

# Introductie tot R7RS

## Algoritmen en Datastructuren 1

9 November 2023

**Sam Van den Vonder**  
[sam.van.den.vonder@vub.be](mailto:sam.van.den.vonder@vub.be)

**Youri Coppens**  
[youri.coppens@vub.be](mailto:youri.coppens@vub.be)

In deze cursus gebruiken we de R7RS-variant van Scheme en de DrRacket ontwikkelomgeving. We leggen enkele praktische aspecten uit om R7RS te gebruiken.

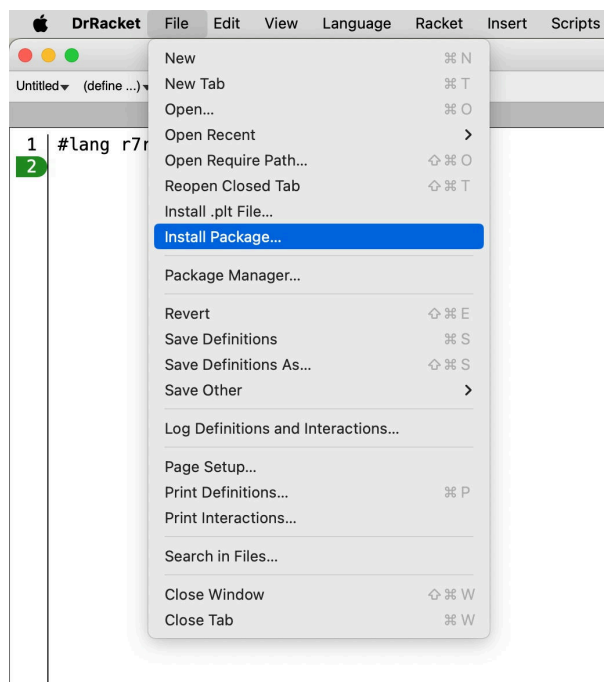
Je bent vrij om een andere code editor te gebruiken, maar daarvoor geven de assistenten geen officiële handleiding/instructies.

## 1 R7RS in DrRacket

DrRacket is een editor waarin men verschillende programmeertalen kan gebruiken, bijvoorbeeld R5RS en “Racket” (een andere variant van Scheme). DrRacket heeft geen ingebouwde ondersteuning voor R7RS, en om R7RS te gebruiken moet je eerst een “package” (een soort van plugin of library) installeren. Je kan deze package installeren via de menu’s van DrRacket.

### 1.1 R7RS installeren in DrRacket

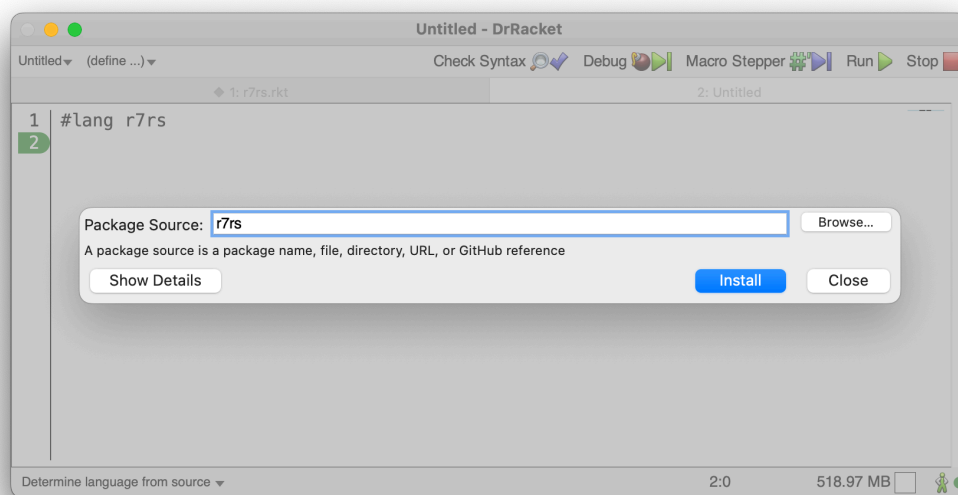
In de menubalk van DrRacket vind je de optie “File -> Install Package...” zoals getoond in Figuur 1. Klik hierop, en er zal een dialoog verschijnen zoals getoond in Figuur 2.



Figuur 1: De “File” dialoog in de menubalk van DrRacket. De optie “Install Package” is gehighlight.

Bovenaan in de geopende dialoog zie je een tekstveld Package source: <tekstveld>. Schrijf “r7rs” in het tekstveld, en druk op de “install” knop.<sup>1</sup> De r7rs package zal nu installeren. De package is klaar met installeren wanneer je onderaan niet meer kan klikken op de “Abort install” knop, en wel op de “Close output” knop. Je mag nu het scherm van de package manager gewoon sluiten.

