ROCZNIKI POLSKIEGO TOWARZYSTWA MATEMATYCZNEGO
Seria II: WIADOMOŚCI MATEMATYCZNE XXIX (1990)

wmptm 1897

Ministerstwo Nauki i Szkolnictwa Wyższego

JANUSZ S. BIEŃ (Warszawa)

Co to jest TeX?*

Wstęp. Niemal od samego początku istnienia komputerów są one w mniejszym lub większym stopniu wykorzystywane również do sporządzania tekstów. Programy komputerowe nie są przecież niczym innym, jak szczególnego rodzaju tekstami, które wprowadza się do komputera i w razie potrzeby wyprowadza w postaci wydruku; odpowiednie środki służące do modyfikowania przechowywanych w komputerze tekstów programów stanowią niezbędny element oprogramowania każdego komputera. Naturalnym krokiem było stopniowe wykorzystywanie tych narzędzi do wprowadzania, modyfikowania i drukowania również dokumentacji opracowywanych programów, a w dalszej kolejności także innych tekstów. Rosnące wymagania użytkowników sprawiły, że coraz częściej celem było uzyskanie wydruku przypominającego bardziej tekst drukowany niż zwykły maszynopis — oczywiście, w granicach możliwości stosowanych przy tym urządzeń drukujących.

Pojawienie się tanich komputerów, w szczególności komputerów domowych i osobistych, poszerzyło krąg autorów stosujących komputery do sporządzania tekstów o osoby nie związane zawodowo z informatyką; ze względu na ich odmienne kwalifikacje i wymagania wpłynęło to istotnie na kierunek rozwoju stosowanych do tego celu programów. Jednocześnie postęp w dziedzinie urządzeń drukujących spowodował, że stosunkowo niedroga drukarka laserowa o jakości druku dorównującej tradycyjnej technice typograficznej mieści się wraz ze sterującym nią komputerem na blacie biurka (stąd angielskojęzyczne określenie *desktop publishing* dla działalności wydawniczej prowadzonej za pomocą takiego sprzętu).

---

* Wstępna wersja tego artykułu powstała w 1988 r. w ramach — kierowanego przez Zygmunta Saloniego — tematu *1.9. Metody formalne i maszynowe w opisie języków słowiańskich* problemu *CPBP 08.11 Język polski — źródła, słownictwo, gramatyka, odmiany i związki z innymi językami.* Autor dziękuje wszystkim, którzy przekazali mu swoje uwagi do niniejszego tekstu w różnych fazach jego powstawania — w szczególności koleżankom Hannie Kołodziejskiej i Annie Borkowskiej, a także koledze Krzysztofowi Szafranowi. Oczywiście, odpowiedzialność za świadome i nieświadome uproszczenia oraz ewentualne pomyłki ponosi wyłącznie autor.

W rezultacie tej ewolucji powstały liczne i różnorodne programy [12], które można podzielić na dwie główne klasy różniące się zasadniczo działaniem. W systemach *adnotacyjnych* (ang. *markup systems*) właściwy tekst opatruje się – w trakcie wprowadzania go do komputera za pomocą osobnego programu nazywanego *edytorem* [33] – specjalnymi adnotacjami, stanowiącymi mniej lub bardziej szczegółowe instrukcje opisujące wygląd finalnej postaci tekstu; tekst można wydrukować lub obejrzeć na ekranie dopiero po przetworzeniu go w całości. System TeX może być zaliczony do tej właśnie klasy systemów, a przykłady tekstów z adnotacjami przed przetworzeniem i po wydrukowaniu stanowią załącznik do niniejszego artykułu.

W systemach typu *co widzisz* [na ekranie] – *to dostaniesz* [na drukarce] (ang. *what you see is what you get*, w skrócie WYSIWYG) obraz na ekranie komputera jest w miarę możliwości wiernym obrazem odpowiedniego fragmentu tekstu po wydrukowaniu, a wszelkie zmiany w tekście są natychmiast uwidocznione – w szczególności już w trakcie wprowadzania tekstu do komputera widzimy go w postaci bardzo zbliżonej do ostatecznej. Oczywiście i systemy adnotacyjne, i systemy *co widzisz – to dostaniesz* mają swoje wady i zalety; choć są do pomyślenia programy integrujące obie koncepcje, droga do nich wydaje się jeszcze daleka.

Wkraczanie komputerów do różnych dziedzin życia nie ominęło również poligrafii, jednak rozwój programów do składania tekstów przebiegał tam inaczej. Niezbędne dla programisty szczegóły techniczne urządzeń składających były często traktowane jako tajemnica firmowa, co w naturalny sposób preferowało programy tworzone przez producentów tych urządzeń lub na ich zlecenie; również koszt eksperymentów niezbędnych przy opracowywaniu programu w przypadku profesjonalnego sprzętu poligraficznego był znacznie wyższy niż w przypadku drukarek komputerowych, co dodatkowo wzmacniało monopolistyczną pozycję producentów sprzętu w pracach nad zastosowaniem komputerów w poligrafii. Programy te były przeznaczone dla odpowiednio przeszkolonych pracowników drukarni, a więc osób z profesjonalnym wykształceniem typograficznym, ale bez wiedzy i nawyków charakterystycznych dla autora czy programisty. Motywacja tych prac była czysto ekonomiczna, komputeryzowano więc przede wszystkim skład łatwy, co w efekcie dawało relatywny wzrost kosztu składu ręcznego tekstów trudnych. Choć możliwości programów stopniowo rosły, wynikiem komputeryzacji drukarń było początkowo obniżenie jakości składu m.in. przez nakłanianie autorów i wydawców do dostosowywania się do możliwości dostępnych programów (np. przez umieszczanie przypisów na końcu rozdziałów lub całego dzieła).

Tak więc przez dłuższy czas istniały dwa niemal całkowicie niezależne nurty: mniej lub bardziej wyrafinowane programy przeznaczone dla autorów – niekiedy tworzone z czystej ciekawości intelektualnej lub dla zaspokojenia ambicji programisty – których jakość druku była ograniczona możli-

wościami komputerowych urządzeń drukujących, oraz stosowane w drukarniach programy nie wykorzystujące w pełni potencjalnych możliwości informatyki i sprzętu poligraficznego. Nurty te dzieliły (głównie z powodów historycznych) również kwestie czysto techniczne — w poligrafii używano przeważnie innej taśmy perforowanej i nadal często używa się innego typu dyskietek niż w informatyce. Nieliczne przypadki przełamania wszystkich barier technicznych, organizacyjnych i psychologicznych polegające na złożeniu w drukarni książki na podstawie dostarczonej przez autora taśmy magnetycznej (ten typ nośnika jest na szczęście identyczny dla obu dziedzin) wymagały dużego wysiłku i stanowiły powód do dumy wszystkich zainteresowanych stron.

Taki stan rzeczy profesor Donald Ervin Knuth — światowej sławy informatyk z Uniwersytetu Stanforda w Stanach Zjednoczonych — potraktował jako wyzwanie i sformułował program badawczy nazwany przez niego *typografią matematyczną* [17], polegający na wykorzystaniu aparatu matematyki do sformułowania reguł składania tekstów, przy czym same litery są w razie potrzeby traktowane jako macierze zer i jedynek, reprezentujących odpowiednio białe i czarne punkty na papierze. Powstały w wyniku tych prac system TeX to, według słów jego twórcy, *system składania tekstów przeznaczony do tworzenia pięknych książek — a zwłaszcza książek zawierających dużo matematyki* ([20], str. V), określenie to wymaga jednak pewnych modyfikacji i uzupełnień. Obecnie TeX to przede wszystkim zdefiniowany w niekonwencjonalny sposób zestaw standardów, a rodzina zgodnych z nimi programów jest coraz bardziej liczna i rozwija się dynamicznie — są one od pewnego czasu dostępne również na takich komputerach osobistych i domowych jak IBM PC, Apple Macintosh, ATARI ST czy Commodore Amiga. Do drukowania tekstu służyć mogą m.in. typowe drukarki komputerowe i drukarki laserowe, a także niektóre urządzenia poligraficzne; lista ta jest otwarta, a dodanie do niej nowego urządzenia wymaga (przynajmniej w założeniu) stosunkowo niewielkiego nakładu pracy programisty.

W systemie TeX sposób formowania tekstu opisujemy, wplatając we właściwy tekst odpowiednie *komendy*, co — z punktu widzenia użytkownika — czyni go bardzo zbliżonym do systemów adnotacyjnych; od typowego programu tej klasy różni się on przede wszystkim tym, że nawet w przypadku trudnych tekstów pozwala osiągnąć jakość składu porównywalną ze składem ręcznym wykonanym przez kompetentnego zecera. Jednocześnie różni się on od typowego programu stosowanego w drukarniach m.in. tym, że nie wymaga od użytkownika specjalistycznej wiedzy czy umiejętności. Jest to możliwe dzięki temu, że wiedza ta jest reprezentowana za pomocą odpowiednich reguł, częściowo wbudowanych w system, a częściowo zawartych w wymiennych tzw. *pakietach makrodefinicji*. W zasadzie zadanie użytkownika (będącego przeważnie autorem publikacji) sprowadza się więc do opisania struktury logicznej przetwarzanego tekstu za pomocą odpowiednich adnotacji o mniej lub bardziej mnemotechnicznych nazwach.

Pod względem ogólności i elastyczności niezbędnej do składania tekstów o wysokiej jakości typograficznej system TeX nie ma dotąd sobie równych [13]. Za ogólność i elastyczność płaci się jednak cenę w postaci złożoności systemu, co ma różnorodne konsekwencje: program jest duży i zużywa dużo mocy obliczeniowej, a do pełnego poznania możliwości systemu niezbędny jest znaczny wysiłek. Zwykły użytkownik nie musi jednak znać wszystkich niuansów systemu, ponieważ przygotowanie nawet skomplikowanego, ale typowego tekstu, np. przez wykorzystanie adnotacji stosowanych w innym artykule z tej samej dziedziny, jest w zasadzie sprawą prostą; głęboka wiedza o systemie jest potrzebna przede wszystkim autorom pakietów makrodefinicji. Z drugiej strony, uciążliwości korzystania z systemu TeX wynikające z ograniczeń obecnych komputerów osobistych będą się stopniowo zmniejszać w miarę wprowadzania nowych generacji sprzętu i oprogramowania podstawowego.

1. Symbol TeX — znaczenie, pisownia i wymowa. Symbol T_EX to zapisany dużymi literami początek greckiego słowa *τέχνη*, oznaczającego zarówno sztukę, jak i technikę; obniżenie litery E ma przypominać, że chodzi o składanie tekstów — kiedy nie jest to technicznie możliwe, należy pisać TeX. Ponieważ w języku angielskim w słowach greckiego pochodzenia odpowiednik tej litery brzmi jak *k* (w szczególności angielska wymowa nazwy litery *χ* może być w przybliżeniu oddana po polsku przez *kaj*), sam fakt, że ostatnią literą symbolu jest *chi*, a nie *iks*, nie przesądza jeszcze o jego wymowie. W krajach anglosaskich najczęściej wymawia się TeX jak *tek*. Niewątpliwie poprawna (i stosowana przez autora niniejszego tekstu) jest również wymowa taka, jak polskie czy niemieckie odczytanie napisu *tech*. Sprawie nazwy systemu poświęcony jest osobny jednostronicowy rozdział w podręczniku [20]; niestety, uwagi dotyczące rekomendowanej wymowy nie są dostatecznie precyzyjne.

2. Program TeX, system TeX i środowisko. Nazwa TeX bywa używana w dwóch znaczeniach. W węższym z nich chodzi o program TeX, który przetwarza zakodowany w komputerze tekst z adnotacjami na niezależną od urządzenia wyjściowego (a więc w szczególności od typu drukarki) reprezentację tekstu drukowanego, czyli tzw. plik DVI (ang. *DeVice Independent file*). Aby przetworzenie to — o którym piszemy więcej nieco dalej — było możliwe, program musi dysponować odpowiednią informacją o dostępnych zestawach znaków, które nazywamy tutaj *fontami* (ang. *font* lub *fount*); informacji tej dostarczają programowi *metryki* fontów zawarte w tzw. plikach TFM (ang. *TeX Font Metric files*); podają one m.in. rozmiary poszczególnych znaków. Plik DVI opisuje postać poszczególnych stron tekstu, który ma zostać wydrukowany, przez wskazanie dla każdego znaku tekstu jego współrzędnych na stronie z dokładnością do kilku nanometrów (10^{-9} m). Same znaki reprezentowane są w pliku DVI tylko przez nazwę fontu i numer znaku w foncie (choć normalnie korzysta się z wspomnianych niżej fontów Computer

Modern, można również używać fontów dostępnych wyłącznie na konkretnym urządzeniu wyjściowym, co oczywiście oznacza świadomą rezygnację z uniwersalności programu TeX).

Aby reprezentowane w postaci plików DVI teksty mogły być rzeczywiście niezależne od konkretnych urządzeń wyjściowych, niezbędne jest opisanie kształtów znaków w dostatecznie abstrakcyjny sposób. Możliwe są przy tym dwa poziomy abstrakcji. Pierwszy z nich to wyrażenie pewnego kształtu przez rozłożenie go na pojedyncze czarne lub białe punkty, reprezentowane z kolei przez odpowiednio dużą macierz zer i jedynek. Macierz taką będziemy nazywać reprezentacją *rastrową*, a jej elementy – *pikslami* (lub *pikselami*; jak podaje *Webster's New World Dictionary of the American Language*, ang. *pixel* to skrót od *pix element*, gdzie *pix* jest amerykańskim żargonowym określeniem filmu lub fotografii). Reprezentację rastrową będziemy odróżniać od *rastra*, który jest reprezentacją rastrową o określonej *rozdzielczości* (ang. *resolution*), tj. wielkości piksli (a więc i odległości między nimi), wyrażanej najczęściej przez liczbę piksli na jednostkę długości. Drugi, wyższy poziom abstrakcji to taka reprezentacja kształtów, która pozwala otrzymać reprezentację rastrową o dowolnej zadanej rozdzielczości; w przeciwieństwie do poprzedniej *dyskretnej* reprezentacji rastrowej tę drugą możemy nazywać *reprezentacją ciągłą*.

Wyrażona w zaprojektowanym przez Knutha formalizmie ciągła reprezentacja fontów (wykorzystująca istotnie funkcje gięte, ang. *spline*) stanowi – wraz z żądaną rozdzielczością – dane dla programu METAFONT [22], który na ich podstawie tworzy dwa typy plików: rastrowe reprezentacje fontów nazywane (trochę myląco) *fontami generycznymi* i ich metryki. Wykorzystując program METAFONT, można osiągnąć bardzo wysoki stopień ogólności przez odpowiednie sparametryzowanie równań określających kształty poszczególnych fragmentów liter, dzięki czemu różne odmiany fontów (a w szczególności różne ich wielkości) mogą być tworzone przez zmianę wartości wybranych parametrów. Nazwa METAFONT nawiązuje właśnie do tych potencjalnych możliwości. Zostały one wykorzystane przez twórcę systemu TeX do skonstruowania zestawu fontów o nazwie *Computer Modern* [24] (ich wcześniejsza wersja, oznaczana skrótem AM, nosiła roboczą nazwę *Almost computer Modern*). Zestaw ten obejmuje 75 fontów po 128 znaków.

Fonty dzielimy na kilka grup w zależności od ich *rozkładu*, tj. rodzaju zawartych w nich znaków i przyporządkowanych im kodów liczbowych. Jedną grupę stanowią *fonty tekstowe* do składania zasadniczej części tekstu, druga to fonty do reprodukowania wydruków programów komputerowych, trzy pozostałe grupy służą do składania formuł matematycznych. W grupie fontów tekstowych znajduje się kilka krojów pisma: antykwa (ang. *roman type*), kursywa (ang. *italic type*, *italics*), pismo pochyłe o rysunku antykwy (ang. *slanted type*), w odmianach różniących się grubością pisma, kształtem liter (występowaniem lub brakiem tzw. szeryfów) oraz wielkością, czyli tzw. *stop-*

niem pisma (ang. *design size*). Wszystkie kroje są dostępne w podstawowym stopniu 10 anglosaskich punktów typograficznych, w systemie TeX oznaczanych *pt* (od ang. *point*), a najczęściej używane fonty – również w stopniach od 5 do 9 pt oraz dodatkowo w stopniach 12 i 17 pt (punkt to tradycyjna miara drukarska wywodząca się podobno od średnicy kropki stosowanej do interpunkcji, a następnie na różne sposoby uściślana; w Polsce przyjęto definicję punktu zaproponowaną przez francuskiego typografa Didot – punkt ten, nieco większy od anglosaskiego, w systemie TeX oznaczany jest skrótem *dd* od ang. *Didot point*).

W odpowiednich fontach zestawu Computer Modern można znaleźć m.in. różne symbole matematyczne, pisane majuskuły łacińskie, duże i małe litery greckie oraz wybrane litery innych alfabetów obcych, powszechnie stosowane w matematyce, np. alef ($\aleph$) czy gotycka litera R ($\mathfrak{R}$); niektóre symbole (np. znak pierwiastka) mogą być w razie potrzeby tworzone w dowolnym rozmiarze z elementów składowych znajdujących się w odpowiednim foncie.

Program METAFONT swoimi rozmiarami i złożonością dorównuje programowi TeX, a może nawet go przewyższa. W praktyce program METAFONT jest wykorzystywany zawsze razem z różnymi programami pomocniczymi, stąd możemy mówić o systemie METAFONT stanowiącym składnik systemu TeX. Typowy użytkownik nie ma na szczęście bezpośredniego kontaktu z METAFONTem, chyba że postanowi go użyć – wbrew jego pierwotnemu przeznaczeniu – do tworzenia ilustracji w postaci rysunków i wykresów.

Rastrowa reprezentacja znaków jest tak wygodna, że obecnie stosuje ją większość drukarek komputerowych i fotoskładarek poligraficznych, tj. urządzeń do naświetlania tekstu na materiale światłoczułym (błonie lub papierze fotograficznym), który po odpowiedniej obróbce służy do wytrawienia formy drukarskiej. W konsekwencji urządzenia te mogą być wykorzystywane do drukowania tekstów przygotowanych za pomocą systemu TeX i wykorzystujących fonty Computer Modern; niezbędne są jednak w tym celu specjalne programy dostosowane do konkretnych urządzeń wyjściowych i w związku z tym nazywane *pilotami* (francuskie *pilote*, angielskie *driver*) tych urządzeń. Dość często chcemy przed wydrukowaniem obejrzeć tekst na ekranie komputera – umożliwiają to piloty ekranów komputerowych (ściślej, konkretnych sterowników graficznych – np. Hercules Graphics Card dla mikrokomputerów typu IBM PC).

Korzystanie z rastrowych urządzeń drukujących i fontów Computer Modern (lub innej abstrakcyjnej reprezentacji znaków) zapewnia pełną niezależność składanego tekstu od wykorzystywanego komputera i użytego urządzenia wyjściowego, ale może pociągać za sobą pewne niedogodności. W przypadku drukarek komputerowych zdarza się często, że urządzenia te działają szybciej, kiedy pracują z mniejszą rozdzielczością niż przewidziana dla dostępnych fontów Computer Modern lub kiedy korzystają z rastrów własnych

fontów przechowywanych w pamięci stałej (ang. ROM, *Read Only Memory*). W przypadku fotoskładu pojawiają się inne problemy różnorodnej natury, co powoduje, że stosowanie fontów Computer Modern ma sens tylko dla nielicznych typów naświetlarek. W przypadku obu typów urządzeń można jednak – przynajmniej teoretycznie – utworzyć metryki dla dostępnych fontów i sporządzić pilota dla danego urządzenia wyjściowego, dostosowanego do jego zasady działania; może to być więc np. drukarka rozetkowa odbijająca na papierze czcionki podobnie jak zwykła maszyna do pisania, lub naświetlarka Monophoto, w której fotograficzne matryce znaków są mechanicznie wybierane do naświetlenia we właściwym miejscu strony za pomocą układu optycznego z ruchomymi lustrami.

Program TeX został zaprojektowany w ten sposób, że niezmienne są tylko pewne podstawowe *komendy* sterujące procesem formowania tekstu, bardziej zaś złożone polecenia czy adnotacje realizuje się za pomocą tzw. *makrodefinicji*. W praktyce TeX jest używany zawsze z pewnym zestawem makrodefinicji, nazywanym dalej *pakietem makrodefinicji* lub krótko *pakietem*. Najbardziej znane są pakiety 'plain TeX' [20], $\mathcal{AMS}$-TeX [35] i LaTeX [30]. Na szczególną uwagę zasługuje ten ostatni, który staje się coraz bardziej popularny, m.in. dzięki wyposażeniu go w dodatkowe możliwości realizowane za pomocą osobnych programów. Najważniejszym z nich jest program BibTeX, ułatwiający tworzenie wykazów cytowanych prac dzięki wykorzystaniu opisów bibliograficznych przechowywanych w komputerze w specjalnych bazach danych. Istnieją również współpracujące z LaTeXem programy ułatwiające tworzenie indeksów i słowniczków.

Reasumując, TeX w szerszym znaczeniu – czyli *system* TeX – to zestaw różnorodnych programów i wykorzystywanych przez nie plików. W procesie *instalacji* systemu na konkretnym komputerze ustala się, które z nich będą wykorzystywane; w rzadkich przypadkach instalacja systemu TeX musi być poprzedzona instalacją podsystemu METAFONT w celu utworzenia fontów o odpowiedniej rozdzielczości. Ze względu na rozmiary systemu twardy dysk jest zawsze pożyteczny, a często nawet niezbędny.

Wykorzystywanie systemu TeX przebiega w kilku fazach: najpierw wprowadza się do komputera tekst wraz z adnotacjami za pomocą jakiegoś edytora, następnie przetwarza się go programem TeX. Niektóre usterki tekstu są wykrywane i sygnalizowane przez program, inne można dostrzec dopiero przy oglądaniu – za pomocą odpowiedniego pilota ekranu – sformowanego tekstu na ekranie monitora. Tak czy inaczej, w zależności od wyniku albo powraca się do fazy pierwszej w celu naniesienia zmian i poprawek, albo drukuje się tekst za pomocą pilota odpowiedniej drukarki. Przeważnie wykorzystuje się do drukowania tekstu drukarkę podłączoną – bezpośrednio lub pośrednio (za pomocą tzw. *sieci lokalnej*) – do danego komputera. Można jednak przygotować tekst do wydruku, zapisując wynik pracy pilota na odpowiednim pliku; do jego wydrukowania nie będzie już potrzebny system

TeX, ale dla niektórych typów drukarek objętość takiego pliku jest znaczna, a jego przeniesienie na inny komputer kłopotliwe.

Sposób, w jaki podczas pracy z systemem TeX przechodzi się z jednej fazy do drugiej, zależy od *środowiska programistycznego*, w szczególności od systemu operacyjnego eksploatowanego na danym komputerze. Ideałem jest, aby niektóre fazy mogły odbywać się równocześnie lub prawie równocześnie – np. praca z edytorem i oglądanie na ekranie złożonego tekstu – najlepiej przez podział ekranu na tzw. okna (ang. *windows*) przyporządkowane odpowiednim programom czy procesom. Niestety, środowiska takie są bardzo trudne do zrealizowania w sposób ogólny dla komputerów o małej mocy obliczeniowej i niewielkiej pamięci operacyjnej, dlatego są one rzadko dostępne na komputerach osobistych i domowych; kosztem pewnych ograniczeń można zamiast nich stosować różne namiastki, które jednak muszą być bardzo starannie dobrane do konkretnego zestawu komputerowego i preferowanego trybu jego wykorzystania.

3. Formalny status systemu TeX. System TeX projektowany był z myślą o jego szerokim upowszechnieniu i maksymalnym uniezależnieniu od konkretnych typów komputerów i stosowanych urządzeń wyjściowych. Dla łatwiejszego spełnienia tych postulatów Donald Knuth stworzył nowatorski system dokumentowania i uaktualniania programów, noszący nazwę WEB. Program zapisany w systemie WEB stanowi integralną całość ze swoją dokumentacją i w zależności od potrzeb jest prezentowany na dwa różne sposoby. Dla wykonania programu na komputerze jego zapis jest przetworzony przez program TANGLE, który m.in. porządkuje fragmenty programu w odpowiedni sposób i usuwa wszystkie komentarze. Dla prezentacji programu wraz z komentarzami jego zapis jest przetworzony przez program WEAVE, a następnie przez program TeX; w rezultacie otrzymujemy tekst, w którym fragmenty programu są zawarte w omawiającym jego działanie tekście i przedstawione w kolejności ustalonej przez autora, a nie wymuszonej składnią języka programowania. Co więcej, WEAVE wprowadza dodatkowe wyróżnienia typograficzne i konwencje notacyjne, niedostępne w językach programowania, a znacznie zwiększające czytelność tekstu.

Wykorzystanie systemu WEB do przechowywania tekstów programów TeX i METAFONT uczyniło możliwym i sensownym ich opublikowanie w postaci książkowej ([21], [23]), przy czym Knuth zrzekł się praw majątkowych do programów, ale zachował pełnię praw autorskich. Oznacza to w praktyce, że każdy zainteresowany jest uprawniony do wykorzystywania tych programów – w szczególności do zaadaptowania ich do dowolnego komputera – ale pod takim warunkiem, że otrzymane w rezultacie tych prac programy tylko wtedy mogą zachować oryginalne nazwy, jeśli ich wyniki są całkowicie zgodne z wynikami programów oryginalnych. W razie zamierzonych lub nie zamierzonych rozbieżności, nazwa programu musi być zmieniona,

a w celu egzekwowania powyższej zasady nazwa TeX została zarejestrowana przez American Mathematical Society (które w różnorodny sposób popiera rozwój systemu) jako zastrzeżony znak handlowy (z podobnych powodów nazwą zastrzeżoną jest również nazwa pakietu $\mathcal{AMS}$-TeX).

W trakcie opracowywania system TeX był sprawdzany przez Knutha na sprzęcie dostępnym w jego macierzystej instytucji, tj. na dużych komputerach firmy Digital Equipment Corporation (DEC10, DEC20), a kolejne etapy pracy były udostępniane zainteresowanym w postaci taśmy magnetycznej, zawierającej również te składowe systemu, które były publikowane tylko w postaci raportów Uniwersytetu Stanforda lub w ogóle nie publikowane (w szczególności wyrafinowany test na zgodność działania programu z intencjami autora, decydujący o tym, czy dany program ma prawo nosić nazwę TeX). Ta autorska wersja systemu nosi nazwę *systemu generycznego*; stanowi ona w gruncie rzeczy zdefiniowany w niekonwencjonalny sposób zestaw standardów, obejmujących między innymi następujące składowe:

– formalizm METAFONT do projektowania znaków,
– zapisane w formalizmie METAFONT fonty rodziny Computer Modern,
– zasady reprezentowania fontów generycznych, czyli struktura plików GF,
– zasady reprezentacji metrycznych własności fontów, czyli struktura plików TFM,
– formalizm TeX do zapisywania tekstów wraz z instrukcjami o sposobie składania,
– zasady reprezentacji tekstów drukowanych, czyli struktura plików DVI.

Standardy te ewoluują w miarę upływu czasu, np. pliki GF zastępują wcześniejszą reprezentację fontów w postaci tzw. plików PXL (ang. *PiXeL file*), a w praktyce coraz częściej stosuje się tzw. reprezentację spakowaną w postaci plików PK (ang. *PacKed file*) tworzonych na podstawie plików GF. Charakter wtórnego standardu mają również niektóre pakiety makrodefinicji – przede wszystkim pakiet LaTeX i związane z nim programy.

Udostępnienie generycznego systemu TeX na taśmie magnetycznej oraz w postaci publikowanej umożliwiło podejmowanie prac nad jego adaptacją do innych komputerów. Powstały w 1980 roku klub użytkowników systemu TeX (TeX Users Group – TUG) koordynował te wysiłki i informował o rezultatach. Z nielicznymi wyjątkami wyniki prac dla dużych komputerów traktuje się jako własność publiczną. Specjalnie powołany ośrodek dystrybucji rozprowadza je na taśmie magnetycznej – podobnie jak wersję generyczną – w zasadzie po kosztach własnych z niewielką dopłatą na cele dalszego rozwoju systemu (aktualnie stanowi to, wraz z kosztem taśmy, ale bez drukowanej dokumentacji, kwotę rzędu 100 $).

W przypadku mikrokomputerów (16-bitowych i 32-bitowych) sprawa przedstawia się inaczej. Z jednej strony, mikrokomputery dzięki swoim możliwościom graficznym zapewniają większą wygodę pracy przy składaniu tekstów niż typowe duże komputery, z drugiej zaś strony złożoność i rozmiary systemu TeX czyniły jego przeniesienie na mikrokomputer zadaniem niełatwym. W rezultacie większość mikrokomputerowych wersji systemu TeX ma charakter produktów komercyjnych – dla IBM PC najwcześniej pojawiły się dwa rywalizujące ze sobą systemy: PC-TeX firmy *Personal TeX, Inc.* i Micro-TeX znanego wydawnictwa Addison-Wesley; jak się wydaje, pierwsza z tych firm dominuje na rynku, chociaż – pomimo wycofania się Addison-Wesley z tego przedsięwzięcia – konkurencja w tej dziedzinie się istotnie zwiększyła. Niemniej jednak prace nad publicznymi wersjami systemu TeX dla mikrokomputerów są również prowadzone, na uwagę zasługuje tutaj m.in. zestaw pilotów dla różnego rodzaju drukarek opracowany przez Nelsona Beebe'a ([1], [2]).

Wydawany przez klub użytkowników TeXa biuletyn *TUGboat* informuje systematycznie – choć niekiedy z pewnym opóźnieniem – o nowych wersjach systemu, zasadach ich rozpowszechniania, dostępności pilotów dla różnych urządzeń wyjściowych itp. Od pewnego czasu podobne biuletyny wydawane są również w Wielkiej Brytanii (*TeXline*) i Francji (*Cahiers GUTenberg*). Najświeższe informacje są dostępne przede wszystkim w tzw. *biuletynach sieciowych*; są one rozpowszechniane za pomocą sieci komputerowych, do których można podłączyć się za pomocą zwykłej linii telefonicznej i komputera wyposażonego w tzw. *modem*. Niestety, w warunkach polskich biuletyny te są w praktyce całkowicie niedostępne.

4. Pierwotne przeznaczenie i historia systemu TeX. Twórca TeXa Donald E. Knuth jest jednocześnie autorem wielotomowej monografii *Sztuka programowania* [16]. Podstawowym impulsem do rozpoczęcia w maju 1977 roku prac nad składaniem tekstów było dostrzeżenie faktu, że nowe wydanie *Sztuki programowania* wykorzystujące skład komputerowy istotnie ustępuje jakością wydaniu wcześniejszemu, sporządzonemu za pomocą tradycyjnego składu monotypowego. Zadanie, jakie postawił sobie Knuth, to stworzenie systemu, który pozwoli utrzymać jakość składu ręcznego w kolejnych wydaniach jego monografii, i to nie tylko przy użyciu aktualnie dostępnych technik fotoskładu, ale również takich technik, które mogą pojawić się w przyszłości.

Pierwszym etapem pracy było zaprogramowanie w języku SAIL prototypowej wersji systemu TeX, nazywanej obecnie TeX78 [18]. Język SAIL dostępny był tylko na dużych komputerach firmy DEC, ale komputery te były bardzo popularne w zachodnich ośrodkach akademickich, co pozwoliło na szybkie rozpowszechnienie się systemu TeX w tych kręgach. Uzyskane doświadczenia zostały wykorzystane przy projektowaniu nowej wersji stosującej szeroko

rozpowszechniony język programowania Pascal – wersja ta nosiła początkowo nazwę TeX82.

W maju 1986 r. Knuth uznał swoje zadanie za zrealizowane [25], wyrażając przy tym pogląd, że TeX zachowa swoją przydatność co najmniej przez kilkadziesiąt lat.

Choć system w intencji jest maksymalnie ogólny, to wzorowanie się na składzie monotypowym i ukierunkowanie uwagi na konkretne dzieło wycisnęły jednak swoje piętno na koncepcji systemu. Na przykład, ponieważ *Sztuka programowania* cytuje takie polskie nazwiska jak *Łukasiewicz* i takie imiona jak *Zdzisław*, *Stanisław* i *Sławomir* (Pawlak, Ulam, Świerczkowski), polska litera *ł* została uwzględniona w fontach rodziny Computer Modern i zestawie makrodefinicji 'plain TeX'; z drugiej strony, ponieważ *Sztuka programowania* nie cytuje takich nazwisk jak np. *Wałęsa*, znak diakrytyczny w postaci ogonka nie został uwzględniony w systemie i jego stosowanie pociąga za sobą pewne komplikacje.

Program TeX może być przystosowany do pracy w niemal dowolnym języku o alfabetycznym systemie pisma przez sformułowanie w odpowiedni sposób metody dzielenia wyrazów i utworzenie – w razie potrzeby – odpowiednich fontów za pomocą systemu METAFONT; możliwości te zostały już praktycznie wykorzystane m.in. dla języka francuskiego [7]. Niestety, ze względu na swoje pierwotne przeznaczenie system TeX jest ukierunkowany na teksty jednojęzyczne, nie jest bowiem możliwa (przynajmniej w zasadzie) zmiana języka tekstu – a ściślej, metody dzielenia wyrazów – w trakcie pracy programu, chociaż pojedyncze słowa w innych językach mogą być cytowane bez większych trudności ze względu na bogaty zestaw znaków diakrytycznych dostępnych w fontach Computer Modern. W praktyce możliwe jest obejście tych ograniczeń przez użycie pewnych sztuczek technicznych [7] lub przez jawne zaznaczanie w tekście dopuszczalnych podziałów słów (ręcznie lub za pomocą odpowiedniego programu). Bardziej wygodne i eleganckie rozwiązanie tego problemu wymaga jednak modyfikacji programu TeX. Jak dotąd, szerzej znane są dwie takie propozycje: TêX (z akcentem nad literą e) do składania tekstów wielojęzycznych za pomocą standardowych fontów Computer Modern [9]–[11] oraz TeX-XeT umożliwiający łączenie tekstów pisanych z lewa na prawo z tekstami pisanymi z prawa na lewo [26].

5. Koncepcja i zasady działania systemu TeX. Podstawową innowacją systemu TeX jest wprowadzenie abstrakcyjnej reprezentacji tekstu drukowanego w postaci *pudeł* (ang. *box*) i *kleju* (ang. *glue*). Najprostszy element pudła to przede wszystkim *czcionka* (ang. *character*), a także pozioma lub pionowa linia (ang. *rule*) o zadanej grubości i długości. Oczywiście, wymiary czcionki – *wysokość*, *głębokość* i *szerokość* – zależą od kształtu jej *oczka*; np. *a*, *b*, c i inne litery bez tzw. wydłużeń dolnych mają głębokość równą zeru, a szerokość litery *m* jest z reguły dwukrotnie większa od szerokości litery *n*.

Słowo tekstu jest reprezentowane jako ciąg czcionek odpowiadających kolejnym literom; wiersz tekstu to *pudło poziome* (ang. *horizontal box*), w którym między słowami znajdują się warstwy kleju reprezentujące odstępy. Strona to *pudło pionowe* (ang. *vertical box*), w którym między pudłami reprezentującymi wiersze tekstu znajdują się odpowiednie warstwy kleju reprezentujące odstępy międzywierszowe. *Klej* najprościej jest opisać jako odpowiednik *justunku*, czyli *ślepego materiału zecerskiego* w składzie ręcznym, służącego do wypełniania odstępów między czcionkami, liniami itp.

W pewnych przypadkach budowa fragmentu tekstu musi być wskazana jawnie przez użytkownika za pomocą odpowiednich komend dla programu TeX; ma to miejsce na przykład wtedy, kiedy przez nałożenie na siebie dostępnych czcionek chcemy zbudować nowy symbol lub literę – ich wzajemne położenie i wymiary otrzymanego pudła określamy, wyrażając odpowiednie wielkości za pomocą stosownych miar typograficznych. Taką sytuację ilustruje dalej Przykład 7.

Znacznie częściej określamy tylko strukturę fragmentu tekstu, ustalenie niektórych wielkości pozostawiając programowi; w takich sytuacjach stosujemy klej, korzystając z faktu, że jest on elastyczny – w razie potrzeby warstwa kleju może zmieniać swoje rozmiary zgodnie z przypisanymi jej współczynnikami *kurczliwości* i *rozszerzalności*. Tak więc np. *wyśrodkowanie* pewnego napisu polega na umieszczeniu w pudle poziomym – o szerokości równej długości wiersza – warstwy kleju o nieskończonym współczynniku rozszerzalności, następnie pudeł z elementami napisu (czcionkami wyrazów i odstępami międzywyrazowymi), i w końcu jeszcze jednej warstwy kleju o nieskończonym współczynniku rozszerzalności – po skompletowaniu zawartości pudła rozszerzający się klej ustawi napis dokładnie pośrodku wiersza. W praktyce jednak całą tę operację wykonuje się prościej, korzystając z odpowiedniej makrodefinicji.

Najbardziej typowy przypadek mamy jednak wtedy, kiedy nie zależy nam na konkretnym podziale tekstu na wiersze, tylko na tym, aby podział ten był poprawny i edytorsko nie rażący – aby np. wyrazy nie były dzielone zbyt często, aby nie było ostatniego wiersza akapitu przeniesionego na nową stronę (czyli tzw. *bękarta*) itp. Program TeX – jak każdy system składania tekstów – może samodzielnie dokonać *łamania tekstu*, czyli jego podziału na wiersze, ma jednak dwie istotnie nowe własności. Po pierwsze, podstawową jednostką tekstu dla systemu jest *akapit* (ang. *paragraph*). Dzieląc akapit na wiersze, program przypisuje pewien koszt nazywany *grzywną* (ang. *penalty*) każdemu przejściu do nowego wiersza; przejście takie proponujemy nazywać technicznie *przełomem* (ang. *break*). Jednocześnie łatwość rozsunięcia przez klej pudeł ze słowami w celu otrzymania zadanej szerokości wiersza jest oceniana przez wyliczenie tzw. *wadliwości* (ang. *badness*); np. użycie do wyśrodkowania napisu kleju o dużej, ale nie wystarczającej rozszerzalności dałoby rezultat wizualnie podobny, ale zasygnalizowany przez system jako prawdopodobnie

błędny ze względu na wysoką wartość wadliwości. Biorąc pod uwagę wadliwość danego wiersza, związaną z nim grzywnę i pewne dodatkowe informacje o wierszach sąsiednich, system oblicza dla każdego wiersza jego *przywary* (ang. *demerits*), a następnie wybiera w akapicie taki ciąg przełomów, który mimimalizuje sumę przywar wszystkich wierszy tego akapitu.

Drugą istotną cechą systemu TeX jest możliwość interwencji użytkownika w proces łamania akapitu — albo przez wprowadzenie do tekstu *jawnych grzywien* (ang. *explicit penalty*) w celu wymuszenia lub zabronienia przełomu we wskazanych miejscach, albo przez manipulowanie współczynnikami liczbowymi określającymi np., jaka wartość grzywny wiąże się z dzieleniem wyrazu lub jaka jest liczbowa wartość przywary związanej z podzieleniem wyrazów na końcu dwóch kolejnych wierszy.

Jeżeli algorytm dzielenia wyrazów nie znajduje właściwego podziału pewnego słowa, to odpowiedni podział można wskazać w sposób ogólny przez wprowadzenie odpowiedniej informacji do *wykazu wyjątków*, lub w sposób ograniczony do konkretnego wystąpienia tego słowa przez użycie tzw. *przełomu uznaniowego* (ang. *discretionary break*). Możliwa jest również pełna wymiana tzw. *wzorców* sterujących algorytmem dzielenia wyrazów w celu np. przystosowania go do innego języka.

Z racji swojego pierwotnego przeznaczenia system posiada również specjalne środki do składania formuł matematycznych, które zilustrujemy przykładami nieco dalej.

6. Klasy użytkowników i zastosowań. Typowy użytkownik systemu składania tekstów często nie jest zainteresowany różnymi szczegółami typograficznymi, takimi jak np. wybór wielkości czy kroju czcionek. Co więcej, dość rozpowszechniony jest pogląd, że dla publikacji wysokiej jakości pewne decyzje tego typu może we właściwy sposób podjąć tylko kompetentny profesjonalista. Prowadzi to w naturalny sposób do podziału użytkowników TeXa na dwie klasy: tych, którzy przygotowują pakiety makrodefinicji i tych, którzy z nich korzystają. W przypadku pakietów $\mathcal{AMS}$-TeX i LaTeX, zadanie użytkownika drugiego typu polega na wybraniu właściwego *stylu* tekstu (np. książka, artykuł, list), a następnie na opisaniu logicznej struktury danego tekstu; dzięki tej dodatkowej informacji, makrodefinicje zawarte w wybranym stylu nadadzą przetwarzanemu tekstowi odpowiednią postać typograficzną. Na przykład dla pakietu LaTeX zasadnicza struktura książki może być określona przez wypełnienie schematu podanego w Przykładzie 1.

W przykładzie tym zasygnalizowaliśmy również pewne dodatkowe możliwości pakietu. Mianowicie po odpowiednim przetworzeniu za pomocą LaTeXa tak sporządzonych danych rozdziały zostaną automatycznie ponumerowane, a ich tytuły umieszczone w spisie treści wraz z właściwymi numerami stron. Pozycje bibliograficzne zostaną uporządkowane w żądany sposób, a symboliczne odwołania będą zastąpione odsyłaczami zgodnie z przyjętą

konwencją. Symboliczne odwołania do numerów stron zastępuje się aktualnymi numerami stron, na których wystąpiły odpowiednie etykiety; w analogiczny sposób można odwoływać się do formuł, tabel, ilustracji itp.

