

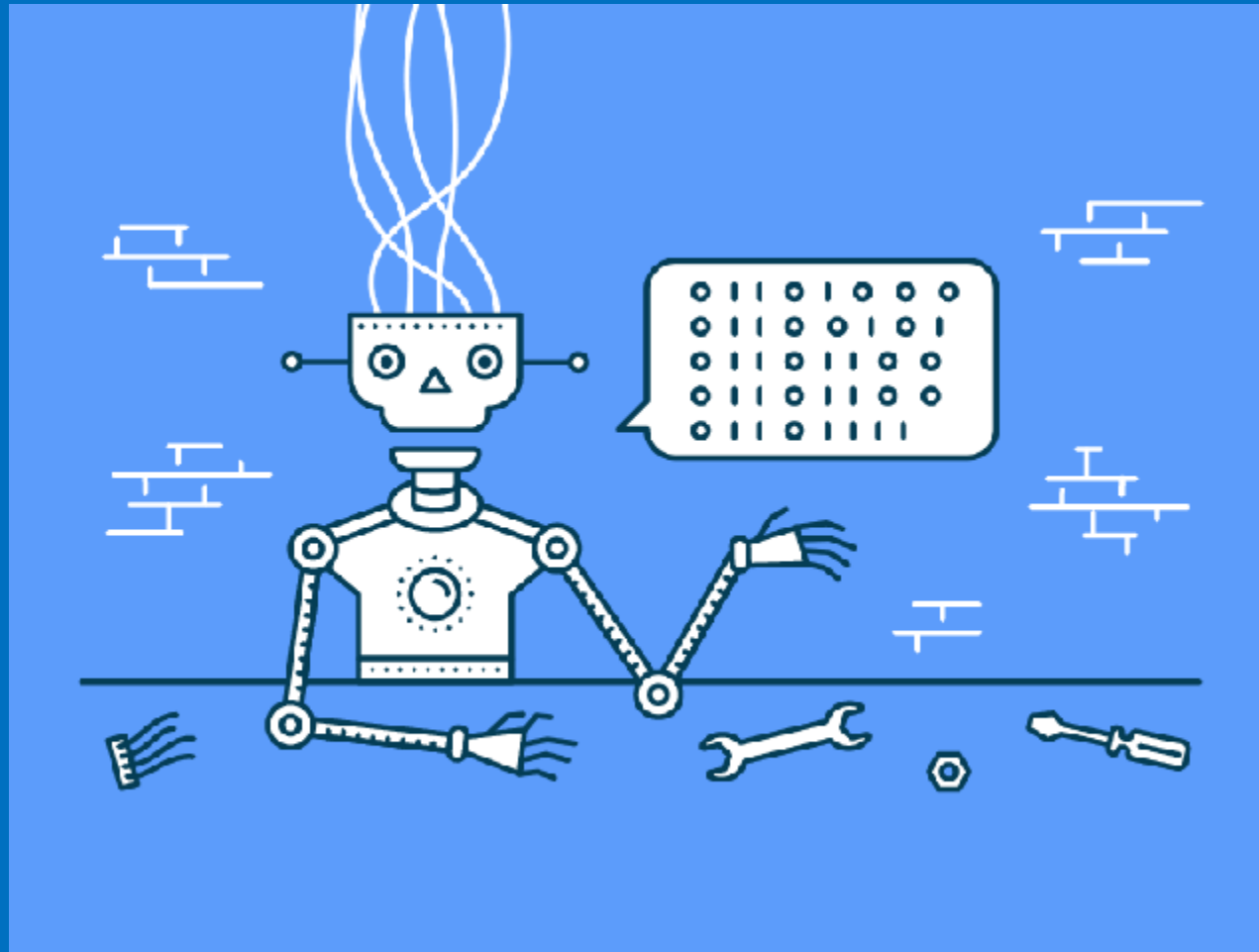
Funkcionális programozás C++-ban

Lazy – a lusta
kiértékelés

Lazy – a lusta kiértékelés

- ▶ bizonyos számítások sok időt vehetnek igénybe
 - ▶ A, B mátrix
 - ```
auto R = A * B; // conditionally use R
```
    - ▶ az eredmény azonnal kiszámításra kerül
    - ▶ jobb lenne, ha csak akkor, amikor tényleg szükséges, pl.:
      - ```
auto P = [A,B] { return A * B; };
```
- ▶ ha P-t többször hívjuk meg, akkor többször fog a költséges művelet elvégződni
 - ▶ lusta kiértékelés, amely csak az első hivatkozásnál értékelődik ki, viszont utána megtartja az értéket.

