

### گزارش فاز ۳ - نگین فیروزیان ۹۴۳۱۰۱۸

بعد تعریف Start Symbol تابع exportIntermediateCode فراخوانی می‌شود و چون این rule در آخر اجرا می‌شود کد و سیمبل تبیل نهایی چاپ می‌شود.

همچنین از سه کلاس Quadruple برای ذخیره کد سه آدرس ، کلاس Eval برای تولید Semantic و کلاس SymbolTable برای ذخیره نماد ها استفاده می‌شود.

نمونه کد ورودی و کد جنریت شده:

int negin76 () comeback;

```
// ////////////////////////////////// Symbol Table ////////////////////////////////// //
```

```
int TempVar0;
```

```
// ////////////////////////////////// Three Address Code ////////////////////////////////// //
```

```
Line0:      TempVar0 = 0;
Line1:      if (TempVar0) goto Line3;
Line2:      goto Line3;
Line3:      printf("Process is terminated with no error.\n");
             getchar();
             return 0;
Line4:      printf("Array Error: Index out of bound!\n");
             getchar();
             return -1;
Line5:      printf("Array Error: Invalid array size!\n");
             getchar();
             return -2;
.
```

برای تولید کد سه آدرس با توجه به rule بخش semantic آن تولید می‌شود.

که به شرح زیر می‌باشد:

تعریف non terminal جدید:

```
yyval = new Eval();
((Eval)yyval).place = ((Eval)(yyvsp.valueAt (1-(1)))) .place;
((Eval)yyval).type = ((Eval)(yyvsp.valueAt (1-(1)))) .type;
((Eval)yyval).nextList = ((Eval)(yyvsp.valueAt (1-(1)))) .nextList;
((Eval)yyval).trueList = ((Eval)(yyvsp.valueAt (1-(1)))) .trueList;
((Eval)yyval).falseList = ((Eval)(yyvsp.valueAt (1-(1)))) .falseList;
```

## تایپ Bool و Boolean

```
yyval = new Eval();
((Eval)yyval).place = newTemp(Eval.TYPE_CODE_BOOLEAN, false);
((Eval)yyval).type = Eval.TYPE_CODE_BOOLEAN;
((Eval)yyval).trueList = Eval.makeList(nextQuad() + 1);
((Eval)yyval).falseList = Eval.makeList(nextQuad() + 2);
((Eval)yyval).nextList = Eval.merge(((Eval)yyval).trueList,
((Eval)yyval).falseList);

emit(":=", "" + String.valueOf(lexChar) + "'", null, ((Eval)yyval).place);
emit("check", ((Eval)yyval).place, null, String.valueOf(nextQuad() + 2)); //
result may be backpatched.
emit("goto", null, null, String.valueOf(nextQuad() + 1));
```

## تایپ Character و Char

```
yyval = new Eval();
((Eval)yyval).place = newTemp(Eval.TYPE_CODE_CHAR, false);
((Eval)yyval).type = Eval.TYPE_CODE_CHAR;
((Eval)yyval).trueList = Eval.makeList(nextQuad() + 1);
((Eval)yyval).falseList = Eval.makeList(nextQuad() + 2);
((Eval)yyval).nextList = Eval.merge(((Eval)yyval).trueList,
((Eval)yyval).falseList);

emit(":=", "" + String.valueOf(lexChar) + "'", null, ((Eval)yyval).place);
emit("check", ((Eval)yyval).place, null, String.valueOf(nextQuad() + 2)); //
result may be backpatched.
emit("goto", null, null, String.valueOf(nextQuad() + 1));
```

## تایپ Integer و Int

```
yyval = new Eval();
((Eval)yyval).place = newTemp(Eval.TYPE_CODE_INTEGER, false);
((Eval)yyval).type = Eval.TYPE_CODE_INTEGER;
((Eval)yyval).trueList = Eval.makeList(nextQuad() + 1);
((Eval)yyval).falseList = Eval.makeList(nextQuad() + 2);
((Eval)yyval).nextList = Eval.merge(((Eval)yyval).trueList,
((Eval)yyval).falseList);

emit(":=", String.valueOf(lexInt), null, ((Eval)yyval).place);
emit("check", ((Eval)yyval).place, null, String.valueOf(nextQuad() + 2)); //
result may be backpatched.
emit("goto", null, null, String.valueOf(nextQuad() + 1));
```

## عملیات جمع:

```
if (((((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_INTEGER ||
(Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_CHAR || ((Eval)
(ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && (((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_INTEGER ||
(Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_CHAR || ((Eval)
(ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_BOOLEAN)
    || (((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_REAL &&
```

```

((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_REAL)) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(((Eval) (ystack.valueAt(2 - (2))))).type, false);
    ((Eval) yyval).type = ((Eval) (ystack.valueAt(2 - (2))))).type;
    emit("+", ((Eval) (ystack.valueAt(2 - (2))))).place, ((Eval) (ystack.valueAt(2 - (2))))).place1, ((Eval) yyval).place);
} else if (((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_CHAR || ((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && ((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_REAL) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval) yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt(2 - (2))))).place, TYPE_STRING_REAL, tmp);
    emit("+", tmp, ((Eval) (ystack.valueAt(2 - (2))))).place1, ((Eval) yyval).place);
} else if (((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_REAL
    && (((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_CHAR || ((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_BOOLEAN) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval) yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt(2 - (2))))).place1, TYPE_STRING_REAL, tmp);
    emit("+", ((Eval) (ystack.valueAt(2 - (2))))).place, tmp, ((Eval) yyval).place);
} else {
    System.err.println("Error! Invalid type for \"+\" operation.");
    return YYABORT;
}

```

عمليات تفريق:

```

if((((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && (((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_BOOLEAN)
    || (((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_REAL &&
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_REAL) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(((Eval) (ystack.valueAt (2-(2))))).type, false);
    ((Eval) yyval).type = ((Eval) (ystack.valueAt (2-(2))))).type;
    emit("-", ((Eval) (ystack.valueAt (2-(2))))).place, ((Eval) (ystack.valueAt (2-(2))))).place1, ((Eval) yyval).place);
} else if (((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && ((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_REAL) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval) yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt (2-(2))))).place, TYPE_STRING_REAL, tmp);
    emit("-", tmp, ((Eval) (ystack.valueAt (2-(2))))).place1, ((Eval) yyval).place);
} else if (((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_REAL

```

```

        && (((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_BOOLEAN)) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval) yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt (2-(2))))).place1, TYPE_STRING_REAL, tmp);
    emit("-", ((Eval) (ystack.valueAt (2-(2))))).place, tmp, ((Eval) yyval).place);
} else {
    System.err.println("Error! Invalid type for \"+\" operation.");
    return YYABORT;
}

```

عمليات ضرب:

```

if (((((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_CHAR || ((Eval)
(ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && (((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_CHAR || ((Eval)
(ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_BOOLEAN)
    || (((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_REAL &&
((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_REAL)) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(((Eval) (ystack.valueAt(2 - (2))))).type, false);
    ((Eval) yyval).type = ((Eval) (ystack.valueAt(2 - (2))))).type;
    emit("=", ((Eval) (ystack.valueAt(2 - (2))))).place, ((Eval) (ystack.valueAt(2
- (2))))).place1, ((Eval) yyval).place);
} else if (((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_CHAR || ((Eval)
(ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && ((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_REAL) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval) yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt(2 - (2))))).place, TYPE_STRING_REAL, tmp);
    emit("*", tmp, ((Eval) (ystack.valueAt(2 - (2))))).place1, ((Eval)
yyval).place);
} else if (((Eval) (ystack.valueAt(2 - (2))))).type == Eval.TYPE_CODE_REAL
    && (((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_CHAR || ((Eval)
(ystack.valueAt(2 - (2))))).type1 == Eval.TYPE_CODE_BOOLEAN)) {
    yyval = new Eval();
    ((Eval) yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval) yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt(2 - (2))))).place1, TYPE_STRING_REAL, tmp);
    emit("=", ((Eval) (ystack.valueAt(2 - (2))))).place, tmp, ((Eval) yyval).place);
} else {
    System.err.println("Error! Invalid type for \"+\" operation.");
    return YYABORT;
}

```

عمليات تقسيم:

```

if((((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && (((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_BOOLEAN)
    || (((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_REAL &&
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_REAL)) {
    yyval = new Eval();
    ((Eval)yyval).place = newTemp(((Eval) (ystack.valueAt (2-(2))))).type, false);
    ((Eval)yyval).type = ((Eval) (ystack.valueAt (2-(2))))).type;
    emit("/", ((Eval) (ystack.valueAt (2-(2))))).place, ((Eval) (ystack.valueAt (2-
2))))).place1, ((Eval)yyval).place);
} else if((((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && ((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_REAL) {
    yyval = new Eval();
    ((Eval)yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval)yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt (2-(2))))).place, TYPE_STRING_REAL, tmp);
    emit("/", tmp, ((Eval) (ystack.valueAt (2-(2))))).place1, ((Eval)yyval).place);
} else if((((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_REAL
    && (((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_BOOLEAN)) {
    yyval = new Eval();
    ((Eval)yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval)yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval) (ystack.valueAt (2-(2))))).place1, TYPE_STRING_REAL, tmp);
    emit("/", ((Eval) (ystack.valueAt (2-(2))))).place, tmp, ((Eval)yyval).place);
} else {
    System.err.println("Error! Invalid type for \"+" operation.");
    return YYABORT;
}

```

عملیات باقی مانده:

```

if((((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && (((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_BOOLEAN)
    || (((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_REAL &&
((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_REAL)) {
    yyval = new Eval();
    ((Eval)yyval).place = newTemp(((Eval) (ystack.valueAt (2-(2))))).type, false);
    ((Eval)yyval).type = ((Eval) (ystack.valueAt (2-(2))))).type;
    emit("%", ((Eval) (ystack.valueAt (2-(2))))).place, ((Eval) (ystack.valueAt (2-
2))))).place1, ((Eval)yyval).place);
} else if((((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_INTEGER ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_CHAR ||
((Eval) (ystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_BOOLEAN)
    && ((Eval) (ystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_REAL) {
    yyval = new Eval();
    ((Eval)yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);

```

```

    ((Eval)yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval)(yystack.valueAt (2-(2))))).place, TYPE_STRING_REAL, tmp);
    emit("%", tmp, ((Eval)(yystack.valueAt (2-(2))))).place1, ((Eval)yyval).place);
} else if(((Eval)(yystack.valueAt (2-(2))))).type == Eval.TYPE_CODE_REAL
    && (((Eval)(yystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_INTEGER ||
    ((Eval)(yystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_CHAR ||
    ((Eval)(yystack.valueAt (2-(2))))).type1 == Eval.TYPE_CODE_BOOLEAN) {
    yyval = new Eval();
    ((Eval)yyval).place = newTemp(Eval.TYPE_CODE_REAL, false);
    ((Eval)yyval).type = Eval.TYPE_CODE_REAL;
    String tmp = newTemp(Eval.TYPE_CODE_REAL, false);
    emit("cast", ((Eval)(yystack.valueAt (2-(2))))).place1, TYPE_STRING_REAL, tmp);
    emit("%", ((Eval)(yystack.valueAt (2-(2))))).place, tmp, ((Eval)yyval).place);
} else {
    System.err.println("Error! Invalid type for \"+\" operation.");
    return YYABORT;
}

```