

РОЗРОБКА ПАРСЕРУ ДЛЯ ІНТЕГРАЦІЇ МЕНЕДЖЕРІВ БІБЛІОГРАФІЙ ТА ПРОГРАМОВАНОЇ СИСТЕМИ КОМП'ЮТЕРНОЇ ВЕРСТКИ TYPST, ЯКИЙ ВІДПОВІДАЄ ВИМОГАМ БІБЛІОГРАФІЧНОГО СТАНДАРТУ ДСТУ 8302:2015А.О. Стопакевич¹, О.А. Стопакевич²¹Державний університет інтелектуальних технологій та зв'язку

1, Кузнечна вул., Одеса, 65023, Україна

²Національний університет «Одеська Політехніка»

1, Шевченка пр., Одеса, 65044, Україна

Email: stopakevich@gmail.com

Менеджер бібліографії (на кшталт Zotero, Mendeley, EndNote) є незамінним інструментом для багатьох науковців. Вони не тільки дозволяють структурувати інформацію, але й здатні тільки за DOI чи ISBN додати в базу менеджера бібліографічну інформацію, яку розмістив видавець чи журнал в таких системах як CrossRef. Проте для науковців, які застосовують національні стандарти бібліографії, зокрема ДСТУ 8302:2015, не існує зручного та повнофункціонального рішення, яке дозволяє отримати коректне оформлене посилання на базі даних, які зберігаються в менеджері бібліографії. Існуючі рішення на базі CSL не забезпечують коректної інтернаціоналізації та гнучкості, отже застосовуючи їх не можливо згенерувати коректне цитування за правилами стилю ДСТУ 8302:2015. Метою роботи є створення програмної технології для автоматичної генерації переліку літературних джерел відповідно до вимог ДСТУ 8302:2015 у середовищі сучасної системи комп'ютерної верстки TYPST, використовуючи як вхідні дані файли форматів BibTeX/BibLaTeX. У роботі розроблено алгоритм парсингу та обробки бібліографічних записів, який включає: лексичний аналіз вхідних даних, евристичне визначення мови джерела, конвертацію LaTeX-макросів та спецсимволів у Unicode, декодування URL-адрес, валідацію полів за типами джерел. Для фінального формування бібліографічного опису застосовано підхід на основі токенів (контент та розділювачі) із системою пріоритетів для вирішення конфліктів пунктуації. Реалізовано програмний модуль мовою TYPST, який не вимагає зовнішніх препроцесорів й враховує неоднозначності та поширені порушення стандартів при оформленні даних в форматі BibTeX. Верифікація роботи парсера проведена на вибірці з понад 500 джерел. Результат роботи дозволяє українським дослідникам, які бажають застосовувати систему TYPST, безпроблемне інтегрувати з нею бібліографічні менеджери, які здатні експортувати інформацію в форматі BibTeX/BibLaTeX.

Ключові слова: ДСТУ 8302:2015; система комп'ютерної верстки TYPST; BibTeX/BibLaTeX; автоматизація бібліографії; парсинг даних; наукова документація; менеджери цитувань.

Вступ. Застосування менеджерів бібліографії загалом у світі протягом останніх 10 років стало звичною практикою для значної кількості (в деяких регіонах й більшості) науковців, викладачів й аспірантів. Стимулює застосування таких систем відсутність необхідності ручного введення бібліографічної інформації. Як правило, якщо монографія чи стаття мають DOI, то висока ймовірність, що менеджер бібліографії зможе знайти та додати необхідну інформацію в свою базу даних автоматично. Для аналізу поточної ситуації будемо застосовувати дані зі звіту [1]. Цей звіт розроблено для інвесторів й у відкритому доступі лише ключові тези, проте інформація в ньому має бути більш об'єктивна ніж абстрактні звіти університетів чи нерепрезентативні опитування. Його ключові моменти наступні:

- сукупний загальнорічний темп зростання - 7.5.%;
- у період з 2025 до 2035 рр очікується зростання ринку у два рази з 0.43 до 0.88 млрд. USD;

- за даними ЮНЕСКО в 2024 р. біля 62% академічних установ впровадили хмарні застосунки для управління бібліографією;
- сегментація ринку: академічний - 68%, корпоративний 21%, державні структури - 11%;
- усереднено за даними International Telecommunication Union 58% студентів та дослідників застосовують мобільні бібліографічні менеджери. Зростання за 5 років на 26%. Проте, ці дані стосуються переважно північної Америки та західної Європи;
- за даними U.K. Intellectual Property Office, британський менеджер бібліографії Mendeley використовується 42% користувачами в Європі на 2024 р, що підтримується партнерством з понад 1500 університетами;
- за даними U.S. National Science Foundation (NSF), американській менеджер бібліографії Zotero мав зростання в 35% відсотків активних користувачів в період з 2021 по 2024.

Менеджери бібліографії реалізують щонайменше наступні функції [2,3]:

- ведення бази даних джерел з врахуванням їх типів з можливістю пошуку та редагування, додавання матеріалів (тексту, анотації);
- імпорт посилань в базу даних в форматі BibTeX (обов'язковий стандарт) та RIS, автоматичне додавання інформації за DOI (використовуються сервіси на кшталт doi2bib) чи іншими параметрами з використанням комерційних та/або відкритих бібліографічних баз даних;
- експорт вибраних посилань в BibTeX та інші подібні формати для використання в системах комп'ютерної верстки (класично - в варіанти TeX, але зараз не тільки в них) та синхронізації даних в різних системах;
- автоматичне оформлення (генерація) бібліографічних посилань вибраних джерел з використанням зазвичай мови CSL, яка забезпечує можливість врахування вимог різних бібліографічних стандартів;
- програмний інтерфейс для взаємодії з іншим програмним забезпеченням та/або плагіни для поширених офлайн (Microsoft Word, Open/LibreOffice Writer тощо) та онлайн (Google Docs, Microsoft 365 тощо) текстових редакторів, за допомогою якого можна зробити посилання в тексті й автоматично згенерувати перелік літературних джерел, які відповідають одному з бібліографічних стандартів.