Podstawową klasą zastosowań systemu TeX są teksty matematyczne i inne teksty korzystające z aparatu matematycznego. Kolejne przykłady posłużą więc do zilustrowania składu formuł matematycznych. Występujące w nich komendy są przeważnie skrótami mnemotechnicznymi angielskich słów lub zwrotów — np. *int* od *integral* czyli *całka*, *frac* od *fraction* czyli *ułamek*, *eqalign* od *equation alignment* czyli *pionowanie* (dosłownie *wyrównanie*, *wyprostowanie*), jednakże do ich pełnego zrozumienia konieczne jest zapoznanie się z odpowiednimi fragmentami właściwych podręczników ([20], [35] lub [30]).

Przykład 2 przedstawia zapis dla pakietu 'plain TeX' lub LaTeX prostej *formuły wystawionej* (nazywanej też *formułą wydzieloną*):

$$\Phi = 2^{2^{2^2}}$$

(Podany w artykule Marka Lao [31] zapis tej formuły nie jest obecnie poprawny, ponieważ był on przeznaczony dla wcześniejszej wersji systemu TeX [19].) Użyta w przykładzie komenda *mit* (ang. *mathematical italics*) oznacza, że litera Φ ma być złożona kursywą. Komenda ta jest potrzebna tylko dlatego, że w typowych użyciach duże litery greckie składa się antykwą; w większości wypadków program samodzielnie wybiera właściwy krój pisma.

Przykład 3 ilustruje tzw. pionowanie:

$$\begin{aligned} x+x+2-2x &= 2x+2-2x \\ &= 2. \end{aligned}$$

Podajemy go w trzech wariantach zapisu: dla pakietu 'plain TeX', $\mathcal{AMS}$-TeX i LaTeX. Pochodzi on również z artykułu [31], ale do jego zapisu wkradły się tam pomyłki.

Dalsze przykłady zaczerpnąłem z instrukcji [36] dla zecerów Wrocławskiej Drukarni Naukowej (prawdopodobnie pochodzą one z polskich publikacji matematycznych), co zainteresowanemu czytelnikowi pozwala porównać zawarte w niej rekomendacje ze składem dokonanym przez system TeX, a konkretnie — przez pakiet LaTeX opisany w książce [30].

Przykład 4 to następująca formuła ([36], str. 22):

$$\frac{1}{\Gamma(a)}\int_0 (t-\tau)^{\alpha-\eta_{n+k_0}} y_\lambda(\lambda,\tau)\,d\tau = \sum_{\nu=1}^{n} \beta^\nu \delta^{\alpha_\nu} \int_0 \frac{(t-\tau)^{\alpha-s_{n+k_0}+w}}{\Gamma(\alpha-\alpha_\lambda)}\,d\tau.$$

Jej zapis dla pakietu LaTeX nie stwarza problemów, ponieważ zawiera ona typowe składowe. Niektóre nawiasy klamrowe występujące w zapisie mogą być pominięte, zostały one użyte dla lepszego uwidocznienia struktury formuły. Warto dodać, że wyrafinowane edytory pozwalają przy wprowadzaniu tekstu łatwo sprawdzić, czy nawiasy są zagnieżdżone we właściwy sposób.

Przykład 5 to następująca formuła ([36], str. 25):

$$\int_{AY^{nt^{bm}}xyz}^{NRAM_{m_{nd^2}}LQ^{rw^{zty}}} \frac{\sin^2 \frac{1}{2}nt}{2\sin^2 \frac{1}{2}t}\,dt = \int_{A}^{CDEU} \frac{1-\cos nt}{(2\sin \frac{1}{2}t)^2}\,dt = \frac{1}{2\operatorname{tg} \frac{1}{2}\alpha^w} + O\left(\frac{1+t_n}{2nam}\right).$$

Ponieważ system TeX przewiduje inny skrót na oznaczenie tangensa, wskazane było wprowadzenie odpowiedniej makrodefinicji przez mechaniczną modyfikację definicji zawartej w pakiecie 'plain TeX' ([20], str. 361).

Przykład 6 ([36], str. 17) okazał się najbardziej skomplikowany:

$$\mathfrak{UB} = \mathfrak{U}(x) = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{pmatrix}.$$

Wystąpiły tutaj dwojakiego rodzaju trudności. Po pierwsze, niedostępność pełnego alfabetu gotyckiego w fontach Computer Modern spowodowała użycie niestandardowego fontu pochodzącego z zestawu opracowanego przez Amerykańskie Towarzystwo Matematyczne [3]. Po drugie, wiersz kropek na całą szerokość macierzy nie został uznany przez twórcę pakietu LaTeX za konstrukcję typową i właściwy zapis tego elementu wymagał dobrej znajomości programu TeX lub dostrzeżenia podobnego przykładu jako zadania dla czytelnika w książce [20] (na str. 245).

Chociaż system TeX został stworzony z myślą o tekstach matematycznych, znalazł on szerokie zastosowanie również do tekstów technicznych i humanistycznych. Więcej informacji na ten temat można znaleźć we wspomnianych wcześniej drukowanych biuletynach, a także w materiałach konferencji użytkowników systemu — np. [32], [8], [14], [6].

Ostatni przykład załączony do niniejszego artykułu ilustruje zadania użytkowników odpowiedzialnych za konserwację systemu i jego adaptację do lokalnych potrzeb. Demonstruje on takie możliwości TeXa, które zbliżają go do języków programowania. Są to m.in. możliwości deklarowania *rejestrów* będących odpowiednikami zmiennych, definiowania nowych komend — w tym także komend z parametrami, stosowania instrukcji warunkowych. Przeznaczenie przytoczonego zestawu definicji — związane z przetwarzaniem tekstów polskich — zostanie omówione w punkcie następnym.

7. TeX w Polsce. Informacja o pierwszej wersji systemu TeX dotarła do Polski jeszcze w 1978 roku wraz z raportami wydziału informatyki Uniwersytetu Stanforda otrzymywanymi przez Instytut Informatyki Uniwersytetu Warszawskiego. Kiedy powstała druga wersja TeXa zrealizowana w szeroko dostępnym języku programowania Pascal, rozpocząłem starania o jej sprowadzenie; wydawało się wówczas, że jedynym wchodzącym w grę urządzeniem wyjściowym może być tylko fotoskładarka poligraficzna w rodzaju Mono-

photo 600 [34]. Moje próby zainteresowania systemem TeX środowiska matematycznego okazały się nieskuteczne. Jedynym ich efektem było otrzymywanie w ciągu kilku lat przez bibliotekę Instytutu Matematycznego PAN wspomnianego wcześniej biuletynu klubu użytkowników *TUGboat*. W końcu dzięki pokryciu kosztów przez jednego z pracowników Instytutu Informatyki UW (przebywającego wówczas w Stanach Zjednoczonych) w styczniu 1983 roku uzyskałem generyczną wersję 0.8 systemu TeX, co pozwoliło rozpocząć pracę nad uruchomieniem systemu na komputerze IBM 370/148 dysponującym prymitywną drukarką graficzną.

Najtrudniejsze etapy pracy — dostosowanie do systemu operacyjnego VM kompilatora języka Pascal opracowanego w Instytucie Podstaw Informatyki PAN dla systemu operacyjnego OS, a następnie rozszerzenie możliwości kompilatora o cechy niezbędne do kompilacji tekstu programu TeX — zostały wykonane przez Piotra Carlsona, który również rozpoczął adaptację odpowiednich programów do dialektu Pascala akceptowanego przez kompilator. Pilot dla drukarki graficznej IBM3287 został zaprogramowany przez Hannę Kołodziejską, która następnie przejęła od Piotra Carlsona pozostałe prace adaptacyjne. W rezultacie w kwietniu 1985 roku zostały złożone pierwsze próbne teksty za pomocą wersji 0.8 systemu TeX, a we wrześniu — za pomocą wersji 1.1 otrzymanej dzięki życzliwości użytkowników TeXa z uniwersytetu w Sztokholmie i z instytutu badawczego IRISA w Rennes. Następnym krokiem miało być przeniesienie systemu na komputer RIAD lub BASF (zgodne odpowiednio z IBM 360 i IBM 370) w Centrum Informatycznym Uniwersytetu Warszawskiego z wykorzystaniem — przynajmniej początkowo — polskiej drukarki graficznej D-200 jako urządzenia wyjściowego. Szybkie rozpowszechnienie się w kraju mikrokomputerów i importowanych drukarek graficznych spowodowało jednak poniechanie tego zamiaru na rzecz skompletowania dla mikrokomputerów typu IBM PC publicznie dostępnej realizacji systemu — w momencie pisania tych słów dysponujemy m. in. systemem DosTeX [15] wersja 2.93 z pilotem ekranu (dla sterownika Hercules Graphics Card) i pilotem mozaikowej drukarki 9-igłowej (typu Epson).

