

KAREL PALA

## SYNTAKTICKÝ ANALYZÁTOR PRO ČEŠTINU

### 1. ÚVOD

V tomto článku budou předmětem naší pozornosti otázky automatické syntaktické analýzy českých vět. Touto problematikou jsme se zabývali již dříve z hlediska procedurální gramatiky (Pala, Machová, Havel, 1978, Pala, 1982) a její implementace v programovacím jazyce LISP 1.5 (Vaníčková, 1977).

Zde se chceme zmínit o pokračování tohoto výzkumu s využitím jiných implementačních prostředků (Benešovský, Šmídek, 1984), na jiném počítači (PDP 11/34) a v interaktivním režimu. Nejde tu jen o pouhé přeprogramování předchozího syntaktického analyzátoru, nové implementační nástroje vedou také k zajímavým pohledům na některé úseky české skladby.

V oddíle 2 si všimneme struktury nového syntaktického analyzátoru (budeme jej nazývat ANAL), jeho jednotlivých částí, slovníku, dále se zmíníme o vlastní implementaci (oddíl 3) a v závěrečném oddílu 4 budeme stručně charakterizovat současný stav a dosažené výsledky.

### 2. STRUKTURA ANALYZÁTORU ANAL

Nově budovaný syntaktický analyzátor ANAL má s výchozím „lispovským“ analyzátozem společné některé základní rysy:

a) u obou analyzátorů je jádrem nepříliš velká nekontextová gramatika češtiny (kolem 70 pravidel),

b) u obou analyzátorů se pravidla nekontextové gramatiky člení přirozeným způsobem do skupin implementovaných jako samostatné procedury, které mají schopnost analyzovat jednotlivé větné složky a vytvářet jejich stromové struktury,

c) slovníky jsou v obou případech definovány odděleně od gramatiky a jsou na ní nezávislé.

Další podstatné rysy jsou již rozdílné, což přirozeně vyplývá z funkcionální povahy jazyka LISP 1.5 proti síťovému charakteru systému TPT a jazyka WANDER.

## 2.1 Nekontextová gramatika pro češtinu

Jádrem analyzátoru ANAL je nekontextová gramatika češtiny definovaná obvyklým způsobem jako uspořádaná čtveřice:

$$G_{\varepsilon} = (V_T, V_N, R, V_H),$$

kde  $V_T$  interpretujeme jako terminální slovník tvořený jednotlivými tvary českých slov,  $V_N$  je množina neterminálních symbolů interpretovaných jako slovní druhy, syntaktické a gramatické kategorie potřebné pro popis českých syntaktických struktur,  $R$  je množina nekontextových přepisovacích pravidel tvaru  $A \rightarrow \omega$ , přičemž platí, že  $A \in V_N$ ,  $n$  je řetěz z  $V_N \cup V_T$ ,  $V_N \cap V_T = \emptyset$  a  $V_H$  chápeme jako vyznačený počáteční symbol (jinde též  $S$ ), jímž musí začínat každá derivace v rámci  $G_{\varepsilon}$ .

V našem případě obsahuje množina  $R$  celkem 67 nekontextových pravidel, jimiž se však jednotlivě nebudeme zabývat, spíše si povšimneme jednotlivých skupin pravidel popisujících základní české syntaktické struktury. Učiníme to tak, že uvedeme nejprve hrubou interpretaci symbolů z  $V_N$  (ve výkladu o typové síti na s. 8 pak i podrobnou), protože tyto symboly se vyskytují na levých stranách pravidel v  $R$ .

Neterminální symboly v množině  $V_N$  se, obecně vzato, člení do dvou skupin:

a) slovnědruhové neterminální symboly, jež z lingvistického hlediska označují jednotlivé slovní druhy odpovídající obvyklým slovním druhům rozlišovaným v tradičních gramatikách (viz např. Havránek—Jedlička, 1960). Uvnitř jednotlivých slovních druhů zavádíme však dosti podrobné subklasifikace dané syntaktickými vlastnostmi jednotlivých podtříd slovních druhů — viz dále na s. 9—11.

b) neterminální symboly interpretované lingvisticky jako označení větných složek, jež v tradiční podobě závislostní koncepce odpovídají zhruba větným členům. Na rozdíl od větných členů jsou ovšem větné složky vymezeny čistě formálně. Pracujeme s následujícími větnými složkami (skupinami) a jim odpovídajícími nekontextovými pravidly:

