λ – Calculus Interpreter

Ορέστης Μελκονιάν *1115201000128*

ΓΕΝΙΚΑ

Το σύστημα που έχω υλοποιήσει είναι ένας διερμηνέας για όρους του λ-λογισμού. Δέχεται είσοδο λ-όρους από το τερματικό και τους φέρνει σε κανονική μορφή.

Το σύστημα έχει υλοποιηθεί σε C++ στο περιβάλλον ανάπτυξης Eclipse. Ακολούθησα μια αυστηρώς αντικειμενοστραφής προσέγγιση. Το σύστημα είναι συμβατό με Unix/Linux.

Για τη διεπαφή στο terminal έχω χρησιμοποιήσει την έτοιμη βιβλιοθήκη linenoise. Αρχικά χρησιμοποιούσα την GNU readline, αλλά τελικά δεν είναι εγκατεστημένη στα linux της σχολής, οπότε αναγκάστηκα να βρω μια minimal βιβλιοθήκη (~1000 γραμμές) ώστε να γίνεται compile μαζί με όλο το δικό μου σύστημα.

SETUP

1. cd /Debug/
2. make clean
3. make all
4. ./LCI

ΤΕΧΝΙΚΕΣ ΥΛΟΠΟΙΗΣΗΣ

Τα βασικά βήματα της διερμηνείας είναι τα εξής:

1. Λεκτική Ανάλυση (Scanning)
   Δέχεται ως είσοδο την εντολή ως συμβολοσειρά και επιστρέφει έναν πίνακα από μεμονωμένες λέξεις (tokens).
2. Συντακτική Ανάλυση (Parsing)
   Δέχεται έναν πίνακα από tokens και χρησιμοποιώντας recursive-descent parsing ανιχνεύει αν ο όρος που έχει δοθεί στην είσοδο μπορεί να παραχθεί από τη γραμματική του λ-λογισμού. Ταυτόχρονα κατασκευάσει ένα συντακτικό δέντρο (abstract syntax tree) που θα χρησιμοποιηθεί για να κανονικοποιηθεί ο όρος στο επόμενο βήμα.
3. Αποτίμηση (Evaluating)
   Δέχεται ως είσοδο ένα abstract syntax tree και, όσο υπάρχουν β-redex ή η-redex στον όρο, εκτελεί β-reductions ή η-conversions αντίστοιχα έως ότου ο όρος φτάσει σε κανονική μορφή. Οι πράξεις αυτές ουσιαστικά είναι μετασχηματισμοί του συντακτικού δέντρου.

ΚΛΑΣΕΙΣ

Παρακάτω αναφέρεται η λειτουργία των σημαντικότερων κλάσεων του συστήματος.

- *Scanner*
  Υλοποιεί τη λεκτική ανάλυση.
- *Parser*
  Υλοποιεί τη συντακτική ανάλυση.
- *Evaluator*
  Υλοποιεί την αποτίμηση του λ-όρου (κανονικοποίηση).
- *AST*
  Η δομή δεδομένων που αναπαριστά το συντακτικό δέντρο.
- *Alias Manager*
  Αναλαμβάνει τη διατήρηση όλων των aliases που έχουν οριστεί από το Prelude ή από τον χρήστη.
- *Consultor*
  Αναλαμβάνει το διάβασμα αρχείων .alias(consulting) και ενημερώνει αναλόγως τον Alias Manager.
- *Church Numerator*
  Κωδικοποιεί φυσικούς αριθμούς ως λ-όρους, σύμφωνα με την τεχνική που πρότεινε ο Alonzo Church.
- *List Manager*
  Μετατρέπει αναπαραστάσεις λιστών σε λ-όρους.
- *Operator Manager*
  Μετατρέπει infix αναπαραστάσεις σε κανονικές. [α + β = (+ α) β]
- *System Commands Manager*
  Εκτελεί εντολές συστήματος (αρχίζουν πάντα με ‘:’).
- *Translator*
  Ομαδοποιεί τις λειτουργίες των Alias Manager, Church Numerator, List Manager και Operator Manager σε μία κλάση.
- *Tester*
  Περιέχει tests για κάθε σημαντική μονάδα/λειτουργία του συστήματος (unit testing).