O ile mi wiadomo, w innych instytucjach w kraju na przełomie 1986 i 1987 roku zaczęto korzystać z systemu TeX na mikrokomputerach typu IBM PC. Jest on obecnie na tyle rozpowszechniony, że co najmniej jedna firma oferuje odpłatne sporządzanie wydruków na drukarce laserowej na podstawie przygotowanych przez klienta za pomocą systemu TeX plików. Można sądzić, że z czasem rozpowszechnią się również — niektóre podobno bardzo udane — realizacje dla mikrokomputerów ATARI, AMIGA i Macintosh.

Choć system TeX jest obecnie systematycznie wykorzystywany m.in. do składania miesięcznika *Delta*, nie wszystkie problemy składu tekstów polskich zostały do końca rozwiązane. Ważnym krokiem w tym kierunku było opracowanie przez Hannę Kołodziejską wzorców do dzielenia wyrazów zgodnie z regułami ortografii polskiej [27]. Pewne idee dotyczące traktowania

polskich znaków diakrytycznych zawiera notatka [4], a dodatkowe informacje na temat przetwarzania TeXem tekstów polskich można znaleźć w referatach [28] i [29].

Do czasu kompleksowego rozwiązania problemu przetwarzania tekstów polskich konieczne jest stosowanie różnych rozwiązań prowizorycznych. Przykładem tego typu rozwiązania jest zestaw definicji PL.STY (nazwany tak od nazwy pliku, w którym jest zwyczajowo przechowywany na dysku lub dyskietce). Jego pierwszą wersję opracowałem jeszcze w 1985 roku na komputerze IBM 370, a fragmenty ostatniej wersji stanowią załączony Przykład 7. Definicje te umożliwiają reprezentowanie polskich liter w jednolity sposób, a mianowicie przez poprzedzenie cudzysłowem (lub innym wybranym przez użytkownika znakiem) odpowiedniej litery angielskiej. Litery z ogonkiem są konstruowane w sposób opracowany przez Jerzego Rylla, przy czym pozycja ogonka może być w razie potrzeby łatwo skorygowana; w definicjach wykorzystałem też pewne sugestie Hanny Kołodziejskiej. Pominięte fragmenty dotyczą głównie tworzenia polskich liter w rzadziej używanych fontach o *rozkładzie terminalowym* (ang. [*tele*]*typewriter layout*).

8. Uwagi końcowe. Ze względu na charakterystyczny dla informatyki szybki rozwój sprzętu i oprogramowania uznałem za niecelowe umieszczanie w niniejszym artykule łatwo dezaktualizujących się informacji dotyczących konkretnych realizacji systemu i ich własności użytkowych. Niektóre informacje tego typu zebrałem w opracowaniu [5], które — jeśli okoliczności na to pozwolą — będę okresowo uaktualniał.

Załączone przykłady, które są reprodukowane z dostarczonych przeze mnie oryginałów, zostały wydrukowane na drukarce 9-igłowej z wykorzystaniem fontów o rozdzielczości 240 piksli na cal w poziomie i 216 piksli na cal w pionie.

P.S. Wersja 3 systemu TeX nie ma już większości ograniczeń dokuczliwych m.in. przy składaniu tekstów polskich.

PRZYKŁAD 1. Schemat zapisu książki dla pakietu LaTeX

```
\title{...}
\author{...}
\documentstyle{book}
\begin{document}
\maketitle
\tableofcontents
\chapter{...}
...
... \cite{JK} ...
...
... \label{l1} ...
...
\chapter{...}
...
... \pageref{l1} ...
...
... \cite{JN} ...
...
\begin{thebibliography}{00}
...
\bibitem{JK}
Jan Kowalski ...
...
\bibitem{JN}
Jan Nowak ...
...
\end{thebibliography}
\end{document}
```

PRZYKŁAD 2

Formuła złożona za pomocą pakietu 'plain TEX'

$$\Phi = 2^{2^{2^{2}}}$$

Zapis powyższej formuły dla pakietu 'plain TEX'

```
$${\mit\Phi}=2^{2^{2^{2^2}}}$$
```

PRZYKŁAD 3a

Formuła złożona za pomocą pakietu 'plain TEX'

$$\begin{aligned} x+x+2-2x &= 2x+2-2x \\ &= 2 \end{aligned}$$

Zapis powyższej formuły dla pakietu 'plain TEX'

```
$$
\eqalign{x+x+2-2x&=2x+2-2x\cr
                 &=2}
$$
```

PRZYKŁAD 3b

Formuła złożona za pomocą pakietu $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

$$\begin{aligned} x+x+2-2x &= 2x+2-2x \\ &= 2 \end{aligned}$$

Zapis powyższej formuły dla pakietu $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

```
$$
  \align
  x+x+2-2x&=2x+2-2x\\
          &=2\\
  \endalign
$$
```

PRZYKŁAD 3c

Formuła złożona za pomocą pakietu LATEX

$$\begin{aligned} x+x+2-2x &= 2x+2-2x \\ &= 2 \end{aligned}$$

Zapis powyższej formuły dla pakietu LATEX:

```
\begin{eqnarray*}
  x+x+2-2x&=&2x+2-2x\\
          &=&2
\end{eqnarray*}
```

PRZYKŁAD 4

Formuła złożona za pomocą pakietu LaTeX

$$\frac{1}{\Gamma(a)}\int_0(t-\tau)^{\alpha-\eta_{n+k_0}}y_\lambda(\lambda,\tau)\,d\tau=\sum_{\nu=1}^{n}\beta^\nu\delta^{\alpha_\nu}\int_0\frac{(t-\tau)^{\alpha-s_{n+k_0}+w}}{\Gamma(\alpha-\alpha_\lambda)}\,d\tau$$

Zapis powyższej formuły dla pakietu LaTeX

```
\[
\frac{1}{\Gamma(a)}
\int\limits_{0}(t-\tau)^{\alpha-\eta_{n+k_{0}}}
                    y_{\lambda}(\lambda,\tau)
\,d\tau
=
\sum_{\nu=1}^{n}
    \beta^{\nu}\delta^{\alpha_{\nu}}
\int\limits_{0}^{}
          \frac{(t-\tau)^{\alpha-s_{n+k_{0}}+w}}
          {\Gamma(\alpha-\alpha_{\lambda})}
\,d\tau
\]
```

PRZYKŁAD 5

Formuła złożona za pomocą pakietu LaTeX

$$\int\limits_{AY^{nt^{bm}}xyz}^{NRAM_{m_{nd^2}}LQ^{rw^{zty}}} \frac{\sin^2\frac{1}{2}nt}{2\sin^2\frac{1}{2}t}\,dt = \int\limits_{A}^{CDEU} \frac{1-\cos nt}{(2\sin\frac{1}{2}t)^2}\,dt = \frac{1}{2\,\mathrm{tg}\,\frac{1}{2}\alpha^w} + O\left(\frac{1+t_n}{2nam}\right)$$

Zapis powyższej formuły dla pakietu LaTeX

```
\newcommand{\tg}{\mathop{\rm tg}\nolimits}
\[
\int\limits_{AY^{nt^{bm}}xyz}
           ^{NRAM_{m_{nd^{2}}}LQ^{rw^{zty}}}
    \frac{\sin^{2}\frac{1}{2}nt}
         {2\sin^{2}\frac{1}{2}t}
\,dt
=
\int\limits_{A}^{CDEU}
    \frac{1-\cos nt}
         {(2\sin\frac{1}{2}t)^{2}}
\,dt
=
\frac{1}
     {2\tg\frac{1}{2}\alpha^{w}}
+O\left(\frac{1+t_{n}}{2nam}
  \right)
\]
```

PRZYKŁAD 6

Formuła złożona za pomocą pakietu LaTeX

$$\mathfrak{U}\mathfrak{B} = \mathfrak{U}(x) = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{pmatrix}$$

Zapis powyższej formuły dla pakietu LaTeX

```
\newfont{\gotyk}{eufm10 scaled \magstep1}
\newcommand{\U}{\hbox{\gotyk U}}
\newcommand{\B}{\hbox{\gotyk B}}
\[
\U\B=\U(x)=
\left(
      \begin{array}
      {rrrrrrr}
       a_{11}x_{1}&+&a_{12}x_{2}&+&\ldots&+&a_{1n}x_{n}\\
       a_{21}x_{1}&+&a_{22}x_{2}&+&\ldots&+&a_{2n}x_{n}\\
       \multicolumn{7}{c}{\dotfill}\\
       a_{m1}x_{1}&+&a_{m2}x_{2}&+&\ldots&+&a_{mn}x_{n}\\
      \end{array}
\right)
\]
```

