appoggiarsi a condizioni informatiche di base, come poteva essere il caso di molti dei nostri studenti; ma tali condizioni minime non sono più quelle di oggi, dieci anni dopo».

Dieci anni dopo, appunto, dovendo preparare nuovi testi per la futura ed ampliata versione del CT (la Ver. 2.0 ed oltre), abbiamo naturalmente seguito ben diverso approccio, appoggiandoci invece ad uno script GAWK (preparato da M. Tomatis, cfr. *infra*) eventualmente da ritarare cammin facendo, e da correggere manualmente laddove l'informazione dell'OVI fosse meno sistematica (come nella paragrafatura, cfr. §§ 8.2.3.2, 8.4.2-3 e 8.5.3.2).

8.9.1 UNO SCRIPT PARAMETRICO PER LA TOKENIZZAZIONE ED IL MARKUP. [MT] Al fine di automatizzare e rendere il più agevole possibile l'intera sequenza di sostituzioni descritta nei §§ 8.3-8.7, è stato predisposto un piccolo programma AWK (`ovi-ct_tm`) che, assumendo in ingresso il testo originale dell'OVI, codificato nel formato utilizzato in GATTO, si fa carico della trasformazione del testo nel formato adottato dal CT, generando in tal modo il file che dovrà successivamente subire il processo di etichettatura morfosintattica.

L'introduzione di un programma appositamente creato con la sola funzione di gestire la corretta manipolazione del file di origine si rivela una strategia vincente in quanto, oltre a evitare potenziali errori introdotti dalle numerose operazioni manuali, permette, mediante l'utilizzo delle espressioni regolari, di operare sostituzioni più dirette e mirate, senza quindi dover ricorrere a strategie di sostituzione intermedie unicamente finalizzate a salvaguardare quelle combinazioni di caratteri che debbono essere necessariamente lasciate invariate.

8.9.1.1 IL LISTATO. Il listato di base, scritto in GAWK, e predisposto per questa procedura, è in sostanza il seguente:

```
{
nf = ""
if ($0 !~ /[\'\,\;\:\.\?\!\^\¨\*]/)
        {
        print $0
        next
        }
else
if ($0 ~ / ¨ /)
        gsub (/ ¨ /, "· ", $0)
else
if ($0 ~ / ¨/)
        gsub (/ ¨/, "· ", $0)
else
if ($0 ~ /\* \*/)
        gsub (/\* \*/, "**", $0)
else
while (nf <= NF)
        {
        if ($nf ~ /[a-zA-Z]'[a-zA-Z]/)
                sub (/'/, "' ", $nf)
        else
        if ($nf ~ /[a-zA-Z],$/)
                sub (/,/, " ,", $nf)
        else
        if ($nf ~ /[a-zA-Z];$/)
                sub (/;/, " ;", $nf)
        else
```

```
        if ($nf ~ /[a-zA-Z]:$/)
                sub (/:/, " :", $nf)
        else
        if ($nf ~ /[a-zA-Z]\?$/)
                sub (/\?/, " ?", $nf)
        else
        if ($nf ~ /[a-zA-Z]\!$/)
                sub (/\!/, " !", $nf)
        else
        if ($nf ~ /[a-zA-Z]\^$/)
                sub (/\^/, " ^", $nf)
        else
        if ($nf ~ /[a-zA-Z]\.$/)
                sub (/\./, " .", $nf)
        else
        if ($nf ~ /\(/ && $nf !~ /\&\(/)
                sub (/\(/, " (", $nf)
        else
        if ($nf ~ /\)/ && $nf !~ /\&\)/)
                sub (/\)/, " )", $nf)
        else
        if ($nf ~ /\[/ && $nf !~ /\&\[/)
                sub (/\[/, " [", $nf)
        else
        if ($nf ~ /\]/ && $nf !~ /\&\]/)
                sub (/\]/, " ]", $nf)
        else
        if ($nf ~ /\&K/)
                sub (/\&K/, "&¦", $nf)
        else
        if ($nf ~ /\&k/)
                sub (/\&k/, "&¦", $nf)
        else
        if ($nf ~ /\&C/)
                sub (/\&C/, "&©", $nf)
        else
        if ($nf ~ /\&c/)
                sub (/\&c/, "&®", $nf)
        nf++
        }
print $0
}
```

Tav. 101: Listato di ovi-ct_tm, script in GAWK.

Data la semplicità del programma, che mediante una opportuna verifica preliminare si limita a operare esclusivamente sulle linee di testo che presentano necessità di intervento, dal punto di vista dell'analisi del funzionamento ci limiteremo a esaminare l'operatività delle espressioni regolari adottate, illustrandone alcuni esempi.

8.9.1.2 TOKENIZZAZIONE DEGLI ELEMENTI DI PUNTEGGIATURA. Il primo caso che prendiamo in esame è quello della tokenizzazione degli elementi di punteggiatura; problema che trova immediata soluzione mediante la regola seguente: