SBORNÍK PRACÍ FILOZOFICKÉ FAKULTY BRNĚNSKÉ UNIVERZITY
STUDIA MINORA FACULTATIS PHILOSOPHICAE UNIVERSITATIS BRUNENSIS
A 30, 1982

KAREL PALA

O PROCEDURÁLNÍ GRAMATICE (PRO ČEŠTINU)

1. Úvod

Tento článek představuje výsledky výzkumu, jehož jedním cílem bylo navrhnout a vypracovat formální gramatiku češtiny vhodnou pro počítačové aplikace (např. dialog s bází dat v podmnožině přirozeného jazyka).

Oddíl 1 je úvodní, oddíl 2 obsahuje stručnou charakteristiku principů, na nichž je založena procedurální gramatika, dále základní množinu nekontextových pravidel a v závěru objasnění vztahů mezi skupinami nekontextových pravidel a jim odpovídajícími procedurami.

V oddíle 3 najdeme popis struktury slovníku pro procedurální gramatiku. Obsahem oddílu 4 je definice procedurální gramatiky pro vymezenou podmnožinu češtiny a dále popis jednotlivých procedurálních pravidel tvořících procedurální gramatiku a jí odpovídající syntaktický analyzátor.

V oddíle 5 je věnována pozornost stromové struktuře věty, která je výstupem z procedurální gramatiky. V oddíle 5 se na příkladě ukazuje průběh analýzy jedné české věty až ke konečnému výsledku.

Závěrečný oddíl 7 poskytuje stručnou informaci o počítačové implementaci popisované procedurální gramatiky v jazyce LISP 1.5 (na počítači TESLA 200).

Při budování formální gramatiky přirozeného jazyka, která má být vhodná pro počítačové aplikace, je žádoucí splnit některé vybrané požadavky:

1. Gramatika nesmí být příliš rozsáhlá, má-li být zvládnutelná v rámci současných počítačových systémů;
2. Gramatika by měla být použitelná jak pro generativní, tak pro rekognoskativní procedury;
3. Gramatika musí být přehledná, čitelná a snadno manipulovatelná;
4. Gramatika má být relativně snadno programovatelná.

Aby nedošlo k nedorozumění, budeme zde gramatikou rozumět množinu formálních pravidel popisujících syntaktické struktury daného přirozeného jazyka (češtiny). Při širším pojetí lze ovšem gramatikou rozumět soubor pravidel postihujících morfologii, syntax i sémantiku daného přirozeného jazyka. Zde se však morfologií zabývat nebudeme a sémantika je předmětem pozornosti jinde [6].

Chceme-li vybudovat konkrétní formální gramatiku jazyka, jako je čeština, musíme zvolit vhodný typ formálního aparátu. V tomto směru existuje řada možností:

(i) nekontextová pravidla (tvaru $A \rightarrow \omega$)
(ii) kontextová pravidla (tvaru $\alpha A \beta \rightarrow \alpha \omega \beta$)
(iii) závislostní pravidla (např. tvaru $\langle x_i, y_j \rangle$)
(iv) transformační pravidla (viz např. [1])
(v) jiné druhy aparátů, viz např. [2] nebo [5]

Jak ukazuje dosavadní praxe, body (i) a (iii) jsou samy o sobě nepřijatelné jako příliš jednoduché. Všechny zajímavější a úspěšné gramatiky představují kombinace bodů (i) nebo (iii) se silnějšími typy pravidel, např. s pravidly kontextovými nebo transformačními.

Pro angličtinu byly v poslední době navrženy dva typy úspěšných a v praxi nyní běžně používaných gramatik. Gramatiky založené na tzv. zesílených přechodových sítích (ATN grammars), jež jsou modifikacemi konečných automatů, navrhl Woods [5]. Naproti tomu Winograd pracuje s nekontextovými pravidly formulovanými jako procedury a programově realizovanými jako funkce v programovacím jazyce LISP 1.5.

Berouce v úvahu rozdíly mezi angličtinou a češtinou, jež se týkají zejména slovosledu a pádů, rozhodli jsme se pro řešení Winogradovo, tj. pro vybudování procedurální gramatiky češtiny.

2. Charakteristika procedurální gramatiky

Procedurální gramatika pro češtinu může být charakterizována ve dvou krocích:

1. Nejprve uvedeme základní množinu nekontextových pravidel. Ta umožňují postihnout syntaktické struktury češtiny, nedostačují však k zachycení kontextových a dalších složitějších souvislostí mezi větnými složkami;

2. Nekontextová pravidla jsou rozdělena do skupin a ty jsou přeformulovány jako procedury. Jako výsledek pak dostáváme soubor procedurálních pravidel, který tvoří to, co budeme dále nazývat procedurální gramatikou.

Dříve než přistoupíme k uvedení základní množiny nekontextových pravidel, pokládáme za vhodné objasnit, co zde rozumíme procedurou. Chápeme ji jako posloupnost kroků, z nichž některé jsou příkazy předpisující provedení určité akce a některé jsou podmíněné příkazy, které předpisují provedení konkrétních testů, jejichž výsledky rozhodují o tom, který krok (příkaz) má být proveden jako následující.

Jako příklad příkazu uveďme: „Vyhledej slovo ve slovníku" nebo „Připoj uzel ke stromu analyzované věty". Příkladem podmíněného příkazu je např. příkaz: „Je slovní druh daného slova adverbium nebo substantivum?" nebo „Je dané substantivum v nominativu?" apod.

Toto chápání procedur je blízké tomu, co se obvykle chápe jako program. Rozdíl lze vidět jen v tom, že příkazy v procedurách mohou být dosti složité, kdežto v odpovídajícím programu bychom je vyjádřili jako posloupnost jednoduchých a často strojově závislých instrukcí.

2.1 Základní množina nekontextových pravidel

Na tuto množinu lze pohlížet jako na následující nekontextovou gramatiku:

$$G = (V_T, V_N, R, VH),$$

kde V_N je množina neterminálních symbolů, V_T je proměnlivá množina terminálních symbolů — českých slov, R je množina přepisovacích pravidel tvaru $A \rightarrow \omega$, přičemž $A \in V_N$ a ω je neprázdný řetěz symbolů patřících do $V_N \cup V_T$. Konečně VH je vyznačený počáteční symbol patřící do V_N. Množina V_N obsahuje následující symboly:
V_N = {VH, VV, NG, NGGEN, PRONG, VG, VGB, PREPG, ADG, PROND, PRONP, PRONUN, PRONPER, PRONQ, PRONR, NUMO, NUMK, AD, A, N, V, VB, VI, VMI, VL, VML, VBL, VM, VPAS, VBC, VREF, PRTREF, NPR}.

