

Wissensvermittlungen im Drama annotieren

Annotationsguideline

Q:TRACK

Melanie Andresen
Benjamin Krautter
Janis Pagel
Nils Reiter

Version 1.0, November 2021

Inhaltsverzeichnis

1	Einleitung	2
2	Annotationstool	2
3	Allgemeines	2
4	Markables	2
5	Tagstruktur	2
6	Figuren	3
7	Figurenrelationen	5
7.1	Gerichtete Relationen	5
7.2	Ungerichtete Relationen	5
7.3	Erläuterung von Relationstags	6
8	Wissen über Wissen	7
9	Sonstiges	7
9.1	Bruder, Oheim & Co.	7
9.2	Automatische Schlüsse	8
10	Zweifelsfälle	8
10.1	Abweichende Wissensbestände	8
10.1.1	Unterschiedliche Empfänger:innen haben unterschiedliches Wissen	8
10.1.2	Sender:in und Empfänger:in haben unterschiedliches Wissen	8
10.2	Unklare Identitäten	8
10.2.1	Hofmannsthals Elektra	8
10.2.2	Kleists Familie Schroffenstein	9
10.2.3	Schillers Braut von Messina	9
10.2.4	Grillparzers Ahnfrau	10
10.2.5	Kriterien zur Vergabe der DraCor-ID (Identität vs. Name)	10
11	Annotationen finalisieren	10

1 Einleitung

Das Projekt Q:TRACK befasst sich mit der Wissensvermittlung in deutschen Dramen. Genauer geht es um die Weitergabe von Wissen über Familienrelationen: Wann sind den Figuren und dem Publikum welche Familienrelationen bekannt? Ein entscheidender Moment für den Weitergang der Handlung und die erhoffte Wirkung eines Dramas besteht oftmals darin, dass eine Figur von einer ihr bisher unbekanntem Familienrelation erfährt. Ein prominentes Beispiel ist die Szene, in der Ödipus (Sophokles: *König Ödipus*) aufdeckt, dass er seinen Vater getötet und seine Mutter geheiratet hat. Um diese Wissensweitergabe zu modellieren, benötigen wir manuell erstellte Annotationen. In einem ersten Schritt wollen wir Textstellen identifizieren, an denen Wissen über Figurenrelationen vermittelt wird. Wie wir dabei vorgehen, wird im Folgenden erläutert.

2 Annotationstool

Als Annotationstool verwenden wir den CorefAnnotator, den man hier herunterladen kann: <https://github.com/nilsreiter/CorefAnnotator/>. Wir nutzen dafür immer die Version, die als „Latest Release“ markiert ist. Insbesondere die Beta-Versionen nutzen wir für die Annotation nicht. Wenn eine neue Version verfügbar ist, wird auf der Startansicht des CorefAnnotators darauf hingewiesen.

3 Allgemeines

Wir annotieren den Text des Dramas als Lesetext. Das Wissen aus der Figurentafel steht den Leser:innen also zur Verfügung und auch die Namen der Sprecher:innen werden als bekannt angenommen.

4 Markables

Annotiert werden alle Textstellen, an denen sich der Wissensstand über familiäre Figurenrelationen einer der beteiligten Figuren und/oder des Publikums verändert ODER an denen sich der Wissensstand über den Wissensstand einer (anderen) Figur ändert (z. B. wenn das Publikum erfährt, dass A weiß, dass B und C Schwestern sind).

Um präzise bestimmen zu können, an welcher Stelle die Wissensvermittlung stattfindet, sollen möglichst kurze Textstellen annotiert werden. Oft handelt es sich dabei um Sätze oder Teilsätze. Die Wissensvermittlung lässt sich jedoch nicht immer an einen einzelnen Satz binden. Wo sich dies nicht vermeiden lässt, können deshalb auch längere Passagen mit Redeanteilen mehrerer Figuren annotiert werden.

5 Tagstruktur

Die annotierten Textstellen werden mit Tags versehen, die die folgenden Informationen zur Wissensvermittlung formalisieren:

1. Wer (oder was) ist die QUELLE des Wissens?
2. Wer ist das ZIEL des Wissens?
3. Welches WISSEN wird vermittelt?
4. optional: ATTRIBUTE, z. B. um Lügen zu markieren

Die Tags enthalten diese Informationen in einer festen Reihenfolge:

```
transfer(QUELLE, ZIEL, WISSEN, ATTRIBUTE)
```

Beispiel:

```
transfer(nathan, saladin, siblings(tempelherr, recha))
```

D. h.: Nathan berichtet Saladin hier, dass der Tempelherr und Recha Geschwister sind. Die einzelnen Komponenten des Beispiels werden im Folgenden näher erläutert.