- (i) substantivní skupiny (obecně NG), jež lze dále členit na:
  - substantivní skupiny v přímých pádech (= NG),
  - pronominální substantivní skupiny (= PRONG),
  - předložkové substantivní skupiny (PREP + NG = PREPG),
  - pronominální předložkové substantivní skupiny (= PREPGP),
  - adjektivní skupiny (= AG) tvořené deverbativními adjektivy majícími schopnost vázat na sebe substantivní skupiny v přímých i předložkových pádech.
- (ii) slovesné skupiny (obecně VG), které se dále rozpadají na dvě velké podskupiny, a to na:
  - slovesné skupiny s plnovýznamovým slovesem ať už v infinitivu nebo v určitém tvaru (= VG) a čítající maximálně tři složky, např. *čtu*, *budu číst*, *chtěl bych číst* ap.,
  - slovesné složky tvořené sponovým slovesem *být* a komplementem typu NG, AG, PREPG, PREPGP, ADG (= VGB).
- (iii) adverbální skupiny, jež lze dále členit podle adverbů:
  - místa (= ADGL),
  - času (= ADGT),
  - způsobu (= ADGM) a podle potřeby i dále.

Jednotlivé adverbialní skupiny jsou tvořeny výhradně adverbii nebo jejich kombinacemi (např. *dněs ráno* či *tam doma* ap.). Adverbialní předložkové pády patří díky své formě do skupin PREPG nebo PREPGP.

(iv) větné skupiny či přímo věty, jež představují základní typy vedlejších vět. Zatím pracujeme s

- VVO, což odpovídá vedlejším větám předmětným,
- VVR, což jsou vedlejší věty přívlastkové,
- VVL, tedy vedlejší věty příslovečné místní,
- VVT, tj. vedlejší věty příslovečné časové,
- VVC, tj. vedlejší věty příčinné (s *protože*),
- VVM, tj. vedlejší věty příslovečné způsobové.

Klasifikace obsažené v tradičních skladbách jsou bohatší, zde nám jde především o postižení základních typů postačujících pro vymezenou podmnožinu češtiny.

Nekontextová pravidla z množiny R odpovídají výše uvedeným skupinám, takže máme pravidla popisující strukturu NG, PRONG, PREPG a PREPGP a AG. Odpovídají (i).

Dále máme pravidla popisující strukturu VG a VGB, tj. pravidla odpovídající (ii).

Samostatnou skupinu tvoří pravidla charakterizující strukturu ADG, tj. odpovídající (iii).

Nekontextová pravidla vymežující strukturu vedlejších vět VV odpovídají (iv).

Navíc je třeba mít ještě finální (či počáteční) pravidlo vymežující strukturu VH, tj. pravidlo předpisující, jak lze z jednotlivých složek (NG, VG, VGB, PREPG, PREPGP, PRONG, ADG, VV) složit celou větu (jednoduchou). Formulovat přesně toto pravidlo je pro jazyk s volným slovosledem, jako je čeština, dosti obtížné, proto v současné verzi ANAL pracujeme s aproximací, která má v jazyce WANDER následující podobu:

- a)  $\langle \text{CONST} [A - \text{VGB}] \text{CONST}$
- b)  $\text{CONST} = (*A - \text{NG}$ 
  - A - PRONG
  - A - PREPG
  - A - PREPGP
  - A - ADGT
  - A - ADGL \*)

Jak je vidět, složky VG nebo VGB jsou ve větě obligatorní, kdežto druhá část pravidla – (b) postihuje fakultativnost, s níž se ve větě mohou vyskytovat ostatní složky. Nejobtížnější je odhadnout všechny možné kombinace větných složek, proto v (b) je charakterizována množina složek, a nikoli jednotlivé posoupnosti složek. Ani pravidlo CONST (a), (b) nelze ještě považovat za definitivní řešení, protože nepostihuje beze zbytku problémy s opakováním větných složek.

## 2.2 Procedury SHODA a PÁDY

Množina R nekontextových pravidel sama o sobě nedostačuje pro adekvátní analýzu českých syntaktických struktur (a obecně vět-přirozeného jazyka), neboť nekontextová pravidla neumožňují překonávat kontextová omezení.

V průběhu syntaktické analýzy však poměrně často narážíme např. na analytické slovesné tvary vyznačující se nespojitými složkami a ke kontextu je třeba pravidelně přihlížet při testování gramatické shody jak uvnitř substantivních skupin, tak mezi substantivními a slovesnými skupinami. Kontext je důležitý rovněž při určování větných funkcí substantivních a předložkových skupin.

To tedy znamená, že nekontextová gramatika  $G_c$  musí být zesílena dalšími pravidly—procedurami, která operují na grafových strukturách vytvořených nekontextovými pravidly a dovedou překonávat kontextová omezení.

Prvním příkladem je procedura SHODA, která testuje shodu uvnitř substantivních skupin a dále pak gramatickou shodu mezi VG a tou NG, která má ve větě funkci gramatického podmětu. Je-li test gramatické shody v dané NG negativní, velmi často to znamená, že analýza proběhla chybně a že je potřeba pokusit se o zpětné prohledávání (backtracking) na nejnižší úrovni. Fungování procedury SHODA je založeno na testování průniku gramatických kategorií spojených s jednotlivými prvky dané substantivní složky.

Dalším příkladem je procedura PÁDY, která pro každou větnou substantivní složku určuje její pád, a v závislosti na určeném pádu jí pak přiřadí náležitou větnou funkci. I zde vznikají konflikty, které se řeší testováním pádů jednotlivých skupin v rámci kontextu celé analyzované věty. Nedochází zde však k přestavbě analyzovaných substantivních skupin, mění se jen jejich ohodnocení v termínech větných funkcí.

Ještě je potřeba dodat, že přímo k procedurám obsahujícím jednotlivá nekontextová pravidla jsou připojeny testy na následující slovní druhy, tj. na vhodných místech se testuje, jaký slovní druh následuje za právě analyzovanou složkou. I tyto testy mají kontextový charakter a velmi redukuje počet falešných kroků v průběhu analýzy. Navíc představují elementy strategie zdola nahoru v rámci celkové strategie shora dolů.

### 2.3 Slovník

Slovník je, jak jsme už naznačili, množina  $V_T$  terminálních symbolů, implementovaná v rámci systému TPT jako speciální síť. Ve slovníku jsou uloženy jednotlivé slovní tvary spolu se všemi svými gramatickými kategoriemi a údajem o slovním druhu. U ohebných slovních druhů ukládáme do slovníku všechny tvary, tj. nepočítáme zatím s morfologickou analýzou, která se patrně vyplatí až při dostatečně velkém rozsahu slovníku. Je však vidět, že u jednotlivých slovních tvarů jsou ve slovníku uloženy právě ty údaje, které by byly získány (včetně homonymií) morfologickou analýzou.

Současný počet slovních tvarů ve slovníku—síti je 510. Ponecháme-li stranou implementační detaily, mají jednotlivá hesla ve slovníku následující podobu:

## a) substantivní slovní druhy

heslo	slovní druh	pád	číslo	rod	životn/ neživotn	sém. rys
„programátor“	N	NOM	SG	MASK	HUM	—
„článek“	N	NOM	SG	MASK	NEZ	—
„článek“	N	AK	SG	MASK	NEZ	—
„předního“	A	GEN	SG	MASK/N	—	—
„předního“	A	AK	SG	MASK	HUM	—
„já“	PRPER	NOM	SG	—	HUM	—
„tím“	PROND	INST	SG	MASK/N	—	—
„dvou“	NUMK	GEN	PL	MASK/N	—	—
„dvou“	NUMK	LOK	PL	MASK/FEM/N	—	—

## b) slovesné slovní druhy

heslo	slovní druh	osoba	číslo	čas	způsob	rod	vid	rodn	sém. rys
„čte“	VJ	TO	SG	PRES	IND	ACT	ND	—	—
„četl“	VL	TO	SG	PRET	IND	ACT	ND	—	—
„čten“	VPAS	—	SG	—	—	PAS	ND	MASK	—
„programovat“	VI	—	—	—	—	ACT	ND	—	—
„chtějí“	VM	TO	PL	PRES	IND	ACT	ND	—	—
„přestaneš“	VF	DO	SG	PRES	IND	ACT	DOK	—	—
„budeme“	VB	PO	PL	FUT	IND	ACT	ND	—	—
„není“	VB	TO	SG	PRES	IND	ACT	ND	—	NEG
„byla“	VBL	TO	SG	PRET	IND	ACT	ND	FEM	—
„byla“	VBL	TO	PL	PRET	IND	ACT	ND	NEUT	—
„bys“	VBC	DO	SG	PRES	CON	ACT	ND	—	—

## 2.4 Hierarchie lingvistických pojmů

Specifickou vlastností systému TPT je, že pro danou aplikační oblast vyžaduje vytvoření tzv. typové sítě, která obsahuje hierarchii pojmů užívaných v dané aplikační oblasti.