ΕΠΕΚΤΑΣΕΙΣ

- *Error Handling*
  Υπάρχει αναγνώριση των σφαλμάτων του χρήστη, κατάλληλη ενημέρωση αυτού και χειρισμός των σφαλμάτων.
- *Αριθμητική*
  Ο χρήστης μπορεί να δώσει ως είσοδο όρους που περιέχουν αριθμούς, τους οποίους το σύστημα αυτόματα μετατρέπει στους αντίστοιχους λ-όρους.
- *Aliasing*
  Το σύστημα είναι ικανό να διατηρεί ένα πλήθος ψευδωνύμων για σύνθετους όρους, καθιστώντας δυνατό ο χρήστης να δίνει ως είσοδο όρους που περιέχουν τέτοια ψευδώνυμα.
- *Αναδρομή*
  Ο χρήστης μπορεί να ορίσει αναδρομικές συναρτήσεις και το σύστημα θα μετατρέψει αυτόματα την συνάρτηση σε έναν έγκυρο λ-όρο χρησιμοποιώντας τον Y-combinator του Curry.
- *Λίστες*
  Ο χρήστης μπορεί να δίνει ως είσοδο όρους που περιέχουν αναπαραστάσεις λιστών. (πχ [1, 2, orestis, var21])
- *Εντολές συστήματος*
  Υπάρχουν εντολές που αλλάζουν settings του συστήματος ή δίνουν στο χρήστη χρήσιμες πληροφορίες.
- *Operators*
  Υποστηρίζεται ορισμός infix operator στα αρχεία .alias.
- *Prelude*
  Το σύστημα διαθέτει built-in ψευδώνυμα και operators για βασικές λειτουργίες, πράξεις, κτλ...
- *Ανίχνευση άπειρων λ-όρων*
  Μπορούν να ανιχνευθούν μερικοί από τους όρους, των οποίων η εκτέλεση δεν τερματίζει.
- *Eager/Lazy evaluation*
  Υποστηρίζεται η εναλλαγή στρατηγικής αποτίμησης μεταξύ eager και lazy.

ΔΙΕΠΑΦΗ – ΧΡΗΣΗ

Ο χρήστης τρέχει το εκτελέσιμο στο τερματικό και εμφανίζεται ένα prompt που δέχεται εντολές. Σαν εντολή μπορεί να δοθεί ένας λ-όρος για να αναχθεί σε κανονική μορφή ή μία εντολή συστήματος για εκτύπωση χρήσιμων πληροφοριών, αλλαγή παραμέτρων του συστήματος, consulting αρχείων, κλπ…

Βασική χρήση είναι η δημιουργία ενός αρχείου .alias, στο οποίο έχουν οριστεί διάφορες συναρτήσεις και κλήση της εντολής :consult <filename> ώστε το σύστημα να αποθηκεύσει τα ψευδώνυμα και τους operators που έχουν ορισθεί στο αρχείο.

Για να δείτε όλες τις διαθέσιμες εντολές εισάγεται την εντολή :help.

REPOSITORY

https://bitbucket.org/orestisMelkonian/calculus-interpreter/

ΣΥΜΒΑΣΕΙΣ – ΠΕΡΙΟΡΙΣΜΟΙ

- Υποστηρίζονται μόνο φυσικοί αριθμοί.
- Απαιτούνται κενά δεξιά και αριστερά των operator.
- Error handling μόνο στο κατώτερο επίπεδο αφαίρεσης (pure λ-calculus). Είναι στην ευθύνη του χρήστη να δώσει έγκυρες εντολές όταν περιέχουν αριθμούς, λίστες, aliases, operators.
- Λόγω περιορισμού στη μνήμη της stack, επιτρέπονται αριθμοί μέχρι το 500, λ-όροι μέχρι 10000 χαρακτήρες, κλπ… (:printLimits για να δείτε όλους τους χωρικούς περιορισμούς)
- Δεν επιτρέπεται η εκτέλεση αναδρομικών συναρτήσεων με eager evaluation.
- Οι λ-όροι πρέπει να είναι πλήρως παρενθετοποιημένοι.
- Υπάρχουν κάποια memory leaks, οπότε χρονοβόρες/πολύπλοκες εκτελέσεις ενδέχεται να οδηγήσουν σε stack overflow.
- Το σύστημα δεν θα καταφέρει να μετατρέψει τα αποτελέσματα σε readable μορφή (αριθμοί, λίστες, aliases…) αν έχει προέλθει προτυποποίηση μεταβλητών (α-conversion).
- Η διάγνωση δεν καλύπτει 100% τα σφάλματα, οπότε για λίγα λάθος input δεν θα δωθεί λεπτομέρεια του λάθους.
- Δεν επιτρέπονται εμφωλευμένες λίστες. (μπορεί να υλοποιηθεί πολύ εύκολα, αλλά ήδη έχουν αρχίσει οι εργασίες του επόμενου εξαμήνου…)
- Γενικότερα, οι απλές πράξεις απαιτούν εκθετικά βήματα μετατροπών, οπότε το σύστημα είναι αρκετά αργό.

ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΚΤΕΛΕΣΗΣ