JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/ClauseCoordinationRule.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/aggregation/BackwardConjunctionReductionRule.html)
* [Next Class](http://docs.google.com/simplenlg/aggregation/ForwardConjunctionReductionRule.html)
* [Frames](http://docs.google.com/index.html?simplenlg/aggregation/ClauseCoordinationRule.html)
* [No Frames](http://docs.google.com/ClauseCoordinationRule.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.2s8eyo1)

simplenlg.aggregation

## Class ClauseCoordinationRule

* java.lang.Object
  + [simplenlg.aggregation.AggregationRule](http://docs.google.com/simplenlg/aggregation/AggregationRule.html)
    - simplenlg.aggregation.ClauseCoordinationRule
* public class ClauseCoordinationRule  
  extends [AggregationRule](http://docs.google.com/simplenlg/aggregation/AggregationRule.html)
* Implementation of a clausal coordination rule. The rule performs the following operations on sentences:
  1. If the sentences have the same subject, a new sentence is returned with that subject, and the VP from the component sentences conjoined. For example *John kicked the ball.* and *John sang a song.* becomes *John kicked the ball and sang a song*.
  2. If the sentences have the same VP, a new sentence is returned with that VP, and the subjects from the component sentences conjoined. For example *John kicked the ball.* and *Mary kicked the ball.* become *John and Mary kicked the ball*.

These operations only apply to sentences whose front modifiers are identical, that is, sentences where, for every pair s1 and s2, s1.getFrontModifiers().equals(s2.getFrontModifiers()).  
**Note:**: it is not recommended to use this rule in addition to [BackwardConjunctionReductionRule](http://docs.google.com/simplenlg/aggregation/BackwardConjunctionReductionRule.html) and/or [ForwardConjunctionReductionRule](http://docs.google.com/simplenlg/aggregation/ForwardConjunctionReductionRule.html).Author: Albert Gatt, University of Malta & University of Aberdeen

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**ClauseCoordinationRule**](http://docs.google.com/simplenlg/aggregation/ClauseCoordinationRule.html#ClauseCoordinationRule())() Constructs an instance of the ClauseCoordinationRule |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**apply**](http://docs.google.com/simplenlg/aggregation/ClauseCoordinationRule.html#apply(simplenlg.framework.NLGElement,%20simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) previous, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) next) Applies aggregation to two NLGElements e1 and e2, succeeding only if they are clauses (that is, e1.getCategory() == e2.getCategory == [PhraseCategory.CLAUSE](http://docs.google.com/simplenlg/framework/PhraseCategory.html#CLAUSE)). |

### Methods inherited from class simplenlg.aggregation.[**AggregationRule**](http://docs.google.com/simplenlg/aggregation/AggregationRule.html)[apply](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(java.util.List)), [apply](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(simplenlg.framework.NLGElement)), [getFactory](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#getFactory()), [setFactory](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#setFactory(simplenlg.framework.NLGFactory))

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### ClauseCoordinationRule public ClauseCoordinationRule() Constructs an instance of the ClauseCoordinationRule

### Method Detail

#### apply public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) apply([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) previous, [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) next) Applies aggregation to two NLGElements e1 and e2, succeeding only if they are clauses (that is, e1.getCategory() == e2.getCategory == [PhraseCategory.CLAUSE](http://docs.google.com/simplenlg/framework/PhraseCategory.html#CLAUSE)).**Specified by:** [apply](http://docs.google.com/simplenlg/aggregation/AggregationRule.html#apply(simplenlg.framework.NLGElement,%20simplenlg.framework.NLGElement)) in class [AggregationRule](http://docs.google.com/simplenlg/aggregation/AggregationRule.html) Parameters:previous - the first sentencenext - the second sentence Returns:an aggregated sentence, if the method succeeds, null otherwise

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/ClauseCoordinationRule.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/aggregation/BackwardConjunctionReductionRule.html)
* [Next Class](http://docs.google.com/simplenlg/aggregation/ForwardConjunctionReductionRule.html)
* [Frames](http://docs.google.com/index.html?simplenlg/aggregation/ClauseCoordinationRule.html)
* [No Frames](http://docs.google.com/ClauseCoordinationRule.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.2s8eyo1)