Оскільки формати та програмні застосунки розроблялись переважно в середовищі, в якому користуються латинкою й переважно англійською, то після їх поширення виникало багато проблем з адаптацією для користувачів з іншими алфавітами (наприклад, програмна реалізація BibTeX ще в 2000-х за замовченням працювала в 7-бітному кодуванні й потім перейшла на 8 бітів). Принципово ці проблеми вирішені в BibLaTeX + Babel, розроблених під час масового переходу на UTF-8. Проте BibTeX залишається загальноприйнятим стандартом для багатьох англomовних журналів, оскільки не вимагає Babel. Введення Babel для англomовного журналу створює велику кількість проблем. Babel дозволяє різні макроси, які мають можуть непередбачено застосовувати певні правила локалізації. Крім того, його правила локалізації входять в конфлікт з класичними одnobайтовими модулями для розташування переносів, сортування, стовпчикowego верстання тощо.

В цілому такі не адаптовані до інтернаціоналізації інструменти як формат BibTeX й мова бібліографічних стилів CSL попри цей недолік в цілому залишаються стандартом. Для українських користувачів це означає, що в межах цих інструментів досягнути високого рівня відповідності вимогам ДСТУ 8302:2015 складно з причини відсутності в них механізмів інтернаціоналізації та гнучкої логіки [4]. Це знижує мотивацію для українських користувачів застосовувати такі інструменти, оскільки дотримання ДСТУ 8302:2015 є вимогою до виконаних в Україні НДР, публікацій в наукових журналах, монографій, кваліфікаційних робіт.

В цій роботі розглянута задача програмної генерації переліку літературних джерел з використанням BibTeX / BibLaTeX (частково) файлу. Ця задача розв'язана для нової системи програмної комп'ютерної верстки T_{up}st, яка має всі шанси в майбутньому стати заміною варіантам систем TeX. За два роки (з 2023) T_{up}st стрімко набирає базу користувачів [5]: систему використовують в понад 3500 академічних установах й кількість зірок на GitHub перейшла поріг у 45 тис. Проте, наданий розв'язок у цілому має універсальний характер й може бути адаптованим для деяких інших пакетів.

Задачі роботи. Розробити програмну технологію для генерації переліку літературних джерел згідно вимог ДСТУ 8302:2015 для системи T_{up}st. Для цього потрібно

1. Проаналізувати недоліки та обмеження наявних інструментів.
2. Запропонувати зручний для користувача алгоритм для генерації переліку літературних джерел, який орієнтований на те, що користувач працює переважно з бібліографічними даними, які згенеровані програмами чи оформлені іншими людьми, при цьому застосовується не лише англійська, але й європейські мови.
3. Провести програмну реалізацію генератора переліку літературних джерел мовою T_{up}st й перевірити її на достатній вибірці згенерованих та вручну оформлених вихідних даних в форматі BibTeX.

Аналіз недоліків та обмежень наявних інструментів. *RIS*. Найпростіший формат, запропонований в 1980 саме з метою обміну даними між різними бібліографічними системами. Дані зберігаються в форматі неунікальний параметр: значення. Це неструктуроване збереження, проте воно зручне тим, що автори чітко відокремлюються один від одного й немає проблеми як їх виділити з одного поля, яке заповнюється вільно й періодично з помилками. Формат принципово мінімалістичний, хоча сучасні версії мають doi, url, language. Він містить менше інформації в порівнянні з усіма наступними форматами. Використовується лише для експорту та імпорту. Проблема кодувань розв'язується на рівні програм.

BibTeX. Являє собою формат представлення інформації для однойменного двигуна в межах LaTeX та його розширень (LuaLaTeX, XeTeX) про джерела як послідовності структурованих даних виду “унікальний параметр: значення”. Запропонований в 1985 р. доступ до структури реалізується за допомогою унікального ключа, який формує користувач. Перелік полів в структурі відрізняється в залежності від типу джерела й є фіксованим. Параметр мови джерела відсутній. Сучасні бібліографічні менеджери/сайти зазвичай експортують дані в BibTeX в кодуванні UTF-8.

В двигуні BibTeX застосовується однобайтове кодування, яке треба вказати (ko_i8-r, cp1251) чи він буде працювати 7-ми бітному режимі. Відсутність інтернаціоналізації призвела до поганої практики застосування символів LaTeX в бібліографії для передачі грецьких символів, діакритичних символів латинки тощо. Двигун застосовує спрощену мову стилів, яка зберігається в файлах з розширенням bst. Ця мова дозволяє базові операції над рядками й умовні переходи, але не має повноцінних структур даних, Unicode-обробки чи локалізації. Наприклад, неможливо реалізувати правила типу "якщо авторів більше трьох – скоротити, але при цьому врахувати кириличні ініціали". Остання стабільна версія двигуна, яка включає певні розширення формату (наприклад, url, doi) з його початкової форми випущена у 2010 р.

Двигун *BibLaTeX* з'явився в 2009 р. як результат перегляду BibTeX з активним застосуванням двигуна Babel для інтернаціоналізації в LaTeX. В порівнянні з BibTeX він має дещо несумісну номенклатуру полів, збільшує кількість ролей (не тільки автор й редактор як в BibTeX, але й перекладач, рецензент тощо) й кількість типів джерел. Підтримує мову CSL, проте логіка може бути достатньо вільно запрограмована (особливо в LuaLaTeX з мовою Lua).

Мова CSL. До CSL фактично кожний бібліографічний менеджер мав свій власний несумісний формат стилів. Перша реалізація CSL з'явилась в 2004 і в 2010 р. стала стандартною мовою стилів для оформлення бібліографічних джерел у всіх

бібліографічних системах. Кількість доступних стилів [6] мовою CSL - біля 10 тис. [7]. Ця мова орієнтувалась на бібліотекарів й передбачала можливість візуального програмування - створення формату готовими блоками з обмеженими настройками. Хоча існують стилі для ДСТУ, ГОСТ тощо, з їх застосуванням неможливо зробити відповідне до вимог посилання. Наприклад, немає можливості писати "С. 20-21" чи "Р. 20-21" залежно від мови, писати авторів до чи після заголовку за певних умов тощо. Хоча й були запропоновані розширення, які цей недолік можуть нейтралізувати, їх не реалізують щоб не зробити систему занадто складною за можливою поведінкою.

Менеджери бібліографічних джерел *Mendeley (Elseiver/Scopus)*, *EndNote*[8] (*Clavirate/Web of Science*), *Zotero* (проект з відкритим кодом). Всі ці застосунки працюють з мовою CSL, проте відрізняються в можливих джерелах отримання даних. Таким чином, згенерувати повноцінне посилання у відповідності до ДСТУ 8302:2015 з їх застосуванням не можливо.

Має свій обмежений вбудований бібліографічний менеджер також *Microsoft Word*, який теж внутрішньо працює з CSL, проте можливо обирати лише зі стандартних стилів, зміна яких користувачеві не дозволена. Для *Word* та аналогічних пакетів менеджери бібліографічних джерел пропонують плагіни. Вони різні за якістю, проте корінний недолік для нашої задачі однаковий.

LaTeX - історично набір макросів для *TeX*, який вийшов за межі цього визначення й фактично сам став мовою зі своїми бібліотеками [9–12]. Його історія починається з кінця 1970-х, коли були прийняті основні архітектурні рішення, деякі з яких стали проблемою потім. Архітектура базувалась на ідеї генерації універсальної розмітки - формат *DVI*. Цей формат максимально конкретний - кожен символ в ньому розміщений на позиції. Далі драйвер принтера мав виконати *DVI* за власною логікою. Коли *PostScript* став стандартом для принтерів звичайним шляхом став *DVI->PostScript*. Як базові шрифти прийняті *Type 1* з однобайтовим кодуванням. Коли на базі *PostScript* розробили *PDF* й принтери стали переважно *GDI*, отримання результату стало тристадійним *DVI->PostScript->PDF*. З кінця 1990-х система почала змінюватись. Спочатку під *LaTeX* стали розуміти *pdflatex* - прямий генератор *PDF* на базі однобайтних *Type 1* шрифтів. Далі в 2004 р. виник окремий проєкт з підтримкою *Unicode* й *TrueType/OpenType* шрифтів - *XeTeX*, а в 2008 р. *LuaLaTeX* - ще й додатковою підтримкою мови програмування *Lua*. Зараз *LaTeX* має велику кількість готових рішень й доволі популярним є його застосування через веб (*OverLeaf*), що дозволяє кодувати й переглядати результат без інсталяції додаткових програм.

LaTeX розповсюджений переважно в академічному середовищі. Його ключові переваги: орієнтація на друк, робота з зображеннями без втрати якості, можливість реалізовувати стандартні шаблони (для статей, звітів, робіт), в тому числі з двома стовпцями, автоматичне (хоч й не ідеально) розміщення таблиць, рисунків тощо, зручність роботи з великою кількістю формул, складно помилитись в цифровій бібліографії, оскільки пишеш й бачиш в коді унікальний текстовий ключ. Проте є й недоліки: синтаксис доволі складний й результат часто не передбачуваний, тому що потрібно застосовувати в складних документах велику кількість бібліотек, які не обов'язково добре співпрацюють (сучасний *HTML5* значно більш очевидний), вимагає компіляції в *PDF* для перегляду результату, що займає час оскільки архітектура далека від ідеальної (особливо складно працювати з багатосторінковими документами й з *Babel*, який реалізований мовою *Perl* й лише запуск якого займає декілька секунд). З цими недоліками можна змиритись, якщо задача – зверстати вже готову роботу. В процесі написання тексту *LaTeX* не зручний, оскільки вимагає постійної уваги до технічної реалізації деталей.

За останні 15 років *LaTeX* більшою мірою втратив свою унікальність в аспекті переваг. За межами друкарської справи істотно більш зручним є *Markdown* з можливістю застосовувати усі функції веббраузерів при необхідності. Двигуни

Markdown звичайно підтримують виведення формул в форматі LaTeX, є можливості ставити посилання з ручним оформленням джерел. Сучасний десктопний Word може працювати з LaTeX формулами як з вбудованими, так й через MathType, також він має опцію не знижувати якість зображень й бібліографію (власну – через ряд вбудованих стилів й через плагіни для доступу до бібліографічних менеджерів).

Typst розробляється як повноцінна заміна *LuaLaTeX*. Написаний з нуля мовою Rust він реалізує основний функціонал власноруч без зовнішніх макросів, бібліотек тощо. Синтаксис його значно більш читабельний й зрозумілий (в тому числі й синтаксис для формул). Його головне – компіляція працює ефективно: перераховується й замінюється лише частина документу, тому в редакторі *Typst* рендеринг реалізується після введення нового коду автоматично. В порівнянні з LaTeX різниця в швидкості у десятки разів. Істотно не зменшують відставання результат такі підходи як, наприклад, "draft mode" в *Overleaf*, коли з документу видаляють важкий контент (зображення тощо) щоб прискорити перегляд результату.

Typst підтримує бібліографію на базі CSL, але й дозволяє запрограмувати свою бібліографічну систему. Його функціонал й швидкість дозволяють розробити генератор переліку джерел посилання відповідно до стилю ДСТУ 8302:2015 достатньої для професійного застосування якості.

Основні вимоги наступні.

1. Підтримка основних типів джерел (книга, глава у книзі, стаття в журналі/газеті, тези конференції, інтернет-джерело) з можливістю введення прямого тексту посилання для специфічних речей.
2. Застосування всіх даних, які можуть бути збережені в форматах BibTeX/BibLaTeX для основних типів джерел. Оскільки дані вводяться вільно, то дані мають бути перевірені й використані спираючись на типові практики заповнення полів. Проте незначні дані для універсальності можуть бути видалені.
3. Повна інтернаціоналізація - джерело оформлюється його мовою. Якщо мова не вказана, то вона автоматично визначається. Підтримуються такі мови як: українська, англійська, іспанська, італійська, німецька, польська, португальська, російська, французька, чеська. Всі службові слова (сторінка, том, номер, редактор, дати) описуються визначеною мовою.
4. Скорочення міст нейтралізується. Для цього застосовується таблиця скорочень й рік. Основа - прийняті в бібліографії України до 2013 року скорочення міст, скорочення міст в CPCR різних років тощо.
5. Декодування URL адрес з ASCII формату (представлення символів за межами 7-біт як послідовність HEX-кодів, кожен з яких починається з проценту).

