

Obecné guardy:

g_1 : isPrimitive = false && ClassSerialization . isTransient = false

g_2 : $\exists i \in \mathbf{N}$: properties [i] . serialization.isID = true && properties [i] .
propertySerialization . IsTransient = false

Case I (třídy bez serializovaného předka)

g_3 : parent = NULL || parent.isAbstract = true || $\forall i \in \mathbf{N}$: \forall předek(i) : předek.isTransient

// předek(i) je třída nacházející se o i úrovní výše v stromu dědičnosti, tzn. Třída, na které je testován guard od ní dědí

// isAbstract nesmi mít neabstract predka, takže se da testovat jen parent

$\omega_1 (\pi_1, g_1 - g_3)$

E e Class

π_1 : E -> { Table, K [Columns], PrimaryKey }

// K [Columns] značí kolekci

// implicitni InheritanceType je Joined

Case II (Třídy s jednoduchou dědičností a jedním typem inheritanceType)

g_5 : parent . inheritanceType = Joined && parent != NULL

g_6 : parent . inheritanceType = SingleTable && parent != NULL

g_7 : parent . inheritanceType = ClassPerTable && parent != NULL

// v podstate by tam mohl byt predek misto parenta pro case, kdy parent nema nadefinovany typ: $\exists i \in \mathbf{N}$: $\forall j \in \mathbf{N}$: $j < i$: předek(j) . inheritanceType = NULL

//E e Class

//K [Columns] značí kolekci

$\omega_2 (\pi_2, g_1 \cup g_2 \cup g_5)$

π_2 : E -> { Table, ForeignKey, K [Columns] }

//guardy = obecné guardy g_1 až g_2 a typ Joined

$\omega_3 (\pi_3, g_1 \cup g_2 \cup g_6)$
 $\pi_3 : E \rightarrow \{ \text{Table}, K[\text{Columns}] \}$
// guardy = obecné guardy g_1 až g_3 a typ SingleTable
// pri InheritanceType SingleTable sice nevznika u serializovatelných potomku nova
Tabulka, ale budeme ji potrebovat
 $\omega_4 (\pi_4, g_1 \cup g_2 \cup g_7)$
 $\pi_4 : E \rightarrow \{ \text{Table}, K[\text{Columns}], \text{PrimaryKey} \}$

=====

Kombinace

$g_8: \exists i \in N: \forall j \in N, j < i: \text{předek}(j) . \text{inheritanceType} = \text{NULL}, (\text{předek}(i) . \text{inheritanceType} = \text{Joined} \parallel \text{předek}(i) . \text{inheritanceType} = \text{TablePerClass}), \text{parent} . \text{InheritanceType} = \text{SingleTable}$

$\omega_5 (\pi_3, g_1 \cup g_2 \cup g_8)$
 $\pi_3 : E \rightarrow \{ \text{Table}, K[\text{Columns}], \text{PK} \}$
// Joined v TablePerClass => SingleTable

$g_9: \exists i \in N: \forall j \in N, j < i: \text{předek}(j) . \text{inheritanceType} = \text{NULL}, (\text{předek}(i) . \text{inheritanceType} = \text{TablePerClass} \parallel \text{předek}(i) . \text{inheritanceType} = \text{SingleTable}), \text{self} . \text{InheritanceType} = \text{Joined}$

$\omega_6 (\pi_1, g_1 \cup g_2 \cup g_8)$
 $\pi_1 : E \rightarrow \{ \text{Table}, K[\text{Columns}] \}$
// SingleTable v TablePerClass => Joined

$g_{10}: \exists i \in N: \forall j \in N, j < i: \text{předek}(j) . \text{inheritanceType} = \text{NULL}, (\text{předek}(i) . \text{inheritanceType} = \text{Joined} \parallel \text{předek}(i) . \text{inheritanceType} = \text{SingleTable}), \text{self} . \text{InheritanceType} = \text{TablePerClass}$

$\omega_7 (\pi_1, g_1 \cup g_2 \cup g_9)$
 $\pi_2 : E \rightarrow \{ \text{Table}, \text{ForeignKey}, K[\text{Columns}] \}$