

Kodowanie tekstów polskich w systemach komputerowych*

Janusz S. Bień[†]

4 stycznia 1999[‡]

1 Wstęp

W niniejszym artykule przedstawiono podstawowe pojęcia kodowania tekstów oraz omówiono *wszystkie* aktualnie obowiązujące polskie normy dotyczące tej problematyki, a także najważniejsze standardy międzynarodowe.

2 Podstawowe pojęcia kodowania tekstów

2.1 Klasyfikacja tekstów

Pojęcie tekstu traktujemy jako pierwotne, którego nie będziemy tutaj definiować, wyróżnimy natomiast dwa podstawowe typy tekstów: *teksty fizyczne* i *teksty elektroniczne*. Za charakterystyczną cechę tekstów elektronicznych

*Referat wygłoszony na konferencji MULTIMEDIA W NAUCZANIU JĘZYKA RODZIMEGO JAKO OBCEGO (Szkoła Języka i Kultury Polskiej, Uniwersytet Śląski, Katowice, 7-8 grudnia 1998) i opublikowany z niewielkimi skrótami w kwartalniku *Postscriptum* nr 27-29 (jesień 1998 — wiosna 1999), s. 4-27 (ISSN 1427-0501). Za zgodą organizatorów i redakcji niniejszy pełny tekst referatu był dostępny pod adresem ftp://ftp.mimuw.edu.pl/pub/polszczyzna/ogonki/katow98.*, obecnie jest dostępny pod adresem <http://www.mimuw.edu.pl/~jsbien/publ/Kodtp98/>.

[†]Dr hab. J. S. Bień, prof. UW jest pracownikiem Katedry Lingwistyki Formalnej Uniwersytetu Warszawskiego. W latach 1998-2003 był kierownikiem Zakładu Zastosowań Informatycznych Instytutu Orientalistycznego Uniwersytetu Warszawskiego. W Instytucie Informatyki UW, w którym pracował do czerwca 1998 r., prowadził m.in. wykłady monograficzne pt. *Wybrane standardy przetwarzania tekstu*. W latach 1992-1993 był członkiem Normalizacyjnej Komisji Problemowej ds. Informatyki, w latach 1997-1999 Normalizacyjnej Komisji Problemowej nr 242 ds. Informacji i Dokumentacji, a w latach 1999-2001 Normalizacyjnej Komisji Problemowej nr 170 ds. Terminologii Informatycznej i Kodowania Informacji Polskiego Komitetu Normalizacyjnego. Adres elektroniczny: jsbien@uw.edu.pl (lub jsbien@mimuw.uw.edu.pl).

[‡]Poprawki i uzupełnienia wprowadzono 28 października 2002 r. Informacje o autorze i adresy internetowe uaktualniono 25 listopada 2005 r.

uznajemy fakt, że nie zmieniają one swojej tożsamości przy zmianie ich nośnika fizycznego: ten sam tekst elektroniczny może być zapisany na dyskietce, twardym dysku komputera lub na płycie CD-ROM i mimo to pozostaje tym samym tekstem. Tymczasem w przypadku tekstów fizycznych każde skopiowanie bez względu na zastosowaną technikę wprowadza mniejsze lub większe zmiany i zniekształcenia — w rezultacie po wielokrotnym kopiowaniu tekst może np. znacznie zmniejszyć swoją czytelność.

Teksty fizyczne dzielimy m.in. na

- teksty mówione,
- teksty pisane ręcznie,
- teksty drukowane.

Do wykazu tego można dołączyć teksty gestykulowane, stosowane przez głuchoniemych. Zasadniczym jednak powodem przedstawienia tego podziału jest podkreślenie istotnej różnicy między tekstami pisanymi ręcznie a tekstami drukowanymi — co widać wyraźnie np. przy wprowadzaniu tekstu do komputera za pomocą optycznego rozpoznawania znaków (OCR, ang. *Optical Character Recognition*); są one przez wielu lingwistów zbyt pochopnie traktowane łącznie jako jednorodna klasa tekstów pisanych.

Niektóre teksty elektroniczne są po prostu mniej lub bardziej wiernym zapisem pewnych tekstów fizycznych — możemy mówić wtedy o tekście pierwotnym (fizycznym) i wtórnym (elektronicznym). Mamy wówczas do czynienia z *kodowaniem akustycznym*, tj. z cyfrowym zapisem dźwięku tekstu mówionego, lub z *kodowaniem wizualnym*, tj. z cyfrowym zapisem informacji wizualnej. Przykładem zastosowania tej ostatniej techniki może być *Elektroniczny przedruk SŁOWNIKA JĘZYKA POLSKIEGO pod red. W. Doroszewskiego* ([23], por. [9]).

Najważniejszym typem kodowania tekstów jest jednak *kodowanie symboliczne*, gdzie pewnym elementom składowym tekstów (w przypadku alfabetycznych systemów pisma będą to m.in. litery i inne znaki pisarskie czy drukarskie) przyporządkowuje się pewne reprezentacje w postaci liczb lub ciągów bitów, a następnie zapisuje się tekst jako ciąg reprezentacji jego elementów składowych. Typowy tzw. *plik tekstowy* zapisany w komputerze stanowi dobry przykład tego typu kodowania; każdy, kto kiedykolwiek musiał się zastanowić, jak reprezentowane są polskie litery w otrzymanym pliku, powinien w pełni uświadamiać sobie umowność i konwencjonalność symbolicznego kodowania tekstów.

Dla pełności obrazu warto wspomnieć, że nie zawsze wymienione typy kodowania występują w postaci czystej. Ponieważ mieszane typy kodowania są praktycznie zawsze podporządkowane własnościom konkretnych urządzeń lub metod przetwarzania, określam je terminem *kodowanie technologiczne*. Przykładem mogą być tutaj formaty Postscript i PDF (*Portable Document*

Format), które — dla ekonomicznego przechowywania dokumentów przeznaczonych do drukowania lub oglądania na ekranie — łączą kodowanie wizualne z symbolicznym.

Rozwój techniki informatycznej zdeaktualizował postulowany przeze mnie wcześniej podział tekstów na naturalne i kodowane (por. [3], s. 254 i [4] s. 9), ponieważ dużo tekstów — np. listy elektroniczne — istnieje tylko w postaci kodowanej, a trudno im odmawiać cechy naturalności.

2.2 Teksty czyste i wzbogacone

Charakterystyczną cechą tekstów elektronicznych stosujących w mniejszym lub większym stopniu kodowanie symboliczne jest możliwość wyboru stopnia dokładności i szczegółowości, z jaką dany tekst jest zapisany. W terminologii anglosaskiej przyjęło się ten fakt odzwierciedlać rozróżniając *plain text* (dosłownie *zwykły tekst*) z jednej strony i *fancy text* (dosłownie *tekst wymyślny*) lub *rich text* (dosłownie *tekst bogaty*); dla *plain text* dość szeroko stosowane — również przeze mnie — jest tłumaczenie *czysty tekst*, termin *rich text* proponuję tutaj tłumaczyć jako *tekst wzbogacony*.

Standard UNICODE, którym zajmujemy się dokładniej w punkcie 7, stwierdza ([61], s. 2-7), że *czysty tekst* reprezentuje podstawową treść tekstu w wymiennej — to znaczy nie związanej z konkretnym oprogramowaniem — postaci. W konsekwencji czysty tekst nie ma określonego swojego wyglądu — kwestie kroju i wielkości znaków, podziału na wiersze i strony, koloru, odsyłaczy hipertekstowych, a nawet języka, w którym tekst lub jego fragmenty są zapisane, należą do domeny *tekstu wzbogaconego*. Dla ścisłości trzeba jednak dodać, że od powyższej zasady UNICODE wprowadza pewien wyjątek — jeśli tekst zawiera wielokrotnie zagnieżdżone fragmenty pisane od lewej do prawej (np. po angielsku) i od prawej do lewej (np. po hebrajsku lub arabsku), to dla poprawnego przedstawienia i odczytania tego tekstu niezbędna jest pewna dodatkowa informacja, która choć odnosi się do wyglądu, jest traktowana jako składnik tekstu czystego.

Zanim skoncentrujemy naszą uwagę na kodowaniu tekstu czystego, warto podać kilka przykładów tekstów wzbogaconych. Oprócz wspomnianych wyżej tekstów w formacie Postscript i PDF można tutaj dodać „rodzime” (ang. *native*) formaty komercyjnych edytorów takich jak MS Word czy WordPerfect, opracowany przez Microsoft format RTF (ang. *Rich Text Format*) itp. Szczególnie ciekawe są jednak takie teksty wzbogacone, które zawierają dodatkową informację o strukturze i innych, niewizualnych ich własnościach. Informacje takie w terminologii angielskiej noszą nazwę *abstract* lub *generalized markup*; historycznie *marking up* oznaczało nanoszenie przez redaktora technicznego uwag dla drukarni na maszynopisie wydawniczym, zaś rzeczownik *markup* oznaczał oczywiście wynik tego procesu. Najbliższymi polskimi odpowiednikami *to mark up* i *markup* są *adiustować* i *adiustacja*; jak się wydaje, pomimo pewnych wahań te polskie terminy są już powszechnie ak-

ceptowane również w ich nowym znaczeniu.

Najczęściej stosowaną postacią adiustacji jest SGML czyli *Standard Generalized Markup Language (Standardowy Uogólniony Język Adiustacyjny)* zdefiniowany w międzynarodowej normie ISO 8879 ustanowionej w 1986 r., a ostatnio uzupełnionej o poprawki i dodatki ([33], [25], por. także [7] i [11]); w 1998 r. Polski Komitet Normalizacyjny podjął decyzję o rozpoczęciu prac nad ustanowieniem polskiego odpowiednika ISO 8879 jako Polskiej Normy¹. Z SGML wywodzi się HTML (*Hypertext Markup Language* — Hipertekstowy Język Adiustacyjny) powszechnie stosowany w Internecie i jego prawdopodobny następca XML (*Extensible Markup Language* — Rozszerzalny Język Adiustacyjny). Dla tematyki niniejszej konferencji szczególnie istotny jest inny, niedawno uaktualniony, również wywodzący się z SGML standard, mianowicie HyTime ([37]), przeznaczony do opisu dowolnie złożonych dokumentów hipertekstowych i multimedialnych; choć jest to norma ISO, tekst jej jest dostępny bezpłatnie w Internecie pod adresem <http://www.hytime.org>. SGML znalazł również zastosowanie w badaniach lingwistycznych, czego przykładem jest specyfikacja TEI (*Text Encoding Initiative*) i pochodne — patrz np. [59]; przykład polskiego tekstu zadiustowanego w SGML można znaleźć na płycie CD-ROM ([24]) przygotowanej z udziałem Uniwersytetu Warszawskiego w ramach międzynarodowego projektu TELRI (*TransEuropean Language Resources Infrastructure* — Transeuropejska Infrastruktura Zasobów Językowych).

2.3 Znaki i kody

Taki element danego systemu pisma, który dla potrzeb kodowania tekstu traktujemy jako podstawowy, nazywamy *znakiem* (ang. *character*). W ogólny sposób nie można pojęcia znaku zdefiniować ściślej, co więcej, w wielu wypadkach jego znaczenie odbiega od potocznej intuicji. Wyróżnienie podstawowych elementów pisma (czyli znaków pisarskich lub czcionek — angielski termin *character* oznacza jedno i drugie) jest w dużym stopniu kwestią konwencji, wywodzącej się z tradycji lub uwarunkowań technicznych — np. niepodzielna dla nas litera „L” może być w systemie T_EX traktowana jako litera „L” oraz specjalny znak diakrytyczny.

Jeśli dysponujemy już asortymentem znaków, możemy im przyporządkować pewne reprezentacje binarne czyli ciągi bitów (w praktyce zapisywane przeważnie jako liczby dziesiętne, szesnastkowe lub ósemkowe); znak z przypisaną mu reprezentacją nazywamy *znakiem kodowym*, a na jego reprezentację mówimy krótko *kod znaku* (nie jest to oczywiście jedyne znaczenie terminu *kod*). Konkretny zestaw znaków (ang. *character set*) z reprezentacjami przypisanymi im zgodnie z pewną spójną konwencją możemy naturalnie nazywać *zestawem znaków kodowych*, osobiście wolę jednak określenie *kodowy*

¹Decyzja ta została później anulowana - uwaga dodana 25.11.2005 r.

zestaw znaków, które wydaje się zreczniejsze stylistycznie i lepiej oddaje fakt, że poszczególne znaki kodowe są podporządkowane wspólnej konwencji. Zgodnie z przyjętą praktyką na kodowy zestaw znaków możemy również mówić krócej *kod*, *kod znaków* lub *kod znakowy*.

Normy ISO poświęcone kodowaniu znaków zawierają wiele definicji związanych z tą problematyką pojęć, nie jest jednak oczywiste, czy definicje te są w pełni wewnętrznie spójne i praktycznie użyteczne. Nie ulega natomiast wątpliwości, że stwarzają one poważne problemy przy tłumaczeniu tych norm na język polski.

Podstawowa definicja znaku brzmi w oryginale (np. w normie ISO/IEC 8859-2, s. 1):

character: A member of a set of elements used for the organisation, control and representation of data

W normie PN-91/T-42115 definicja ta została przetłumaczona jako

znak — element zbioru służącego do organizacji i sterowania lub przedstawiania danych

zaś w normie PN-ISO/IEC 2022 z 1996 r. jako

znak: element zbioru stosowany do organizacji, sterowania i reprezentacji danych

Jak widać, tłumacze odmiennie zinterpretowali strukturę składniową oryginału („... zbioru używanego ...”, „element ...stosowany ...”), a jednocześnie żaden z nich nie próbował wiernie przetłumaczyć *A member of a set of elements*, bez czego niemożliwe jest — moim zdaniem — pełne oddanie sensu oryginału. Definicja w normie PN-91/T-42115 jest prawdopodobnie zaadoptowana z normy PN-93/T-42109/01 będącej polskim odpowiednikiem ISO/IEC 646:1991, stąd „tłumaczenie” *and przez lub* — w ISO/IEC 646 w tym miejscu jest rzeczywiście *or*; tłumaczenie *representation* jako *przedstawianie* należy uznać za błąd. Ponadto obaj tłumacze popełnili ten sam błąd gramatyczny („sterowania ... danych”). Jak się wydaje, znacznie bliższe intencjom oryginału jest następujące sformułowanie:

znak: jeden z elementów pewnego zbioru, którego elementy służą do organizacji danych, sterowania nimi i ich reprezentowania.

Kłopoty z przetłumaczeniem definicji znaku to tylko przedsmak problemów z tłumaczeniem następującej definicji występującej m.in. w normie ISO/IEC 8859-2:

coded character set; code: A set of unambiguous rules that establishes a character set and the one-to-one relationship between each character of the set and its coded representation.

(w normach ISO/IEC 646:1991 i ISO/IEC 2022:1994 zakończenie definicji brzmi nieco inaczej: ... *between the characters of the set and their bit combinations*).

W normie PN-93/T-42109/01 będącej tłumaczeniem ISO/IEC 646:1991 mamy

kodowany zestaw znaków, kod — ustalony według jednoznacznych zasad zestaw znaków i przyporządkowane im kombinacje bitowe w taki sposób, że określa się wzajemnie jednoznaczny związek między każdym znakiem zestawu i jego kombinacją bitową.

W normie PN-91/T042115 będącej tłumaczeniem ISO/IEC 8859-2:1987 mamy

zestaw kodowanych znaków; kod — zespół jednoznacznych reguł, które określają pewien zestaw znaków i wzajemnie jednoznaczny związek między każdym znakiem tego zestawu a jego kodowym odpowiednikiem.

W normie PN-ISO/IEC 2022, będącej — jak wskazuje jej symbol zbudowany zgodnie z nowymi zasadami — tłumaczeniem ISO/IEC 2022 (z 1994 r.) mamy natomiast

zestaw zakodowanych znaków; kod: Zbiór jednoznacznych reguł, za pomocą których jest ustanawiany zestaw znaków i wzajemnie jednoznaczny związek między znakami tego zestawu i ich kombinacjami bitowymi.

Norma ta jest dodatkowo opatrzona deskryptorami, wśród których znajdziemy (wyróżnienie moje): *0391035* — *zestaw znaków*, *0611660* — **zestaw znaków kodowych**, *0614355* — *reprezentacja kodowa*, natomiast w tekście normy znajdujemy również użyte terminy *kodowany zestaw znaków* (uwaga na s. 9), *zestaw kodowanych znaków* (s. 16 i następne).

Pozostawiając zainteresowanemu czytelnikowi ocenę wierności i poprawności stylistycznej przytoczonych wyżej tłumaczeń, chciałbym skoncentrować się na problemie tłumaczenia terminu *coded character set*. Ani z jego postaci, ani z treści jego definicji nie wynika jasno jego struktura — tj. czy kodowany jest znak czy zestaw; możemy jednak przyjąć w tym względzie za rozstrzygające francuskie tłumaczenie tego terminu (*jeu de caractères codés* występujące m.in. w francuskojęzycznych równoległych tytułach norm). Pozwala nam to wyeliminować jako błędne tłumaczenie *kodowany zestaw znaków*. Jeśli ograniczymy się do tłumaczeń, które w ten czy inny sposób pojawiły się już w polskich normach, to pozostają nam do wyboru trzy możliwości: *zestaw zakodowanych znaków*, *zestaw kodowanych znaków*, *zestaw znaków kodowych*.

Warto w tym momencie powiedzieć, że żadna z wymienionych norm nie definiowała osobno pojęcia *coded character*. Definicja ta pojawiła się dopiero w normie ISO/IEC 10646-1:1993, która prędzej czy później powinna stać się polską normą (formalny wniosek w tej sprawie został skierowany do Polskiego Komitetu Normalizacyjnego przez Normalizacyjną Komisję Problemową nr 246 do spraw Informacji i Dokumentacji w listopadzie 1998 r.):

coded character: A character together with its coded representation.

Coded character oznacza więc nie wynik procesu kodowania znaku, ale właśnie sam znak rozpatrywany jako element pewnego kodu — przesądza to moim zdaniem o tym, że jedynym adekwatnym tłumaczeniem tego terminu jest *znak kodowy*. Tłumaczenie to nie tylko dobrze oddaje znaczenie tego terminu, ale też pozwala zarezerwować czasowniki *kodować* i *zakodować* jako odpowiedniki angielskiego *to encode*, bez czego byłoby skrajnie trudno tłumaczyć na język polski takie angielskie teksty, w których np. mowa o tym, że znaki kodowe (*coded characters*) są na potrzeby poczty elektronicznej zakodowane (*encoded*) metodą „drukowalnego cytowania” (*quoted-printable*).

W konsekwencji powyższej decyzji właściwym tłumaczeniem terminu *coded character set* jest *zestaw znaków kodowych*; w sytuacjach, gdy nie obowiązuje stosowanie terminologii oficjalnie ustanowionych norm, można również posługiwać się proponowanym przeze mnie terminem *kodowy zestaw znaków*.

2.4 Glify i fonty

Jak się wydaje, termin *glyph* został wprowadzony do angielskojęzycznej literatury informatycznej przez normę ISO/IEC 9541-1:1991. Okazał się on bardzo pożyteczny, stąd potrzeba jego adaptacji do języka polskiego. Mając do wyboru pisownię taką, jak w słowie *hieroglif*, oraz taką, jak w słowie *glyf* oznaczającym pewien detal architektoniczny, zdecydowałem się na to pierwsze rozwiązanie. Oryginalna definicja z ISO/IEC 9541 brzmi następująco:

glyph: A recognizable abstract graphic symbol which is independent of any specific design.

a w moim swobodnym tłumaczeniu

glif: Abstrakcyjny symbol graficzny, którego kształt jest określony w stopniu pozwalającym na jego rozpoznanie i identyfikację, ale bez przesądzania konkretnych cech jego wyglądu.

Język polski nie dostarcza niestety — a może raczej na szczęście — dobrych przykładów na różnice między znakami i glifami. Po klasyczne przykłady musimy sięgnąć do języka angielskiego, w którym regułą jest stosowanie tzw. ligatur dla pewnych połączeń liter — m.in. sekwencja liter „ff” oraz

„i” drukowana jest jedną czcionką, której tzw. oczko nie jest prostym złożeniem tych liter, ponieważ „i” jest pozbawione kropki; jeśli abstrahujemy od wielkości i kroju tej czcionki, to otrzymamy glif ligatury „fi”.

Języki bardziej egzotyczne dostarczają bardziej wyrazistych przykładów: w języku arabskim kształt konkretnej litery zależy od tego, czy występuje ona na początku, w środku czy na końcu słowa, a jeszcze inny wygląd ma litera pisana samodzielnie — innymi słowy, w zależności od kontekstu litera jest pisana, drukowana lub wyświetlana jako jeden z czterech możliwych glifów.

Z pojęciem glifu jest ściśle związane pojęcie fontu (ang. *font*, w pisowni brytyjskiej *fount*). W przeciwieństwie do neologizmu *glyph*, słowo *font* jest tradycyjnym terminem typografii anglosaskiej, nie mającym dokładnego odpowiednika w typografii polskiej, wywodzącej się z tradycji kontynentalnej, stąd celowość zapożyczenia tego słowa, nie stwarzającego żadnych problemów fleksyjnych. Nawiasem mówiąc, słowo to ma ten sam źródłosłów co „fontanna” — pierwotnie oznaczało ono jeden odlew wykonany przez producenta czcionek drukarskich. Stosowane w spolonizowanych programach edycyjnych tłumaczenie *font* jako *czcionki* może być bardziej intuicyjne dla zwykłego użytkownika, jest natomiast mało wygodne przy bardziej technicznych rozważaniach dotyczących typografii komputerowej.

Nie próbując tutaj formułować precyzyjnej definicji, można powiedzieć, że font to zbiór reprezentacji glifów tak zaprojektowanych, aby dzięki pewnym wspólnym cechom graficznym mogły one być wykorzystywane razem w czytelny i estetyczny sposób do prezentowania (drukowania lub wyświetlania) tekstów. Reprezentacja glifu jest obiektem znacznie bardziej konkretnym niż sam glif, charakteryzuje się bardziej szczegółowym wyglądem, np. przesądza już ona o tym, czy litery mają tzw. szeryfy czy też są **bezszerypowe**; pewne aspekty kształtu, łatwo modyfikowalne algorytmicznie, jak wielkość czy postura (pismo proste lub *pochylone*), mogą być ustalone dopiero w trakcie prezentacji, tj. przy tworzeniu obrazów glifów postrzeganych przez użytkownika.

Ponieważ przy kodowaniu tekstu czystego powinniśmy abstrahować od jego wyglądu, czyli zajmować się znakami, a nie glifami, należy traktować przekształcenie znaków na odpowiednie glify jako osobne zadanie. Trzeba jednak pamiętać o tym, że większość kodów pozwala zapisywać teksty w różnych językach naturalnych, zaś wiele języków wymaga specyficznych glifów — np. akcent nad dużą literą „A” inaczej umieszcza się w języku francuskim, a inaczej w węgierskim. Jeśli zależy nam na wysokiej jakości druku, należy w miarę możliwości stosować font specjalnie zaprojektowany dla danego języka. Oznacza to w szczególności, że fonty obsługujące pewien konkretny kod (np. UNICODE) nie mogą uwzględnić wszystkich, często sprzecznych, wymagań różnych języków, i należy w pełni uświadamiać sobie ich ograniczenia, które zresztą przy pewnych zastosowaniach, np. przy oglądaniu obcojęzycznych stron WWW, są całkowicie nieistotne.

3 Standaryzacja i normalizacja kodowania

3.1 Informacje wstępne

Terminy *standaryzacja* i *normalizacja* są normalnie używane wymiennie, ja proponuję jednak wprowadzić pewne rozróżnienie. Przez standaryzację zamierzam rozumieć wszelkie działania zmierzające do ujednoczenia pewnych produktów lub metod, w szczególności wprowadzanie tzw. standardów przemysłowych przez producentów dominujących na rynku, zaś przez normalizację — analogiczną działalność, ale prowadzoną lub oficjalnie uznawaną przez administrację państwową — w szczególności przez Polski Komitet Normalizacyjny i organizacje międzynarodowe, których jest on członkiem.

Jeśli traktować metrologię jako dział normalizacji, to jej początków można dopatrywać się w zamierzchłej przeszłości, ale początki normalizacji w obecnym znaczeniu to połowa XIX wieku, kiedy w Anglii, w USA i we Francji dokonano niezależnej standaryzacji gwintów (informacje historyczne podaję za Wielką Encyklopedią Powszechną PWN [60]). Systematyczne prace normalizacyjne jako pierwsze rozpoczęło Stowarzyszenie Inżynierów Niemieckich, a w Polsce Stowarzyszenie Elektryków Polskich. Pierwsza krajowa organizacja normalizacyjna została powołana w Anglii w 1901 r., Polski Komitet Normalizacyjny utworzono w 1924 r.

3.2 Normy ISO

Choć pierwszą międzynarodową organizację normalizacyjną powołano już w 1926 r., współpraca międzynarodowa nabrała impetu znacznie później, prawdopodobnie pod wpływem doświadczeń wojennych — skutek różnic w profilu gwintu calowego w latach 1940–45 jako części zamienne do wykorzystywanego przez aliantów sprzętu amerykańskiego sprowadzono konwojami do Europy ponad 2 miliony ton śrub i nakrętek. W r. 1947 powstała Międzynarodowa Organizacja Normalizacyjna ISO (*International Organisation for Standardization* — ISO nie jest skrótem!), będąca federacją krajowych organizacji normalizacyjnych; do ustanowienia normy międzynarodowej potrzebna jest zgoda co najmniej 75% głosujących krajów.

ISO ściśle współpracuje z Międzynarodową Komisją Elektrotechniczną (IEC, *International Electrotechnical Commission*), co przejawia się w istnieniu wspólnego komitetu technicznego JTC1 poświęconego technice informacyjnej (*information technology*). Komitet dzieli się na podkomitety, z których najważniejszy jest dla nas SC2 *Character sets and information coding* (*Zestawy znaków i kodowanie informacji*). Podkomitet ten prowadzi swoją działalność w sposób bardzo otwarty, co niestety jest ciągle jeszcze rzadkością w działalności normalizacyjnej. W szczególności prawie wszystkie robocze dokumenty podkomitetu — łącznie z planami pracy — są dostęp-

ne w Internecie pod adresem <http://osiris.dkuug.dk/jtc1/sc2/open>². W Polsce analogicznymi zagadnieniami zajmuje się Normalizacyjna Komisja Problemowa nr 170 ds. Terminologii Informatycznej i Kodowania Informacji.

W ISO problematyką zestawów znaków zajmuje się również komitet techniczny TC46 *Information and Documentation*, a konkretnie podkomitet SC4 *Computer Applications in Information and Documentation* (*Zastosowania komputerów w informacji i dokumentacji*), o którym bliższe dane można znaleźć w Internecie pod adresem <http://lcweb.loc.gov/loc/standards/isotc46>³. Wybitny specjalista w zakresie normalizacji i autor kilku norm międzynarodowych Johan W. van Wingen w podręczniku normalizacyjnym, rekomendowanym przez holenderskie ministerstwo spraw wewnętrznych, pisał o normach znakowych opracowanych przez TC 46 w następujący sposób ([62], p. 47):

ISO 5426 i inne bibliograficzne zestawy znaków kodowych zostały opracowane przez ISO TC46/SC4. Jeśli popatrzyć łącznie na całą tę serię, to widać wyraźnie, jak niska jest techniczna jakość tych norm w porównaniu z normami opracowanymi przez JTC1. Są one pełne niejasności, wieloznaczności i niezręcznych sformułowań. Od żadnej biblioteki uniwersyteckiej nie uzyskano informacji, że ISO 5426 było kiedykolwiek stosowane w Holandii. Trzeba sobie w związku z tym w pełni zdawać sprawę, że powoływanie się na implementacje ISO 5426 jest prowadzeniem w ślepy zaulek.

Analogiem TC46 jest w Polsce Normalizacyjna Komisja Problemowa nr 242 ds. Informacji i Dokumentacji. Mając bliski kontakt z normami TC46/SC4 jako członek tej komisji w pełni potwierdzam ocenę van Wingena.

3.3 Normalizacja w Polsce

W dniu 1 stycznia 1994 weszła w życie ustawa o normalizacji z dnia 3 kwietnia 1993 r. (opublikowana w Dzienniku Ustaw nr 55 jako pozycja 252; do ustawy wprowadzono później kilka nieistotnych dla niniejszego artykułu zmian), której artykuł 19 ustęp 1 stwierdza, że stosowanie Polskich Norm jest dobrowolne (z pewnymi wyjątkami, których nie warto tutaj wyliczać); sytuacja prawna normalizacji w Polsce jest więc obecnie identyczna jak w większości krajów świata.

Fakt, że normy nie są obowiązkowe z mocy ustawy, nie wyklucza tego, że mogą się powoływać na nie inne akty prawne. Tragikomicznym przykładem jest tutaj ustawa o zamówieniach publicznych (z 10.06.94, Dz.U. z 4.07.94 nr 76 poz. 344 z późniejszymi zmianami), która w artykule 17 pkt. 1 stwierdza

²Aktualnie — 25.11.2005 — pod adresem <http://anubis.dkuug.dk/jtc1/sc2/open/>.

³Adres nieaktualny. Komitet zmienił nazwę i zakres działania — por. <http://www.niso.org/international/SC4/>. Uwaga dodana 25.11.2005 r.

Przedmiot zamówienia określa się za pomocą obiektywnych cech technicznych i jakościowych przy przestrzeganiu Polskich Norm . . . Gdyby w całości potraktować ją serio, to uczelnia państwowa (w przeciwieństwie do prywatnej) nie mogłaby w drodze przetargu zakupić serwera internetowego, ponieważ standardy Internetu nie są polskimi normami.

Pomimo powyższego przykładu niekompetencji i nadgorliwości naszych ustawodawców polski system normalizacyjny w zasadzie nie budzi poważniejszych zastrzeżeń. W teorii proces ustanawiania norm jest w pełni demokratyczny: okresowo przeprowadzane tzw. ankietowanie programu prac normalizacyjnych pozwala każdemu zainteresowanemu zgłaszać swoje uwagi, również konkretne projekty norm podlegają tzw. ankiecie powszechnej ogłaszanej w specjalnym dodatku do miesięcznika *Normalizacja*. Niestety, w praktyce system ten nie jest w pełni sprawny, stąd w ustanawianych normach częste są błędy merytoryczne i językowe.

