

1. כן, let expression הוא special form מכיוון שיש לו שיטת איולויאציה שונה מסדר האיולויאציה הרגיל שבו עושים evaluate לביטוי הנוכחי. כלומר, קודם כל מחשבים את המשתנים המוגדרים על ידו ואז מפעיל על כל אחד מהמשתנים את clouser שבער אוולויאציה ב body של let.

כלומר let שונה בדרך האיולויאציה שלו.

2. התשובה אינה חד משמעית, תלוי במימוש הLET שלנו. כלומר אם יצרנו LAMBDA כלשהי בתוך הLET אז נוצר CLOSURE בתוך הLET ואחרת לא נוצר LAMBDA ולכן אין CLOSURE. אנו יודעים כי CLOSURE נוצר רק בתוך הLAMBDA מכיוון שהוא הVALUE שלה.

3.

-כאשר יש חלוקה באפס.

-כאשר השתמשנו ב type הלא נכון, לדוגמה התבקשנו להכניס פרמטר מסוג Boolean והעברנו Number .

-כאשר החלק אותו אנחנו בודקים בביטוי הוא אינו מטיפוס בוליאני – לדוגמה (14+5? #t: #f).

-כאשר מגדירים משתנה וסביבה והסביבה אינה נכונה.

4. באמצעות valueToLitExp מתבצעת החלפה של ערכים לביטויים. כלומר פותר את הבעיה במודל אשר מחשב ערכים, בגוף של CLOSURE.

5. כאשר מופעל closure בשיטת normal evaluation, האופרנדים מחושבים לאחר חישוב האופרטור ולאחר מכן גוף הclosure מחושב, כאשר האופרנדים הם עדיין ביטויים. האופרנדים יומרו לערך רק כאשר צריך לחשב בשלב האחרון של החישוב אופרטור פרימיטיבי. בשום שלב בחישוב לא עוברים ערכים לפונקציה שמקבלת ביטויים, לכן אין צורך ב-valueToLitExp.

6. ההבדל הוא באופן החישוב של האינטרפרטר. אופרטור פרימיטיבי מחושב ע"י פונקציה מוגדרת מראש בשפת ה-meta איתה כותבים את האינטרפרטר של השפה. בכל אופרטור פרימיטיבי, על מנת לבצע הפעלה שלו, יש לחשב בצורה מלאה את כל הארגומנטים שנקראו ולא ניתן להשאיר חלקים לא מחושבים.

Special form לעומת זאת, הוא ביטוי מורכב שיש לו חוק חישוב אחר. לדוגמה, ביטוי מצורה if מורכב משלושה חלקים – test, then, else, כאשר קודם מחשבים את test ולאחר מכן, בהתאם לתוצאה נחליט האם לחשב את then או את else, כלומר לא כל החלקים מחושבים.

7. סיבה טובה לעבור לשיטת הסביבות הינה שפעולת החלפת שמות המשתנים היא מסובכת ויקרה.

בשיטת ההחלפה נדרשות פעולות יקרות כגון שינוי שמות המשתנים החופשיים והחלפתם וכד', מה שיגרור העתקה של ב-AST של ביטויי הלמבדה. דבר זה מוביל לשימוש רב בזיכרון ומאט את זמן הרצה של התוכנית. בשיטת הסביבות אין אנו נדרשים לבצע כאלו פעולות, לכן אנו חוסכים זמן ריצה ושימוש מופרז בזיכרון, לכן נעדיף לעבור לעבוד בשיטה זו.