Запропонований алгоритм роботи для *Typst*. *Отримання бібліографічних джерел.* Ми спираємось на стандартний механізм бібліографії. Правила сортування та цитування в тексті задаються елементарним шаблоном мовою CSL. Ми застосуємо сортування за порядком цитування, цифровий індекс з можливістю об'єднання послідовних цитувань в цитування виду [2-4]. Бібліографічний опис сформуємо елементарний – номер та заголовок (його надалі приховаємо й будемо застосовувати лише для відлагодження). Бібліографічні посилання розміщуються в коді сторінки *Typst* як ключі @key. У місті для бібліографічного опису встановимо виклик функції генерації опису, яка буде приймати як аргументи ім'я поточного файлу й ім'я bib файлу.

За допомогою функції `bibliography` задається CSL шаблон, bib файл й стандартний бібліографічний опис. При цьому цитування в тексті залишаються. Далі ми будемо генерувати власний бібліографічний опис програмним чином.

Завантаження bib файлу будемо реалізовувати за допомогою бібліотеки `citegeist`. Ця бібліотека є інтерфейсом для програми-парсера BibTeX/BibLaTeX мовою Rust. Ключова її функція завантаження перевіряє синтаксис й завантажує усі дані в словник. Якщо є помилки, процедура генерації верстки зупиняється з повідомлення про помилку.

Бібліотека також робить деякі заміни, наприклад видаляє нерозривні пробіли LaTeX й певні перевірки синтаксису, алгоритм яких нами не досліджувався.

Оскільки кількість записів в bib не має бути еквівалентною кількості цитованих записів, то потрібно отримати перелік саме цитованих записів. Стандартного механізму для цього нажаль поки немає, тому ключі отримуються прямо з коду файлу з видаленими коментарями, з якого була викликана функція з застосуванням регулярного виразу `@([a-zA-Z0-9_-:/]+)(?:.[a-zA-Z0-9_-:/]+)*`. Таким чином, ми можемо отримати словник тільки з цитованими джерелами. В поточній реалізації розглядається тільки порядок вживання. Алфавітний порядок сортування джерел не реалізуємо, краще дочекатись поки вказаний недолік буде вирішений на рівні T_{ypst}.

Зменшення розміру полів записів citegeist, як і формат bib, не лімітує розмір полів. Іноді в записах зберігають непотрібні для задачі великі обсяги даних, такі як анотація, зміст чи навіть вступ до книги тощо. Ця інформація для нас зайва. Раціональна межа для різних полів принципова різна: для ISBN досить 32 символів, для назви книги – 255.

Перевірка записів. Наявність та наповненість полів перевіряється за наступним словником:

```
"book": ("title", "address|location", "year"),
"incollection": ("title", "booktitle", "address|location", "year|date"),
"article": ("author", "title", "journal|journaltitle", "year"),
"online": ("title", "url"),
"phdthesis": ("author", "title", "address|location",
"school|institution|organization", "year|date"),
"mastersthesis": //аналогічно phdthesis
"inproceedings": ("author", "title", "booktitle", "address|location|venue",
"year|date"),
"misc": ("title",)
```

Маємо 8 типів джерел на вході: книга (book), розділ книги (incollection), стаття (article), інтернет-ресурс (online), дисертація (phdthesis), магістерська робота (masterthesis), тези конференції (inproceedings), інше (misc). Запис з типом inbook прирівнюється до incollection. Записи з типами booklet, manual, proceedings прирівнюються до book. Для кожного джерела вказані обов'язкові поля, деякі вказані як альтернатива (відмінності між BibTeX й BibLaTeX). Відсутність чи незаповнення вказаних полів призводить до відмови генерації. Замість джерела виводиться помилка з ім'ям необхідного параметра. Додатково перевіряється зміст misc. Припустимо два варіанти: title="free-form" й заповнено поле note й title!="free-form" й заповнено поле url (є практика замість online використовувати misc). В першому випадку формується джерело типу mff з полем note_for_freeform, в другому - джерело типу murl з полями title й url.

Обробка записів. Далі нам потрібно реалізувати перетворення словника з записами в словник обробленими записами, якій буде містити коректну інформацію в коректних полях. Якщо поле переноситься без змін, то його назва не змінюється, в протилежному випадку додається префікс "fin_". Процедура обробки полів наступна.

- Мова (language) – якщо є таке поле, то обробляємо його зміст. Розповсюджена практика писати мову повністю, двома та трьома літерами, тому для визначення належності до потрібних мов маємо спробувати всі варіанти. Наприклад, german/de/deu/ger чи czech/cz/ces/cze. У випадку якщо поля немає, то орієнтуємось на тип алфавіту (кирилиця/латинка) й на виключні для певної мови символи. Наприклад, для української мови серед підтримуваних мов притаманна кирилиця з особливими літерами "ї", "є", "г", а для польської мови притаманна латинка з особливими літерами "ą", "ę", "ł", "ń", "ś", "ź", "ż", "ć". Якщо в змісті не кирилиця й мову встановити не вдалось, приймаємо англійську (en). Всі текстові складові локалізуються визначеною в зазначеній процедурі мовою, вказувати на це в описі кожного поля не будемо.
- Актори (fin_author/fin_editor) та їх кількість (num_authors/num_editors). Поле authors й editors може містити кілька акторів - вони розділяються "and", неповний список -

закінчується на "and others". Актори можуть бути записані в форматі "Прізвище,Ім'я" чи без коми, тоді "Ім'я Прізвище". Нам потрібно залишити лише ініціали. При цьому врахувати, що ініціали можуть бути складними, наприклад І.-А. Якщо кількість авторів перевищує 3 чи є "and others", то `num_authors=-1`. При кількості авторів 1-3 порядок запису прямий - "Прізвище І.", інакше зворотній - "І. Прізвище та ін.". Для редакторів стоїть префікс "за ред." й порядок завжди зворотній. Якщо є "and others" замінюється на "та ін.".

- Назва журналу (`fin_journal`) - записується `journal` чи `journaltitle`.
- Видання (`fin_edition`) - якщо є цифра, залишається тільки вона й записується у відмінку, наприклад "3-тє видання", інакше записуємо значення поля `edition`.
- Номер (`fin_number`) - якщо є цифра з опціональним варіантом тире чи крапкою, вона записується як "№ 2". Остання крапка видаляється.
- Номер (`fin_volume`) - аналогічно номеру записується як "Т. 3".
- Кількість сторінок (`fin_page_count`) чи їх інтервал (`fin_page_interval`). Якщо є `num_pages` з BibLaTeX, то нам зрозуміло що мова йде про кількість сторінок. Проте, зазвичай кількість сторінок й номера сторінок розміщують в `pages`, тому їх треба диференціювати (С. 10 і 10 С.). Це робиться за типом джерела. Інтервали можуть в BibTeX позначатись як "5-10", "5-10, 15-20", "5+". Цю логіку, хоча вона не зафіксована в ДСТУ ми не змінюємо. Лише замінюємо всі варіанти символів, які можуть трактуватись як тире "[—]" з опціональними пробілами між ними на запис виду "5-10" з `en-dash`.
- Видавець (`fin_publisher`) - якщо є один з варіантів душок, то беремо текст тільки в дужках, інакше – беремо повністю.
- Місце захисту (`fin_school`) - записується `school` чи `insitution`.
- URL (`fin_url`) - записується URL (з поля `url` чи якщо немає, то з поля `note` для джерел типу `misc`), який відповідає рег. виразу
`"https?:/[a-zA-Z0-9+&@#/%?~_|!:,.;()]*[a-zA-Z0-9+&@#/%?~_|()]"`
 Якщо отриманий URL містить %, то він декодується. Оскільки в `typst` немає роботи з байтами, а лише з символами, то реалізувати алгоритм `urldecode` складно, а стандартної функції немає. Цілком достатньо зробити заміну в межах алфавітів цільових мов та ще грецької (стародавньої й сучасної). Наприклад "%C3%83" → "Ã", "%C3%A3" → "ã", "%C3%84" → "Ä", "%C3%A4" → "ä", "%D0%84" → "€", "%D1%94" → "ё". Це порядку 200 замін на пару закодованих символів. Для того, щоб операція не вимірювалась в мілісекундах чи їх десятках, ми можемо зробити масову заміну одним викликом. Функції як в `python translate/maketrans` немає, проте ми можемо зменшити кількість надлишкових викликів (`overhead`) шляхом формування одного регулярного виразу заміни, який шукає за виразом типу `(%D0%90|D0%B0|. |%20)` й замінює кожний знайдений підрядок на підрядок від лямбда-функції, яка за ключем (підрядком) визначає потрібне значення. В результаті отримуємо оптимальний однопрохідний алгоритм.
- Заголовок (`fin_title`) - в полі `title`, звичайно статей зустрічаються макроси LaTeX для виведення математичних символів, це поширена практика в статтях з математики й фізики. На відміну від URL таких символів не буде дуже багато в рядку. Проте дуже багато потенційних макросів. Припустивши, що навряд буде дріб, матриця чи система був виділений раціональний перелік: грецькі літери `"\alpha": "\alpha"`, оператори та символи `"\times": "\times"`, стрілки `"\to": "\rightarrow"`, логіка та множини `"\emptyset": "\emptyset"`. Ці макроси можуть бути в блоках внутрішньої математики `\beta\beta` чи `(\alpha \cdot \beta)`. Якщо зустрічаються в рядку "\$" чи "\$" з використанням регулярних виразів для визначення блоків
`"\${^$}+\$|\\\\\\\\([^\$])+\\\\\\\\")`
 та макросів в них
`"\\\\\\\\[a-zA-Z]+|\\\\\\\\[,;:]")`

циклічно перебираємо всі блоки, в кожному виділяємо макроси й якщо такий ключ є в словнику, то робимо заміну. Ідея заміни аналогічна URL, проте заміна робиться як `latex_mapping.at(mac.text, default: mac.text)`, тобто якщо ключ не знайдений, то залишається макрос як є без помилки.

- `fin_day`, `fin_month`, `fin_year` - якщо `day`, `month`, `year` присутні, то базуємось на їх значеннях. Часто `month` записується як три літерне скорочення місяця - `jan`, `feb`, тоді за першими трьома літерами треба їх перевести в число. Якщо ж даних немає, але вони є в `date` в форматі рік-місяць-день, то треба їх виділити й перенести. Зустрічається ще нестандартний варіанти, коли в `date` тільки рік, тоді беремо його як ціле число. Відсутній чи некоректний рік для всіх джерел за винятком `online` та `misc` призводить до відмови від парсингу джерела. Розглядається випадок, коли рік невідомий (знак питання наприкінці) чи є декілька варіантів (роки записані через `or`). Такі роки записуються в квадратних дужках.
- Дата доступу (`fin_urldate`) - зазвичай записується в форматі рік-місяць-день й якщо так, то переводиться в день.місяць.рік, інакше – залишаємо як є.
- Примітка `fin_note` - допускається застосування елементарних HTML тегів: `<b>`, `<i>`, `<sup>`, `<sub>`. Для цього реалізується простий парсер з підтримкою вкладень. Знайдені блоки оформлюються за допомогою команд `Typst strong`, `emph`, `sub`, `super`.

Генерація бібліографічного опису. Проблема генерації полягає в тому, що стиль передбачає застосування різних розділювачів для різного типу інформації. При чому частина інформації може бути відсутньою, що призводить до необхідності виключати певні розділювачі. Пряма послідовна генерація має бути або істотно зв'язаною (дивитись наперед чи коректувати вже згенероване) або буде призводити до артефактів, які наприкінці потрібно грубо видаляти.