<sup>1</sup>Indien een package al geïnstalleerd is zie je een “update” knop in plaats van “install”.



Figuur 2: De “Install Package...” dialoog van DrRacket. “r7rs” werd ingevoerd bij “Package Source”.

## 1.2 R7RS gebruiken in DrRacket

Om R7RS code te kunnen schrijven moet je DrRacket “helpen” om te weten in welke programmeertaal je document is geschreven. Selecteer links onderaan in DrRacket “Choose Language...”, en selecteer vervolgens “Determine language from source”. Indien je dit niet ziet staan, selecteer dan “Choose Language...” en selecteer “The Racket Language”. Voortaan kan je als eerste lijn van een R7RS bestand de lijn “`#lang r7rs`” plaatsen. Hierdoor weet DrRacket dat je code in R7RS geschreven is.

## 2 Libraries in R7RS

### 2.1 Importeren van R7RS libraries

R7RS laat, in tegenstelling tot sommige andere Scheme-varianten zoals R5RS, toe om code onder te verdelen in modules die “libraries” heten. Libraries zijn een handige manier om functies te groeperen, en vormen op die manier een extra niveau van abstractie.

Alle Scheme-functies, dus ook `define`, bevinden zich in R7RS in een library. Standaard zijn er dus nauwelijks functies beschikbaar tenzij je deze expliciet importeert. Daarom zal je bovenaan je eigen code, vlak onder de lijn die de taal specificeert, meestal de `import` functie oproepen met als parameters de libraries die je in je code wil gebruiken. Een typische import bovenaan een bestand is de volgende:

```
1 (import (scheme base)
2       (scheme write))
```

R7RS

De eerste lijn geeft aan dat we de basisfuncties van Scheme willen importeren, en de tweede lijn betreft een aantal eenvoudige I/O functies, zoals bv. `display`. Een volledig overzicht van de standaard R7RS libraries is te vinden in de documentatie (zie Hoofdstuk 4). Onder de import schrijf je vervolgens gewoon je eigen Scheme-code.

### 2.2 Definiëren van een library

Naast het importeren van bestaande libraries, kunnen we uiteraard ook zelf libraries gaan definiëren. Dat kan door middel van de `define-library` functie, die opnieuw vlak onder de lijn die de taal specificeert moet komen. Een minimale library die slechts één functie `f` definieert ziet er als volgt uit:

```

1 #lang r7rs
2
3 (define-library ()
4   (export f)
5   (import (scheme base)
6           (scheme write))
7   (begin
8     (define (f x) x)))

```

R7RS

De definitie van een library bestaat uit vier delen:

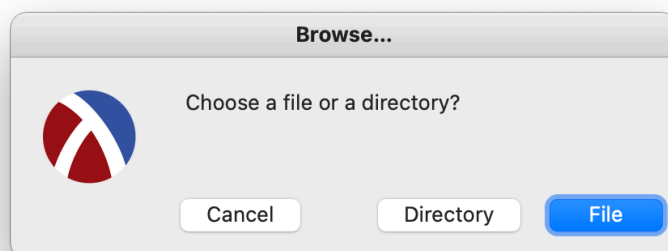
1. Het eerste argument bevat (in principe) de naam van de library, en de collectie(s) waarin de library is ondergebracht. Echter, de R7RS voor DrRacket doet niets met deze informatie, dus geven we om verwarring te vermijden een leeg argument (). In de code van de cursus (bv. de “a-d” folder) zal je hier vaak extra informatie zien staan. Dit is correct R7RS code, maar de R7RS plugin doet hier dus niets mee.
2. Het tweede argument is een export die bepaalt welke functies door de library aan haar gebruikers worden aangeboden. In dit geval bieden we enkel de functie “f” aan.
3. Het derde argument is een import die bepaalt welke functies onze library zelf kan gebruiken. In dit geval importeren we (scheme base) en (scheme write).
4. Het laatste argument is een body die bestaat uit een begin met daarin de definities van de library. Hier moeten we dus elke functie die we exporteren gaan definiëren, door gebruik te maken van de functies die we importeren. In dit geval definiëren we dus de geëxporteerde functie f door gebruik te maken van de (uit base) geïmporteerde functie “define”. Je mag in de body ook functies definiëren die niet geëxporteerd worden, bv. hulpfuncties.

## 2.3 Eigen libraries gebruiken in DrRacket

Er zijn 2 manieren om zelfgemaakte libraries bruikbaar te maken in je code. Oftewel werk je via de Package Manager van DrRacket (dit is aanbevolen) of je kan de locatie van je library toevoegen aan de Collection Path(s) van DrRacket.