Symboly z V_N mohou být interpretovány následovně:

1. věty — VH, jednoduchá věta, hlavní
   VV, jednoduchá věta, vedlejší
2. skupiny (složky) — NG, substantivní skupina libovolné složitosti v kterémkoli pádě a čísle (tj. v NOM, DAT, AK, LOK, INST, a to v SG nebo PL, mimo GEN)

NGGEN, substantivní skupina v genitivu
PREPG, předložková skupina v kterémkoli pádě a čísle (tj. v GEN, DAT, AK, LOK, INST)
PRONG, pronominální substantivní skupiny, tzv. nevyjádřený podmět
VG, slovesná skupina
VGB, slovesná skupina obsahující sponu „být" a skupinu NG
ADG, adverbiální skupina

3. slovní druhy — PROND, ukazovací zájmeno

PRONP, posesívní zájmeno
PRONPER, osobní zájmeno
PRONUN, neurčité zájmeno
PRONQ, tázací zájmeno (slovo)
PRONR, vztažné zájmeno
NUMO, řadová číslovka
NUMK, číslovka základní
AD, příslovce místa, času nebo způsobu
A, adjektivum
N, substantivum
NPR, substantivum — vlastní jméno
V, sloveso v určitém tvaru (v přítomném nebo budoucím čase)
VB, sloveso být v určitém tvaru
VBL, minulé participium slovesa být
VBC, kondicionálová forma slovesa být
VREF, určitý tvar reflexiva tantum
VI, infinitiv
VL, minulé participium slovesa
VPAS, trpné participium slovesa
VM, modální sloveso v určitém tvaru
VMI, infinitiv modálního slovesa

VML, minulé participium modálního slovesa
VF, určitý tvar fázového slovesa
VFI, infinitiv fázového slovesa
PREP, předložka
CON, spojka souřadicí
CONV, spojka podřadicí

Množina V_T může být charakterizována jako měnitelná množina českých slovních tvarů tvořících slovník, jehož podoba bude specifikována dále.

Množina R obsahuje následující nekontextová pravidla:

1. VH → {(ADG) (PREPG) (NG) (ADG) VG (PREPG) (NGGEN) (NG) (ADG) (NG)
   (ADG) (PREPG) (NG) (ADG) VGB (ADG)}
2. NG → {(PROND) (PRONP) (NUMO) (NUMK) (AD) (A) (N) (NPR)
   PRONUN
   PRONPER
   PRONQ
   PRONR
   NG NGGEN
   NG PREPG}
3. PREPG → {PREP NG}
4. NGGEN → {NG (S RYSEM GEN)}
5. VG → {V
   VREF PRTREF
   VB VMI VI
   VB VML VI
   VB VBL VPAS
   VB VL
   VB VPAS
   VBC VBL VPAS
   VBC VML VI
   VBC VBL
   VBC VL
   VBL VB VPAS
   VBL VBC VPAS
   VL VB
   VL VBC
   VPAS VB
   VPAS VM VB
   VPAS VML VB
   VI VBC VML
   VI VM
   VI VML
   VML VB VI
   VML VBC VI
   VM
   VML VI
   VM VI
   VML}
6. VGB → {VB
   VB VBL
   VBC VBL
   VBL VBC
   VBL VB
   VGB NG}
7. ADG → {AD
   AD_{QUA} AD}

Je vidět, že gramatické kategorie jako pád, rod jmenný, číslo, osoba, čas, způsob, vid, rod slovesný aj. nejsou obsaženy ani ve V_N, ani v R. Má-li gramatika být co nejjednodušší, musí tomu tak být i za cenu nepříjemných důsledků spočívajících v tom, že gramatika G generuje substantivní nebo předložkové skupiny nebo i celé věty bez ohledu na gramatickou shodu. Je však třeba najít řešení, které zachová jednoduchost nekontextové gramatiky G, ale zbaví nás nepříjemných důsledků, jež jsou s tím spojeny. Toto řešení vypadá následovně:

(i) použijeme nekontextového rámce, jak byl uveden výše, protože nám dává jednoduchý a průzračný popis základních syntaktických struktur češtiny v termínech slovních druhů a skupin;

(ii) tato nekontextová pravidla přeformulujeme a dáme jim podobu nezávislých procedur, které mohou řídit své vlastní chování, dovedou nacházet strukturu větných složek nebo celých vět s ohledem na kontext a shodu a testovat gramatickou správnost složek i vět.

Nyní je třeba zodpovědět otázku, jak začlenit informace o gramatických kategoriích do procedur. Tyto informace jsou potřebné pro vybudování syntaktické, sémantické a pragmatické struktury věty.

Dřívější pokusy začlenit gramatické kategorie do nekontextových pravidel spočívaly v tom, že ke kategoriálním symbolům se přidávaly speciální indexy, tj. zaváděly se komplexní symboly jako $N_{\text{NOMSGMASK}}$ nebo $V_{\text{3SGPUTINDACT}}$ apod. Tyto symboly jsou však vskutku příliš složité a obvykle nečlenitelné, takže se s nimi obtížně manipuluje. Nadto způsobují zvětšení počtu nekontextových pravidel v gramatice až o 40 %.

Lepší východisko spočívá v tom, že se skupiny nekontextových pravidel formulují jako samostatné procedury, které mohou dělat následující věci:

1. dovedou rozpoznat, z jakých konkrétních slovních druhů se skládají větné složky (a tudíž celá věta);

2. dovedou rozpoznat gramatické kategorie (zde je budeme dále nazývat rysy) jednotlivých slov a rovněž jejich slovní druhy;

3. s použitím rysů dovedou kontrolovat a zjistit gramatickou shodu a další kontextové závislosti v rámci věty a větných složek;

4. pro rozpoznané větné složky (nebo celé věty) dovedou vybudovat stromové struktury (grafy-stromy) a tak zachytit jejich syntaktickou strukturu.

Procedury tohoto druhu budeme nazývat procedurální pravidla (dále P-pravidla). Skládají se obvykle ze dvou částí:

(i) z části gramatické, která obsahuje příslušná gramatická pravidla v podobě nekontextových pravidel,

(ii) z části tvořené souborem pomocných operací, které dovedou vyhledávat potřebné rysy (viz výše bod (2)), budovat příslušné stromové struktury (bod (4)), pohybovat se uvnitř grafu-stromu, testovat specifické podmínky, které mohou být kladeny na jednotlivé elementy stromových struktur, a řídit chování celého P-pravidla.

Např. P-pravidlo, které bude odpovídat již uvedenému nekontextovému pravidlu (2), lze charakterizovat jako proceduru, která na vstupu přijímá část analyzované věty, vyhledá ve slovníku její jednotlivá slova a přiřadí jim jejich slovní druhy a též příslušné gramatické rysy. Je-li třeba, testuje gramatickou shodu uvnitř analyzované složky tím, že provádí průnik trojic tvořených

rysy „pád-číslo-rod jmenný". Jestliže výsledek testu je negativní, analyzovaná složka je hodnocena jako gramaticky nesprávná, P-pravidlo to oznámí na výstupu a skončí svou činnost. V pozitivním případě P-pravidlo pokračuje v činnosti a buduje příslušnou stromovou strukturu. Poté ještě zjistí, zda má opět volat sebe samo nebo nějaké další P-pravidlo, jemuž v tom případě předá i řízení.

3. Slovník

Lze říci, že jeden z hlavních rozdílů mezi nekontextovými pravidly a P-pravidly spočívá v tom, jak definují slovník.

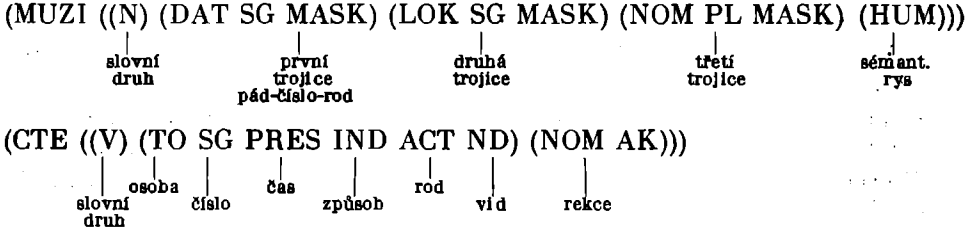
Jak jsme už naznačili, v nekontextové gramatice je slovník terminálních symbolů tvořen množinou V_T a lze jej získat jako podmnožinu těch pravidel z R, která mají formu např.

