```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                         %
% File:            serbocro.dtr                                           %
% Purpose:         Serbo Croat noun inflection                            %
% Author:          Gerald Gazdar, 24 March 1989                           %
% Version:         1.01                                                   %
%                                                                         %
%      Copyright (c) University of Sussex 1989.  All rights reserved.     %
%                                                                         %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% Not to be taken too seriously as a linguistic analysis: stress is ignored.

Pnoun:   <nom sing> == "<root>"
         <gen sing> == "<nom plur>"
         <dat sing> == <nom plur>
         <acc sing> == <nom sing>
         <loc> == "<dat>"
         <ins sing> == "<root>" om
         <nom plur> == "<root>" i
         <gen plur> == "<gen sing>"
         <dat plur> == "<root>" ima
         <acc plur> == "<nom plur>"
         <ins plur> == "<loc plur>"
         <gender> == feminine.

Fnoun:   <> == Pnoun
         <nom sing> == "<root>" a
         <acc sing> == "<root>" u
         <nom plur> == "<root>" e
         <gen plur> == "<root>" a
         <dat plur> == "<root>" ama.

Mnoun:   <> == Pnoun
         <gen sing> == "<root>" a
         <dat sing> == "<root>" u
         <acc plur> == "<root>" e
         <gender> == masculine.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

Banka:   <> == Fnoun
         <root> == bank
         <dat sing> == banci
         <meaning> == 'BANK'.

Dinar:   <> == Mnoun
         <root> == dinar
         <meaning> == 'MONEY'.

Frula:   <> == Fnoun
         <root> == frul
         <meaning> == 'FLUTE'.

Glad:    <> == Pnoun
         <root> == glad
         <ins sing> == "<root>" u
         <meaning> == 'HUNGER'.

Kost:    <> == Pnoun
         <root> == kost
         <ins sing> == koshcu
         <meaning> == 'BONE'.
```

```
Noga:   <> == Fnoun
        <root> == nog
        <dat sing> == nozi
        <meaning> == 'LEG'.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% Some example theorems:
%
% Dinar:
%       <meaning> = MONEY
%       <gender> = masculine
%       <root> = dinar
%       <nom sing> = dinar
%       <gen sing> = dinar a
%       <acc sing> = dinar
%       <dat sing> = dinar u
%       <loc sing> = dinar u
%       <ins sing> = dinar om
%       <nom plur> = dinar i
%       <gen plur> = dinar a
%       <acc plur> = dinar e
%       <dat plur> = dinar ima
%       <loc plur> = dinar ima
%       <ins plur> = dinar ima.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

# hide  Pnoun Fnoun Mnoun.

# show  <meaning>
        <gender>
        <root>
        <nom sing>
        <gen sing>
        <acc sing>
        <dat sing>
        <loc sing>
        <ins sing>
        <nom plur>
        <gen plur>
        <acc plur>
        <dat plur>
        <loc plur>
        <ins plur>.

% The next line is the Revision Control System Id: do not delete it.
% $Id: archive.dtr 5.2 96/02/04 19:35:39 geraldg Exp $
```

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                         %
% File:           sbcnx.dtr                                               %
% Purpose:        Serbo-Bosno-Croatian noun inflection                    %
% Authors:        Gerald Gazdar, 24 March 1989                            %
%                 James Kilbury, 24 May 1995                              %
% Version:        2.02, 15 June 2007                                      %
% Related files:  serbocro.dtr, hrvatski.dtr, croat.dtr, sbcn.dtr         %
%                                                                         %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

# nc seq node:<root>.                           % node conversion (QDATR)

MG:     <> == Else                              % morphographemics
        <+> == <>
        <g + i> == z i <>
        <k + i> == c i <>
        <d + j> == dj <>
        <s t + j> == sh c <>.

Else:   <*> ==
        <$x> == $x MG:<>.


Pnoun:  <gender> == feminine                    % gender

        <mor> ==  MG:<"<root>" + "<flex>" *>    % morphotactics

                                                % global syncretism
        <flex loc>      == "<flex dat>"         %  - strict
        <flex ins plur> == "<flex dat plur>"    %
                                                %
        <flex acc> == "<flex nom>"              %  - defeasible

        <flex gen sing> == <flex dat sing>      % local syncretism
        <flex dat sing> == <flex nom plur>      %

        <flex nom sing> ==                      % inflections
        <flex ins sing> ==  j u                 %    see stvarju
        <flex nom plur> ==  i                   %    (short i)
        <flex gen plur> ==  i                   %    (long i)
        <flex dat plur> ==  i m a.              %

Mnoun:  <> == Pnoun
        <gender> == masculine
        <flex gen sing> ==  a
        <flex dat sing> ==  u
        <flex ins sing> ==  o m
        <flex gen plur> ==  a                   %    (long a)
        <flex acc plur> ==  e .

Fnoun:  <> == Mnoun
        <gender> == feminine
        <flex nom sing> ==  a                   %    (short a)
        <flex acc sing> ==  u
        <flex gen sing> ==  e                   %    (long e)
        <flex dat sing> ==  i
        <flex nom plur> ==  e                   %    (short e)
        <flex dat plur> ==  a m a.


% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

Bank:   <> == Fnoun                  % dat sing  banci < bank + i
        <meaning> == 'BANK'.
```

```
Dinar:  <> == Mnoun
        <meaning> == 'MONEY'.

Frul:   <> == Fnoun
        <meaning> == 'FLUTE'.

Glad:   <> == Pnoun                 % ins sing  g l a dj u < glad + ju
        <meaning> == 'HUNGER'.

Kost:   <> == Pnoun                 % ins sing  k o sh c u < kost + ju
        <meaning> == 'BONE'.

Nog:    <> == Fnoun                 % dat sing  nozi < nog + i
        <meaning> == 'LEG'.

Stvar:  <> == Pnoun
        <meaning> == 'THING'.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% Some example theorems:
%
% Dinar:
%       <meaning> = MONEY
%       <gender> = masculine
%       <root> = dinar
%       <mor nom sing> = d i n a r
%       <mor acc sing> = d i n a r
%       <mor gen sing> = d i n a r a
%       <mor dat sing> = d i n a r u
%       <mor loc sing> = d i n a r u
%       <mor ins sing> = d i n a r o m
%       <mor nom plur> = d i n a r i
%       <mor acc plur> = d i n a r e
%       <mor gen plur> = d i n a r a
%       <mor dat plur> = d i n a r i m a
%       <mor loc plur> = d i n a r i m a
%       <mor ins plur> = d i n a r i m a.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

