# Überapproximation

Generelles Ziel: Erkenntnis zur Überapproximation in den verschiedenen Regelsätzen gewinnen!

Vorgehen:

* Zwei Möglichkeiten:
  + Regel „on demand“ innerhalb des Programmcodes anpassen
    - Gefahr: Eine Kontrolle der „preserve“ Regeln ist nicht mehr möglich
  + Regeln in extra Dateiverzeichnis (“preserve“) zwischenspeichern
    - Nachteil: ansonsten überflüssige Dateien werden erzeugt
* Bericht erstellen
  + Ergebnisse (Overapproximation)
  + Berechnungsgrundlage (ess.CPA oder CR)
  + Berechnungsdauer

Implementierungsziele:

* Zu jeder Messung eine ausführbare Datei vorliegen haben und somit Ergebnisse reproduzieren können.
* Möglichst vollautomatischer Prozess im Hinblick auf Gewinnung der Auswertungen.

Auf Grundlage welches mathematischen Operators soll die Überapproximation berechnet werden?

Vermutliche eher auf Basis der Differenz (-) als auf Basis des Quotienten

## DONE @ XLS writer:

* Add metric values on each sheet!
* Build **Overapproximation** sheet!
* Calculate values @ overapproximation sheet

## Problem beim Exportieren von UML Regeln zu AGG:

Es gibt Vererbungszyklen im Metamodell. (Sowohl programmatisch herausgefunden, als auch bekannte Aussage von Siegener Kollegen)

Warum ist das ein Problem?

Der HenshinAGGExporter geht beim Exportieren so vor, dass er mit einer exportierten EClass auch die Referenzen zu allen super classes exportiert. Bei einem Vererbungszyklus ist dies nicht möglich.

## Lösungsvorschlag:

Umbauen des Exporters, sodass dieser Vererbungszyklen unterstützt.!

Dazu zuerst die EClasses mit allem wie bisher exportieren und im Anschluss die Vererbungen behandeln. (Dann sind alle super classes vorhanden!)

Dazu:

* ein kleines Beispiel aufsetzen zum Ausprobieren.
* Erneutes Ausführen der overapproximation Runner und prüfen, dass die Ergebnisse mit dem neuen Exporter nicht abweichen.
* Ausführen der CPA Tests (in org.eclipse.emf.henshin.tests.cpa)

## Gespräch mit Nebras:

completeness – Correctness

Ein Ansatz mit der Overapproximation umzugehen wäre es alle Ergebnisse bei denen ein löschender Knoten der zweiten Regel überlappt zu ignorieren. Dies würde jedes Ergebnis, dass zu einer dangling edge führen kann ignorieren.

Allerdings würden dabei auch valide Ergebnisse verloren gehen. Der Vorteil ist, dass alle Ergebnisse die gemeldet werden korrekt sind. Der Nachteil ist, dass nicht alle Ergebnisse in der Ergebnismenge enthalten sind!