QUELLE. Die häufigste Quelle von weitergegebenem Wissen sind andere Figuren, die die Information verbal vermitteln. In diesem Fall wird das Argument mit der ID der Figur gefüllt (siehe Abschnitt 6). Die Figur, die die Quelle des Wissens ist, muss nicht unbedingt in der Szene anwesend sein, wenn das Wissen etwa durch einen Brief medial vermittelt wird. Bei Botenberichten ist im Normalfall die:der Sender:in des Boten als Quelle des Wissens zu annotieren. Ausnahmen sind möglich, wenn die Boten die Weitergabe des Wissens selbst entscheidend beeinflussen, also beispielsweise die Nachricht verfälschen oder eigene Interessen damit verfolgen.

In manchen Fällen stammt das Wissen nicht von einer anderen Figur, sondern wird durch ein Objekt oder eine Beobachtung ausgelöst. Für diese Art der Informationsquelle ist keine Formalisierung vorgesehen, sie kann deshalb in Anführungsstrichen frei formuliert werden, z. B.:

```
transfer("Handschrift in Buch", saladin, parent_of("Assad", tempelherr))
```

D. h.: Aus der Handschrift im Buch leitet Saladin ab, dass Assad der Vater des Tempelherrn ist. Wird als Quelle eine Figur angegeben, so bedeutet dies, dass die Figur das Wissen verbal geäußert hat. Non-verbale Wissensvermittlungen werden ebenfalls umschrieben, z. B.: "optische Ähnlichkeit mit Assad".

ZIEL. Als Ziel des weitergegebenen Wissens kommen andere Figuren des Dramas oder das Publikum in Frage. Zu ihrer Formalisierung bei der Annotation siehe Abschnitt 6. Voraussetzung für die Annotation als Ziel der Wissensvermittlung ist es, dass die entsprechende Information von der Figur (oder den Figuren) auch verstanden wurde.

WISSEN. Gegenstand des für unser Annotationsprojekt relevanten Wissens sind Familien- und Liebesrelationen zwischen den Dramenfiguren. Abschnitt 7 führt aus, welche Relationen auf welche Weise annotiert werden. Außerdem annotieren wir auf einer Metaebene das Wissen über das Wissen anderer Figuren, siehe Abschnitt 8.

ATTRIBUTE. In zusätzlichen Attributen können wir bedeutsame Merkmale der Wissensvermittlung vermerken. Attribute sind optional. Folgende Attribute sind vorgesehen:

- **lie:** Mit diesem Attribut werden Lügen markiert. Die:Der Sender:in vermittelt in diesem Fall eine Information, die nicht mit ihrem:seinem eigenen Wissensstand übereinstimmt. Mit dem Attribut **lie** wird eine Aussage über die **QUELLE** des Wissens getroffen.
- **uncertain:** Mit diesem Attribut kann ausgedrückt werden, dass das vermittelte Wissen nicht gesichert ist, es sich also beispielsweise nur um eine Vermutung handelt.

```
transfer(tempelherr, nathan, relatives(tempelherr, recha), uncertain)
```

Eine Aussage des Tempelherrn weckt in Nathan hier die Vermutung, dass der Tempelherr und Recha verwandt sein könnten. Mit diesem Attribut wird eine Aussage über das **ZIEL** des Wissens getroffen.

- **already-known:** Mit diesem Attribut wird markiert, dass nur die Quelle des Wissens der Ansicht ist, dass sie neues Wissen vermittelt, die vermittelte Information dem Ziel aber bereits bekannt ist.

```
transfer(appiani, marinelli, engaged(appiani, emilia), already-known)
```

Appiani informiert Marinelli, dass er mit Emilia verlobt ist. Marinelli ist diese Information aber bereits bekannt.

Ergänzungen der Liste im weiteren Verlauf des Annotationsprojekts sind möglich.

6 Figuren

Figuren werden in den Tags durch ihre DraCor-ID repräsentiert. Diese kann auf der Webseite von GerDraCor nachgeschlagen werden: <https://dracor.org/ger>. Im TEI-XML-Dokument zu jedem Drama werden die IDs am Anfang des Dokuments definiert. Alternativ ist zu jedem Drama auch eine Figurenliste (List of characters) als Download verfügbar.