# hide  MG Else Pnoun Fnoun Mnoun.

# show  <meaning> <gender> <root>
        <mor nom sing> <mor acc sing> <mor gen sing>
        <mor dat sing> <mor loc sing> <mor ins sing>
        <mor nom plur> <mor acc plur> <mor gen plur>
        <mor dat plur> <mor loc plur> <mor ins plur>.
```

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                             %
% File:            hcr.dtr                                                    %
% Purpose:         historical-comparative reconstruction                      %
% Author:          James Kilbury, 11 November 1997                            %
% Related files:   phon_fst.dtr                                               %
% Version:         1.03 (2 June 2004)                                         %
%                                                                             %
%      Copyright (c) University of Duesseldorf, 1997.  All rights reserved.   %
%                                                                             %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% The following illustrates the method of historical-comparative reconstruc-
% tion with realistic but invented data for a fictitious language family
% consisting of the languages A, B, C, and D. Common features of C and D
% allow the reconstruction of Proto-CD ("PCD") in addition to Proto-ABCD
% ("PABCD"). Similarities to Indoeuropean are not accidental, but an adequate
% example for the latter would require much more depth.
% Note that in transcriptions, starred forms refer to reconstructed segments
% or forms of a proto-language.
%
% background literature:
% Raimo Anttila, Historical and Comparative Linguistics, Amsterdam: John
%    Benjamins, 1989 (2nd ed.)
% Winfred P. Lehmann, Historical Linguistics, London & New York: Routledge,
%    1992 (3rd ed.)
% ------------------------------------------------------------------------
%
% the reconstructed Proto-ABCD sound system
%  - consonants
%
%    i  @  u       p b   t d   k g
%    e     o             s z
%       a            m     n
%                    r
%  - vowel correspondences
%
%    PABCD   *i  *@  *e  *a  *o  *u
%    A        i   i   a   a   a   u
%    B        i   a   e   a   o   u
%    C        i   a   e   a   a   u
%    D        i   o   e   o   o   u
%
% principal sound shifts
%
% (A) complete merger
%  1) *@ merges with *a except in A, where it merges with *i.
%  2) The mid vowels *e and *o merge with *a in A.
%  3) *a and *o merge in C and D but are realized differently,
%     i.e. as /a/ in C and as /o/ in D.
%
% (B) partial merger
%  4) Voiceless obstruents merge intervocalically with voiced in B.
%
% (C) split
%  5) The affricates /c j/ arise in A through the regressive palatali-
%     zation of /k g/ by a following front vowel. [NB: /i/ arising
%     from *@ does not cause palatalization. How could the definition
%     be changed to require feeding order?]
%
% (D) rephonologization
%  6) In C voiceless stops become fricatives except after /s/,
%     and voiced stops become voiceless.
```

```
%
% (E) epenthesis
%  7) *sr becomes /str/ in C and D.
%
% (F) elision
%  8) Prevocalic clusters of *s plus a voiceless stop lose *s in D.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# vars $mid:    e o.                          % mid vowels
# vars $front:  i e.                          % front vowels
# vars $vowel:  i @ e a o u.                  % vowels
# vars $vowel2: i @ e a o u.

# vars $vel:    k g.                          % velar stops
# vars $cvl:    p t k s.                      % voiceless consonants
# vars $stop:   p t k b d g.                  % stops


Else:                                         % the default transducer (Idem)
   <> ==
   <$X> == $X "<>".

PABCD:
   <> == Else
   <@> == "<a>".

A:
   <> == PABCD
   <@> == "<i>"
   <$mid> == "<a>"
   <$vel $front> == "<Palatalization:<$vel> $front>".

B:
   <> == PABCD
   <$vowel $cvl $vowel2> == "<$vowel Voicing:<$cvl> $vowel2>".

PCD:
   <> == PABCD
   <o> == "<a>"
   <s r> == "<s t r>".

C:
   <> == PCD
   <$stop> == Shift "<>"                      % consonant shift
   <s $stop> == s $stop "<>".                 % consonant shift blocked

D:
   <> == PCD
   <a> == o "<>"
   <s $stop $vowel> == "<$stop $vowel>".


% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

Palatalization:
   <k> == c
   <g> == j.

Voicing:
   <p> == b
   <t> == d
   <k> == g
```

```
   <s> == z.

Shift:
   <p> == f
   <t> == th
   <k> == x
   <b> == p
   <d> == t
   <g> == k.


% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

# show <k e t>  <g i b a>  <k @ p u>  <e s r o>
       <d @ s a g>  <o s p u n>  <s k a p i m>.

# hide Else PABCD PCD Palatalization Voicing Shift.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% Some theorems arranged as cognate sets:
%
%    A:<k e t> = c a t.
%    B:<k e t> = k e t.
%    C:<k e t> = x e th.
%    D:<k e t> = k e t.
%
%    A:<g i b a> = j i b a.
%    B:<g i b a> = g i b a.
%    C:<g i b a> = k i p a.
%    D:<g i b a> = g i b o.
%
%    A:<k @ p u> = k i p u.
%    B:<k @ p u> = k a b u.
%    C:<k @ p u> = x a f u.
%    D:<k @ p u> = k o p u.
%
%    A:<e s r o> = a s r a.
%    B:<e s r o> = e s r o.
%    C:<e s r o> = e s t r a.
%    D:<e s r o> = e s t r o.
%
%    A:<d @ s a g> = d i s a g.
%    B:<d @ s a g> = d a z a g.
%    C:<d @ s a g> = t a s a k.
%    D:<d @ s a g> = d o s o g.
%
%    A:<o s p u n> = a s p u n.
%    B:<o s p u n> = o s p u n.
%    C:<o s p u n> = a s p u n.
%    D:<o s p u n> = o p u n.
%
%    A:<s k a p i m> = s k a p i m.
%    B:<s k a p i m> = s k a b i m.
%    C:<s k a p i m> = s k a f i m.
%    D:<s k a p i m> = k o p i m.
```

```
! sbc20.txt                                        James Kilbury, 30 June 2009

! based on sbc15.txt with a dummy tag FLX for use in realization rules
! designed for use with the trace facility


read lexc < sbc_lx20.txt
define lex


! tag classes of morphological categories
define FClass   ["+PN"|"+FN"|"+MN"] ;                  ! inflectional classes
define Num      ["+Sg"|"+Pl"] ;                        ! numerus
define Cas      ["+Nm"|"+Ac"|"+Gn"|"+Dt"|"+Lc"|"+In"]; ! case
define AnyTag   [ FClass|Num|Cas] ;

! natural classes of phological segments
define Vwl [i|e|a|o|u|j u] ;                           ! vowels
define CnsVls [p|t|k|s|c] ;                            ! voiceless cons
define CnsVcd [b|d|g|z|m|n|l|r|j] ;                    ! voiced cons
define Cns [CnsVls|CnsVcd] ;                           ! consonants


! inflectional morphology
! syncretism, inheritance, and realization (order: class - FLX - num - case)

define flx      [[..] -> FLX || FClass _ ] ;           ! insert dummy tag FLX

define dsync    ["+Lc" -> "+Dt" ]          .o.         ! syncretism of dative,
                 ["+In" -> "+Dt" ||"+Pl" _ ]  ] ;      ! locative and instr

define acreal   [FLX -> u ||"+FN" _ "+Sg" "+Ac",,      ! realization & blocking
                 FLX -> e ||"+MN" _ "+Pl" "+Ac" ] ;    ! of syncretism in acc

define nsync    ["+Ac" -> "+Nm"] ;                     ! nom/acc syncretism

define pnsync   ["+Sg" "+Gn" -> "+Pl" "+Nm" ,          ! syncretism in class PN
                 "+Sg" "+Dt" -> "+Pl" "+Nm"
                             || "+PN" FLX _   ] ;

define pnflx    [[FLX -> j u ||"+PN" _ "+Sg" "+In",,   ! realization in class PN
                  FLX -> i   ||"+PN" _ "+Pl" "+Nm",,
                  FLX -> i 3 ||"+PN" _ "+Pl" "+Gn" ] .o.
                 ["+PN" -> "+MN"]                  ]; ! inheritance

define fnflx    [[FLX -> a     ||"+FN" _ "+Sg" "+Nm",, ! realization in class FN
                  FLX -> e     ||"+FN" _ "+Sg" "+Gn",,
                  FLX -> i     ||"+FN" _ "+Sg" "+Dt",,
                  FLX -> e     ||"+FN" _ "+Pl" "+Nm",,
                  FLX -> a m a ||"+FN" _ "+Pl" "+Dt"] .o.
                 ["+FN" -> "+MN"]                  ]; ! inheritance

define mnflx    [FLX -> 0     || _ "+Sg" "+Nm",,       ! realization in class MN
                 FLX -> a     || _ "+Sg" "+Gn",,       ! (the default class)
                 FLX -> u     || _ "+Sg" "+Dt",,
                 FLX -> o m   || _ "+Sg" "+In",,
                 FLX -> i     || _ "+Pl" "+Nm",,
                 FLX -> a 3   || _ "+Pl" "+Gn",,
                 FLX -> i m a || _ "+Pl" "+Dt"  ] ;

define mb1      [[..] -> "+" || FClass _ ] ;          ! insert morph boundary

define tgdel    [AnyTag -> 0] ;                        ! tag deletion
```

```
define Morph    [flx .o. dsync .o. acreal .o. nsync .o. pnsync .o.
                 pnflx .o. fnflx .o. mnflx .o. mb1 .o. tgdel      ] ;


! morphophonemics (NB: '3' designates vowel length)

define p1   [k -> c, g -> z || _ "+" i] ;              ! palatalization

define p2   [q -> a || _ Cns "+" .#.] ;                ! jer vocalization

define p3   [q -> 0 || _ Cns "+" Vwl] ;                ! jer deletion

define p4   [[..] -> a || Cns _ Cns "+" a 3 ] ;        ! a epenthesis (+Pl+Gn)

define p5   [l -> o || Vwl _ ("+") [Cns|.#.]] ;        ! l vocalization

define p6   [b -> p, d -> t, g -> k || _ CnsVls "+" ]; ! voicing assimilation