$N_{1\,sgmask} \rightarrow$ {učitel, muž, plavec, ...} nebo
$V_{3\,sgpresind} \rightarrow$ {čte, spí, zpívá, ...},

tj. jsou to ta pravidla v R, která mají na pravých stranách jen terminální symboly a na levých stranách jen kategoriální symboly, které jsou, jak patrno, symboly komplexními. Protože komplexní symboly nejsou dále analyzovatelné v rámci množiny R, není možné testovat a zpracovávat jednotlivé gramatické rysy a tím řídit průběh syntaktické analýzy.

V rámci P-pravidel je možná jiná, výhodnější organizace slovníku. Především slovník není součástí žádného z P-pravidel a je definován nezávisle na nich, a to jako hierarchický seznam. Protože slovník je nezávislý na jednotlivých P-pravidlech, lze věci zařídit tak, že každé pravidlo má přímý přístup ke slovníku a může z něho získávat libovolné rysy libovolného slova.

Na příkladu ukážeme typické heslo pro substantivum a pro sloveso:



Je vidět, že uvedená hesla obsahují všechny gramatické rysy dané slovoformy. Substantivum muži je homonymní, protože může odpovídat třem pádům DATivu, LOKálu (SG) a NOMinativu (PL). Je úlohou příslušného P-pravidla rozhodnout, o který pád v daném kontextu jde (často je třeba brát v úvahu kontext v rámci celé věty). Uvedená hierarchická organizace slovníkového hesla se vyznačuje následujícími vlastnostmi:

(i) tvar hesla je velmi blízký předpokládanému výstupu z morfologického analyzátoru;

(ii) heslo lze doplnit o libovolné údaje týkající se slova, jež heslo obsahuje. Mohou to být gramatické, sémantické nebo i pragmatické rysy a mohou být výhodně užívány jednotlivými P-pravidly;

(iii) údaje obsažené v heslu mohou být dále strukturovány v závislosti na použitém systému rysů;
(iv) heslo budované tímto způsobem je snadno programovatelné v jazycích typu LISP.

4. Definice procedurální gramatiky pro češtinu

Můžeme nyní přistoupit k definování procedurální gramatiky češtiny.

DEF. 1: Procedurální gramatika (dále P-gramatika) je množina P-pravidel, kde každé P-pravidlo má tvar A(X, Y, ..., Z). A je jméno P-pravidla a symboly X, Y, ..., Z označují skupiny (větné složky), které lze pravidlem analyzovat.

Poznámka: V našem případě jsou P-pravidla programově realizována jako funkce v programovacím jazyce LISP 1.5. Z toho plyne, že P-pravidla lze chápat jako jména funkcí + seznamy jejich argumentů. Jsou tudíž možné, jak uvidíme dále, i funkce bez argumentů.

Konkrétně pro češtinu:

DEF. 2: $PG_č$ = {ANAL(NG PREPG NGGEN), ANALVG( ), ANALAD( ), ANALVETA(ZDROJ)}.

P-gramatika pro češtinu je tedy tvořena čtyřmi pravidly, která lze charakterizovat následovně:

ANAL(NG PREPG NGGEN) je P-pravidlo založené na výše uvedených nekontextových pravidlech (2), (3) a (4) a analyzuje české substantivní a předložkové skupiny ve všech pádech a číslech a testuje shodu uvnitř těchto skupin.

ANALVG( ) je P-pravidlo založené na nekontextových pravidlech (5) a (6) a analyzuje české slovesné skupiny tvořené nejvýše třemi slovesnými složkami. Je též schopno analyzovat slovesné skupiny obsahující tzv. verbonominální predikát (= VGB).

ANALAD( ) je P-pravidlo založené na nekontextovém pravidle (7) a analyzuje české adverbiální skupiny času, místa a způsobu, pokud jsou tvořeny adverbii.

ANALVETA(ZDROJ) je řídicí P-pravidlo založené na nekontextovém pravidle (1). Dovede hlídat a usměrňovat činnost ostatních P-pravidel, dokončuje průběh analýzy a buduje stromovou strukturu analyzované věty.

4.1 P-pravidlo ANAL

Ukázali jsme již, že gramatický rámec tohoto pravidla je založen na nekontextových pravidlech (2), (3), (4). Tato pravidla lze charakterizovat jako maximální v tom smyslu, že obsahují všechny možné složky, z nichž se může nějaká substantivní nebo předložková skupina skládat. P-pravidlo ANAL však analyzuje jakékoli substantivní či předložkové skupiny (tj. maximální i minimální — v jiné terminologii eliptické).

Předložkové skupiny jsou analyzovány stejně jako substantivní s výjimkou

předložky, která ve většině případů napomáhá jednoznačně identifikovat pád analyzované předložkové skupiny. Shoda je v předložkových i substantivních skupinách testována stejným způsobem — pravidlo ANAL obsahuje podproceduru SHODNOST, která toto zajišťuje.

Na příkladě ukážeme způsob práce P-pravidla ANAL. Mějme následující českou substantivní skupinu

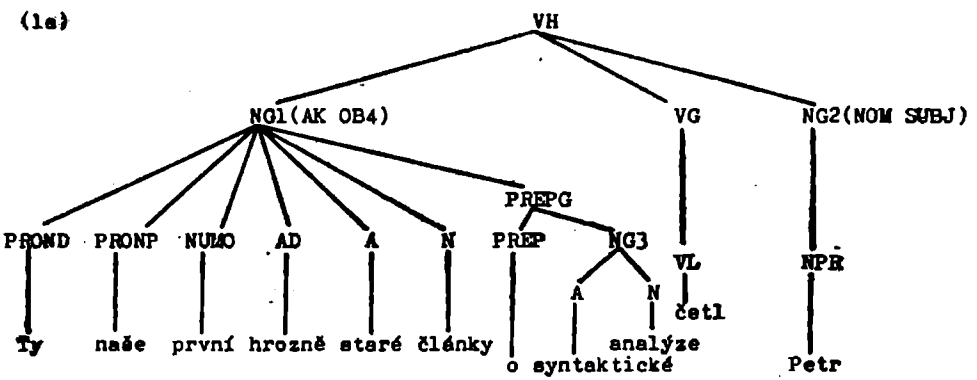
(1) ty naše první hrozně staré články o syntaktické analýze

Pravidlo ANAL začíná svou činnost testováním slovních druhů slov vyskytujících se ve skupině. Jestliže se slovní druhy nalezené ve skupině shodují (i co do pořadí) s tím, co požaduje nekontextové pravidlo (2) zabudované v P-pravidle ANAL, činí se pokus stanovit pád analyzované skupiny a testuje se gramatická shoda uvnitř skupiny. K úspěšnému určení pádu analyzované skupiny je však často třeba přihlížet ke kontextu v rámci celé věty a též testovat pořadí jednotlivých substantivních skupin ve větě (zejména v případě homonymie NOM-AK). Je-li analyzovaná skupina (= NG) první ve větě a poskytuje-li možnost jak NOM tak AK, vybírá se první možnost a NG je charakterizována jako SUBJ věty. Je-li v pořadí až druhou NG ve větě, vybírá se AK a tato NG je charakterizována rysem OBJ, ovšem pád první NG se znovu testuje, abychom se vyhnuli chybné analýze. Obecně lze strategii zabudovanou v pravidle ANAL formulovat takto: jsou-li ve větě dvě NG a obě mohou být NOM nebo AK a obě mají rys NEŽ (neživotnost), pak první je brána jako NOM a dostává rys SUBJ. Druhá je pak AK a OBJ. Má-li jedna z NG rys HUM (životnost) nebo je-li tvořena osobním zájmenem, je situace jasná a pád této NG lze stanovit jednoznačně.

