1. Rabin

1.1. Опишіть вашу предметну область словами

Предметна область «Магазин алкогольних напоїв».

Об'єкти:

1. Алкогольний напій — ідентифікатор, тип, назва, об'єм, ціна.
2. Магазин — назва, список-асортимент.

Запити:

1. Знайти напої в магазині, дешевші за задану ціну.
2. Знайти напої в магазині, дорожчі за задану ціну.
3. Знайти напої в магазині, більші за заданий об'єм.
4. Купити напій.

1.2. Запропонуйте подання [частини] даних Вашої задачі з використанням списків Прологу

Представимо асортимент магазину у вигляді списку. Для цього спочатку опишемо можливі напої без використання списків:

```
domains
    bev_id, store_id, bev_type = symbol
    bev_name = string
    bev_volume, bev_price = real

clauses
    % напій(ID напою, ім'я, тривалість виконання (с)).
    % ID напою = jd05 — (J)ack (D)aniels, (0.5) L
    beverage(jd05, whiskey, "Jack Daniels", 0.5, 100.0).
    beverage(jb05, whiskey, "Jim Beam", 0.5, 100.0).
    beverage(bl05, liqueur, "Bailey's", 0.5, 75.0).
    beverage(pl05, beer, "Paulaner", 0.5, 25.0).
```

Тепер представимо асортимент магазину у вигляді списків:

```
domains
    % ... Типи з попереднього завдання
```

```
    stocklist = symbol*

clauses
    % ... Факти і правила з попереднього завдання
    % магазин(ID магазину, [асортимент магазину за ID]).
    store(s0, [jd05, jb05, bl05, pl05]).
    store(s1, [jb05, jd05, pl05]).
    store(s2, [jb05, bl05, pl05]).
```

1.3. Запропонуйте предикати для розв'язання одного з запитів Вашої задачі з використанням списків Прологу.

Реалізуємо запит для пошуку напоїв в магазині, дешевших за задану ціну:

```
% Рекурсивна перевірка, чи є напій в асортименті
% bev_in_stocklist(напій, наявність)
bev_in_stocklist(Bev, [Bev | _]).
bev_in_stocklist(Bev, [_ | T]) :-
    bev_in_stocklist(Bev, T).

% Перевірка, чи є напій `BevID` у магазині `StoreID`
store_has_bev(StoreID, BevID) :-
    store(StoreID, Stocklist),
    bev_in_stocklist(BevID, Stocklist).

% Пошук напоїв у магазині, які коштують дорожче за Price
% Запит, щоб знайти напої в магазині s0, дешевші за 35:
% store_bev_under(s0, Bev, 35).
store_bev_under(StoreID, BevID, Price) :-
    % Знайти асортимент `Stocklist` магазину `StoreID`
    store(StoreID, Stocklist),
    % Знайти напій `BevID` в асортименті
    bev_in_stocklist(BevID, Stocklist),
    % Знайти ціну напою
    beverage(BevID, _, _, _, BevPrice),
    % Істина, якщо ціна менше заданої
    BevPrice < Price.
```

1.4. Запропонуйте подання [частини] даних Вашої задачі з використанням динамічних баз даних Прологу.

Представимо магазини у вигляді динамічної бази даних:

```
% Створюємо динамічну базу даних `stores` –
% список існуючих магазинів, де визначений предикат `store`
database - stores
    store(store_id, stocklist)
```

Тепер оголошений динамічний предикат `store/2`, який використовується так само, але зберігається в динамічний базі даних `stores`.

1.5. Запропонуйте предикат(и) для розв'язання одного з запитів Вашої задачі з використанням динамічних баз даних Прологу

Оголосимо предикат, який симулює покупку напою в магазині:

```
predicates
    % ... Предикати для напоїв з попередніх завдань
    % Предикат `store/2` відсутній у цій секції,
    % оскільки він описаний у секції динамічних баз даних
    buy_bev(bev_id, stocklist, stocklist)
    buy_store_bev(store_id, bev_id)

clauses
    % ... Факти і правила з попередніх завдань

    % Рекурсивний предикат, щоб купити один напій Item
    % з асортименту (видалити перший елемент `Item` зі
    % списку)
    buy_bev_one(_, [], []).
    % Відсікання, щоб отримувати лише один результат
    buy_bev_one(Item, [Item | Tail], Tail) :- !.
    buy_bev_one(Item, [Head | Tail], [Head | Res]) :-
        buy_bev_one(Item, Tail, Res).

    % Придбати напій `BevID` в магазині `StoreID`
    % Якщо напій придбали, він видаляється з асортименту
```

```
    buy_store_bev(StoreID, BevID) :-
        store(StoreID, Stocklist),
        buy_bev_one(BevID, Stocklist, NewStocklist),
        retract(
               store(StoreID, _)
        ),
        % Додати магазин `StoreID` з асортиментом `Stocklist`
        % в кінець динамічної БД `stores`
        assertz(
               store(StoreID, NewStocklist)
        ).
```

1.6. Запропонуйте опис [частини] даних Вашої задачі з використанням засобів мови Ліспо.

Представимо відомості про магазин мовою Ліспо. Мовою Пролог вони описані так:

```
store(s0, [jd05, jb05, bl05, pl05]).
store(s1, [jb05, jd05, pl05]).
store(s2, [jb05, bl05, pl05]).
```

Тоді мовою Lisp створюємо ідентичні змінні `s0`, `s1`, `s2` і присвоїмо їм відповідний зміст за допомогою виразу (expression) `let`:

```
(let ((s0 '(jd05 jb05 bl05 pl05))
      (s1 '(jb05 jd05 pl05))
      (s2 '(jb05 bl05 pl05)))
)
```