به نام آنکه جان را فکرت آموخت

امیر ارسلان یاوری - ۴۰۲۲۰۳۴۹۷

امتحان درس سیستم‌های تحمل‌پذیر اشکال دانشگاه صنعتی شریف — استاد: دکتر اجلالی

پاسخ سوال سوم — نیمسال اول ۱۴۰۴-۱۴۰۳

---

۳) یک قطعه کد Maple یا Mathematica توسعه دهید که توصیف یک نمودار RBD را در قالب یک فایل متنی در ورودی دریافت کند (برای توصیف RBD در یک فایل متنی خودتان یک قرارداد ارائه دهید) و سپس عبارت جبری Reliability کل سیستم را در خروجی ارائه دهد. دقت کنید که خروجی باید عبارت جبری باشد و نه یک عدد.

برای حل این سوال کد maple زیر نوشته شده است که در ادامه مفصلا آن را توضیح خواهم داد:

```
SeriesReliability := proc(blocks::list)
    local i, R;
    R := 1;
    for i in blocks do
        R := R * i;
    end do;
    return R;
end proc:

ParallelReliability := proc(blocks::list)
    local i, R;
    R := 1;
    for i in blocks do
        R := R * (1 - i);
    end do;
    return 1 - R;
end proc:

ParseRBD := proc(rbdString::string)
    local parsedString;
    parsedString := StringTools:-SubstituteAll(rbdString, "S(", "SeriesReliability([");
    parsedString := StringTools:-SubstituteAll(parsedString, "P(", "ParallelReliability([");
    parsedString := StringTools:-SubstituteAll(parsedString, ")", "])");
    return eval(parse(parsedString));
end proc:

RemoveB := proc(rbdString::string)
    local openBracketPos, closeBracketPos, cleanedString;
    cleanedString := rbdString;
    while StringTools:-Search("B(", cleanedString) <> 0 do
        openBracketPos := StringTools:-Search("B(", cleanedString);
        closeBracketPos := StringTools:-Search(")", cleanedString, openBracketPos);
        # Remove "B(" and the corresponding closing ")"
        cleanedString := StringTools:-Substitute(cleanedString, cleanedString[openBracketPos..openBracketPos + 1], "");
        cleanedString := StringTools:-Substitute(cleanedString, cleanedString[closeBracketPos - 2..closeBracketPos - 2], "");
    end do;
    return cleanedString;
end proc:

ReadRBDFromFile := proc(filename::string)
    local rbdString;
    rbdString := FileTools:-Text:-ReadFile(filename);
    rbdString := StringTools:-Trim(rbdString);
    # Suppressing the print here to avoid showing the raw RBD string
    rbdString := RemoveB(rbdString);  # Remove all occurrences of B() blocks
    return rbdString;
end proc:

ComputeParametricReliabilityFromFile := proc(filename::string)
    local rbdString, reliability;
    rbdString := ReadRBDFromFile(filename);
    reliability := ParseRBD(rbdString);
    return reliability;
end proc:

# Sample file path
filename := "/home/andre/code/uni/fault/midterm/Q3/parametric_rbd.txt":
reliability := ComputeParametricReliabilityFromFile(filename):
print("The symbolic reliability of the system is: ", reliability);
```

قسمت اول کد یک تابع به نام SeriesReliability است که قابلیت اطمینان یک سیستم سری را محاسبه می‌کند. تابع لیستی از مقادیر را دریافت می‌کند، آن‌ها را ضرب می‌کند و نهایتا حاصل‌ضرب را به عنوان خروجی برمی‌گرداند.

قسمت دوم کد یک تابع به نام ParallelReliability تعریف می‌کند که قابلیت اطمینان یک سیستم موازی را محاسبه می‌کند. تابع لیستی از مقادیر را دریافت می‌کند، احتمال از کار افتادن هر بلوک را محاسبه کرده و آن‌ها را در هم ضرب می‌کند، سپس در نهایت با تفریق حاصل از عدد ۱، قابلیت اطمینان کلی سیستم موازی را برمی‌گرداند.

قسمت سوم کد یک تابع پارسر است به طوری که ورودی خوانده شده از فایل را تبدیل به جمله‌ای اجرایی می‌کند این تابع تمامی عبارت‌های S(را به SeriesReliability([و همچنین تمامی عبارت‌های P(را به ParallelReliability([تبدیل می‌کنه بعدش همه‌ی پرانتزهای بسته) را با]) جایگزین می‌کند تا برای فراخوانی توابع SeriesReliability و ParallelReliability آماده باشند. درآخر جمله‌ی پردازش‌شده را اجرا و نتیجه را بر میگرداند.

تابع چهارم وظیفه حذف تمام مواردی که به شکل B( ) شروع می‌شوند و معادل پرانتز بسته‌ی مربوط به آن‌ها از یک جمله را دارد. ابتدا موقعیت B(و سپس موقعیت پرانتز بسته مربوطه پیدا می‌شود، و هر دو از جمله حذف می‌شوند. پس از فراخوانی این تابع جمله‌ی بدون B(برگردانده می‌شود. این قسمت اضافه شده تا امکان محاسبه برای سیستم‌های دارای Bridge را هم فراهم کند.

تابع پنجم ب اسم ReadRBDFromFile فایل حاوی RBD را به فرمت گفته شده در ادامه می‌خواند، بعدش از تابع RemoveB استفاده می‌کند تا تمامی B( ) های آن را حذف کند.
فرمت فایل به صورتی است که کامپونت‌هایی که موازی هستند داخل یک پرانتز باز و بسته P نوشته می‌شوند برای مثال:

P(R1 , R2)

برای کامپونت‌های سریال نیز به همین صورت اما داخل پرانتز S نوشته می‌شوند.

برای بریج هم باید رابطه‌ی بریج نوشته شود و داخل پرانتز B نوشته شود. برای مثال:

B(P(S(R1, R2) , S(R3, R4)) + S(P(R1, R3) , P(R2, R4)))

تابع آخر نیز ابتدا از تابع ReadRBDFromFile استفاده می‌کند تا فایل RBD را می‌خواند و پردازش کرده بعدش با استفاده از تابع ParseRBD این جمله را به فرمول قابلیت اطمینان تبدیل کرده و در نهایت مقدار محاسبه‌شده‌ی قابلیت اطمینان سیستم را برمی‌گرداند.

در پایان هم متغیرهای لازمه مقدار دهی شده و توابع فراخوانی و نتیجه چاپ می‌شود:

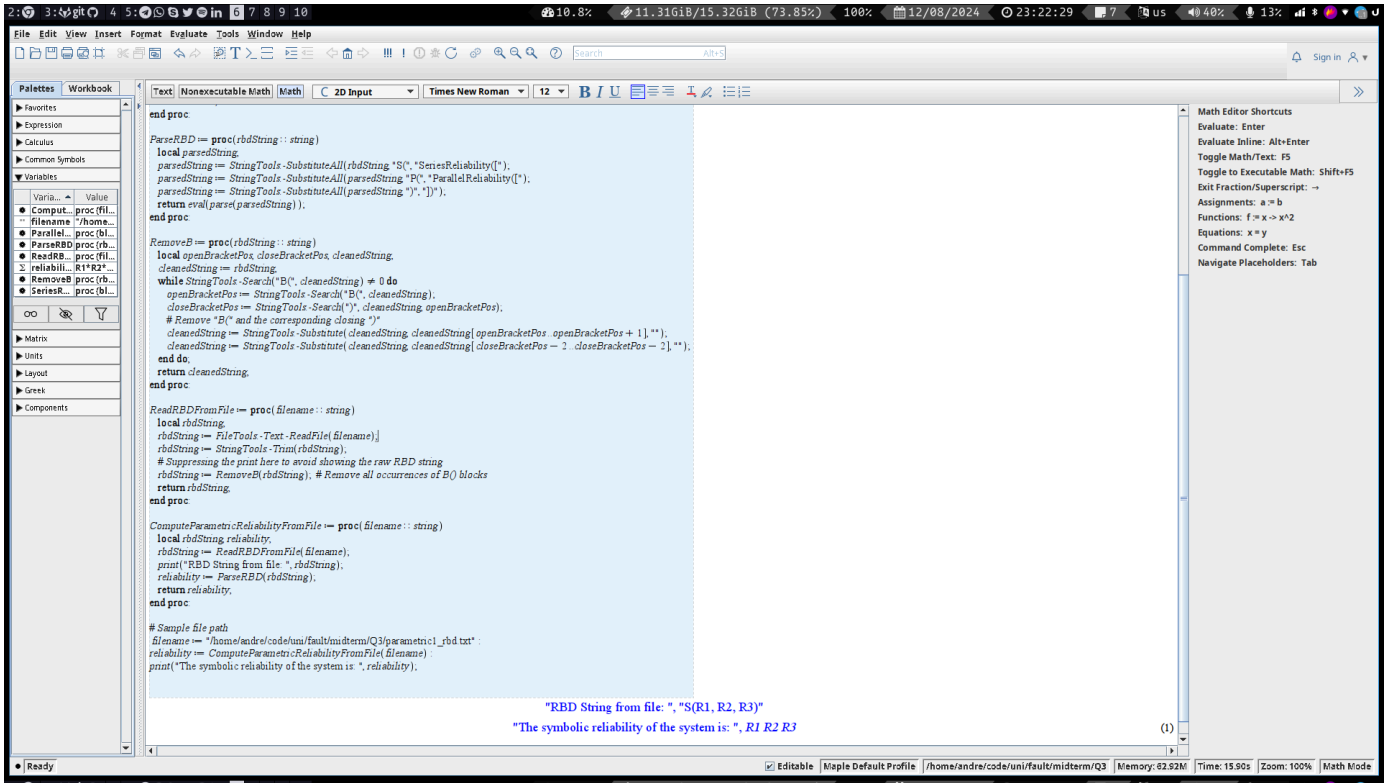
filename := "/home/andre/code/uni/fault/midterm/Q3/parametric_rbd.txt":

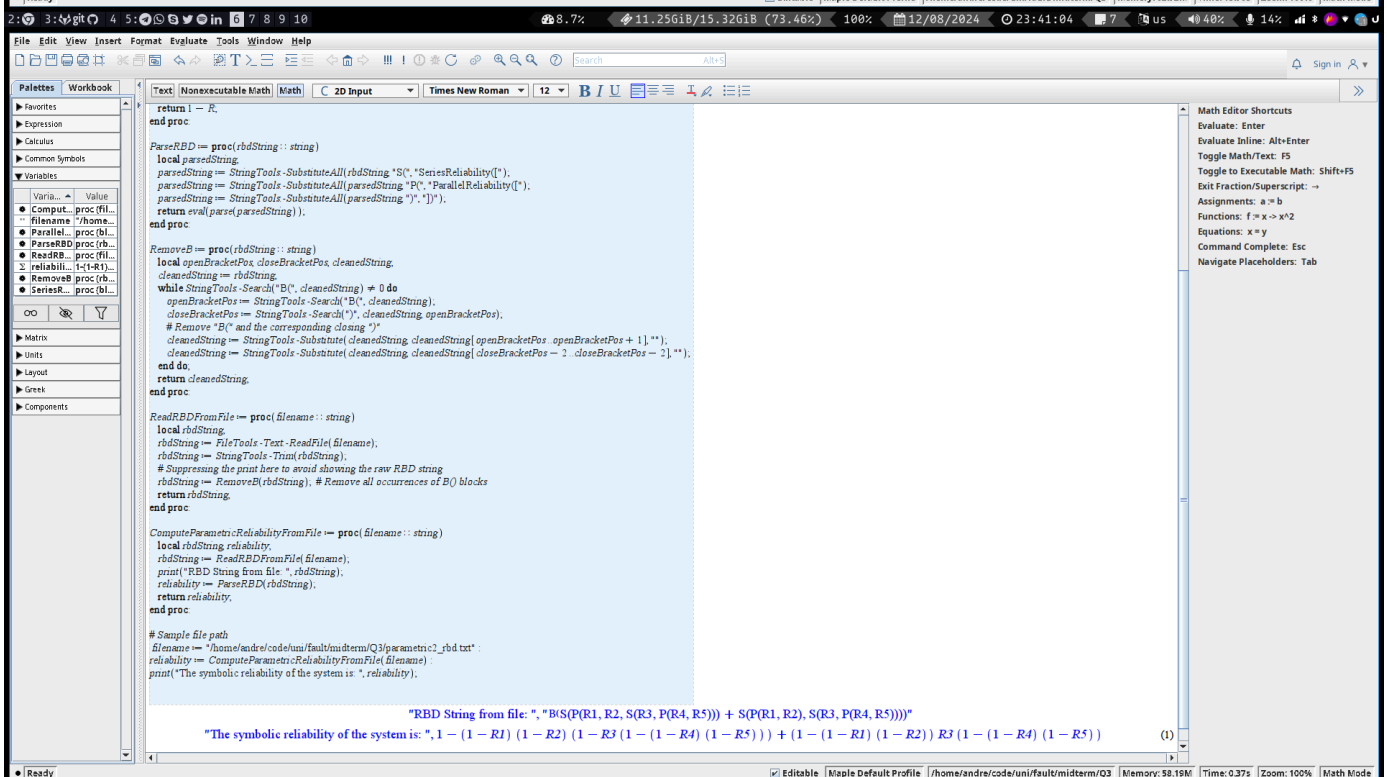
reliability := ComputeParametricReliabilityFromFile(filename):

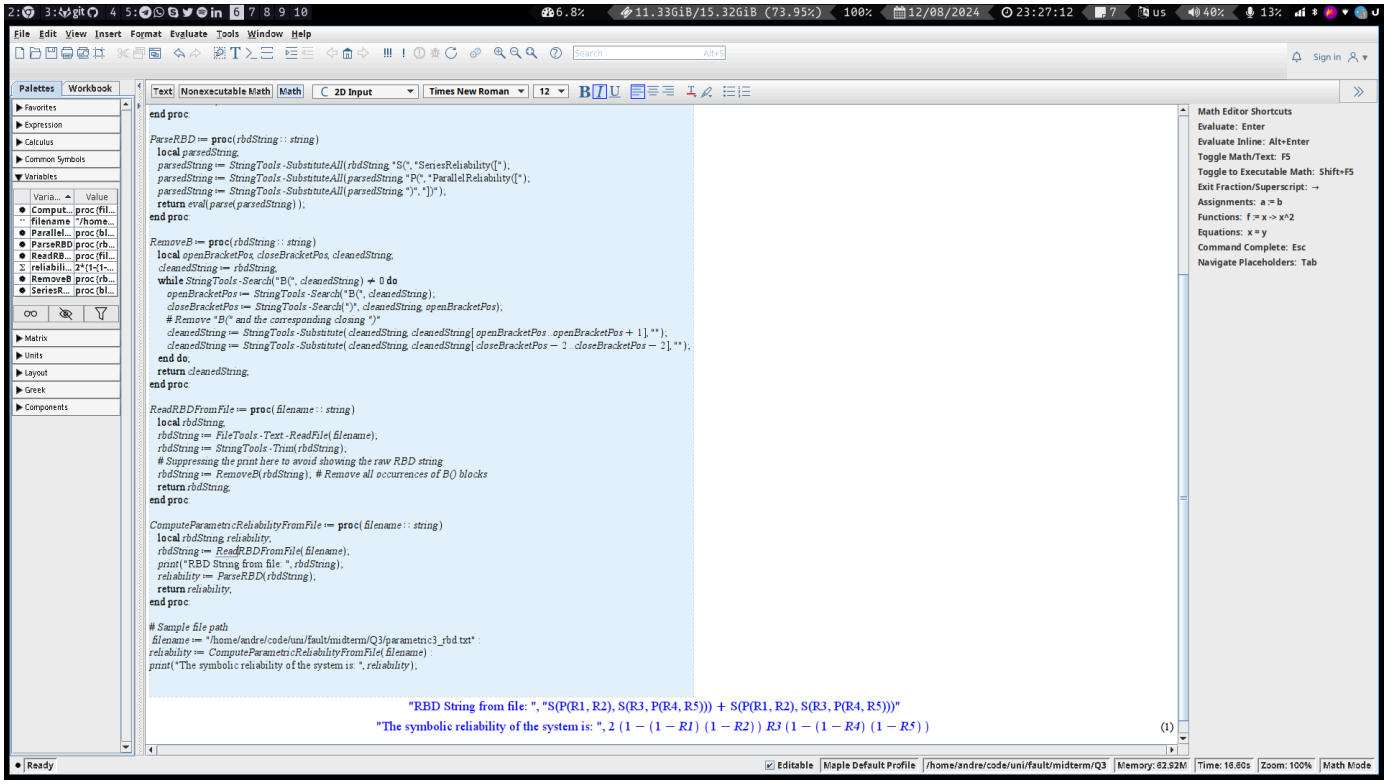
```
print("The symbolic reliability of the system is: ", reliability);
```

نتیجه‌ی اجرا برای حالات مختلف:

```
andre  (e) base  main  …  fault  midterm  Q3  for i in `ls para
metric*`; do echo $i && cat $i; done
parametric1_rbd.txt
S(R1, R2, R3)
parametric2_rbd.txt
B(S(P(R1, R2), S(R3, P(R4, R5))) + S(P(R1, R2), S(R3, P(R4, R5))))
parametric3_rbd.txt
S(P(R1, R2), S(R3, P(R4, R5))) + S(P(R1, R2), S(R3, P(R4, R5)))
andre  (e) base  main  …  fault  midterm  Q3
```







مطابق با تصاویر بالا کد برای مقادیر متفاوت پاسخ صحیح می‌دهد.

با تشکر :)