```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                         %
%      File:          ch2_polish.dtr                                      %
%      Purpose:       polish nouns                                        %
%      Author:        Dunstan Brown, December 14, 2011                    %
%      Email:         d.brown@surrey.ac.uk                                %
%      Address:       SMG, University of Surrey, Guildford GU2 7XH        %
%      Documentation: 'Network Morphology', Brown & Hippisley 2012        %
%      Related Files: ch2_polishlex.dtr                                   %
%      Version:       15.03                                               %
%                                                                         %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% This fragment and the theoretical issues which it deals with are        %
% discussed in Brown and Hippisley (2012) Chapter 2 Section 2.3.2.3.      %
%                                                                         %
%                                                                         %
% NOTE ON THE REPRESENTATION OF THE FORMS                                 %
% Note that this is not in the standard Polish orthography, but uses      %
% a lower ascii transcription to represent the phonology, with the        %
% following correspondences (where upper case C stands for any consonant  %
% in the transcription or the orthography):                               %
%                                                                         %
%     Transcription     Writing                                           %
%     Ci                Cy                                                %
%     C'                Ci or Ć                                           %
%     e&                ę                                                 %
%     o&                ą                                                 %
%     ts                c                                                 %
%     ts'               ci or ć                                           %
%     u                 ó                                                 %
%     v                 w                                                 %
%     w                 ł                                                 %
%     x                 ch                                                %
%     z'                zi or ź                                           %
%     zz                ż (with a dot on top)                             %
%                                                                         %
%                                                                         %
% It should be noted that we have written ć (or ci before another vowel) as %
% as /ts'/, as this alternates with dental /t/and the alternation is one of %
% softening and affrication.                                              %
%                                                                         %
% The transcription system also assumes an automatic phonological rule    %
% that softens velars before /e/ (cf. /bikem/ and written bykiem).        %

% Load the lexicon of Polish nouns.                                       %

# load 'ch2_polishlex.dtr'


MOR_NOMINAL:
    <> == "<stem closed>"
    <mor hardness> == MOR_HARDNESS:< "<stem final type>" >
    <mor case> == nom
    <mor case masc person> == gen
    <mor sg acc> == ACCUSATIVE:< sg "<syn gender>" "<syn animacy>" >
    <mor pl acc> == ACCUSATIVE:< pl "<syn gender>" "<syn animacy>" >
    <mor pl inst> == "<stem>" "<mor vowel>"  _m'i
    <mor pl loc> == "<stem>" "<mor vowel>" _x
    <mor pl gen> == "<mor pl loc>".

MOR_ADJ:
    <> == MOR_NOMINAL
    <mor sg nom masc> == "<stem>" _i
    <mor sg nom fem> == N_II
    <mor sg nom neuter> == "<stem>" _e
    <mor sg acc fem> == "<stem>" _o&
    <mor sg gen> == "<stem>" _ego
    <mor sg gen fem> == "<stem>" _ej
    <mor sg dat> == "<stem>" _emu
    <mor sg dat fem> == "<mor sg gen fem>"
    <mor sg inst> == "<stem>" _im
    <mor sg inst fem> == N_II
    <mor sg loc> == "<mor sg inst>"
    <mor sg loc fem> == "<mor sg dat fem>"
    <mor pl nom> == PL_NOM2:<>
    <mor pl dat> ==  "<stem>" _im
    <mor vowel> == _i.

MOR_NOUN:
    <> == MOR_NOMINAL
    <mor sg voc> == "<mor sg loc>"
    <mor sg gen> == "<stem>" _i
    <mor sg dat> == "<mor sg loc>"
```

```
    <mor sg loc> == "<stem i>" _e
    <mor pl voc> == "<mor pl nom>"
    <mor pl nom> == PL_NOM1:< "<mor hardness>" "<syn gender>" >
    <mor pl gen> == MGP:< "<mor hardness>" >
    <mor pl dat> == "<stem>" _om
    <mor vowel> == _a.

N_O:
    <> == MOR_NOUN
    <mor sg gen> == GENITIVE:< "<syn gender>" "<syn animacy>" >
    <mor sg inst> == "<stem>" _em
    <mor sg loc> == SG_LOC:< "<stem final type>" >.