ex_0: move ctor



Dinamikus programozás

Dinamikus programozás

- ▶ egy olyan technika, amellyel úgy oldunk meg összetettebb problémákat, hogy kisebb problémákra bontjuk őket
- ▶ amikor egy részprobléma megoldódik, akkor annak eredményét eltároljuk, hogy később felhasználjuk az összetett probléma megoldására
- ▶ számos valós való életben használt algoritmusnál alkalmazzák
 - ▶ legrövidebb út keresés
 - ▶ stringek közti távolság

Levenshtein távolság

- ▶ Egy metrika, amely megadja, hogy két string milyen távol van egymástól
 - ▷ szovjet matematikus után Vladimir Levenshtein (1965)
- ▶ Informálisan: az egyszeres szerkesztések száma (törlés, beszúrás, csere), amellyel a két string egymásba transzformálható
- ▶ Definíció:
 - ▷ a, b sztringek, $|a|$ és $|b|$ hosszúsággal

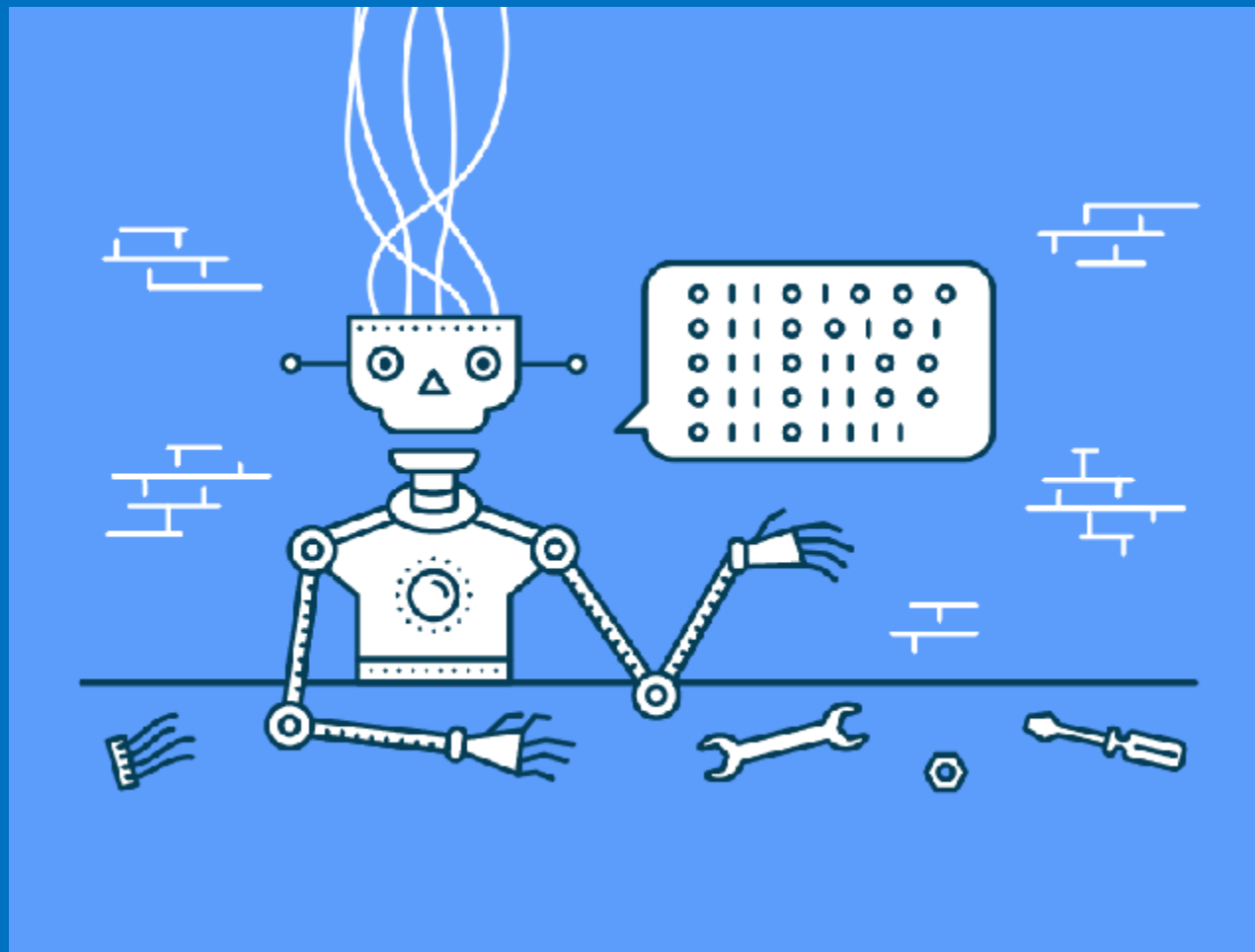
$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} \end{cases}$$

- ▷ ahol, $1_{a_i \neq b_j}$ indikátor, 1 értékű, ha $a_i \neq b_j$, 0 egyébként



if $\min(i, j) = 0$,
otherwise.

