```
theory Diff_Arr_Rel
  imports Cell
begin


fun diff_arr_rel' ("(_ ⊢ _ ∼⇩_ _)" [51, 51, 51, 51] 50)  where
  "diff_arr_rel' t xs 0 a ⟷ (a, Array' xs) ∈L t"
| "diff_arr_rel' t xs (Suc n) a ⟷ (∃i x a' xs'.
        (a, Upd' i x a') ∈L t
      ∧ diff_arr_rel' t xs' n a'
      ∧ xs = xs'[i:=x]
      ∧ i < length xs'
)"


definition diff_arr_rel ("(_ ⊢ _ ∼ _)" [51, 51, 51] 50) where
  "diff_arr_rel t xs a ≡ ∃n. t ⊢ xs ∼⇩n a"

lemma diff_arr_rel'_cons: "t ⊢ xs ∼⇩n diff_arr ⟹ x # t ⊢ xs ∼⇩n diff_arr"
proof(induction t xs n diff_arr rule: diff_arr_rel'.induct)
  case 1
  then show ?case by auto
next
  case (2 t xs n a)
  then show ?case
    apply auto
    subgoal for i v a' xs'
      apply(rule exI[where x = "i"])
      apply(rule exI[where x = "v"])
      apply(rule exI[where x = "a'"])
      by auto
    done
qed

lemma diff_arr_rel'_cons': "t ⊢ xs ∼⇩n diff_arr ⟹ ∃n. x # t ⊢ xs ∼⇩n diff_arr"
  using diff_arr_rel'_cons
  by blast

lemma diff_arr_rel_cons: "t ⊢ xs ∼ diff_arr ⟹ x # t ⊢ xs ∼ diff_arr"
  unfolding diff_arr_rel_def
  using diff_arr_rel'_cons by blast

end
```