DraCor-IDs stehen nur für die Figuren zur Verfügung, die im Drama auch selbst auftreten. Figuren, über die nur gesprochen wird, haben keine DraCor-ID. Für diese Figuren vergeben wir eigene IDs, die im Normalfall den am häufigsten für sie verwendeten Namen entsprechen. Von uns vergebene Figuren-IDs, die nicht in den DraCor-Daten enthalten sind, setzen wir in Anführungszeichen, also `recha` und `nathan`, aber "Assad".

Darüber hinaus kommen Erwähnungen von Figuren ohne Namen vor. Wenn sie etwa nur über ihre Relation zu einer anderen Figur erwähnt werden. Beispielsweise gibt Nathan zunächst nur bekannt, dass Recha einen Bruder hat:¹

NATHAN.

Besonders hat ein Bruder sich gefunden,

Bei dem Ihr um sie werben müßt.

→ `transfer(nathan, tempelherr, siblings(recha, SIBLING_OF[RECHA]))`

In solchen Fällen verwenden wir eine Variable, die – wie im Beispiel – die entsprechende Relation ausdrückt. Diese Variable wird in Großbuchstaben geschrieben. Sollte es in einem Drama mehrere Figuren geben, die über die gleiche Relation eingeführt werden, dann werden die Variablen zur Disambiguierung nummeriert. In *Die natürliche Tochter* von Goethe werden beispielsweise zwei Kinder des Herzogs zunächst ohne Namen eingeführt. Diese bezeichnen wir als `CHILD-1[HERZOG]` und `CHILD-2[HERZOG]`.

Im weiteren Verlauf des Textes wird die Identität der Figuren häufig aufgelöst. In *Nathan der Weise* stellt sich kurz nach der zitierten Textstelle heraus, dass es sich bei Rechas Bruder um den Tempelherrn handelt. Diese Gleichsetzung wird anschließend mit der `identity`-Relation annotiert:

TEMPELHERR.

... Aber Rechas Bruder? Rechas Bruder ...

NATHAN.

Seid Ihr!

→ `transfer(nathan, tempelherr, identity(SIBLING_OF[RECHA], tempelherr))`

Die Auflösung der Variablen muss immer über die `identity`-Relation erfolgen. Im Beispiel kann deshalb nicht einfach `SIBLING(recha, tempelherr)` annotiert werden, weil der konkrete Bezug zur Variable dann unklar bleiben würde.

Variablen können auch im Plural stehen, etwa `SIBLINGS[IPHIGENIE]`. Bei solchen Gruppen von Figuren muss die Auflösung der Variablen aber nicht unbedingt gleichzeitig erfolgen. Im Beispiel von Goethes *Iphigenie* erfahren wir etwa zunächst, dass Elektra Iphigenies Schwester und erst später, dass Orest ihr Bruder ist. Wir stellen die Verbindung zur Gruppe über das gleiche Schlüsselwort (`SIBLING`) her, anstelle des Plural-s erfolgt auch hier eine Nummerierung: `identity(SIBLING-1[IPHIGENIE], "Elektra")`.

Manche Figuren werden in einem Drama nur am Rande erwähnt, haben aber keine weitere Bedeutung für den Text. Relationen dieser Figuren werden nicht annotiert. Konkret gilt das für Figuren, die

- selbst nicht im Drama auftreten und
- für die Handlung des Dramas irrelevant sind.

Letzteres kann ggf. erst nach Lesen des ganzen Dramas beurteilt werden. Die Figur Assad tritt in *Nathan der Weise* beispielsweise nicht selbst auf, ihre Relationen zu anderen Figuren sind für die Handlung aber trotzdem entscheidend.

Zusätzlich zum Wissensstand der Figuren interessieren wir uns auch für den Wissensstand des Publikums. Für das Publikum wird die ID `audience` verwendet. Die ersten Wissensbestände des Publikums entstehen üblicherweise durch die Figurentafel am Anfang des Textes. Wissensvermittlungen an das Publikum können, insbesondere am Anfang des Textes, relativ unmarkiert erfolgen. Wenn etwa eine Figur eine andere als *Bruder* anspricht, ist diese Relation für beide Figuren (normalerweise) schon vorher bekannt, für das Publikum aber unter Umständen eine neue Information.