Для того, щоб уникнути двох крайнощів реалізуємо лексичний аналізатор з пріоритетним вирішенням конфліктів. Для цього створимо масив незалежних об'єктів (токенів) двох типів: контент й розділювач. Фінальну генерацію будемо проводити за масивом з застосуванням правил фільтрації та пріоритетів. Кожен тип розділювача має свій пріоритет:

```
#let SEP_SPACE = (text: " ", weight: 10)
#let SEP_DOT = (text: ". ", weight: 20)
#let SEP_COMMA = (text: ", ", weight: 30)
#let SEP_COLON = (text: " : ", weight: 40)
#let SEP_SEMICOLON = ( text: "; ", weight: 50)
#let SEP_SLASH = (text: " / ", weight: 100)
#let SEP_DBL_SLASH = (text: " // ", weight: 110)
```

Вирішення конфліктів реалізується за наступними правилами:

- якщо розділювачі йдуть один за одним, то залишається той, в якого більший пріоритет;
- якщо пріоритет однаковий, залишається один розділювач (всі мають різні пріоритети, тому проблем не має).

Перед генерацією програма перевіряє для надійності контент на пустоту (`none`, `""`, `[]`, `styled(child:[])`). Пусті токени виділяються. Це запобігає утворення подвійних крапок (..) чи "вісячих" ком.

Заповнення токенів й розділювачів будемо реалізовувати за допомогою послідовних операцій, перелік яких задається словником:

```
#let formats = (
  "book": ("authors1", "title", "authors2", "edition", "editors",
    "publisher_address", "year", "volume", "pages", "isbn", "doi", "url"),
  "incollection": ("authors1", "title", "authors2", "booktitle", "edition",
    "editors", "publisher_address", "year", "volume", "pages", "isbn", "doi",
    "url"),
  "article": ("authors1", "title", "authors2", "journal", "year",
    "volume_num", "day_month", "pages", "doi", "url"),
```

```

"inproceedings": ("authors1", "title", "authors2", "booktitle", "editors",
"publisher_address", "year", "volume", "pages", "isbn", "doi", "url"),
"phdthesis": ("authors1", "title", "thesis_type", "school_or_org", "year",
"pages", "url"),
"mastersthesis": ("authors1", "title", "thesis_type", "school_or_org",
"year", "pages", "url"),
"online": ("authors1", "title", "authors2", "site_name", "url"),
"murl": ("title", "url"),
"mff": ("note_for_freeform",),
)

```

Опишемо кожну операцію окремо.

- **authors1**: якщо `num_authors` від 1 до 3, то додати об'єкт з `fin_author` та розділювач `SEP_SPACE`
- **authors2**: якщо `num_authors > 3` чи `-1`, то додати розділювач `SEP_SLASH`, об'єкт з поля `fin_author` та розділювач `SEP_DOT`
- **booktitle**: якщо є, додати з відповідного поля, видаливши останню крапку й додати розділювач `SEP_DOT`
- **day_month**: якщо немає `fin_number` і `fin_volume`, але є місяць і опціонально день, то записати їх з назвою місяця згідно правил й додати як об'єкт, також додати розділювач `SEP_DOT`
- **doi**: якщо є, додати значення поля як об'єкт з префіксом `"DOI:"` й додати розділювач `SEP_DOT`
- **edition**: якщо є, додати як об'єкт, видаливши останню крапку й додати розділювач `SEP_DOT`
- **editors**: якщо `num_editors > 0`, то додати розділювач `SEP_SLASH`, об'єкт з поля `fin_editor` та розділювач `SEP_DOT`
- **isbn**: якщо є, додати значення поля як об'єкт, потім додати розділювач `SEP_DOT`
- **journal**: якщо є, додати значення поля як об'єкт, потім додати розділювач `SEP_DOT`
- **note_for_freeform**: передати як об'єкт в поле запису `fin_note` як є
- **pages**: записуємо кількість чи інтервал сторінок з `fin_page_count / fin_page_interval`
- **publisher_address**: обов'язково додається адреса як об'єкт. Якщо видавництво є, то додається воно як об'єкт після розділювача `SEP_COLON`. Оскільки далі йде обов'язково кома з роком, то розділювач після не потрібний.
- **school_or_org**: обов'язково додається адреса як об'єкт. Якщо видавництво також є `school` чи `organization`, то додається значення наявного поля як об'єкт після розділювача `SEP_COLON`
- **site_name**: якщо наявна `organization` чи заповнене поле `fin_journal` – додається перша першого з наявних полів як об'єкт, далі додається розділювач `SEP_DOT`
- **thesis_type**: береться значення з `type` чи `fin_note` й видаляється остання крапка за наявності. Якщо значення є, то додаємо `SEP_COLON`, потім об'єкт зі значенням, потім `SEP_DOT`
- **title**: додається значення поля `fin_title` як об'єкт. Якщо наприкінці `"?"` чи `"!"`, то вставляється розділювач `SEP_SPACE`, інакше - `SEP_DOT`
- **url**: вставляється значення об'єкту з поля лише, якщо немає `doi`. Вставляється URL як об'єкт. Якщо є дата, то вона також вставляється як об'єкт з `urldate` після оформлення. Закінчуємо розділювачем `SEP_DOT`
- **volume**: вставляється якщо є `fin_volume` як об'єкт з розділювачем `SEP_DOT`
- **volume_num**: якщо є тільки значення поля `fin_volume` чи `fin_number`, то віно вставляється як об'єкт з розділювачем `SEP_DOT`. Якщо є обидва значення, то між ними додається розділювач `SEP_COMMA`
- **year**: вставляється розділювач `SEP_COMMA`, об'єкт зі значенням поля `fin_year` та розділювач `SEP_DOT`.

Виведення. Після завершення операцій з кожним записом згенерований текст вставляється у абзац (`par`) з відповідним номером з фіксованою відстанню після крапки.

