Pala, Karel

Česká syntax v PROLOGu

Sborník prací Filozofické fakulty brněnské univerzity. A, Řada jazykovědná. 1989-1990, vol. 38-39, iss. A37-38, pp. [111]-120

ISBN 80-210-0168-2
ISSN 0231-7567

Stable URL (handle): https://hdl.handle.net/11222.digilib/101690
Access Date: 14. 12. 2024
Version: 20220831

SBORNÍK PRACÍ FILOZOFICKÉ FAKULTY BRNĚNSKÉ UNIVERZITY
STUDIA MINORA FACULTATIS PHILOSOPHICAE UNIVERSITATIS BRUNENSIS
A 37—38, 1989—1990

KAREL PALA

ČESKÁ SYNTAX V PROLOGU

1. ÚVOD

V tomto článku budeme věnovat pozornost formálnímu popisu některých základních českých syntaktických struktur a pokusíme se ukázat, jak lze některé české syntaktické struktury, např. jmenné skupiny, adverbiální skupiny a další, popsat nekontextovými pravidly a pak je převést na gramatiky konečných klauzulí v PROLOGu (= definite clause grammars, tedy DCG, viz např. Periera, 1983). Naší snahou bude poskytnout čtenáři návod, jak DC gramatiky v PROLOGu budovat a jak si vytvářet experimentální počítačově testovatelné popisy fragmentů české syntaxe.

Postupy uvedené v tomto článku poskytují lingvistům nespecializovaným v oblasti strojové lingvistiky programové vybavení pro práci s formálními gramatikami způsobem, který jim může být relativně blízký a přístupný.

2. NEKONTEXTOVÉ GRAMATIKY A GRAMATIKY V PROLOGU

Nekontextové gramatiky (Chomsky, 1957) zůstávají přes své dílčí nedostatky vhodným nástrojem pro popis struktur na jednotlivých rovinách jazyka. Poskytují prostředky pro dynamický popis jednotlivých jazykových struktur a lze na ně pohlížet buď jako na generativní procedury, jež vyjmenovávají prvky nějaké dané množiny — v našem případě množiny českých vět —, nebo jako na procedury rekognoskativní, jež pro předloženou množinu vět rozpoznávají, jsou-li české či nikoli.

Samy o sobě jsou nekontextové gramatiky ovšem jen semialgoritmy, takže pro praktické užívání je ještě musíme doplnit o další algoritmy, které z nich pak činí generátory nebo analyzátory českých vět nebo třeba slovních tvarů v závislosti na tom, kterou rovinu jazyka

se rozhodneme popisovat. Chceme-li z nekontextových gramatik učinit praktický a běžný nástroj popisu na jednotlivých rovinách jazyka, musíme je formulovat jako programové vybavení, které umožní lingvistovi vytvářet a testovat teoretické popisy s použitím dostupné výpočetní techniky.

Donedávna byla s tímto způsobem práce spojena řada konkrétních překážek, např. byly tu omezené možnosti přístupu k výpočetní technice nebo neexistence lingvisticky orientovaného programového vybavení. V poslední době se však zdá, že zavádění osobních počítačů i na humanitně orientovaná pracoviště poskytne každému lingvistovi možnost pracovat v oblasti teoretického popisu jazyka s použitím vhodných formálních gramatik. Průlom v oblasti programového vybavení pak představuje programovací jazyk PROLOG (tj. PROgramming in LOGic, viz např. Clocksin, Mellish, 1986), který poskytuje značné možnosti pro práci s přirozeným jazykem.

2.1. PROLOG

V dalším se pokusíme ukázat, jak lze nekontextová pravidla vyjádřit ve formalismu jazyka PROLOG. K tomu musíme nejprve uvést a objasnit některé základní pojmy, na nichž je PROLOG založen. Jako východisko nám posloužily práce: Pereira, 1983 a Clocksin, Mellish, 1986.

PROLOG je založen na podmnožině predikátového kalkulu 1. řádu (proto jde o PROgramování v LOGice), konkrétně na Hornových či konečných klauzulích.

Konečné klauzule mají formu:

a) $P{:-}\ Q_1, \ldots, Q_n.$
   Tuto formuli lze interpretovat jako implikaci, která říká, že „P je pravdivé, jsou-li pravdivé $Q_1, \ldots, Q_n$".
b) P.
   Tu interpretujeme „P je pravdivé".

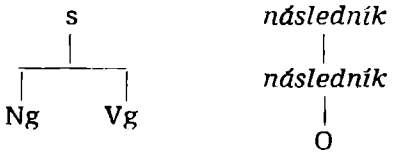
P je hlava klauzule, $Q_1, \ldots, Q_n$ jsou cíle tvořící tělo klauzule. Hlava a cíle reprezentují výskyty predikátů uvedením predikátových konstant a případných argumentů uvedených v kulatých závorkách oddělených čárkami. Argumenty odpovídají individuovým konstantám nebo individuovým proměnným. Příklady predikátů: miluje(X,Y), číslo(0), pravda.

Je vidět, že výskyt predikátu reprezentuje relaci mezi jeho argumenty, tedy „miluje(X,Y) označuje relaci „milovat" mezi objekty X a Y.

Argumenty jsou termy zastupující částečně specifikované objekty, a to:

i) proměnné (individuové) — označují nespecifikované, nevymezené objekty a značíme je vždy velkými písmeny, např. Y,X, G (=gramatický rod), Nu (=gramatické číslo), K (=pád substantiva) aj.
ii) atomy (individua) — označují konkrétní objekty a značíme je malými písmeny, např. nom (=nominativ), 3 (=číslo tři), [] (=prázdný seznam) ap.
iii) složené termy — označují složené objekty, např. s(Ng,Vg), následník (následník(O)) ap. Složené termy jsou tvořeny funktorem a odpovídajícími argumenty. Je vhodné chápat je jako stromové struktury.

Příklady stromových struktur:



Důležitým případem termu je seznam: [] je prázdný seznam. Pro seznamy existuje speciální notace: [a,b] nebo [X|Y].

Klauzule mohou mít různou podobu: tak klauzuli
(1) otec(Honza,Marie).
interpretujeme jako fakt (výrok) „Honza je otec Marie". Podobně klauzuli
(2) otec(X,Marie).
interpretujeme díky proměnné X jako neúplnou „někdo je otec Marie". Konečně klauzule
(3) děda(X,Z) :— otec(X,Y), rodič(Y,Z).
představuje pravidlo, které říká: „X je děda Z, jestliže X je otec Y a Y je rodič Z".

Množina konečných klauzulí uložených do paměti počítače tvoří program a také současně databázi. Program definuje relace označené predikáty objevujícími se v hlavách klauzulí. Když použijeme interpretu, tj. unifikačního mechanismu, pro konečné klauzule, můžeme požadovat splnění určitých cílů. Příkladem příkazu, jímž vyžadujeme splnění cíle, je např.
(4) ? — P.
Tímto příkazem říkáme, že požadujeme ty výskyty relací, které splňují P.

2.2. Gramatická pravidla v PROLOGU

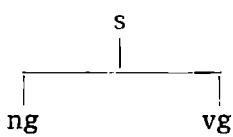
Konečné klauzule určitého tvaru lze chápat jako gramatická pravidla, která umožňují vybudovat gramatiku konečných klauzulí, tj. DC gramatiku.

Vhodné východisko pro formulování DC pravidel poskytují nekontextová pravidla. Mějme např. nekontextové pravidlo

(5) s —→ ng vg,
které říká, že věta s se skládá ze jmenné složky ng a slovesné složky vg. Pravidlo (5) lze převést na konečnou klauzuli
(6) s(SO,S) :— ng(SO,S1), vg(S1,S).,
kterou interpretujeme „v řetězu mezi body SO a S se nachází věta, jestliže mezi body SO a S1 se nachází složka ng a mezi body S1 a S složka vg.

Podobně nekontextové pravidlo s terminálním symbolem (konkrétním slovním tvarem)
(6) pnd —→ ten
říká, že pnd je ukazovací zájméno „ten". Lze je převést na jednoduchou konečnou klauzuli
(7) pnd([ten | S], S).,
kterou čteme „v bodě S řetězu se nachází ukazovací zájméno „ten".

Neterminální symboly z nekontextových pravidel mohou mít v DC pravidlech obecně podobu predikátů s argumenty a terminální symboly mohou být libovolné termy. Chceme-li navíc, aby interpret PROLOGU každému gramatickému pravidlu přiřazoval i jeho odpovídající stromovou strukturu můžeme původní konečnou klauzuli (6) převést na
(8) s(s(Ng,Vg),SO,S) :— ng(Ng,N,SO,S1), vg(VG,NS1,S).
Toto pravidlo říká, že věta s je tvořena jmennou skupinou se strukturou ng a gramatickou kategorií čísla N, za níž následuje slovesná skupina se strukturou vg a shoduje se s ng v čísle N. Navíc ještě (8) přiřadí větě s stromovou strukturu díky tomu, že na levé
(9)



i pravé straně pravidla jsme přidali argumenty (proměnné) Ng a Vg.

Obecně lze tedy neterminální symbol z nekontextového pravidla překládat do DC pravidla jako predikát s n + 2 argumenty, z nichž poslední dva jsou body v řetězu — viz např. pravidlo (6). Tento předpis zároveň naznačuje, že místo konečných klauzulí jako (8) můžeme psát jednodušeji
(10) s(s(Ng,Vg)) —→ ng(Ng,N), vg(Vg,N).

Pravidlu, jako je (10), budeme říkat DC p r a v i d l o a všechna další pravidla budeme psát v této podobě. Pro práci s těmito pravidly potřebujeme však k interpretu PROLOGu připojit jednoduchý preprocesor napsaný rovněž v PROLOGU (Clocksin, Mellish, 1986).

Pro pravidla s terminálními symboly zavedeme rovněž jednotný způsob zápisu. Je z něho vidět, že konkrétní hodnoty gramatických
(11) pnd(pnd(ten), [1,1,4],sg,[mz,mn,mn]) —→ [ten].

kategorií pádu, čísla a rodu zachycujeme jako konstanty nebo seznamy konstant.

3. DC GRAMATIKA ROLE1

V dalším se pokusíme ukázat, jak lze v hrubých rysech vybudovat DC gramatiku popisující základní české syntaktické struktury. Vycházíme z nekontextové gramatiky, která byla vytvořena již dříve a publikována v práci (Pala, 1982).

Pravidla gramatiky, kterou budeme budovat, se přirozeným způsobem člení do dvou skupin:

a) lexikální pravidla s terminálními symboly — tvary českých slov na pravé straně (samostatný soubor SLOVNIK.PLG).

b) syntaktická pravidla popisující jednotlivé české syntaktické struktury — různé typy větných složek (samostatný soubor ROLE1.PLG).

3.1. Lexikální pravidla

Tímto způsobem budujeme slovník dané gramatiky v případě, že nepočítáme s morfologickou analýzou ohebných slov vyskytujících se ve vstupních větách (o morfologické analýze viz Halasová, 1987, Pala, Halasová, Franc, 1987).

Na levých stranách pravidel máme tedy neterminální symboly označující jednotlivé slovní druhy, na pravých stranách jsou konkrétní slovní tvary. Kromě toho levé strany ještě obsahují údaje o gramatických kategoriích spojených s daným slovním druhem a slovním tvarem. V případě tvarové homonymie se přirozeně uvádějí všechny možnosti.

Následují příklady konkrétních lexikálních pravidel, jak byla použita v souboru SLOVNIK.PLG. Pracujeme se všemi slovními druhy (kromě citoslovcí) a pro každý slovní druh uvádíme jeden příklad.

1) substantiva, n
   n(n(psa), [2,4],sg,mz) --> [psa].
2) adjektiva, a
   a(a(zlý),[1,1,4],sg,[mz,mn]) --> [zlý].
3) zájmena ukazovací, pnd
   pnd(pnd(ti),1,pl,mz) --> [ti].
3.1) zájmena osobní, per
   per(per(mnou),1,7,sg,—) --> [mnou].
3.2) zájmena přivlastňovací, pos
   pos(pos(jeho),—,—,—) --> [jeho].
3.3) zájmena vztažná, pr
   pr(pr(kterému),3,sg,[m,n,f]) --> [kterému].