`transfer("Figurentafel", audience, fosterparent_of(nathan, recha))`

¹Wir annotieren also zunächst nur, dass wir von der Existenz einer solchen Figur wissen. Die Existenz mancher Figuren kann natürlich bereits durch *Common Sense* angenommen werden, etwa dass eine Figur einen Vater hat. Um formale Einheitlichkeit zu gewährleisten, werden solche Fälle aber trotzdem annotiert.

In vielen Fällen erhalten mehrere Figuren und/oder das Publikum Informationen zur gleichen Zeit. In diesem Fall wird als ZIEL eine Liste aus allen Entitäten angelegt, für die das transferierte Wissen neu ist. Listen werden von eckigen Klammern umschlossen, die Elemente der Liste werden durch Kommata voneinander getrennt:

```
transfer(nathan, [recha, tempelherr, saladin, sittah], siblings(tempelherr, recha))
```

Als Quelle von Wissen treten selten mehrere Figuren gemeinsam auf, grundsätzlich kann aber auch an dieser Stelle eine Liste von Figuren verwendet werden.

7 Figurenrelationen

Bei der Annotation der Figurenrelationen folgen wir diesem Muster:

```
relation(CHARACTER-A, CHARACTER-B)
```

CHARACTER-A und CHARACTER-B stehen jeweils für die beteiligten Figuren. Beispiel:

```
Nathan ist der Vater von Recha.  
→ parent_of(nathan, recha)
```

Tabelle 1 fasst die zu annotierenden Relationen zusammen. Zu Relationen, für die es inhaltlich sinnvoll ist, können außerdem Varianten mit den Präfixen **grand-**, **step-**, **foster-**, **god-** oder **ex-** sowie dem Suffix **-in-law** verwendet werden. Des Weiteren ist es möglich, alle Relationen bei Bedarf mit einem Ausrufezeichen vor der Relation zu negieren:

```
Nathan ist nicht der Vater von Recha.  
→ !parent_of(nathan, recha)
```

Auch in den Relationen können an Stelle von CHARACTER-A und CHARACTER-B in Einzelfällen mehrere Figuren in Form einer Liste in eckigen Klammern stehen:

```
cousins([marschallin, der_baron], "Jörger")
```

7.1 Gerichtete Relationen

In gerichteten Relationen wird die Relation aus der Perspektive einer der beiden beteiligten Figuren benannt. Die Relation ist asymmetrisch: Wenn A der Vater von B ist, ist B nicht auch der Vater von A. Die Reihenfolge der Figuren ist deshalb wichtig. Wir folgen hier dem Muster „A steht in dieser Relation zu B“:

```
Nathan ist der Vater von Recha.  
→ parent_of(nathan, recha)
```

Im Fall gerichteter Relationen gibt es oft eine komplementäre Relation. Die Vater-Tochter-Relation von Nathan und Recha kann etwa durch `parent_of(nathan, recha)` und `child_of(recha, nathan)` ausgedrückt werden. Bei der Annotation verschiedener für die Wissensweitergabe wichtiger Erwähnungen dieser Relation orientieren wir uns an der Perspektive, die an der Textoberfläche eingenommen wird. Innerhalb eines Dramas können also durchaus beide Relationsformen vorkommen.

7.2 Ungerichtete Relationen

In ungerichteten Relationen gibt es keine bestimmte Reihenfolge der genannten Figuren A und B. Wenn A ein Geschwister von B ist, ist B auch ein Geschwister von A:

```
Recha und der Tempelherr sind Geschwister.  
→ siblings(recha, tempelherr)
```

Bei ungerichteten Relationen spielt die Reihenfolge der Figuren in der Klammer keine Rolle.

A) Verwandtschaft

Gerichtete Relationen	Ungerichtete Relationen
<code>parent_of(PARENT, CHILD)</code> <code>child_of(CHILD, PARENT)</code>	
	<code>siblings(SIBLING-A, SIBLING-B)</code>
<code>aunt:uncle_of(AUNT:UNCLE, NIECE:NEPHEW)</code> <code>niece:nephew_of(NIECE:NEPHEW, AUNT:UNCLE)</code>	<code>cousins(COUSIN-A, COUSIN-B)</code>