N_I:
    <> == N_O
    <mor pl gen> == "<stem>" _uv
    <mor sg dat> == "<stem>" _ov'i
    <mor gender> == masc.

N_IV:
    <> == N_O
    <mor sg nom> == "<stem>" _o
    <mor sg voc> == "<mor sg nom>"
    <mor sg dat> == "<stem>" _u
    <mor pl nom> == "<stem>" _a
    <mor gender> == neuter.

N_A:
    <> == MOR_NOUN
    <mor sg acc> == "<stem>" _e&
    <mor sg inst> == "<stem>" _o&
    <mor gender> == fem.

N_II:
    <> == N_A
    <mor sg nom> == "<stem>" _a
    <mor sg voc> == "<stem>" _o.

N_III:
    <> == MOR_NOUN
    <mor hardness> == soft
    <mor sg loc> == "<mor sg gen>"
    <mor sg inst> == N_II
    <mor gender> == fem.

N_V:
    <> == N_A
    <mor hardness> == soft
    <mor sg nom> == "<mor sg gen>"
    <mor sg loc> == N_III
    <mor pl gen> == "<stem>".

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                           %
%                       LEXEMIC HIERARCHY                                   %
%                                                                           %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

NOMINAL:
    <> == undefined
    <syn> == SYNTAX
    <mor> == "<declensional_class>"
    <stem final type> == "<root final type>" %velar and so on
    <stem> == "<root>" "<root final shape>".

ADJ:
    <> == NOMINAL
    <declensional_class> == MOR_ADJ:<mor>.


% The node NOUN (see below) within the Lexemic Hierarchy: the value for     %
% <sem sex> is undifferentiated (fourth equation). The value for            %
% <sem sex> is used to determine the semantic animacy of a noun at the      %
% node ANIMACY (seventh equation). The default undifferentiated value will  %
% lead to the noun being assigned the value 'inanimate'. (See example 77 in %
% Chapter 2.) At NOUN also gender will be assigned in the first instance on %
% the basis of syn animacy. By default syn animacy will be the same as      %
% sem animacy (which has three values: person, animate, inanimate).         %
%                                                                           %


NOUN:
    <> == NOMINAL
```

```
    <declensional_class> == DECLENSION:< "<sem sex>" >
    <syn cat> ==  n
    <sem sex> == undifferentiated
    <syn gender> == GENDER:< "<syn animacy>" >
    <syn animacy> == "<sem animacy>"
    <sem animacy> == ANIMACY:< "<sem sex>" >.


% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                         %
%                      INTERDEPENDENCIES                                  %
%                                                                         %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% The interdependency node ANIMACY (see below): the value for             %
% <sem sex> is used to determine the animacy of a noun at this node.      %
% The default undifferentiated value specified at NOUN will lead to the   %
% noun being assigned the value 'inanimate' (third equation).  If the noun %
% denotes a male it will be assigned the value 'person'.  Otherwise       %
% the noun will be assigned the value animate (which is also the          %
% exceptional case default). See the discussion of example (79) in        %
% Chapter 2.                                                              %
%                                                                         %
% A note on the exceptional case default for animacy: we argue that       %
% this accounts for both type 1 devirlized nouns (see example 80 in       %
% in chapter 2) and semantically inanimate nouns such as banan 'banana'.  %
% The lexical entry accesses the node ANIMACY directly but without        %
% specifying anything more about the value to be inherited. This means that %
% the least specific path <> is matched and the value 'animate' is used.  %
% For type 1 devirlized nouns it also means that the main gender will be  %
% masc, rather than masc person, as gender evaluation is normally         %
% dependent on animacy evaluation.                                        %

ANIMACY:
    <> == animate
    <male> == person
    <undifferentiated> == inanimate.

DECLENSION:
    <male> == N_I:<mor>
    <female> == N_II:<mor>.

PL_NOM1:
    <> == "<stem>" _i
    <soft> == PL_NOM2
    <hard masc person> == PL_NOM2:<masc person>.

PL_NOM2:
    <> == "<stem>" _e
    <masc person> == "<stem i>" _i.