V našem případě předpokládejme, že (1) je první ve větě a že druhá relevantní NG má rys HUM jako ve větě

(2) Ty naše první hrozně staré články o syntaktické analýze četl Petr.

P-pravidlo ANAL bere tedy skupinu (1) jako NOM a tedy i SUBJ, ale později zjistí, že druhá NG tvořená slovem Petr je NPR (vlastní jméno) a tedy HUM. To vede k opravě a skupina (1) je definitivně analyzována jako AK a OBJ4. Poté se uvnitř (1) testuje gramatická shoda a v pozitivním případě je strom skupiny (1) vybudovaný v průběhu analýzy připojen ke stromové struktuře celé věty. Výsledek činnosti pravidla ANAL má tedy v našem případě formu následujícího podstromu:



4.2 P-pravidlo ANALVG

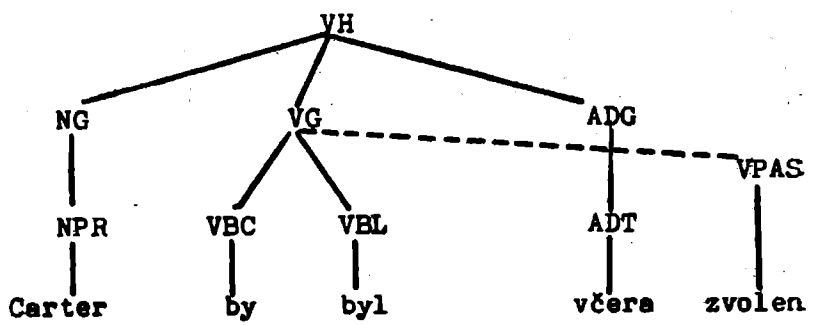
Gramatickým jádrem pravidla ANALVG je nekontextové pravidlo (5) uvedené výše. Analyzuje české slovesné skupiny tvořené jednou, dvěma nebo třemi slovesnými složkami a buduje jejich stromové struktury a připojuje je do stromu-grafu věty.

Patrně nejvýraznějším a nejzávažnějším znakem pravidla ANALVG je jeho kontextovost — bez schopnosti analyzovat kontextově bychom si neporadili s českými analytickými a často nespojitými slovesnými tvary. ANALVG má schopnost procházet celou větu a vyhledávat jednotlivé slovesné složky a stanovit slovesné skupiny, jež jsou těmito složkami tvořeny. K tomu účelu obsahuje ANALVG tři speciální podprocedury MOVE, MOVE1 a MOVE2.

Na příkladech ukážeme činnost ANALVG — zejména rozpoznávání stejných slovesných skupin (=VG) v různých kontextech:

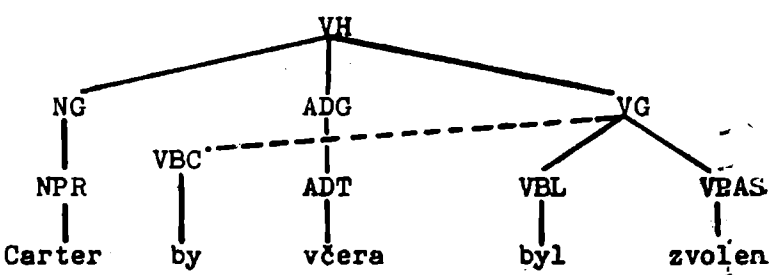
(3) Carter by byl včera zvolen.

(3a)



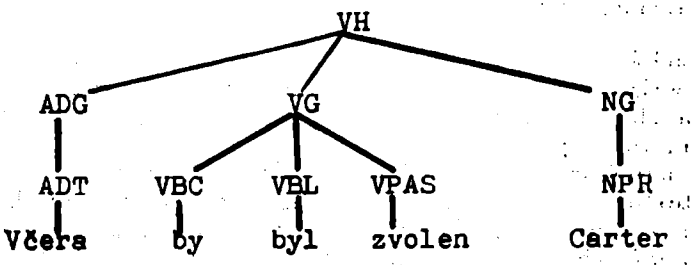
(4) Carter by včera byl zvolen.

(4a)



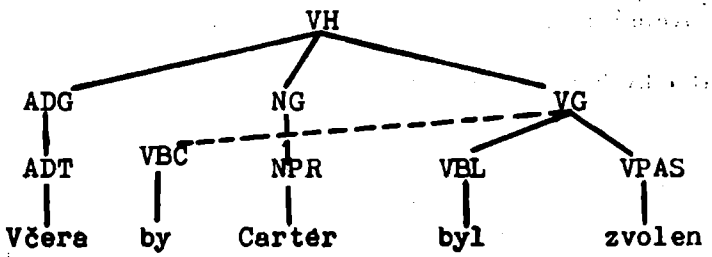
(5) Včera by byl zvolen Carter.

(5a)



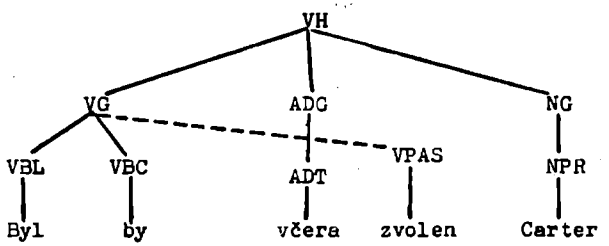
(6) Včera by Carter byl zvolen.

(6a)



(7) Byl by včera zvolen Carter.

(7a)



Kdyby tuto analýzu mělo provádět nekontextové pravidlo (5), výsledek by nebyl úspěšný, protože by nebylo možné analyzovat slovesné skupiny obsahující nespojité složky (tj. (3), (4), (6), (7)), nebylo by možno testovat shodu mezi složkami uvnitř VG a odpovídající subjektovou NG a vybudovat podstrom analyzované VG a připojit jej pod uzel VH.

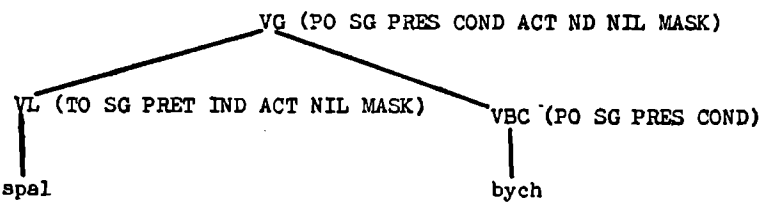
Fungování P-pravidla ANALVG ukážeme na slovesné skupině spal bych. ANALVG začíná tím, že vyhledá slova, z nichž se složka skládá, ve slovníku, čímž je získána následující informace:

(SPAL ((VL) (TO SG PRET IND ACT ND NIL MASK) (NOM)))
(BYCH ((VBC) (PO SG PRES COND)))

Rysy prvního tvaru spal ukazují, že jde o participium minulé (= VL), patrně ve 3. osobě (= TO), v jednotném čísle (= SG), čas minulý (= PRET),

způsob oznamovací (= IND), rod činný (= ACT), vid nedokonavý (= ND), bez negace (= NIL), rod jmenný mužský (= MASK). Tvar bych je pomocný tvar slovesa být (= VBC) a má rysy — 1. osoba (= PO), jednotné číslo (= SG), čas přítomný (= PRES), a způsob podmiňovací (= COND). Protože první tvar je minulé participium, ANALVG ví na základě nekontextového pravidla (5), že má hledat jako další složku buď VB (jako ve VL VB = = spal jsem), nebo VBC, což je právě náš případ. Jakmile bylo VBC nalezeno, porovnají se relevantní rysy obou složek a ty, které vyhoví potřebným podmínkám, jsou vybrány a umístěny do seznamu rysů uzlu VG. Např. rys TO (= 3. osoba) je u VL potlačen a od tvaru VBC je vybrána 1. osoba (= PO), protože VBC je pomocné sloveso. Totéž platí o čase a způsobu — do výsledného seznamu se vybírají rysy tvaru VBC (tj. PRES a COND). Všechny tyto podmínky jsou v pravidle ANALVG obsaženy v podobě podmíněných příkazů, které se testují v průběhu analýzy. VG, která je výsledkem analýzy, bude mít následující graf-strom:

(8)



Lze tedy říci, že P-pravidla vracejí výsledky své činnosti ve formě stromových struktur, tj. každé P-pravidlo buduje a poskytuje jako výsledek stromovou strukturu, jež je typická pro větnou složku, k jejíž analýze je P-pravidlo určeno.

