Интерпретатор на Scheme

Идея

Идеята на проекта е да симулира прост интепретатор на езика Scheme, написан на Haskell. Интерпретаторът работи в REPL (Read-Eval-Print Loop) режим, с възможност за по-нататъшно разширяване до четене от файлове.

Структура

Проектът се състои от дата тип Val, съдържащ различните възможни стойности на един атом, и Sexpr който представлява S-expression. Направени са различни парсъри (добавени са Parsec библиотеки за помощ с обработката), с които да се разчита входът. Има списък с примитиви, които представят базови Scheme примитиви и функциите им. Създаден е и тип, представляващ среда, в която се пазят дефинираните променливи и стойностите им по време на REPL манипулацията. Има функция eval, която оценява специални форми и изрази, която работи частично (cond не може да дефинира сложни функции). Завършва с функция за REPL изпълнението, която взема ред и го евалюира, вкарва резултата в средата и принтира резултата, очакваща нов вход при вече обновена среда и изкарващ грешка при проблем.

Стартиране

Проектът за пускане изисква пакета Parsec, който е предварително сложен в dependancy-то му. Проектът се стартира със стандартното stack run, след което се отваря автоматично REPL режимът и входът се въвежда по стандартния Scheme формат, “:q” за изход.