

Laboration 3

Predikatlogik och substitution

Gör en objektorienterad modell för att representera ett predikatlogiskt uttryck. Modellen skall tillhandahålla en metod för att göra substitutioner och en för att konvertera representationen till en sträng. Alla objekt i modellen skall vara *omuterbara*, dvs det får inte finnas några metoder som modifierar ett befintligt objekt.

Under föreläsning F10 konstruerades en modell för *termer*. Denna finns tillgänglig via hemsidan som ett Java-paket. Den innehåller en `toString`-metod och en metod `collectVariables` som är analog med den du gjort i föregående laboration. Du skall själv komplettera med en metod, `Term substitute(Variable, Term)` som implementerar $t_1[x \setminus t_2]$ från kapitel 3 i Diskreta strukturer.

Du skall själv göra en modell för att representera predikatlogiska uttryck. Modellen skapas i ett eget paket. För att minimera arbetet är det tillräckligt att den kan representera

- predikat, $P(t_1, \dots, t_n)$, där $n \geq 0$ och t_i är termer,
- implikationer $(e_1 \rightarrow e_2)$, där e_1 och e_2 är predikatlogiska uttryck och
- allkvantifieringar, $(\forall x.e)$, där x är en variabel och e är ett predikatlogiskt uttryck.

Du skall implementera `toString()` och `substitute(Variable, Term)` för sådana uttryck. Observera att substitution i $(\forall x.e)[y \setminus t]$ innehåller tre fall:

1. x och y är samma variabel. Då skall man returnera $(\forall x.e)$.
2. x och y är olika variabler och x förekommer inte i t . Då skall man utföra substitutionen $[y \setminus t]$ på e och returnera ett ny allkvantifiering med samma variabel och det nya uttrycket.
3. x och y är olika variabler och x förekommer i t . I detta fall måste man byta ut kvantifieringsvariabeln mot en ny variabel och sedan agera som ovan. Konstrueraren i klassen `term.Variabel` utan parameter genererar en sådan ny variabel. Den nya variabeln skall ersätta x i e .

På hemsidan finns två paket, ett för termer och ett för predikatlogiska uttryck som du skall utgå ifrån. De innehåller testklasser att användas för att testa din implementering. Om du använder samma konstruerare i din implementering så blir det enkelt att använda den.