Перевірка результатів. Для перевірки були взяті три джерела:

- приклад коректного оформлення джерел оформлення з сайту Grafiati [13], схвалений Книжковою палатою України;
- перелік для перевірки парсеру для верстки дисертації в LaTeX з [14] (включає 13 записів з прикладами еталонного оформлення різних типів джерел [15]);
- експортована власна база даних з програми Zotero (~400 джерел).

Всього - понад 500 джерел.

Зі списком Grafiati є такі відмінності:

- "2-ге видання, стер". та "3-тє видання, випр. і доп." замінюється на "2-ге видання". та "3-тє видання". Подібна деталізація на наш погляд не потрібна;
- не відрізняємо "за ред.", "ред.", "за заг. ред" тощо - тільки перший варіант;
- тип укладача не реалізуємо, лише автор чи редактор;
- назва організації в книзі не пишеться, як й назва тома.
- не вказуємо дату звернення для DOI, але пишемо DOI, а не URL.
- в англ. варіанті URL не "date of access", а "accessed".
- в тезах конференції не вказується редколегія, місце зустрічі тощо, тільки рік видання й видавець.
- посилання з типом джерела "диск", "закон", "нормативний акт", "препринт", "патент", "стандарт", "архівні матеріали" може бути реалізовано вручну через misc з free-title - в BibTeX таких джерел немає, в BibLaTeX частково є, але поля складно зіставити.

У цілому вони, на наш погляд не критичні й в цілому результат генератора можна вважати відповідним до стандарту наскільки це можливо з BibTeX/BibLaTeX файлів. Приклад генерації показаний на рис. 1.

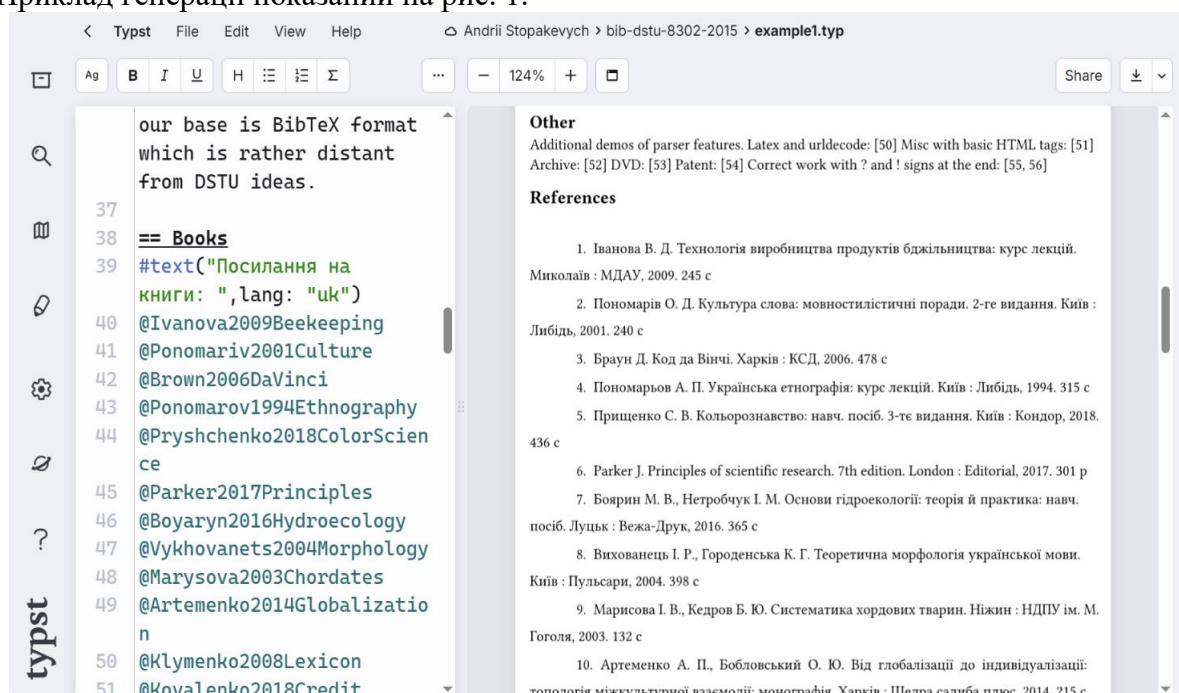


Рис.1. Генерація бібліографії з власноруч написаного BibTeX файлу для відтворення оформлення з сайту Grafiati

Зі списку для перевірки парсеру не генеруються такі посилання:

- тип thesis – не підтримується й стандартними парсерами (треба phdthesis, mastersthesis);
- типи unpublished, techreport, patent не підтримуються, misc - лише для ручної цитати чи як BibTeX варіант online з BibLaTeX;
- якщо немає адрес в певних типах джерел (вимога ДСТУ, можна якщо не можливо вручну поставити).

В експортованому списку додаткових до вказаних вище артефактів не виявлено. Таки чином, можемо сказати, що результат цілком задовільний. Вихідний код доступний за адресою <https://typst.app/project/rqwMZQBnT2VFdkbaMgz87y>. Для перегляду файлів - відкрийте зліва список за допомогою кнопки "Explore files". Генерація за другим й третім списком відобразиться шляхом перемикання "ока" на файли `example2.typ` й `example3.typ`.

Висновки. Розглянута задача програмної генерації переліку літературних джерел з використанням BibTeX / BibLaTeX (частково) файлу. Ця задача розв'язана для нової системи програмної комп'ютерної верстки Typst. Продемонстровані недоліки та обмеження існуючих інструментів, показано що застосування менеджерів бібліографії для користувачів з нелатинськими алфавітами має багато проблем. Обґрунтовані переваги Typst як альтернативи LuaLaTeX. Запропоновано для користувачів менеджерів бібліографічний алгоритм. Оскільки користувачі сучасних менеджерів переважно не самі створюють посилання в форматі BibTeX, а завантажують їх з різних сайтів з метою імпорту їх в свій менеджер бібліографії, алгоритм намагається виправити можливі недоліки вихідної інформації й дозволяє реалізувати посилання з власним оформленням джерела. Проведена програмна реалізація генератора переліку літературних джерел мовою Typst, яка перевірена на вибірці у понад 500 джерел. Результати є цілком задовільними для використання в практичній роботі. Покликання на вихідний код надано.