ACCUSATIVE:
    <sg> == "<mor sg "<mor case>" >"
    <pl> == "<mor pl "<mor case>" >"
    <sg masc animate> == <sg masc person>.

% The interdependency node GENDER (see below): this assigns genders       %
% according to semantics and morphological information in line with       %
% (83) in Chapter 2. Recall that this node is accessed in the normal case %
% via the NOUN node, where the value for <syn animacy> has been evaluated. %
% As the values for this are 'person', 'animate' or 'inanimate', in all   %
% cases of inheritance via the NOUN node the evaluable path will match with %
% the first equation <> == < "<sem sex>" >, because the other LHS paths   %
% do not begin with 'person', 'animate' or 'inanimate'. This means that the %
% values for <sem sex>  will be extended by the values for <syn animacy>  %
% inserted by the evaluation at NOUN. We therefore get the following      %
% implict combinations by evaluation: male person; male animate;          %
% female person; female animate; undifferentiated. The last of these will %
% now match with the second equation and gender is assigned by declension %
% (the declension's value for <mor gender>).  (Recall that sex            %
% undifferentiated entities are specified as inanimate at the node        %
% ANIMACY.) This is how we get the normal case default for inanimates,    %
% listed in (83) as masc/fem/neut, depending on declension. Male animates %
% will match with the path <male> and be assigned masc. Female animates   %
% will match with the path <female> and be assigned fem. Male persons will %
% match with the path <male person> and be assigned masc person, where the %
% RHS path <male> will inherit the value masc, which will be combined with %
% person. These account for the normal case for male animates, male humans, %
% female animates and female humans.                                      %
%                                                                         %
% How the exceptional case default works: because the lexical entry       %
% accesses (i.e. inherits from) the node GENDER directly without          %
% specification of the path, the least specific path involved is accessed. %
% This is the one which requires <sem sex> to be evaluated, of course.    %
```

```
% The effect of this is to circumvent the evaluation of animacy at the      %
% node NOUN, and so a noun which denotes male humans will not use the       %
% value 'person' and so will just use the attribute 'male' on the LHS.      %
% The effect is that it will then be assigned masc rather than masc person. %
% This accounts for the 'devirlization' type 2 where the main gender        %
% switches from 'masc person' to 'masc' but the values for animacy remains  %
% as in the normal case. See the discussion of example (82) in Chapter 2.   %


GENDER:
    <> == < "<sem sex>" >
    <undifferentiated> == "<mor gender>"
    <male> == masc
    <female> == fem
    <male person> == <male> person.

GENITIVE:
    <> == "<stem>" _a
    <masc inanimate> == N_IV:<mor sg dat>.

SG_LOC:
    <> == MOR_NOUN:<mor sg loc>
    <velar> == N_IV:<mor sg dat>.

MGP:
    <> == "<stem>" _i
    <hard> == "<mor>".

MOR_HARDNESS:
    <> == hard
    <soft consonant> == soft.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                             %
%                      MORPHONOLOGY                                           %
%                                                                             %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %


CONS:
    <> ==
    <root final type> == hard consonant
    <root final grade i> == _' .

STOP:
    <> == CONS
    <root final shape> == "<root final stop>"  "<root final affric>"
                          "<root final grade>".

FRIC:
    <> == CONS
    <root final shape> == "<root final fric>" "<root final grade>".

I_II_STOP:
    <> == STOP
    <root final grade i i i> == .

I_II_FRIC:
    <> == FRIC
    <root final grade i i i> == _z.

FRIC_STOP:
    <> == CONS
    <root final shape> == FRIC STOP.

I_II_FRIC_STOP:
    <> == FRIC_STOP
    <root final grade i i i> == I_II_FRIC.

T:
    <> == I_II_STOP
    <root final stop> == _t
    <root final affric i> == S:<root final fric i>.

D:
    <> == I_II_STOP
    <root final stop> == _d
    <root final affric i> == Z:<root final fric i>.

S:
    <> == I_II_FRIC
    <root final fric> == _s.

Z:
```

```
    <> == I_II_FRIC
    <root final fric> == _z.

