



Technische Universität Dresden
Fakultät Informatik
Institut für Theoretische Informatik
Lehrstuhl für Automatentheorie

Diplomarbeit

Modellierung zellbiologischer Prozesse mittels Brane- und κ -Kalkül

Sebastian Voigt

Eingereicht am 6. Dezember 2007

Betreuender Hochschullehrer: Prof. Dr. Franz Baader

Betreuerin: Dr. Monika Sturm

Aufgabenstellung Diplomarbeit

Modellierung zellbiologischer Prozesse mittels Brane- und κ -Kalkül

Bearbeiter: Sebastian Voigt (Informatik)

Die Anwendung von Prozesskalkülen zur Beschreibung parallel ablaufender Prozesse steht im zunehmenden Maße auch im Interesse der Systembiologie. Bekannt sind Ansätze der Anwendung von Prozesskalkülen im Bereich des DNA-Computing, wie z.B. zur Verifikation DNA-basierter Algorithmen. Die Systembiologie setzt Prozesskalküle mit dem Ziel ein, Abläufe in Zellen, in Zellverbänden und in ganzen Organismen mithilfe von Simulation zu erschliessen, um damit ein komplexes Verständnis der Lebensprozesse aufzubauen und zu vermitteln. In der Theoretischen Informatik ist bekannt, dass eine Anwendung der klassischen Prozesskalküle in der Biologie nicht ohne spezielle Anpassung möglich und als neue Herausforderung zu betrachten ist. In den letzten Jahren (konkret seit 2003) konnten von der zugehörigen Community erste Ergebnisse veröffentlicht werden. Die Diplomarbeit soll zwei Ansätze vorstellen und mit einer konkreten Adaption einen Beitrag für das Fachgebiet leisten.

- Zu der Aufgabenstellung ist eine Literaturrecherche anzufertigen.
- Der Brane- und der κ -Kalkül (Luca Cardelli, 2004, Vincent Danos, Cosimo Laneve, 2003) sind ausführlich vorzustellen.
- Zur formalen Beschreibung eines ausgewählten biologischen Prozesses (z.B. Prozess zur Proteininteraktion) sind beide Kalküle einzusetzen.
- Auf der Basis der Untersuchung beider Kalküle hinsichtlich einer universellen Anwendbarkeit in der Systembiologie ist ein Ansatz zu konzipieren, der sich partiell aus Syntax und Semantik beider Kalküle zusammensetzt und einer breiteren Anwendung zur Verfügung steht.
- Bewertung der Ergebnisse

Folgende Literatur wird empfohlen:

- Luca Cardelli. *Brane Calculi. Interactions of Biological Membranes*. Computational Methods in Systems Biology, 257-278, 2004.
- Vincent Danos, Cosimo Laneve. *Formal Molecular Biology*. December 2003.
- Cosimo Laneve, Fabien Tarissan. *A simple calculus for proteins and cells*. MeC-BIC 2006.

Betreuerin: Dr. M. Sturm

Betreuender Hochschullehrer: Prof. F. Baader

Bearbeitungszeitraum: 11.06.2007 - 11.12.2007

Abschlußleistung: Diplomarbeit, Verteidigung mit Vortrag

Erklärung an Eides statt

Hiermit erkläre ich an Eides statt, diese Diplomarbeit selbstständig und ohne Verwendung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt zu haben.

Sebastian Voigt
Dresden, 6. Dezember 2007

Danksagung

Ich möchte mich an dieser Stelle herzlich bei Frau Dr. Sturm bedanken, die durch exzellente fachliche und menschliche Betreuung ein wichtiger Begleiter meines gesamten Studiums und insbesondere dieser Diplomarbeit gewesen ist. Weiterer Dank gilt Florian Stenger für zahlreiche interessante Gespräche zur Thematik, die Bereitstellung seiner Bilder und das intensive Korrekturlesen der Arbeit sowie Sophie Haufe für die geduldige Beantwortung medizinischer Fragen, besonders bedanken möchte ich mich darüberhinaus für deren große Unterstützung als enge Freunde. Weiterhin danke ich meiner Familie sowie allen anderen, ohne die mein Studium und diese Diplomarbeit nicht möglich gewesen wäre.

Inhaltsverzeichnis

1	Einleitung	1
2	Informatik in der Systembiologie	3
2.1	Aufbau und Funktionen einer Zelle	4
2.1.1	Proteine	5
2.1.2	Biologische Membranen	5
2.2	Abstraktion von der Biologie	6
2.2.1	Einige Modellierungsansätze	8
2.3	Prozesskalküle	8
2.3.1	β -Binder	10
2.3.2	Bio-Ambients	11
3	Der Brane-Kalkül	13
3.1	Syntax	13
3.2	Strukturelle Kongruenz	15
3.3	Reduktionsrelation	16
3.3.1	Phago/Exo/Pino	17
3.3.2	Mate/Bud/Drip	20
3.3.3	Bind and Release	22
3.4	Weitere Arbeiten zum Brane-Kalkül	24
3.4.1	Eigenschaften des Kalküls	25
3.4.2	Verwandte Prozesskalküle	25
3.5	Anwendungsbeispiel: HIV	26
3.5.1	Aufbau und Funktionsweise	27
3.5.2	Modellierung	29
3.5.3	Diskussion	37
4	Der κ-Kalkül	39
4.1	Syntax	39
4.2	Semantik	41
4.2.1	Weitere Betrachtungen	48
4.3	Anwendungsbeispiel: HIV-Proteininteraktionen	50
4.3.1	Modellierung	50
4.3.2	Diskussion	52
4.4	Integration von Konzepten der Membran-Maschine	52
4.4.1	Erweiterung um Membranen	52

4.4.2	Membranfusion und -aktivierung	55
4.4.3	Transmembraner Transport und Phagozytose	57
4.4.4	Diskussion	60
5	Der Protein-Brane-Kalkül	63
5.1	Syntax	63
5.2	Semantik	65
5.3	Modifikationen des Kalküls	72
5.4	Anwendungsbeispiel: HIV mit Proteininteraktionen	73
5.4.1	Modellierung	73
5.4.2	Diskussion	78
6	Bewertung und Ausblick	81

Kapitel 1

Einleitung

Das Verständnis des Aufbaus und der Funktionsweise biologischer Systeme steht zunehmend im Interesse aktueller Forschung und wird aufgrund der sich schnell fortentwickelnden technologischen Möglichkeiten beständig vorangetrieben. Die immense Komplexität biologischer Systeme bietet zahlreiche Aufgabenstellungen für viele Fachbereiche und hat zur Entwicklung des hochgradig interdisziplinären Forschungsgebietes der *Systembiologie* geführt. Aufgrund der enormen Geschwindigkeit des Wissenszuwachses ist die Konstruktion geeigneter Mittel des Zugriffs und der Repräsentation erlangter Erkenntnisse erforderlich und liefert neue Herausforderungen für die Informatik. Neben dem Aufbau von Datenbanken für die strukturierte Abfrage gewonnener Zusammenhänge steht die Entwicklung mathematischer Modelle im Vordergrund. Deren Anforderungen bestehen in der exakten formalen Beschreibung einzelner biologischer Prozesse und darauf aufbauend in der Unterstützung von Laborexperimenten durch automatisierte Analyse und Simulation. Als gegenwärtig geeignet hat sich die Konzentration auf die äußerst vielschichtigen Abläufe innerhalb einzelner Zellen und deren Interaktionen untereinander herausgestellt. Erst auf Grundlage plausibler Modelle für dieses Abstraktionsniveau kann schrittweise der jeweils größere Komplexitätsgrad von Geweben, Organen und schließlich gesamten Organismen bewältigt werden.

Innerhalb der letzten Jahre haben sich *Prozesskalküle* als Modellierungsansatz in der Systembiologie etabliert. Ursprünglich für die Beschreibung nebenläufiger Systeme im Bereich der Telekommunikation entwickelt, stellt ihr auf unabhängigen Agenten und einer modularen Struktur basierendes Konzept ein geeignetes Mittel zur Beschreibung systembiologisch relevanter Inhalte dar. Die spezifischen biologischen Mechanismen benötigen jedoch geeignete Anpassungen der Kalküle und bieten deshalb zahlreiche Aufgabenstellungen insbesondere für die theoretische Informatik. Dabei haben sich einige Ansätze herausgebildet, die sich auf einzelne Teilfunktionalitäten der Zelle konzentrieren und aufgrund ihrer Spezialisierung auch bei vielen Details eine direkte und übersichtliche Darstellung erreichen. Eine solche Teilfunktionalität stellen *biologische Membranen* dar. Über ihre Funktion zur räumlichen Abgrenzung einzelner Zellbereiche hinaus kommt ihnen große Bedeutung bei der Organisation sowohl intra- als auch interzellulärer biologischer Prozesse zu. Dennoch ist die Zahl diesbezüglich vorhandener Modellierungsansätze gering, der *Brane-Kalkül* als einer der Hauptvertreter erfährt darüberhinaus nur in wenigen Modellierungen Verwendung. Die Untersuchung seiner Fähigkeiten zur Modellierung systembiologisch relevanter Fragestellungen bildet den