3.4) zájmena tázací, pq
pq(pq(kdo),1,sg,mz) —→ [kdo].
3.5) zájmena neurčitá, pun
pun(pun(nic),[1,4],sg,n) —→ [nic].
3.6) zájmena záporná, pne
pne(pne(někom),6,sg,mz) —→ [někom].
4. číslovky řadové, no
no(no(prvnímu),3,pl,[mz,mn,n]) —→ [prvnímu].
4.1) číslovky základní, nk
nk(nk(tří),2,pl,[mz,mn,f,n]) —→ [tří].
5) slovesa plnovýznamová patřící do jisté třídy, v1
v1(v(beru),1,1,sg,—) —→ [beru].
5.1) sloveso pomocné být, vb
vb(vb(jsi),2,1,sg,—) —→ [jsi].
6) adverbia času, dt, místa, dl, způsobu, dm, míry, dq
dl(dl(tady)) —→ [tady].
dt(dt(veřer)) —→ [večer].
dm(dm(skvěle)) —→ [skvěle].
dq(dq(velice)) —→ [velice].
7) předložky, pre
pre(pre(o),[4,6]) —→ [o].
8) spojky, cze, ca
cze(cze(že)) —→ [že].
ca(ca(a)) —→ [a].
9) partikule zvratná, ptr, srovnávací, pFs
ptr(ptr(se)) —→ [se].
pts(pts(jako)) —→ [jako].
10) hraniční symboly, fm, qm, sm, em
fm(fm(,)) —→ [,].
qm(qm(?)) —→ [?].
sm(sm(.)) —→ [.].
em(em(!)) —→ [!].

Uvedené příklady představují, jak lze snadno vidět, základní schéma českého slovníku pro DC gramatiku. Slovník lze libovolně rozšiřovat tak, aby obsahoval potřebné tvary i hesla. U některých slovních druhů nejsou uvedeny všechny možné subklasifikace, nebylo by však obtížné je doplnit.

3.2. Syntaktická pravidla

Poté, co jsme vybudovali slovník a máme tak k dispozici seznam neterminálních symbolů odpovídajících slovním druhům, můžeme přistoupit k sestavování DC pravidel pro jednotlivé české syntaktické struktury. Pro nedostatek místa zde ovšem nebudeme probírat všechny a hlavní

pozornost soustředíme na jmenné skupiny, na nichž se pokusíme demonstrovat postup vytváření DC gramatiky.

3.2.1. JMENNÉ SKUPINY

Jednou ze základních syntaktických struktur v rámci jednoduché české věty jsou jmenné skupiny. Jsou tvořeny určitým typickým seskupením nominálních slovních druhů, mezi nimiž jako konstruktivní prvek funguje substantivum či zájmeno v přímém nebo předložkovém pádě, pokud ponecháme stranou případy s elipsou. Jmenné skupiny jsou rekurzívní: jednak fungují jako základní větné složky (mimo sloveso), jednak v sobě mohou obsahovat další jmenné skupiny a též relativní věty, které opět obsahují jmenné skupiny jako své základní složky.

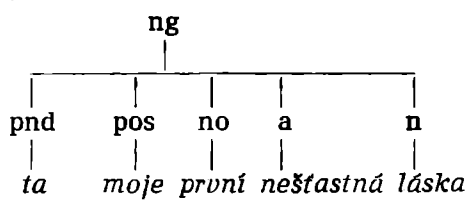
Strukturu jmenné složky lze velmi dobře definovat v termínech slovních druhů, jak byly uvedeny výše, a to prostřednictvím nekontextového pravidla

(12) ng —→ {(pnd) (pos) (no) (nk) (*a)
(n) (ng2) (ap) (ppg) (sr)},

které říká, že jmenná složka ng je tvořena danými slovními druhy v uvedeném pořadí. Kulaté závorky značí fakultativnost, takže pravidlo (12) zachycuje jak složitější české ng, např. ng typu „ti moji první dva noví šikovní kamarádi z našeho sídliště", tak i eliptické ng jako „ta" ve větě „Ta to neviděla." Překlad (12) na odpovídající DC pravidlo má jistě více řešení, v našem případě jsme zvolili alternativu

```
(13) ng(ng(Png,Pos,No,Nk,A,N,X),K,Nu,G) —→
                {w(pnd(Pnd,K,Nu,G))   ; nil(Pnd) ),
                (w(pos(Pos,K,Nu,G))   ; nil(Pos) ),
                (w(no(No,K, Nu,G))    ; nil(No) ),
                (w(a(A,K,Nu,G))       ; nil(A) ),
                (w(n(N,K,Nu,G,TL))),
               (ng(X,2,—,—,—) ; ppg(X,—,—) ; ap(X,K,Nu,G);
                 sr(X,Nu,G) ; nil(X)) ; nil(N), nil(X)),
          {not (Pnd='*',Pos='*', No='*', A='*',X='*',N='*')}.
```