define p7   [t c -> c ] ;                              ! tc simplification

define p8   [3 -> 0 ] ;                                ! vowel length

define pmb2 ["+" -> 0 ] ;                              ! morpheme boundary del

define MPhon [p1 .o. p2 .o. p3 .o. p4 .o. p5 .o. p6 .o. p7 .o. p8 .o. pmb2] ;


read regex [ lex .o. Morph .o. MPhon ] ;


! playlist definition

! define sbc [lex .o. flx .o. dsync .o. acreal .o. nsync .o. pnsync .o. pnflx
.o. fnflx .o. mnflx .o. mb1 .o. tgdel .o. p1 .o. p2 .o. p3 .o. p4 .o. p5 .o. p6
.o. p7 .o. p8 .o. pmb2] ; # QQ
```

```
# hcr4.txt                                        James Kilbury, 04.06.2008
#                                                                11.05.2009

# historical-comparative reconstruction

# based on a tree model of language relationships


# natural phonological classes of the proto-language X (=PrABCD)

define VwlMid  [e|o] ;                        # mid vowels
define VwlFrt  [i|e] ;                        # fromt vowels
define Vwl     [i|e|"@"|a|o|u] ;              # vowels

define Stp     [p|t|k|b|d|g] ;                # stops
define CnsVls  [p|t|k|s] ;                    # voiceles consonants
define CnsVcd  [b|d|g|m|n|r] ;                # voiced consonants
define Cns     [CnsVls|CnsVcd] ;              # consonants

define Seg     [Vwl|Cns] ;                    # segments


# phonotactics of X

define PL      [X ([Cns|s k]) Vwl Cns ([(Cns) Vwl (Cns)])] ;
                                              # NB: 'X' is a tag.
# define PL    [X Seg*] ;               alternative smaller net



# isoglosses/rules defining the daughter languages of X

define r1a     ["@" -> a] ;

define r1b     ["@" -> i] ;

define r2      [VwlMid -> a] ;

define r3      [ [a|o] -> ao ] ;

define r3a     [ ao -> a ] ;

define r3b     [ ao -> o ] ;

define r4      [p -> b, t -> d, k -> g , s -> z
                              || Vwl _ Vwl] ;

define r5      [k -> c, g -> j || _ VwlFrt] ;

define r6      [p -> f, t -> th, k -> x,    # a hack!
                b -> p, d -> t,  g -> k || \s  _ Vwl,
                                          Vwl _ Vwl,
                                          Vwl _ .#. ];

define r7      [s r -> s t r] ;

define r8      [s -> 0 || _ Stp Vwl ] ;



# historical phonology (hp) of daughter languages

define Ahp     [[X -> A] .o. r1b .o. r5 .o. r2] ;
```

```
define PrBCDhp [[X -> PrBCD] .o. r1a] ; # proto-BCD

define Bhp     [[PrBCD -> B] .o. r4] ;

define PrCDhp  [[PrBCD -> PrCD] .o. r3 .o. r7] ;

define Chp     [[PrCD -> C] .o. r3a .o. r6] ;

define Dhp     [[PrCD -> D] .o. r3b .o. r8] ;


# historical reconstruction of the family (Stammbaum)

define HCR     [PL .o.
                   [ Ahp |
                    [PrBCDhp .o.
                              [ Bhp |
                               [PrCDhp .o.
                                       [Chp |
                                        Dhp   ]]]]]] ;


read regex HCR ;


# cognate sets

define CSet1   [ [HCR .o. {Acat}].u   & [HCR .o. {Bket}].u &
                 [HCR .o. [C x e th]].u  & [HCR .o. {Dket}].u ] ;
#                                "th" is a multicharacter symbol!

define CSet2   [ [HCR .o. {Ajiba}].u & [HCR .o. {Bgiba}].u &
                 [HCR .o. {Ckipa}].u & [HCR .o. {Dgibo}].u  ] ;

define CSet3   [ [HCR .o. {Acipu}].u & [HCR .o. {Bkabu}].u &
                 [HCR .o. {Cxafu}].u & [HCR .o. {Dkopu}].u  ] ;

define CSet4   [ [HCR .o. {Aasra}].u & [HCR .o. {Besro}].u &
                 [HCR .o. {Cestra}].u & [HCR .o. {Destro}].u ] ;

define CSet5   [ [HCR .o. {Adisag}].u & [HCR .o. {Bdazag}].u &
                 [HCR .o. {Ctasak}].u & [HCR .o. {Ddosog}].u  ] ;

define CSet6   [ [HCR .o. {Aaspun}].u & [HCR .o. {Bospun}].u &
                 [HCR .o. {Caspun}].u & [HCR .o. {Dopun}].u   ] ;

define CSet7   [ [HCR .o. {Askapim}].u & [HCR .o. {Bskabim}].u &
                 [HCR .o. {Cskafim}].u & [HCR .o. {Dkopim}].u   ] ;



define CSet11  [ [HCR .o. {Ckipa}].u & [HCR .o. {Dgibo}].u ] ;

define CSet12  [ [HCR .o. {Ajiba}].u & [HCR .o. {Ckipa}].u ] ;

define CSet13  [ [HCR .o. {Bgiba}].u & [HCR .o. {Ckipa}].u ] ;

define CSet14  [ [HCR .o. {Ajiba}].u & [HCR .o. {Bgiba}].u ] ;
```

```
# hcr5.txt                                    James Kilbury, 04.06.2008
#                                                            17.05.2009

# historical-comparative reconstruction

# based on a wave model of language relationships


# natural phonological classes of the proto-language X (=PrABCD)

define VwlMid  [e|o] ;                       # mid vowels