B) Partnerschaft

Gerichtete Relationen	Ungerichtete Relationen
<code>in_love_with(LOVER, TARGET)</code>	<code>lovers(LOVER-A, LOVER-B)</code> <code>couple(PARTNER-A, PARTNER-B)</code> <code>engaged(PARTNER-A, PARTNER-B)</code> <code>spouses(PARTNER-A, PARTNER-B)</code>
<code>widow:er_of(WIDOW:ER, DEAD-PARTNER)</code>	

C) Identitäten

Gerichtete Relationen	Ungerichtete Relationen
<code>has_name(A, NAME)</code>	<code>identity(A, B)</code>

Tabelle 1: Zu annotierende Figurenrelationen. Wo notwendig können die Präfixe `grand-`, `step-`, `foster-`, `god-` und `ex-` sowie das Suffix `-in-law` ergänzt werden.

7.3 Erläuterung von Relationstags

Nicht alle der Relationstags sind selbsterklärend. Die folgenden Erläuterungen sollen mögliche Unklarheiten adressieren:

Liebesrelationen. Bei Liebesrelationen werden mehrere Phänomene unterschieden: Die beiden Tags `lovers(LOVER-A, LOVER-B)` und `in_love_with(LOVER, TARGET)` decken Liebesverhältnisse ab, die sowohl einseitiger als auch gegenseitiger Natur sein können.² Entscheidend für die Unterscheidung dieser beiden Tags ist die Frage, ob nur A B liebt (oder wir jedenfalls nur davon wissen \rightarrow `in_love_with(LOVER, TARGET)`) oder ob ein gegenseitiges Liebesverhältnis besteht (\rightarrow `lovers(LOVER-A, LOVER-B)`). Für Liebesrelationen, die sich als längerfristige Beziehung oder Partnerschaft charakterisieren lässt, verwenden wir das Tag `couple(PARTNER-A, PARTNER-B)`. Diese Relation kann nur als ungerichtet, also auf Gegenseitigkeit beruhend annotiert werden. Die Art der Liebesbeziehung ist dabei nicht festgelegt. Längerfristige Affären sind ebenso denkbar wie publik gemachte offizielle Beziehungen. Liebesrelationen werden ihrer Natur gemäß meist weniger explizit im Dramentext benannt als Verwandtschaftsverhältnis. Nach Möglichkeit annotieren wir Textstellen, die die Liebesrelation zweifelsfrei annehmen lassen. In vielen Fällen werden aber auch vagere Aussagen und Handlungen annotiert werden müssen, die dem Publikum in Summe jedoch trotzdem vermitteln, dass eine Liebesrelation vorliegt.

`has_name(A, NAME)`. Hierbei handelt es sich nicht um eine Relation im eigentlichen Sinne. Das Tag wird verwendet, wenn ein bisher unbekannter Name einer Figur eingeführt wird. Namen schreiben wir in Anführungsstrichen.

```
transfer(tempelherr, nathan, has_name(tempelherr, "Curd von Stauffen"))
```

D. h.: Der Tempelherr teilt Nathan mit, dass er den Namen Curd von Stauffen trägt.

²Der Einfachheit halber werden hierunter z. B. auch Heiratsabsichten gefasst, bei denen die Motivation nicht-emotionaler Natur ist.

`identity(A, B)`. Dieses Tag wird verwendet, wenn zwei Figuren sich im Verlauf des Dramas als die gleiche herausstellen oder ihre Identität behauptet wird. Beispielsweise behauptet der Baron im *Rosenkavalier* dem Kommissar gegenüber, bei der vermeintlichen Kammerzofe handele es sich um seine Verlobte Sophie:

```
transfer(baron, kommissar, identity(sophie, kammerzofe), lie)
```

Das Tag kann ebenfalls verwendet werden, wenn eine Figur sich einer zweiten Identität bedient, indem sie sich beispielsweise verkleidet.

```
transfer("Octavian verkleidet sich", [marschallin, audience], identity(kammerzofe, octavian))
```

D. h.: Dadurch, dass Octavian sich verkleidet, erfahren die anwesende Marschallin und das Publikum, dass es sich bei der Kammerzofe um Octavian handelt.

Der häufigste Anwendungsfall der `identity`-Relation ist die Auflösung von Variablen, wie in Abschnitt 6 beschrieben.

8 Wissen über Wissen

Zusätzlich zu unmittelbarem Wissen über Figurenrelationen ist an vielen Textstellen auch das Wissen über das Wissen anderer Figuren von Bedeutung. In solchen Fällen kann an der Stelle `WISSEN` ein weiteres Wissens-Tag verwendet werden.

