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/ForwardConjunctionReductionRule.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/ClauseCoordinationRule.html)
* [Next Class](http://docs.google.com/simplenlg/aggregation/FunctionalSet.html)
* [Frames](http://docs.google.com/index.html?simplenlg/aggregation/ForwardConjunctionReductionRule.html)
* [No Frames](http://docs.google.com/ForwardConjunctionReductionRule.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 ForwardConjunctionReductionRule

* java.lang.Object
  + [simplenlg.aggregation.AggregationRule](http://docs.google.com/simplenlg/aggregation/AggregationRule.html)
    - simplenlg.aggregation.ForwardConjunctionReductionRule
* public class ForwardConjunctionReductionRule  
  extends [AggregationRule](http://docs.google.com/simplenlg/aggregation/AggregationRule.html)  
  Implementation of the forward conjunction rule. Given two sentences s1 and s2, this rule elides any constituent in the left periphery of s2 which is also in s1. The left periphery in simplenlg is roughly defined as the subjects, front modifiers and cue phrase of an [SPhraseSpec](http://docs.google.com/simplenlg/phrasespec/SPhraseSpec.html).  
  This rule elides any constituent in the left periphery of s2 which is *lemma-identical* to a constituent with the same function in s1, that is, it is headed by the same lexical item, though possibly with different inflectional features and/or modifiers. Note that this means that *the tall man* and *the man* are counted as "identical" for the purposes of this rule. This is only justifiable insofar as the two NPs are co-referential. Since SimpleNLG does not make assumptions about the referentiality (or any part of the semantics) of phrases, it is up to the developer to ensure that this is always the case.  
  The current implementation is loosely based on the algorithm in Harbusch and Kempen (2009), which is described here: <http://aclweb.org/anthology-new/W/W09/W09-0624.pdf>  
  **Implementation note:** The current implementation only applies ellipsis to phrasal constituents (i.e. not to their component lexical items).  
  **Note:**: this rule can be used in conjunction with the [BackwardConjunctionReductionRule](http://docs.google.com/simplenlg/aggregation/BackwardConjunctionReductionRule.html) in [Aggregator](http://docs.google.com/simplenlg/aggregation/Aggregator.html).Author: agatt

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**ForwardConjunctionReductionRule**](http://docs.google.com/simplenlg/aggregation/ForwardConjunctionReductionRule.html#ForwardConjunctionReductionRule())() Creates a new ForwardConjunctionReduction. |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**apply**](http://docs.google.com/simplenlg/aggregation/ForwardConjunctionReductionRule.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 forward conjunction reduction 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)) and the clauses are not passive. |

### 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

#### ForwardConjunctionReductionRule public ForwardConjunctionReductionRule() Creates a new ForwardConjunctionReduction.

### 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 forward conjunction reduction 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)) and the clauses are not passive.**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 phrasenext - the second phrase Returns:a coordinate phrase if aggregation is successful, 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/ForwardConjunctionReductionRule.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/ClauseCoordinationRule.html)
* [Next Class](http://docs.google.com/simplenlg/aggregation/FunctionalSet.html)
* [Frames](http://docs.google.com/index.html?simplenlg/aggregation/ForwardConjunctionReductionRule.html)
* [No Frames](http://docs.google.com/ForwardConjunctionReductionRule.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)