3 Strojový překlad – různé

3.1 Transfer

Pravidla transferu lze rozdělit na:

- lexikální
- strukturní: obecná nebo specifická (jejichž aplikace závisí např. na morfologických kategoriích nebo na lexikálním obsazení prvků struktury)

Pravidla transferu mohou být užita na:

- stromovém grafu
- sestavě rysů
- souboru kategorií popisujících nejmenší jednotky roviny

Pravidla transferu mohou mít přístup:

- pouze k části reprezentace (např. k hloubkové struktuře)
- ke všem údajům reprezentace (včetně povrchového slovosledu, morfologických kategorií)

Při aplikaci pravidel je třeba dodržovat určitý ppostup, který respektuje:

- rekursivní vlastnosti representace (výrazu jazyka): transfer postupuje od nejvyšší úrovně struktury k nižším úrovním
- míru obecnosti pravidel transferu: specifičtější pravidla se uplatňují přednostně

Ukázka slovníku systému APAČ

```
ARM == N(RAMEN(NO2),*C,*PART).
ARRANGE == V(USPOR3A2D(50N),IL,*FIN,0).
ARRANGEMENT == N(USPOR3A2DA2NI2(N12),*A,*RACT,*NVT,*UNCC,*METH,0).
ARRAY == ARRANGEMENT.
ARRIVE == V(DOSPI2V(50I33,DOSPE3J),I,*FIN,*PRO,AT(K3),7,0).
ARTICLE == N(C3LA2NK(M04),G,ON(06),?(06)).
ASCERTAIN == FIND.
ASCRIBE == V(PR3IPISUJ(30N),IRI,*IO,*FIN,*PRO,0).
ASK == V(Z3A2D(50N),IJ,*FIN,*SCE,FOR(04),*PRO,0).
-- == V(PT(51N),1(*H,/),2(*SS,*A,*H,/),*PAT(2),*RFXX,*AUTH,ABOUT(NA4),AFTER(NA4),*QUOT,0).
ASPECT == N(HLEDISK(N08),*A,*METH,0).
ASSEMBLE == V(MONTUJ(30N),1(M),2(*OB,T),*FIN,*VAD2).
-- == V(SHROMA2Z3D(43P30,SHROMAZ3D3UJ),1(P),2(I,*PART,*INST),*RFX,*FIN,*VAD2).
ASSEMBLY == N(SESTAVENI2(N12),B,*NVT,0).
ASSERT == V(PROSAZUJ(30N),IR,*THT,*AUTH,0).
ASSESS == V(ODHADUJ(30I21,ODHADN),IL,*THT,AT(NA4),5,*AUTH,0).
ASSESSMENT == N(ODHAD(M03),*A,*RACT,*NVT,*UNCC).
ASSIGN == V(URC3(42N),IRI,*IO,*FIN,*PRO,0).
ASSIGNMENT == N(STANOVENI2(N12),F,*RACT,*METH,*NVT,0).
ASSIST == V(POMA2H(50N),IN,*PAT(3),*ACIF,IN(PR3I6),0).
ASSISTANT == N(ASISTENT(A22),*H,*NA,0).
-- == AD(POMOCNE2(3),M,*NA,*OGRD,0).
```

Ukázka reprezentace věty v systému APAČ

```
-001-$sn(13)+.($gs(c,4e,s))+the+program+reads+the+data+qpiod+>-002-

-001-$($sn(13),$gs(c,4e,s))+
      s(v(c3t(sl(15i15,#pr3e),/,*sg),
         n(program(l,$ag,*def,*a,jm(m01),/,*sg)),
         n(dat(r,$pat,*def,*sp,*a,jm(n15),/,*pl))
      )) +.-002-

-001-$sn(13)+$gs(c,4e,s)+program+c3te+data+.-002-
```

Ukázka postupu překladu jedné věty v systému APAČ

```
/-1- $SN(13) + $GS(C,4E,S) + < + THE + METHOD + TAKES + INTO + ACCOUNT + THE + COMPRESSIBILITY
+ OF + THE + AIR + AS + WELL + AS + THE + FRICTIONAL + LOSSES + IN + THE + PIPE($GT) + . + >-2-

PHASE NO.   2
-001- .($sn(13),$gs(c,4e,s))
+ art(*def)
+ n(metod(f01),*a,*meth,*strm,0)
+ v(take,il,*rel1,from(od2),aptc(off),aptc(out),yptc(u2c3t,of),aptc(up),eptc(place),
  xptc(into,u2vah),xptc(into,u2c3t),5,0) + #(s)
+ p(into)
-002-
  -002- v(u2c3tuj(30n),io,*vb,*tht,aptc(for),*caus,*pro,*vad2) -003-
  -002- n(u2c3t(m10),*vb,g) -003-
-003- art(*def)
+ n(stlac3itelnost(f09),e,*scl,0)
+ p(of)
+ art(*def)
+ n(vzduch(m11),*c,*mtr,*lq,compressor)
+ conj(aswas)
+ art(*def)
+ ad(tr3eci2(9),t,*0grd,*mnr)
+ n(ztra2t(f01),g)
+ #(s)
+ p(in)
+ art(*def)
+ n(trubic(f12),*c,*inst,0)
+ *(.) -004-

PHASE NO.   5
-001- .($sn(13),$gs(c,4e,s))
+ art(*def)
+ n(metod(jm(f01),/,*sg),;,*a,*meth,*strm,0)
+ v(takeintou2c3t(/,*sg),;,1(*a,*c,*h,/),2(*ob,*a,*c,/),#,*rel1,from(od2),aptc(off),aptc(out),
  yptc(u2c3t,of),aptc(up),eptc(place),xptc(into,u2vah),*ph,5,0)
+ art(*def)
+ n(stlac3itelnost(*a,jm(f09),/,*sg),;,*scl,0,*a,*prop)
+ p(of)
+ art(*def)
+ n(vzduch(jm(m11),/,*sg),;,*c,*mtr,*lq,compressor)
+ conj(aswas)
+ art(*def)
+ ad(tr3eci2(pj(9),/),;,1(*c,/),*0grd,*mnr)
+ n(ztra2t(*a,*c,jm(f01),/,*pl),;,*a,*c,0)
+ p(in)
```

```
+ art(*def)
+ n(trubic(jm(f12),/,*sg),;,*c,*inst,0)
+ . -002-

PHASE NO.   12
-001- .
+ $($sn(13),$gs(c,4e,s))
+ n(metod(*def,*a,jm(f01),/,*sg),;,*meth,*strm,0,*a)
+ v(takeintou2c3t(/,*sg),;,1(*a,*c,*h,/),2(*ob,*a,*c,/),#,*rel1,from(od2),aptc(off),aptc(out),
  yptc(u2c3t,of),aptc(up),eptc(place),xptc(into,u2vah),*ph,5,0)
+ n(stlac3itelnost(*def,*a,jm(f09),/,*sg),
  n(vzduch(r,$atr(2),*def,*c,jm(m11),/,*sg)),;,p(of),*scl,0,*a,*prop,#10,0)
+ conj(aswas) + n(ztra2t(*def,*a,*c,jm(f01),/,*pl),ad(tr3eci2(l,$atr,pj(9),/)),;,*a,*c,0,0)
+ p(in)
+ n(trubic(*def,*c,jm(f12),/,*sg),;,*inst,0,*c)
+ . + $ -002-

PHASE NO.   16
-001- $($sn(13),$gs(c,4e,s))
+ s(v(takeintou2c3t(/,*sg),
    n(metod(l,$ag,*def,*a,jm(f01),/,*sg)),
    n(coor(r,$pat,aswas,/,*pl),
      n(stlac3itelnost(*def,*a,jm(f09),/,*sg),
        n(vzduch(r,$atr(2),*def,*c,jm(m11),/,*sg))),
      conj(aswas),
        n(ztra2t(*def,*a,*c,jm(f01),/,*pl),
          ad(tr3eci2(l,$atr,pj(9),/)))),
          n(trubic(r,$adv(v6),*def,*c,jm(f12),/,*sg))))
+ . -002-

PHASE NO.   18
-001-
$($sn(13),$gs(c,4e,s),s)
+ n(metod(l,$ag,#r,*def,*a,jm(f01),*um,/,*sg))
+ v(takeintou2c3t(%8,/,*sg))
+ n(stlac3itelnost(r,$pat,#r,aswas,*cd,*def,*a,jm(f09),#m,/,*sg))
+ n(vzduch(r,$atr(2),*def,*c,jm(m11),#m,/,*sg))
+ conj(aswas)
+ ad(tr3eci2(l,$atr,pj(9),/))
+ n(ztra2t(r,$pat,#r,aswas,*crd,*def,*a,*c,jm(f01),#m,/,*pl))
+ $adv(v6)
+ n(trubic(r,$advb(v6),#r,*def,*c,jm(f12),*um,/,*sg))
+ . -002-

PHASE NO.   23
-001- $($sn(13),$gs(c,4e,s),s)
+ nm(m,e,t,o,d,*($subj,#r,*def,*a,*um,f01,/,1,f,*sg))
+ prs(b,e,r,*(%8,*sg,11,/,1,f,*sg)) + vt(v) + vt(u2vahu)
+ nm(s,t,l,a,c,3,i,t,e,l,n,o,s,t,*(r,$pat,#r,aswas,*ccd,*def,*a,#m,*0,f09,/,4,f,*sg))
+ nm(v,z,d,u,c,h,*(r,$atr(2),*def,*c,#m,m11,/,2,m,*sg))
+ i + adj(t,r,3,e,c,i,2,*(l,$atr,9,/,4,f,*pl))
+ nm(z,t,r,a,2,t,*(r,$pat,#r,aswas,*ccrd(4,m,*pl),*def,*a,*c,#m,*atr,f01,/,4,f,*pl))
+ v + nm(t,r,u,b,i,c,*(r,$advb(v6),#r,*def,*c,*um,*0,f12,/,6,f,*sg))
+ . -002-

PHASE NO.   25
-001- $sn(13) + $gs(c,4e,s)
+ metoda + bere + v + u2vahu + stlac3itelnost + vzduchu + i + tr3eci2 + ztra2ty + v + trubici + . -002-
```