- Wenn es um das Wissen über einen Wissenstransfer geht, wird hier wiederum das Prädikat `transfer()` verwendet.

```
transfer(recha, sittah, transfer(daja, recha, fosterparent_of(nathan, recha)))
```

D. h.: Recha eröffnet Sittah, dass sie von Daja erfahren hat, dass Nathan ihr Pflegevater ist.

- Handelt es sich um Wissen, das nicht gerade vermittelt wird, sondern schon vorhanden war, wird stattdessen das Prädikat `knowledge(WISSENDE:R, WISSEN, ATTRIBUTE)` verwendet. Das Argument zur Quelle des Wissens fällt hier weg.

```
transfer(daja, nathan, knowledge(recha, fosterparent_of(nathan, recha)))
```

D. h.: Daja gibt an Nathan weiter, dass Recha weiß, dass Nathan ihr Pflegevater ist.

Ebenfalls annotiert wird das Wissen über Nichtwissen. Hierzu kann das Tag `!knowledge()` verwendet werden. In *Nathan der Weise* erfährt beispielsweise der Tempelherr von Daja, dass Recha nicht weiß, dass Nathan nicht ihr Vater ist:

```
transfer(daja, tempelherr, !knowledge(recha, !parent_of(nathan, recha)))
```

Wissen über Wissen muss nicht begleitend zu jeder Wissensannotation annotiert werden: Wenn Daja etwa Recha erzählt, dass Nathan nicht Rechas Vater ist, dann weiß Daja anschließend über Rechas Wissen Bescheid (und andersherum weiß auch Recha über Dajas Wissen Bescheid und das Publikum weiß, dass beide es wissen). Diese Zusammenhänge sind logisch zu erschließen und werden nicht gesondert annotiert.

9 Sonstiges

9.1 Bruder, Oheim & Co.

Bezeichnungen wie *Bruder*, *Oheim* oder *Vetter* werden oft für Personen verwendet, die im engeren Sinne dieses Verwandtschaftsverhältnis nicht ausweisen. Wenn eine Figur eine andere z. B. als Vetter bezeichnet, ist deshalb oft nicht eindeutig, ob das Verwandtschaftsverhältnis besteht oder nicht. Wir annotieren solche Textstellen deshalb zunächst mit dem Attribut `uncertain`. Wenn sich das wahre Verwandtschaftsverhältnis im Textverlauf klärt, wird die Annotation an entsprechender Stelle ohne das Attribut wiederholt.

9.2 Automatische Schlüsse

Aus vielen Relationen lassen sich weitere Relationen logisch ableiten. Wenn etwa gilt, dass `PARENT_OF(A, B)`, dann gilt auch `CHILD_OF(B, A)`. Letzteres muss deshalb nicht gesondert annotiert werden, sondern gilt ebenfalls sofort als bekannt.

Bei anderen Relationen sind die Schlüsse weniger deterministisch: Wenn gilt, dass `PARENT_OF(A, B)` und `PARENT_OF(C, B)`, so gilt (jedenfalls im Dramenkopus) mit hoher Wahrscheinlichkeit auch `SPOUSES(A, C)`. Hierzu gibt es aber natürlich zahlreiche Ausnahmen. Wir gehen bei der Annotation im Zweifelsfall davon aus, dass das Wissen nicht automatisch erschließbar ist und annotiert werden muss.

10 Zweifelsfälle

Bei den im Folgenden dargelegten Zweifelsfällen handelt es sich um spezifische Einzeltextstellen, deren Annotation anhand der bisher vorgestellten Guideline nicht eindeutig vorzunehmen ist. Wenn eine größere Zahl an Zweifelsfällen zu verwandten Phänomenen vorliegen, ergeben sich daraus mittelfristig Generalisierungen im Umgang mit ihnen, die dann Eingang in den Hauptteil der Guideline finden.

10.1 Abweichende Wissensbestände

10.1.1 Unterschiedliche Empfänger:innen haben unterschiedliches Wissen

Durch eine Aussage kann sich für eine Figur oder das Publikum eine Variable auflösen, für eine andere aber nicht. Beispiel aus *Iphigenie auf Tauris*: Iphigenie berichtet Thoas, dass ihr Vater Agamemnon ist. Thoas wird hier eine neue Relation vermittelt: `parent_of("Agamemnon", iphigenie)`. Das Publikum hingegen hat schon vorher Informationen über den Vater Iphigeniens erhalten, die der Variable `PARENT-1[IPHIGENIE]` zugeordnet wurden. Die neue Information für das Publikum ist demnach `identity(PARENT-1[IPHIGENIE], "Agamemnon")`. Wir lösen diesen Konflikt durch eine Doppelannotation der Textstelle mit folgenden beiden Labels:

```
transfer(iphigenie, audience, identity(PARENT-1[IPHIGENIE], "Agamemnon"))
transfer(iphigenie, thoas, parent_of("Agamemnon", iphigenie))
```

10.1.2 Sender:in und Empfänger:in haben unterschiedliches Wissen

Der Klosterbruder teilt Nathan mit, dass er ihm damals die Tochter von Wolf von Filneck gebracht hat. Nathan weiß, dass es sich um Recha handelt, der Klosterbruder aber hat sie seither nicht mehr gesehen. Die folgende Annotation ist deshalb nicht korrekt, weil man annehmen würde, dass das vermittelte Wissen auch dem Sender bekannt ist:

```
transfer(klosterbruder, [nathan, audience], parent_of("Wolf von Filneck", recha))
```

Lösungsansätze: 1) Der Wissensbestand des Klosterbruders wird für den weiteren Verlauf des Dramas als nicht zentral eingeschätzt und die Ungenauigkeit kann in Kauf genommen werden. 2) Doppelannotation mit einer Hilfsannotation zum Wissen von Nathan und dem Publikum:

```
transfer(klosterbruder, [nathan, audience], parent_of("Wolf von Filneck",
FOSTERCHILD[NATHAN]))
knowledge([nathan, audience], identity(FOSTERCHILD[NATHAN], recha))
```

10.2 Unklare Identitäten

10.2.1 Hofmannsthals Elektra

Elektra und Orest treffen aufeinander, erkennen sich aber zunächst gegenseitig nicht. Orest gibt sich als Fremder aus, Elektra als Magd am Hof. Später erkennen sie sich dann. Wir annotieren nur die Auflösung der Identitäten, ohne dass die Magd und der Fremde in den vorangehenden Annotationen auftauchen.

```
transfer("Elektra gibt sich als Magd aus", audience, identity(MAGD, elektra))
```

Evtl. explizit das Nicht-Wissen von Orest annotieren:

```
!knowledge(orest, identity(MAGD, elektra))
```

Später Auflösung:

```
transfer(elektra, orest, identity(MAGD, elektra))
```

10.2.2 Kleists Familie Schroffenstein

Ottokar gesteht Agnes seine Liebe, als ihre gegenseitige Identität noch nicht geklärt ist. Ottokar vermutet, dass es sich bei dem Mädchen um Agnes handelt, Agnes weiß vermutlich nichts über seine Identität. Wir nehmen eine dreifache Annotation vor, die einmal die Perspektive Ottokars, einmal die Perspektive Agnes' und einmal die des Publikums einnimmt:

```
transfer(ottokar, LOVER[OTTOKAR], in_love_with(ottokar, LOVER[OTTOKAR]))
transfer(LOVER[AGNES], agnes, in_love_with(LOVER[AGNES], agnes))
transfer(ottokar, audience, in_love_with(ottokar, agnes))
```

Agnes wird später im Wald von einem Fremden (vermeintlich) angegriffen. Jeronimus erkennt ihn im Anschluss als Johann, den unehelichen Sohn von Rupert.

```
transfer(jeronimus, [sylvester, gertrud, agnes], identity(FREMDER-IM-WALD, johann))
```

Wichtig: Unterscheidung zwischen

- I weiß, dass A der Sohn von B ist
- I erkennt eine anwesende Person als A, den Sohn von B

In der gegebenen Szene nehmen wir an, dass die Anwesenden bereits wissen, dass Johann der uneheliche Sohn von Rupert ist.

10.2.3 Schillers Braut von Messina

Don Manuel berichtet, dass er eine Liebhaberin hat, die seine Identität noch nicht kennt:

```
transfer(don_manuel, [audience, chor-1], !knowledge(LOVER[DON_MANUEL],
identity(LOVER[LOVER[DON_MANUEL]], don_manuel)))
```

(Mehrfache Verschachtelung in Variablen kommt häufiger vor: PARENT[LOVER[DON_MANUEL]])

Don Cesar gesteht Beatrice seine Liebe, beide kennen aber die Identität der:s anderen nicht. Wir annotieren deshalb (analog zur *Familie Schroffenstein* oben) dreifach für die Perspektiven von Don Cesar, Beatrice und dem Publikum:

```
transfer(don_cesar, LOVER[DON_CESAR], in_love_with(don_cesar, LOVER[DON_CESAR]))
transfer(FREMDER-AUS-KIRCHE, beatrice, in_love_with(FREMDER-AUS-KIRCHE, beatrice))
transfer(don_cesar, audience, in_love_with(don_cesar, beatrice))
```

Dann eröffnet Don Cesar seine Identität gegenüber Beatrice:

```
transfer(FREMDER-AUS-KIRCHE, beatrice, identity(FREMDER-AUS-KIRCHE, don_cesar))
```

Deshalb kann die folgende „Verlobung“ nicht mehr analog zur Liebeserklärung annotiert werden. Die Annotation aus Beatrice' Perspektive:

```
transfer(don_cesar, beatrice, engaged(don_cesar, beatrice))
```

Diese Annotation darf keine Implikationen für den Wissensbestand von Don Cesar haben. Dieses Problem lösen wir, in dem wir an der Stelle der Liebeserklärung eine vierte Hilfsannotation ergänzen:

```
knowledge([beatrice, audience], identity(LOVER[DON_CESAR], beatrice))
```

Bei der „Verlobung“ annotieren wir dann:

```
transfer(don_cesar, [LOVER[DON_CESAR], audience], engaged(don_cesar, LOVER[DON_CESAR]))
```

In Kombination mit der Hilfsannotation ermöglicht das den Schluss, dass sowohl Beatrice als auch das Publikum wissen, dass Beatrice Gegenstand dieser Verlobung ist.

10.2.4 Grillparzers Ahnfrau

Jaromir trifft nachts auf die Ahnfrau, die er für Bertha hält, und gesteht ihr seine Liebe:

```
transfer(jaromir, BERTHA_ÄHNLICHE, in_love_with(jaromir, BERTHA_ÄHNLICHE))
knowledge(jaromir, identity(BERTHA_ÄHNLICHE, bertha))
```

10.2.5 Kriterien zur Vergabe der DraCor-ID (Identität vs. Name)

Insbesondere bei Figuren, die erst als Variablen eingeführt werden, ist manchmal unklar, ab wann sie mit der DraCor-ID gleichgesetzt werden können.

Bsp. *Komtesse Mizzi*: Der Fürst erzählt, dass er einen Sohn hat → CHILD(FUERST) wird eingeführt. Der Name des Sohns wird zum ersten Mal genannt → reicht das für die Einführung der ID aus?

Bsp. *Die Ahnfrau*: Der Graf erzählt Bertha, dass einer Sage zu Folge die Ahnfrau der Familie aufgrund ihrer Sünden nicht zur Ruhe kommen kann, bis ihr Stamm ausgestorben ist. Zunächst annotiert als RELATIVE[DER_GRAF]. Der Graf und Bertha berichten von nächtlichen Erscheinungen. Günther äußert den Verdacht, dass es sich um die Ahnfrau aus der Sage gehandelt haben könnte. Daraus könnte identity(RELATIVE[DER_GRAF], NÄCHTLICHE-GESTALT) folgen. Wann vergeben wir der Ahnfrau die DraCor-ID? Lösung: Die Ahnfrau wird direkt mit ihrer ID annotiert. Hierbei handelt es sich bis auf Weiteres um Einzelfallentscheidungen.

11 Annotationen finalisieren

Um die Korrektheit und Konsistenz der Annotationen sicherzustellen, empfehlen wir, nach Abschluss der Annotation folgende Checkliste durchzugehen:

- Werden alle Figuren mit DraCor-IDs auch darüber referenziert?
- Stehen alle anderen Figurenbezeichnungen in Anführungsstrichen?
- Werden alle Klammern am Ende geschlossen?
- Gibt es sonstige Tippfehler?
- Prüfe für jede annotierte Wissensweitergabe, welche Figuren dieses Wissen im Verlauf des Dramas erlangen. Gibt es weitere Figuren, die dieses Wissen erhalten, aber bislang noch nicht annotiert sind?
- Prüfe insbesondere, ob das Publikum als Ziel des Wissens an bestimmten Textstellen ergänzt werden muss.