4.3 P-pravidlo ANALAD

Toto pravidlo má za úkol analyzovat adverbia, která tvoří samostatné adverbiální skupiny (= ADG) ve větě. Gramatická část pravidla ANALAD se zakládá na jednoduchém nekontextovém pravidle (7) a dovede rozpoznávat ADG sestávající z adverbií času, místa a způsobu (včetně adverbií míry). Jsou to ADG jako dnes večer, doma, tam doma, rychle, velice rychle apod. Navíc ANALAD obsahuje strategii pro rozpoznávání kontextových závislostí následujícího druhu: často je potřeba testovat slovní druh slova následujícího za právě analyzovaným adverbiem, protože adverbia míry se mohou vyskytovat buď uvnitř NG (např. velmi krásná dívka), nebo uvnitř VG (např. velice plakala nebo velice smutně plakala).

Na druhé straně není ANALAD určeno pro analýzu tzv. adverbiálních pádů, tj. předložkových pádů, které fungují jako ADG místa (na stole), času (v poledne, na začátku roku), způsobu (udělal to šroubovákem), příčiny (zemřel na tyfus) aj. Rozpoznat sémantický status předložkového pádu je obtížné a vyžaduje to použít aparát sémantických rysů spojených jak se substantivy, tak s předložkami. Potřebná síť sémantických rysů se teprve vypracovává a bude patrně součástí pravidla ANAL.

4.4 P-pravidlo ANALVETA

Gramatická struktura tohoto pravidla je založena na nekontextovém pravidle (1) a jeho hlavním úkolem je řídit činnost ostatních P-pravidel a koordinovat jejich akce. ANALVETA tedy aktivuje ostatní P-pravidla na základě formálních signálů získávaných z analyzované věty, případně znovu analyzuje některé skupiny nebo i celou větu, buduje výsledný graf-strom analyzované věty a jeho ohodnocené uzávorkování.

Vývojový diagram pravidla ANALVETA (následující samostatný list) ukazuje, jak jsou propojeny strategie shora dolů a zdola nahoru: analýza začíná vždy vyhledáním slovního druhu prvního vstupního slova věty (zdola nahoru) a podle toho, jaký slovní druh byl zjištěn, volá se příslušné P-pravidlo, které řídí další chod analýzy. Volané P-pravidlo analyzuje potom příslušnou větnou složku strategií shora dolů. Tato kombinace strategií poskytuje P-gramatice značnou pružnost a odstraňuje často zpětné vyhledávání (backup).

P-pravidlo ANALVETA podobně jako pravidlo ANAL obsahuje co do své gramatické struktury maximální přepisovací pravidla, protože je potřeba pokrýt pokud možno všechny možné kombinace větných složek. Může tedy rozpoznávat jak dlouhé věty obsahující třeba pět nebo šest skupin, jako je tomu ve větě

(9) Petr četl včera večer ve vlaku ten náš společný článek o syntaktické analýze českých vět.,

tak i krátké věty tvořené jen jednou slovesnou skupinou, jako je

(10) Spí.

Z vývojového diagramu P-pravidla ANALVETA je vidět, že v dosavadní verzi se zatím nepočítá s analýzou vedlejších vět. K tomu je potřeba doplnit výchozí nekontextové pravidlo (1) a též samo P-pravidlo. Pravidlo (1) lze upravit zhruba následovně

(1') VH → {(ADG) (PREPG) (NG)(ADG) VG (VV) (PREPG) (NG / NGGEN) (NG) (ADG)}.

Do P-pravidla je třeba doplnit další podmíněné příkazy, které testují výskyt spojek a spojovacích výrazů, případně dalších formálních signálů. Po těchto úpravách může ANALVETA analyzovat věty jako

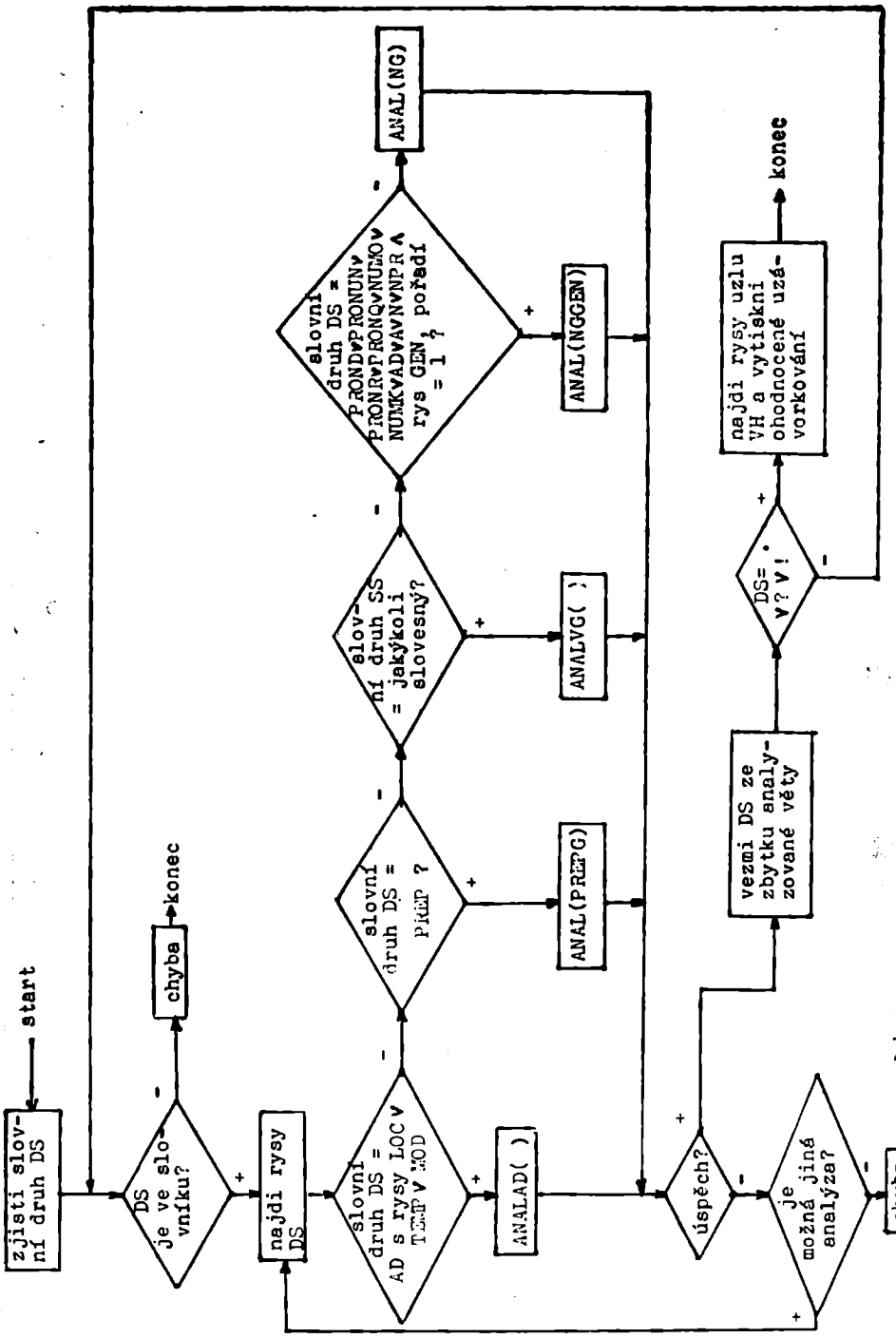
(11) Věděl jsem, že si koupil auto.