### 2.3.1 Installeren via de DrRacket Package Manager (aanbevolen)

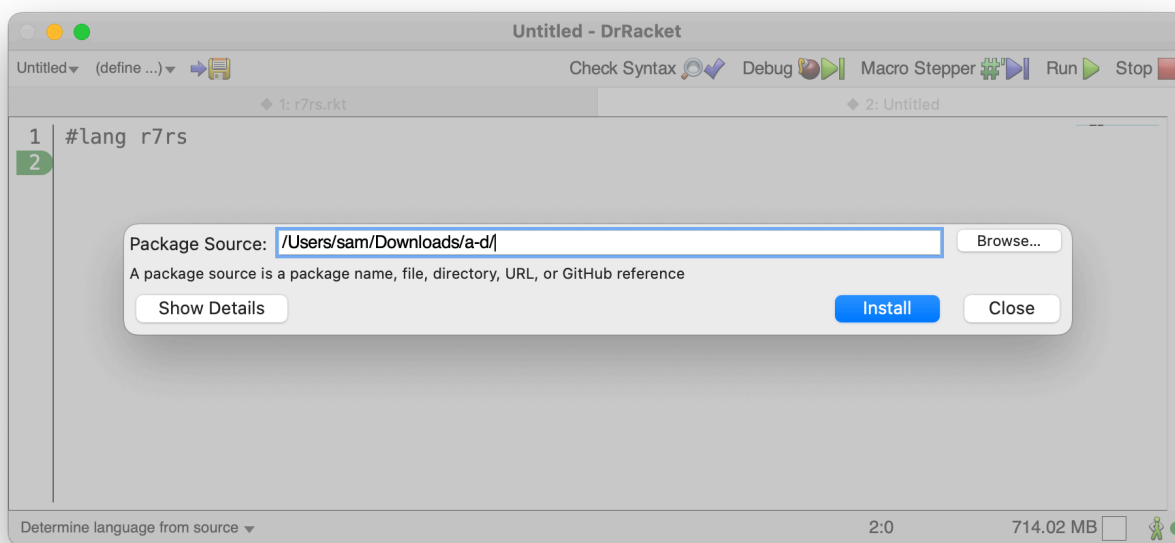
Het installeren van zelfgemaakte libraries verloopt gelijkaardig aan het installeren van de R7RS package (zie Hoofdstuk 1.1). In de “Install Package...” dialoog, klik op “Browse”. Je krijgt een dialoog die eruit ziet zoals Figuur 3. Selecteer “Directory”.



Figuur 3: De “Browse...” dialoog van DrRacket.

In principe kan elke folder kan geïnstalleerd worden als library, maar wees voorzichtig met naamgeving van folders. DrRacket kan niet goed overweg met speciale karakters en spaties in de naam van folders of files. Gebruik dus steeds simpele namen.

Bijvoorbeeld, de code van de cursus wordt gegeven als een folder genaamd “a-d” (deze kan je downloaden van Canvas). In deze folder zitten libraries die we voor deze cursus hebben geschreven. Deze folder kan geïnstalleerd worden via die “Install Package...” dialoog zoals voorheen. Hier selecteer je dan de gehele “a-d” folder zoals afgebeeld in Figuur 4. Druk daarna op “Install”. De package manager is klaar met installeren wanneer je onderaan niet meer kan klikken op de “Abort install” knop, en wel op de “Close output” knop. Je mag nu het scherm van de package manager gewoon sluiten.



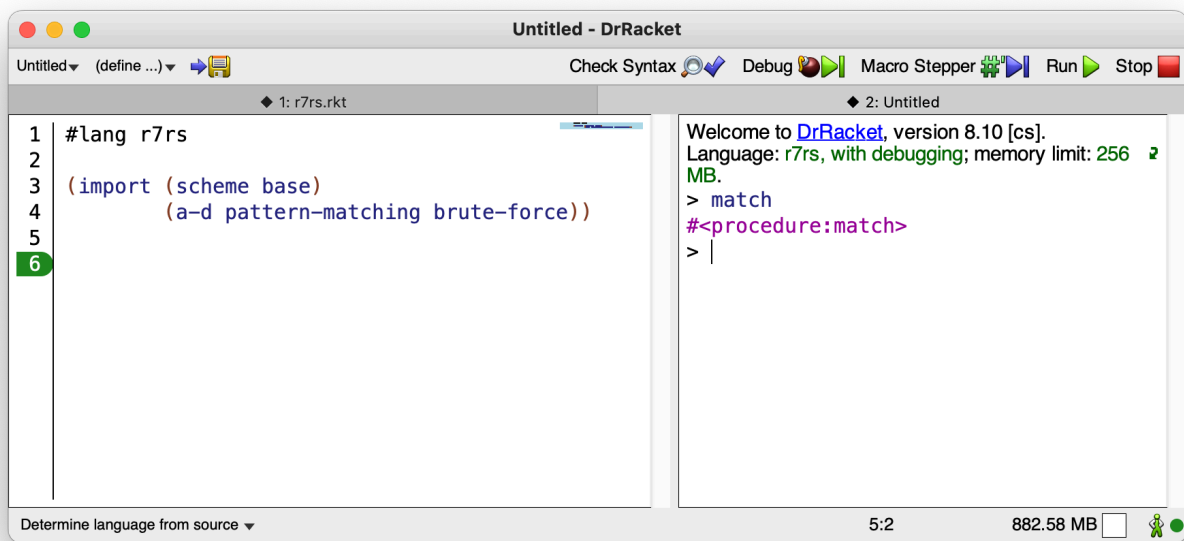
Figuur 4: De “Install Package...” dialoog van DrRacket. De libraries van de cursus zitten in een “a-d” folder, welke wordt geïnstalleerd.