Z lingvistického hlediska to znamená vytvořit formalizovanou taxonomii slovních, syntaktických a gramatických kategorií, jež potřebujeme pro popis českých syntaktických struktur a pro analýzu českých vět tyto struktury obsahujících, a přeložit ji do grafové struktury (grafu—stromu) — sítě, s níž může pracovat počítač. Tato taxonomie je pak využívána syntaktickým analyzátozem ANAL a umožňuje kontrolovat správnost pojmů užitých při analýze českých vět.

Jednotlivé prvky v typové síti jsou trojího typu:

- slovní druhy
- syntaktické složky (viz výklad o gramatice  $G_{\mathcal{E}}$  na s. 112—113)
- gramatické kategorie.

K prvkům typu (a) poznamenáváme, že jde o obvyklou klasifikaci podle slovních druhů často však doplněnou o dosti podrobné subklasifikace. Stranou jsou ponechána citoslovce. Následuje přehled použitých slovních druhů:

- (i) substantivní: N — substantivum (obecně)  
 NPR — substantivum, vlastní jméno  
 A — adjektivum  
 AV — adjektivum deverbativní  
 NUM — číslovka  
 NUMO — číslovka řadová  
 NUMK — číslovka základní  
 PRON — zájmeno  
 PROND — zájmeno ukazovací  
 PRONP — zájmeno přivlastňovací  
 PRONR — zájmeno vztažné  
 PRONQ — zájmeno tázací  
 PRONUN — zájmeno neurčité  
 PRPER — zájmeno osobní  
 PREP — předložka
- (ii) slovesné: V — sloveso (obecně)  
 VJ — určitý tvar plnovýznamového slovesa  
 VB — určitý tvar slovesa *být*  
 VBC — kondicionálový tvar slovesa *být*  
 VBL — minulé přičestí slovesa *být*  
 VBI — infinitiv slovesa *být*  
 VL — přičestí minulé  
 VPAS — přičestí trpné  
 VM — určitý tvar modálního slovesa  
 VR — určitý tvar reflexivního slovesa  
 VF — určitý tvar fázového slovesa

Analogicky lze mít VMI, VML, VRI, VRL, VFI, VFL apod.

- (iii) ostatní: AD — adverbia (obecně)  
 ADQ — adverbia míry  
 ADM — adverbia způsobu  
 ADL — adverbia místa  
 ADT — adverbia času  
 KON — spojky (obecně)  
 KONK — spojky souřadící  
 CA — spojka *a*  
 CNEB — spojka *nebo*  
 CAL — spojka *ale*  
 KONS — spojky podřadící  
 CZE — spojka *že*  
 CAB — spojka *aby*  
 CKD — spojka *když*  
 CAZ — spojka *až*  
 CPR — spojka *protože*  
 PRT — částice (obecně)  
 PRTR — částice reflexivní (*se*)

K typu (b) poznamenáváme, že pracujeme se všemi obvyklými gramatickými kategoriemi jmennými i slovesnými.

## (i) gramatické kategorie jmenné:

ROD, hodnoty MASK, FEM, NEUT

CIS, hodnoty SG, PL

PAD, hodnoty NOM — nominativ

GEN — genitiv

DAT — dativ

AK — akuzativ

LOK — lokál

INST — instrumentál

## (ii) gramatické kategorie slovesné:

OS, hodnoty PO — první

DO — druhá

TO — třetí

CIS, stejně jako u jmenných kategorií

CAS, hodnoty PRES — přítomný

PRET — minulý

FUT — budoucí

ZPUS, hodnoty IND — indikativ, oznamovací

IMP — imperativ, rozkazovací

CON — kondicionál, podmiňovací

RODV, hodnoty ACT — rod činný

PAS — rod trpný

VID, hodnoty ND — vid nedokonavý

DOK — vid dokonavý

ROD, stejně jako u jmenných kategorií

PN, hodnoty POZ — sloveso bez negace

NEG — sloveso s negací

## (iii) ostatní kategorie či sémantické rysy

TYPV, hodnoty DECL — věta oznamovací

IMPE — věta rozkazovací

QYN — otázka zjišťovací

QW — otázka doplňovací

PERS, hodnoty ZIV, HUM — životnost

NEZ — neživotnost

PER — personálnost

NPE — nepersonálnost

STUP, hodnoty PZ — pozitiv u adjektiv a adverbii

KOMP — komparativ “ “

SUP — superlativ “ “

SRNG, hodnoty SUBJ — subjekt ve větě

OB4 — akuzativní (přímý) objekt

OB3 — dativní (nepřímý) objekt

OB2 — genitivní objekt

INS7 — určení nástroje, prostředku

K typu (c), tj. k syntaktickým složkám, poznamenáváme, že vzhledem k použití složkové syntaktické koncepce pracujeme s větnými složkami vymezenými čistě formálně. Jejich sémantický status je v průběhu analýzy věty teprve rozpoznáván a ohodnocován sémantickými rysy typu SRNG (u nichž počítáme s pochopitelným rozšířením inventáře). Použité větné složky:

- (i) substantivní, tedy obecně typu NG (viz s. 116)
- (ii) slovesné, tedy obecně mající typ VG
- (iii) adverbialní, tj. obecně typu ADG
- (iv) větné, tj. obecně typu VV (vedlejší věty)
- (v) složka VH, tj. typu jednoduché věty, kořen větného grafu —stromu.

Uvedený přehled je poměrně blízký skutečné podobě typové sítě CZECH. WAN, jak je užívána počítačem PDP 11/34, je však v porovnání s ní čitelnější díky doplněné lingvistické interpretaci. Je patrné, že typová síť může být (a také bude) podle potřeby upravována a doplňována zejména o sémantické rysy.

### 3. IMPLEMENTACE

Implementačním nástrojem pro analyzátor ANAL je systém TPT a jazyk WANDER — oba původně vyvíjené (Benešovský, Šmídek, 1984) pro testování správnosti programů. Má řadu vlastností, které jej činí vhodným prostředkem pro budování syntaktických a nepochybně i sémantických analyzátorů přirozeného jazyka.

Nejdůležitější z nich spočívá v tom, že TPT i WANDER jsou speciálně orientovány na práci s gramatikami (WANDER), grafy a sítěmi (TPT). K tomu účelu je v nich definována řada jednoduchých procedur, které se v jiných programovacích jazycích musí vytvářet poměrně obtížně.

Dále je to okolnost, že součástí aplikace TPT je vytvoření typové sítě, o níž jsme se už zmínili. Formulovat typovou síť znamená, že uživatel si musí přesně vymežit hierarchii pojmů ve své aplikační oblasti a navíc, musí svých pojmů užívat korektně. Typová síť je totiž vybudována tak, že v průběhu práce systému ANAL kontroluje korektnost uživatelského zacházení s definovanými pojmy v různých datových strukturách systému. Dojde-li k uživatelem zaviněným nesrovnalostem, systém ANAL ohlásí chybu.

Velmi důležitou vlastností systému TPT je, že umožňuje uživateli práci v interaktivním režimu prostřednictvím obrazovky terminálu připojeného k počítači PDP 11/34.

Uživatel zadá vstupní českou větu na obrazovce terminálu. Nejprve je vytvořena tzv. L-struktura vstupní věty, tj. věta se transformuje na lineární grafovou strukturu, v níž základními uzly jsou slova vstupní věty.

Následuje lexikální analýza a rozpoznání slovních druhů slov obsažených v L-struktuře. Její průběh se objevuje na obrazovce, uživatel—lingvista může tedy přímo sledovat správnost rozpoznávání. Není-li slovo ze vstupní věty ve slovníku, na obrazovce se vypíše „Slovo xxx není ve slovníku“ a pokus o analýzu končí neúspěšně.

Jinak analýza dále pokračuje tak, že strategií shora dolů se postupně aplikují pravidla z  $G_c$  a v případě úspěchu se vybuduje příslušný podstrom v celé stromové struktuře vstupní věty. Pokud došlo k chybným krokům, nepotřebné



uzly se ruší a analýza pokračuje až do konce, tj. až jsou zpracována všechna slova vstupní věty. Již při budování stromové struktury věty se aplikují procedury SHODA a PADY a pokud i ony ukončí svou práci úspěšně, na obrazovce terminálu se objeví prompt označující, že analýza vstupní věty proběhla úspěšně. Poté je možno použít příkazu DRAW a pomocí programu PEPNET nakreslit na obrazovce terminálu výslednou stromovou strukturu, která je syntaktickou analýzou zadané vstupní věty. Jestliže uživatel pracoval u terminálu delší dobu, může všechny získané stromové struktury vytisknout i na řádkové tiskárně použitím příkazu PLOT.