(12) Petr mu řekl, aby přinesl pivo.

(13) Poznal jsem, kdo přišel.

Mají-li být analyzovány vztažné věty, je potřeba doplnit podobným způsobem P-pravidlo ANAL, které musí hlídat výskyt vztažných výrazů ve větě.

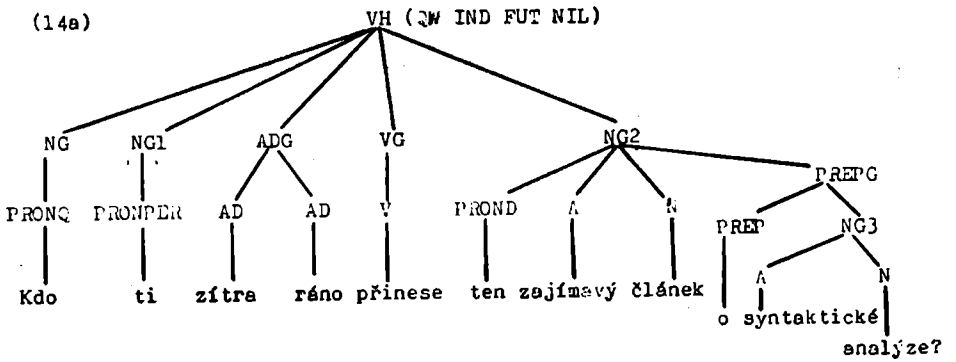
Samostatnou problematiku představují úpravy, které umožní, aby ANALVETA mělo schopnost rozpoznávat souvětí s koordinací. Jsou připravovány pro další verzi P-gramatiky.



procedurální pravidlo ANALVETA

5. STROMOVÁ STRUKTURA VĚTY

Výsledkem činnosti P-gramatiky nad vstupní českou větou je stromová struktura (ohodnocený graf-strom) a jemu odpovídající ohodnocené uzávorkování (viz dále). Strukturu analyzované věty
(14) Kdo ti zítra ráno přinese ten zajímavý článek o syntaktické analýze?
můžeme graficky reprezentovat grafem-stromem



Uvedená stromová struktura se sestává z uzlů a větví. Uzly jsou ohodnoceny a jejich ohodnocení (jména) reprezentují tři druhy jednotek:

(i) slovní druhy
(ii) skupiny (větné složky)
(iii) věty (klauze)

S každým uzlem může být spojen jeho seznam rysů, který může být z tradičního hlediska charakterizován jako soubor gramatických kategorií náležejících konkrétnímu uzlu, např. substantivní skupině (= NG). Přesněji řečeno, rysy jsou pomocné symboly umožňující specifikovat libovolnou vlastnost uzlu, k němuž jsou připojeny. Podle toho, jaké vlastnosti uzlů rysy specifikují, můžeme je rozdělit do následujících skupin:

(a) syntakticky relevantní vlastnosti uzlu
(b) sémanticky relevantní vlastnosti uzlu
(c) pragmaticky relevantní vlastnosti uzlu

Syntakticky relevantní rysy jsou potřebné v průběhu syntaktické analýzy věty. Nesou informaci nutnou k vybudování stromové struktury analyzované věty. Např. první NG v grafu (14a) má rysy (SUBJ NOM SG HUM). První tři rysy mají syntaktickou povahu, protože jsou potřebné pro testování shody jak uvnitř NG tak mezi NG a VG. Rysy SUBJ a SG mají však nadto i povahu sémantickou, např. rys SG může mít značný význam v průběhu následující sémantické analýzy, protože může indikovat určitost NG, u níž se vyskytuje.

Sémanticky relevantní rysy jsou potřebné pro sémantickou analýzu věty, která se snaží najít sémantickou reprezentaci věty. Např. uzel VH v grafu (14a) má rysy (QW IND FUT NIL). Zde je sémanticky relevantním rysem např. rys FUT (budoucí), který indikuje gramatický čas, mající přímý vliv na výslednou podobu sémantické reprezentace (konstrukce). Podobně je sémanticky relevantní i rys NIL, neboť signalizuje, že v sémantické reprezentaci věty se nemá objevit negace (~). Lze ukázat, že některé rysy mohou mít nejen dvojí, ale i trojí status.

Pragmaticky relevantní rysy jsou důležité jednak pro interní (vnitřní) pragmatiku, jednak pro pragmatiku externí (vnější). V oblasti pragmatiky vnitřní jde o možné postoje mluvčího k propozici, která je „za" analyzovanou větou. V seznamu uzlu VH z grafu (14a) je to např. rys QW, který indikuje, že postoj mluvčího k propozici stojící „za" analyzovanou větou (14) je tázací (doplňovací otázka). Pro pragmatiku vnější jsou předmětem pozornosti slovní druhy nebo složky věty, které mají indexickou povahu. Pokud se vyskytují ve větě, vede analýza takové věty k tzv. otevřené konstrukci. Např. v grafu (14a) je NG1, která má ve svém seznamu rysů i rys PRON, což signalizuje, že tato NG1 je pronominální (s osobním zájmenem), takže výsledná konstrukce bude otevřená — bude obsahovat volnou proměnnou typu individua. Jaká konstanta bude za tuto proměnnou dosazena, to určí až pragmatická analýza, která musí jednoznačně zjistit, které individuum je označováno výrazem ti.

Poznamenejme ještě, že PRON je opět příkladem rysu s dvojí povahou, protože je i syntakticky relevantní. Zde bychom rádi řekli, že nám nejde o striktní kategorizaci jednotlivých rysů, spíše je pro nás důležité registrovat všechny druhy relevance a vědět, že daný rys bude důležitý jak pro syntaktickou, tak pro sémantickou analýzu apod. Protože rysy jsou umístěny v seznamech spojených s jednotlivými uzly, jsou snadno dostupné jak pro syntaktický, tak pro sémantický analyzátor.

Právě použití systému seznamů rysů činí P-gramatiku pružnou a relativně jednoduchou. Díky seznamům rysů lze popsat české syntaktické struktury jednoduchými nekontextovými pravidly a současně mít k dispozici vhodný mechanismus pro zachycení kontextových závislostí a vztahů souvýskytu. Navíc je systém rysů pohodlně realizovatelný v programovacím jazyce LISP 1.5, který je určen pro práci se všemi druhy hierarchických struktur a stromů.

6. PŘÍKLAD ANALÝZY

Postup analýzy ukážeme na větě
(15) Kdo ti zítra ráno přinese tu novou knihu o počítačových systémech?,
která byla počítačem analyzována při testování P-gramatiky [3].