Wanneer de installatie compleet is (zonder errors) kan je nu gebruik maken van de “a-d” libraries. Bijvoorbeeld, Figuur 5 toont een voorbeeld in DrRacket waar één van de “a-d” libraries wordt geïmporteerd. In dit geval zie je op lijn 4 een import van “(a-d pattern-matching brute-force)”. Deze import komt exact overeen met een bestand in de “a-d” folder die je hebt geïnstalleerd, namelijk het volgende bestand: “a-d/pattern-matching/brute-force.rkt”. Dit bestand bevat een R7RS library zoals uitgelegd in Hoofdstuk 2.2, welke dus wordt geïmporteerd.

Je kan zelfgemaakte libraries importeren op een gelijkaardige manier door ze in een folder te steken (bv. “mijn-libraries”) en je library hierin op te slaan onder een bepaalde bestandsnaam (by. “mijn-library.rkt”). Daarna installeer je de folder (“mijn-libraries”) als DrRacket package, en kan je je eigen R7RS library importeren via:

```
1 (import (scheme base)
2         (mijn-libraries mijn-library))
```

R7RS



Figuur 5: Na het installeren van een eigen library kan je deze library importeren in R7RS. Let op dat het pad van de geïmporteerde library moet overeenkomen met de folderstructuur die je hebt geïnstalleerd.

### 3 Importeren van libraries met name clashes

Wanneer je code twee libraries importeert die allebei dezelfde functie exporteren, toont DrRacket een foutmelding. In dat geval kan immers niet bepaald worden welke implementatie er moet gebruikt worden wanneer je die functie oproept. Dit probleem noemt men een “name clash”, omdat de namen van de functies als het ware met mekaar botsen. In dat geval kan je echter één van beide functies hernoemen, waardoor de name clash verdwijnt.

Stel dat de functie `+` de oorzaak van de name clash is. In dat geval kan je bijvoorbeeld het volgende doen:

```

1 (import (rename (scheme base) (+ base:)))
2       (my-collection my-library))

```

R7RS

De eerste lijn geeft aan dat we alle functies uit (scheme base) willen importeren, maar dat we (enkel) de functie `+` willen hernoemen in `base:+`. Merk op dat we het voorvoegsel “base:” kiezen omdat een voorvoegsel met “:” makkelijk aangeeft waar een hernoemde functie vandaan komt. In principe kan je een functie hernoemen naar eendert wat, en dit hoeft niet te beginnen met “base:”, en het hoeft ook niet de naam van de originele functie te bevatten.

Om alle functies die door een library worden geëxporteerd in één keer te hernoemen, kunnen we het volgende doen:

```

1 (import (prefix (scheme base) base:)
2       (my-collection my-library))

```

R7RS

Nu geeft de eerste lijn aan dat we alle functies uit (scheme base) willen importeren, maar dat we elk van die functies willen hernoemen naar de oorspronkelijke naam, met vlak daarvoor de prefix “base:”. De ingebouwde define wordt dus “base:define”, “+” wordt “base:++”, enzovoort, zonder dat we alle functies uit (scheme base) zelf moeten opsommen.

Alle mogelijke variaties rond importeren vind je terug in de R7RS documentatie.

## 4 R7RS documentatie

- Officiële documentatie: <https://standards.scheme.org/official/r7rs.pdf>
- Webversie van de documentatie: <https://standards.scheme.org/corrected-r7rs/r7rs-Z-H-1.html>
- Regels rondom importeren: <https://standards.scheme.org/corrected-r7rs/r7rs-Z-H-7.html>
- Overzicht van de standaard libraries [https://standards.scheme.org/corrected-r7rs/r7rs-Z-H-10.html#TAG: \\_\\_tex2page\\_chap\\_A](https://standards.scheme.org/corrected-r7rs/r7rs-Z-H-10.html#TAG: __tex2page_chap_A)