Assignment 7.1 Linearization I

Consider the following inheritance expressions: \( A(B, C) \quad B(E, G) \quad C(D, F) \quad D(G) \quad F(E) \)

Give the linearization order for \( A \) of the following methods:

1. Leftmost Preorder Depth-First Search with Duplicate Cancelation
2. C3

Suggested Solution 7.1

1. DFS: \( ABEGCDFGE \)
   \( L(A) = ABCDGFE \)
2. \( L(D) = DG, L(F) = FE \)
   \( L(C) = C \cdot [DG] \cdot [FE] \cdot [DF] = CDGFE \)
   \( L(B) = BEG \)
   \( L(A) = A \cdot [BEG] \cdot [CDGFE] \cdot [BC] = ABCD \cdot [EG] \cdot [GFE] \text{ error} \)

Assignment 7.2 Linearization II

Consider the following classes: \( A(B, C) \quad B(D, E) \quad C(F, G) \quad D(G) \quad E(F) \)

Give the linearization order for \( A \) of the following methods:

1. LPDFS with Duplicate Cancellation
2. Reverse Postorder Rightmost DFS
3. C3
Suggested Solution 7.2

1. \( L[A] = ABDECFG \)
   Principles 1 and 2 are satisfied! However, the extension principle (aka monotonicity) is not satisfied: \( L[B] = BDGEF \) from which we derive that \( G \rightarrow F \) holds but this is not the case for \( L[A] \) where \( F \rightarrow G \) holds.

2. Same as for LPDFS with Duplicate Cancellation

3. 
   \[
   \begin{align*}
   L[F] & = F \\
   L[G] & = G \\
   L[E(F)] & = EF \\
   L[D(G)] & = DG \\
   L[C(F, G)] & = C \cdot \bigcup(F, G, FG) = CFG \\
   L[B(D, E)] & = B \cdot \bigcup(DG, EF, DE) = BDGEF \\
   L[A(B, C)] & = A \cdot \bigcup(BDGEF, CFG, BC) = ABDC \cdot \bigcup(GEF, FG) = \text{fail}
   \end{align*}
   \]

Assignment 7.3 (Multiple) Inheritance

1. Consider the following C++-Classes:
   
   ```
   class A { public: int a; virtual void f(); } \\
   class B : public A { public: int b; virtual void f(); } \\
   class C : public B { public: int c; virtual void f(); } 
   ```
   
   Draw a memory representation diagram for a C-Object, and the virtual table diagram for class C!

2. Consider the following C++-Classes:
   
   ```
   class A { public: int a; void f(); } \\
   class B : public A { public: int b; void f(); } \\
   class C { public: int c; void f(); } \\
   class D : public C, public B { public: int d; void f(); } 
   ```
   
   Draw a memory representation diagram for a D-Object!

Suggested Solution 7.3

1. 
Δ^B \{ 
\begin{align*} 
C & \quad \text{int c} \\
A & \quad \text{int a} \\
B & \quad \text{int b} \\
D & \quad \text{int d} 
\end{align*} 
\}