PRZYKŁAD 7

Fragmenty zestawu makrodefinicji PL.STY

```
\newdimen\plak
...
\newdimen\plZr
\def\ogonekbox#1{%
     \hbox{\setbox0\hbox{\ogonekfont\ogonekchar}%
          \dp0 #1\box0}}
\chardef\ogonekchar'054% \lhook
\def\diakryt#1{\relax
     \leavevmode
     \ifx a#1a\kern\plak
               \lower\plal\ogonekbox{\plad}%
               \kern\plaw\else
     \ifx A#1A\kern\plAk
               \lower\plAl\ogonekbox{\plAd}%
               \kern\plAw\else
     \ifx c#1\accent'023 c\else
     \ifx C#1\accent'023 C\else
     \ifx e#1e\kern\plek
               \lower\plel\ogonekbox{\pled}%
               \kern\plew\else
     \ifx E#1E\kern\plEk
               \lower\plEl\ogonekbox{\plEd}%
               \kern\plEw\else
     \ifx l#1\poprzeczka\else
     \ifx L#1\Poprzeczka\else
     \ifx n#1\accent'023 n\else
     \ifx N#1\accent'023 N\else
     \ifx o#1\accent'023 o\else
     \ifx O#1\accent'023 O\else
     \ifx r#1\kropka\else
     \ifx R#1\Kropka\else
     \ifx s#1\accent'023 s\else
     \ifx S#1\accent'023 S\else
     \ifx z#1\accent'023 z\else
     \ifx Z#1\accent'023 Z\else
       \errmessage
          {niepoprawna sekwencja: diakryt litera}
     \fi\fi\fi\fi\fi\fi\fi\fi\fi
     \fi\fi\fi\fi\fi\fi\fi\fi\fi}
```

```
\def\tpoprzeczka{\char'401}
\def\tPoprzeczka{\setbox0=\hbox{L}%
                 \hbox to \wd0{\hss\char'40L}}
...
\def\tkropka{\accent'137z}
\def\tKropka{\accent'137Z}
...
\def\tlay#1{\font\ogonekfont=#1%
    \let\poprzeczka\tpoprzeczka
\let\Poprzeczka\tPoprzeczka
    \let\kropka\tkropka\let\Kropka\tKropka}
...
\plak-.21em\plAk-.21em\plek-.26em\plEk-.26em
\plal.79ex\plAl.79ex\plel.77ex\plEl.77ex
...
\tlay{cmmi7}
\catcode`"\active
\let"\diakryt
% P"OJD"Z, KI"N-"RE T"E CHMURNO"S"C W G"L"AB FLASZY
```

Prace cytowane

[1] Nelson H. F. Beebe, *Public domain TeX DVI driver family*, TUGboat 8 (1) (April 1987), 41–42.

[2] –, *A TeX DVI driver family*, w: [14], 71–114.

[3] Barbara Beeton, *Mathematical symbols and cyrillic fonts ready for distribution* (revised), TUGboat 6 (2) (November 1985), 124–128.

[4] Janusz S. Bień, *Polish language and TeX*, TeXline 8 (January 1989), 2.

[5] –, *Wykorzystanie systemu TeX na komputerach typu IBM PC*, Instytut Informatyki Uniwersytetu Warszawskiego, 1988 (tekst powielony).

[6] Malcolm Clark (red.), *Tex: Applications, Users, Methods. Proceedings of the Third European TeX Conference*, Ellis Horwood, Chichester 1990.

[7] Jacques Désarménien, *The use of TeX in French: hyphenation and typography*, w: [32], 41–59.

[8] – (red.), *Proceedings of the Second European Conference on TeX for Scientific Documentation*, Springer-Verlag, Berlin 1986.

[9] Michael J. Ferguson, *A multilingual TeX*, TUGboat 6 (2) (July 1985), 57–58.

[10] –, *A (hopefully) final extension of multilingual TêX*, TUGboat 8 (2) (July 1987), 102–103.

[11] –, *A multilingual TêX*, w: [8], 65–73.

[12] Richard Furuta, Jeffrey Scofield, Alan Show, *Document formatting systems: survey, concepts and issues*, ACM Comput. Surveys 14 (3) (September 1982), 417–472.

[13] Richard Goldstein, James Loomis, Avram Tetewsky, *Technical word processors for the IBM PC and compatibles. Report by the Boston Computer Society*, Notices Amer. Math. Soc. 34 (1987) (1) (January), 15–32 (Part I); (2) (February), 262–281 (Part IIA); (3) (April), 462–491 (Part IIB).

[14] Dean Guenther (red.), *Proceedings of the Eighth TeX Users Group Annual Meeting*, TeXniques Series No. 5, TeX Users Group, Providence, R.I., 1988.
[15] Alan Hoening, Mitch Pfeffer, *Grapevine report of inexpensive versions of TeX*, TUGboat 9 (1) (April 1988), 47–48.
[16] Donald E. Knuth, *The Art of Computer Programming*, Vol. 1: *Fundamental Algorithms*, Vol. 2: *Seminumerical Algorithms*, Vol. 3: *Sorting and Searching*, Addison-Wesley, Reading, Mass., 1968, 1969, 1973.
[17] —, *Mathematical Typography*, Report Stan-CS-78-648, Stanford University, Department of Computer Science, 1978; także Bull. Amer. Math. Soc. (N. S.) 1 (2) (March 1979), 337–372, oraz w [19].
[18] —, *TAU EPSILON CHI. A System for Technical Text*, Report Stan-CS-78-675, Stanford University, Department of Computer Science, 1978. Także w [19].
[19] —, *TeX and Metafont: New Directions in Typesetting*, Amer. Math. Soc. and Digital Press, Bedford, Mass., 1979.
[20] —, *The TeXbook*, Computer and Typesetting Series Vol. A, Addison-Wesley, Reading, Mass., 1986.,
[21] —, *TeX: The Program*, Computer and Typesetting Series Vol. B, Addison-Wesley,Reading, Mass., 1986.,
[22] —, *The METAFONTbook*, Computer and Typesetting Series Vol. C, Addison-Wesley, Reading, Mass., 1986.
[23] —, *METAFONT: The Program*, Computer and Typesetting Series Vol. D, Addison-Wesley, Reading, Mass., 1986.
[24] —, *Computer Modern Typefaces*, Computer and Typesetting Series Vol. E, Addison-Wesley, Reading, Mass., 1986.
[25] —, *Remarks to celebrate the publication of Computers & Typesetting*, TUGboat 7 (2) (June 1986), 95–98.
[26] Donald [E.] Knuth, Pierre MacKay, *Mixing right-to-left texts with left-to-right texts*, TUGboat 8 (1) (April 1987), 14–25.
[27] Hanna Kołodziejska, *Dzielenie wyrazów polskich w systemie TeX*, Sprawozdania Instytutu Informatyki Uniwersytetu Warszawskiego nr 165, 1987.
[28] —, *TeX*, Komputerowe wspomaganie prac wydawniczych. Wydawnictwa Szkolne i Pedagogiczne, Ośrodek Doskonalenia Kadr Technicznych RS NOT, Warszawa 1987, 49–54. Także: Komputerowe wspomaganie prac wydawniczych 1 (9.02.1988); Wydawnictwa Szkolne i Pedagogiczne, Koło Użytkowników Mikrokomputerów Profesjonalnych [ODKT RS NOT], Warszawa 1988, 14–18. Także: Mikroklan 1 (1988), 12.
[29] —, *Le traitement des textes polonais avec le logiciel TeX*, Cahiers GUTenberg, Numéro zéro (Avril 1988), 3–10.
[30] Leslie Lamport, LaTeX — *A Document Preparation System*, Addison-Wesley, Reading, Mass., 1985.
[31] Marek J. Lao, *Sztuka TeXowania*, Wiadom. Mat. 27 (1986), 81–87.
[32] Dario Lucarella (red.), *Proceedings of the First European Conference on TeX for Scientific Documentation*, Addison-Wesley, Reading, Mass., 1985.
[33] Norman Meyrowitz, Andries van Dam, *Interactive editing systems*, ACM Comput. Surveys 14 (3) (September 1982), 319–352 (Part I), 353–415 (Part II).
[34] *Programming Information for the 'Monophoto' 600 filmsetting system*, The Monotype Corporation Limited, Salfords, 1971.
[35] Michael Spivak, *The Joy of TeX. A Gourmet Guide to Typeseting with the AMS-TeX Macro Package*, Amer. Math. Soc., Providence, R.I., 1987.
[36] *Układ druków matematycznych. Instrukcja wewnętrzna dla pracowników drukarni*, Wrocławska Drukarnia Naukowa, Wrocław 1953.