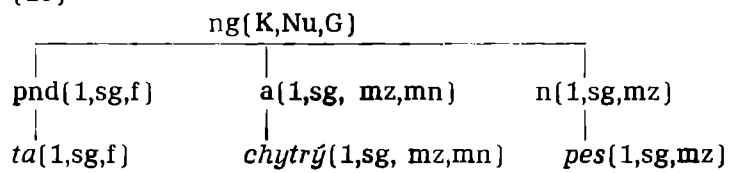
Levá strana pravidla (13) obsahuje neterminální symbol ng a proměnné slovních druhů, aby interpret PROLOGU mohl při analýze konkrétní české ng vytvořit její odpovídající stromovou strukturu. Byla-li na vstupu třeba ng „ta moje první nešťastná láska", bude jí přiřazena stromová struktura



Pravá strana pravidla (13) je poněkud složitější než odpovídající pravá strana nekontextového pravidla (12), je to však výborná ukázka toho, co PROLOG při psaní gramatických pravidel umožňuje. Jednotlivé řádky na pravé straně pravidla (13) uvozené predikátem w zachycují fakultativnost výskytu jednotlivých slovních druhů v ng. Predikáty s proměnnou X zase vyjadřují skutečnost, že pokud se v ng vyskytlo substantivum, může za ním následovat další ng v genitivu („ta dívka dobrého srdce") nebo adjektivní skupina ap („dopis napsaný v rozrušení"), předložková skupina ppg („jeho žena se svým třetím milencem"), případně relativní věta sr („můj přítel, který se mi stále smál").

Pravidlo (13) promě toho dovede testovat gramatickou shodu mezi slovními druhy, z nichž se skládá. K tomu slouží proměnné gramatických kategorií: K = pád, Nu = číslo, G = jmenný rod. Protože analýza v PROLOGU postupuje shora dolů, objevují se u jednotlivých uzlů proměnné gramatických kategorií jako u uzlu ng ve stromové struktuře

(15)



Jakmile se však ve slovníku najdou konkrétní slovní tvary ze vstupní ng a k nim jejich konkrétní hodnoty gramatických kategorií, pokusí se interpret PROLOGU o unifikaci v souladu s pravidlem (13). Unifikace však může uspět jen v případě, když stejným proměnným odpovídají stejné konstanty. To však u analyzované ng „ta chytrý pes" nelze splnit, proto její analýza neuspěje a její stromová struktura se nevytvoří.

Obdobně se testuje shoda mezi slovesem a podmětovou jmennou skupinou v rámci věty, rozdíl je však v testovaných proměnných — gramatických kategoriích. Pokud chceme či potřebujeme některou proměnnou ignorovat, značíme ji podtržítkem „_" jako tzv. anonymní proměnnou.

Z pravidla (13) odvodíme velmi snadno DC pravidlo pro jmennou skupinu v předložkovém pádě:

(16) ppg(ppg(Pre,Ng),K) --> pre(Pre,K), ng(Ng,K,Nu,G).

K pravidlům (13) a (16) uveďme ještě DC pravidla pro pronominální skupiny s osobním zájmem (17) a se zájmem neurčitým (18).

(17) png(png(Per),P,K,Nu,G) --> per(Per,K,Nu,G).

(18) png(png(Pun),P,K,Nu,G) --> pun(Pun,K,Nu,G).

Podobně bychom postupovali při formulování DC pravidel pro jmenné a pronominální skupiny fungující v tázacích a vztažných větách. Rovněž ponecháme stranou DC pravidla popisující strukturu adverbiálních skupin, ať už jsou vyjádřeny adverbii či adverbiálními pády.

Na závěr si povšimneme DC pravidel popisujících strukturu jednoduché české věty. I zde lze volit různá řešení, např. lze pracovat s kategorií větného členu, jak je zavedena v tradičních gramatikách (Bauer, Grepl, 1972, Šmilauer, 1966). To by vedlo k DC pravidlům definujícím třeba podmět jako ng v nominativu, přímý předmět jako ng v akuzativu, neshodný přívlastek genitivní jako ng v genitivu atd.

Chceme-li však DC pravidel využít pro popis sémantické stavby jednoduché české věty, jeví se jako vhodné užít jiného přístupu, v němž se lze obejít bez kategorie větného členu. Je založen na pojmu slovesného rámce (Fillmore, 1968) či větného vzorce (Daneš, 1981). Jádro tohoto postupu spočívá v tom, že slovesné rámce sloves patřících do jednotlivých sémantických tříd přímo formulujeme jako DC pravidla. Tak pro sloveso vidět z třídy sloves smyslového vnímání máme DC pravidlo

(19) s(s(Ag,M,V2,Ob,T,L),P,K,Nu,G) —→
ag(AG,P,K,Nu,G),m(M),v2(V2,P,K,Nu,G),ob(Ob),t(T),l(L).