Список літератури

1. Reference Management Tools Market Size, Growth, Size, Share, and Industry Analysis. Regional Forecast To 2035. Last updated 10 nov 2025. URL: <https://www.businessresearchinsights.com/market-reports/reference-management-tools-market-101541>
2. Mendes K. D. S., Silveira R. C., Galvão C. M. Use of the bibliographic reference manager in the selection of primary studies in integrative reviews. *Texto & Contexto Enfermagem*. 2019. V. 28. DOI:10.1590/1980-265x-tce-2017-0204
3. Studies and analysis of reference management software: A literature review. *El Profesional de La Información*. 2015. No.24(5). P. 680. DOI:10.3145/epi.2015.sep.17
4. Попов Р. О., Карпенко Н. В. Проблеми використання системи LaTeX та BibTeX в українському науковому середовищі. Матеріали XVII міжнародної науково-практичної конференції «Інформаційні технології і автоматизація». Одеса: ОНТУ, 2024. С. 757-761
5. Two years and counting: How we are building the future of technical writing. URL: <https://typst.app/blog/2025/future/>
6. Боженко О., Корян Ю., Федорець М. Міжнародні стилі цитування та посилання в наукових роботах. Київ : УБА, 2016. 118 с.
7. Official repository for Citation Style Language (CSL) citation styles. URL: <https://github.com/citation-style-language/styles>
8. Баскакова С.О., Вигівська В.О. Бібліографічний менеджер EndNote для науковців : (із досвіду роботи бібліотеки Криворізького національного університету). Матеріали науково-практичної інтернет-конференції «Сучасна бібліотека: проблеми, досвід та вектори розвитку». Харків, 2019. С. 9–13.
9. Грищенко Т.Б., Нікітенко О.М., Дейнеко Ж.В. Створення електронних підручників засобами видавничої системи LaTeX. Поліграфічні, мультимедійні та web-технології: колективна монографія. Харків: Мадрид, 2021. С. 80-96
10. Азаренков В.І., Федоріщева В.О. Дослідження можливостей автоматизації розробки шаблонів документів всередовищі LATEX. *Системи управління, навігації та зв'язку*. 2021. Т.1. №63. С.71-73
11. Подошвелев Ю. Програмування власного класу в системі LaTeX. Збірник наукових праць викладачів, аспірантів, магістрантів і студентів фізико-математичного факультету ПНПУ імені В. Г. Короленка. Полтава : Астроя, 2022. С. 112-114

12. Baranovskyi O. LaTeX classes for doctoral theses in Ukraine: Interesting tips and painful problem. *The 43rd Annual Conference of the TeX Users Group*. 2022. URL: <https://www.imath.kiev.ua/~baranovskyi/talks/20220724tug2022.pdf>
13. Приклади оформлення посилань за ДСТУ 8302:2015 у списку використаних джерел. URL: <https://www.grafiati.com/uk/info/dstu-8302-2015/examples/>
14. Phd-LaTeX-Disseration-Template. URL: <https://github.com/AndreyAkinshin/>
15. BibTeX Style Examples. URL: <https://verbosus.com/bibtex-style-examples.html>

DEVELOPMENT OF A PARSER FOR THE INTEGRATION OF BIBLIOGRAPHY MANAGERS AND THE TYPST PROGRAMMED COMPUTER LAYOUT SYSTEM THAT MEETS THE REQUIREMENTS OF THE BIBLIOGRAPHIC STANDARD DSTU 8302:2015

A.O. Stopakevych¹, O.A. Stopakevych²

¹State University of Intellectual Technologies and Telecommunications

1, Kuznechna Str., Odesa, 65023, Ukraine

²National Odesa Polytechnic University

1, Shevchenko Ave., Odesa, 65044, Ukraine

Email: stopakevich@gmail.com

Bibliography managers (e.g. Zotero, Mendeley, EndNote) have become an essential instrument for numerous researchers. The functionality of such systems extends beyond mere information structuring, encompassing the capacity to import bibliographic details from publishers and journals. This process utilizes the unique identifiers provided by CrossRef, facilitating seamless integration into the management database. The use of a single identifier, whether it be a DOI or ISBN, enables the seamless synchronization of bibliographic data, streamlining the process of bibliographic management. However, researchers who use national bibliography standards, in particular DSTU 8302:2015, face a lack of a convenient and fully functional solution that allows them to obtain correctly formatted references based on the database stored in the bibliography manager. Existing CSL-based solutions have been observed to lack the capacity to provide accurate internationalization and flexibility. Consequently, it is not possible to generate correct citations according to the rules of DSTU 8302:2015 when using them. The objective of this research is to develop software technology capable of automatically generating a list of bibliographic references in compliance with the requirements of DSTU 8302:2015. This will be achieved within the framework of a contemporary computer typesetting system, Typst, utilizing BibTeX/BibLaTeX format files as the source of input data. The work presents the development of an algorithm for parsing and processing bibliographic entries, encompassing lexical analysis of input data, heuristic determination of the source language, conversion of LaTeX macros and special characters to Unicode, decoding of URLs, and validation of fields by source type. In the final formation of the bibliographic description, a token-based approach (content and separators) is employed, in conjunction with a priority system for resolving punctuation conflicts. A software module has been implemented in the Typst language that functions independently of external preprocessors. It is notable for its ability to address ambiguities and common violations of standards when formatting data in BibTeX format. The parser has been verified on a sample of over 500 BibTeX entities from diverse sources. The result allows Ukrainian researchers who wish to use the Typst system to seamlessly integrate bibliographic managers that can export information in BibTeX/BibLaTeX format.

Keywords: DSTU 8302:2015; Typst computer typesetting system; BibTeX/BibLaTeX; bibliography automation; data parsing; scientific documentation; citation managers.