Dla przykładu, w normie PN-ISO/IEC 2382-23 z grudnia 1996 pt. *Technika informatyczna — Terminologia — Przetwarzanie tekstu* mamy na s. 10 ewidentny błąd polegający na pomyleniu przypisu (ang. *footnote*) ze stopką (ang. *footer*). Na tej samej stronie zamiast linii pisma (ang. *baseline*) mamy dziwaczne neologizmy *linia odniesienia* i *linia bazowa wiersza*. Zdziwienie budzi również sformułowanie na s. 17 *Stopki i nagłówki stron mogą być drukowane na łamaniami stron*. Pomimo tych błędów tekst normy został zaaprobowany m.in. przez kilka instytucji (nie było wśród nich ani Instytutu Informatyki, ani Zakładu Zastosowań Informatycznych IO UW) poproszonych o opinię w ramach tzw. ankiety adresowanej przez Zespół Informatyki i Telekomunikacji, a także zatwierdzony przez Wydział Kontroli Norm Biura Polskiego Komitetu Normalizacyjnego.

3.4 Standardy przemysłowe

Stosunek przemysłu komputerowego do norm znakowych jest bardzo zróżnicowany. Dominujące na rynku firmy, takie jak IBM i Microsoft, w praktyce przez dłuższy czas je ignorowały, czyniąc nieliczne wyjątki pod wpływem nacisku użytkowników. Niektóre duże firmy komputerowe, jak Digital Equipment Corporation, starały się zastosować w praktyce wybrane normy międzynarodowe, ale nie mogły sobie pozwolić na opóźnianie wprowadzenia swoich produktów na rynek do czasu zakończenia procesu normalizacyjnego, stąd *DEC Multinational Coded Character Set* stanowi w istocie wstępną wersję ISO/IEC 8859-1 ([62], s. 43).

Nie jest rzeczą zaskakującą, że właśnie w Europie najbardziej odczuwano potrzebę stworzenia międzynarodowych norm znaków odpowiadających różnorodnym potrzebom użytkowników. Tutaj właśnie powstało w 1961 Europejskie Stowarzyszenie Producentów Komputerów (ECMA, *European Computer Manufacturers Association*), które istnieje do dzisiaj, choć z czasem zmieniło swój charakter i określa się teraz — zachowując nadal symbol

ECMA, ale już nie traktując go jak skrót — jako międzynarodowe stowarzyszenie normalizacji systemów informacyjnych i komunikacyjnych. Podstawową formą działalności stowarzyszenia jest wydawanie własnych standardów, w szczególności dotyczących kodowania znaków — wszystkie standardy tej grupy stały się potem standardami ISO, niekiedy w nieco innej formie redakcyjnej, a niekiedy bez zmian. Ma to poważne znaczenie praktyczne, ponieważ zarówno normy ISO jak i ich polskie tłumaczenia (często wątpliwej jakości) są kosztowne, tymczasem ECMA wszystkie swoje standardy udostępnia bezpłatnie, w szczególności przez Internet — patrz www.ecma.ch⁴.

Największe firmy komputerowe zaczęły poważnie interesować się normalizacją kodowania znaków dopiero wtedy, kiedy — jak się wydaje — zaczęły się obawiać nasycenia się rynku amerykańskiego i zachodnioeuropejskiego. Aby zmniejszyć koszty dostosowywania swoich produktów do lokalnych, w tym również dalekowschodnich, rynków (czyli kosztów tzw. lokalizacji), zawiązały one w 1991 konsorcjum UNICODE, lansujące tzw. uniwersalny kod znaków o tej samej nazwie ([61]). Jest bardzo możliwe, że to właśnie konsorcjum UNICODE przesądziło o ostatecznym kształcie normy ISO/IEC 10646 ([36]) i ma istotny wpływ na dalszy jej rozwój.

Do problematyki uniwersalnych kodów znaków powrócimy w dalszej części niniejszego artykułu.

3.5 Inne organizacje normalizacyjne

Ze względu na brak miejsca nie zajmiemy się tutaj bliżej Europejskim Komitetem Normalizacyjnym (CEN, *Comité Européen de Normalisation* — por. <http://www.cenorm.be>), który w 1995 roku zdefiniował Minimalny Podzbiór Europejski (MES, *Minimal European Subset*) zestawu ISO-10646, liczący tylko 926 znaków, oraz nieco większy Rozszerzony Podzbiór Europejski (EES, *Extended European Subset*) zestawu ISO 10646. Aktualnie⁵ podzbiory te mają status tzw. prestandardu (ENV 1973:1995, por. <http://www.indigo.ie/egt/standards/mes.html>⁶ i [ees.html](http://www.indigo.ie/egt/standards/ees.html)⁷), ale zgodnie z zasadami działania Komitetu uzyskają one status normy krajowej we wszystkich krajach Wspólnoty Europejskiej. Jednym z warunków wstąpienia Polski do Wspólnoty Europejskiej jest uzyskanie członkostwa w CEN, co z kolei wymaga ustanowienia jako polskich norm 80% norm europejskich; można zatem sądzić, że wspomniane wyżej normy (oprócz takich norm jak EN 29983:1991 o oznaczaniu niezapisanych dyskieciek i EN 28630:1992 o pa-

⁴Aktualny adres <http://www.ecma-international.org/> — uwaga dodana 25.11.2005 r.

⁵Obecnie — 25.11.2005 — są to już normy europejskie i krajowe, por. np. PN-EN 1923:2000.

⁶Aktualnie — 25.11.2005 — pod adresem <http://www.evertype.com/standards/mes.html>.

⁷Aktualnie — 25.11.2005 — pod adresem <http://www.evertype.com/standards/ees.html>.

rametrach technicznych dyskietek o średnicy 5,25 cala, już zaplanowanych do ustanowienia odpowiednio w 2000 i 2001 roku) prędzej czy później staną się również polskimi normami.

Warto tutaj podkreślić, że normalizacja nie powinna być nigdy celem samym w sobie. W szybko rozwijających się dziedzinach przedczesne przyjęcie za standard niedojrzałych rozwiązań może w praktyce hamować postęp zamiast go stymulować. Również sama biurokratyczna procedura normalizacyjna może się w takich wypadkach okazać zbyt powolna i mało wygodna — nie jest chyba przypadkiem, że prawie wszystkie liczące się normy znakowe powstały poza strukturą ISO i krajowych organizacji normalizacyjnych, a nadanie im statusu norm międzynarodowych lub krajowych było tylko formalnością. Nie jest również chyba przypadkiem, że najważniejsza w tej chwili — również dla zwykłego człowieka — dziedzina informatyki jaką jest Internet, stworzyła swoje własne, oryginalne i demokratyczne procedury normalizacyjne, opierające się przede wszystkim na pracy wolontariuszy, tylko pośrednio wspieranej przez zainteresowane instytucje.

Wszystkie oficjalne dokumenty Internetu mają postać elektroniczną i są dostępne bezpłatnie za pomocą sieci; zgodnie z tradycją sięgającą 1969 roku noszą one nazwę *Prośby o uwagi* (RFC, *Request for Comments*); ukażało się ich dotąd około dwóch i pół tysiąca. Są one umieszczane w Internecie pod adresem `ftp://ds.internic.net/rfc/`⁸ i replikowane na wielu komputerach — w Polsce najłatwiej je znaleźć w Interdyscyplinarnym Centrum Modelowania UW pod adresem `ftp://ftp.icm.edu.pl` lub `http://www.icm.edu.pl`⁹. Procedurę normalizacyjną opisują przede wszystkim RFC 2026 i RFC 2282. Informacje na ten temat można znaleźć także w moim opracowaniu [8], do którego odsyłam w szczególności Czytelników zainteresowanych problematyką kodowania tekstów w sieciach komputerowych.

4 Kod ASCII i pochodne

ASCII to *American Standard Code for Information Interchange* czyli *Amerykański Standardowy Kod do Wymiany Informacji*. Jest to 7-bitowy kod, wprowadzony normą USAS X3.4 z 1967 roku, a w swojej aktualnej postaci określony normą ANSI X3.4 z 1986 roku. Znaczenie jego polega przede wszystkim na tym, że stał się on podstawą wielu innych kodów 7- i 8-bitowych. Jego bezpośrednie wykorzystanie jest obecnie w zaniku, ma ono miejsce niemal wyłącznie w Internetowej poczcie elektronicznej, przy czym dzięki różnym technikom konwersji dowolnego kodu na kod ASCII nie ogranicza to stosowania języków narodowych w korespondencji elektronicznej (patrz np. [8]).

⁸Aktualnie — 25.11.2005 — pod adresem `http://www.rfc-editor.org/rfc.html`.

⁹Aktualnie — 25.11.2005 — najwygodniejszą formą dostępu do RFC wydaje się `http://zvon.org/tmRFC/RFC_share/Output/front.html`; ICM UW chyba już ich nie replikuje.

Należy sobie wyraźnie uświadamiać, że termin ASCII jest potocznie używany także w innych znaczeniach, w szczególności w znaczeniu tekstu czytego zapisanego w jednym z kodów 8-bitowych. Do niedawna takie właśnie znaczenie ASCII stosowała firma Microsoft: jeśli np. w polskiej wersji edytora Word 6.0 wykonamy komendę *Zachowaj plik w formacie: Plik tekstowy ASCII*, to plik zostanie zapisany w jednej z tzw. stron kodowych omawianych w punkcie następnym (prawdopodobnie w stronie kodowej 852); nowsze wersje tego edytora stosują poprawne określenie *Plik tekstowy DOS*. Aby uniknąć tego typu nieporozumień, standardy Internetowe zalecają powoływać się na wspomnianą normę za pomocą skrótu USASCII.

Standard ASCII został przystosowany do wymagań europejskich przez ECMA. Choć oryginalny standard dopuszczał pewne drobne modyfikacje kodu w celu uwzględnienia lokalnych potrzeb, standard ECMA-6 w swoim trzecim wydaniu wprowadził formalnie pojęcie *wersji* kodu oraz specjalnej, domyślnej wersji nazywanej *międzynarodową wersją wzorcową* (IRV, *International Reference Version*). Standard ECMA-6 został następnie przyjęty przez ISO jako norma ISO 646:1972.

Jednym z najbardziej kontrowersyjnych aspektów normy ISO 646 był zestaw znaków międzynarodowej wersji wzorcowej; wiele krajów (w tym również zachodnich) nie chciało ze względów politycznych pełnej zgodności tej wersji z kodem ASCII, dlatego znak dolara zastąpiono w niej tzw. międzynarodowym symbolem waluty. Decyzja ta okazała się mało praktyczna i w aktualnie obowiązującej wersji ISO 646:1991 międzynarodowa wersja wzorcowa jest całkowicie zgodna z ASCII.

Pierwsze polskie normy znakowe ze względów politycznych opisywały jednocześnie kodowanie alfabetu łacińskiego i cyrylicznego. Kod ISO 646:1983 został wprowadzony jako kod N_0 (łącznie z kodem N_1 dla cyrylicy) w normie PN-79/T-42109/01, później zastąpionej przez PN-88/T-42109/01 ([44]), przy czym niektóre ustalenia ISO 646 zostały zawarte w odrębnej normie PN-78/T-42108, zastąpionej przez PN-89/T-42108 ([45]).

Nieco później w normie PN-84/T-42109/02 ([42]) wprowadzono krajowy zestaw znaków ZU0 (ZU jest prawdopodobnie skrótem od *zestaw uniwersalny*) będący wariantem ISO 646:1983; ponieważ jednak ISO 646 na znaki narodowe rezerwowało tylko 10 pozycji, umożliwiło to stosowanie oprócz wszystkich małych liter narodowych tylko dużej litery „Ł”; funkcje brakujących dużych liter miały pełnić odpowiednie małe litery, np. PAMIĘĆ. Oprócz kodu ZU0 norma ta wprowadzała pozbawiony w ogóle polskich liter kod ZU1 przeznaczony dla komputerów ODRA (zgodnych z angielskimi komputerami ICL 1900), dopuszczała jednocześnie stosowanie w ZU1 polskich znaków na dowolnych pozycjach na podstawie prozumuienia zainteresowanych stron.

Po ukazaniu się trzeciego wydania ISO 646 z 1991 r. zostało opracowane jego polskie tłumaczenie, które otrzymało symbol PN-93/T-42109/01. Jakość tego tłumaczenia jest daleka od ideału, w szczególności słowo *reference* jest bez uwzględniania kontekstu tłumaczone jako *odniesienie*, stąd wersja

wzorcową jest tam nazywana „wersją odniesienia”.

Kod ASCII i pochodne w sposób jawny dopuszczają reprezentowanie znaków spoza ich zakresu przez sekwencje powodujące nadrukowanie kształtu jednego znaku na drugim, a także przez sekwencje wykorzystujące specjalne znaki sterujące takie jak ESCAPE (*ucieczka*).

W momencie pisania tych słów (styczeń 1999) w Polsce obowiązują wszystkie wymienione normy: PN-84/T-42109/02, PN-89/T-42108 i PN-93/T-42109/01; dopiero niedawno rozpoczęto procedurę wycofywania dwóch najstarszych norm.

5 Kody ośmiobitowe

5.1 EBCDIC

Zanim pojawiły się komputery, w Stanach Zjednoczonych i innych krajach do przetwarzania danych wykorzystywano mechaniczne urządzenia operujące na kartach perforowanych, na których informacje zapisywano stosując tzw. kod Holleritha; produkcją takich urządzeń zajmowała się firma International Business Machines czyli IBM. Kiedy mniej więcej w 1965 r. firma ta zaanonsowała rozpoczęcie produkcji komputerów serii 360, dla komputerów tych zaprojektowano 8-bitowy kod silnie nawiązujący do kodu Holleritha. Kod ten otrzymał nazwę EBCDIC (*Extended Binary Coded Decimal Interchange Code* — objaśnienie znaczenia tej nazwy wymagałoby zbyt obszernego omówienia jego historii) i pierwotnie wykorzystywał tylko niewielką część 256 pozycji kodowych. Z czasem IBM zaczął tworzyć różne narodowe odmiany EBCDIC, nazywając je *stronami kodowymi* (CP, *code page*), a także *rozszerzonymi krajowymi stronami kodowymi* (CECP, *Country Extended Code Page*). Do końca lat siedemdziesiątych był to kod najczęściej stosowany w praktyce (ASCII zaczęło się upowszechniać dopiero z systemem operacyjnym UNIX i komputerami osobistymi).

W krajach RWPG produkowano z gorszym lub lepszym skutkiem kopie komputerów IBM 360 pod nazwą komputerów Jednolitego Systemu, potocznie nazywanego RIAD, które oczywiście stosowały kod EBCDIC. Pomimo tego kod ten nigdy nie stał się normą — ani ogólnokrajową ani branżową; jest to bardzo ciekawe dlatego, że zgodnie z obowiązującym wówczas prawem stosowanie polskich norm było obowiązkowe, a ich nieprzestrzeganie narażało na karę aresztu do lat dwóch.

5.2 ISO 6937

W 1983 r. ukazało się pierwsze wydanie (w dwóch częściach) normy ISO 6937, uaktualnionej w 1994 r. ([31]). Jej podstawowe znaczenie polega na tym, że określiła ona jawnie zestaw liter i znaków stosowanych we wszystkich żywych językach europejskich pisanych alfabetem łacińskim. Zestaw

ten, nazywany repertuarem ISO 6937, jest przywoływany w wielu innych normach międzynarodowych, np. dotyczących układu klawiatury.

Od strony technicznej ISO 6937 opisywała kod 8-bitowy, który przewidywał traktowanie niektórych znaków jako złożonych z litery podstawowej i znaku diakrytycznego, przy czym znak diakrytyczny odpowiadał tzw. martwym klawiszom stosowanym w maszynach do pisania, tj. nie powodował przesuwu karetki. Stosując terminologię ISO, należy powiedzieć, że znak diakrytyczny jest *non-spacing*; w niektórych polskich normach termin ten tłumaczono jako „niespacjujący”, co jest mylące, ponieważ sugeruje jakiś związek z typograficznym terminem *spacjowanie* (oznaczaającym wyróżnienie tekstu przez wprowadzenie specjalnych odstępów między literami). W nowszych normach ISO termin *non-spacing* zastąpiono przez *combining*, co w normie PN-ISO/IEC 2022 przetłumaczono jako *składalny*; moim zdaniem jest to również mylące, ponieważ sugeruje, że znak „składalny” może, ale nie musi być złożony z innym. W istocie *combining character* nie może być wykorzystywany samodzielnie i dlatego proponuję nazywać go *znakiem składowym*.

Tak więc ISO 6937 niektóre znaki kodowała za pomocą jednego 8-bitowego bajtu, pozostałe zaś za pomocą dwóch kolejnych 8-bitowych bajtów. Powodowało to różne komplikacje przy przetwarzaniu tak zapisanych tekstów, a także przy ich wyświetlaniu na monitorach komputerowych. Uznano wówczas, że komplikacje te nie są warte korzyści płynących z użycia ISO 6937, stąd wykorzystanie tej normy ograniczyło się do teletekstu (wyświetlania na ekranie telewizora informacji tekstowej wysyłanej razem z sygnałem telewizyjnym) i wideotekstu (teletekst z możliwością przekazywania informacji przez użytkownika za pomocą telefonu lub specjalnego terminala). Jedyne znany mi wyjątek od tej reguły, to stosowany m.in. przez Uniwersytet Warszawski i kilka innych bibliotek w Polsce amerykański system biblioteczny VTLS, który stosuje nieco rozszerzony kod ISO 6937 do przechowywania danych bibliograficznych.

W miarę wzrostu ogólnej złożoności systemów komputerowych komplikacje powodowane przez znaki składowe zaczęły stosunkowo tracić na znaczeniu, stąd idea ta pojawiła się na nowo, choć w nieco innej formie, w uniwersalnych kodach znaków omawianych dalej.

5.3 ISO 8859

Ze względu na wymienione wyżej słabe strony ISO 6937 i ograniczenie kodu ASCII do 128 znaków, w 1982 r. zarówno ECMA jak i ANSI (*American National Standard Institute*) rozpoczęły prace nad serią kodów 8-bitowych, w których każdy znak byłby zawsze reprezentowany przez dokładnie jeden bajt, ułatwiając w ten sposób przetwarzanie tekstów i ich wyświetlanie na monitorach ekranowych (które, w przeciwieństwie do stosowanych przed nimi dalekopisów, nie pozwalały na nadrukowywanie jednego znaku na dru-

gim). Oczywiście, kod 8-bitowy o 256 znakach nie był w stanie uwzględnić wszystkich potrzebnych znaków, stąd konieczność stworzenia standardu wieloczęściowego, w którym każda część była przeznaczona dla pewnego rejonu geograficznego.

W rezultacie w latach 1985–1992 powstała seria standardów ECMA, później zatwierdzonych jako odpowiednie części standardu ISO 8859: ECMA-94 (ISO 8859-1, -2, -3 i -4), ECMA-113 (ISO 8859-5), ECMA-114 (ISO 8859-6), ECMA-118 (ISO 8859-7), ECMA-121 (ISO 8859-8), ECMA-128 (ISO/IEC 8859-9), ECMA-144 (ISO/IEC 8859-10).

Ogólna struktura tych kodów jest określona przez osobne normy ECMA-35 (ISO/IEC 2022) i ECMA-43 (ISO/IEC 4873), o których jeszcze będziemy mówić w punkcie następnym. Ich uzupełnieniem jest norma ECMA-48 (ISO/IEC 6429, PN-ISO/IEC 6429:1996 — patrz [53]) wprowadzająca dodatkowe znaki sterujące.

Największym problemem rodziny kodów ISO 8859 jest to, że repertuar znaków był ustalany na podstawie kryteriów politycznych, co w przypadku norm ISO 8859-1 i 8859-2 można nazwać syndromem żelaznej kurtyny: w kodzie 8859-2 można napisać tekst po polsku, czesku, słowacku, chorwacku, słoweńsku, serbsku (oczywiście w zapisie łacińskim), albańsku, węgiersku, rumuńsku i niemiecku (a przy okazji również po angielsku, fińsku i irlandzku), ale nie można np. zacytować tekstu po francusku. Problemy takie rozwiązywano tworząc coraz to nowe kody tej rodziny, np. 8859-9 (tzw. Latin-5; numer alfabetu łacińskiego nie zawsze pokrywa się z numerem opisującej go części normy) powstał z 8859-1 przez dołączenie liter języka tureckiego kosztem liter islandzkich. Z obecnej perspektywy wspomniane wyżej metody usuwania wad kodów rodziny 8859 wydają się jednak tylko półśrodkami.

Zarezerwowanie zgodnie z ISO/IEC 4873 aż 32 pozycji na dodatkowe kody sterujące wydaje się obecnie marnotrawstwem, ponieważ w praktyce nie są one używane; równie mało przydatne są także samodzielne znaki diakrytyczne (np. ogonek). W rezultacie repertuar znaków jest stosunkowo ubogi — np. ISO 8859-2 zawiera 26 znaków mniej niż repertuar strony kodowej 1250 (np. poprawne polskie cudzysłowy są dostępne w kodzie 1250, ale nie w kodzie ISO 8859-2).

ISO 8859-2 ma swój odpowiednik polski w postaci normy PN-91/T-42115 ([46]). Dziedziczy ona oczywiście wszystkie kontrowersyjne własności oryginału, dodając do tego nowe problemy związane z nienajlepszym tłumaczeniem tekstu normy ISO na język polski.

6 Kody rozszerzalne

Jak wspominaliśmy wcześniej, kod ASCII przewidywał możliwość rozszerzania zestawu znaków za pomocą specjalnych kodów sterujących. W 1971 r. zostało opublikowane pierwsze wydanie standardu ECMA-35, opisują-

ce szczegółowe reguły rozszerzania kodów 7- i 8-bitowych; szóste wydanie ECMA-35 z 1994 r. jest identyczne z normą ISO/IEC 2022:1994, posiadającą polski odpowiednik PN-ISO/IEC 2022 ([51]) z 1996 roku.

Jedna z technik opisanych w normie ISO/IEC 2022 została wykorzystana w 1986 roku do zdefiniowania 7-bitowego zestawu znaków ZU2 w normie PN-86/T-42109/03 ([43]).

Przewidując, że kody 7-bitowe będą wychodzić z użycia, w 1974 r. wydano standard ECMA-43, którego trzecie wydanie z 1991 roku jest równoważne normie ISO/IEC 4873:1991, posiadającej polski odpowiednik PN-93/T-42112 ([49]). Normy te można traktować jako uproszczone wersje ISO/IEC 2022, ograniczone do problematyki kodów 8-bitowych, a jednocześnie rozszerzone o zagadnienia ogólne, które w przypadku kodów 7-bitowych były omówione w ISO/IEC 646.

ISO/IEC 2022 i ISO/IEC 4873 przewidywały prawie nieograniczoną możliwość rozszerzania zestawu znaków za pomocą sekwencji znaków zaczynających się od ESCAPE. Sekwencjom tym można było przypisywać znaczenie lokalnie, ale można też było podawać je do publicznej wiadomości rejestrując je w ISO zgodnie z procedurą określoną w normie ISO 2375; wyznaczona przez ISO instytucja — początkowo ECMA, obecnie ITSCJ (*Information Technology Standards Commission of Japan*) — utrzymuje odpowiedni rejestr i wysyła zainteresowanym tablice zarejestrowanych w nim kodów. Od niedawna rejestr ten jest dostępny w Internecie pod adresem www.itscj.ipsj.or.jp/ISO-IR/.

Reguły rozszerzania kodów zdefiniowane w tych normach były tak skomplikowane, a teksty zapisane za ich pomocą tak nieporęczne do przetwarzania, że techniki te — z bardzo nielicznymi wyjątkami — nie są w ogóle stosowane w praktyce. Natomiast zarejestrowanie sekwencji rozszerzającej dla danego kodu jest najwygodniejszą metodą podania tablicy tego kodu do publicznej wiadomości, i w tej swojej wtórnej funkcji Międzynarodowy Rejestr (IR, *International Register*) nadal skutecznie funkcjonuje. Z polskiego punktu widzenia interesująca jest rejestracja IR 179 *Baltic Rim* (kod pobraża Bałtyku) zawierający m.in. litery polskie, litewskie, łotewskie i skandynawskie.

Jak się wydaje, ISO JTC1/SC2 dość szybko zdał sobie sprawę ze znikomej praktycznej przydatności tych technik rozszerzania kodu, dlatego ustanowił tylko jedną normą korzystającą z nich w sposób istotny, mianowicie ISO/IEC 10367:1991; norma ta, dotycząca również języka polskiego, ma swój polski odpowiednik w postaci normy PN-93/T-42118 ([50]).

Całkowicie odmiennie potraktował normy ISO/IEC 2022 i ISO/IEC 4873 komitet techniczny ISO TC46 *Information and Documentation*, mianowicie odniósł się do nich z niezrozumiałym entuzjazmem. Entuzjazm ten w dużym stopniu podzieliła m.in. polska Normalizacyjna Komisja Problemowa nr 242 ds. Informacji i Dokumentacji oraz Ośrodek Informacji i Dokumentacji Biura Polskiego Komitetu Normalizacyjnego.

Oprócz normy ISO 6630 (polski odpowiednik PN-93/N-09128 — patrz [47]), komitet TC 46 w latach 1983–1996 ustanowił 12 norm zakładających (jak się wydaje, całkowicie niesłusznie) praktyczną dostępność wyrafinowanych technik rozszerzania kodu. Następujące z tych norm mają polskie odpowiedniki:

ISO 5426:1983 — PN-84/N-09120 ([39]); norma ta określa jeszcze jeden sposób kodowania polskich liter.

ISO 5427:1984 — PN-85/N-09122 ([41]).

ISO 5428:1984 — PN-85/N-09121 ([40]).

ISO 6862:1996 — PN-ISO 6862; norma skierowana do ustanowienia w grudniu 1998 r.

W roku 1998 doszło do porozumienia między JTC1 i TC 46, w wyniku którego martwe normy kodowe ustanowione przez TC 46 zostaną wchłonięte przez uniwersalny kod znaków zdefiniowany przez ISO/IEC 10646 bez formalnego ich wycofywania (ponieważ niektóre z nich były ustanowione zaledwie 2 lata temu, byłoby to jawne stwierdzenie niekompetencji TC 46). Odbędzie się to w ten sposób, że w Międzynarodowym Rejestrze zostaną zarejestrowane wszystkie kody wprowadzone we wspomnianych normach, ale razem z ich odpowiednikami w ISO/IEC 10646; jednocześnie wystąpiono z wnioskiem o poszerzenie ISO/IEC 10646 o pewne znaki, występujące w normach TC 46, a aktualnie niedostępne w ISO/IEC 10646.

W momencie pisania tych słów wszystkie wymienione polskie normy są obowiązujące, dopiero niedawno rozpoczęto procedurę wycofywania PN-86/T-42109/03.

6.1 Strony kodowe IBM i Microsoft

Kiedy firma IBM wprowadziła na rynek komputer osobisty PC (współpracował on z magnetofonem i telewizorem i był pozbawiony twardego dysku, który pojawił się dopiero później w modelu XT — *eXtended Technology*), był on produkowany w dwóch odmianach. Jedna z nich, przeznaczona do przetwarzania tekstów, była całkowicie pozbawiona możliwości graficznych, dlatego zaprojektowany dla niej 8-bitowy kod (później nazwany stroną kodową 437) zawierał w sobie oprócz pełnego zestawu ASCII dużo znaków nietypowych oraz tzw. znaki semigraficzne służące do wyświetlania pojedynczych i podwójnych ramek oraz tzw. grafiki blokowej. Pod wpływem potrzeb użytkowników europejskich IBM wprowadziło dodatkowe kody, mianowicie stronę kodową 850 i 852. Ich repertuary zawierały w sobie odpowiednio repertuar ISO 8859-1 i 8859-2, ale dla zachowania maksymalnej zgodności z kodem 437 znaki narodowe nie mogły się znaleźć na tych samych pozycjach, co w normach ISO. Co więcej, choć repertuar znaków ISO jest stosunkowo

ubogi ze względu na zarezerwowanie pewnych pozycji na znaki sterujące, to jednak jego liczebność nie pozwalała zachować wszystkich znaków semigraficznych kodu 437. W tej sytuacji IBM zrezygnowało z znaków służących do łączenia ramek pojedynczych z podwójnymi, co wydaje się decyzją słuszną i pozbawioną istotnych konsekwencji praktycznych; pomimo tego — jak pokazały wypowiedzi prasowe — wielu polskich użytkowników odebrało to jako szykanę i nie pocieszał ich fakt, że w identycznej sytuacji znaleźli się użytkownicy zachodnioeuropejscy. Jednym z efektów takiej postawy była spora popularność lokalnego kodu MAZOVIA. IBM wprowadził później jeszcze wiele innych stron kodowych dla różnych języków.

Kompletnie nowa seria stron kodowych pojawiła się z inicjatywy Microsoftu, gdy zaczął on oferować graficzny interfejs użytkownika MS Windows. Do tego celu znaki semigraficzne były całkowicie nieprzydatne i zajęte przez nie pozycje można było wykorzystywać do innych celów. Microsoft skorzystał jednak z okazji, aby dokonać kompleksowej reorganizacji stron kodowych i przybliżyć je do znajdujących się w opracowywaniu standardów. W rezultacie strona kodowa 1252 w dużym stopniu pokrywa się z ISO 8859-1, a strona 1250 z ISO 8859-2, są one jednak od nich bogatsze dzięki wykorzystaniu większości pozycji zarezerwowanych w normach ISO dla znaków sterujących. Z czasem pojawiły się również inne strony kodowe, także dla języków orientalnych.

7 Uniwersalne kody znaków i ich zastosowania

Idea Uniwersalnego Systemu Znaków (UCS, *Universal Character Set*) powstała niezależnie w dwóch środowiskach. Jednym z nich było środowisko producentów komputerów i oprogramowania, które stworzyło wspomniane wcześniej konsorcjum UNICODE. Drugim środowiskiem były osoby i instytucje związane z ISO, które zaproponowały nową normę międzynarodową o symbolu ISO/IEC 10646. Początkowo te środowiska były ze sobą w konflikcie, ale z czasem doszło do uzgodnienia poglądów i połączenia wysiłków, w związku z czym kod UNICODE może być traktowany jako szczególne zastosowanie ISO/IEC 10646-1:1993.

Norma ISO 10646 przewiduje stosowanie w ogólnym wypadku aż 4 bajtów na reprezentowanie jednego znaku — liczba różnych znaków, które mogą być w ten sposób reprezentowane jest tak duża, że aż trudna do wyobrażenia; w konsekwencji przewiduje się stosowanie różnych podzbiorów. Największy z nich, noszący nazwę Podstawowej Płaszczyzny Wielojęzycznej (BMP, *Basic Multilingual Plane*) i oznaczany symbolem UCS-2, pozwala jeden znak reprezentować jako 2 bajty; repertuar UCS-2 w praktyce pokrywa się z UNICODE, który liczy obecnie prawie 40 000 pozycji; składają się na to litery i inne znaki praktycznie wszystkich języków żywych (w tym języków stosujących ideograficzne lub inne nielacińskie systemy pisma) i ważniejszych

języków martwych; pewna liczba pozycji jest nadal niewykorzystana i może być użyta w przeszłości do rozszerzenia UNICODE o nowe znaki.

Warto podkreślić, że twórcy UNICODE i ISO-10646 starali się jak najlepiej uwzględnić specyfikę poszczególnych języków, dlatego np. skomplikowanym zasadom pisma koreańskiego poświęcono znaczącą część tych standardów pomimo niewielkiego znaczenia tego języka z ogólnościowego punktu widzenia. Wprowadzanie nowoczesnej technologii wcale nie musi oznaczać rezygnacji z tradycji czy jej zubożenia, co zresztą wiadomo co najmniej od czasu tzw. apelu z Tours o zachowanie europejskiego dziedzictwa językowego (*Tours Manifesto for the Safeguarding of Europe's Heritage of Language*, przedstawionego przez sekretarza generalnego Rady Europy i uchwalonego przez uczestników kolokwium *The language products industry — the stakes for Europe*, które odbyło się w Tours w 1986 r.).

W celu ułatwienia stosowania Uniwersalnego Zestawu Znaków w już istniejących systemach komputerowych zdefiniowano tzw. formaty transformacyjne (UTF, *UCS/UNICODE Transformation Format*), z których najważniejszym jest UTF-8 (por. np. [57]). W formacie tym każdy znak jest reprezentowany przez 1 lub więcej (maksymalnie 6) bajtów w taki sposób, że tak zapisany tekst może być przetwarzany przez starsze oprogramowanie, traktujące UTF-8 jako jeszcze jeden z 8-bitowych kodów. Z tego powodu UTF-8 jest coraz częściej stosowany w stronach WWW i obsługiwany przez nowoczesne przeglądarki.

Pewne aspekty UNICODE są już uwzględniane w takich systemach i środowiskach operacyjnych jak Linux, MS Windows 3.x z rozszerzeniem Win32s oraz MS Windows 95 i 98, natomiast system operacyjny MS Windows NT od początku został zaprojektowany w taki sposób, aby w pełni wykorzystać możliwości UNICODE. Tekst w UNICODE można w Windows NT zapisać m.in. za pomocą notatnika, a w repertuarze UNICODE można się zorientować przeglądając za pomocą tabeli znaków dostępne fonty, np. Lucida Unicode ([10]). Niestety, polonizacja tablicy znaków została dokonana w sposób skrajnie niekompetentny, przejawiający się m.in. w mechanicznym tłumaczeniu słowa *form* (tutaj *forma*, *postać*) jako „formularz”, co prowadzi do absurdów. Dla przykładu, występujące w kodach pism ideograficznych *formy pełno- i niepełnowymiarowe* (*halfwidth and fullwidth forms*) przetłumaczono jako „Formularze o połowie i całej szerokości”, *number forms* jako „Formularze liczbowe” zamiast *formy liczbowe*, *arabskie formy prezentacyjne* (*arabic presentation forms*) jako „Arabskie formularze prezentacyjne” itd. O kompletnym niezrozumieniu istoty sprawy świadczy przetłumaczenie *combining diacritical marks* jako „połączenia znaków diakrytycznych”, chodzi tu bowiem o *składowe znaki diakrytyczne*; jeszcze gorzej zostało przetłumaczone *combining diacritical marks for symbols* — „Połączenia znaków diakrytycznych (symbole)”, w rzeczywistości chodzi tutaj o *składowe znaki diakrytyczne dla symboli*. Określenie *control pictures* oznaczające *graficzne symbole znaków sterujących* (choć w angielskim można na *control characters*

powiedzieć *controls*, w polskim analogiczny skrót nie wydaje się możliwy) przetłumaczono na nic nie mówiące „Rysunki formantów” itd.

UNICODE jest obsługiwany także przez wszystkie programy MS Office 97; warto przy tym wspomnieć, że polonizacja terminologii Unicodowej w MS Word 97 jest w zasadzie poprawna, tłumaczowi należy się nawet uznanie za bardzo zręczne przetłumaczenie określeń typu *enclosed alphanumeric* jako *otoczone znaki alfanumeryczne* itd. (są to np. liczby i litery ujęte w kółka lub nawiasy albo uzupełnione kropką).

Na zakończenie warto powiedzieć, że zgodność oprogramowania z ISO/IEC 10646 nie oznacza tego samego, co zgodność z UNICODE. Skrajnie upraszczając, ISO/IEC 10646 jest standardem kodowania tekstów, a UNICODE — standardem ich przetwarzania.

8 Wnioski

Użytkownik preferujący kodowanie tekstów polskich zgodnie z obowiązującymi polskimi normami może zrobić to na jeden z 5 sposobów: zgodnie z PN-84/T-42109/02, PN-91/T-42115, PN-86/T-42109/03, PN-93/T-42118 lub PN-84/N-09120. Użytkownik preferujący stosowanie aktualnych norm międzynarodowych może kodować teksty polskie na 6 sposobów: ISO 6937, ISO 8859-2, ISO IR 179, ISO/IEC 10367, ISO 5426, ISO/IEC 10646. Fakt, że niektóre z tych norm są przestarzałe, a inne ze względu na błędne założenia nigdy nie weszły do użytku, świadczy dobitnie, że oficjalne instytucje normalizacyjne mają tendencje do życia własnym życiem, a pozbawione stałego nacisku i kontroli użytkowników zaczynają działać zgodnie z prawem Parkinsona (raz powołana komisja nie rozwiązuje się po wykonaniu zadania, lecz wymyśla sobie nowe zajęcia).

Znacznie bliżej faktycznych potrzeb użytkowników jest mało sformalizowana struktura standaryzacyjna Internetu. Jego standardy nie ograniczają w sposób sztywny stosowanych kodów znaków, ale pewne z nich preferują przez umieszczenie ich w specjalnym rejestrze ([56], por. [8]). Dla języka polskiego aktualnie znajdują się tam strony kodowe 852, 1250, kod ISO 8859-2 i ISO/IEC 10646. Jako rozwiązanie przyszłościowe zdecydowanie preferowany jest kod ISO/IEC 10646 ([55]).

Jak widać, wbrew często głoszonym poglądom (m.in. w grupie wiadomości sieciowych `pl.comp.ogonki` i na tzw. Polskiej Stronie Ogonkowej `http://www.cyf.gov.pl/ogonki/pl.html`¹⁰), względy formalne nie pozwalają wskazać jednego powszechnie obowiązującego sposobu kodowania polskich liter. Co więcej, moim zdaniem wszelkie kategoryczne twierdzenia dotyczące wyższości jednego kodu nad drugim (przypominające często dyskusje nad wyższością świąt Bożego Narodzenia nad świętami Wielkiejnocy) stanowią w istocie zamach na prawo autora i wydawcy samodzielnego decydowania o

¹⁰Aktualnie — 25.11.2005 — pod adresem `http://www.ogonki.agh.edu.pl/`.

formie swojej wypowiedzi. Podobnie jak wydanie książki w formacie kieszonkowym adresuje ją do innego czytelnika niż wydanie jej w twardej oprawie w formacie A3, tak np. zapisanie tekstu w ISO 10646 adresuje go do czytelników na całym świecie dysponujących dostatecznie nowoczesnym oprogramowaniem, zaś zapisanie tego samego tekstu w kodzie 1250 preferuje polskich użytkowników oprogramowania Microsoft. Posiadający najbardziej oficjalny status kod nazywany potocznie ISO Latin-2 (ISO 8859-2, PN-91/T-42115) jest nadal najwygodniejszym sposobem kodowania polskich liter w systemach operacyjnych UNIX, ale nie jest to wystarczający powód, aby narzucać go użytkownikom innych systemów.

Reasumując, użytkownik komputera PC wyposażonego w typowy system operacyjny MS Windows i bezpłatne oprogramowanie sieciowe typu MS Internet Explorer lub Netscape Communicator nie powinien napotykać na problemy przy posługiwaniu się językiem polskim, zarówno lokalnie jak i w sieci Internet. Nieco gorzej wygląda sytuacja z innymi systemami operacyjnymi lub innymi typami komputerów, ale ostra konkurencja między producentami sprzętu i oprogramowania powinna wkrótce doprowadzić do zlikwidowania mogących jeszcze występować trudności w tym zakresie — oczywiście pod warunkiem, że klienci i użytkownicy będą się tego domagać.