V naší DC gramatice ROLE1 je ovšem použito složitější DC pravidlo, než je (19), protože při popisu struktury jednoduché věty vzniká řada problémů s obligatorností a fakultativností jednotlivých aktantů. Dále je potřeba řešit značně komplikované otázky českého volného slovosledu, což vede k výrazně složitějším DC pravidlům.

V pravidle (19) jsme kromě slovesa z třídy v2 zavedli aktanty agens (ag) a objekt (ob). Nyní stačí jen dalšími jednoduchými DC pravidly specifikovat, jak se tyto aktanty realizují v povrchové struktuře věty, — učiníme to v pravidlech (20) a (21):

(20) ag(ag(Ng),3,K,Nu,G) —→ ng(Ng,1,Nu,G).
(21) ob(ob(Ng)) —→ ng(Ng,Nu,G).

Zde jsou uvedeny jen základní formální realizace obou aktantů, není však obtížné formulovat další DC pravidla a testovat pak jejich adekvátnost. Zároveň je vidět, že při tomto postupu nevzniká potřeba zavádět kategorie větných členů.

4. ZÁVĚR

Všechna pravidla uvedená výše a s nimi spojené úvahy pocházejí z experimentální DC gramatiky ROLE1, která byla implementována v PROLOGU NU7 na počítači PDP 11/34 v ÚVT UJEP. Omezená paměť tohoto počítače nás však nutí k přechodu na jiný počítač (Commodore PC 40 AT) a také na jiný PROLOG, konkrétně Arity PROLOG, který má již jak interpretační, tak i kompilační verzi. Doplněná a ověřená verze české DC gramatiky ROLE1 bude v blízké budoucnosti k dispozici pro všechny potenciální zájemce.

Pro čtenáře, kteří mají k dispozici osobní počítač s vhodným PROLOGEM, může DC gramatika ROLE1 posloužit jako východisko či vzor pro

další experimenty, ale také jako mohutný nástroj pro získávání dalších poznatků v oblasti syntaktické roviny jazyka a pro jejich bezprostřední testování. Jsme přesvědčeni, že PROLOG a na něm založené DC gramatiky si také poměrně rychle najdou cestu do výuky skladby nejen na oboru bohemistika, ale i na dalších filologických oborech.

LITERATURA

CLOCKSIN, W. F.—MELLISH, C. S.: Programming in PROLOG (Programování v PROLOGU), český překlad. ZAK Slušovice, 1986, s. 240—263, s. 300—303.
DANEŠ, F.—HLAVSA, Z.: Větné vzorce v češtině. Praha, 1981.
FILLMORE, C.: The Case for Case. In: Universals in Linguistic Theory, ed. E. Bach a R. T. Harms, New York 1968.
GREPL, M.—KARLÍK, P.: Skladba spisovné češtiny. Praha, 1986.
HALASOVÁ, K.: Algoritmický popis české morfologie. Diplomová práce, Brno 1987.
CHOMSKY, N.: Syntaktické struktury, český překlad. Praha 1966.
PALA, K.: O procedurální gramatice (pro češtinu). SPFFBU, A 30, 1982, s. 103—122.
PALA, K.—HALASOVÁ, K.— FRANC, S.: Česká morfologie a syntax v PROLOGU. In: SOFSEM '87, VVS Bratislava, 1987.
PEREIRA, F.: Logic for Natural Language Analysis, Technical Note 275, SRI, 1983.

CZECH SYNTAX IN PROLOG

In the presented paper it is our main purpose to tackle two problems: a) to give the basic information about programming language PROLOG with regard to linguistic needs consisting in writing and testing computer grammars for Czech, b) to describe the way of building definite clause (=DC) grammars for Czech. We start with context-free rules describing the selected syntactic structures of Czech, such as noun groups, and we show how one can translate them into DC grammar rules. In fact, we give the instruction how to assemble DC grammar for the main syntactic structures in Czech.

All the presented examples and analyses are based on the experimental DC grammar ROLE1 that has been implemented in PROLOG NU7 on PDP 11/34. A new version of ROLE1 is being developed for RC AT and is going to be written in ARITY PROLOG.