ex_1: Levenshtein

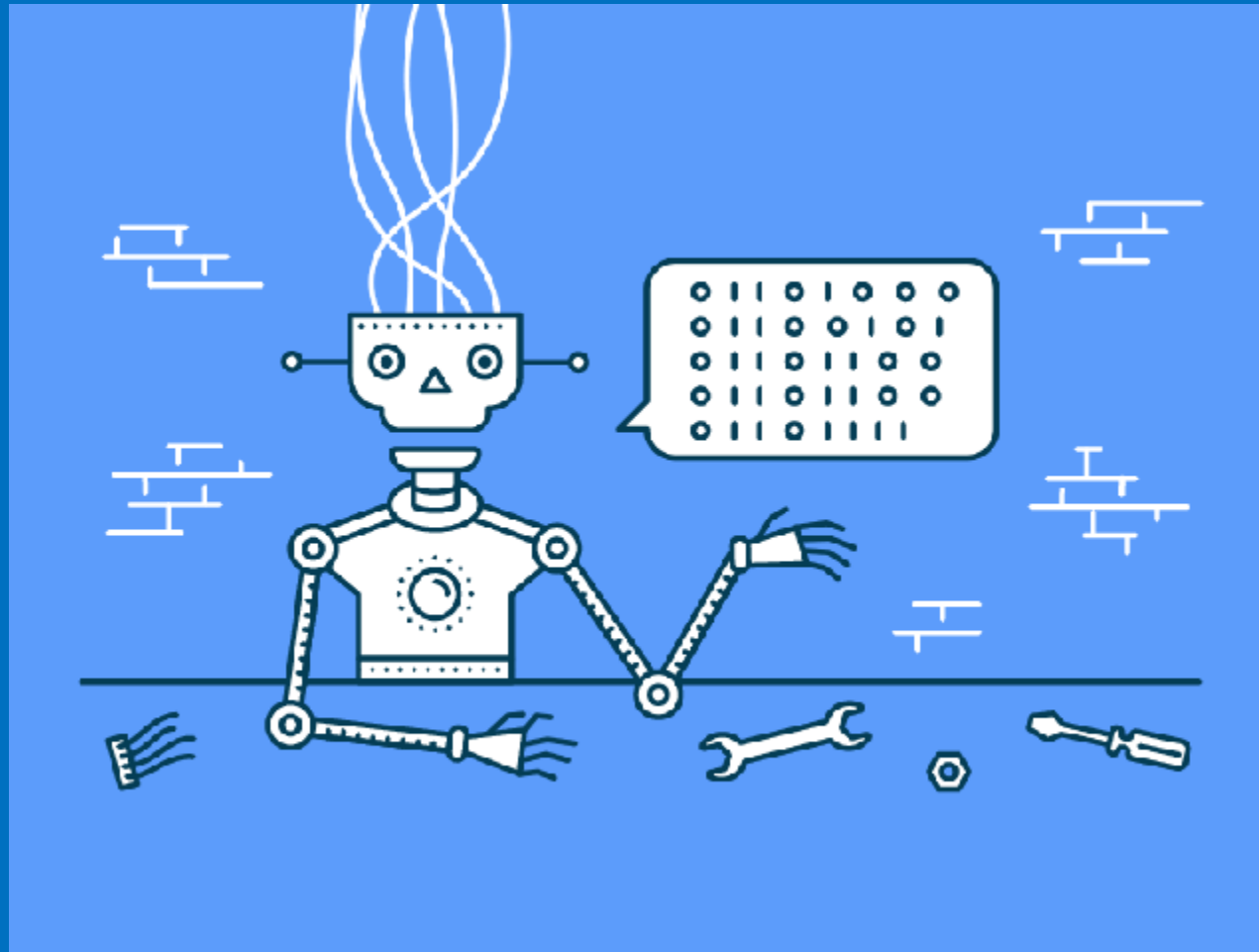


Általános
memoizáció

Általános memoizáció

- ▶ a Levenshtein távolságnál használt módszer általánosítható
- ▶ tiszta függvények használata (referenciálisan transzparens függvények)
 - ▶ minden tiszta függvény egy leképezés (*mapping*), ezért természetesen valamilyen asszociatív tárolót fogunk alkalmazni
 - ▶ rekurzív implementációnál a memoizált változatot kell használni!

ex_3: generikus memoizáció



Köszönöm a figyelmet!