ersten Schwerpunkt dieser Arbeit. Anhand einer umfangreichen Beispielmodellierung demonstrieren wir seine zahlreichen Möglichkeiten und zeigen biologische Verhaltensweisen auf, deren Darstellung basierend auf Membraninteraktionen allein nur schwer oder gar nicht erreichbar ist. Wir stellen anschließend den eigens für die Modellierung komplexer proteinbasierter Signaltransduktionswege entwickelten κ -Kalkül vor und demonstrieren seine Eignung zur Formulierung der jeder Membraninteraktion zugrunde liegenden steuernden Proteinprozesse. Eine vorhandene Erweiterung dieses Ansatzes, der *bio* κ -Kalkül, integriert Möglichkeiten zur Beschreibung von Membranen und ermöglicht somit die Einbettung von Proteininteraktionen in ihren natürlichen Kontext. Wir zeigen auf, dass der *bio* κ -Kalkül im Vergleich zum Brane-Kalkül jedoch einige membranbezogene Modellierungsaspekte vermissen lässt. Der zweite Schwerpunkt dieser Arbeit besteht in der Entwicklung des *Protein-Brane-Kalküls*, eines auf dem Brane-Kalkül basierenden Ansatzes, der die umfangreichen Möglichkeiten des *bio* κ -Kalküls zur Modellierung komplexer Proteininteraktionen vollständig integriert. Darüberhinaus erweitert er die im Brane-Kalkül darstellbaren Membranprozesse um die zusätzliche Betrachtung ihrer zugrunde liegenden Proteininteraktionen. Unser Kalkül erreicht die Modellierbarkeit verschiedener entweder im Brane- oder im *bio* κ -Kalkül nicht darstellbarer biologischer Prozesse unter Wahrung der Vorteile der beiden Ansätze. Wir demonstrieren seine erweiterte Ausdruckstärke abschließend anhand unserer Beispielmodellierung.

In Kapitel 2 führen wir zunächst an das Gebiet der Systembiologie heran und betrachten einige grundlegende Bestandteile und Funktionsprinzipien ihres derzeitigen zentralen Modellierungsobjektes, der biologischen Zelle. Wir gehen auf die Bedeutung von Prozesskalkülen in der Systembiologie ein und stellen mit den *β -Bindern* und den *Bio-Ambients* zwei Vertreter vor, die spezielle Konzepte zur Modellierung von Membraninteraktionen besitzen. Der ausführlichen Darstellung des Brane-Kalküls widmen wir uns in Kapitel 3 und geben einen kurzen Überblick über unmittelbar mit ihm im Zusammenhang stehende weitere Arbeiten. Anschließend modellieren wir die Infektion einer Zelle durch das *humane Immundefizienzvirus (HIV)* und einige mit der Infektion in Zusammenhang stehende Effekte zur Demonstration der Möglichkeiten des Brane-Kalküls, und arbeiten einige seiner Grenzen heraus. In Kapitel 4 stellen wir den κ -Kalkül vor, zeigen seine Anwendbarkeit exemplarisch an einigen mit der HIV-Infektion in Zusammenhang stehenden Proteininteraktionen und diskutieren seine Erweiterung zum *bio* κ -Kalkül. In Kapitel 5 entwickeln wir den Protein-Brane-Kalkül und schließen unsere Betrachtungen mit der erneuten Modellierung einiger mit der HIV-Infektion in Zusammenhang stehender Effekte ab. Kapitel 6 dient der Einordnung der Ergebnisse und zeigt Möglichkeiten zur weiteren Arbeit auf.

