

2 settembre 2021

In un sistema concorrente sono presenti due gruppi di thread, detti rispettivamente produttori e consumatori, che si appoggiano ad una struttura dati condivisa per passarsi dati generici di tipo `T`. La struttura dati è *thread-safe* ed implementa il concetto di buffer circolare: al suo interno ospita un array di `N` elementi (con `N` specificato a livello di tipo) nel quale vengono depositati i valori ricevuti dai singoli produttori, in attesa che vengano passati ad un consumatore qualunque che ne faccia richiesta. I dati sono trattati in modalità FIFO (First-In-First-Out).

- **`pub fn insert(&self, t: T) {...}`** → Inserisce un elemento. Se il buffer risulta pieno nel momento in cui un produttore prova ad inserire un nuovo valore, l'operazione si blocca, senza consumare CPU, in attesa che si crei uno spazio a seguito della conclusione di un'operazione di lettura da parte di un consumatore.
- **`pub fn extract(&self) -> T {...}`** → Estrae un elemento. Analogamente, se un consumatore prova a leggere un valore mentre il buffer è vuoto, deve attendere, senza consumare CPU, che un produttore inserisca un nuovo valore.

Per generalità, accanto alle operazioni base di inserimento ed estrazione di un valore, la struttura dati offre anche una coppia di operazioni con attesa limitata temporalmente

- **`pub fn try_insert_for(&self, t: T, d: Duration) -> TimeoutResult {...}`** → Cerca di inserire un elemento, se non riesce entro l'intervallo `duration`, restituisce `false`; in caso di successo, restituisce `true`.
- **`pub fn try_extract_for(&self, d: Duration) -> (Option<T>, TimeoutResult) {...}`** → Cerca di estrarre un elemento, se non riesce entro l'intervallo `duration` restituisce un `Option` vuoto; altrimenti restituisce un `Option` con l'elemento estratto.

Si implementi la classe generica di seguito proposta utilizzando le funzionalità offerte dal linguaggio **Rust**, aggiungendo le parti eventualmente necessarie per ottenere il comportamento descritto.

```
struct CircularBuffer<T: Send + 'static> {...}

impl<T: Send + 'static> CircularBuffer<T> {...}
```