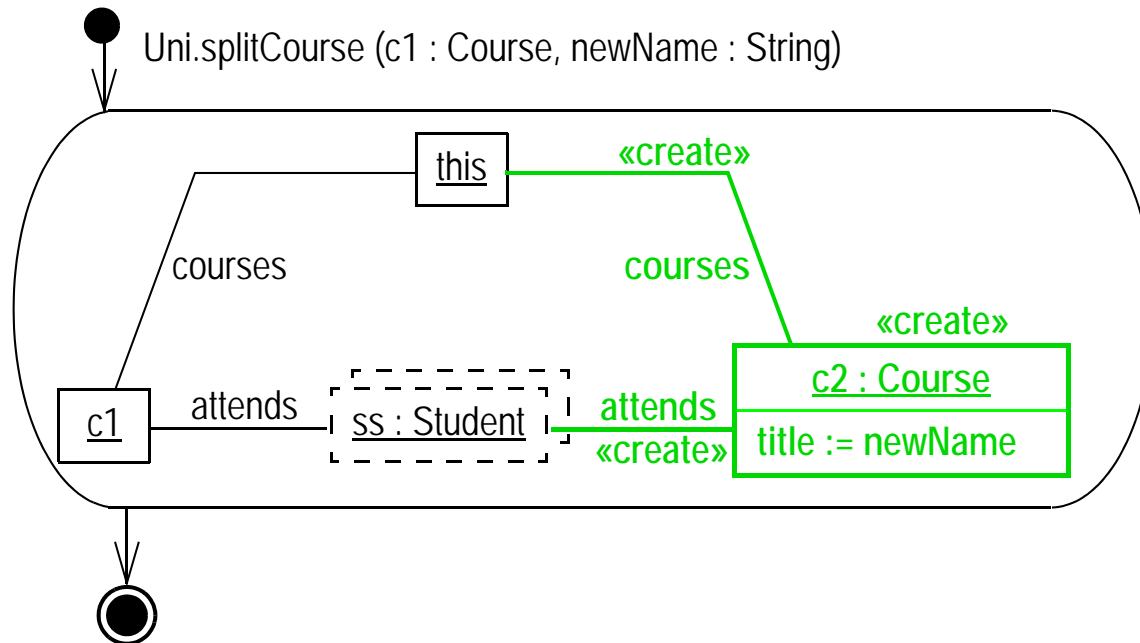
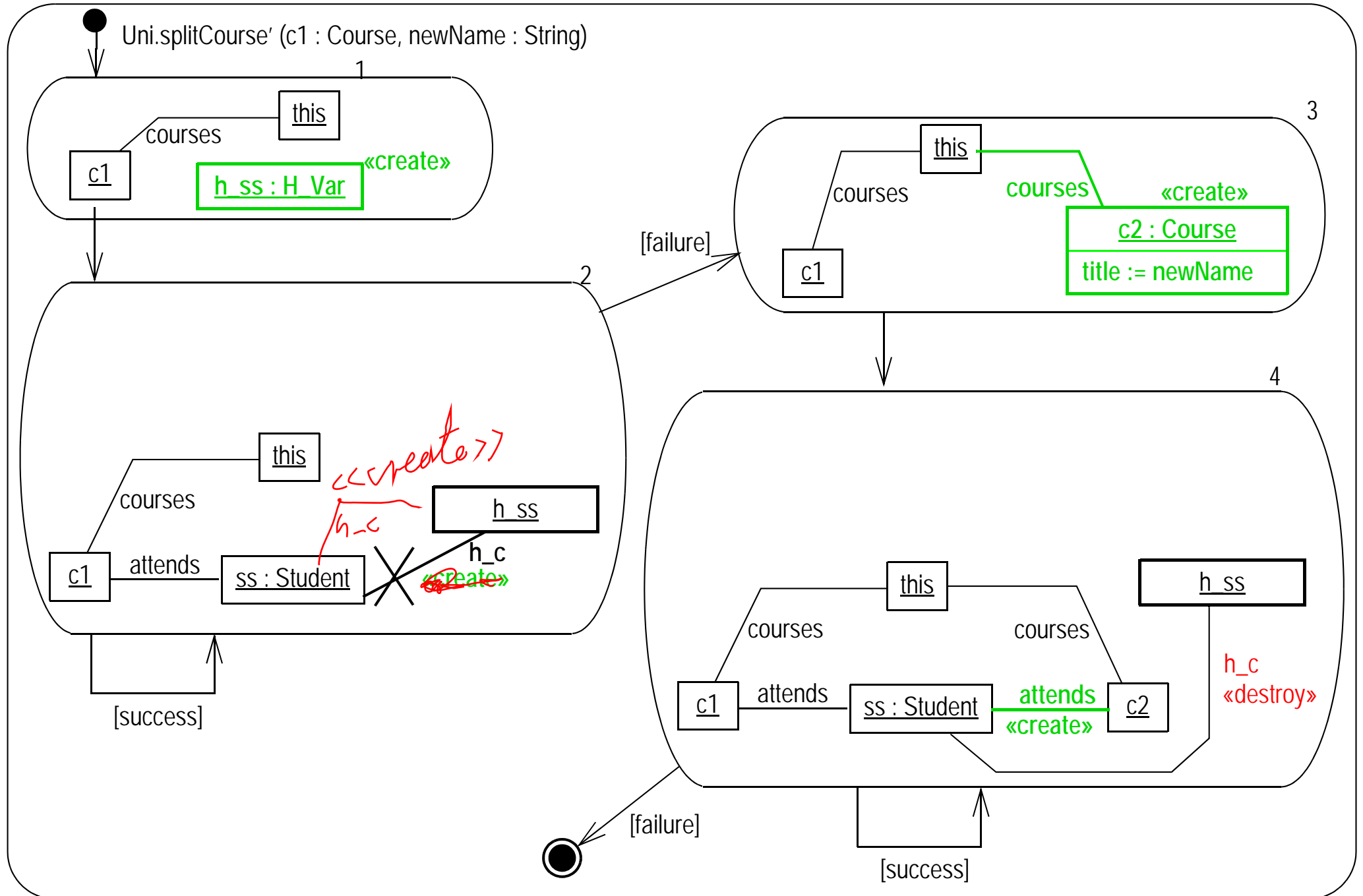


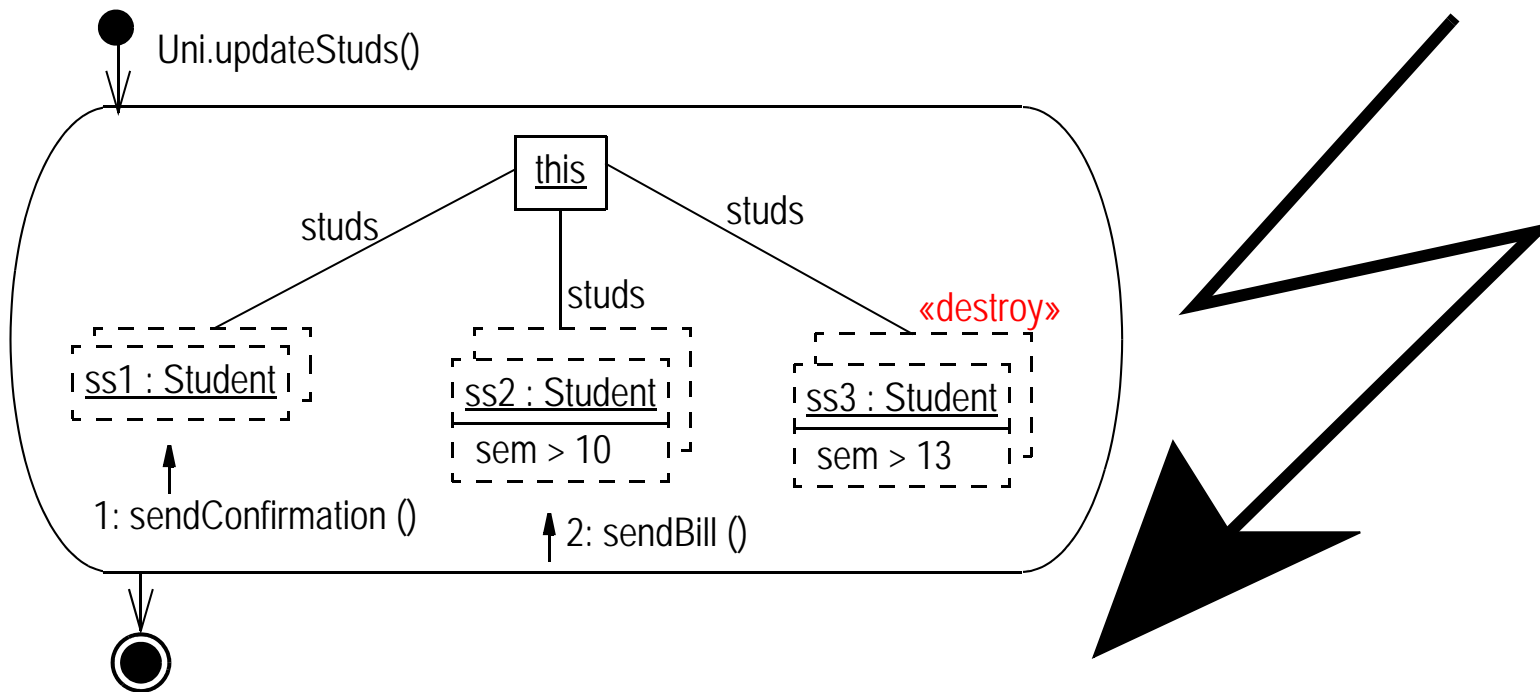
## Mengenknoten



- 1: suche Kern der Regelanwendung
- 2: markiere alle möglichen Matches für den Mengenknoten
- 3: führe Kern der Regel aus
- 4: für Effekte des Mengenknotens für alle markierten Matches aus

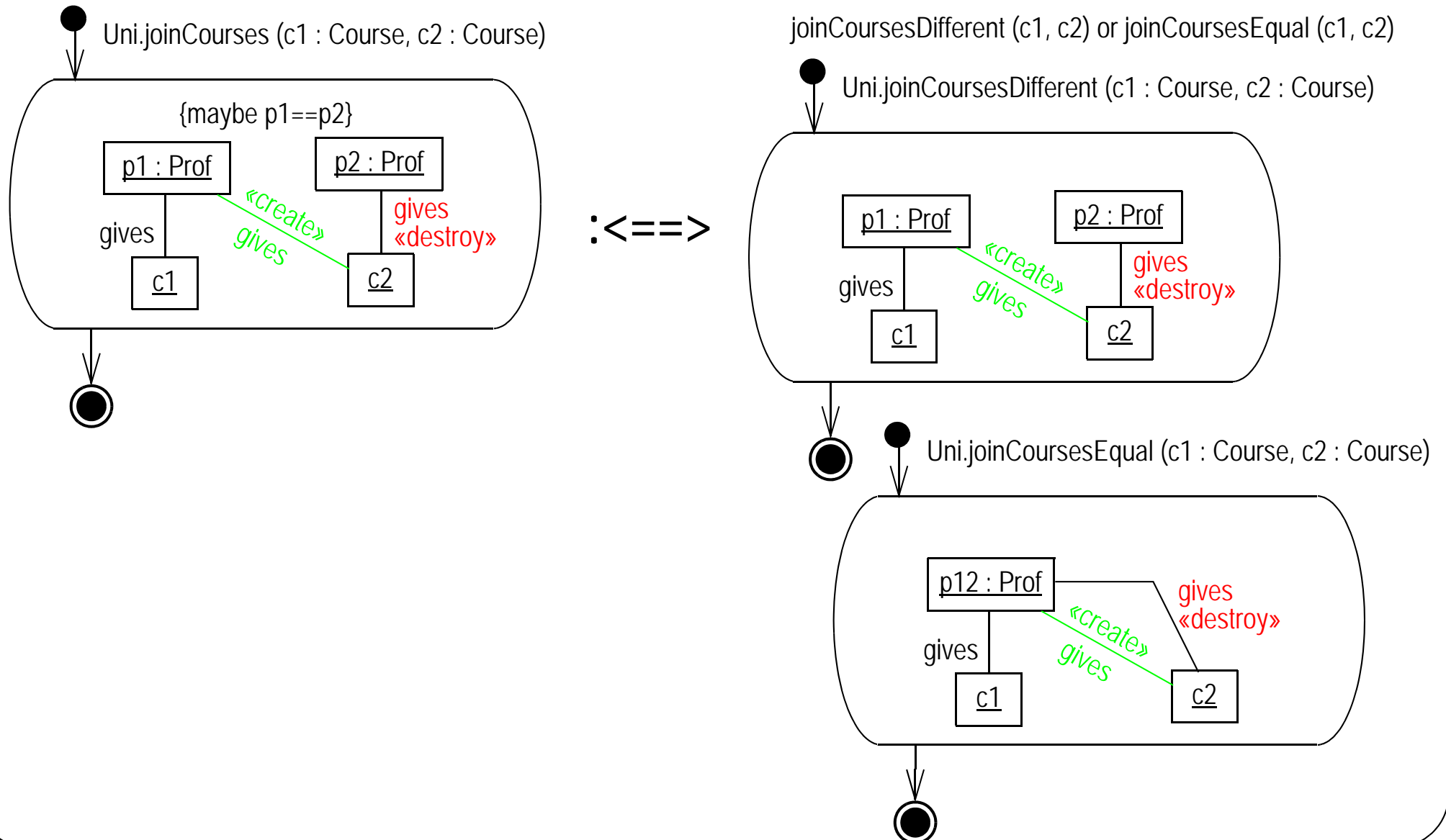


## mehrere Mengenknoten

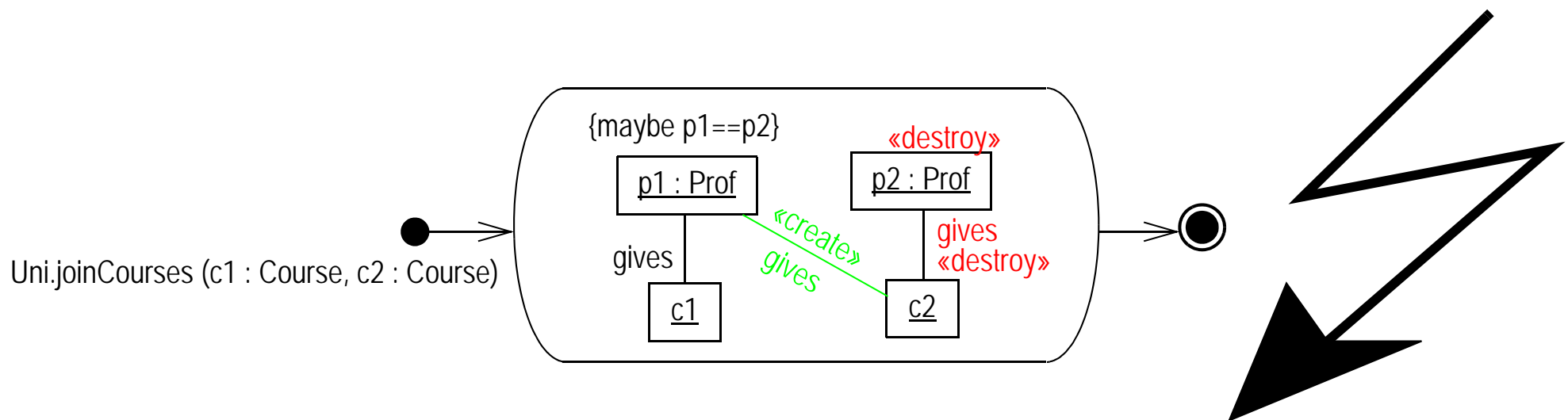


- Aufteilung der Studenten ist nicht eindeutig / zufällig => Effekt der Regel wäre zufällig
- Daher: Kein Knoten darf zur Laufzeit mehreren Mengenknoten zugeordnet werden können
- Sonst Laufzeitfehler

## Maybe-Klausel (nicht isomorphe Teilgraphensuche):

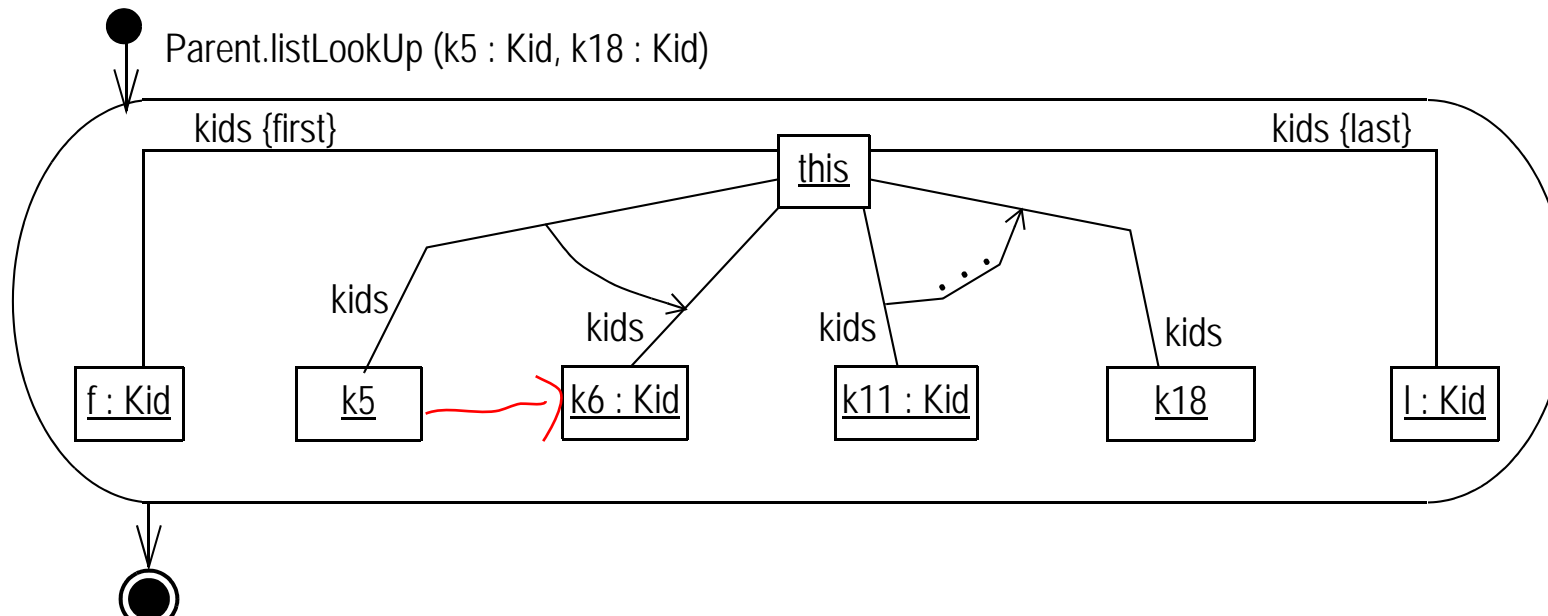


- mehr als 2 Knoten in einer Maybe-Klausel möglich
- mehrere Maybe-Klauseln oder mehrere Äquivalenzklassen in einem Story Pattern gehen:  
{ maybe a==b==c, d==e==f }
- ein Knoten kann in mehreren Maybe-Klauseln auftauchen
- in einer Maybe-Klausel können Knoten unterschiedlicher Kardinalität enthalten sein  
{ maybe a in s }
- die Knoten einer Maybe-Klausel müssen gleich (kompatiblen) Typ haben
- die Knoten einer Maybe-Klausel müssen bezüglich Löschen/Ersetzen gleich behandelt werden  
(identification condition):

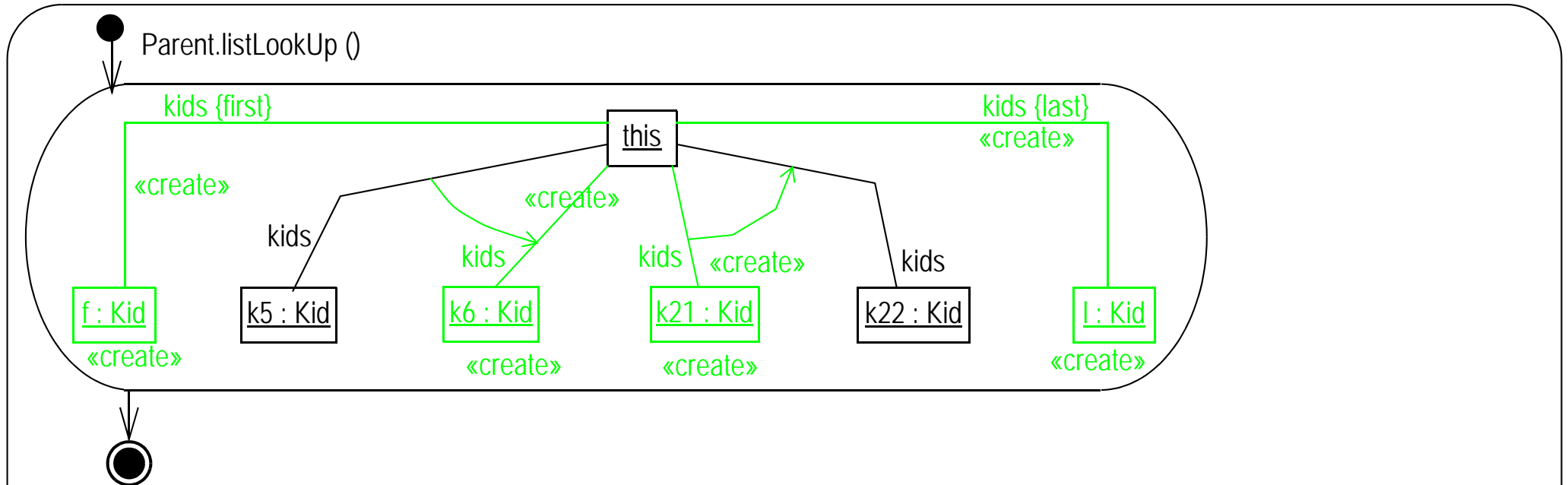




## Behandlung von Listen / geordneten oder sortierten Assoziationen:



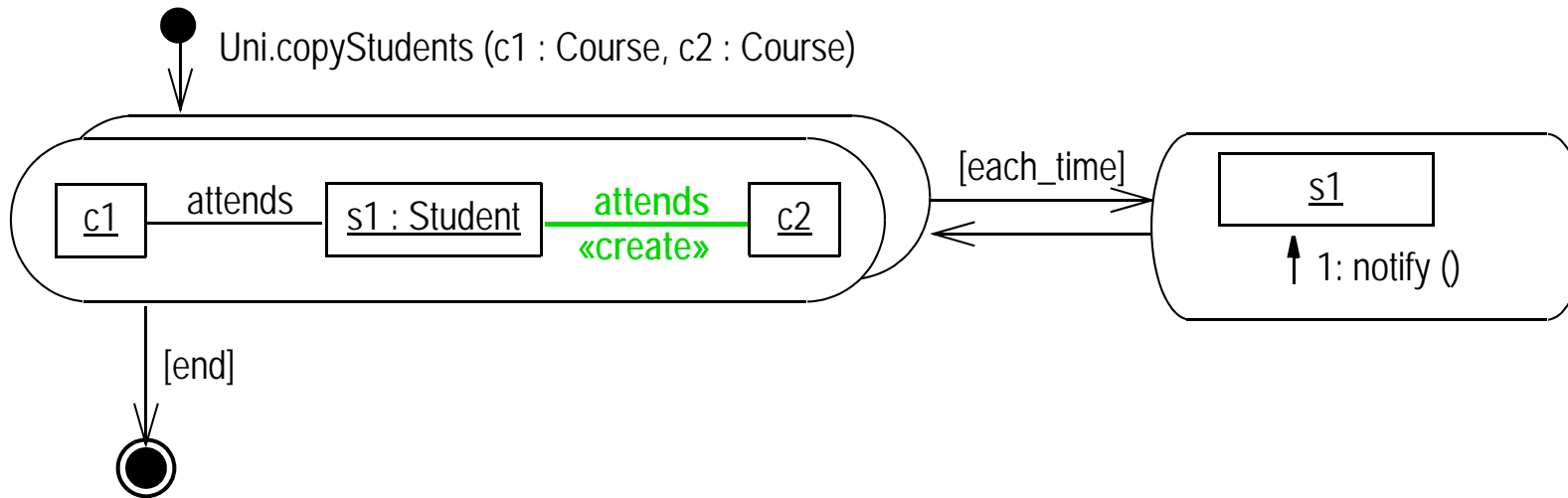
- {first} sucht erstes Element der Assoc
- {last} sucht letztes Element der Assoc
- $\longrightarrow$  zwischen zwei Links kennzeichnet direkt aufeinanderfolgende Elemente (next)
- $\cdots\longrightarrow$  zwischen zwei Links kennzeichnet indirekt aufeinanderfolgende Elemente (next \*)



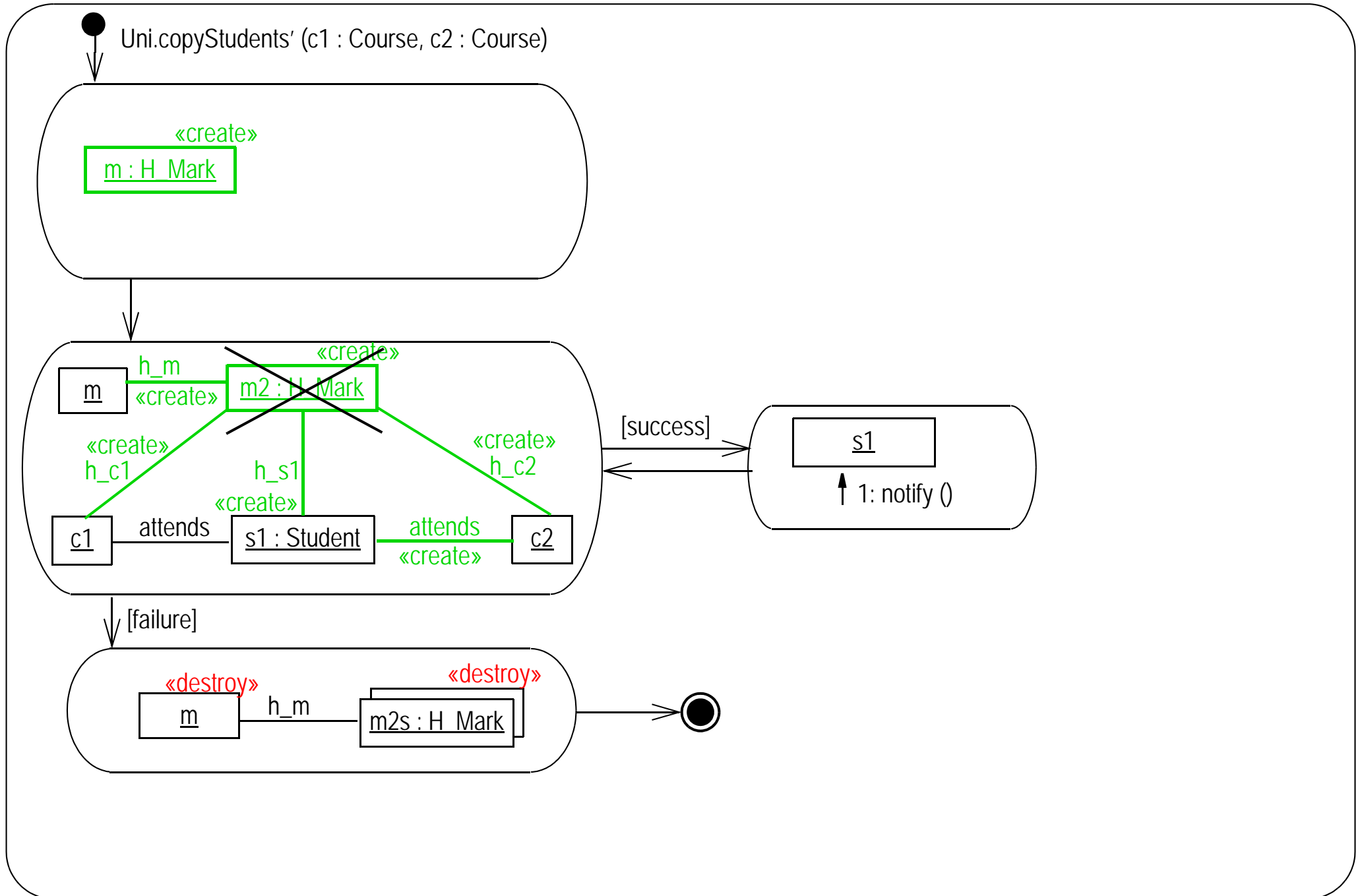
Beim Einfügen in Listen kann die Einfügeposition bezeichnet werden:

- `«create» {first}` Einfügen vorne
- `«create» {last}` Einfügen hinten
- `«create»`  $\longrightarrow$  Einfügen direkt vor oder nach gegebenem anderen Element

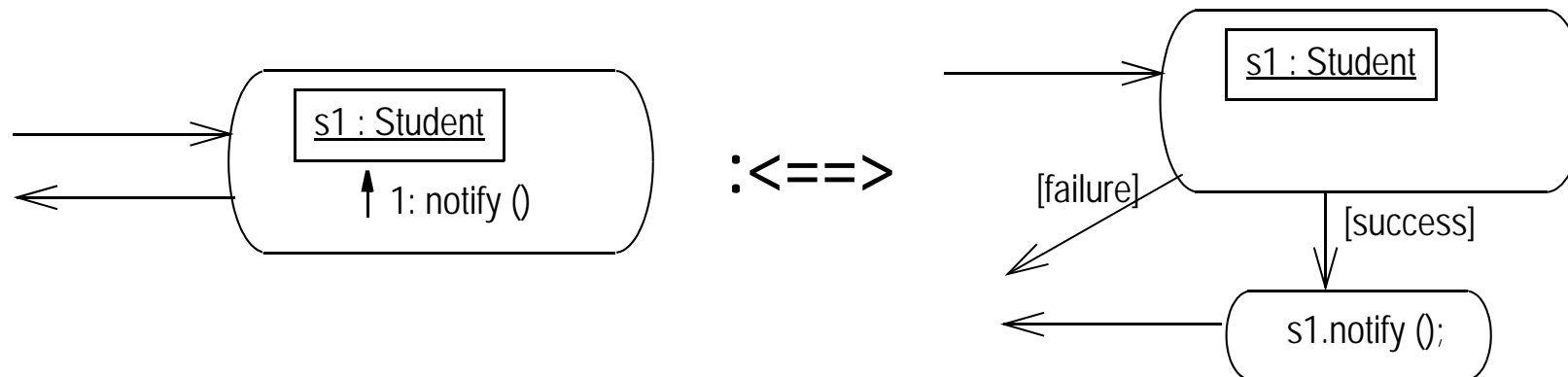


**For-All-Patterns:**

- bei jeder Ausführung:
  - markiere schon benutzte Stellen bei der Ausführung
  - stelle sicher, dass neue Stelle erwischt wurde
- am Schluss alle Markierungen entfernen



## Collaborations-Nachrichten:



- geschachtelte-Nummerierung: 1, 2, 2.1, 2.2, 3, 3.1, 3.1.1, ...
- if-Statement: 4 [<boole expr>]: , 4.1, 4.2, ...
- (else fehlt)
- for-Schleife: 5 [ i = 1..100]: ,...
- while-Schleife 6 [while <bool expr>]: , ...
- wird auf entsprechende Java-Statements abgebildet

# Story Diagrams Complete Syntax:

