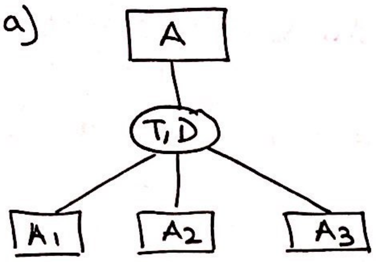
Diseño lógico de Generalización / Especialización.

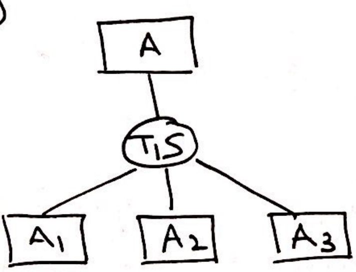
1º Recordemos los tipos de especialización:

a)



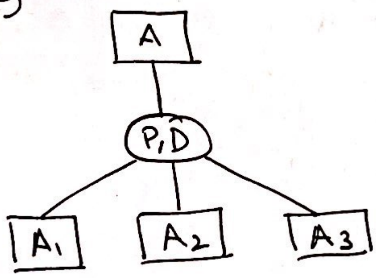
Total, disjunta: A debe especializar siempre. Ninguna ocurrencia de A puede aparecer en subclases distintas.

b)



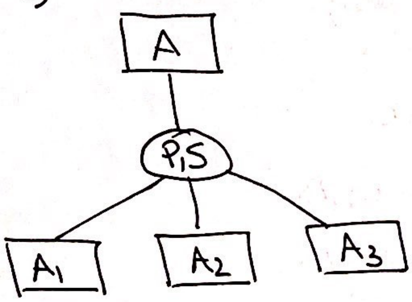
Total, solapada: A debe especializar siempre. Toda ocurrencia de A puede aparecer en subclases distintas

c)



Parcial, disjunta: A no está obligada a especializar, pero si especializa, no puede aparecer en subclases distintas

d)



Parcial, solapada: A no está obligada a especializar, pero si especializa, puede aparecer en subclases distintas.

2º ¿Y cómo se realiza el diseño lógico en estos casos?

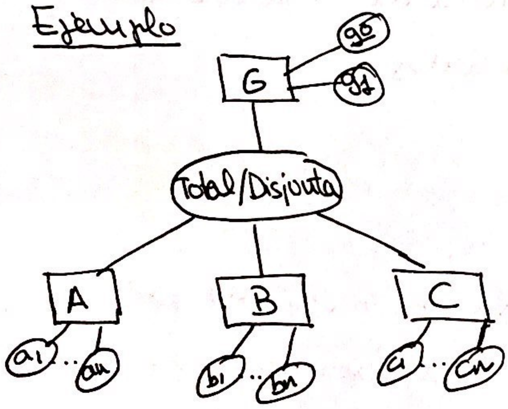
Dependiendo del tipo de especialización nos encontraremos con unas restricciones u otras. En el caso que vamos a poner de ejemplo, la especialización es total y disjunta; por ello se deberán añadir las dos restricciones.

Si la especialización fuera total y solapada, se debería añadir la restricción de total.

Si fuera parcial y disjunta, se debería añadir la restricción de disjunta.

Y si fuera parcial y solapada, no se añadiría nada.

Ejemplo



$G = \underline{g_0} + g_1$

$A = \underline{g_0} + a_1 + \ldots + a_n$
$CAj: \{g_0 \to G\}$

$B = \underline{g_0} + b_1 + \ldots + b_n$
$CAj: \{g_0 \to G\}$

$C = \underline{g_0} + c_1 + \ldots + c_n$
$CAj: \{g_0 \to G\}$

Se indican las restricciones para:

a) Total $\Rightarrow$ $Gx : G \quad Ax : A \quad Bx : B \quad Cx : C$

$$\forall Gx (G(Gx) \to (\exists Ax (A(Ax) \wedge Ax.g_0 = Gx.g_0)) \vee (\exists Bx (B(Bx) \wedge Bx.g_0 = Gx.g_0)) \vee (\exists Cx (C(Cx) \wedge Cx.g_0 = Gx.g_0)))$$

b) Disjunta $\Rightarrow$ $Gx : G \quad Ax : A \quad Bx : B \quad Cx : C$

$$\neg\exists Ax (A(Ax) \wedge \neg\exists Bx (B(Bx) \wedge Bx.g_0 = Ax.g_0)) \wedge$$
$$\neg\exists Ax (A(Ax) \wedge \neg\exists Cx (C(Cx) \wedge Cx.g_0 = Ax.g_0)) \wedge$$
$$\neg\exists Bx (B(Bx) \wedge \neg\exists Cx (C(Cx) \wedge Cx.g_0 = Bx.g_0))$$