define VwlFrt  [i|e] ;                       # fromt vowels
define Vwl     [i|e|"@"|a|o|u] ;             # vowels

define Stp     [p|t|k|b|d|g] ;               # stops
define CnsVls  [p|t|k|s] ;                   # voiceles consonants
define CnsVcd  [b|d|g|m|n|r] ;               # voiced consonants
define Cns     [CnsVls|CnsVcd] ;             # consonants

define Seg     [Vwl|Cns] ;                   # segments


# phonotactics of X

define PL      [X ([Cns|s k]) Vwl Cns ([(Cns) Vwl (Cns)])] ;
# NB: 'X' is a tag for the language.


# isoglosses/rules defining the daughter languages of X

define r1a     ["@" -> a               || $§r1a _] ;

define r1b     ["@" -> i               || $§r1b _] ;

define r2      [VwlMid -> a            || $§r2 _] ;

define r3      [ [a|o] -> ao           || $§r3 _] ;

define r3a     [ ao -> a               || $§r3a _] ;

define r3b     [ ao -> o               || $§r3b _] ;

define r4      [p -> b, t -> d, k -> g ,
                        s -> z         || $§r4 Vwl _ Vwl] ;

define r5      [k -> c, g -> j         || $§r5_ VwlFrt] ;

# The next rule is a terrible hack!
define r6      [p -> f, t -> th, k -> x,
                b -> p, d -> t,  g -> k || $§r6 \s  _ Vwl,
                                          $§r6 Vwl _ Vwl,
                                          $§r6 Vwl _ .#. ] ;

define r7      [s r -> s t r           || $§r7 _] ;

define r8      [s -> 0                 || $§r8 _ Stp Vwl] ;


# relative chronology of sound changes (rough)
define RChron  [r1a .o. r1b .o.
                r5  .o.
                r2  .o. r3  .o.
                r3a .o. r3b .o.
```

```
                r4  .o. r6  .o. r7 .o. r8] ;

# isogloss/rule distribution in daughter languages
define Ahp     [X -> A §r1b §r2 §r5] ;
define Bhp     [X -> B §r1a §r4] ;
define Chp     [X -> C §r1a §r3 §r3a §r6 §r7] ;
define Dhp     [X -> D §r1a §r3 §r3b §r7 §r8] ;
define RTag    [§r1a|§r1b|§r2|§r3|§r3a|§r3b|§r4|§r5|§r6|§r7|§r8] ;
define DelT    [RTag -> 0] ;
define HCR     [PL .o. [Ahp|Bhp|Chp|Dhp] .o. RChron .o. DelT ] ;

read regex HCR ;
```

```
! ocsn3.txt                                        James Kilbury, 28 June 2011

! Old Church Slavonic noun inflection


read lexc < ocsn_lx.txt
define lex


! tag classes of morphological categories
define FClass ["+aST"|"+oST"|"+iST"] ;                  ! inflectional classes
define Num    ["+Sg"|"+Pl"] ;                           ! numerus
define Gen    ["+Ms"|"+Fm"|"+Nt"] ;                     ! genus
define Cas    ["+Nm"|"+Ac"|"+Gn"|"+Dt"|"+Lc"|"+In"|"+Vc"]; ! casus
define AnyTag [FClass|Gen|Num|Cas] ;

! natural classes of (morpho)phonological segments (cf. Lunt, Lindstedt)
define VwlFrt [I|i|i1|i2|e|A|A1|A2|E|JU|JA] ;          ! front vowels
define VwlBck [U|u|y|    o|a|      O      ] ;          ! back vowels
define Vwl    [VwlFrnt|VwlBck] ;                        ! vowels
define CnsVls [p|t|k|c|C|s|S|x                ] ;      ! voiceless cons
define CnsVcd [b|d|g|D|  z|Z|v|m|n|l|r|N|L|R|j] ;      ! voiced cons
define Cns    [CnsVls|CnsVcd] ;                         ! consonants


! inflectional morphology
! syncretism, inheritance, and realization
! order: FLX - class - gen - num - case)

define flx    [ [..] -> "+" FLX || _ FClass ] ;         ! insert dummy tag FLX

define sync  [["+Vc"  -> "+Nm" ||       "+Pl" _ ,
                                 "+Nt" "+Sg" _ ] .o.

              ["+Ac"  -> "+Nm" || "+Nt"  ?    _ ,
                                  "+Fm" "+Pl" _ ,
                                  "+Ms" "+Sg" _ ]  ] ;


define oflx  [[FLX -> U       || _ "+oST" "+Ms" "+Sg" "+Nm"] .o.   ! class oST
              [FLX -> e       || _ "+oST" "+Ms" "+Sg" "+Vc"] .o.
              [FLX -> o       || _ "+oST" "+Nt" "+Sg" "+Nm"] .o.
              [FLX -> a       || _ "+oST" ?     "+Sg" "+Gn"] .o.
              [FLX -> u       || _ "+oST" ?     "+Sg" "+Dt"] .o.
              [FLX -> A1      || _ "+oST" ?     "+Sg" "+Lc"] .o.
              [FLX -> o m I   || _ "+oST" ?     "+Sg" "+In"] .o.

              [FLX -> i1      || _ "+oST" "+Ms" "+Pl" "+Nm"] .o.
              [FLX -> y       || _ "+oST" "+Ms" "+Pl" "+Ac"] .o.
              [FLX -> a       || _ "+oST" "+Nt" "+Pl" "+Nm"] .o.
              [FLX -> U       || _ "+oST" ?     "+Pl" "+Gn"] .o.
              [FLX -> o m U   || _ "+oST" ?     "+Pl" "+Dt"] .o.
              [FLX -> y       || _ "+oST" ?     "+Pl" "+In"] .o.
              [FLX -> A1 x U  || _ "+oST" ?     "+Pl" "+Lc"]  ] ;

define iflx  [["+Ac" -> "+Nm" ||   "+iST" ?     "+Sg"   _  ] .o.   ! class iST
              [FLX -> 0       || _ "+iST" ?     "+Sg" "+Nm"] .o.
              [FLX -> j O     || _ "+iST" "+Fm" "+Sg" "+In"] .o.
              [FLX -> m I     || _ "+iST" "+Ms" "+Sg" "+In"] .o.

              [FLX -> j e     || _ "+iST" "+Ms" "+Pl" "+Nm"] .o.
              [FLX -> j       || _ "+iST" ?     "+Pl" "+Gn"] .o.
              [FLX -> m U     || _ "+iST" ?     "+Pl" "+Dt"] .o.
```

```
                [FLX -> m i     || _ "+iST" ?     "+Pl" "+In"] .o.
                [FLX -> x U     || _ "+iST" ?     "+Pl" "+Lc"] .o.

                [FLX -> i       || _ "+iST" ?      ?     ?  ]  ] ;  ! default

define aflx  [["+Lc" -> "+Dt" ||   "+aST" ?     "+Sg"  _    ] .o.   ! class aST
                [FLX -> a       || _ "+aST" ?     "+Sg" "+Nm"] .o.
                [FLX -> o       || _ "+aST" ?     "+Sg" "+Vc"] .o.
                [FLX -> O       || _ "+aST" ?     "+Sg" "+Ac"] .o.
                [FLX -> y       || _ "+aST" ?     "+Sg" "+Gn"] .o.
                [FLX -> A1      || _ "+aST" ?     "+Sg" "+Dt"] .o.
                [FLX -> o j O   || _ "+aST" ?     "+Sg" "+In"] .o.

                [FLX -> y       || _ "+aST" ?     "+Pl" "+Nm"] .o.
                [FLX -> U       || _ "+aST" ?     "+Pl" "+Gn"] .o.
                [FLX -> a m U   || _ "+aST" ?     "+Pl" "+Dt"] .o.
                [FLX -> a m i   || _ "+aST" ?     "+Pl" "+In"] .o.
                [FLX -> a x U   || _ "+aST" ?     "+Pl" "+Lc"]  ] ;

define tagdel [AnyTag -> 0] ;                          ! tag deletion

define Morph  [flx .o. sync .o. oflx .o. iflx .o. aflx .o. tagdel] ;

! morphophonemics

define p1   [k -> c,  g -> D, x -> s || _ "+" [i1|A1]] ;   ! 1st palatalization

define p2   [k -> C,  g -> Z, x -> S || _ "+" [e|i2|A2]];  ! 2nd palatalization

define p3   [[i1|i2] -> i, [A1|A2] -> A] ;                 ! diacritic merger

define p4   [I "+" i -> i] ;                               ! cf. iflex

define p5   ["+" -> 0 ] ;                          ! morph boundary deletion

define MPhon [p1 .o. p2 .o. p3 .o. p4 .o. p5] ;

read regex [ lex .o. Morph .o. MPhon ] ;

! playlist definition

! define ocsn [lex .o. flx .o. sync .o. iflx .o. oflx .o. aflx .o. tagdel .o.
!               p1  .o. p2  .o. p3   .o. p4   .o. p5                  ] ; # QQ
```

```
! orsn7.txt                                    James Kilbury, 28 June 2011

! Old Church Slavonic / Russian / Serbo-Coatian noun inflection

read lexc < ocsn_lx.txt
define lex


! tag classes of morphological categories
define FClass ["+aST"|"+oST"|"+iST"] ;                ! inflectional classes
define Num    ["+Sg"|"+Pl"] ;                         ! numerus
define Gen    ["+Ms"|"+Fm"|"+Nt"] ;                   ! genus
define Cas    ["+Nm"|"+Ac"|"+Gn"|"+Dt"|"+Lc"|"+In"|"+Vc"]; ! casus
define AnyTag [ FClass|Gen|Num|Cas] ;

! natural classes of (morpho)phonological segments (cf. Lunt, Lindstedt)
define VwlFrt [I|i|i1|i2|e|A|A1|A2|E|JU|JA] ;         ! front vowels
define VwlBck [U|u|y|    o|a|      O      ] ;         ! back vowels
define Vwl    [VwlFrnt|VwlBck] ;                      ! vowels
define CnsVls [p|t|k|c|C|s|S|x                ] ;    ! voiceless cons
define CnsVcd [b|d|g|D|  z|Z|v|m|n|l|r|N|L|R|j] ;    ! voiced cons
define Cns    [CnsVls|CnsVcd] ;                       ! consonants


define lgs    [ [..] ->
                ["*OCS"|"*RUS"|"*SCR"] || Cas _ ] ;   ! language tags


! inflectional morphology
! syncretism, inheritance, and realization
! order: FLX - class - gen - num - case)

define flx    [ [..] -> "+" FLX || _ FClass ] ;       ! insert dummy tag FLX


define sync  [["+Ac"  -> "+Nm" || "+Nt"  ?    _ ,
                                   "+Fm" "+Pl" _ ,
                                   "+Ms" "+Sg" _ ] .o.
              ["+Vc"  -> "+Nm" ||       "+Pl" _ ,
                                   "+Nt" "+Sg" _ ] .o.
              ["+Vc"  -> "+Nm" ||            _ "*RUS"] .o.
              ["+Lc"  -> "+Dt" ||            _ "*SCR"] .o.
              ["+In"  -> "+Dt" || "+Pl"      _ "*SCR"] .o.

              ["+oST" -> "+iST" || _ ? "+Pl" "+Dt" "*SCR"] .o.
              ["+aST" -> "+oST" || _ ? "+Sg" "+In" "*SCR",
                                   _ ? "+Pl" "+Gn" "*SCR" ] .o.
              ["+iST" -> "+aST",
               "+oST" -> "+aST" || _ ? "+Pl" ["+Dt"|"+Lc"|"+In"] "*RUS"] ] ;


define oflx  [[FLX -> U       || _ "+oST" "+Ms" "+Sg" "+Nm"] .o.   ! class oST
              [FLX -> e       || _ "+oST" "+Ms" "+Sg" "+Vc"] .o.
              [FLX -> o       || _ "+oST" "+Nt" "+Sg" "+Nm"] .o.
              [FLX -> a       || _ "+oST" ?     "+Sg" "+Gn"] .o.
              [FLX -> u       || _ "+oST" ?     "+Sg" "+Dt"] .o.
              [FLX -> A1      || _ "+oST" ?     "+Sg" "+Lc"] .o.
              [FLX -> o m     || _ "+oST" ?     "+Sg" "+In" "*RUS"] .o.
              [FLX -> o m I   || _ "+oST" ?     "+Sg" "+In"] .o.

              [FLX -> y       || _ "+oST" "+Ms" "+Pl" "+Nm" "*RUS"] .o.
              [FLX -> i1      || _ "+oST" "+Ms" "+Pl" "+Nm"] .o.
              [FLX -> y       || _ "+oST" "+Ms" "+Pl" "+Ac"] .o.
              [FLX -> a       || _ "+oST" "+Nt" "+Pl" "+Nm"] .o.
              [FLX -> o v     || _ "+oST" ?     "+Pl" "+Gn" "*RUS"] .o.
              [FLX -> a       || _ "+oST" ?     "+Pl" "+Gn" "*SCR"] .o.
```

```
              [FLX -> U       || _ "+oST" ?     "+Pl" "+Gn"] .o.

              [FLX -> o m U   || _ "+oST" ?     "+Pl" "+Dt"] .o.
              [FLX -> y       || _ "+oST" ?     "+Pl" "+In"] .o.
              [FLX -> A1 x U  || _ "+oST" ?     "+Pl" "+Lc"]  ] ;

define iflx  [["+Ac" -> "+Nm" ||   "+iST" ?     "+Sg"   _  ] .o.   ! class iST
              [FLX -> 0       || _ "+iST" ?     "+Sg" "+Nm"] .o.
              [FLX -> j O     || _ "+iST" "+Fm" "+Sg" "+In"] .o.
              [FLX -> m I     || _ "+iST" "+Ms" "+Sg" "+In"] .o.

              [FLX -> j e     || _ "+iST" "+Ms" "+Pl" "+Nm"] .o.
              [FLX -> i       || _ "+iST" ?     "+Pl" "+Gn" "*SCR"] .o.
              [FLX -> j       || _ "+iST" ?     "+Pl" "+Gn"] .o.
              [FLX -> i m a   || _ "+iST" ?     "+Pl" "+Dt" "*SCR"] .o.
              [FLX -> m U     || _ "+iST" ?     "+Pl" "+Dt"] .o.
              [FLX -> m i     || _ "+iST" ?     "+Pl" "+In"] .o.
              [FLX -> x U     || _ "+iST" ?     "+Pl" "+Lc"] .o.

              [FLX -> i       || _ "+iST" ?      ?     ?  ]  ] ;  ! default

define aflx  [["+Lc" -> "+Dt" ||   "+aST" ?     "+Sg"   _   ] .o.   ! class aST
              [FLX -> a       || _ "+aST" ?     "+Sg" "+Nm"] .o.
              [FLX -> o       || _ "+aST" ?     "+Sg" "+Vc"] .o.
              [FLX -> O       || _ "+aST" ?     "+Sg" "+Ac"] .o.
              [FLX -> y       || _ "+aST" ?     "+Sg" "+Gn"] .o.
              [FLX -> A1      || _ "+aST" ?     "+Sg" "+Dt"] .o.
              [FLX -> o j     || _ "+aST" ?     "+Sg" "+In" "*RUS"] .o.
              [FLX -> o j O   || _ "+aST" ?     "+Sg" "+In"] .o.

              [FLX -> y       || _ "+aST" ?     "+Pl" "+Nm"] .o.
              [FLX -> U       || _ "+aST" ?     "+Pl" "+Gn"] .o.
              [FLX -> a m a   || _ "+aST" ?     "+Pl" "+Dt" "*SCR"] .o.
              [FLX -> a m U   || _ "+aST" ?     "+Pl" "+Dt"] .o.
              [FLX -> a m i   || _ "+aST" ?     "+Pl" "+In"] .o.
              [FLX -> a x U   || _ "+aST" ?     "+Pl" "+Lc"]  ] ;


define tagdel [AnyTag -> 0] ;                          ! tag deletion

define Morph  [flx .o. sync .o. oflx .o. iflx .o. aflx .o. tagdel] ;


! morphophonemics

! 1st palatalization
define p1   [k -> c,  g -> D, x -> s || _ "+" [i1|A1]   $["*OCS"|"*SCR"]] ;
! 2nd palatalization
define p2   [k -> C,  g -> Z, x -> S || _ "+" [e|i2|A2] $["*OCS"|"*SCR"]];
! diacritic merger
define p3   [[i1|i2] -> i, [A1|A2] -> A] ;

define p4  [[I "+" i -> i] .o.                              ! cf. iflex

            [I "+" m I  -> e m  ,
             I "+" j e  -> i    || _ $["*RUS"|"*SCR"]] .o.

            [I "+" j O  -> j u  || _         $"*SCR"] .o.

            [I "+" j O  -> I JU ,
             I "+" a    -> JA   || _ $"*RUS"] .o.

            [I "+" j    -> e j  || _ $"*RUS"]  ] ;

define p5   [A -> i, y -> e || "+" _ $"*SCR"] ;             ! cf. aflex
```

```
define p6   ["+" -> 0 ] ;                                    ! boundary del

define MPhon [p1 .o. p2 .o. p3 .o. p4 .o. p5 .o. p6] ;


! phonology

define s1  [[ U -> 0  || _ ["*RUS"|"*SCR"]] .o.              ! final jers
            [ I -> 0  || _         "*SCR" ]  ] ;

define s2  [[ O -> u  || _ $["*RUS"|"*SCR"]] .o.             ! nasal vowels
            [ E -> JA || _  $"*RUS"]         .o.
            [ E -> e  || _         $"*SCR"]   ] ;

define s3   [ A -> e  || _  $"*RUS"] ;                      ! jat'

define s4   [ y -> i  || _         $"*SCR"] ;                ! jery

define s5   [ s t j -> S c || _    $"*SCR"] ;

define Phon [s1 .o. s2 .o. s3 .o. s4 .o. s5] ;


read regex [ lex .o. lgs .o. sync .o. Morph .o. MPhon .o. Phon] ;


! playlist definition

! define ocsn1 [lex .o. lgs .o. sync .o.
!               flx .o. iflx .o. oflx .o. aflx .o. tagdel .o.
!               p1 .o. p2 .o. p3 .o. p4 .o. p5 .o. p6 .o.
!               s1 .o. s2 .o. s3 .o. s4 .o. s5                ] ; # QQ
```