Rendszertervezés laboratórium 1

1. Laboratórium feladatai

A laboratórium időtartama alatt végezze el az alábbi feladatokat, és dokumentálja a munkáját! A munkája értékelésénél a dokumentációt, az elkészített forráskódot vesszük figyelembe.

Megjegyzések:

- A munkát párban végezzük, így elég az egyik gépen megoldani a feladatot.
- A munkájához vezessen egyszerű online jegyzőkönyvet (mondjuk google docs alkalmazás segítségével), melybe emelje ki a fontos kódrészleteket, és illessze be a kért képernyőrészleteket.
- A 2., 3. és 4. feladat után mutassa be az elkészült munkát a laborvezetőnek.
- A feladatokat úgy készítse el, hogy a korábbi megoldásai továbbra is működjenek.
- Amennyiben szükséges, az összekészített kiindulási állapot megtalálható a tantárgy oldalán.

1. **Előkészületek**
   1.1. Nyissa meg a VMWareImages/ReLab1Lab1/MDSD2017.vmx virtuális gépet
   1.2. Indítsa el a virtuális gépet, jelentkezzen be a meres/LaborImage felhasználónév/jelszó párost.
   1.3. Nyissa meg az asztalon elhelyezett „ReLab1” nevű parancsikonnal az Eclipse alkalmazást.
2. **Modell Bejárása**
   2.1. Indítsa el a `hu.bme.mit.yakindu.analysis/RunWithSmallModel.launch` konfigurációval az alkalmazást, tekintse meg a konzol kimenetet és a generált .
   2.2. Egészítse ki az alkalmazást, hogy az állapotok mellett a tranzíciókat is kiírja az alábbi formában: `Kiinduló állapot neve -> Cél állapot neve`
   2.3. Egészítse ki az alkalmazást, hogy az keresse meg azokat a csapda állapotokat, amiből nem vezet ki él, és írassa ki a nevüket! A program működését demonstrálja egy példával (rajzolja át a modellt).
   2.4. Egészítse ki az alkalmazást, hogy keresse meg azokat az állapotokat, amelyeknek nincs neve, és nevezze el őket! A program működését demonstrálja egy példával.
3. **Yakindu kódgenerátor használata**
   3.1. Egy összetett kódgenerátor jellemzően több módosítható paraméterrel rendelkezik. Nyissa meg a `hu.bme.mit.yakindu.analysis\model_input\example.sgen` állományt, és adja hozzá az alábbi paramétereket. Amennyiben az egeret az új funkció neve fölé viszi, megjelenik egy dokumentáció. Írja le hogy ezeknek a paramétereknek a bevezetésével milyen változást idézett elő!

```
statechart example {
        …
        feature GeneralFeatures {
                TimerService = true
                RuntimeService = true
        }
}
```

3.2. Tekintse át az alábbi kódrészletet! A kódrészlet megtalálható az alábbi állományban is:

`hu.bme.mit.yakindu.analysis/src/hu/bme/mit/yakindu/analysis/workhere/RunStatechart.java`

A kód az alábbi műveleteket végzi:

- Létrehoz és felparaméterez egy állapotgépet (első három sor).
- Inicializálja az állapotgépet az `init` és `enter` metódusokkal.
- Az állapotgépen tetszőleges sorrendben meghívhatjuk eseményeket a `raise<Esemény neve>()` metódusokkal. Fontos, hogy az események kiváltása után hívjuk meg a `s.runCycle()` függvényt!
- A `s.getSCInterface()` interface-en keresztül lekérdezhetjük az állapotgép változóinak értékét. Például `s.getSCInterface().getWhiteTime()` mutatja a világos játékos hátralévő idejét.

```
public static void main(String[] args) throws IOException {
	ExampleStatemachine s = new ExampleStatemachine();
	s.setTimer(new TimerService());
	RuntimeService.getInstance().registerStatemachine(s, 200);

	s.init();
	s.enter();

	s.runCycle();
	print(s);

	s.raiseStart();
	s.runCycle();

	System.in.read();

	s.raiseWhite();
	s.runCycle();

	print(s);
	System.exit(0);
}

public static void print(IExampleStatemachine s) {
	System.out.println("W = " + s.getSCInterface().getWhiteTime());
	System.out.println("B = " + s.getSCInterface().getBlackTime());
}
```

3.3. Futtassa az alkalmazást az alábbi konfigurációval:
`hu.bme.mit.yakindu.analysis/RunStatechart.launch`

3.4. A kódrészlet alapján készítsen olyan alkalmazást, amely a konzolról beolvas sorokat, és amennyiben a beolvasott szöveg megegyezik egy esemény nevével (`start`, `white` vagy `black`), meghívja az adott eseményt. Minden beolvasott sor után írjuk ki az összes változó értékét!

4. Saját kódgenerátor készítése

4.1. Futtassa a `hu.bme.mit.yakindu.analysis/RunWithBigModel.launch` konfigurációval az alkalmazást, és tekintse át a kirajzolt gráfot! Milyen változást tapasztal?

4.2. Mutassa meg a kirajzolt gráfban, hogy hogy néz ki a whiteTime-=1 kifejezés absztrakt szintaxis fája! A gráfrészletet dokumentálja!

4.3. A 2. ponthoz hasonlóan járja be a modellt, és írja ki az összes belső változó és bemenő esemény nevét! Az eseményekhez definiációjának megtalálásához sokat segíthet a generált gráfnézetet.

4.4. **Saját kódgenerátor írása**: Írjon egy olyan alkalmazást, ami az események és a változók nevét az alábbi formában írja ki:

4.4.1. **Változók**: Írassa ki az állapotgép változóit az alábbi formában (azaz ez a szöveg jelenjen meg a konzolon):

```
public static void print(IExampleStatemachine s) {
System.out.println("W = " + s.getSCInterface().get<Első változó neve>());
      …
System.out.println("B = " + s.getSCInterface().get<Utolsó változó neve>());
}
```

4.4.2. **Események**: Az állapotgép beolvasása után írja ki az eseményeket olyan formában, hogy annak a 3.3-as feladatban használt alkalmazásnak a forráskód-részletét kapjuk, ami a konzolról beolvassa az eseményeket.

4.4.3. **Kódgenerátor**: Mutassa meg, hogy a konzolra kiírt szöveg megegyezik a 3.3-as feladathoz készített forráskód nagy részével.