ST:
    <> == I_II_FRIC_STOP
    <root final stop> == T
    <root final affric> == T
    <root final fric> == S.

ZD:
   <> == I_II_FRIC_STOP
   <root final stop> == D
   <root final affric> == D
   <root final fric> == Z.

P:
    <> == STOP
    <root final stop> == _p.

B:
    <> == STOP
    <root final stop> == _b.

F:
    <> == FRIC
    <root final fric> == _f.

V:
    <> == FRIC
    <root final fric> == _v.


M:
    <> == STOP
    <root final stop> == _m.

N:
    <> == STOP
    <root final stop> == _n.

W:
    <> == CONS
    <root final> == _w
    <root final shape i> == _l.

R:
    <> == CONS
    <root final> == _r
    <root final shape i> == _zz.

X:
    <> == CONS
    <root final> == _x
    <root final type> == velar consonant
    <root final shape i> == _sz.


K:
    <> == I_II_STOP
    <root final type> == velar consonant
    <root final stop> == _k
    <root final shape i> == _ts
    <root final shape i i> == _cz.

G:
    <> == I_II_STOP
    <root final type> == velar consonant
    <root final stop> == _g
    <root final shape i> == _dz
    <root final shape i i> == _zz.

SK:
    <> == I_II_FRIC_STOP
    <root final type> == velar consonant
    <root final shape i> == _s K
    <root final shape i i> == X K.

ZG:
    <> == I_II_FRIC_STOP
    <root final type> == velar consonant
    <root final shape i> == _z G
    <root final shape i i> ==  _zzdzz.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
```

```
%                                                                       %
%                       VOWELS                                          %
%                                                                       %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

A_E:
    <root vowel> == _a
    <root vowel i> == _e.

O_U:
    <root vowel> == _o
    <root vowel closed> == _u.

E:
    <root vowel> ==
    <root vowel closed> ==  _'_e.

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                       %
%                       DUMMY SYNTAX                                    %
%                                                                       %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

SYNTAX:
    <> ==
    <syn form> == "<mor>"
    <syn head> == "<syn>"
    <syn $number $case / noun> ==
      ' [ '  <syn form $number $case> ' ] '
    <syn $number $case / adj noun> ==
      ' [ '  "Novi:<syn form $number $case <syn head gender> >"
       <syn $number $case / noun> ' ] '
    <syn $number acc / adj noun> ==
      ' [ ' "Novi:<syn form $number acc <syn head gender> <syn head animacy> >"
    <syn $number acc / noun> ' ] '.

# show
    <gloss>
    <syn sg nom / noun>
    <syn sg nom / adj noun>
    <syn sg voc / noun>
    <syn sg acc / noun>
    <syn sg acc / adj noun>
    <syn sg gen / noun>
    <syn sg gen / adj noun>
    <syn sg dat / noun>
    <syn sg dat / adj noun>
    <syn sg inst / noun>
    <syn sg inst / adj noun>
    <syn sg loc / noun>
    <syn sg loc / adj noun>
    <syn pl nom / noun>
    <syn pl nom / adj noun>
    <syn pl voc / noun>
    <syn pl acc / noun>
    <syn pl acc / adj noun>
    <syn pl gen / noun>
    <syn pl gen / adj noun>
    <syn pl dat / noun>
    <syn pl dat / adj noun>
    <syn pl inst / noun>
    <syn pl inst / adj noun>
    <syn pl loc / noun>
    <syn pl loc / adj noun>
    <syn gender>
    <syn animacy>.

# hide
    MOR_NOUN N_A N_O N_I N_II N_III N_IV N_V NOUN ANIMACY DECLENSION GENDER
    ACCUSATIVE GENITIVE PL_NOM1 PL_NOM2 ADJ MOR_ADJ CONS STOP FRIC I_II_STOP
    I_II_FRIC I_II_FRIC_STOP MOR_NOMINAL NOMINAL SYNTAX Novi T D S Z ST ZD P
    B F V M N W R X K G SK ZG SG_LOC FRIC_STOP A_E E O_U MGP MOR_HARDNESS.
```