Analýza jedné české věty o 8–10 slovech trvá v současné verzi ANAL nejvýše 20 sec, přičemž analyzátor zabírá celkem asi 100 Kb paměti (vnitřní paměť i disky).

Souhrnně je syntaktický analyzátor implementován jako:

- a) síť – slovník, tj. síť VOCAB.WAN čítající 510 českých slovních tvarů,
- b) síť představující vlastní program syntaktické analýzy v jazyce WANDER, tj. síť ANAL.WAN,
- c) typová síť CZECH.WAN obsahující již zmíněnou hierarchii použitých lingvistických pojmů a vztahů mezi nimi,
- d) pracovní síť WORK.WAN, do níž se v průběhu analýzy ukládají dílčí výsledky – podstromy jednotlivých větných složek a v případě úspěchu nakonec i hlavní výsledek – výsledná stromová struktura vstupní české věty.

#### 4. SOUČASNÝ STAV A VÝSLEDKY

Ke konci r. 1985 byl ANAL schopen pracovat s vymezenou podmnožinou českých vět obsahující jednoduché české věty skládající se z libovolně složitých substantivních skupin, adjektivních skupin, adverbiálních a slovesných skupin (maximálně tříložkových). Dále ANAL dovede analyzovat souvětí zatím se spojkami *že* a *aby*, připravena je analýza souvětí se spojovacími výrazy *kdo*, *co*, *který*, *jak*, *až*, *když*, *protože*, *jestliže*, (*pak*). Testy byly prováděny na testovacím souboru asi o 150 větách s úspěšností 94–95 %.

Získávané stromové struktury nejsou zatím jednoznačné, doplnění o heuristické procedury však povede k výraznému omezení víceznačností. I když výsledné stromové struktury odpovídají spíše povrchovým než hloubkovým strukturám, otázka, zda je potřeba usilovat o získání hloubkových struktur, bude v plném rozsahu zodpovězena až při přebudovávání ANAL na sémantický analyzátor, jehož úkolem bude přiřazovat českým větám z vymezené podmnožiny sémantické reprezentace v rámci transparentní intenzionální logiky (systému TIL, Tichý 1976).

## LITERATURA

1. BENEŠOVSKÝ, M.—ŠMÍDEK, M.: Testování programů. In: Sborník SOFSEM'84, VVS Bratislava 1984.
2. HAVRÁNEK, B.—JEDLIČKA, A.: Česká mluvnická. Praha, SPFFBU 1960.
3. PALA, K.: O procedurální gramatice (pro češtinu). SPFFBU, A 30, 1982, s. 103—122.
4. PALA, K.—MACHOVÁ, S.—HAVEL, I. M.: Přirozený jazyk jako prostředek komunikace člověka s počítačem. In: Sborník SOFSEM '78, VVS Bratislava 1978.
5. TICHÝ, P.: Intensional logic. Manuscript. University of Otago, 1976, 730 s.
6. VANÍČKOVÁ, I.: Syntaktický analyzátor pro české věty. Rigorózní práce, PFF UJEP Brno, 1977.

## SYNTACTIC ANALYZER FOR CZECH

In our previous research in this field we have worked out a procedural grammar for Czech (Pala, 1982) and implemented it as a syntactic analyzer for written Czech sentences in programming language LISP 1.5.

In this paper we present a new version of the earlier analyzer and especially its new interactive implementation on the PDP 11/34 computer. The new version is based on the implementation tools quite different from LISP 1.5, particularly, it is based on the system TPT and language WANDER (Benešovský, Šmídek, 1984). The new approach to implementation has led also to new interesting solutions both in the construction of the analyzer and in the description of some parts of Czech syntax.

The analyzer consists of the basic set of 67 context-free rules describing the main syntactic structures in Czech, i.e. noun groups, adjective groups, verb groups, adverbial groups, clauses and sentence. It also contains procedures (e.g. SHODA—agreement, PÁDY—cases, etc.) that are able to cope with context of the whole sentence. The main strategy is from top to bottom, the result of the analysis is a tree-structure of the input Czech sentence displayed on the screen of terminal.

There are four parts—nets forming together the analyzer:

1. Net VOCAB.WAN—a dictionary of Czech word forms (about 550 forms),
2. Net ANAL.WAN—the program of the analysis, i.e. context-free rules and all the additional procedures and other rules,
3. Basic net CZECH.WAN containing the hierarchy of all linguistic concepts necessary for description Czech syntactic structures,
4. Temporary net WORK.WAN in which the course of the analysis and the resulting tree-structures can be stored and from which it can be drawn or printed.