

תוכנה 1

שחר פרץ

11 בנובמבר 2024

COMPILING AND INTERPITATION (1)

בהתינתן שפת ביטויים שנרצה לחשב. יהיו שתי אפשרויות: לקחת את השפה, לחשב, ולהציג תוצאה. השנייה, לעשות הכנה מראש, חישוב, ותוצאה.

ההכנה מראש תיקח זמן, אבל תגרום לחישוב להיות יותר מהיר. interpreter לא יעשה הכנה מראש, בעוד compiler הוא השלה של ההכנה מראש.

המרצה כתב שפה עם קומפילר, אופטימיזר (משפר את הקוד המקומפל) ואינטפרטר לשפה שהוא המציא. השפה היא שפה של ביטויים אלגבריים.

נקרא לשפה ביטויים פשוטים. נאשר פרמטרים (משתנים) כגון x, y, z_{37} וכו'. נאשר מספרים שלמים, ופעולות חשבון (רק $+, -, \cdot, ^{-1}$). נאשר סוגריים, אך פרט לכך אין סדר פעולות חשבון. לדוגמה, $2 + 3 \cdot 4 = 20$.

כלומר, no operator precedence (כלומר, אין קדימות לאופרטורים כאלו ואחרים). לדוגמה, ב-java: `if x + 5 < 8 && y + 7 < 3`

השפה היא right accosiative (לא משנה כי אין קדימות), כלומר החישוב מימין לשמאל. רוב השפות הפוכות.

נשים סוגריים כדי להגדיר קדימות: $((x + 5) < 8) \&\& ((y * 7) < 2)$

נעזוב את java לבינתיים.

“SIMPLE EXPRESSIONS” COMPILING (2)

lexical analysis – ניתוח של השפה. לדוגמה:

1. רצף ספרות ספרות

2. אות – אחריה ספרות או אותיות פרמטר

3. $+, -, *, /$ אופרטורים

2.1 Regex

regular expressions. דוגמאות:

dt/ – מספר בין 0-9

[a - zA - Z] – רצף תווים

אגב, בג'אווה regexים מתקמפלים בנפרד.

שיטת האינטרפריטציה שלהן, כדלהלן:

• מספר \leftarrow LV (תשמור בצד ותמשיך).

• פרמטר \leftarrow שלוף מה-dict.

• אופרטור \leftarrow רוקורסיה על מה שנשאר, על ביטוי אטומי (מספר או פרמטר).

• סוגריים \leftarrow רוקורסיה על מה שנמצא אחרי הסוגריים, עד שרואים סגור סוגריים.

ככה נקבל דברים מדהימים כמו $1 - 4 + 5 = -4$.

2.2 AST

abstrcut syntax tree, מתייחס להתליך של לקחת את ה-string, לחסוך את תהליך prasing (לנתח מחרוזת והבנת התחביר בה) בכך שה-output יהפוך לעץ. לא נעשה את זה.

NOTES ABOUT JAVA (3)

Map הוא dict ב-java. יש כמה אימפלמנטציות ל-dict, כמו hash map, tree map וכו'.
לליטרלים יש סוג. לדוגמה 5 הוא ליטרל מסוג int, בעוד 5L יהיה long, יתקיים 3.14 ש-double ואחרונה 3.14F יהיה float.
אל תשוו float באמצעות == כי זה לא מדויק מספיק. פר פלטפורמה (אפילו mac vs. amd או cpu vs. gpu) התוצאה יכולה להשתנות במקצת.
תמיד להשוות עד לכדי אפסילון (טולרנס).
סוגים פרימיטיביים – int, float, double, long וכו'. הם סוגים שקיימים ב-JVM ובד"כ גם קיימים בחומרה. לעומתם, יש גם classes כמו Float.
לעומת זה, float הוא לא class, ולכן אין לו methods. יש לו static methods כמו Float.parseFloat. זה חשוב, כי בכל מקום שבו ג'אווה מחזיק
containers, ג'אווה יכול להחזיק רק pointers, ולכן צריך להשתמש ב-class wrappers של הפרימיטיביים. ה-exception היחיד הוא array, שהוא
פרימיטיבי שמנוהל ע"י ה-JVM.

HENIACOINS (4)

big-endian/little-endian – מגדיר את הסדר בו דברים נשלחים ברשת. (כאשר big endiann אומר להתחיל מהחזקה הגדולה). לשיעור הבא:
להרחיב את ה-blockchain מעבר ל-1, ולעשות validation ל-blockchain. כלומר, צריך:

- לייצר blockchain של 3 מטבעות
- לעשות validate ל-chain.

אימיל:

tal.frangi@gmail.com