-časované).

P-časovaná síť je Petriho síť, u které musí být splněna podmínka, že jestliže je značka umístěna do místa P_i , potom musí zůstat v tomto místě nejméně po dobu d_i . Říkáme, že značka není po dobu d_i k dispozici. Po uplynutí této doby je značka k dispozici a může uvolnit další přechod.

Dle [11] je u **T-časované Petriho sítě** ke každému přechodu T_j přiřazena doba $d_j \geq 0$, po kterou je značka rezervována přechodem T_j .

4.2.4.3 Modelování inteligentních systémů pomocí Petriho sítí

Inteligentní systémy pracující v nějaké předmětné oblasti musí rozlišovat objekty a vztahy mezi nimi. Proto je pro přehledné zobrazení situace vhodné užít grafy se dvěma typy vrcholů. Jeden typ vrcholu znázorňuje objekty, druhý typ vztahy mezi objekty.

Mezi objekty probíhají změny v čase nebo v informačním obsahu. To se znázorní nejlépe pohybem značek po vrcholech grafu. Na tomto principu jsou založeny klasické Petriho sítě.

Označení sítě $G = (P, T, F)$ je funkce $\mu: P \rightarrow \mathbf{N}$, kde \mathbf{N} je množina nezáporných celých čísel, $\mu(p_j)$ se znázorní příslušným počtem značek.

Vývoj (funkce) označení Petriho sítě (P, T, F, μ) je proces změny označení podle následujících pravidel:

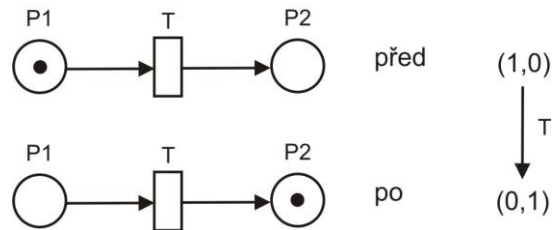
- Jestliže pro přechod $t \in T$ mají všechny vstupní pozice nenulové označení, zahájí se přechod. Výsledkem přechodu je změna označení a to pouze na vstupních a výstupních místech tohoto přechodu, takže ve všech vstupních místech se označení sníží o 1 a na všech výstupních místech se označení o 1 zvýší.
- Jestliže dva nebo více přechodů může přechod zahájit a tyto přechody nemají společné žádné vstupní místo, provedou se oba v libovolném pořadí nebo paralelně.
- Lze-li zahájit více než jeden přechod a dva přechody mají společné vstupní místo, provede se jen jeden z nich. Tato situace se nazývá konflikt a vede k nedeterminismu, pokud tento konflikt není vyřešen jiným pravidlem, vně pravidla pro síť.

Necht' X_1, X_2, \dots, X_m jsou množiny zkoumaných objektů. Vztah nebo relace s doménami X_1, X_2, \dots, X_m je libovolná podmnožina kartézského součinu $X_1 \times X_2 \times \dots \times X_m$.

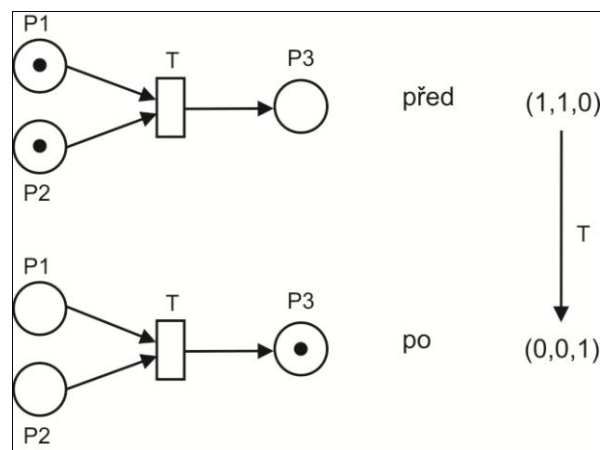
Nyní lze definovat následující modelovací pravidla:

- **Modelovací pravidlo 1** – každému objektu x předmětné oblasti odpovídá místo Petriho sítě s jedinou značkou. Místo bez značky označuje množinu X objektů, které jsou vždy spojeny s každým místem. Jedna značka v daném místě znamená konkrétní objekt $x \in X$.
- **Modelovací pravidlo 2** – každý vztah mezi objekty předmětné oblasti odpovídá jednomu nebo několika přechodům v Petriho síti a incidentním hranám, užitým při těchto přechodech. Přechod mezi místy odpovídá objektům $x_1 \in X_1, x_2 \in X_2, \dots, x_m \in X_m$, tj. vektoru (x_1, x_2, \dots, x_m) , který je prvkem relace R s doménami X_1, X_2, \dots, X_m .

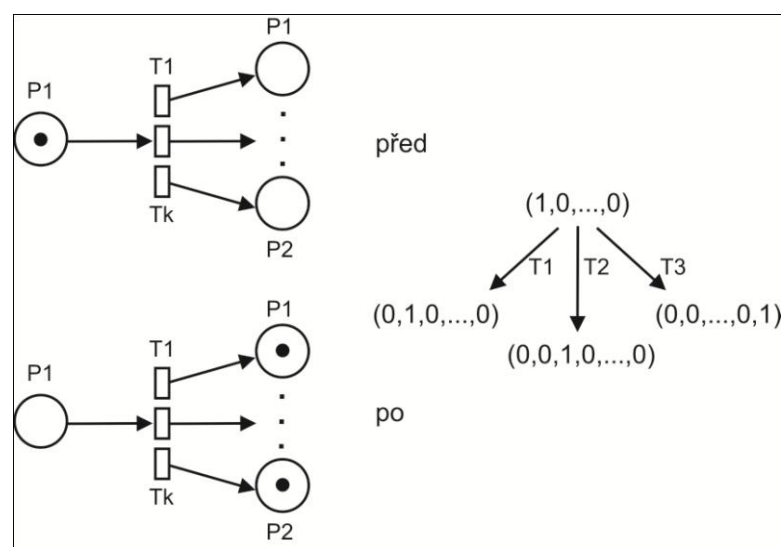
Výše uvedené lze pomocí Petriho sítě vyjádřit i následující funkční závislosti pro funkci jedné či dvou proměnných i pro funkci s alternativním nedeterministickým charakterem.



Obr. 11 – Příklad pro funkci jedné proměnné



Obr. 12 – Příklad pro funkci dvou proměnných



Obr. 13 – Příklad pro funkci s alternativním nedeterministickým charakterem

Právě tímto způsobem je možné pomocí Petriho sítě vhodným způsobem zachytit vztah celek – část, respektive lze takto zobrazit jakoukoliv hierarchii. Směr hran v Petriho síti je od části k celku.

Poslední případ může pak znamenat, že týž objekt může být částí (dědit) více celků. Například pro množinu všech lidí lze zobecnit takto:

- **P1** – množina všech domů,
- **P2** – množina všech zaměstnavatelů,
- **P3** – množina všech národností.

Takto lze například modelovat vztah příčina – následek, včetně toho případu, že jedna příčina může mít více následků a jeden následek více příčin.

4.3 Možnosti modelování výukového systému

Tato kapitola disertační práce si klade za cíl shrnout známé případy modelování průchodu studenta elektronickým výukovým kurzem za pomoci Petriho sítě i dalších vybraných formálních matematických modelů. V kapitole jsou obsaženy modely e-learningových systémů českých i zahraničních autorů.

4.3.1 Modelování výukového systému pomocí barevné Petriho sítě

Zajímavá myšlenka pro využití Petriho sítě k modelování elektronického výukového systému je obsažena v disertační práci [6]. Dle myšlenek obsažených v této práci lze výukový systém modelovat pomocí barevné (ohodnocené) Petriho sítě. Síťový výukový systém lze definovat jako množinu všech podsítí, z nichž každá tvoří určitý vyšší logický celek (modul, kapitolu, výukovou lekci, apod.) a každá může být připojena i k další podsíti, která na prezentované učivo navazuje.

Výukový systém se síťovou strukturou vazeb mezi lekcemi umožňuje implementovat vztahy, které jsou součástí prezentovaného učiva. Jednou ze základních podmínek, která musí být při implementaci splněna, je správná návaznost vyučované látky, neboť není vhodné vysvětlovat určitý odvozený pojem, pokud student nezná

význam pojmu základního. Dalším principem je možnost aktivního zapojení studenta, u něhož je zapotřebí podle povahy učiva ověřovat teoretické znalosti nebo mu uložit trénování a zmechanizování postupů.

Obyčejná Petriho síť může být použita například pro obecné modelování pojmové sítě daného předmětu. Je-li Petriho síť použita jako řídicí mechanismus výukového systému, umožňuje implementovat nejen zmíněné principy, ale i například variantní cesty výkladu nebo testování, což zmírňuje stereotypní projevy stroje a zvyšuje vypovídací schopnost testů.

Výukový systém navržený za pomoci barevné Petriho sítě lze vystavět jako prázdný výukový systém, jehož naplněním teprve učitel určí charakter vyučovaného nebo zkoušeného předmětu.

4.3.1.1 Interpretace sítě

Dle [6] je před definováním modelu výukového systému nezbytné interpretovat všechny prvky barevné (ohodnocené) Petriho sítě vzhledem k vazbám na výukový systém:

- **Místa** – představují pozice výukového systému, které lze přesně charakterizovat určitou množinou zvládnutých pojmů (vymezení pojmů, které má daná výuková množina vysvětlit a protestovat vytváří přesně stanovené místo v síti).
- **Přechody** – představují výukové procedury, každý přechod je jednoznačně charakterizován vysvětlovaným pojmem (nebo množinou pojmů, která je v rámci systému považována za nedělitelný celek a v jiných souvislostech se vždy v jednom celku vyskytuje).
- **Značky** – označení stavu znázorňuje úspěšné dosažení daného stavu, cílem průchodu systémem (výukového nebo examinačního tahu) je označení celé množiny stavů, kterou stanoví učitel. Podle charakteru množiny skutečně označených stavů lze určit obsah zvládnutého učiva.

Proto, aby výukový proces mohl být zahájen z libovolného místa, začíná každý přechod testem porozumění a schopností užít bezprostředně předcházející pojmy. Po jeho úspěšném průchodu následuje vlastní prezentace. Přejchod bez předchůdců (každý jeho uzel náleží do množiny „počátečních uzlů“) neobsahuje test, ale pouze odkazy na množinu pojmů, které se pokládají vzhledem k danému systému za všeobecně známé (např. cílové pojmy jiné podsítě). Výstup systému pak tvoří přechody, z nichž vycházejí pouze „koncové uzly“. Tyto uzly obsahují pouze test, který ověřuje dosažení cíle podsítě. Při vynechání prezentace pojmů v určité množině přechodů systém pouze diagnostikuje vědomosti studenta.

Test u přechodu ověřuje znalost všech bezprostředně předchozích pojmů. Jestliže je v testu znalost určitého pojmu z množiny bezprostředních předchůdců úspěšně ověřena, příslušný uzel dostane značku. V opačném případě je proces výuky/examinace přesunut do uzlu, který odpovídá nezvládnutému pojmu. Tímto postupem lze při čisté examinaci z jakéhokoliv počátečního uzlu dospět do stavu, kdy bude označena množina právě těch uzlů, jejichž pojmy student úspěšně zvládnul.

Podsít', směřující z množiny určitých výchozích stavů k určitému koncovému stavu, nazývá autor disertační práce „výukovým tahem“. Výukový tah charakterizuje samostatnou část učiva, kterou chce pedagog prezentovat (např. výuková lekce, vyučovací blok, jedna kapitola). Dále lze definovat „examinační tah“, tj. takový tah, jehož všechny přechody mají blokováno prezentování pojmů. Doplnkem k tahu předchozímu je „prezentační tah“, který lze definovat jako množinu přechodů, které mají blokováno testování.

4.3.1.2 Systém přechodů

Dle [6] je každý student v síti modelován čtveřicí (J, D, P, L), kde jednotlivé složky mají tento význam:

J – jednoznačná identifikace studenta v daném okamžiku,

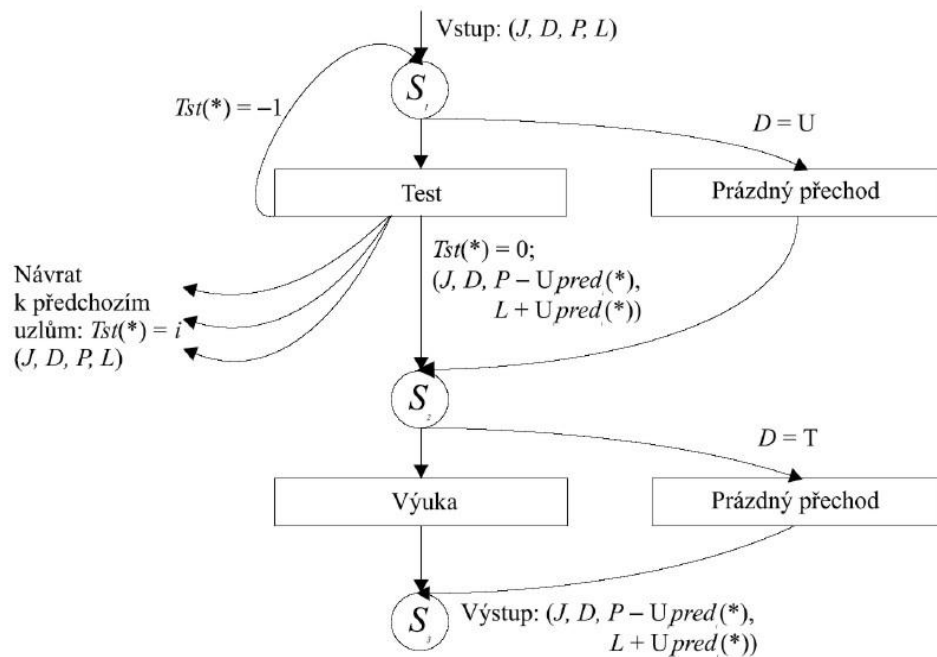
D – druh průchodu sítí, přičemž lze předpokládat tyto druhy průchodů:

- *examinační* – není prováděna výuka, probíhá testování studenta ($D = T$),
- *výukový* – je prováděna výuka i testování ($D = V$),

- čistě výukový (prezentační) – je doplňkem předchozích dvou, provádí se pouze výuka, systém funguje pouze jako učebnice ($D = U$).

P – množina uzlů, kterou má student v síti absolvovat, množinu definuje na počátku studia lektor,

L – množina skutečně absolvovaných uzlů v síti, v daném okamžiku tato množina udává stav, v němž se student nachází – tj. rozdíl mezi množinou P a L umožňuje přesně diagnostikovat vědomosti studenta.



Obr. 14 – Model přechodu výukového systému [převzato z 6]

Při vstupu do přechodového kroku se čtveřice objeví v místě S_1 . Podle složky D pak prochází testem nebo prázdným přechodem. V okamžiku dosažení místa S_2 je množina L obohacena o identifikaci všech bezprostředních předchůdců daného přechodu (identifikace daného přechodu je znázorněna hvězdičkou, předchůdce přechodu vyplývající ze struktury sítě je dán funkcí $pred$, předchůdců může být obecně víc). Současně s obohacením množiny L je o tentýž seznam identifikací ochuzena množina P .

Další postup z místa S_2 závisí opět na složce D čtveřice – buď je nebo není provedena výuka příslušného nového pojmu. Výstupní místo S_3 je současně vstupním místem pro další přechodový krok.

Výsledek testování je popsán celočíselnou funkcí $Tst(idUzel)$, jejíž výsledek určuje další postup čtveřice (provedení přechodu). Hodnotou 0 je modelován úspěch při testu, hodnotou 1 pak stav, kdy je tentýž test opakován (s modifikovanými otázkami – pro určité příklady, kdy výsledek testování není průkazně vyhovující). Kladné hodnoty i testovací funkce označují návrat po i -té cestě k příslušnému předchozímu přechodu.

Testovací funkce vychází z konstrukce testu a je individuální pro každý přechod, protože u každého přechodu jsou na test kladeny jiné nároky a je testován různý počet pojmů.

4.3.1.3 Struktura systému

Autor definoval výukový systém jako konkrétní implementaci simulátoru barevné Petriho sítě. Hlavní částí sítě je množina uzlů, představující jednotlivé pojmy, neboli pojmové skupiny, mezi nimiž jsou definovány orientované hrany vyjadřující relaci logické následnosti těchto pojmů.

Informace o každém uzlu jsou uloženy v textovém souboru, přičemž soubory jsou dány definicemi předchůdce a následníka. Systém lze pak rozšiřovat přidáváním dalších souborů, na něž se vytvoří vazba přidáním nového jména v příslušných předchůdcích.

Definovaný výukový systém se tedy skládá z uzlů, reprezentujících jednotlivé výukové lekce, které jsou sestaveny podle logických návazností učiva. „Vstupem“ do každé lekce je didaktický test, složený z elementů, které zahrnují všechny logické předchůdce daného uzlu. V případě neúspěchu studenta v testu je systémem vyhodnoceno, do kterého předchozího uzlu je předáno řízení.

Autorův návrh výukového systému pomocí barevné Petriho sítě práce se jeví jako velice zajímavý. Pro další výzkum v oblasti výuky může být tento navržený směr přínosem. Tato práce může být základem dalšího výzkumu v modelování e-learningových kurzů.

4.3.2 Možnosti řízení výukových procesů s použitím Petriho sítí a fuzzy logiky

Zajímavý nápad na zobecnění Petriho sítě pro účely modelování výukového procesu je uveden v příspěvku [25] na konferenci o distančním vzdělávání v aplikované informatice. Autoři nejprve shrnují přednosti klasických Petriho sítí pro modelování procesů, spočívající v tom, že je udávají jako nástroj pro modelování procesů, které mohou probíhat souběžně (paralelně) a to jak v asynchronním, tak i synchronizovaném režimu a mohou být distribuované a do určité míry i nedeterministické.

Příspěvek naznačuje, že výukové procesy mají či mohou mít vlastnosti, které klasický model Petriho sítí může postihnout, nepostihuje ale nejasnost procesů, která je pro výuku typická. Snaží se tedy naznačit, že aparát Petriho sítí by bylo vhodné zobecnit tak, aby byly použitelné i pro fuzzy modelování. Bohužel příspěvek lze považovat spíše za naznačení jedné z možných cest, jak rozšířit aparát Petriho sítí tak, aby výukový proces popsal věrohodněji, než za řešení daného problému.

Definici klasické Petriho sítě autoři modifikují tak, že orientované hrany směřující z míst do přechodů charakterizují pomocí zobrazení I množiny $P \times T$ do $\{0,1\}$ a orientované hrany směřující od přechodů do míst sítě pomocí zobrazení O množiny $T \times P$ do $\{0,1\}$. Záměrem autorů bylo zřejmě definovat fuzzy Petriho síť tak, že změní obor hodnot oboru zobrazení I a O , vyjadřujících orientované hrany z dvouprvkové množiny $\{0,1\}$ na kontinuum $\langle 0,1 \rangle$. Tím by bylo možné zachytit mlhavost existence hran v Petriho síti, která by mohla přiblížit model realitě. Fuzzy model však popsán není. Příspěvek nebyl doveden ani do takové hloubky, aby byla navržena definice fuzzy Petriho sítě. Tím méně v příspěvku chybí naznačení toho, jaké neurčitosti ve výukovém procesu by bylo možné fuzzy Petriho sítěmi popsat a jaký by byl přínos fuzzy Petriho sítí k řízení výukového procesu. Příspěvek také pomíjí

problém zpětné vazby a navigace studenta výukovým kurzem. Lze jej tedy považovat jako upozornění na možný směr, jak postihnout vhodným modelem neurčitosti, které ve výukovém procesu mohou nastat. V článku není ani nastíněn popis takového modelu a možnosti jeho využití.

4.3.3 Technologie Petriho sítí při návrhu e-learningových kurzů

O možnosti využití Petriho sítě jako modelu pro průchod studenta elektronickým výukovým kurzem se zmiňuje i příspěvek [26]. V článku jsou zopakovány obecné zásady e-learningu a poukázáno na požadavek, aby kurzy byly orientovány projektově a byly dostatečně flexibilní. Pro tento cíl je třeba věnovat pozornost zejména průvodci studenta kurzem. Pro modelování tohoto průvodce je navrženo užít Petriho síť. V článku je dále stručně popsána známá základní definice Petriho sítě. Chybí však pokus naznačit jak je této definice využito pro modelování průvodce kurzem. Je pouze konstatováno, že na Pedagogické fakultě Ostravské univerzity v Ostravě probíhá kurz Vzdělávací technologie, v jehož realizační fázi se Petriho síť využívají. Příspěvek však neuvádí, jak jsou využity. Z příspěvku však nevyplývá, zda jde pouze o záměr či zda byl průchod studenta výukovým kurzem pomocí Petriho sítě vymodelován a jaké jsou zkušenosti s využitím těchto modelů.

4.3.4 GLIF model průchodu studenta výukovým systémem

GLIF model primárně slouží pro strukturovanou reprezentaci převážně lékařských doporučení ve formě orientovaného grafu obsahujícího pět základních prvků (stav, rozhodování, akce, větvení a synchronizace). Tento model je nejčastěji používaný pro modelování převážně procedurálních znalostí obsažených v textových oborových doporučeních. Dle [12a] byl **GLIF** (*Guideline Interchange Format*) v nynější podobě vyvinut ve spolupráci Univerzity Columbia, Harvardské, McGillovy a Stanfordské univerzity.

Jako každý formální matematický model, tak i GLIF je možné upravit takovým způsobem, aby jej bylo možné využít i v jiných oblastech. Jednou z oblastí je i elektronická výuka. V článku [12a], na kterém se autorka disertační práce spolupodílela, je nastíněn nový návrh, jak druhotně GLIF model aplikovat na sledování

výuky. V následujícím textu je popsána úprava GLIF modelu určeného pro zobrazení průchodu studenta elektronickým výukovým systémem.

Výsledkem výše zmíněného zobecnění GLIF modelu je orientovaný graf $G_{GLIF} = (S, H)$, kde:

- $S = \{s_1, \dots, s_n\}$ je neprázdná konečná množina vrcholů zvaných kroky,
- H je konečná množina orientovaných hran takových, že každé hraně je přiřazena dvojice vrcholů s_i a $s_j \in S$, kde $s_i \neq s_j$.

4.3.4.1 Základní typy vrcholů

Každý vrchol (krok) s_i z množiny S může mít žádný nebo konečný počet vstupů (hran vedoucích do vrcholu s_i) a výstupů (hran vedoucích z vrcholu s_i). Vrchol s_i může být pouze jeden z následujících možných typů:

- **Akce** – představuje specifickou činnost nebo událost. Akcí může být i podgraf, který dále zjemňuje danou činnost. Akce, značená obdélníkem, má právě jeden vstup a právě jeden výstup.
- **Rozhodování** – představuje větvení (výběr následného kroku) na základě splnění logického kritéria (kritérií), kdy další postup grafem je dán výsledkem aritmetického nebo logického výrazu nad konkrétními daty, a nebo rozhodnutí uživatele, kterou částí grafu bude dále pokračovat. Rozhodování, značené kosočtvercem, má právě jeden vstup a dva nebo více výstupů.
- **Větvení a synchronizace** – větvení se používá při modelování nezávislých kroků, které mohou probíhat paralelně a synchronizace slouží pro tyto kroky jako slučovací bod po splnění synchronizační podmínky. Větvení, značené trojúhelníkem s vrcholem nahoru, má právě jeden vstup a dva nebo více výstupů. Synchronizace, značená trojúhelníkem s vrcholem dolů, má dva nebo více vstupů a právě jeden výstup.
- **Stav** – značí stav, ve kterém se zkoumaný objekt nachází při vstupu do modelu nebo po provedení některého předchozího kroku. Stav, značený oválem, má žádný nebo jeden vstup a žádný nebo jeden výstup.

4.3.4.2 Rozhodovací kritéria

Dle [12a] jsou v každém rozhodovacím kroku $s_i \in S$ pro každý jeho výstup $h_{ij} \in H$, kde $j = 1 \dots m$, definována rozhodovací kritéria:

- **strict-in** (h^{SI}_{ij}) – je-li toto kritérium splněno, bude se určitě pokračovat výstupem h_{ij} ,
- **strict-out** (h^{SO}_{ij}) – je-li toto kritérium splněno, výstup h_{ij} je pro další postup zakázán,
- **rule-in** (h^{RI}_{ij}) – při splnění tohoto kritéria je výstup h_{ij} doporučen pro další postup,
- **rule-out** (h^{RO}_{ij}) – při splnění tohoto kritéria není výstup h_{ij} doporučen pro další postup.

4.3.4.3 Model průchodu studenta výukou

Předmětem zkoumání v [12a] je student denního studia, jehož cílem je získat během jednoho semestru zápočet z hypotetického předmětu. Pro absolvování předmětu (dosažení cíle) je třeba splnit následující podmínky:

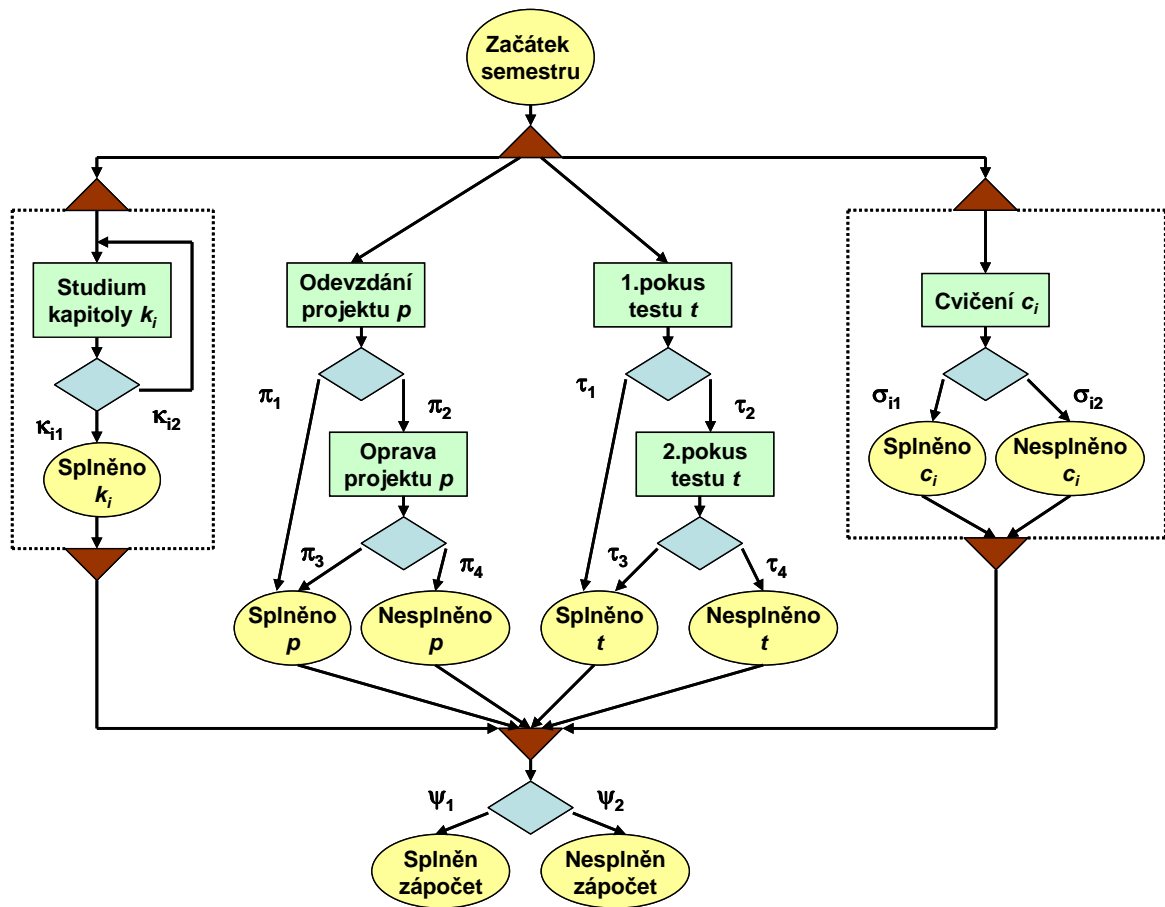
- **prostudování 10 kapitol s teorií týkající se předmětu** – na konci každé kapitoly je několik kontrolních otázek, na které musí student správně odpovědět (odpovídat může v několika pokusech),
- **docházka na cvičení** – je třeba získat minimálně 70% účast na cvičeních,
- **samostatný projekt** – odevzdání samotného projektu na zadané téma. Student musí získat hodnocení minimálně 70%. V samostatném projektu je možné provést jednu opravu,
- **zápočtový test** – kontrolní test z probírané problematiky. Úspěšné absolvování testu znamená získat minimálně 70% z možného hodnocení. Test je možné absolvovat maximálně ve dvou pokusech.

Průběh studia, tj. docházka a výsledky dílčích podmínek, je zaznamenán v systému pro podporu elektronického vzdělávání (e-learningovém systému). Ve stejném systému jsou studentovi k dispozici i potřebné studijní materiály (teoretické kapitoly).

Modelová situace je v GLIF modelu znázorněna jako čtyři paralelní větve (viz obrázek 15), kde jedna větev představuje studium deseti kapitol (k_i), druhá větev absolvování 14-ti cvičení (c_i), třetí větev splnění samostatného projektu (p) a poslední větev absolvování zápočtového testu (t). Tečkované čáry znamenají opakování téže struktury, tj. 10 krát studium kapitoly a 14 krát absolvování cvičení.

- **studium kapitoly (k_i)** – student opakuje studium kapitoly k_i dokud neodpoví správně na kontrolní otázky, tj. dokud není strict-in kritérium větve κ_{i1} pravdivé,
- **absolvování cvičení (c_i)** – každého cvičení c_i se student buď zúčastní (pravdivé strict-in kritérium σ_{i1}) nebo nezúčastní (pravdivé strict-in kritérium σ_{i2}). Účast (neúčast) je zaznamenána vedoucím cvičení,
- **splnění projektu (p)** – student splní projekt v případě, že je splněno strict-in kritérium hrany π_1 nebo π_3 . Kritéria jsou definována následovně $\pi_1^{SI} = \text{hodnocení}(p) \geq 0,7$ a $\pi_3^{SI} = \text{hodnocení}(p) \geq 0,7$. Strict-in kritéria větví π_2 a π_4 jsou negací strict-in kritérií hran π_1 a π_3 ,
- **absolvování testu t** – student úspěšně absolvuje zápočtový test v případě, že je splněno strict-in kritérium hrany τ_1 nebo τ_3 . Kritéria jsou definována následovně $\tau_1^{SI} = \text{hodnocení}(t) \geq 0,7$ a $\tau_3^{SI} = \text{hodnocení}(t) \geq 0,7$. Strict-in kritéria větví τ_2 a τ_4 jsou negací strict-in kritérií hran τ_1 a τ_3 ,
- **zápočet** – zápočet student získá, jestliže je splněno strict-in kritérium hrany ψ_1 , tzn. student splní studium všech kapitol (k_i), absolvuje minimálně 70% cvičení c_i , splní projekt p a úspěšně absolvuje test t . Strict-in kritérium hrany ψ_2 je negací strict-in kritéria hrany ψ_1 .

Všechna kritéria strict-out jsou ve všech případech negací strict-in kritérií. Kritéria rule-in a rule-out nejsou v tomto modelu použita vůbec.



Obr. 15 – GLIF model průchodu studenta vyučováním předmětem [12a]

Dle autorky této disertační práce použití GLIF modelu pro modelování průchodu studenta elektronickým výukovým systémem skýtá jistá omezení. Pokud student prokáže neznalost z konkrétního tématu vyučované látky, tj. nesplní podmínky testu, týkající se kapitoly k_i (větev κ v obrázku 17), navrátí jej systém pomocí cyklu pouze k opětovnému prostudování učiva kapitoly k_i . Systém však nedokáže zjistit, zda student zvládl nastudovat látku, která učivu v kapitole k_i předchází a je nutná ke zvládnutí této kapitoly. Na rozdíl od modelu zobecněné Petriho sítě, navržené v této práci, se nelze v GLIF modelu pohybovat zpětně (proti směru šipek) a testovat, zda student ovládá základní látku, jejíž znalost je předpokladem učiva navazujícího.

4.3.5 Využití Petriho sítí k analýze chování studentů v e-learningovém prostředí

Těsně před dokončením této disertační práce, v době posledních úprav textu měla autorka možnost seznámit se s článkem [27], který může mít značný význam pro další výzkum modelů počítačové podpory výuky na základě Petriho sítí. Článek je věnován modelování e-learningového systému s využitím webových služeb.

V článku je podrobně popsána tzv. LBPN-síť (*Learning Behavioral Petri Net*), která je modelem stavu e-learningového systému jako celku. Na rozdíl od přístupu využitého v této disertační práci není tedy cílem modelu postihnout navigaci konkrétního studenta výukovým systémem, ale modelovat stav systému spolu se všemi jeho uživateli, pracujícími se systémem paralelně.

Model LBPN je založen na obarvené Petriho síti se značkami tří barev. Místa v této síti se skládají ze čtyř typů uzlů (jsou obarvena čtyřmi barvami $P = P_N \cup P_T \cup P_S \cup P_P$), kde:

- $P_N = \{p_{N1}, p_{N2}, \dots, p_{Nr}\}$ reprezentují jednotlivé stavy systému,
- $P_T = \{p_{T1}, p_{T2}, \dots, p_{Tn}\}$ reprezentují skutečnost, že student s_i zůstává ve výukové jednotce po specifikovanou dobu,
- $P_S = \{p_{S1}, p_{S2}, \dots, p_{Sv}\}$ jsou místa určená pro zapamatování výkladu látky a testů,
- $P_P = \{p_{P1}, p_{P2}, \dots, p_{Pw}\}$ jsou místa pro rozhodnutí o následujícím stavu.

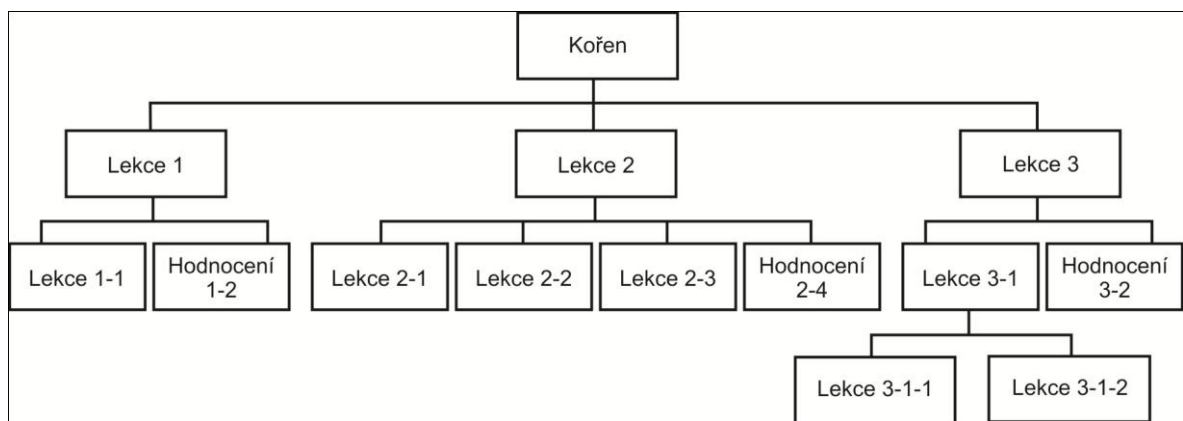
V síti se pohybují tři typy značek znázorněné černými, šedými a bílými kolečky. Množina všech značek, jejichž umístění popisuje konkrétní stav sítě je sjednocením $\Sigma = S \cup C \cup A$, disjunktních množin S , C a A , kde:

- $S = \{s_1, s_2, \dots, s_x\}$ označuje jednotlivé studenty,
- $C = \{c_1, c_2, \dots, c_y\}$ označuje jednotlivé výukové celky a
- $A = \{a_1, a_2, \dots, a_z\}$ označuje jednotlivé testy ve výukovém systému.

Černé značky označují jednotlivé studenty z množiny S , šedivé výukové jednotky z C a bílé testy z množiny A . V článku jsou podrobně popsány algoritmy pro realizaci následujících akcí:

- opuštění kurzu studentem,
- postup studenta kurzem strategií do hloubky (*dept-first*),
- postup studenta kurzem strategií do šířky (*breath-first*),
- vynechání dané jednotky studia studentem,
- opakování (opravné) studium dané jednotky studentem.

Celý model je ilustrován na příkladu jednoduchého kurzu s hierarchickou strukturou výkladu a testů, který je vyobrazen na následujícím obrázku 16.



Obr. 16 – Příklad struktury kurzu (převzato z [27])

Článek dále uvádí příklady průchodu studentů kurzem dané struktury a výsledky vyhodnocení pokusu, kdy tento kurz v roce 2004 navštívilo 177 studentů. Studenti byli rozdělení do tří typů:

- A – pomalí,
- B – přechodní,
- C – vytrvalí.

V článku jsou porovnány skutečné výsledky studentů s odhady výsledků získaných z daného modelu a je poukázáno, že model je relativně úspěšný

v předpovídání konečného úspěchu či neúspěchu studenta ze sledování jeho pohybu kurzem.

Článek může být základem dalšího výzkumu v modelování e-learningových kurzů jako celku při jejich paralelním používání více studenty. Jeho zaměření je však odlišné od cíle, který byl vytýčen pro tuto disertační práci.

4.3.6 Výzkum zatížení e-learningového systému založeného na Petriho sítích

Využitím Petriho sítí pro modelování e-learningu se zabývá i příspěvek [28]. Příspěvek naznačuje užití Petriho sítí pro modelování diskrétních událostí ve výukovém systému jako celku. Přednost Petriho sítí spatřují autoři především v tom, že sítě umožňují sledovat „diskrétní události se spojeným chováním“. Cílem příspěvku je především naznačit, jak lze využít Petriho síť pro modelování zátěže výukového systému jeho uživateli. Příspěvek není věnován otázce jak řídit a hodnotit průchod jednotlivých studentů výukovým systémem. Je tedy zaměřen jiným směrem, než je cíl této disertační práce. Nicméně obsahuje některé zajímavé možnosti zobecnění Petriho sítí.

V příspěvku je popsána tak zvaná GSPN síť (*Generalized Stochastic Petri Net – zobecněná statistická Petriho síť*), v níž jsou místa interpretována pouze jako omezení ve výukovém procesu a veškeré akce se provádějí v přechodech.

Autor předpokládá, že takovýto model by mohl být využit ke sledování možných zatížení modelovaného výukového systému paralelně přistupujícími studenty. Konstrukci modelu však obecně nepopisuje. Článek pouze obsahuje obrázek, ve kterém je naznačeno, jak by mohl vypadat model pro zjednodušený výukový problém, neuvádí se však využití daného modelu pro vyhodnocení zátěže. Není zde ani obsažena diskuse v čem spočívá stochastičnost navrženého modelu. Příspěvek lze tedy považovat pouze za podnět k jednomu možnému využití Petriho sítí, nikoliv za rozpracování ani za přesný popis předpokládaného modelu.

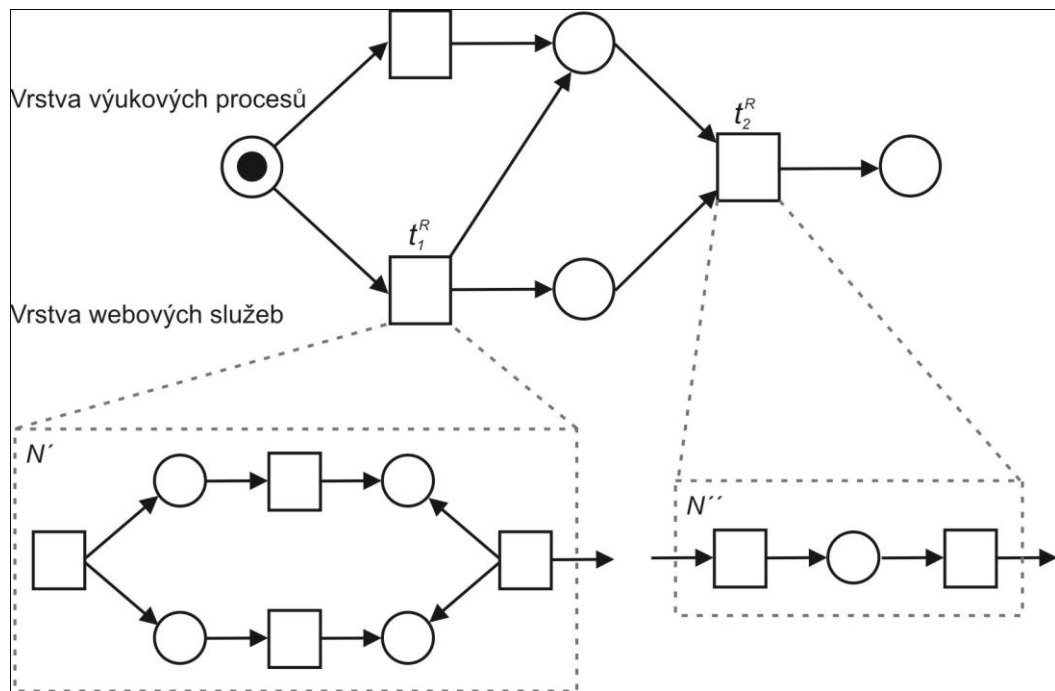
Pokud by se ukázalo, že cesta naznačená autorem by byla využitelná, týkala by se spíše problému odhadu možného provozního zatížení výukového systému studenty, než problému jak ve výukovém systému využít zpětnou vazbu a navigovat práci studenta, ke kterému chce přispět předkládaná disertační práce.

4.3.7 Návrh e-learningového systému pomocí hierarchického modelu Petriho sítí

Aplikací modelu založeného na Petriho sítích se zabývá i článek [29]. V tomto příspěvku se proces strojového učení rozkládá do dvou hierarchických úrovní. V obecnější úrovni se definují objekty výuky (*learning objects*), kterými mohou být libovolné entity (digitalizované i nedigitalizované), se kterými pracují nebo které využívají či na ně odkazují technologické prostředky pro podporu výuky. Mezi takovéto objekty patří vyučovaná témata, řešené problémy i zadávané úlohy. Tyto objekty jsou charakterizovány vstupními předpoklady a podmínkami, které mají splňovat výstupy. Proces strojem podporovaného vyučování je modelován na úrovni těchto objektů výuky.

Pro realizaci výukového procesu jsou užívány zpravidla webové služby. Tyto služby tvoří nižší vrstvu v hierarchické úrovni strojové podpory vyučovacího procesu, nazývanou autory úrovní webových služeb. Do ní patří například kalendáře, diskusní fórum, oběžníky, zprávy, systémy pro zkoušení i navigace studenta kurzem. Jednotlivé akce vyšší úrovně výukového procesu jsou realizovány systémem několika webových služeb. Jeden krok výukového procesu (prvek vyšší úrovně) je realizován několika webovými službami užitými v dané struktuře.

Autoři ukazují, že obě vymezené úrovně lze popsat pomocí klasických Petriho sítí s tím, že jednomu uzlu v síti vyšší hierarchické úrovně pro výukový proces odpovídá celá síť, zachycující systém webových služeb užitých pro realizaci daného kroku. Konceptuální kostra výukového systému je pak znázorněna na obrázku 17 převzatého z citovaného článku.



Obr. 17 – Konceptuální kostra e-learningového systému (převzato z [29])

Vyšší úroveň představuje síť výukového procesu, na nižší hierarchické úrovni je pak popsána její realizace, která spočívá v rozkladu jednotlivých přechodů v této úrovni na několik Petriho sítí, které modelují webové služby realizující tyto přechody. V práci jsou uvedeny příklady jak realizovat síť webových služeb pro typické přechody v síti výukového procesu.

Hlavní cíl navrženého dvouúrovňového hierarchického popisu výukového procesu pomocí Petriho sítí je možnost formální verifikace korektnosti navrženého procesu z hlediska možnosti dosažení cíle, prevence uváznutí a nekonečných cyklů. Autoři podrobně diskutují možnosti formální verifikace takto popsaného návrhu.

Příspěvek je třeba hodnotit jako důležitý z toho důvodu, že umožní navrhovateli výukového systému od sebe oddělit koncepční věcné úvahy o konstrukci výukového systému od úvah o jeho realizaci pomocí dostupných webových služeb a soustředit se tak vždy pouze na jeden aspekt problému. Práce však neřeší problém zpětné vazby a navigace studenta kurzem, což je úkolem řešeným v této disertační práci.

4.3.8 Návrh výukového procesu pomocí zrcadlí Petriho sítě

V příspěvku [30] autoři navrhují popis vývoje elektronického výukového systému pomocí Petriho sítě na dvou úrovních. U e-learningového systému autoři odlišují dva procesy:

1. Proces průchodu studenta počítačem podporovaným kurzem v situaci, kdy struktura kurzu i jeho obsah je statický,
2. Proces postupného vývoje kurzu na základě zkušeností získaných z jeho využívání studenty a požadavků pedagogů, kteří s kurzem pracují a jsou jeho účastníky.

Oba tyto procesy autoři navrhují popsat pomocí Petriho sítí. Stav systému v daný okamžik je modelován tak zvanou sítí základní úrovně (*base-level Petri Net*). Tato síť zobrazuje, jak systém reaguje (mění svůj stav) při dané události. Na vyšší úrovni je vývoj e-learningového systému modelován pomocí další Petriho sítě, nazvané metaúrovňová síť (*meta-level Petri Net*). Tato metaúrovňová síť, která je obarvenou Petriho sítí, modifikuje síť základní úrovně tím, že vybraná místa základní sítě obarví a tím její funkci pozmění. Síť základní úrovně tak naviguje studenty a pedagogy ve výukovém procesu podle svého „programu“. Tento program však může být měněn v závislosti na vývoji metasítě. Pokud základní síť pracuje v souladu s danou strategií, zůstává beze změny. V případě, že je třeba funkci pozměnit, dojde k vývoji metasítě, která prostřednictvím rozhraní obou sítí změni obarvení základní sítě ve shodě s potřebnou změnou systému.

Tento vztah obou sítí se nazývá zrcadlení. Pro rozhraní obou sítí se navrhuje užít jazyka, jehož syntaxe je inspirována jazykem T. Hoara pro popis komunikace sekvenčních procesů (viz [31]). Spolupráci základní sítě a metasítě je pak navrženo realizovat pomocí superpozice vybraných míst obou sítí.

Příspěvek lze považovat spíše za vytýčení možného směru jak formalizovat vývoj e-learningových systémů v čase na základě změny požadavků na systém a jak korektnost vývoje následně formálně verifikovat, než za ucelené řešení daného problému.

Metoda formálního popisu pomocí Petriho sítí na dvou úrovních není však v článku dokonale formálně zpracována. Nicméně pro další výzkum může být navržený směr zajímavý.

5. Návrh zobecněné Petriho sítě

5.1 Využití zobecněné Petriho sítě pro e-learning

V této kapitole, která se snaží být novým přínosem disertační práce, se pokusíme nejprve zařadit problematiku výukových systémů do širších teoretických souvislostí. Nejprve budeme definovat pojem tak zvaného inteligentního systému a ukážeme, že výukový systém lze považovat za zvláštní typ inteligentních systémů.

Inteligentní systémy pracují s informacemi. Pojem informace budeme chápat v sémantickém smyslu slova, jako soubor poznatků o vlastnostech prvků univerza. Tyto vlastnosti mohou být ostré (jisté) i neostré (nejisté). V sémantické teorii informace lze poznatky charakterizovat pomocí příslušnosti prvku univerza k určité podmnožině. U jistých (binárních) poznatků ke klasické množině, u nejistých k fuzzy množině. Informace je pak ultrafiter na množině poznatků o prvcích univerza.

Následně zobecníme pojem Petriho sítě tak, aby byla schopna pracovat s informacemi a umožňovala zpětnou vazbu při průchodu studenta výukovým systémem.

Petriho sítě je vhodné zobecnit tak, aby byly schopny zachytit více procesů, které mají vzájemnou vazbu. Z tohoto důvodu je cílem a přínosem této disertační práce zobecnění pojmu Petriho sítě tak, aby lépe popisoval zpětnou vazbu, kterou inteligentní systémy využívají. Zobecněná Petriho síť je pak použita pro modelování inteligentních systémů, speciálně e-learningových systémů zachycujících proces učení a hodnocení studentů.

Původní myšlenkou této práce je modelovat výukový proces při e-learningu pomocí Petriho sítí s využitím principu inteligentních systémů a sémantické teorie informace. Výklad základů této teorie je obsažen například v [33]. Zde se omezíme pouze na stručný přehled základních myšlenek, které bezprostředně ovlivní navržený model zobecněné Petriho sítě užitý pro modelování průběhu výukovým procesem.

5.2 Poznatky a informace

Jedním ze základních pojmů teorie poznání je **informace**. Informaci lze označit jako veličinu určující míru jistoty, že nastane jistá událost. Jinak řečeno je to míra poznání okolního světa či jakéhokoliv jiného reálného či předpokládaného systému.

Získané poznatky je vhodné zaznamenávat a vzájemně si předávat. Aby toto bylo možné, je nutné je popsat prostřednictvím zpráv, jež je možné přenášet a zpracovávat. Takovýto proces lze nazvat kódování informace. Konečně dlouhý řetězec symbolů, jež je vybrán z neprázdné, konečné a ve většině případů alespoň dvouprvkové množiny symbolů se nazývá **zpráva**. Obsahem zprávy jsou **data**.

5.2.1 Sémantická teorie informace

Matematická teorie, která popisuje informace na základě množin, se nazývá sémantická teorie informace. Základní myšlenkou této teorie je, že každý dílčí poznatek o zkoumaných entitách z nějakého univerza je možné vyjádřit jako vlastnost prvku tohoto univerza. Základní poznatek o prvku tohoto univerza je možné popsat pomocí příslušnosti daného prvku univerza do podmnožiny univerza. Do této množiny náleží pouze prvky, které mají danou vlastnost, ostatní prvky do množiny nenáleží.

Obecný poznatek o prvku množiny je pak určen funkcí, která zobrazuje univerzum do intervalu $\langle 0, 1 \rangle$, tj. tzv. fuzzy množinou, která je na univerzu definována. Hodnota funkce pak určuje míru jistoty, že prvek danou vlastnost má. Pokud je její hodnota 1, prvek danou vlastnost má, při hodnotě 0 prvek danou vlastnost jistě nemá. Hodnoty mezi 0 a 1 značí další odlišné stupně jistoty o vlastnosti prvku.

Pro účely této disertační práce je dle [33] účelné osvětlit některé vlastnosti prvků univerza U . Označme $\delta(x)$ jako základní poznatek o prvku $x \in X$, který říká, že prvek x má vlastnost δ . $\delta \subseteq U$ je množina všech prvků univerza, které mají vlastnost δ . K vyjádření toho, že poznatek je platný s mírou jistoty α , $\alpha \in \langle 0, 1 \rangle$ slouží zápis ve tvaru $(\alpha)\delta(x)$. Pak lze říci, že $(0)\delta(\alpha) = \neg\delta(x) = (U \div \delta)(x) = \delta'(x)$.

Zmiňme se dále o „ostrých“ vlastnostech prvků univerza. Pokud je $x \in U$ libovolný prvek univerza a $\delta_1(x)$ a $\delta_2(x)$ poznatky o daném prvku, platí, že jsou $(\delta_1 \cup \delta_2)(x)$, $(\delta_1 \cap \delta_2)(x)$ a $(U \div \delta_1)(x)$ také poznatky o daném prvku. Pokud platí, že $\delta_2(x) \subseteq \delta_1(x)$, pak lze nazvat poznatek $\delta_1(x)$ obecnějším nebo stejně obecným jako poznatek $\delta_2(x)$. Relace \subseteq je relace slabého uspořádání na množině $\exp(U) = 2^U$ všech podmnožin univerza. K libovolným dvěma poznatkům o prvku x existuje právě jeden minimálně obecný a právě jeden maximálně obecný poznatek, jež je méně nebo stejně obecný jako oba dva tyto poznatky:

$$\begin{aligned} \sup(\delta_1(x), \delta_2(x)) &= \delta_1(x) \vee \delta_2(x) = (\delta_1 \cup \delta_2)(x), \quad \inf(\delta_1(x), \delta_2(x)) = \delta_1(x) \wedge \delta_2(x) \\ &= (\delta_1 \cap \delta_2)(x). \end{aligned}$$

Množina všech možných poznatků o daném prvku tvoří distributivní svaz. Nejobecnější „poznatek“ $\delta^* = U$ je maximálním prvkem daného svazu. O daném prvku nic nevyovídá. Jednoprvkový poznatek $\delta_* = \{x\}$ je nejméně obecný poznatek o prvku x , který daný prvek jednoznačně charakterizuje. Žádný další prvek univerza danou vlastnost δ_* již nemá.

V sémantické teorii informace je základní informace o prvku $x \in U$ definována jako neprázdna množina $J(x)$ poznatků pro kterou platí následující podmínky:

- $\delta(x) \in J(x) \Rightarrow \delta \neq \emptyset$ – každý poznatek o prvku x je vlastností alespoň jednoho prvku univerza,
- $(\delta_1(x) \in J(x) \wedge \delta_1 \subseteq \delta_2) \Rightarrow \delta_2(x) \in J(x)$ – pokud má prvek x nějakou vlastnost, má i všechny obecnější vlastnosti,
- $(\delta_1(x) \in J(x) \wedge \delta_2(x) \in J(x)) \Rightarrow \delta_1(x) \cap \delta_2(x) \in J(x)$.

Nosič informace o prvku $x \in U$ lze nazvat libovolnou množinou poznatků $J(x) \subseteq \Delta(x)$. Patří do informace o daném prvku a platí pro ni, že pro libovolný poznatek $\delta(x)$, který patří do dané informace, existuje poznatek, který patří do nosiče $\Delta(x)$ a je méně obecný jako poznatek $\delta(x)$.

Každá informace o prvku univerza je jednoznačně dána svým libovolným nosičem. Některé informace o tomto prvku mají nosič tvořený jedním prvkem, tj. prvek je nosičem určen jednoznačně. Jestliže informace o prvku má nějaký nosič s konečným počtem prvků, má také jednoprvkový nosič. Tento lze získat jako průnik poznatků v nosičích s jedním prvkem.

5.3 Inteligentní systém

Název inteligentní systém budeme užívat bez ohledu na to, zda jde o člověka, skupinu lidí, stroj, či systém tvořený lidmi či stroji. Typické jednoduché systémy tohoto typu, které jsou prvky složitějších inteligentních systémů jsou:

- **Inteligentní čidlo** – tento systém získává poznatky typu $(p)\delta(x)$ pro jednotlivé objekty (předměty či jevy) reálného světa. Přitom x je prvek (bod univerza) a δ nějaká vlastnost prvků univerza, chápaná jako podmnožina univerza U . Číslo $p \in \langle 0,1 \rangle$ udává míru jistoty, že prvek x má vlastnost δ . Poznaček $(1)\delta(x)$ někdy zapisovaný pouze jako $\delta(x)$ znamená jistotu, že prvek x má vlastnost δ . Poznaček $(0)\delta(x)$, někdy zapisovaný pouze jako $\delta'(x)$ znamená jistotu, že x vlastnost δ nemá tedy, $\delta \notin x$, čili $x \in U \div \delta$. Inteligentní čidlo může realizovat člověk, kolektiv lidí (komise), technické zařízení nebo lidé za pomoci techniky. Podstata inteligence je, že inteligentní čidlo má schopnost zjištěnou (naměřenou) hodnotu spojit s objektem, kterého se údaj týká. Teploměr sám o sobě inteligentním čidlem není. Měří-li však současně čas a své umístění a naměřenou hodnotu spojí ze sémantickým indikátorem místa a času, roli inteligentního čidla již plní.
- **Inteligentní výkonný prvek (realizátor)** – tento systém realizuje vyhledání objektu na základě poznatků ve tvaru $(p)\delta(x)$, které jsou o tomto objektu známy. Opět může jít o člověka, kolektiv, stroj či systém, složený z lidí a strojů. Příkladem je každá výroba, vyhledání knihy knihovníkem či operace pokladníka.
- **Datový sklad** – činnost tohoto inteligentního systému spočívá ve vedení souborů, kartoték a bází dat a dokumentů, představujících poznatky o objektech a zpřístupnění údajů o těchto poznacích uživatelům. Údaje ve

tvaru $(p)\delta(x)$ jsou ve skladu vždy spojeny s objektem x , kterého se týkají. Výstup ze skladu je řízen vnějšími požadavky.

- **Transformátor poznatku (Ultrafunkce)** – transformátor je systém, který na základě poznatků o objektech množiny X vytváří poznatky o objektech množiny Y . Transformátorem může být člověk, který na základě daných poznatků přijímá nějaké rozhodnutí, odborník či kolektiv, řešící nějaký problém, projektant nebo projekční kancelář, řidič automobilu, pilot letadla nebo technické zařízení regulující výrobní proces i každý počítačový program, provádějící nějaký algoritmus. Transformátor převádí poznatky typu $(p)\delta(x)$ pro $x \in X$ na poznatky typu $(q)\beta(y)$ pro $y \in Y$. Zvláštní, ale častý případ je $X = Y$. V tom případě lze transformátor nazvat překladač.

Z uvedených čtyř základních typů inteligentních systémů lze vytvářet složitější, například:

- **Expertní systém** – je síť základních inteligentních systémů složená z čidel, datových skladů a transformátorů. Tato síť může být víceúrovňová, hierarchická. Na nejnižší úrovni nemá výkonné prvky (realizátory).
- **Řídicí systém** – jako síť inteligentních systémů, která obsahuje vždy čidla a výkonné prvky.

Každý inteligentní systém musí mít vymezen předmětnou oblast, která není jeho součástí, ale se kterou je činnost systému svázána. Tato oblast je tvořena objekty daného typu a vztahy mezi nimi. Tvoří univerzum, pro které jsou definovány poznatky a informace.

5.4 Zobecněná Petriho síť

Informační oblast intelektuálních systémů lze modelovat jako síť základních prvků a datových skladů. Pro modelování takové sítě formálními prostředky bude vhodné zobecnit pojem Petriho sítě tak, aby bylo možné zobrazit jak proces dotazů tak zároveň i vyhledávání odpovědí.

Inteligentní systém pracuje tak, že nejprve analyzuje požadavky a pak se je snaží zodpovědět. To lze popsat změnou stavů zobecněné Petriho sítě takto:

- Dotazy (požadavky) se znázorňují značkami *. Odpovědi se znázorňují značkami •.
- Znak * se do sítě vloží z nějakého datového skladu nebo od výkonného prvku nebo z vnějšku tak, že představuje odkaz na nějakou vlastnost nějakého prvku univerza. V tom případě se do daného místa zobecněné Petriho sítě vloží *. V místě, označeném * se vytvoří odpověď. Je-li v tomto místě dostatečná informace pro odpověď (tj. informace pokrývající požadavek, změní se * na •). K této značce • přísluší odpověď typu $(p)\delta(x)$, vyjádřená nejlépe daty $\Delta(x) = \{(p_1)\delta_1, (p_2)\delta_2, \dots, (p_n)\delta_n\}(x)$ – těmito daty jsou jednotlivé značky typu • “obarveny“.
- V případě, že v daném místě (datovém skladu) informace pokrývající dotaz není nebo není dostatečná, znak * v tomto bodě zůstane a oproti směru hran v zobecněné Petriho síti vznikne znak * ve všech vstupních přechodech tohoto místa.

Dotazy lze vyřizovat podle zvolené strategie, buď postupně nebo paralelně. Jsou-li v příslušném místě potřebné informace, změní se * na •. Pokud ne, pokračuje se rekurzivně, dokud se na některém místě neobjeví •.

Totéž lze využít pro řízení toho, kterou část látky má student znovu studovat a procvičovat pro to, aby měl předpoklady zvládnout další poznatky či získat potřebné návyky.

Hlavním přínosem této disertační práce je návrh takového zobecnění, které by odpovídalo potřebám modelování automatizovaného výukového systému. Zobecnění Petriho sítě pro tento účel spočívá v doplnění dalšího typu značky, kterou budeme v grafických reprezentacích označovat symbolem *, kterou mohou být podle přesně definovaných pravidel označeny některá místa v síti. Prioritu mají nadále v síti značky typu •, jejichž výskyt se i u zobecněné Petriho sítě řídí běžnými pravidly pro klasické Petriho sítě, viz kapitola 4.2.4.1 této práce.

Zavedení nového typu značky *, která se na rozdíl od značek • může v každém místě vyskytnout pouze jediná a nemůže v žádném místě být současně se značkou •, je vynuceno tím, že zobecněné Petriho sítě mají být nástrojem pro sledování dvou různých procesů současně – procesu dotazu a procesu hledání odpovědi na dotaz. Toto zobecnění Petriho sítí autorka publikovala jako svůj původní výsledek v [15a].

Definice č. 1: Zobecněnou Petriho síť nazýváme bipartitní orientovaný multigraf $Q = (P, T, H, \varphi)$, kde P je konečná neprázdná množina vrcholů (uzlů) grafu, nazývaných místa, T je konečná neprázdná množina vrcholů (uzlů) grafu, nazývaných přechody, $P \cap T = \emptyset$ a $P \cup T$ je množina všech uzlů orientovaného multigrafu, H je množina jeho orientovaných hran a φ je zobrazení H do $(P \times T) \cup (T \times P)$. Každá orientovaná hrana tedy vede buď z místa do přechodu nebo z přechodu do místa.

Bipartitní orientovaný multigraf $Q = (P, T, H, \varphi)$ je chápán jako Petriho síť se všemi jejími náležitostmi, především s možností kapacitních omezení míst sítě a váhami hran sítě, tak jak bylo uvedeno v [10] a popisovaného v předchozím kapitolách disertační práce. Rozdíl je pouze v tom, že zobecněná síť připouští různé typy značek a má doplňující pravidla pro vývoj sítě.

Uzly grafu jsou obarveny dvěma typy značek podle pravidel, která budou popsána v následujících definicích. Pracuje v diskrétních krocích, při kterých se mění obarvení míst podle pravidel, která budou také popsána v následujících definicích.

Definice č.2: Značkou prvního typu (dotazem), kterou budeme označovat symbolem * v zobecněné Petriho síti $N = (P, T, H, \varphi)$ se rozumí zobrazení χ množiny $P \cup T$ do množiny $\{0,1\}$ (v každém vrcholu může být nejvýše jedna značka *). Rozmístění značek * zapisujeme jako $(m+n)$ -místný vektor

$$\chi = (\chi_1, \chi_2, \dots, \chi_m, \chi^1, \chi^2, \dots, \chi^n), \text{ kde}$$

- $\chi_i = \chi(p_i)$ pro $i = 1, \dots, m$ a
- $\chi_j = \chi(t_j)$ pro $j = 1, \dots, n$.

Je-li $\chi(p_i) = 1$, je v místě p_i hvězdička, je-li $\chi(p_i) = 0$, hvězdička v místě p_i není. Obdobná pravidla jsou platná i pro přechody.

Definice č. 3: Krokem prvního typu pro zobecněnou Petriho síť nazýváme proces změny označení sítě spočívající ve změně zobrazení χ , tedy ve změně rozmístění značek prvního typu podle těchto pravidel:

- Jestliže přechod $t \in T$ nemá označení * a alespoň jedno z jeho následných výstupních míst má nenulovou hodnotu $\chi(p)$, tedy je označeno *, potom dochází ke změně, v jejímž důsledku přechod získá označení * a všechna jeho vstupní místa získají též znak * (ohodnocení χ bude rovno 1). Ohodnocení výstupních míst tohoto přechodu se nezmění. Pokud počáteční podmínce tohoto odstavce vyhovuje více přechodů, popsaná změna označení se provede u všech z nich v libovolném pořadí či paralelně. Tento proces probíhá tak dlouho, pokud podmínce vyhovuje alespoň jeden přechod.
- Když již podmínka pro změnu označení * není u žádného přechodu splněna, získáme tak podsít' $Q^* \subseteq Q$ s místy $P^* \subseteq P$ a přechody $T^* \subseteq T$ ohodnocenými znakem * (hodnota zobrazení χ bude u nich rovna 1) a příslušnými orientovanými hranami $H^* \subseteq H$, které jsou s vybranými místy a přechody incidentní. Zúžené zobrazení $\varphi: H \rightarrow (P \times T) \cup (T \times U)$ na $P^* \cup T^*$ označme φ^* .

Definice č. 4: Značkou druhého typu (odpovědí), kterou budeme označovat symbolem \bullet , v zobecněné Petriho síti $N = (P, T, H, \varphi)$ se rozumí zobrazení (funkce) μ množiny P^* všech míst podsítě $(P^*, T^*, H^*, \varphi^*)$ do množiny \mathbf{N} všech nezáporných celých čísel. Hodnota této funkce má význam počtu značek umístěných v příslušném místě podsítě Q^* z uzlů označených značkami prvního typu (*). Počet značek \bullet v místě p^*_j podsítě Q^* je dán hodnotou funkce $\mu(p^*_j)$.

Definice č. 5: Krokem druhého typu pro zobecněnou Petriho síť $G = (P^*, T^*, H^*, \varphi^*)$ se nazývá proces změny jejího značení podle pravidel klasické Petriho sítě, tedy s přihlédnutím ke kapacitnímu omezení míst a váhou hran v původní

síti G . Přitom po provedení libovolného přípustného přechodu $t^* \in T^*$ se u tohoto přechodu značka $*$ umaže (hodnota $\chi(t)$ se změní na hodnotu 0). Po provedení tohoto kroku se zruší značky $*$ na vstupních místech, na kterých se objevil znak \bullet . Platí tedy $\mu(p) \neq 0 \Rightarrow \chi(p) = 0$. Pokud se na nějakém místě objeví znak \bullet , ruší se tím znak $*$.

Definice č. 6: V zobecněné Petriho síti $G = (P, T, H, \varphi)$ probíhají kroky prvního a druhého typu současně, přičemž platí, že značka \bullet druhého typu je „silnější“ než značka $*$ prvního typu. To přesně znamená následující:

- Pokud se během vývoje sítě v nějakém místě objeví znak \bullet , ruší se tím znak $*$ na tomto místě.
- Po provedení kroku druhého typu (změně značení \bullet) se ruší znaky $*$ ve všech vstupních místech provedeného přechodu a současně i u tohoto přechodu samotného, pokud tento přechod nemá nějaké další vstupní místo označené znakem $*$.

5.4.1 Vývoj zobecněné Petriho sítě

Vývoj zobecněné Petriho sítě probíhá následujícím způsobem. Na počátku je zadána Petriho síť s kapacitami míst a váhami hran, případně s dalšími omezeními. Tato síť představuje celkový rámec problémové oblasti. V případě automatizované výuky celý kurz. Vstupním místem sítě je takové místo, do kterého nevede žádná hrana. Konkrétní úkol je zadán vymezením nějaké podsítě tak, že příslušným uzlům (místům i přechodům) přiřadíme znak $*$, tedy definujeme počáteční značení prvního typu $(\chi_1^0, \chi_2^0, \dots, \chi_m^0, \chi_0^1, \chi_0^2, \dots, \chi_0^n)$, kde $\chi_j^0 \in \{0, 1\}$, $\chi_0^i \in \{0, 1\}$. Celý dílčí krok pak probíhá pouze v takto vymezené podsíti původní zobecněné Petriho sítě.

Obecné použití tohoto modelu pro popis a modelování funkce inteligentních systémů může probíhat takto:

1. Každému nosiči informace (datovému skladu) přiřadíme nějaké místo v zobecněné Petriho síti, zobrazujeme jej kroužkem.
2. Každému modelovanému systému zpracování informace přiřadíme nějaký přechod v zobecněné Petriho síti (zobrazujeme jej obdélníkem).

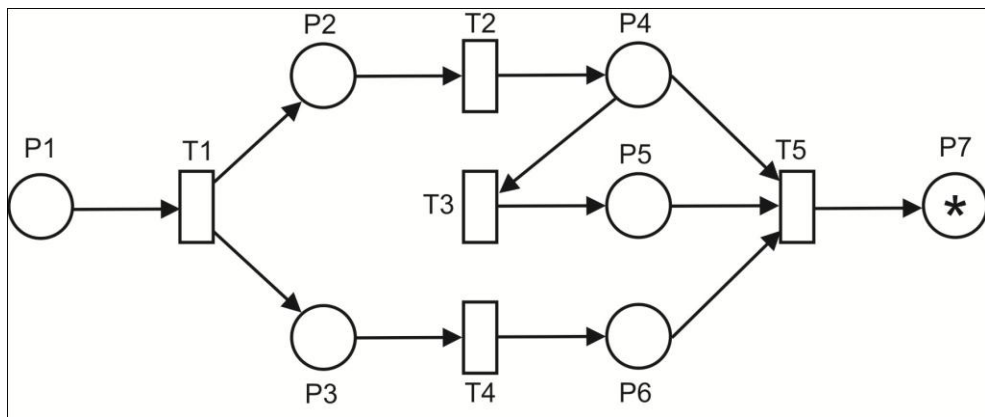
3. Informační spojení systémů a nosičů informace zobrazíme orientovanými hranami.
4. Pro značení dotazů užíváme znak *, pro značení odpovědí znak •. Znaky * užíváme jak pro místa (nosiče informace), tak pro přechody (systémy zpracování informací). Znaky • užíváme jen pro místa.

Automatizovaný výukový proces lze považovat za optimální typ inteligentního systému. Zvládnutí látky lze formulovat jako požadavek na dodání informace ze strany studenta. Tento požadavek se projeví jako značení prvního typu (znak *) umístěné do příslušného místa zobecněné Petriho sítě. V případě, že test je úspěšný a prokazuje, že student látku do požadovaného rozsahu a hloubky zvládl (pochopil a umí získané znalosti a dovednosti použít), změní se znak * na znak • a výuka může podle pravidel klasické Petriho sítě pokračovat.

V případě, že výsledek testu vyhovující není, znak * se v daném místě ponechá a dotaz se proti směru orientace hran grafu rozšíří na vstupní přechody daného místa a ke všem místům, která tomuto přechodu bezprostředně předcházejí. To znamená, že student se musí ve výukovém schématu vrátit k některé látce, která je předpokladem ke zvládnutí tématu, u kterého test nebyl úspěšný a studovat či trénovat jej znovu. Znaky * se tedy rozšíří do předchozího přechodu a míst v síti.

V případě neúspěchu i zde se test rekurzivně opakuje tak dlouho, dokud nelze nahradit znak * znakem druhého typu •, tedy úspěšnou odpovědí tak, aby mohl pokračovat vývoj klasické Petriho sítě.

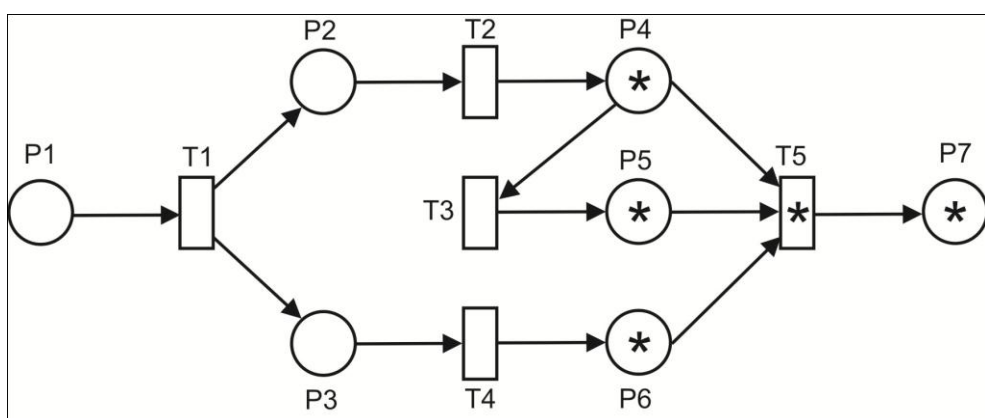
Pro objasnění procesu práce sítě uveďme nejprve obecný příklad sítě složené ze sedmi vzájemně provázaných témat, vyjádřené následujícím grafem vzájemné závislosti. Předpokládejme, že má být otestována znalost celku reprezentovaná místem P7 na které je směřován dotaz, viz obrázek číslo 18.



Obr. 18 – Příklad sítě složené ze sedmi vzájemně provázaných témat

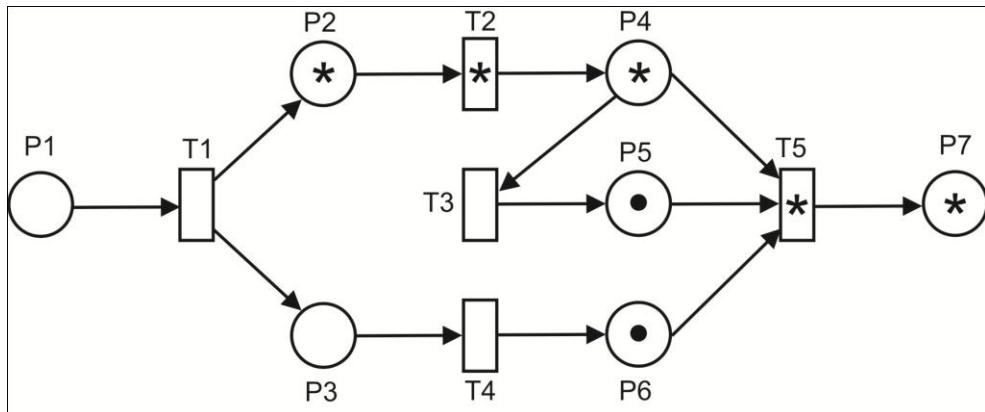
Připusťme, že test P7 nebyl úspěšný. Pokud by úspěšný byl, bylo by možné obarvit místo P7 značkou druhého typu • a výukový proces by se mohl dále řídit pravidly klasické Petriho sítě, tj. studiem následující látky, pro níž je znalost P7 předpokladem.

Neúspěšný výsledek znamená, že je nutné prověřit znalost předchozích témat, která jsou pro pochopení předpokladem. To se projeví krokem prvního typu ve zobecněné Petriho síti, po jehož provedení bude podle pravidel zavedených v této disertační práci stav obarvení sítě následující, viz obrázek číslo 19.



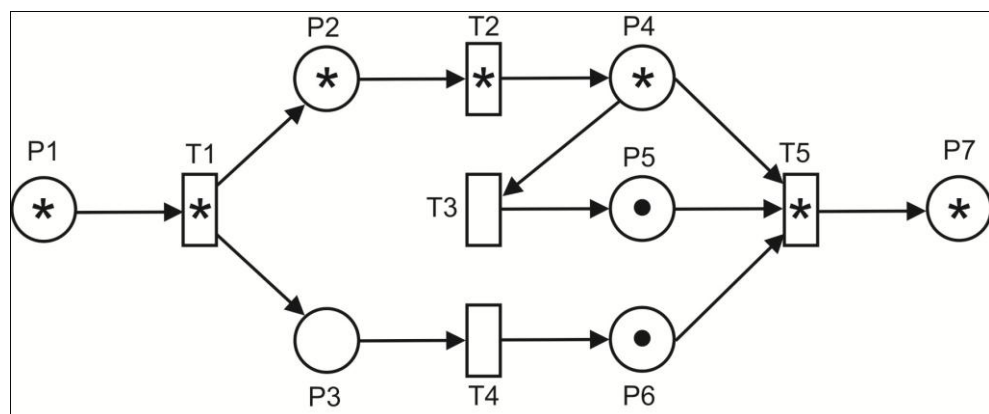
Obr. 19 – Příklad kroku prvního typu ve zobecněné Petriho síti

Předpokládáme například, že testy P5 a P6 proběhnou úspěšně, že při testu P4 student selže. Nyní lez nahradit značky * prvního typu v místech P5 a P6 značkami druhého typu •, avšak značku * u místa P4 je nutno rozšířit na přechod T2 a jeho předchozí místo P2, protože je třeba, aby si student tuto látku zopakoval. Po provedení tohoto kroku bude mít zobecněná Petriho síť obarvení uvedené v obrázku číslo 20.



Obr. 20 – Obarvení zobecněné Petriho sítě při neúspěchu studenta při testu P4

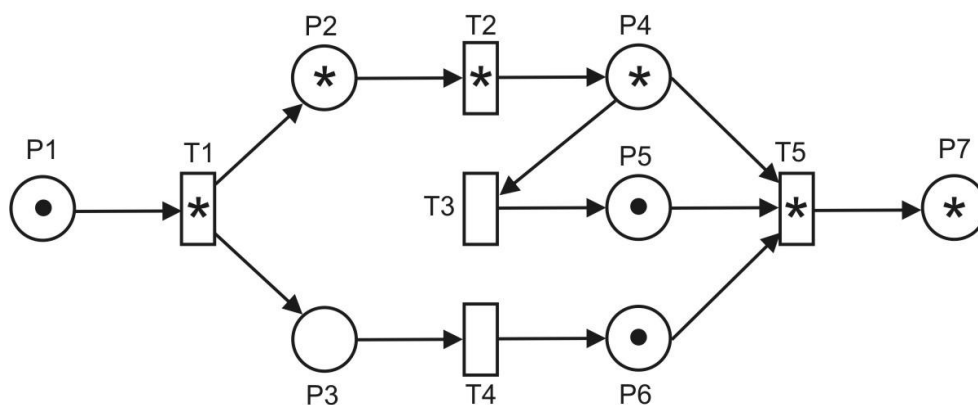
Je-li test P2 opět neúspěšný, je nutné, aby se student vrátil až k výchozímu bodu P1 a látku si zde zopakoval a podrobil se testu, viz obrázek číslo 21.



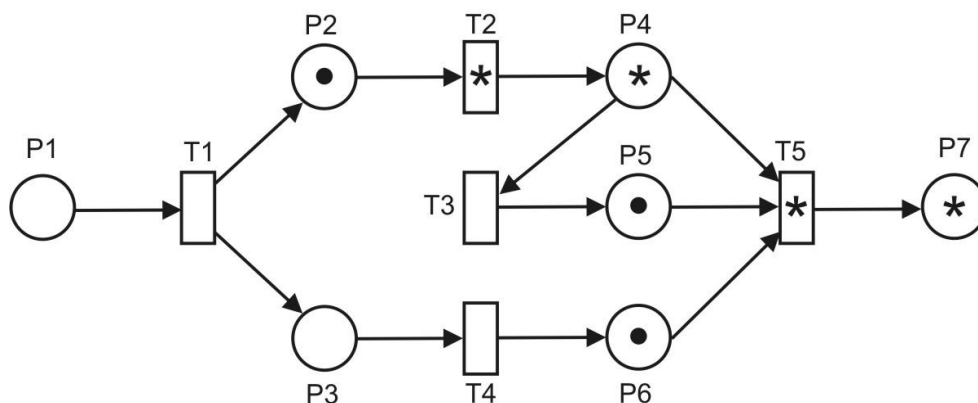
Obr. 21 – Obarvení zobecněné Petriho sítě při neúspěchu studenta při testu P2

Předpokládejme, že nyní již po zopakování výchozí látky student úspěšný bude a v dalším postupu testování celku se nebude nutné již znovu vracet k předchozí látce.

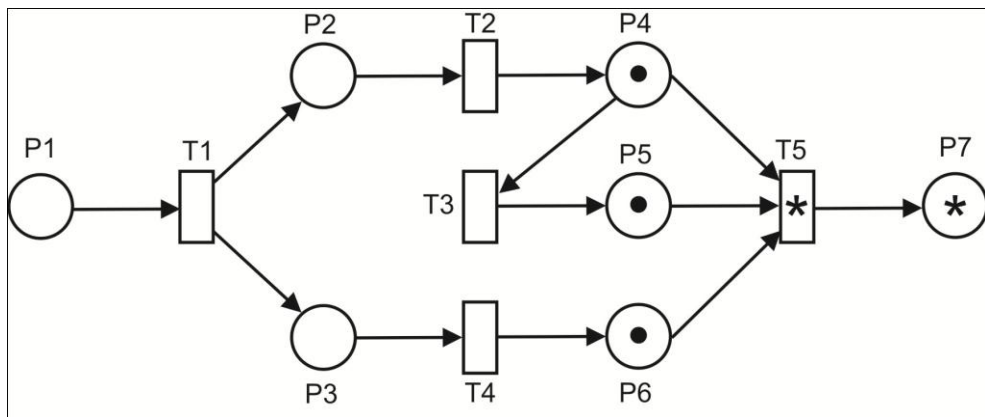
Při vývoji zobecněné Petriho sítě, která proces modeluje, budou postupně nahrazovány značky prvního typu (*) u míst značkami druhého typu (●) a odstraňovány značky prvního typu (*) u přechodů. Postupný vývoj značení ve zobecněné Petriho síti je zobrazen na obrázcích 22 – 25.



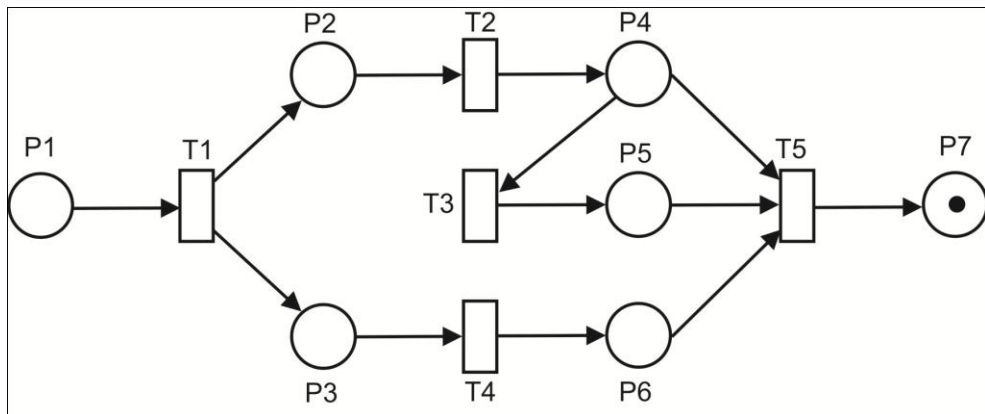
Obr. 22 – Obarvení zobecněné Petriho sítě při úspěchu studenta při testu P1



Obr. 23 – Obarvení zobecněné Petriho sítě při úspěchu studenta při testu P2



Obr. 24 – Obarvení zobecněné Petriho sítě při úspěchu studenta při testu P4



Obr. 25 – Obarvení zobecněné Petriho sítě při úspěšném průchodu studenta kurzem

Stojí za povšimnutí, že navržený způsob sledování zpětné vazby v automatizovaných výukových systémech umožňuje snadné zobrazení nejrůznějších vzájemných souvislostí vyučované látky formou orientovaného grafu zobrazujícího, které znalosti a dovednosti jsou předpokladem pro zvládnutí daného tématu.

Je velmi vhodný i pro postupné zjemňování návrhu výukových systémů metodou shora-dolů (top-down). Každý přechod je totiž možno považovat za systém nižší úrovně a v následném kroku zjemňování návrhu je možné jej nahradit vloženou zobecněnou Petriho sítí, která vyjádří detailnější pohled na výukový systém.

Zobecněné Petriho sítě jsou vhodné i pro sběr informací v postupu studenta. Jednotlivé kroky vedoucí k průchodu sítí je možné monitorovat. Je-li to žádoucí, může o nich být informován jeho pedagog a mohou být využity k odhadu studijního typu konkrétních studentů k identifikaci jejich silných stránek a slabin, například zda jim činí větší potíže teorie a potřeba abstrakce či aplikace poznatku nebo rutinní provádění jednotlivých aplikačních úkolů.

5.5 Příklady užití zobecněné Petriho sítě na konkrétní výuková témata

Tato kapitola disertační práce je zaměřena na praktické ukázky příkladů na využití v práci navrženého modelu zobecněné Petriho sítě pro konkrétní výukové tématické celky.

Vzhledem k tomu, že se autorka disertační práce podílí na Katedře informačního inženýrství PEF ČZU v Praze na výuce předmětů **Výpočetní systémy** a **Architektura počítačů**, budou uvedené příklady znázorňovat průchody studenta tématickými celky obsaženými v těchto předmětech.

Vývoj zobecněné Petriho sítě bude osvětlen na modelech průchodu studenta dvěma tématickými celky. První příklad je zaměřen na výuku problematiky konstrukce **kombinačních logických obvodů pomocí tranzistorů**, tento tématický celek je důležitou součástí předmětu Architektura počítačů. Pomocí druhého příkladu je zobrazena výuka způsobu **uložení reálných čísel v počítači**, konkrétně pomocí formátu IEEE 754 pro čísla v pohyblivé řádové čárce. Tato látka je vyučována v předmětu Výpočetní systémy.

5.5.1 Příklad konstrukce kombinačního logického obvodu pomocí tranzistorů

Prvním příkladem, který demonstruje vývoj zobecněné Petriho sítě použité pro modelování průchodu studenta výukovým procesem je případ testující dovednost studenta zkonstruovat logický obvod pomocí tranzistorů. Síť je složena z patnácti

následujících vzájemně provázaných témat (reprezentovaných jednotlivými místy zobecněné Petriho sítě):

- **P1 – logické hodnoty, výroková logika**, – představuje znalosti logických hodnot a dále pravdivostních tabulek nejpoužívanějších logických spojek (negace, logický součin, logický součet, nonekvivalence, ekvivalence, Piercova funkce, Shefferova funkce, implikace).
- **P2 – výrokový počet a Booleova algebra** – představuje znalosti pojmu výrok – jednoduchý a složený (formule výrokového počtu), tautologie a kontradikce výrokového počtu, ekvivalence formulí, vztah výrokového počtu a Booleovy algebry a dále schopnost vyhodnocování logických formulí.
- **P3 – převod výrokové formule na tautologicky ekvivalentní formuli v disjunktivní a konjunktivní normální formě** – pokrývá znalosti definice a konstrukce disjunktivní a konjunktivní normální formy (DNF a KNF) a znalosti převodu výrokové formule do DNF a KNF pomocí ekvivalentních úprav, pravdivostní tabulky či pomocí Karnoughových map.
- **P4 – logické obvody** – představuje znalost pojmu logický obvod a rozdělení logických obvodů na základní typy dle závislosti vstupních hodnot obvodu na kombinační a sekvenční logické obvody.
- **P5 – kombinační logické obvody** – představuje znalost pojmu kombinační logický obvod, znalosti nejpoužívanějších typů kombinačních obvodů (kodér, dekodér, multiplexor, demultiplexor), jejich popisu pomocí booleovských funkcí a pravdivostních tabulek a dále využití kombinačních logických obvodů.
- **P6 – hradla a jejich symbolické značky** – představuje znalost symbolických značek základních nejpoužívanějších kombinačních obvodů – hradel (NON, OR, NOR, AND, NAND, XOR, EQV).

- **P7** – *analýza kombinačního logického obvodu* – představuje znalost pojmu a postupu analýzy kombinačního logického obvodu a sestavení booleovské funkce popisující daný obvod.
- **P8** – *syntéza kombinačního logického obvodu* – představuje znalost pojmu a postupu syntézy kombinačního logického obvodu z hradel:
 - AND, OR, NON.
 - pouze NAND (Piercova funkce),
 - pouze NOR (Shefferova funkce).
- **P9** – *technická realizace logických hodnot pomocí úrovně napětí* – zahrnuje znalost reprezentace logických hodnota 0 a 1 v elektronických logických obvodech, tj. obvykle reprezentací pomocí velikosti (úrovně) elektrického napětí (U) v pozitivní i negativní logice. Pokud v elektronickém logickém obvodu elektrické napětí nabývá hodnoty z intervalu $\langle U_{min}, U_{max} \rangle$, platí:
 - $U \in \langle U_{min}, U_L \rangle$, napětí U je ve stavu L (*low*),
 - $U \in (U_L, U_H)$, napětí U je v nedefinovaném stavu,
 - $U \in \langle U_H, U_{max} \rangle$, napětí U je ve stavu H (*high*).V **pozitivní logice** hodnota napětí U na úrovni H znamená logický stav 1, je-li napětí U na úrovni L, znamená to logický stav 0. V **negativní logice** nízké úrovni U odpovídá logická hodnota 1 a vysoké úrovni U odpovídá logická hodnota 0.
- **P10** – *základní fyzikální pojmy (pojmy napětí, proud, odpor, apod.)* – prostudování těchto základních pojmů je nezbytné pro pochopení souvislostí při výuce dalších tematických celků, zejména učiva týkajícího se rozdělení látek dle odporu a typu polovodičů.
- **P11** – *rozdělení látek dle odporu a typy polovodičů* – zahrnuje znalost rozdělení látek dle měrného fyzikálního odporu, tj. na nevodiče, polovodiče

a vodiče. Dále jsou předpokládány znalosti typů polovodičů využívaných v elektronickém průmyslu:

- *polovodiče typu N* – které vznikají dotací krystalu křemíku pětímocným prvkem (např. arsen nebo fosfor),
 - *polovodiče typu P* – které vzniknou dotací krystalu křemíku třímocným prvkem (galiem či indiem).
- **P12 – typy polovodičových prvků** – předpokládá znalosti konstrukce základních polovodičových prvků, které vznikají složením polovodičů typu P a N a jejich použití:
- *dioda* – která vzniká složením dvou příměsových polovodičů – typu N (katoda) a typu P (anoda),
 - *tranzistor* – součástka, kterou tvoří dvojice přechodů P a N.
- **P13 – tranzistor a jeho funkce** – předpokládá znalost základních skupin tranzistorů, tj.:
- *bipolární tranzistory (tranzistory řízené proudem)* – které jsou ovládány elektrickým proudem mezi bází a emitorem, velikostí tohoto proudu se ovládá proud v obvodu, procházející mezi emitorem a kolektorem. Dle uspořádání oblastí polovodičů P a N rozdělujeme na NPN a PNP tranzistory.
 - *unipolární tranzistory (tranzistory řízené elektrickým polem)* – využívají k řízení proudu mezi D (*drain*) a S (*source*) elektrostatické pole, vytvořené v obvodu řídicí elektrody G (*gate*).

Poznámka: pro úspěšné absolvování učiva zaměřeného na návrh kombinačního obvodu pomocí tranzistorů je důležitá zejména znalost principů a funkce bipolárních tranzistorů.

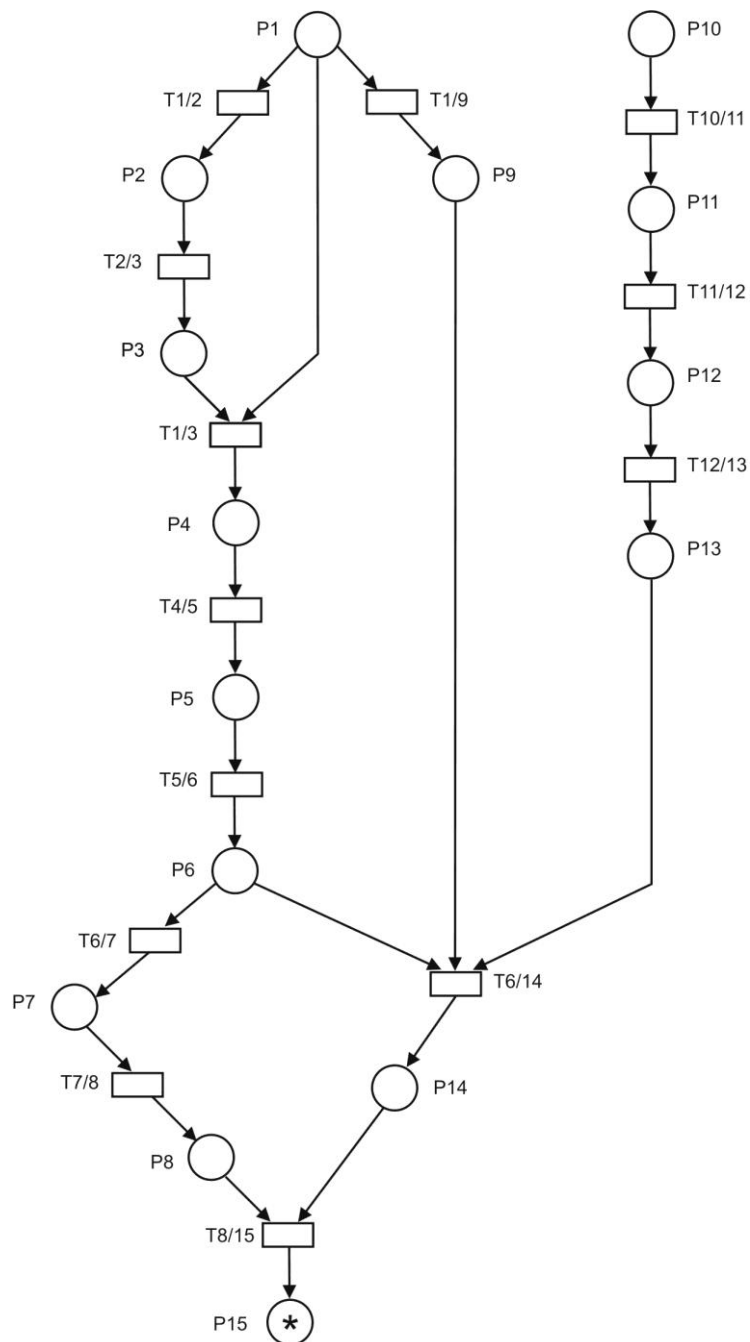
- **P14 – realizace hradel tranzistorů** – zahrnuje znalost realizace jednotlivých hradel (AND, OR, NON) pomocí bipolárních tranzistorů v negativní i pozitivní logice.

- **P15** – *návrh kombinačního obvodu pomocí tranzistorů* – zahrnuje schopnost realizace celého kombinačního logického obvodu pomocí tranzistorů.

Podrobnější výklad problematiky výrokové logiky, Booleovy algebry a kombinačních logických obvodů je k dispozici v [19a] a [21a].

V tomto příkladu má být otestována znalost celku P15, tedy skutečnosti, zda je student na základě znalostí tématických celků P1 až P14 schopen *navrhnout kombinační logický obvod z polovodičových prvků*, konkrétně pomocí tranzistorů. Na místo P15 je proto směřován dotaz. Zobecněná Petriho síť popisující tento příklad je znázorněna na obrázku 26.

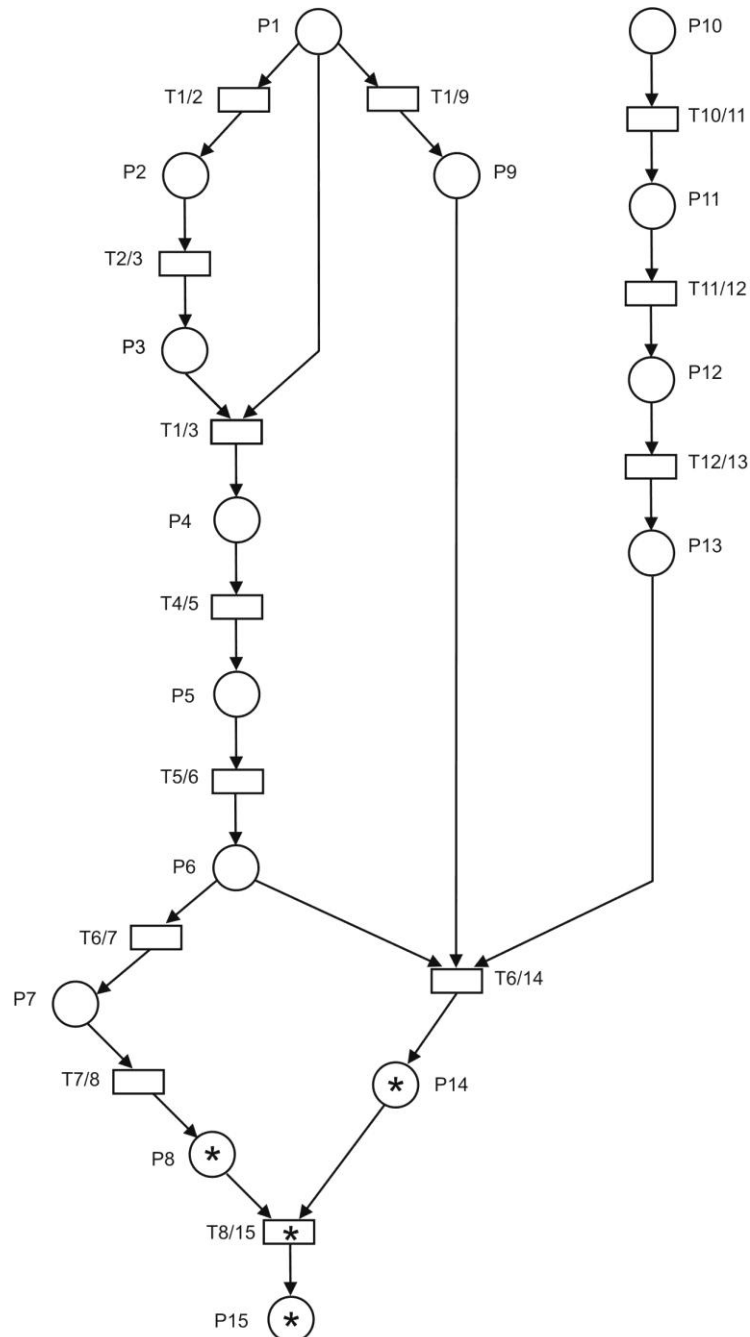
Poznámka: pro zpřehlednění příkladu jsou jednotlivé přechody navržené zobecněné Petriho sítě označeny pořadovým číslem vstupního a výstupního místa. U přechodů s více výstupními místy je použito označení pomocí vstupního místa s nejnižším pořadovým číslem a číslem výstupního místa. Pokud by v modelu existoval přechod s více výstupními místy, bylo by k označení přechodu použito číslo vstupního místa a výstupního místa s nejnižším pořadovým číslem.



Obr. 26 – Síť zobrazující příklad konstrukce kombinačního logického obvodu pomocí tranzistorů

Pokud by student test v místě P15 absolvoval úspěšně, toto místo je možné obarvit značkou druhého typu (●). Výuka by za těchto okolností pokračovala směrem k dalším tématům, které na téma P15 navazují. To znamená, že síť by se vyvíjela podle pravidel klasické Petriho sítě. Ukažme ale případ, ve kterém student test v místě P15 úspěšně neabsolvoval. V tom případě je třeba otestovat znalost témat, která tématu P15

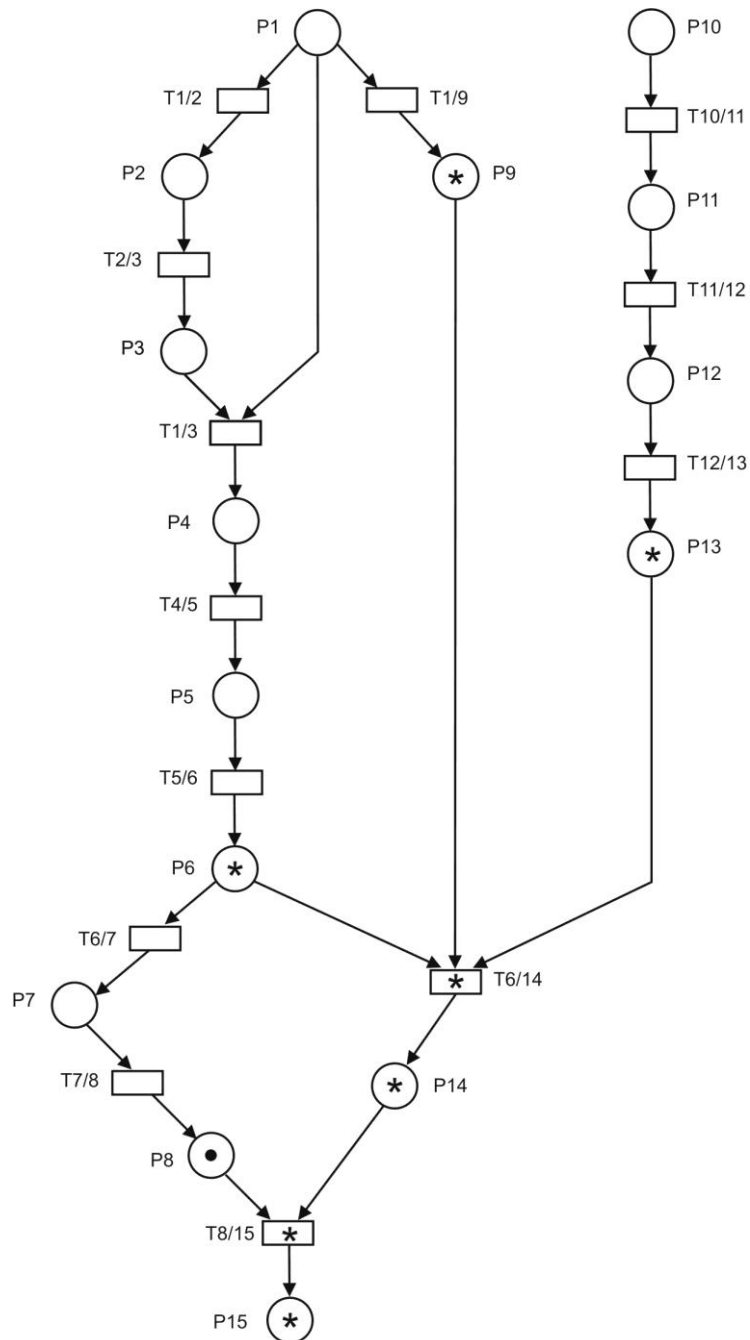
předchází, tj. P8 (*syntéza kombinačního logického obvodu*) a P14 (*realizace hradel tranzistory*). To se v síti projeví krokem prvního typu, viz obrázek 27.



Obr. 27 – Síť zobrazující neúspěch při testu v místě P15

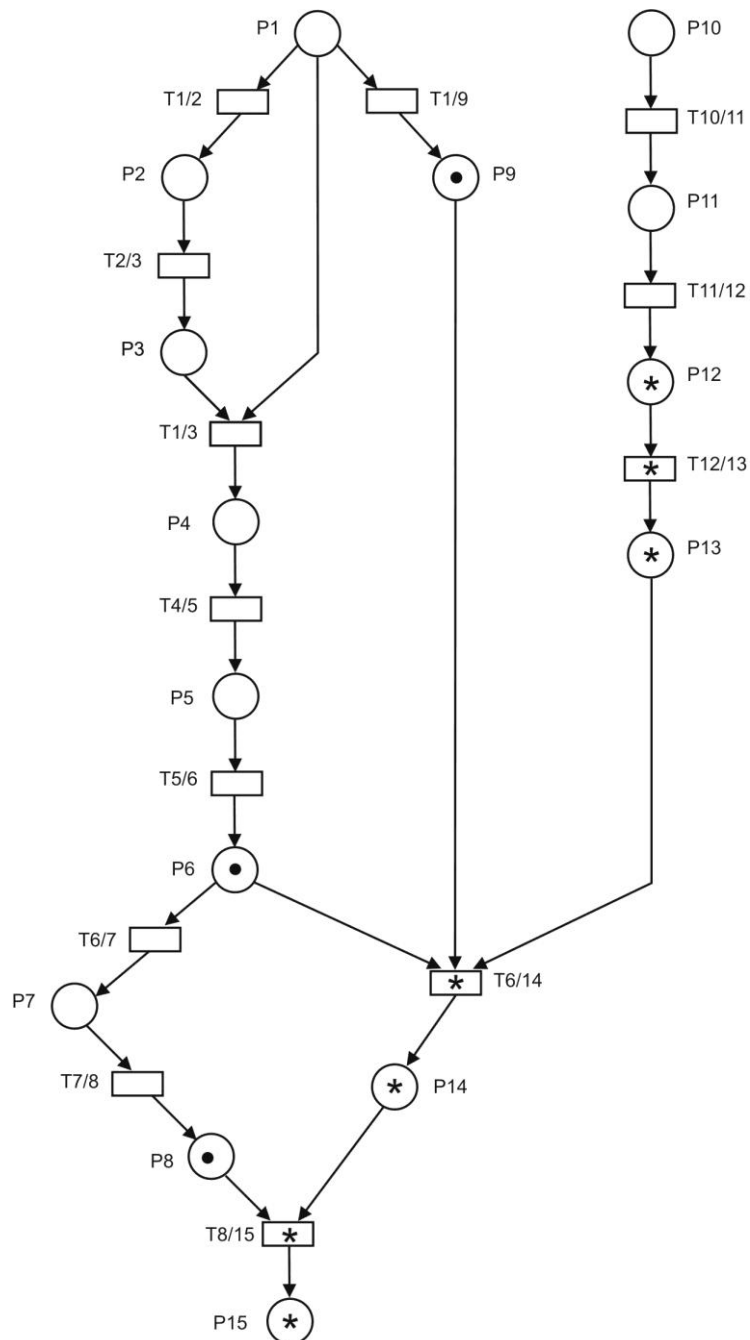
V následujícím kroku předpokládáme, že student ovládl problematiku *syntézy kombinačního logického obvodu*, reprezentovanou místem P8. Proto byla v místě P8

značka prvního typu (*) nahrazena značkou druhého typu (●). Student ale neovládl problematiku *realizace hradel tranzistorů*, reprezentovanou místem P14, proto je nutné otestovat téma předcházející, tj. *technická realizace logických hodnot pomocí úrovně napětí a tranzistor a jeho funkce*. Z tohoto důvodu je nutné rozšířit značku * na přechod T6/14 a jemu předcházející místa P9 a P13, viz obrázek 28.



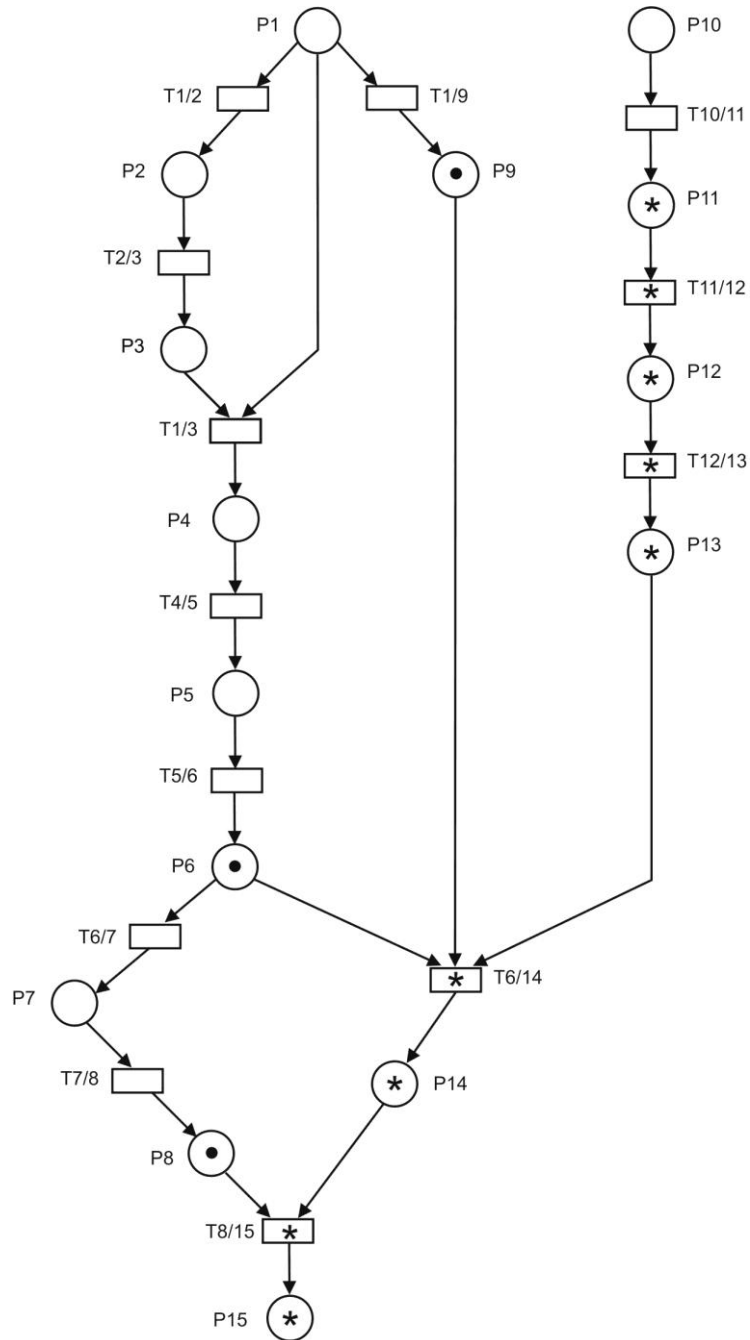
Obr. 28 – Síť zobrazující neúspěch studenta při testu v místě P14

Pokud ani tentokrát nebyl student v testu v místě P13 úspěšný a síť jej tudíž bude navigovat k prostudování tématu předcházejícímu, tj. *typy polovodičových prvků*, reprezentovaným místem P12. Proto se bude značka * šířit na přechod T12/3 a jemu předcházející místo P12. Naopak tematiku technické *realizace logických hodnot pomocí úrovně napětí* již student nastudoval a podmínky testu splnil a nemusí se proto vracet k předchozí látce. Proto je v místě P9 značka * nahrazena značkou ●, obrázek 29.



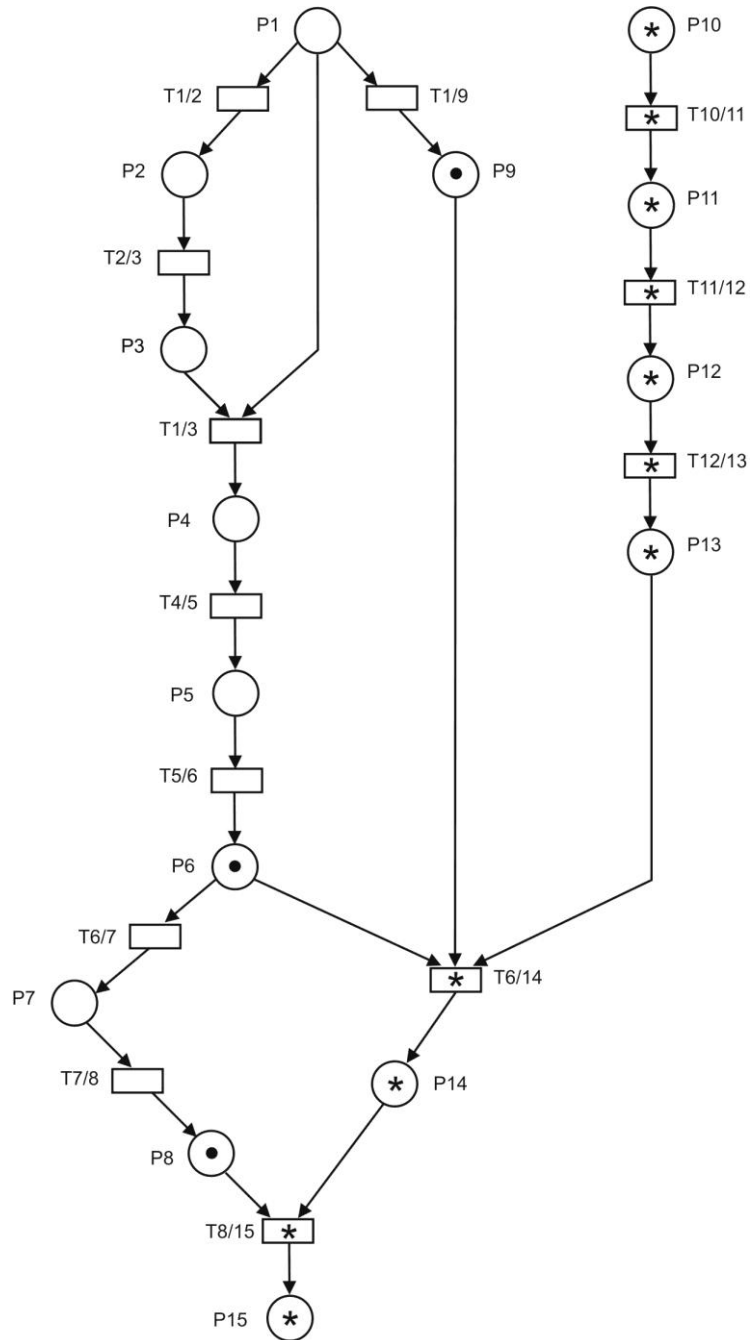
Obr. 29 – Síť zobrazující neúspěch studenta při testu v místě P13

Student byl však opět neúspěšný při testování tematiky *typy polovodičových prvků*, reprezentované místem P12. Síť jej tedy opět naviguje k prostudování předcházejícího tématu, tj. *rozdělení látek dle odporu a typy polovodičů* (místo P11). Značka * se proto rozšíří přes přechod T11/12 do místa P11, viz obrázek 30.



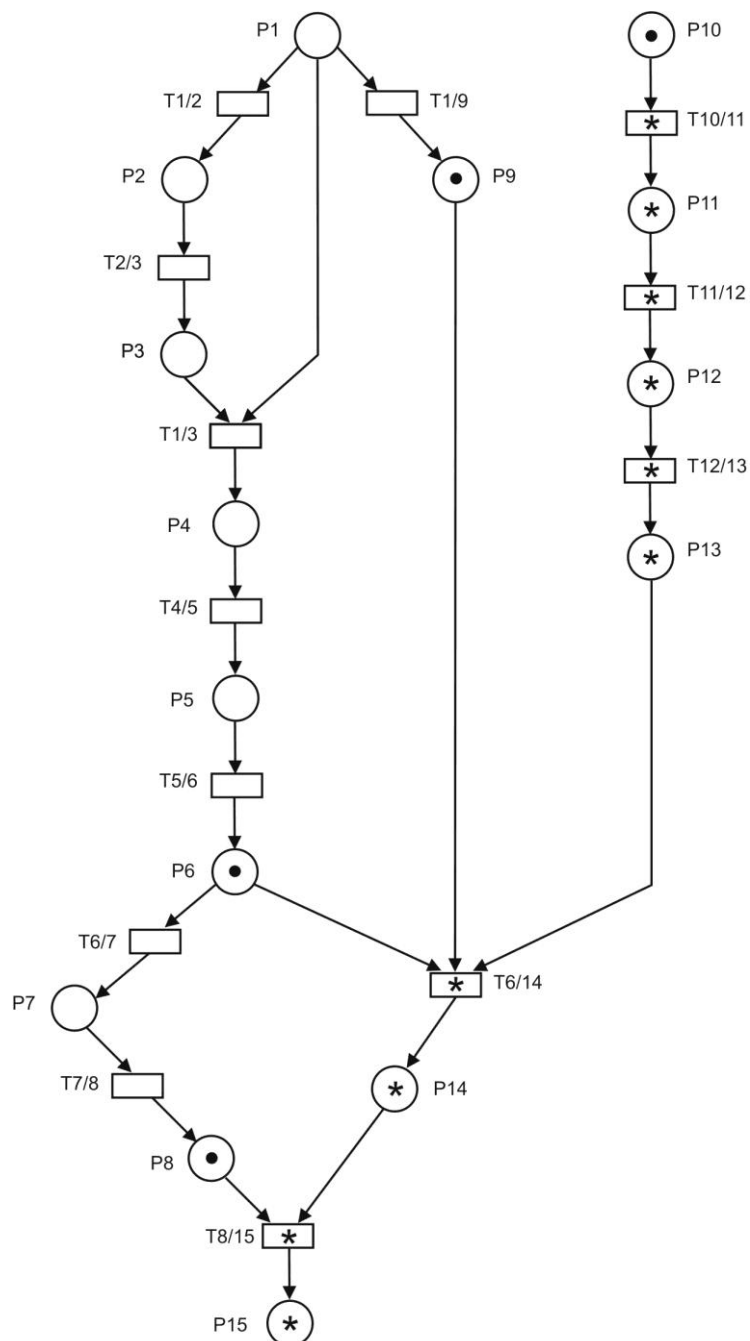
Obr. 30 – Síť zobrazující neúspěch studenta při testu v místě P12

Student ani v tomto případě nebyl úspěšný a podmínky testu pro látku *rozdělení látek dle odporu a typy polovodičů* (místo P11) nesplnil. Je tedy nutné, aby nastudoval ještě téma předcházející, tedy téma týkající se *základních fyzikálních pojmů* v místě P10. Proto je značka * opět rozšířena přes přechod T10/11 do místa P10. Vývoj sítě je viditelný na obrázku 31.



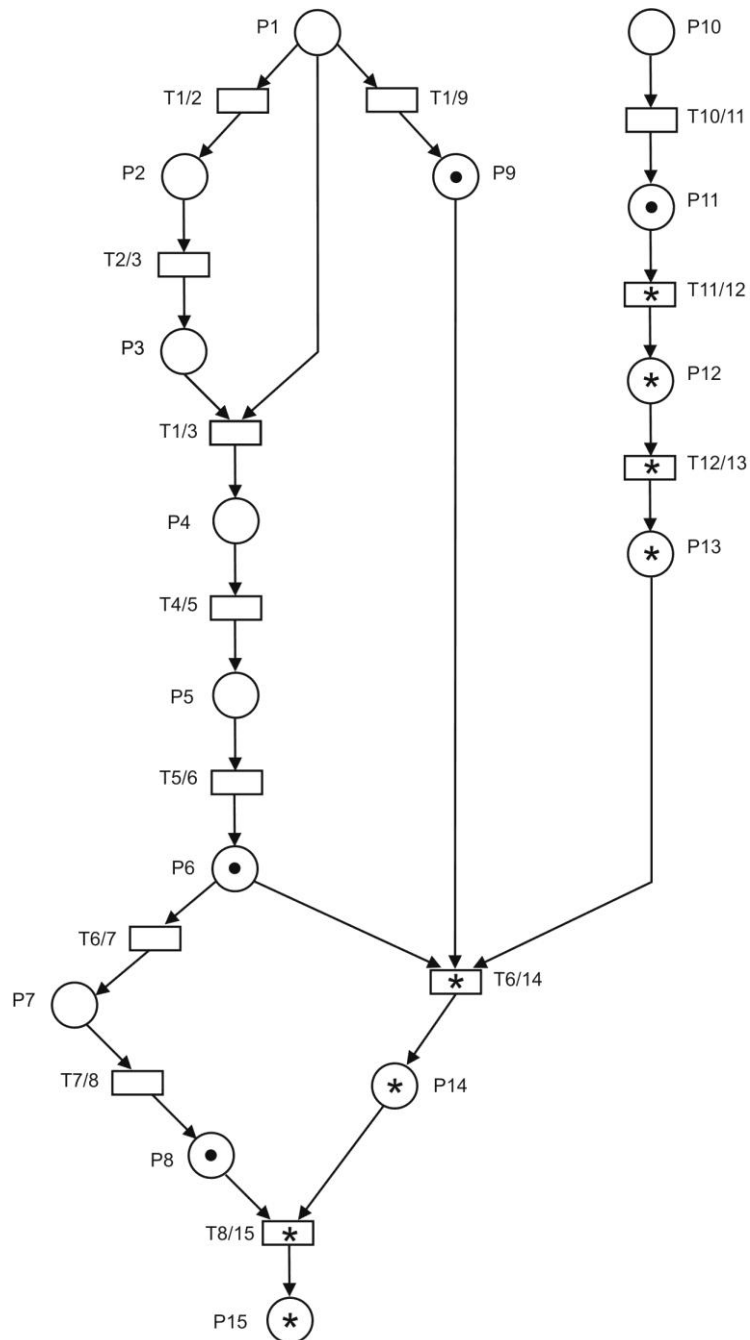
Obr. 31 – Síť zobrazující neúspěch studenta při testu v místě P11

Student byl sítí doveden až k jednomu ze základních témat, tj. *základní fyzikální pojmy* (místo P10), které již po znovuprostudování ovládl a test na toto téma úspěšně absolvoval. Proto dojde v místě P10 k nahrazení značky prvního typu (*) značkou druhého typu (●). Vývoj sítě pro tento případ lze shlédnout na obrázku 32.

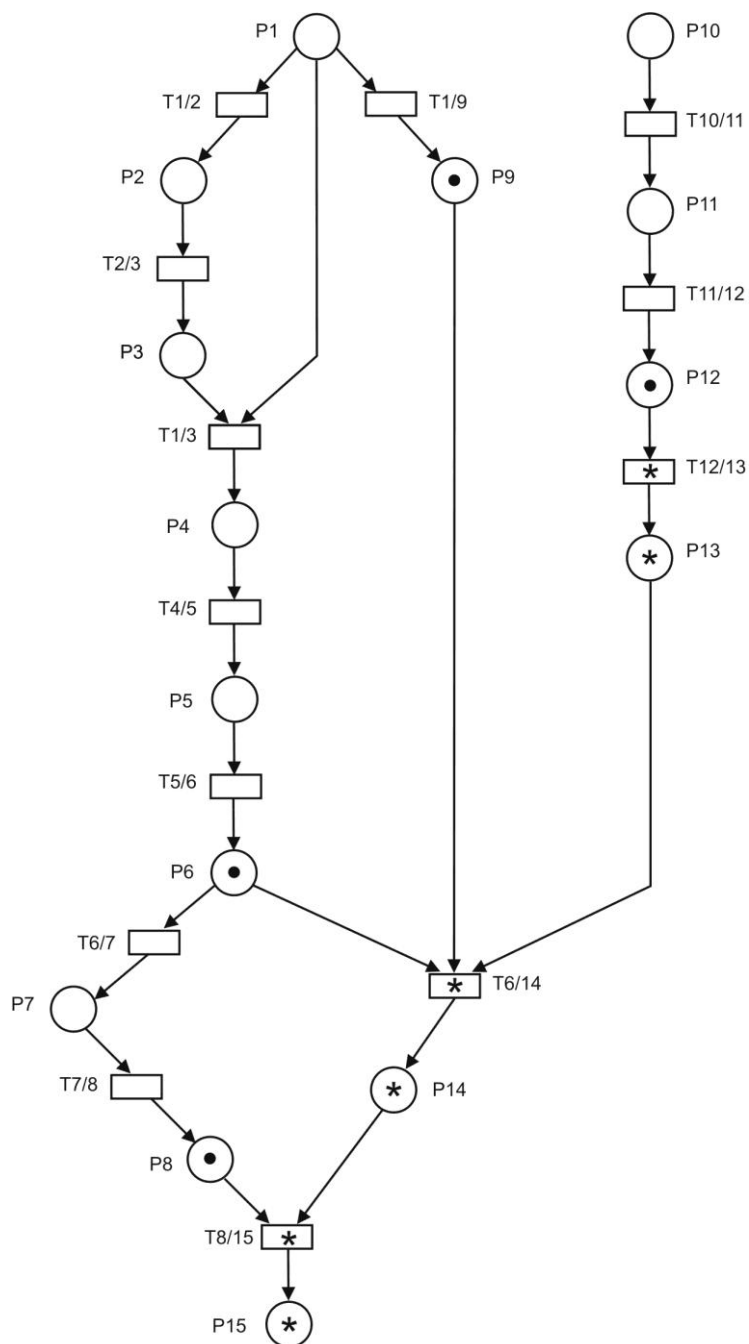


Obr. 32 – Síť zobrazující úspěšné absolvování testu v místě P10

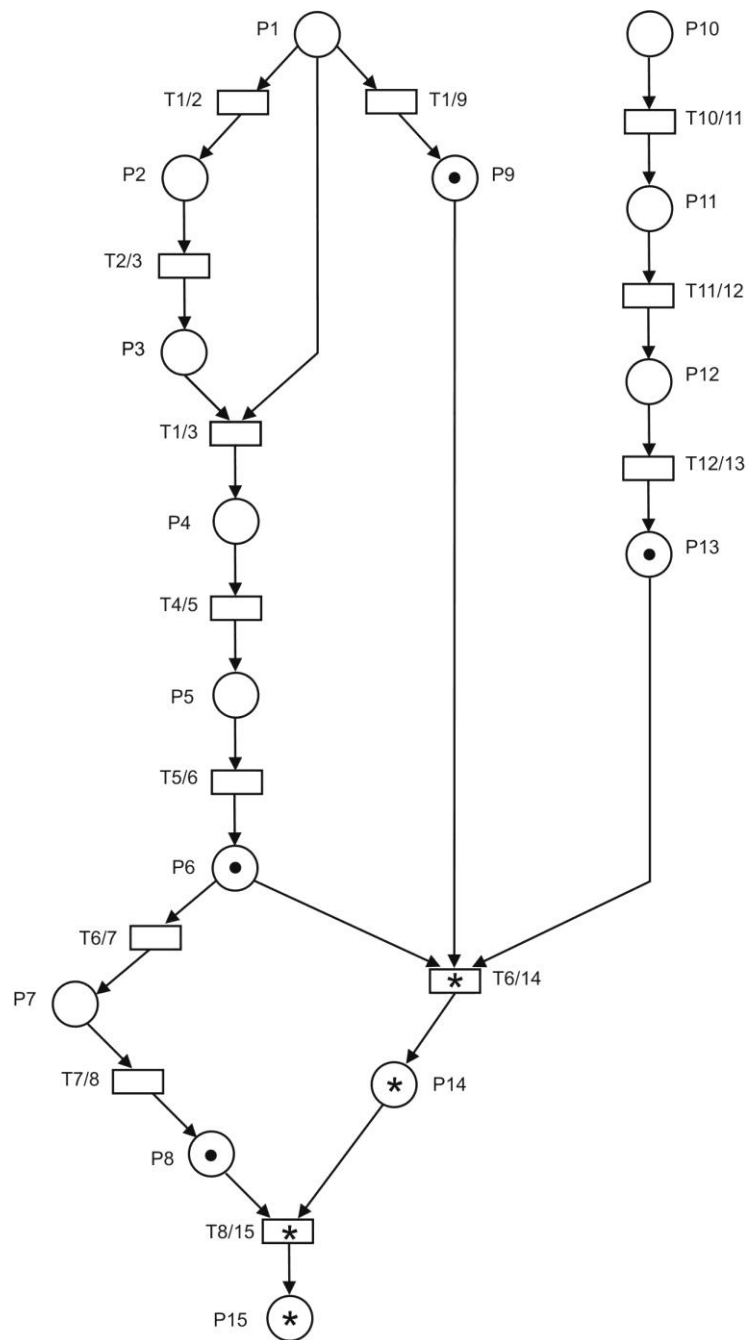
Pro tento příklad již nyní budeme předpokládat, že student napodruhé úspěšně absolvuje všechny testy a nebude sítí nucen se opětovně vracet k předchozí látce. To se ve zobecněné Petriho síti projeví postupným nahrazováním značek prvního typu (*) značkami druhého typu (•) u míst a odstraňováním značek prvního typu (*) z přechodů. Postupný vývoj sítě je viditelný na obrázku 33 až 37.