Literatura

- [1] Jacques André, Michel Goossens. Codage des caractères et multilinguisme: de l'ASCII à UNICODE et ISO/IEC-10646. *Cahiers GUTenberg* no 20, mai 1995, pp. 1–53. Por. <http://www.gutenberg.eu.org/publications/cahiers/46-cahiers20.html> (dostęp 25.11.2005).
- [2] ANSI X3.4-1986. *Coded Character Set — 7-bit American Standard Code for Information Interchange*.
- [3] Janusz S. Bień. Lingwistyka informatyczna we Francji. *Polonica*, III:234–255, 1977.
- [4] Janusz S. Bień. *Koncepcja słownikowej informacji morfologicznej i jej komputerowej weryfikacji*, Rozprawy Uniwersytetu Warszawskiego t. 383. Wydawnictwa Uniwersytetu Warszawskiego, Warszawa, 1991.
- [5] Janusz S. Bień. Polskie litery na PC (głos w dyskusji). *ComputerWorld PL* nr 4(10), 16.II.1991, s. 9-11,14,16,19,21.
- [6] Janusz S. Bień. Strona kodowa 852 i syndrom żelaznej kurtyny. *Biuletyn Polskiego Towarzystwa Informatycznego* r. X nr 5, s. 3, 1991.
- [7] Janusz S. Bień. Wybrane standardy przetwarzania tekstów. Referat wygłoszony na konferencji *Komputerowa Analiza Tekstu*, Karpacz, 16–18

- listopada 1993, zorganizowanej przez Instytut Filologii Polskiej Uniwersytetu Wrocławskiego i Seminar für Slavistik, Universität Bochum. Por. <http://www.mimuw.edu.pl/~jsbien/publ/Wstp98/JSB-Wstp98.pdf>
- [8] Janusz S. Bień. *Język polski w sieciach komputerowych*. Raport Instytutu Informatyki UW TR98-01 (250), marzec 1998. Patrz także <http://www.mimuw.edu.pl/~jsbien/publ/Jpsk98/JSB-Jpsk98.pdf>.
- [9] Janusz S. Bień, Jadwiga Linde-Usiekiewicz. Elektroniczne słowniki języka polskiego. *Postscriptum* nr 24-25, zima '97 – wiosna '98, ISSN 1427-0501, s. 17-25.
- [10] Charles Bigelow, Kris Holmes. Création d'une police UNICODE. *Cahiers GUTenberg* no 20, mai 1995, pp. 81-102. Por. <http://www.gutenberg.org/publications/cahiers/46-cahiers20.html> (dostęp 25.11.2005).
- [11] Piotr Bolek. SGML w praktyce. *Biuletyn Grupy Użytkowników Systemu T_EX*, zeszyt 7, 1996, s.46-50.
- [12] ECMA-6. *7-Bit Coded Character Set*. Sixth Edition (December 1991), Reprinted August 1997. <http://www.ecma-international.org/publications/standards/Ecma-006.htm> (dostęp 25.11.2005).
- [13] ECMA-35. *Character Code Structure and Extension Techniques*. Sixth Edition (December 1994). <http://www.ecma-international.org/publications/standards/Ecma-035.htm> (dostęp 25.11.2005).
- [14] ECMA-43. *8-Bit Coded Character Set Structure and Rules*. Third Edition (December 1991). <http://www.ecma-international.org/publications/standards/Ecma-043.htm> (dostęp 25.11.2005).
- [15] ECMA-48. *Control Functions for Coded Character Sets*. Fifth Edition (June 1991). Reprinted June 1998. <http://www.ecma-international.org/publications/standards/Ecma-048.htm> (dostęp 25.11.2005).
- [16] ECMA-94. *8-Bit Single-Byte Coded Graphic Character Sets — Latin Alphabets No. 1 to No. 4*. Second Edition (June 1986). <http://www.ecma-international.org/publications/standards/Ecma-094.htm> (dostęp 25.11.2005).
- [17] ECMA-113. *8-Bit Single-Byte Coded Graphic Character Sets — Latin/Cyrillic Alphabet*. Second Edition (July 1988). <http://www.ecma-international.org/publications/standards/Ecma-113.htm> (dostęp 25.11.2005).
- [18] ECMA-114. *8-Bit Single-Byte Coded Graphic Character Sets — Latin/Arabic Alphabet*. June 1986. <http://www.ecma-international.org/publications/standards/Ecma-114.htm> (dostęp 25.11.2005).

- [19] ECMA-118. *8-Bit Single-Byte Coded Graphic Character Sets — Latin/Greek Alphabet*. December 1986. <http://www.ecma-international.org/publications/standards/Ecma-118.htm> (dostęp 25.11.2005).
- [20] ECMA-121. *8-Bit Single-Byte Coded Graphic Character Sets — Latin/Hebrew Alphabet*. July 1987. <http://www.ecma-international.org/publications/standards/Ecma-121.htm> (dostęp 25.11.2005).
- [21] ECMA-128. *8-Bit Single-Byte Coded Graphic Character Sets — Latin Alphabet No. 5*. July 1988. <http://www.ecma-international.org/publications/standards/Ecma-128.htm> (dostęp 25.11.2005).
- [22] ECMA-144. *8-Bit Single-Byte Coded Character Sets — Latin Alphabet No. 6*. Second Edition (December 1992) <http://www.ecma-international.org/publications/standards/Ecma-144.htm> (dostęp 25.11.2005).
- [23] Elektroniczny przedruk *Słownika języka polskiego PAN* pod red. W. Doroszewskiego. PWN 1997, ISBN 83-01-12321-4. Por. www.pwn.com.pl/rsjp¹¹.
- [24] Tomaz Erjavec, Ann Lawson, Laurent Romary (eds.). *East meets West — A Compendium of Multilingual Resources*. TELRI Association 1998.
- [25] Charles F. Goldfarb. *The SGML Handbook*. Clarendon Press: Oxford 1990.
- [26] ISO/IEC 646:1991. *Information technology — ISO 7-bit coded character set for information interchange*.
- [27] ISO/IEC 2022:1994. *Information technology — Character code structure and extension techniques*.
- [28] ISO 2375:1985. *Data processing — Procedure for registration of escape sentences*.
- [29] ISO 4873:1991. *Information technology — ISO 8-bit code for information interchange — Structure and rules for implementation*.
- [30] ISO/IEC 6429. *Information technology — Control functions for coded character sets*
- [31] ISO 6937:1984. *Coded graphic character set for text communication — Latin alphabet*

¹¹Aktualnie — 25.11.2005 — strona nie istnieje. Wersje demonstracyjne słownika dostępne są obecnie pod adresem <http://www.mimuw.edu.pl/~jsbien/PWN/>.

- [32] ISO-8859-2:1987. *Information Processing — 8-bit Single-Byte Coded Graphic Character Sets — Part 2: Latin alphabet No. 2.*
- [33] ISO 8879:1986. *Information processing — Text and office systems — Standard generalized Markup Language (SGML).*
- [34] ISO/IEC 9541-1: 1991. *Information technology — Font information interchange — Part 1: Architecture.*
- [35] ISO 10367:1991. *Information technology — Standardized coded graphic character sets for use in 8-bit codes.*
- [36] ISO/IEC 10646-1:1993. *Information Technology — Universal Multiple-octet Coded Character Set (UCS).* Patrz również uzupełnienia i poprawki redakcyjne. Por. <ftp://dkuug.dk/JTC1/SC2/WG2/docs/>.
- [37] ISO/IEC 10744:1992. *Information Technology — Hypermedia/Time-based Structuring Language.*
- [38] Charles E. Mackenzie. *Coded character sets, History and Development.* Addison-Wesley 1980. ISBN 0-201-14460-3.
- [39] PN-84/N-09120. *Zestaw dodatkowych znaków alfabetu łacińskiego do wymiany informacji bibliograficznych.*
- [40] PN-85/N-09121. *Zestaw znaków alfabetu greckiego do wymiany informacji bibliograficznej.*
- [41] PN-85/N-09122. *Zestaw znaków alfabetu cyrylicznego do wymiany informacji bibliograficznej.*
- [42] PN-84/T-42109/02. *Przetwarzanie informacji i komputery. Kod 7-bitowy. Krajowe zestawy znaków.*
- [43] PN-86/T-42109/03. *Przetwarzanie informacji i komputery — Kod 7-bitowy — Krajowy zestaw znaków wprowadzony techniką rozszerzania kodu.*
- [44] PN-88/T-42109/1. *Przetwarzanie informacji i komputery. Kod 7-bitowy. Tablica kodu i zestawy znaków ISO i RWPG.*
- [45] PN-89/T-42108. *Przetwarzanie informacji i komputery. Znaki alfanumeryczne. Klasyfikacja, nazwy i symbole.*
- [46] PN-91/T-42115. *Przetwarzania informacji — Zestaw znaków graficznych w jednobajtowym kodzie 8-bitowym — Alfabet łaciński nr 2.*
- [47] PN-93/N-09128. *Bibliograficzne znaki sterujące.*

- [48] PN-93/T-42109/01. *Technika informatyczna. Zestaw znaków w kodzie 7-bitowym ISO przeznaczony do wymiany informacji.*
- [49] PN-93/T-42112. *Technika informatyczna — Kod 8-bitowy przeznaczony do wymiany informacji — Budowa i zasady wdrażania.*
- [50] PN-93/T-42118. *Technika informatyczna — Znormalizowane zbiory znaków graficznych przeznaczone do stosowania w kodach 8-bitowych.*
- [51] PN-ISO/IEC 2022. *Technika informatyczna — Struktura kodu znaków i techniki rozszerzania.* Grudzień 1996.
- [52] PN-ISO/IEC 2382-23. *Technika informatyczna — Terminologia — Przetwarzanie tekstu.* Grudzień 1996.
- [53] PN-ISO/IEC 6429. *Technika informatyczna — Funkcje sterujące dla zestawów kodowanych znaków.* 1996.
- [54] RFC 2026/BCP 9. *The Internet Standards Process – Revision 3.* S. Bradner. October 1996.
- [55] RFC 2277/BCP 18. *IETF Policy on Character Sets and Languages.* H. Alvestrand, January 1998.
- [56] RFC 2278/BCP 19. *IANA Charset Registration Procedures.* N. Freed, J. Postel, January 1998.
- [57] RFC 2279. *UTF-8, a transformation format of ISO 10646.* F. Yergeau, January 1998.
- [58] RFC 2282/BCP 10. *IAB and IESG Selection, Confirmation, and Recall Process: Operation of the Nominating and Recall Committees.* J. Galvin, February 1998.
- [59] Sperberg-McQueen, C. M., & Burnard, L. (Eds.). *Guidelines for electronic text encoding and interchange (TEI P3).* Text Encoding Initiative, Chicago-Oxford 1994.
- [60] Andrzej Suski. Normalizacja. *Wielka Encyklopedia Powszechna* t. 8, s. 11-12. Państwowe Wydawnictwo Naukowe: Warszawa 1966.
- [61] **The Unicode Standard, Version 2.0**, The Unicode Consortium, Addison-Wesley, 1996. ISBN 0-201-48345-9. Por. także www.unicode.org.
- [62] Johan W. van Wingen. *Manual — Standards for the electronic exchange of personal data — Part 5: character sets.* Ministry of Interior: The Hague 1995. ISBN 90-5414-019-4.