Analýza začíná vyhledáním prvního slova věty (15) ve slovníku. Není-li slovo ve slovníku, hlásí se chyba a analýza se zastaví. V našem případě bylo první slovo věty nalezeno a byl hned testován jeho slovní druh. Tato informace totiž určuje, které konkrétní P-pravidlo je třeba volat. V našem případě je slovní druh prvního slova PRONQ (tázací zájmeno), takže je voláno pravidlo ANAL, které začíná svou činnost s předpokladem, že první analyzovanou

skupinou ve větě bude NG, tj. založí se uzel NG a připojí se pod právě založený uzel VH (věta). Po těchto akcích jsou rysy prvního slova (stále kdo — PRONQ) vzaty ze slovníku a připojeny k uzlu PRONQ vytvořenému současně s tím. Rodičovským uzlem pro PRONQ je přirozeně uzel NG, který ve stromu již je. V seznamu rysů uzlu PRONQ najdeme pro pády jen jediný, a to NOM, což znamená, že analyzovaná NG může okamžitě dostat do svého seznamu rys SUBJ (tj. že je subjektem analyzované věty). Vzhledem k tomu, že slovním druhem právě analyzovaného slova (stále kdo) je PRONQ, znamená to pro ANAL, že analyzovaná NG je tvořena jen jednou složkou (jedním slovem). To vede k bezprostřednímu ukončení analýzy této NG tím, že se vytvoří úplný seznam rysů pro tuto NG a připojí se k uzlu NG.

V dalším kroku se ve slovníku hledá následující slovo věty (je to t i — PRONPER čili osobní zájmeno) a testuje se jeho slovní druh, a protože jím je PRONPER, P-pravidlo ANAL volá znovu samo sebe. Jeho činnost je podobná jako v předchozím případě, konkrétně se v průběhu analýzy zjistí, že PRONPER samo o sobě tvoří další NG (tj. NG1) ve větě a je v dativu, takže NG1 jako celek dostane rys OB3 (dativní objekt). Je vytvořen seznam rysů pro NG1 a sama NG1 je opět připojena pod uzel VH.

V tomto okamžiku se přikročí k vyhledání a testování dalšího slova ve větě, jímž je AD (adverbium) zítra. To vede k aktivaci pravidla ANALAD, které zjistí, že analyzované adverbium je časové. To je pro ANALAD signál k prozkoumání okolí slova zítra a k testu, zda slovo následující za právě analyzovaným slovem není opět časové adverbium. Tento test je v našem případě pozitivní (dalším slovem je opět AD času ráno), takže ANALAD vytvoří z těchto dvou adverbií jednu ADG (adverbiální skupinu) času. ADG s rysem TEMP (čas) je pak připojena pod uzel VH a řízení analýzy je předáno P-pravidlu ANALVĚTA, které zjistí, že dalším vstupním slovem je sloveso (V) přinese v jednoduchém tvaru, a zavolá tedy P-pravidlo ANALVG. Úkolem ANALVG je analyzovat slovesnou skupinu ve větě. Zde je první složkou slovesné skupiny V (jednoduchý slovesný tvar), což také znamená, že VG je tvořena právě touto jedinou složkou. Poté se vyberou ze slovníku rysy V a je vytvořen seznam rysů pro celou VG. Zahrnuje následující rysy (TO SG FUT IND ACT DOK NIL), tj. po řadě 3. osoba, singulár, budoucí čas, indikativ, rod slovesný aktivní, dokonavý vid, celý tvar bez negace. Uzel VG je pak připojen pod uzel VH jako jeho dceřinný uzel.

Analýza pokračuje hledáním dalšího slova věty a testováním jeho slovního druhu, což provádí P-pravidlo ANALVETA nebo též (podle okolností) to pravidlo, které předtím analyzovalo příslušnou skupinu. Tentokrát je další slovo PROND (ukazovací zájmeno tu), takže se volá pravidlo ANAL s tím, že analyzovanou složkou bude další NG (NG2). ANAL začne svou obvyklou činnost, tj. vyhledá rysy PROND, vytvoří uzel PROND a připojí jej pod NG2. Poté ANAL hledá další slovo, protože PROND je sice prvním slovem právě analyzované NG2, ale s velkou pravděpodobností ne posledním. Na základě nekontextového pravidla zabudovaného v ANAL je následujícím možným slovem slovo se slovním druhem PRONP (posesívní zájmeno). Test na PRONP je však v našem případě negativní, takže ANAL provádí test na AD, které je dalším možným slovním druhem v rámci NG. Test je opět negativní, ale v dalším kroku, kdy ANAL hledá A (adjektivum), je test úspěšný, protože ve větě bylo nalezeno jako další slovo adjektivum

novou. Hned nato následuje test, zda ve vstupní větě není ještě další A (nebo i víc), ale ten je v našem případě negativní. To vede k dalšímu testu, a to na N (substantivum), jímž je knihu. ANAL může tedy vytvořit rysy pro jednotlivé složky této NG a může také testovat gramatickou shodu mezi nimi. Ještě předtím provede ANAL poslední test, a to, zda ve vstupní větě není za N ještě NPR (vlastní jméno), aby se zachytily případy jako ten známý spisovatel Čapek, nebo zda NG není rozvita další NG (= NGGEN) nebo PREPG. V naší větě je za substantivem knihu předložka o, což je pro P-pravidlo ANAL signál, že po dokončení analýzy skupiny NG2 přejde k analýze PREPG začínající právě předložkou o. Informace o předložce se zaregistruje a pravidlo ANAL provede test na gramatickou shodu mezi složkami analyzované NG2. Tento test je úspěšný, a proto ANAL ihned vytvoří výsledný seznam rysů pro tuto NG2, jenž má podobu (OB4 AK SG FEM NEZ). Hned nato je NG2 připojena pod uzel VH jako dceřinný uzel. Tím ANAL skončí svou činnost a může přejít k dalšímu slovu věty, jímž — jak už víme — je předložka (PREP) o signalizující, že pravidlo ANAL znovu zahajuje svou činnost. Dalším slovem po předložce o je adjektivum (A) počítačových, takže ANAL pokračuje stejným způsobem, jako kdyby analyzovalo normální NG. Hledá tedy další vstupní slovo, jímž je substantivum (N) systémech. Jakmile je úspěšně nalezeno, testuje se shoda mezi předložkou a ostatními složkami této NG3 tak, že se provádí průnik, který musí obsahovat jediný pád, a to LOK. Pak ANAL vybuduje všechny příslušné uzly včetně výsledného uzlu PREPG, který je nakonec připojen pod uzel NG2 jako jeho dceřinný uzel. Toto připojení PREPG k předcházející NG2 je ovšem choulostivou záležitostí, protože je vždy těžké rozhodnout, jakou má přesně povahu daná PREPG — je-li adverbiálním pádem nebo je-li ve větě objektem, nebo je-li atributem v rámci nějaké NG, jak je tomu v tomto případě. Je potřeba konstatovat, že popisovaná verze P-pravidla ANAL bude muset být ještě doplněna a upravena, aby byla schopna spolehlivě rozpoznávat všechny uvedené případy.

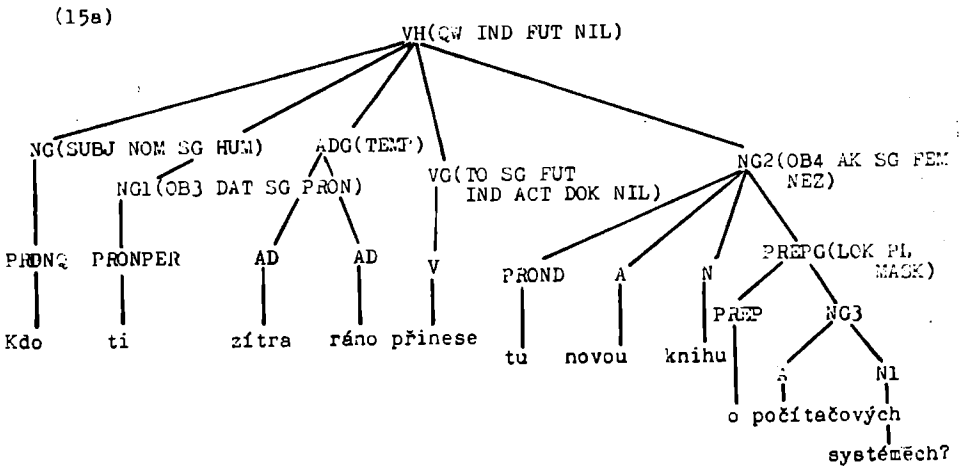
Po skončení analýzy NG2 a PREPG je řízení předáno P-pravidlu ANAL-VETA, které testuje další slovo věty. Tentokrát to však není slovo, nýbrž otazník (?). To je signál, že analýza věty bude končit, a celá věta jako celek bude tázací, a to doplňovací. P-pravidlo ANAL pak dokončí celou analýzu tím, že sestaví výsledný seznam rysů pro uzel VH a dokončí budování syntaktického stromu analyzované věty a jemu odpovídajícího ohodnoceného uzávorkování. V naší větě obsahuje seznam rysů u uzlu VH následující rysy:

(i) protože subjektová NG obsahuje PRONQ a protože na konci věty je otazník, je do seznamu zařazen rys QW indikující, že věta je doplňovací otázka;
(ii) IND (slovesný způsob ve větě je indikativ);
(iii) FUT (sloveso je v budoucím čase);
(iv) NIL (signalizuje, že VG neobsahuje negaci).

Výsledný seznam rysů u uzlu VH má tedy tuto podobu:
VH (QW IND FUT NIL).

Pokud analýza až do tohoto bodu probíhala dobře a nebylo potřeba přistupovat ke zpětnému prohledávání, vyvolají se procedury pro vytištění

výsledné stromové struktury a ohodnoceného uzávorkování, které vytisknou syntaktický strom (15a) analyzované věty (15) a odpovídající ohodnocené uzávorkování (15b).



(15b) ohodnocené uzávorkování

((((KDO) PRONQ) NG(SUBJ NOM SG HUM)
((TI) PRONPER) NG1(OB3 DAT SG PRON)
((ZÍTRA) AD (RÁNO) AD) ADG(TEMP)
((PŘINESE) V) VG(TO SG FUT IND ACT DOK NIL)
((TU) PROND (NOVOU) A (KNIHU) N
((O) PREP ((POČÍTAČOVÝCH) A (SYSTÉMECH) N1) NG3(LOK PL MASK)
PREPG(LOK PL MASK))
NG2(OB4 AK SG FEM NEZ))
VH(QW IND FUT NIL))

7. POČÍTAČOVÁ IMPLEMENTACE

P-gramatika češtiny popsaná výše byla napsána v jazyce LISP 1.5 a stala se tak jádrem syntaktického analyzátoru pro češtinu ([3]), který byl testován nejprve na počítači TESLA 200 v Laboratoři počítacích strojů VUT v Brně od počátku r. 1977.

Čtyři uvedená P-pravidla byla zapsána jako LISPovské funkce se stejným názvem. Tyto čtyři hlavní funkce užívají kolem 30 pomocných LISPovských funkcí, které zajišťují soubory rutinních a opakujících se operací (budování stromu, pohybování se ve stromu, sestavování seznamů rysů apod.).

V průběhu testování (až do současnosti) bylo testováno cca 300 vět různých typů. Výsledky lze pokládat za velmi uspokojující, avšak úplné testování může být provedeno až v druhém pololetí 1980, kdy bude celý analyzátor přenesen na počítač EC 1033, který nemá v porovnání s TESLOU 200 omezení na paměť.

V současné verzi je analyzátor schopen analyzovat jakékoli jednoduché české věty bez ohledu na to, jak bohatě jsou rozvity. V souvislosti s přenosem analyzátoru na počítač EC 1033 budou provedeny další úpravy, které zajistí:

1. analyzátor bude schopen analyzovat i základní typy souřadných a podřadných souvětí v češtině;
2. analyzátor bude doplněn o schopnost rozpoznávat sémantickou povahu předložkových pádů, tj. zda jde o pád adverbiální a jakého typu nebo zda jde o pád objektový nebo zda se jedná o předložkový pád fungující jako neshodný atribut;
3. propojení syntaktického analyzátoru s analyzátorem sémantickým, resp. vytvoření jednoho programu z obou těchto složek.

Závěrem bychom rádi konstatovali, že popsaný syntaktický analyzátor je po provedení zmíněných úprav vhodný i pro praktické využití, i když byl zpočátku orientován převážně experimentálně. V porovnání s vývojem analyzátorů např. pro angličtinu je však nutno říci, že současný trend jde směrem k analyzátorům kompaktnějším, robustnějším, spojujícím syntax a sémantiku v jeden pevný celek a schopným pracovat i s poměrně nestandardními vstupy. Proto soudíme, že pro praktické využití by přece jen bylo vhodnější přistoupit k rozpracování nové verze analyzátoru pro češtinu, který by byl orientován právě zmíněným směrem a splňoval naznačené požadavky.

LITERATURA

[1] Chomsky, N. (1965), *Aspects of the theory of syntax*, Cambridge, Mass.
[2] Sgall, P. et al. (1969), *A functional approach to syntax*, in: Generative description of language, Elsevier, New York.
[3] Vaníčková-Palová, I. (1976), *Syntactic analyzer for Czech*, Papers at the Conference on Cybernetics, Prague.
[4] Winograd, T. (1972), *Understanding natural language*, Academic Press, New York—London.
[5] Woods, A. W. (1969), *Augmented transitional networks for natural language analysis*, Report No-CS-1 (Aiken Computational Laboratory, Harvard University).
[6] Svoboda, A., Materna, P., Pala, K. (1979), *The ordered-triple theory continued*, Brno Studies in English 12, 119—168.

ON THE PROCEDURAL GRAMMAR (FOR CZECH)

The purpose of the present paper is to present a formal grammar of Czech which may serve as a core of the syntactic analyzer suitable for computer processing and applicable, e.g., to dialogue systems, etc.

The explanation begins in Section 2, which contains a brief characterization of the principles on which the presented formal grammar is built. This grammar makes use of the basic set of context-free rules describing the main syntactic structures in Czech, i.e. noun groups, verb groups, adverbial groups and (in the recent version) simple clauses. The context-free rules are transformed into procedures (programs). The collection of these procedures represents a formal grammar called a procedural grammar (for Czech).

Sect. 3 contains the description of the structure of the vocabulary which is used by the procedural grammar. Sect. 4 offers the definition of the procedural grammar for bounded subset of Czech and the description of the individual procedural rules.

Sect. 5 contains the description of the output from the procedural grammar. It is a tree-structure (a tree-graph) of the analyzed sentence. Sect. 6 is an example of the analysis of an Czech sentence (the example was obtained during the testing of the grammar in the TESLA 200 computer).

The closing Sect. contains the information about the computer implementation of the described procedural grammar in the programming language LISP 1.5 in the TESLA 200 computer (1976—1979).