

מבנה הנתונים בו השתמשנו הוא ה- *node* שניתן בקבצי העזר:

```
typedef struct node {
    char *type;
    char *value;
    struct node *sibling;
    struct node *child;
} ParserNode;
```

כאשר *type* הוא סוג האסימון כפי שהוגדרו בתרגיל בית רטוב 1, *value* הוא הערך המסויים של האסימון, *sibling* הוא מצביע אל צומת אח (אוסף *siblings* מציין את כל הרכיבים ששייכים לאותו כלל גזירה) ו- *child* הוא מצביע אל רשימת ה- *siblings* הנגזרת מסימבול כלשהו.

המנתח הלקסיקלי שלנו (*lexer_part2.lex*), אשר נקרא על ידי המנתח התחבירי (*bison_part2.ypp*), מחלץ אסימונים מקובץ הקלט. עבור כל אסימון המתח התחבירי קובע את הערך הסמנטי שלו להיות צומת עלה חדש ומחזיר קוד מספרי המייצג את האסימון. עבור כל כלל גזירה, המנתח התחבירי יוצר רשימה של צמתים אחים עבור כל סימבול או טרמינל שנמצאים בצד ימין של כלל הגזירה וצומת יחיד עבור הסימבול שבצד שמאל של כלל הגזירה. הצומת של הסימבול שבצד שמאל של הכלל נקבע להיות ההורה של רשימת הצמתים. לבסוף אנו מבצעים השמה של אותו צומת הורה לערך הסמנטי של הסימבול שבצד שמאל של כלל הגזירה. כך הצמתים "מחלחלים" והעץ נבנה מלטה למעלה – אותו סימבול הורה יופיע בהמשך בצד ימין של כלל גזירה ויהיה חלק מרשימה עם הורה אחר. לבסוף, כאשר נגיע לסימבול ההתחלה נדפיס את העץ.

הפונקציות ליצירת ולשרשור הצמתים נתונות בקבצי העזר.

הנחות:

- סדר קדימות האופרטורים האריתמטיים והלוגיים והאסוציאטיביות שלהם נקבעה בהתאם למקובל.
- ביטוי *else* משויך ל- *if* הראשון שלפניו.
- איתור והדפסת שגיאות לקסיקליות מבוצעים בקובץ המנתח הלקסיקלי כולל ציון הביטוי הלא מוכר ומיקומו בקובץ הקלט.

- איתור והדפסת שגיאות תחביריות מבוצעים בקובץ המנתח התחבירי כולל ציון הביטוי
הלא מוכר ומיקומו בקובץ הקלט.