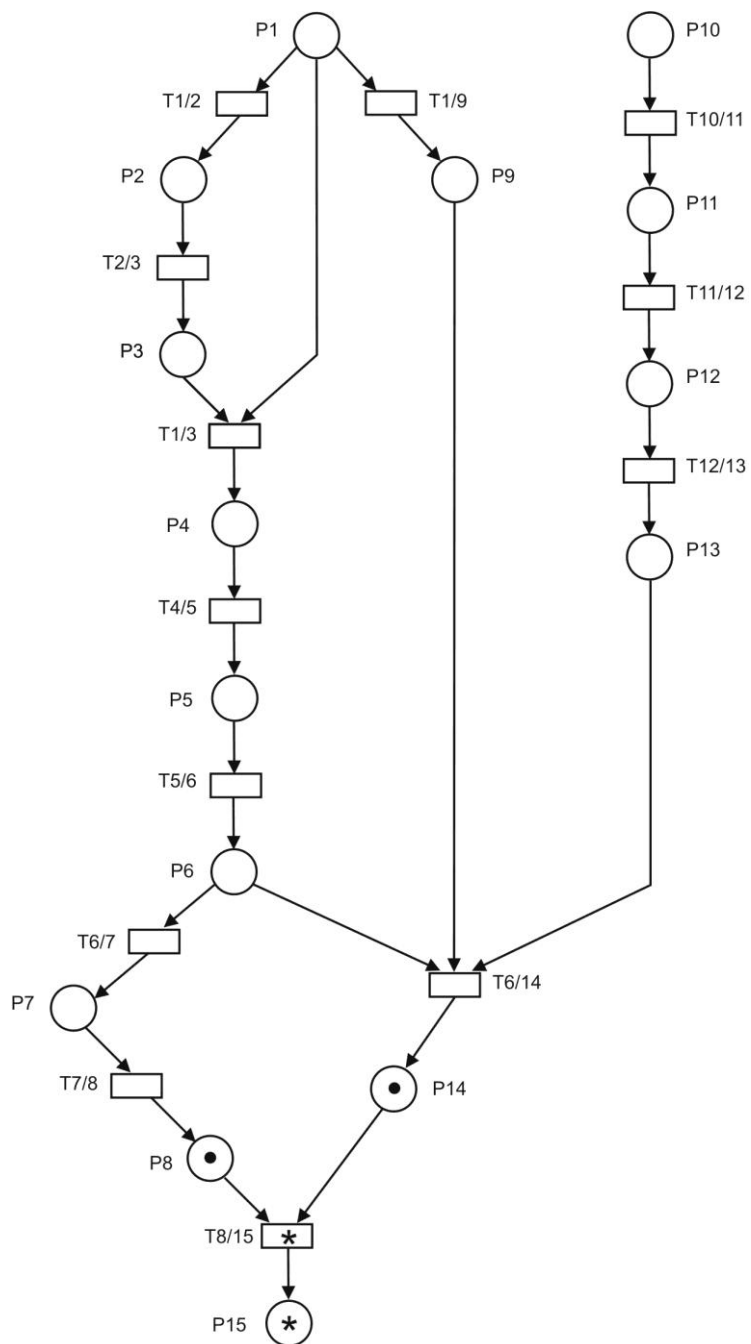
Obr. 33 – Síť zobrazující úspěšné absolvování testu v místě P11



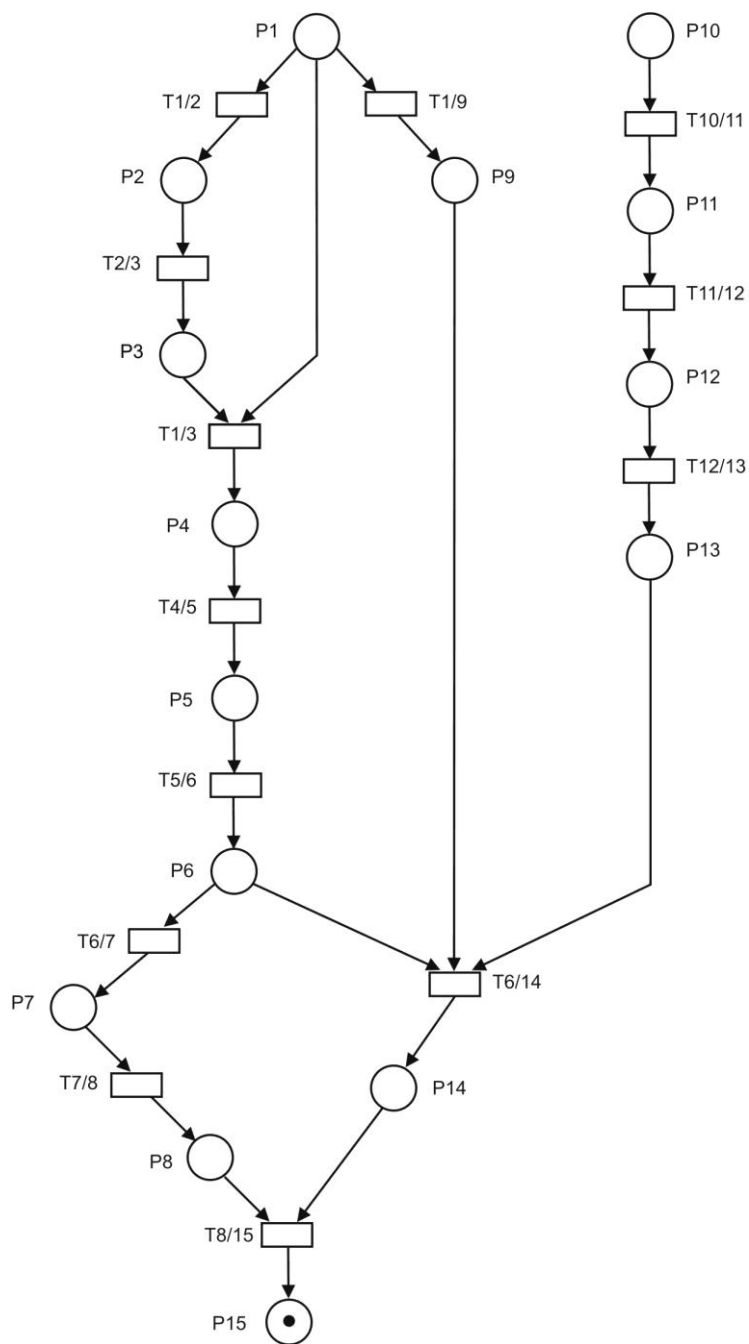
Obr. 34 – Síť zobrazující úspěšné absolvování testu v místě P12



Obr. 35 – Síť zobrazující úspěšné absolvování testu v místě P6, P9 a P13



Obr. 36 – Síť zobrazující úspěšné absolvování testu v místě P8 a P14



Obr. 37 – Síť zobrazující úspěšné zvládnutí problematiky konstrukce kombinačních logických obvodů pomocí tranzistorů

5.5.2 Příklad vyjádření reálného čísla ve formátu IEEE 754

Druhým příkladem, který popisuje vývoj zobecněné Petriho sítě při modelování výukového procesu je případ testující studentovu znalost uložení reálného čísla do 32-bitového formátu IEEE 754 pro čísla v plovoucí řádové čárce. Síť je složena z devíti následujících vzájemně provázaných témat (reprezentovaných jednotlivými místy zobecněné Petriho sítě):

- **P1 – číselné soustavy** – zahrnuje schopnost definovat pojem číselná soustava a schopnost studenta vyjádřit celé číslo a racionální číslo zapsané v desítkové soustavě s konečným počtem dekadických cifer za desetinnou čárkou v soustavě o jiném základu přibližně na takový počet míst, které jsou za řádovou čárkou v dané soustavě k dispozici.
- **P2 – jednotky informace** – zahrnuje znalosti základních jednotek informace v informatice (bit, Byte) a dalších odvozených jednotek (KB, MB, GB, ...) a jejich vzájemných převodů.
- **P3 – převod mezi číselnými soustavami** – představuje schopnost převodů mezi číselnými soustavami používanými ve výpočetní technice, konkrétně mezi desítkovou, binární, osmičkovou a šestnáctkovou číslovou soustavou.
- **P4 – kód a kódová slova** – vyžaduje znalost pojmu kódování dat a redundance v informatice, definice kódu, rozšíření kódu o paritu, typy kódů dle úrovně zabezpečení – zabezpečující a samoopravné kódy.
- **P5 – uložení čísel v pevné řádové čárce** – zahrnuje schopnost zakódovat celé číslo ve vybraném kódu:
 - *přímý kód* – n -bitový přímý kód je zobrazení množiny celých čísel x z intervalu $\langle -2^{n-1} + 1, 2^{n-1} - 1 \rangle$ na množinu binárních kódových slov x' délky n . Pokud binární kódová slova x' vyjádříme celým číslem, lze toto zobrazení popsat jako:

$$x' = x \quad \text{pokud } 0 \leq x \leq 2^{n-1} - 1$$

$$x' = 2^{n-1} - x \quad \text{pokud } -2^{n-1} + 1 \leq x \leq 0$$

- *kódu s posunutou nulou* – n -bitový kód s posunutou nulou je zobrazení množiny celých čísel (kladných i záporných) x na množinu binárních kódových slov x' délky n . Pokud binární kódová slova x' vyjádříme celým číslem, lze toto zobrazení popsat jako $x' = x + \textit{báze_posunutí}$.
- *doplňkový kód* – n -bitový doplňkový kód je zobrazení množiny celých čísel x z intervalu $\langle -2^{n-1}, 2^{n-1} - 1 \rangle$ na množinu binárních kódových slov x' délky n . Pokud binární kódová slova vyjádříme celým číslem, lze toto zobrazení popsat jako:

$$x' = x \quad \text{pokud } 0 \leq x \leq 2^{n-1} - 1$$

$$x' = x + 2^n \quad \text{pokud } -2^{n-1} \leq x < 0.$$

- **P6** – *převod reálného čísla z desítkové do dvojkové číselné soustavy* – představuje schopnost převést reálné číslo $N = \sum_{i=-1}^{-m} a_i \cdot 2^i$, kde $a_i = 0$ nebo

1 z desítkové do dvojkové soustavy pomocí postupu:

$$2 \cdot N = a_{-1} + \underbrace{\frac{1}{2} a_{-2} + \frac{1}{4} a_{-3} + \dots + \frac{1}{2^{m-1}} a_{-m}}_{0 \leq M < 1}, \text{ kde:}$$

$$2 \cdot N < 1 \Leftrightarrow a_{-1} = 0 \text{ a } M > 0,$$

$$2 \cdot N > 1 \Leftrightarrow a_{-1} = 1 \text{ a } M > 0,$$

$$2 \cdot N = 1 \Leftrightarrow a_{-1} = 1 \text{ a } M = 0.$$

- **P7** – *kód s posunutou nulou použitý ve formátu IEEE 754* – představuje znalost přesného formátu kódu s posunutou nulou používaného u 32-bitového IEEE standardu pro uložení čísel v pohyblivé řádové čárce, tj.

$$x' = 2^{n-1} - 1 + x, \text{ kde:}$$

- x' – je n -bitové kódové slovo uložené v kódu s posunutou nulou,
- $2^{n-1} - 1$ – je *báze posunutí* pro formát IEEE754, přičemž n značí počet bitů, ve kterých je celé číslo v kódu s posunutou nulou uloženo,
- x – představuje celé číslo zobrazené v kódu s posunutou nulou.

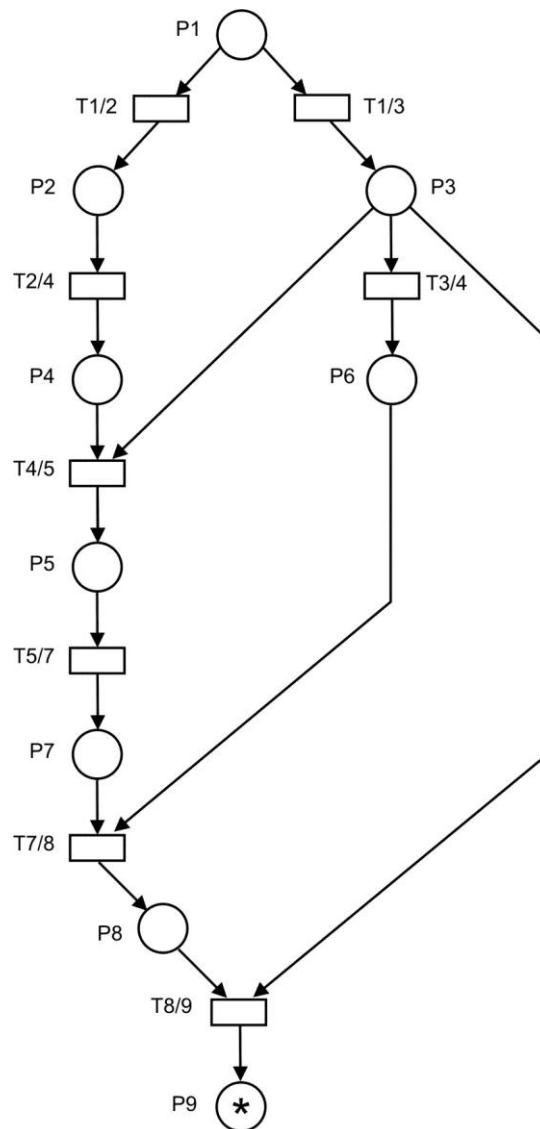
- **P8 – 32-bitový formát IEEE 754** – představuje znalost jednotlivých částí čísla ve 32-bitovém IEEE formátu pro čísla v pohyblivé řádové čárce, tj.:
 - *znaménkový bit* – je první bit zleva u čísla ve formátu IEEE 754, pokud je hodnota tohoto bitu 0, číslo je kladné, pokud je hodnota bitu 1, uložené číslo je záporné,
 - *exponent* – exponent e je 8-bitové kódové slovo reprezentující celé číslo uložené v kódu s posunutou nulou,
 - *mantisa* – je vždy reálné číslo, které je ve 32-bitovém formátu IEEE 754 vždy uloženo ve 23 bitech, mantisu lze vyjádřit dle předpisu
$$m = \sum_{i=1}^n a_i \cdot 2^{-i},$$
 kde:
 - n – počet bitů, ve kterých je mantisa zaznamenána, u 32-bitového formátu IEEE 754 je počet bitů mantisy 23,
 - $a_i = 1$ nebo $a_i = 0$ dle konkrétní hodnoty bitu mantisy.

Poznámka: Pokud je hodnota mantisy z intervalu $< 1, 2$, číslo je v tzv. normalizovaném tvaru. Každé číslo v pohyblivé řádové čárce lze normalizovat, tj. převést do normalizovaného tvaru.
- **P9 – uložení reálného čísla do 32-bitového formátu IEEE754** – představuje znalost uložení reálného čísla x do formátu IEEE 754 a jeho převod do hexadecimální soustavy.

Podrobnější výklad problematiky číselných soustav a kódování dat je k dispozici v [19a] a [20a].

Účelem tohoto příkladu je otestovat znalost celku P9, tedy zda je student na základě znalostí tematických celků P1 až P8 schopen **uložit reálné číslo do 32-bitového formátu IEEE 754 pro čísla v plovoucí řádové čárce**. Z tohoto důvodu je na místo P9 směřován dotaz. Zobecněná Petriho síť znázorňující daný příklad je zobrazena na obrázku 38.

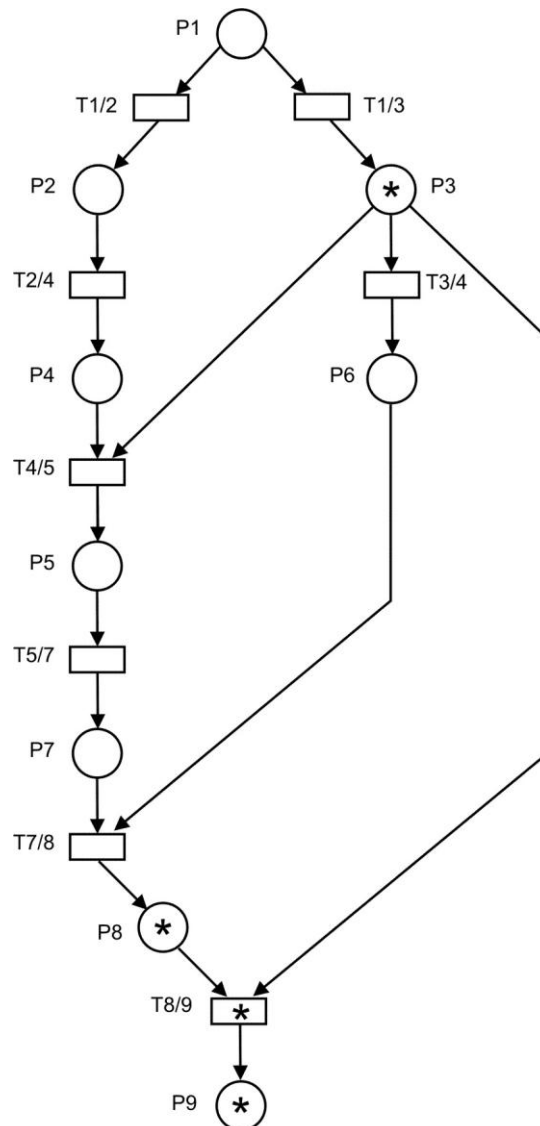
Pokud bude student v testu v místě P9 úspěšný, místo P9 bude obarveno značkou druhého typu (•). To v praxi znamená skutečnost, že student bude moci přejít k následujícím tématům, pro které je znalost tématu P9 předpokladem. Výukový proces se v tomto případě bude vyvíjet opět dle pravidel klasické Petriho sítě.



Obr. 38 – Síť znázorňující příklad vyjádření reálného čísla ve formátu IEEE 754

Uvažujme ale opět případ, kdy student test v místě P9 úspěšně neabsolvoval, proto se další vývoj bude řídit pravidly zobecněné Petriho sítě a značka prvního typu (*) se bude šířit přes přechod T8/9 na všechna jeho vstupní místa, tj. na látku týkající se **32-bitového formátu IEEE 754** (místo P8) a tématiku týkající se **převodu mezi**

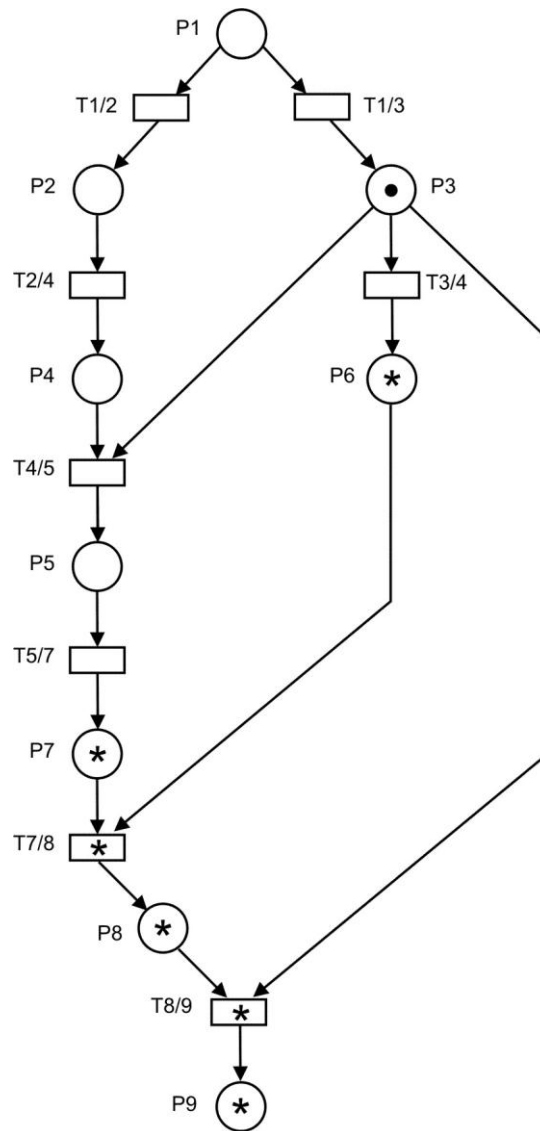
číselnými soustavami (místo P3), jejíž nastudování je pro úspěšný převod reálného čísla do 32-bitového formátu IEEE 754 předpokladem. Vývoj zobecněné Petriho sítě je znázorněn na obrázku 39.



Obr. 39 – Síť znázorňující neúspěch studenta při testu v místě P9

V následujícím kroku nastane situace, kdy student úspěšně absolvoval test, ověřující znalost *převodu mezi číselnými soustavami* v místě P3. Proto se v tomto místě nahradí značka prvního typu (*) značkou druhého typu (●). Student ale není stále úspěšný v látce týkající se *32-bitového formátu IEEE 754* (místo P8), proto jej síť vede k prostudování témat předcházejících, tedy k tématu *kódu s posunutou nulou pro*

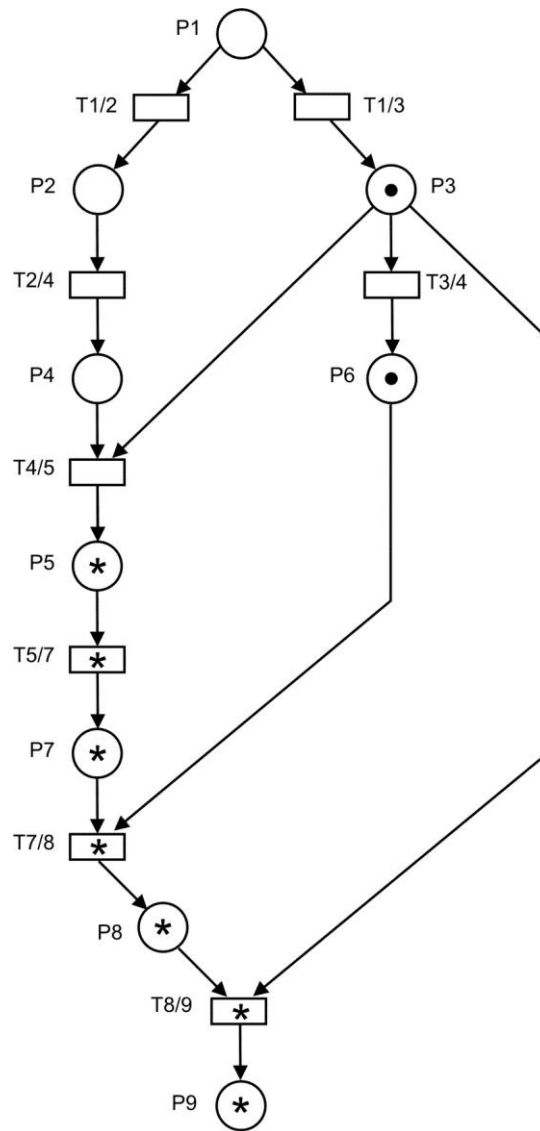
formát IEEE 754 a definici *kódu a kódového slova*. Značka prvního typu se proto rozšíří přes přechod T7/8 do míst P7 a P4. Tento vývoj sítě lze sledovat na obrázku 40.



Obr. 40 – Síť znázorňující neúspěch studenta při testu v místě P8 a úspěch v testu v místě P3

Ani v testu, kterým se ověřuje znalost *kódu s posunutou nulou pro formát IEEE 754* nebyl student úspěšný. Z tohoto důvodu jej opět bude síť navigovat k tématu předcházejícímu, tj. látce zaměřené na *uložení čísel v pevné řádové čárce* (P5). Z tohoto důvodu se v síti bude značka prvního typu (*) šířit přes přechod T5/7 do místa P5. Naopak test, který byl zaměřen na problematiku *převodu reálného čísla z desítkové do*

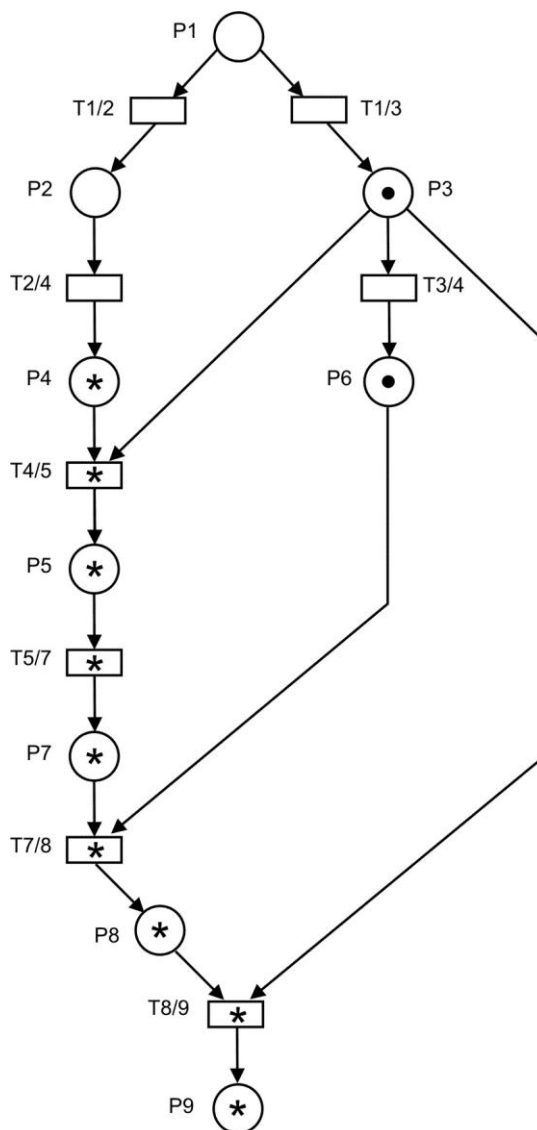
dvojkové číselné soustavy student absolvoval úspěšně, proto se v místě P6 změnila značka * na značku druhého typu •. Vývoj sítě je znázorněn na obrázku 41.



Obr. 41 – Síť znázorňující neúspěch studenta při testu v místě P7 a úspěchu v místě P6

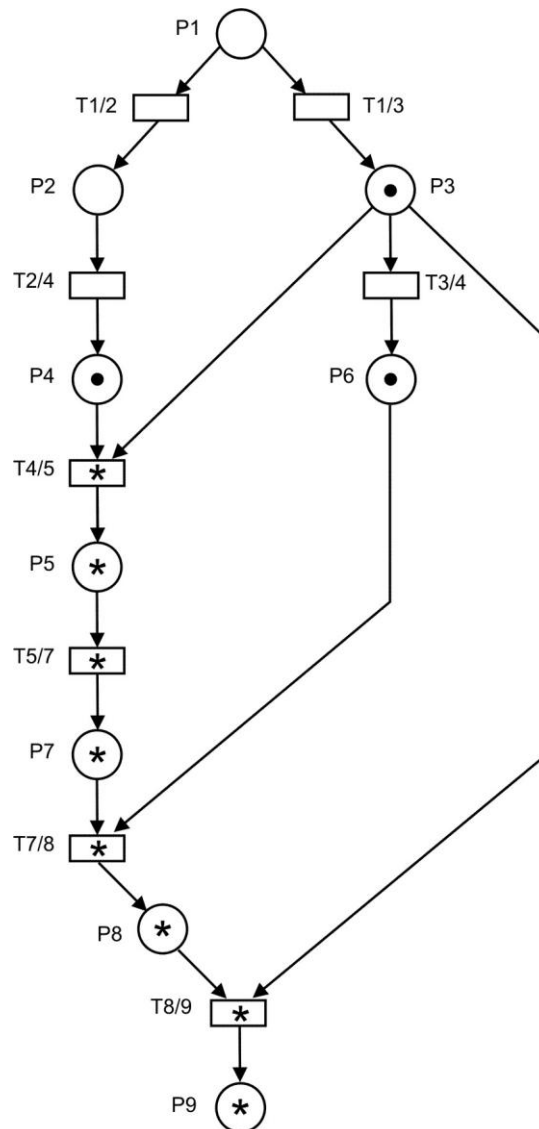
Ani v testu, který je zaměřen na problematiku *uložení čísel v pevné řádové čáře* (místo P5) student úspěšný nebyl. Proto jej síť bude opět navigovat k opětovnému prostudování témat, která problematice uložení čísel v pevné řádové čáře předcházejí. Značka * se proto bude šířit přes přechod T4/5 do místa P4, které reprezentuje problematiku *kódu a kódových slov*. Další vstupní místo přechodu T4/5 je místo P6, které reprezentuje problematiku *převodu reálného čísla z desítkové do dvojkové číselné*

soustavy. Vzhledem k tomu, že ale již student test v tomto místě absolvoval úspěšně, znak prvního typu * se do místa P6 šířit nebude. Vývoj sítě je znázorněn na obrázku 42.



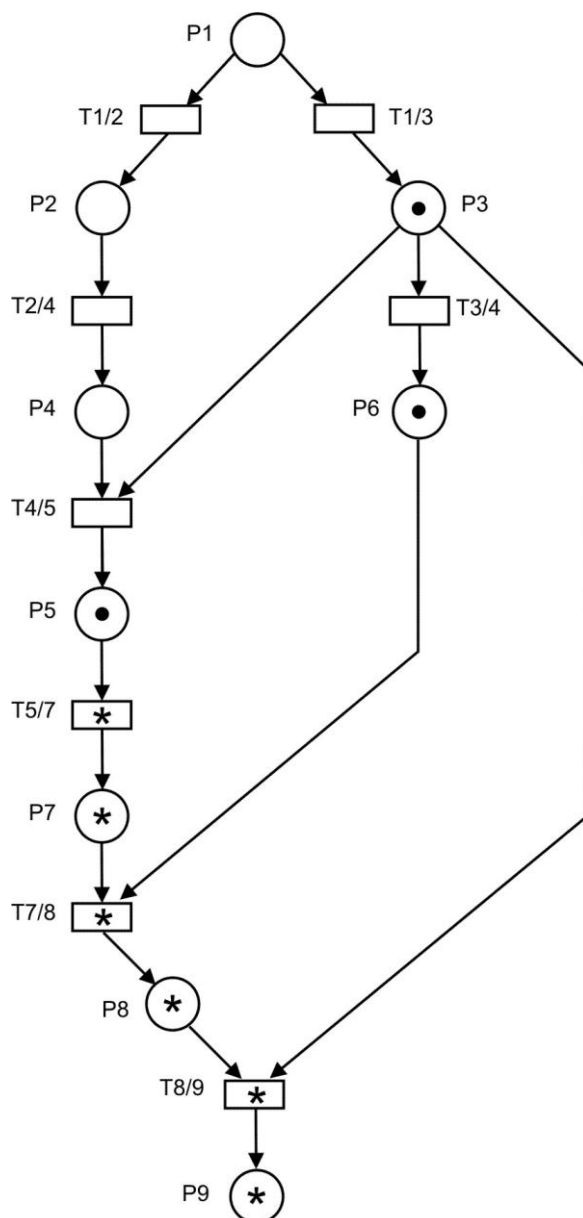
Obr. 42 – Síť znázorňující neúspěch studenta při testu v místě P5

Nyní již student úspěšně absolvoval test pokrývající tematiku *kódu a kódového slova* v místě P4. V síti proto může dojít v místě P4 k nahrazení značky prvního typu (*) značkou druhého typu (•), tento vývoj je patrný na obrázku 43.

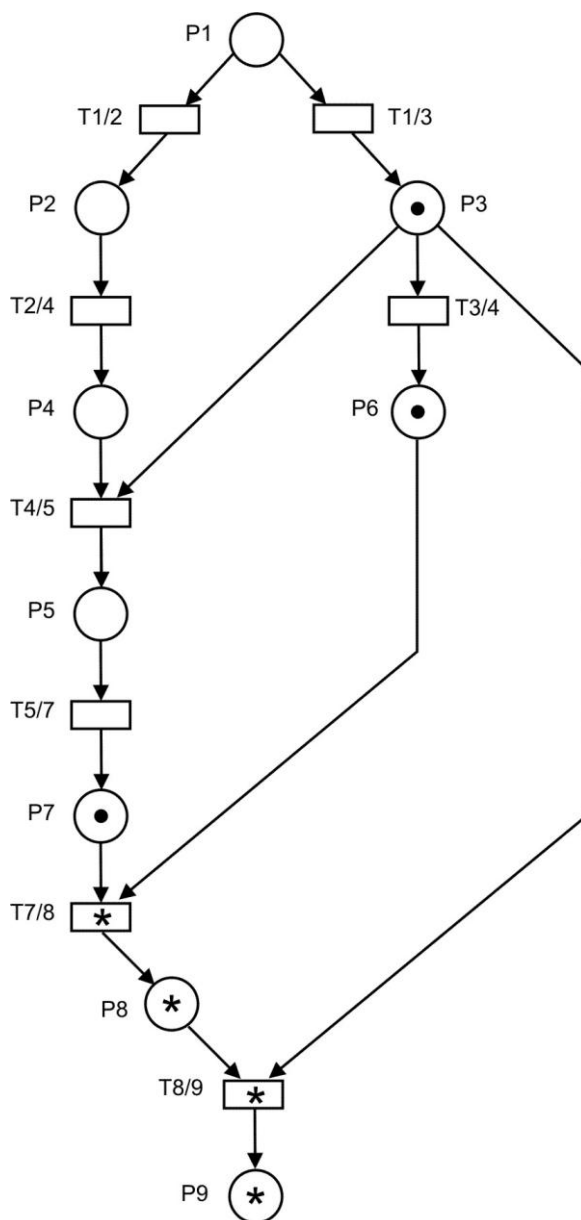


Obr. 43 – Síť znázorňující úspěch studenta při testu v místě P4

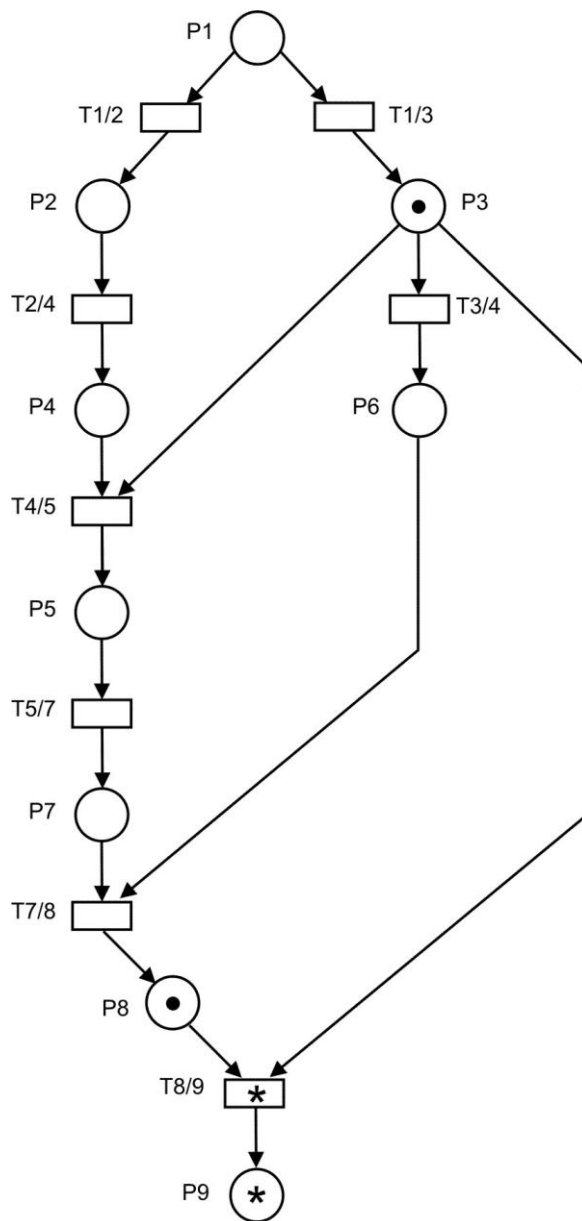
Student již nyní nastudoval všechna témata, potřebná k *převodu reálného čísla do 32-bitového formátu IEEE 754* pro čísla v plovoucí řádové čárce a úspěšně i absolvoval všechny testy. Nemusí se proto již vracet k opětovnému prostudování látky. Proto budou v zobecněné Petriho síti postupně nahrazovány u míst značky prvního typu (*) značkami druhého typu (•) a u přechodů značky prvního typu (*) odstraňovány. Postupný vývoj sítě je znázorněn na obrázcích 44 – 47.



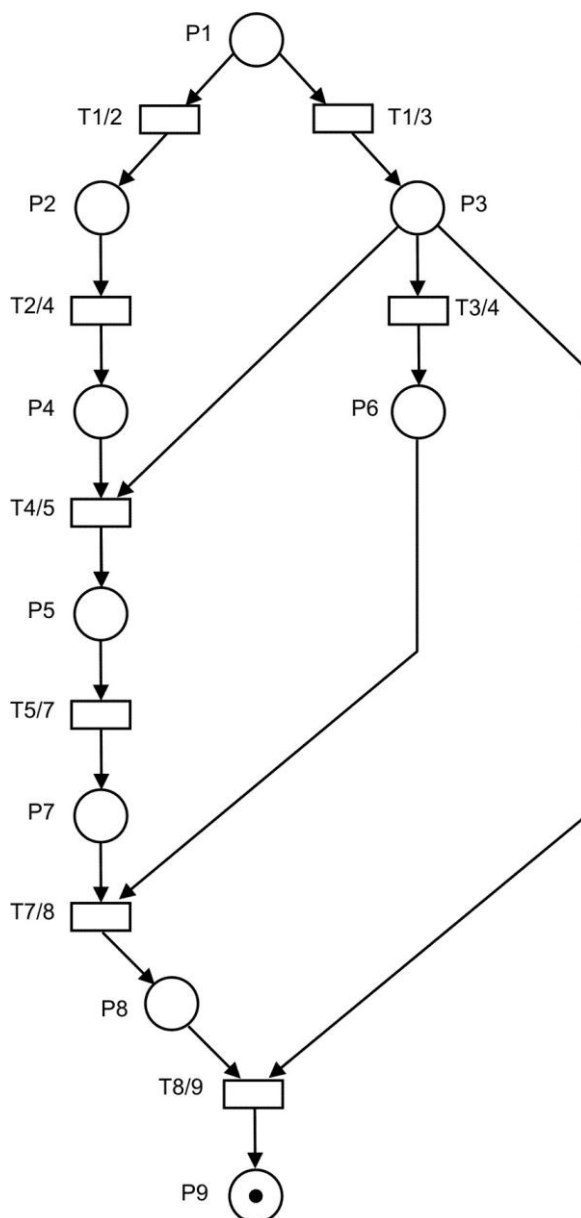
Obr. 44 – Síť znázorňující úspěch studenta při testu v místě P5



Obr. 45 – Síť znázorňující úspěch studenta při testu v místě P7



Obr. 46 – Síť znázorňující úspěch studenta při testu v místě P8



Obr. 47 – Síť znázorňující úspěšné nastudování problematiky převodu reálného čísla do 32-bitového formátu IEEE 754

6. Závěry práce

6.1 Shrnutí a diskuse

Prudký rozvoj informačních a komunikačních technologií spolu se snahou minimalizace spotřeby času se promítají do téměř všech oborů lidské činnosti. Ani oblast výuky a vzdělávání není proto výjimkou. Z tohoto důvodu se velmi rozšířila distanční forma studia a vzdělávání prostřednictvím elektronických výukových aplikací, které přináší úsporu času pedagogů i studentů.

Téměř všechny instituce, které se zabývají vzděláváním, využívají v procesu výuky elektronické výukové programy, ať již jde o aplikace vyvinuté přímo dle svých konkrétních potřeb či programy typu „open source“.

Tak jako využití všech technologií, i využití e-learningu v procesu vzdělávání přináší kromě výhod i některé nevýhody. Z tohoto důvodu se autorka práce věnovala v rešeršní kapitole 3.1 definováním pojmu e-learning a shrnutí jeho výhod i úskalí využití e-learningu ve výuce.

Za jednu z největších nevýhod distančního typu vzdělávání a vzdělávání prostřednictvím elektronických výukových kurzů lze považovat absenci přímého kontaktu studenta a pedagoga. Pedagog při prezenčním typu výuky koriguje postup studenta probíranou látkou, dle zjištěných znalostí studentů pedagog například některé části látky znovu zopakuje a se studenty procvičí či přejde rovnou k vysvětlování učiva navazujícího. Tyto funkce musí v elektronickém výukovém systému zastávat zpětnovazební prvky. Funkcemi a možnostem realizace zpětné vazby v elektronickém výukovém systému se autorka věnovala v rešeršní kapitole 3.2.

Vzhledem k tomu, že je vhodné studenta ke studiu také motivovat, byla do disertační práce zařazena kapitola 3.3, která je věnována možnostem hodnocení odpovědí studenta na testovací otázky. Kapitola byla věnována především úvaze, zda připouštět odpověď „nevím“ jako jednu z možných odpovědí na testovou otázku. Dle pedagogických zkušeností autorky práce a jejich spolupracovníků byla možnost

odpovědi „nevím“ připuštěna a byl navržen systém hodnocení u základních testových typů otázek – dichotomických (ano/ne), otázek s $N > 2$ odpověďmi, které se nevylučují ani není jisté, zda některá nastane, otázek na výběr pouze jedné z několika alternativ a úloh typu seřazení množiny N prvků dle hodnoty nějakého atributu. Součástí této kapitoly byla i úvaha o stanovení kritéria V pro úspěšné absolvování testu. Kritérium V bylo stanoveno s ohledem na přidělenou váhu otázky testu a počtu dosažených bodů za odpověď na otázku. Úvaha obsažená v této kapitole se snaží být jedním z přínosů disertační práce.

Elektronický výukový systém tedy musí být navržen tak, aby vhodně vedl studenta výukovým procesem a na základě znalostí studenta doporučoval další směr studia a získával zpětnou vazbu pro pedagoga i studenta samotného. Kontrolování práce studentů, přezkušování jejich znalostí a získávání zpětné vazby z těchto informací je proto jednou z nejdůležitějších funkcí výukového systému. Proto v kapitole 4 autorka srovnala formální modely výpočtu, které by bylo možné využít i v elektronické výuce pro modelování procesu průchodu studenta výukovým kurzem a využít je jak pro jeho navigaci, tak i pro zprostředkování zpráv vyučujícímu i studentovi. Z řady formálních výpočetních modelů by přicházelo k modelování průchodu studenta kurzem využít konečný automat, zásobníkový automat, Turingův stroj či Petriho síť. V rešeršní podkapitole 4.2 bylo proto provedeno stručné zhodnocení těchto modelů z hlediska využití pro modelování procesu průchodu studenta výukovým kurzem. Podkapitola 4.3 si kladla za cíl shrnutí existujících případů modelování průchodu studenta elektronickým výukovým kurzem za pomoci Petriho sítí i dalších formálních matematických modelů. V kapitole jsou obsaženy modely e-learningových systémů českých i zahraničních autorů.

Dle hodnocení provedeného v kapitole 4.2 byl z nejčastěji užívaných modelů výpočtu zvolen jako nevhodnější pro modelování procesu průchodu studenta výukovým kurzem model Petriho sítě.

Aby Petriho síť lépe odrážela studentův průchod výukovým kurzem, bylo vhodné Petriho síť zobecnit tak, aby bylo možné zobrazit jak proces dotazů tak

i vyhledávání odpovědí. Proto byla v kapitole 5.4 nově navržena „zobecněná“ Petriho síť, která lépe odpovídá potřebám modelování automatizovaného výukového systému. Dotazy jsou ve zobecněné Petriho síti znázorněny značkami *, odpovědi se znázorňují značkami •. Zvládnutí látky lze formulovat jako požadavek na dodání informace ze strany studenta. Požadavek se projeví jako značení prvního typu (znak *) umístěné do příslušného místa zobecněné Petriho sítě. Pokud student látku ovládá (splnil podmínky testu), je změněn znak * na znak •. V případě nevyhovujícího výsledku testu je znak * v daném místě ponechán a dotaz se proti směru orientace grafu rozšíří na vstupní přechody daného místa a všem místům bezprostředně předcházejícím (tzn. že student se musí ve výukovém schématu navrátit k látce, která je předpokladem ke zvládnutí látky, u které nebyl úspěšný a musí ji prostudovat znovu). Pokud je student neúspěšný i zde, test se opakuje tak dlouho, dokud lze nahradit znak * znakem • (tj. úspěšnou odpovědí). Zavedení pojmu zobecněné Petriho sítě a definování jejího vývoje je původní myšlenkou předkládané disertační práce.

Pravidla zobecněné Petriho sítě a její vývoj byly ověřeny v kapitole 5.5 předkládané disertační práce. Tato kapitola obsahuje praktické ukázky příkladů na využití v práci navrženého modelu zobecněné Petriho sítě pro konkrétní výukové tématické celky, konkrétně na problematiku návrhu kombinačních logických obvodů pomocí tranzistorů a na tématický celek věnující se vyjádření reálného čísla ve 32-bitovém formátu IEEE 754 pro čísla v pohyblivé řádové čárce.

6.2 Rekapitulace výsledků práce

Disertační práce kladla za cíl vystihnout důležité zásady, kterými se má řídit elektronický výukový systém, konkrétně se jedná o:

- **Navigaci studenta ve výukovém procesu** – dle výsledků testů studenta navrhování postupu studenta jednotlivými kapitolami probírané látky,
- **Získávání zpětné vazby pro:**
 - pedagoga za účelem hodnocení výsledků studenta,
 - studenta za účelem ověřování vlastních znalostí a dovednost.

Další cíle předkládané disertační práce byly:

- **Vytvoření obecného modelu datové struktury**, popisující stavový prostor používání elektronického výukového kurzu, tak aby umožňoval navigaci studenta, generování rad a doporučení studentovi a získávání zpětné vazby pro lektora i studenta.
- **Navržení omezení a konkretizace modelu** při navigaci studenta výukovým systémem, zabezpečující základní požadavky, tj. navigaci studenta a získávání zpětné vazby.

Pro modelování výukového systému, který respektuje výše uvedené zásady, byla autorkou navržena původní modifikace Petriho sítě, tzv. **zobecněná Petriho síť**, která umožňuje zobrazit proces dotazů a zároveň i vyhledávání odpovědí. Tato zobecněná Petriho síť odpovídá potřebám modelování automatizovaného výukového systému.

Pomocí zobecněné Petriho sítě byl v disertační práci navržen **model průchodu studenta** konkrétním úsekem probíraného učiva. Tento model byl ověřen na příkladech věnovaných problematice návrhu kombinačních logických obvodů pomocí tranzistorů a vyjádření reálného čísla ve 32-bitovém formátu IEEE 754 pro čísla v pohyblivé řádové čárce. Model průchodu studenta elektronickým výukovým systémem a jeho ověření autorka považuje za svůj původní přínos k rozvoji poznatků.

V neposlední řadě jsou přínosem předkládané disertační práce úvahy o hodnocení odpovědí studentů v testech. Úvaha spočívá v připuštění odpovědi „nevím“ a návrhu bodového hodnocení této odpovědi v různých typech testových otázek. Snahou existence této alternativy odpovědi je upřednostnit studenty kritické ke svým znalostem, kteří možnost odpovědi „nevím“ využijí, oproti studentům, kteří vybírají odpověď náhodným způsobem. Součástí této úvahy byl i návrh jakým způsobem stanovit kritérium pro úspěšné absolvování testu a to na základě nastavení váhy jednotlivých otázek a počtu dosažených bodů za jednotlivé odpovědi. Tyto úvahy jsou založeny na pedagogických zkušenostech autorky práce a jejích kolegů.

6.3 Aplikace výsledků práce

Výsledky předkládané disertační práce je možné využít v následujících oborech a odvětvích:

- **Při výuce odborných předmětů** s využitím elektronických výukových systémů (zejména v distanční formě výuky) – jako podklad pro navigaci studenta elektronickým kurzem, generování rad a doporučení studentovi a získávání zpětné vazby o jeho studiu,
- **V akademické sféře** – jako další příklad o možnosti využití Petriho sítí a modelů procesu založených na principech Petriho sítí pro modelování zpětné vazby v e-learningu.

6.4 Možnosti dalšího rozvoje tématu

Problematika optimalizace průchodu studenta elektronickým výukovým systémem, kterou se disertační práce zabývá, je poměrně rozsáhlá a je možné ji nadále rozvíjet v řadě směrů.

Důležitým rozšířením by mohla být například analýza průběhu studenta výukovým kurzem s cílem odhadnout jeho studijní typ. Určit jeho silné stránky a slabiny (například zda u něj činí potíž abstrakce, či naopak aplikace zvládnuté teorie na praktický úkol) a podle zjištěných poznatků pak přizpůsobit výuku. Toto přizpůsobení by mohlo spočívat například ve vytvoření několika variant kurzu a doporučení nejvhodnější z těchto variant podle zjištěného studijního typu.

Dalším směrem rozvoje navržené teorie by mohlo být zavedení vícehodnotového posouzení kontrolních otázek. V předloženém návrhu je zvládnutí posuzovaného úseku výuky posuzováno binárně. V případě, že student dílčí téma zvládl, může pokračovat studiem dalšího tématu. Pokud ne, musí se vrátit. Tento návrat je dosud prováděn do stejného místa nezávisle na tom, zda student nesplnil žádnou z podmínek nebo pouze některé a v tom případě, kde jsou jeho nedostatky. To může postup výuky zpomalit, protože pokud není téma rozčleněno dostatečně jemně, může být požadováno i opakování toho, co již student zvládl. Stojí za úvahu klasifikovat odpovědi ve více

stupních a volit pro tyto stupně různé orientované hrany pro navigaci studenta. Pro tento účel bude asi vhodnější volit hodnocení úrovně splnění testu v několika málo stupních. Užití „spojité“ fuzzy logiky se předběžně jako vhodné nejeví.

Perspektivně přichází do úvahy i kombinace víceúrovňového hodnocení odpovědí s diferenciací navigace podle studijního typu konkrétního studenta, zjišťovaného analýzou jeho průchodů výukovým systémem.

V této disertační práci jsou uvedeny modely popisující průchod studenta vybranými tématy z předmětů, na jejichž výuce se autorka podílí. Model zobecněné Petriho sítě lze nadále rozvíjet pro další tématické celky, síť lze například namodelovat průchod celými předměty.

V rámci další vědecko-výzkumné činnosti je možné model zobecněné Petriho sítě dále rozvíjet a modifikovat. V aplikační oblasti lze model zabudovat do existujících elektronických výukových programů či navrhnout nové praktické výukové aplikace, respektující pravidla, která byla pro model zobecněné Petriho sítě definována.

7. Seznam použité literatury

- [1] KVĚTOŇ Karel, *Základy online výuky a eLearningu*, Praha: nakladatelství CESNET, 2003, 1. vyd., 52 s.
- [2] VOTAVA Václav, HÁN Jan., *E-Learning dostupný pro malé firmy i jednotlivce*, Praha: E-learning 2004, 2004, ISSN 80-86433-3 -7.
- [3] HÁN Jan, *Architektura epizodálního on-line kurzu*, sborník konference ICTE 2001, Rožnov pod Radhoštěm, 201, ISBN 80-7042-808-2.
- [4] FOJTÍK Rostislav, *Možnosti testování v e-learningu*, Konference ICTE 2002, Rožnov pod Radhoštěm 2002, ISBN 80-7042-828-7.
- [5] CROSS Jay, *Neformální historie e-learningu* [online], 2004, [cit 2004-4-1]. <<http://www.skolaonline.cz/scripts/detail.php?id=4205>>.
- [6] ŠORM Milan, *Systém pro podporu distančního vzdělávání*, Brno: Masarykova univerzita, 2003, 88 s., Rigorózní práce. Kapitola 8, *Modelování průchodu studenta studiem*, s. 64 – 72.
- [7] ČERNÁ Hana, *Návrh a implementace síťového výukového systému*, Brno: MZLU, 1999, 83 s., Diplomová práce. Kapitola 2, *Informační systémy ve výuce*, s. 8 – 12, Kapitola 4, *Implementace síťového výukového systému*, s. 16 – 26.
- [8] RYBIČKA Jiří, *Konstrukce testů ve výukových informačních systémech*, Informační systémy a jejich aplikace, Brno: PEF MZLU, 1995.
- [9] RYBIČKA Jiří, *Problémy implementace výukových informačních systémů*, Informační systémy a jejich aplikace, Brno: PEF MZLU, 1997.
- [10] ČEŠKA Milan, *Petriho síť*, Brno: Akademické nakladatelství CERM, 1994, 94 s., ISBN 80-85867-35-4.
- [11] BAYER Jiří, HANZÁLEK Zdeněk, ŠUSTA Richard, *Logické systémy pro řízení*, Praha: vyd. ČVUT, 2000, ISBN 80-01-02147-5.

- [12] SVOBODA Luboš, *Editor Petriho sítí*, Praha: FEL ČVUT, 1993, 62 s., Diplomová práce. Kapitola 1, *Petriho síť*, s. 5 – 11.
- [13] ČEŠKA Milan a kol., *Petriho síť – přednášky*, [online], květen 2010, Dostupné z: <<http://www.fit.vutbr.cz/study/courses/PES/public/#Literatura>>
- [14] PELIKÁN J., *Programovaná výuka v kombinaci s hypertextem*, Zpravodaj ÚVT MU, ISSN 1212-0901, 1998, roč. 9, č. 2, s. 9 – 13.
- [15] CHYTIL Michal, *Automaty a gramatiky*, Praha: Nakl. techn. literatury, 1984, 1. vyd., 331 s.
- [16] HOPCROFT John E., ULLMAN Jeffrey D., *Formálne jazyky a automaty*, Bratislava: vyd. Alfa, 1978, 342 s., ISBN 63-096-78.
- [17] KOCUR Pavel, *Úvod do teorie konečných automatů a formálních jazyků*, Plzeň: vyd. Západočeské univerzity, 2001, 104 s., ISBN 80-7082-813-7.
- [18] PAPÍK Martin, *Nástroje procesního modelování*, Praha: PEF ČZU, 2005, 74 s., Diplomová práce. Kapitola 3, *Petriho síť*, s. 28 – 44.
- [19] BARAK Miri, RAFAELI Sheizaf, *Online question-posing and peer-assessment as means for web-based knowledge sparing in learning*, in International Journal of Human-Computer Studies, vol 61, issue 1, July 2004, ISSN 1071-5819.
- [20] VONDRÁK Ivo, *Jazyky PN a Chomského hierarchie*, [online], září 2006, Dostupné z: <<http://www.cs.vsb.cz/markl/pn/data/NNPN32.pdf>>.
- [21] VANÍČEK Jiří et al., *Mathematical Theory of programs*, Part 1: computation models, Praha: PEF ČZU, 2004, 1. vyd., 44 s., Kapitola 6, *Finite automata*, s. 17 – 28.
- [22] RACEK Stanislav, ROUBÍN Miroslav, *Pravděpodobnostní modely počítačů*, vyd.: Západočeská univerzita, Plzeň, 1997, 1. vyd., 154 s., ISBN 80-7082-300-3.

- [23] ČEŠKA Milan a kol., *Turingovy stroje – přednášky*, [online], duben 2010, Dostupné z: < <http://www.fit.vutbr.cz/study/courses/TIN/public/Prednasky/tin-pr07-ts1.pdf> >.
- [24] KUPČA Vojtěch, *Teorie a perspektiva kvantových počítačů – Klasický Turingův stroj*, [online], leden 2001, <<http://cml.fsv.cvut.cz/~kupca/qc/node13.html>>.
- [25] BALOGH Zoltán, KLIMEŠ Cyril, *Možnosti riadenia výukových procesov s použitím Petriho sietí a fuzzy logiky*, Sborník příspěvku z konference DIVAJ 2008, Nitra: Univerzita Konštatína Filozofa, 2008, s. 92-97, ISBN 978-80-809431-7-2.
- [26] NAGYOVÁ Ingrid, *Modelování průchodů studentů e-learningovým kurzem Petriho sítěmi*, Sborník 6. ročníku konference o elektronické podpoře výuky SCO 2009., Brno: Masarykova univerzita, 2009, s. 117-122, ISBN 978-80-210-4878-2.
- [27] CHANG YC, CHU CP, *Applying learning behavioral Petri nets to the analysis of learning behavior in web-based learning environments*, Information Sciences, volume 180, issue 6, 2010, ps 995 – 1009, ISSN 0020-0255.
- [28] ZHANG BH, *Research on evaluation of e-learning modelling based on Petri nets*, 2008 INTERNATIONAL CONFERENCE ON ADVANCED COMPUTER THEORY AND ENGINEERING, IEEE COMPUTER SOC, 2008, ps. 699-703, ISBN 978-0-7695-3489-3.
- [29] HE F, LE J, *Hierarchical Petri-nets model for the design of e-learning system*, Edutainment 2007, LECTURE NOTES IN COMPUTER SCIENCE, volume 4469, ps. 283-292, ISSN 0303-9743.
- [30] DIAZ P, IGNACIO A, MORA E, *Evolutionary design of collaborative learning processes through reflective Petri nets*, 8th IEEE International Conference on Advanced Learning Technologies, IEEE COMPUTER SOC, 2008, ps. 423-427, ISBN 978-0-7695-3167-0.

- [31] HOARE Antony Richard Charles., *Communicating Sequential Processes*, (online at <http://www.usingscp.com/> in PDF format), Prentice Hall International Series in Computer Science, 1985, ISBN 0-13-153271-5 hardback or ISBN 0-13-153289-8 paperback.
- [31] MUDRÁK David, *Modular Object-Oriented Dynamic Learning Environment*, [online], červen 2005, <<http://moodle.cz/moodle/doc/>>.
- [32] ISO/IEC JTC1/SC7 3941 FCD, 24765 Systems and software engineering – Vocabulary, 2008, 298 ps, (pracovní materiál tvůrců norem ISO a IEC zpřístupněný školitelem).
- [33] VANÍČEK Jiří a kol., *Teoretické základy informatiky*, vyd.: Kernberg Publishing, s.r.o., Praha, 2007, 1. vyd., 431 s., ISBN 978-80-903962-4-1.

8. Přehled publikovaných prací uchazečky

8.1 Příspěvky ve sborníku

- [1a] VYNIKAROVÁ D., *Využití e-learningu při firemních školeních*, Firma a konkurenční prostředí, sborník prací, vyd.: PEF MZLU Brno, 2003, ISBN 80-7157-702-2.
- [2a] VYNIKAROVÁ D., *Trendy využití geografických informačních systémů*, Agrární perspektivy XII., sborník prací, vyd.: PEF ČZU Praha, 2003, ISBN 80-213-1056-1.
- [3a] VYNIKAROVÁ D., *Metodika tvorby elektronických výukových kurzů*, Doktorandský seminář, sborník prací, vyd.: PEF ČZU Praha, 2004, ISBN 80-213-1150-9.
- [4a] BUCHTELA, D., MERUNKA V., PAVLÍČEK J., PÍCKA M., VANÍČEK, J., VYNIKAROVÁ D., *Pokus o sondu studijních předpokladů pro informatiku*, Informatika XV., sborník prací, vyd.: PEF MZLU Brno, 2004, ISBN 80-7302-006-1.
- [5a] VYNIKAROVÁ D., *Problematika návrhu a zavádění elektronické výuky do praxe*, Agrární perspektivy XIII., sborník prací, vyd.: PEF ČZU Praha, 2004, ISBN 80-213-1190-8.
- [6a] VYNIKAROVÁ D., BUCHTELA D., *Analýza úspěšnosti výuky předmětu výpočetní systémy II*, Informatika XVI., sborník prací, vyd.: PEF MZLU Brno, 2005, ISBN 80-7302-083-1.
- [7a] VYNIKAROVÁ D., *Možnosti zpětné vazby v elektronických výukových kurzech*, Doktorandský seminář, sborník prací, vyd.: PEF ČZU Praha, 2005, ISBN 80-213-1314-5.

- [8a] VYNIKAROVÁ D., *Několik poznámek ke způsobům realizace zpětné vazby v e-learningu*, Agrární perspektivy XVII., sborník prací, vyd.: PEF ČZU Praha, 2005, ISBN 80-213-1372-2.
- [9a] VYNIKAROVÁ D., *Analýza technologií pro tvorbu e-learningových materiálů*, Agrární perspektivy XV., sborník prací, vyd.: PEF ČZU Praha, 2006, ISBN 80-213-1531-8.
- [10a] BUCHTELA D., VYNIKAROVÁ D., *Zkušenosti z používáním elektronického výukového systému (ELVYS)*, INFORMATIKA XVIII, sborník prací, vyd.: KONVOJ, spol. s r. o., Brno, 2006, ISBN 80-7302-111-0.
- [11a] VYNIKAROVÁ D., *Modelování průchodu studenta zpětnovazebními prvky elektronického výukového kurzu*, Agrární perspektivy XVI, sborník prací, vyd.: PEF ČZU Praha 2007, ISBN 978-80-213-1675-1.
- [12a] BUCHTELA D., VYNIKAROVÁ D., PAVLÍČEK, J., *Znalostní model průchodu studenta výukou*, Agrární perspektivy XVII, sborník prací, vyd.: PEF ČZU Praha 2008, ISBN 978-80-213-1813-7.
- [13a] PAVLÍČEK, J., BUCHTELA, D., VYNIKAROVÁ, D., *Inteligentní vyhledávač Webových služeb*, Agrární perspektivy XVII, sborník prací, vyd.: PEF ČZU Praha 2008, ISBN 978-80-213-1813
- [14a] BUCHTELA D., VYNIKAROVÁ D., *Standardized Knowledge and Data Interface Model of Branch Guidelines*, Agrární perspektivy XVIII, sborník prací, vyd.: PEF ČZU Praha 2009, ISBN 978-80-213-1965-3.
- [15a] BUCHTELA, D., MACH, J., VYNIKAROVÁ D., *Integrace systému odevzdej.cz a Moodle ČZU*, Informatika XXIII., sborník abstraktů, vyd.: Mendelova univerzita v Brně, 2010. s. 17. ISBN 978-80-7375-394-8.
- [16a] BUCHTELA, D., VYNIKAROVÁ D., *E-learningový systém Moodle ČZU*. Trendy a inovace informačních systémů vysokých škol., Sborník příspěvků, vyd: Západočeská univerzita v Plzni, 2010. s. 5 -9. ISBN 978-80-7043-892-3.

8.2 Publikace v recenzovaném časopise

- [17a] VYNIKAROVÁ D., *Using generalized Petri nets for e-learning system modeling*, Scientia Agriculturae Bohemica, 2008, No. 39, ps. 102 – 107, ISSN 1211-3174.

8.3 Kapitola v monografii

- [18a] BUCHTELA D., VESELÝ A., VYNIKAROVÁ D., *Guideline Knowledge Representation Model (GLIKREM) in Knowledge Management and Modern Information Technologies*, 1. vyd., Praha: nakladatelství Alfa, 2010, s. 26 – 41, ISBN 978-80-87197-31-8.

8.4 Skripta

- [19a] BUCHTELA, D., VYNIKAROVÁ, D., *Výpočetní systémy – cvičení*, vyd.: PEF ČZU Praha, 2004, 202 s., ISBN 80-213-1170-3.
- [20a] BUCHTELA, D., VYNIKAROVÁ, D., *Cvičebnice z předmětu Výpočetní systémy*, vyd.: PEF ČZU Praha, 2009, 123 s., ISBN 978-80-213-2014-8.
- [21a] BUCHTELA, D., VYNIKAROVÁ, D., *Cvičebnice z předmětu Architektura počítačů*, vyd.: PEF ČZU Praha, 2010, 83 s., ISBN 978-80-213-2073-4.

9. Seznam použitých obrázků a tabulek

Obr. 1 – Struktura výukového systému (převzato z [6], [7])	27 -
Obr. 2 – Struktura systému s lineárním základem (převzato z [6], [7])	29 -
Obr. 3 – Struktura systému se stromovým základem (převzato z [6], [7])	29 -
Obr. 4 – Struktura systému se síťovým základem (převzato z [6], [7])	30 -
Obr. 5 – Chomského hierarchie tříd jazyků	33 -
Tab. 1 – stavy deterministického konečného automatu	33 -
Obr. 6 – Znázornění konečného automatu se stavy S_0 , S_1 a S_2	34 -
Obr. 7 – Deterministický Turingův stroj (převzato z [23])	40 -
Obr. 8 – Prvky Petriho sítě	43 -
Obr. 9 – Příklady aktivace Petriho sítě [11]	44 -
Obr. 10 – Příklad konfliktu mezi přechody T1 a T2 [11]	45 -
Obr. 11 – Příklad pro funkci jedné proměnné	50 -
Obr. 12 – Příklad pro funkci dvou proměnných	50 -
Obr. 13 – Příklad pro funkci s alternativním nedeterministickým charakterem	50 -
Obr. 14 – Model přechodu výukového systému [převzato z 6]	54 -
Obr. 15 – GLIF model průchodu studenta vyučovaným předmětem [12a]	61 -
Obr. 16 – Příklad struktury kurzu (převzato z [27])	63 -
Obr. 17 – Konceptuální kostra e-learningového systému (převzato z [29])	66 -
Obr. 18 – Příklad sítě složené ze sedmi vzájemně provázaných témat	79 -
Obr. 19 – Příklad kroku prvního typu ve zobecněné Petriho síti	79 -
Obr. 20 – Obarvení zobecněné Petriho sítě při neúspěchu studenta při testu P4	80 -
Obr. 21 – Obarvení zobecněné Petriho sítě při neúspěchu studenta při testu P2	80 -
Obr. 22 – Obarvení zobecněné Petriho sítě při úspěchu studenta při testu P1	81 -
Obr. 23 – Obarvení zobecněné Petriho sítě při úspěchu studenta při testu P2	81 -
Obr. 24 – Obarvení zobecněné Petriho sítě při úspěchu studenta při testu P4	82 -
Obr. 25 – Obarvení zobecněné Petriho sítě při úspěšném průchodu studenta kurzem	82 -
Obr. 26 – Síť zobrazující příklad konstrukce kombinačního logického obvodu pomocí tranzistorů	88 -
Obr. 27 – Síť zobrazující neúspěch při testu v místě P15	89 -
Obr. 28 – Síť zobrazující neúspěch studenta při testu v místě P14	90 -
Obr. 29 – Síť zobrazující neúspěch studenta při testu v místě P13	91 -
Obr. 30 – Síť zobrazující neúspěch studenta při testu v místě P12	92 -
Obr. 31 – Síť zobrazující neúspěch studenta při testu v místě P11	93 -
Obr. 32 – Síť zobrazující úspěšné absolvování testu v místě P10	94 -

Obr. 33 – Síť zobrazující úspěšné absolvování testu v místě P11	- 95 -
Obr. 34 – Síť zobrazující úspěšné absolvování testu v místě P12	- 96 -
Obr. 35 – Síť zobrazující úspěšné absolvování testu v místě P6, P9 a P13	- 97 -
Obr. 36 – Síť zobrazující úspěšné absolvování testu v místě P8 a P14	- 98 -
Obr. 37 – Síť zobrazující úspěšné zvládnutí problematiky konstrukce kombinačních logických obvodů pomocí tranzistorů	- 99 -
Obr. 38 – Síť znázorňující příklad vyjádření reálného čísla ve formátu IEEE 754	- 103 -
Obr. 39 – Síť znázorňující neúspěch studenta při testu v místě P9	- 104 -
Obr. 40 – Síť znázorňující neúspěch studenta při testu v místě P8 a úspěch v testu v místě P3	- 105 -
Obr. 41 – Síť znázorňující neúspěch studenta při testu v místě P7 a úspěchu v místě P6	- 106 -
Obr. 42 – Síť znázorňující neúspěch studenta při testu v místě P5	- 107 -
Obr. 43 – Síť znázorňující úspěch studenta při testu v místě P4	- 108 -
Obr. 44 – Síť znázorňující úspěch studenta při testu v místě P5	- 109 -
Obr. 45 – Síť znázorňující úspěch studenta při testu v místě P7	- 110 -
Obr. 46 – Síť znázorňující úspěch studenta při testu v místě P8	- 111 -
Obr. 47 – Síť znázorňující úspěšné nastudování problematiky převodu reálného čísla do 32-bitového formátu IEEE 754	- 112 -

10. Rejstřík použitých pojmů a zkratek

V této kapitole jsou vysvětleny některé odborné termíny a zkratky, které jsou v předkládané disertační práci užívány. Pokud v této kapitole některý termín obsažen není či mohou vzniknout pochybnosti o tom, v jakém smyslu je termín využíván, je tento použit ve smyslu uvedeném v [32].

Bipartitní graf – dle [33] graf, kde množina jeho vrcholů V je disjunktním sjednocením množin R a S (R a S tvoří rozklad V), přičemž všechny jeho hrany mají jeden vrchol v R a druhý v S . U orientovaných bipartitních grafů navíc požadujeme, aby všechny hrany byly orientovány od vrcholů v R k vrcholům v S , nebo všechny opačně.

Computer-based learning (CBL) – dle [1] označuje vzdělávání podporované počítačem. Kurz nebo studijní materiál, který lze samostatně spustit na počítači. Na rozdíl od Web-based learning nevyžaduje CBT, aby počítač byl zapojen do internetové sítě a neposkytuje odkazy na vzdělávací prostředky mimo kurz.

E-learning – dle [1] výraz e-learning vznikl propojením slovního základu „learning“ (učení) a předpony „e“ (elektronické), tedy e-learning lze pojmut jako elektronické učení či výuku, přesněji jako elektronickou podporu výuky. E-learning vznikl propojením procesu vzdělávání s informačními a komunikačními technikami.

Hodnocení studenta – měření do jaké míry student zvládl danou látku. Výsledkem je míra vyjádření zpravidla v ordinální stupnici, kde jednotlivé úrovně jsou dány čísly či slovně.

Chomského hierarchie gramatik a jazyků – je hierarchie tříd formálních gramatik generujících formální jazyky. Byla vytvořena Noamem Chomským v roce 1956. Chomského hierarchie se skládá z následujících tříd:

- *Gramatiky typu 0* – zahrnují v sobě všechny formální gramatiky, generují právě ty jazyky, které mohou být rozpoznány nějakým Turingovým strojem.

- *Gramatiky typu 1* (kontextové gramatiky) – generují kontextové jazyky. Tyto jazyky jsou právě jazyky rozpoznatelné lineárně ohraničeným Turingovým strojem.
- *Gramatiky typu 2* (bezkontextové gramatiky) – generují bezkontextové jazyky. Tyto jazyky jsou právě jazyky rozhodnutelné nějakým nedeterministickým zásobníkovým automatem.
- *Gramatiky typu 3* (regulární gramatiky) – generují regulární jazyky. Tyto jazyky jsou právě jazyky rozhodnutelné konečným automatem.

Jazyk – libovolná podmnožina množiny všech slov nad abecedou daného jazyka.

Learning Management System (LMS) – dle [1] řídicí výukový systém (systém pro řízení výuky), tedy aplikace řešící administrativu a organizaci výuky v rámci e-learningu. LMS jsou aplikace, které v sobě integrují zpravidla nejrůznější on-line nástroje pro komunikaci a řízení studia (nástěnka, diskusní fórum, chat, tabule, evidence ad.) a zároveň zpřístupňují studentům učební materiály či výukový obsah on-line nebo i off-line. LMS aplikací je řada, od jednoduchých typů přes LMS z akademické sféry až po rozsáhlé a složité komerční aplikace (Adobe Connect, Fronter, Blackboard). Řada LMS je šířených i jako free nebo open source software (například systém Moodle).

Last In – First Out (LIFO) – způsob manipulace s daty charakteristický pro zásobník, data uložena jako poslední budou čtena jako první. Pro manipulaci s uloženými datovými položkami se udržuje tzv. *ukazatel zásobníku*, který udává relativní adresu poslední přidané položky, tzv. *vrchol zásobníku*.

Model – struktura popisující významné vlastnosti vybrané entity či procesu formálními prostředky.

Moodle („*Modular Object-Oriented Dynamic Learning Environment*“) – dle [31] modulové objektivě orientované dynamické vzdělávací prostředí, je softwarové prostředek, určený pro prezenční i distanční výuku prostřednictvím online kurzů, dostupných na www stránkách. Moodle je volně šiřitelný software s otevřeným kódem,

který pracuje pod operačními systémy Unix, Linux, Windows, Mac OS a na jakémkoliv systému, který podporuje technologii PHP. Data jsou ukládána v jediné databázi, např. MySQL, Access, ODBC, Oracle, apod.

Multigraf – dle [33] graf, který není prostý. Pokud u orientovaného grafu „zrušíme orientaci hran“, to je nahradíme uspořádanou dvojicí vrcholů (a, b) množinou $\{a, b\}$, získáme z orientovaného grafu graf neorientovaný. Této operaci říkáme symetrizace grafu. Z prostého orientovaného grafu můžeme ovšem získat neorientovaný graf, který již prostý není, tedy multigraf.

Navigace studenta – návod či doporučení jak uspořádat jednotlivé fáze výukového procesu.

Neorientovaný graf – dle [33] určen množinou V všech vrcholů grafu, množinou H všech hran a incidenčním zobrazením φ , které však nyní přiřazuje hraně dva vrcholy nebo jediný vrchol, které hrana spojuje. Prvek množiny V se nazývá vrchol [vertex] či uzel [node], prvek množiny H hrana grafu. Zobrazení φ je tedy nyní zobrazení množiny H všech hran do množiny $V_1 \cup V_2$, kde V_1 je množina všech jednoprvkových a V_2 všech dvouprvkových podmnožin V . Toto zobrazení přiřazuje každé hraně dva vrcholy, které hrana spojuje nebo jediný vrchol v případě, že jde o smyčku v tomto vrcholu. Prvkům množiny $V_1 \cup V_2$ budeme pro zjednodušení říkat dvojice vrcholů.

Neorientovaný strom – neorientovaný souvislý graf bez cyklů. Lze charakterizovat jako souvislý graf, u kterého počet hran je o 1 nižší než počet uzlů.

Ordinalní měřicí stupnice – dle [33] představuje typ měření, kde grupu přípustných transformací tvoří všechny funkce typu $y = f(x)$, kde f je libovolná rostoucí funkce. Takové měření nepotřebuje zvažovat operaci skládání na empirické struktuře a z čísel vůbec nevyužívá aritmetiku. Využívá pouze toho, že čísla tvoří uspořádanou množinu. Typickým příkladem je klasifikace výsledků zkoušky nebo jakékoliv hodnocení typu byl jsem: („nadšen“, velmi spokojen“, „spokojen“, „v podstatě spokojen“, „vážám s hodnocením“, „mám výhrady“, „nespokojen“, „velmi nespokojen“, „znechucen“). U

měření ordinálního typu nejsou smysluplné výsledky aritmetických výpočtů (tedy ani aritmetický průměr), interpretovat v empirické struktuře lze pouze to, co závisí jen na pořadí. Tedy maxima, minima, medián.

Orientovaný graf – dle [33] zadán dvěma konečnými množinami. Množinou vrcholů (neboli uzlů) [vertex or node] a množinou orientovaných hran (neboli šipek). Dále pak pravidlem, které stanoví odkud kam, tedy z kterého vrcholu do kterého daná orientovaná hrana směřuje. Tím je určen tak zvaný počáteční vrchol orientované hrany a koncový vrchol orientované hrany. Matematicky lze tuto konstrukci formalizovat tak, že orientovaný graf definujeme jako uspořádanou trojici (V, H, φ) , kde φ je zobrazení H do $V \times V$. Takové zobrazení φ se nazývá incidenční zobrazení v orientovaném grafu.

Orientovaný (kořenový) strom – vznikne z neorientovaného stromu vybráním nějakého vrcholu jako kořene a orientací všech hran stromu od kořene postupně k dalším uzlům.

Petriho síť s prioritami – je P/T Petriho síť, ve které je každému přechodu přiřazeno celé nezáporné číslo udávající prioritu přechodu. Priority přechodů upravují pravidla pro jejich provádění. V Petriho síti s prioritami je přechod t povolen, je-li proveditelný v odpovídající P/T síti bez priorit.

Petriho síť barevná – je P/T Petriho síť, ve které je možno pracovat s více typy značek („značkami různých barev“) a každému místu je přiřazena třída značek, která se může v daném místě nacházet (místu nemusí být přiřazen pouze jeden typ značek, ale i více).

Prostý neorientovaný graf – neorientovaný graf, u kterého libovolné dva vrcholy spojuje nejvýše jedna (jedna nebo žádná) hrana

Prostý orientovaný graf – je orientovaný graf, u kterého libovolnou uspořádanou dvojici vrcholů spojuje nejvýše jedna (jedna nebo žádná) hrana. Prostému orientovanému grafu jednoznačně odpovídá binární relace na množině jeho vrcholů.

Slovo (formálního jazyka) – konečný řetězec symbolů jeho abecedy.

Testování – je metoda sloužící k verifikaci nebo falzifikaci určité hypotézy. V případě testování studenta k verifikaci či falzifikaci tvrzení, že student zvládl látku na určité úrovni. Spočívá v zadávání otázek či úkolů a vyhodnocování odpovědí.

Web-based learning (WBL) – dle [1] znamená vzdělávání na webu. Přístup ke vzdělávacímu obsahu se realizuje prostřednictvím webového prohlížeče.

Zásobník (stack) – v informatice obecná datová struktura (tzv. abstraktní datový typ) používaná pro dočasné ukládání dat. Pro zásobník je charakteristický způsob manipulace s daty - data uložena jako poslední budou čtena jako první. Proto se používá také výraz *LIFO (Last In – First Out)*.

Zpětná vazba – je prvek řízení systému spočívající v tom, že systému ovlivňuje jeho vstup a tím stav systému.