Kapitel 2

Informatik in der Systembiologie

Die Systembiologie ist ein junges, hochgradig interdisziplinäres Forschungsgebiet. Ausschlaggebend für seine schnelle Entwicklung ist ein beständig zunehmendes Interesse an den komplexen Vorgängen biologischer Systeme, welches nicht zuletzt das Erzielen weiterer Fortschritte im Gesundheitswesen und in der Arzneimittelenwicklung zum Ziel hat. Die zahlreichen Erkenntnisse aktueller Forschung verlangen die Entwicklung von Konzepten, welche über grafische Prinzipskizzen hinaus exakte Beschreibungen konkreter Zusammenhänge gestatten. Weiterhin ermöglichen formale Modelle den massiven Einsatz aktueller Computertechnik zur Beschleunigung der Entwicklung und Klassifizierung neuer Ergebnisse. Zwei Hauptrichtungen lassen sich feststellen ([11], [10]):

- Das eigenständige Gebiet der *Bioinformatik* beschäftigt sich mit der Organisation von Daten in speziellen biologischen Teilbereichen. Ein Beispiel ist das *Humangenomprojekt*, dessen Ziel die komplette Offenlegung der Abfolge von Basenpaaren in der menschlichen DNA gewesen ist. Ähnliche Zielstellungen findet man unter Oberbegriffen wie *Proteomik* oder *Metabolomik*. Maßgeblich ist der Einsatz von Datenbanken zur strukturierten Speicherung und Abfrage der enormen Datenmengen. Der Wunsch nach einem offenen Austauschformat für diese Dateninhalte hat darüber hinaus zur Entwicklung der *Systems Biology Markup Language*, kurz SBML, geführt.
- Die Richtung, welche in dieser Arbeit näher beleuchtet werden soll, konzentriert sich hingegen auf das Verständnis der Operationsprinzipien biologischer Systeme durch die Entwicklung mathematischer Modelle, die auf der Interaktion ihrer Komponenten basieren. Gemeinsames Ziel der verschiedenen existierenden Ansätze ist die korrekte Beschreibung und die darauf aufbauende Simulation von Systemverhalten, um eine Unterstützung und teilweise Ersetzung von Laborversuchen zur Gewinnung neuer Erkenntnisse zu ermöglichen. Im Zentrum des Interesses steht derzeit das Verständnis von Abläufen auf Ebene der einzelnen Zelle, um später mithilfe des gewonnenen Wissens den Organisationsprinzipien komplexerer Strukturen wie Geweben oder ganzen Organen nachgehen zu können.

Die in dieser Arbeit vorgestellten Konzepte und Modellierungen konzentrieren sich auf das biologische System Zelle. Zum besseren Verständnis geben wir deshalb im nächsten Abschnitt zunächst einen grundlegenden Überblick über deren wichtigste Bestandteile

und Organisationsprinzipien. Für detailliertere Informationen seien [40] und [32] empfohlen.

2.1 Aufbau und Funktionen einer Zelle

Es werden zwei Arten von Zellen unterschieden, *Prokaryoten* und *Eukaryoten*. Prokaryoten, entweder *Bakterien* oder *Archaeen*, bezeichnen einzellige Organismen mit einfacher Struktur. Sie bestehen im Wesentlichen aus einem einzigen flüssigkeitsgefüllten Raum, dem *Cytoplasma*, welcher von einer Membran umhüllt ist. Darin befinden sich die *Ribosomen*, für die Synthese von Proteinen zuständige Komplexe aus Ribonukleinsäure (RNA) und Proteinen, und das *Genom* des Organismus.

Alle anderen sowohl einzelligen als auch mehrzelligen Lebewesen bestehen aus Eukaryoten, welche eine weitaus komplexere Struktur besitzen. In ihrem Cytoplasma befinden sich zahlreiche *Organellen*, bei denen man solche mit Membranhülle, die sogenannten *Kompartimente*, und solche ohne unterscheidet. Der *Zellkern* ist mit einer Doppelmembran vom Cytoplasma abgegrenzt und enthält das Genom der Zelle. Die ebenfalls mit einer Doppelmembran ausgestatteten *Mitochondrien* dienen der Bereitstellung von Energie aus dem Abbau von Glykose. Weitere Kompartimente sind das *glatte Endoplasmatische Retikulum* zur Lipidsynthese und Entgiftung und das *rauhes Endoplasmatische Retikulum*, dessen angelagerte Ribosomen verschiedene Membran- und Organellenproteine und nahezu alle von der Zelle abzusondernden Proteine synthetisieren. Mittels abgeschnürter kleiner Bläschen, sogenannter *Vesikel*, werden die Proteine dem Transportsystem der Zelle zugeführt, dem *Golgi-Apparat*. Dieser bildet zum einen *Sekretvesikel*, welche mit den vom rauen Endoplasmatischen Retikulum synthetisierten Proteinen gefüllt an die Zellmembran geliefert werden, um mit ihr zu verschmelzen und damit ihren Inhalt aus der Zelle abzugeben. Zum anderen produziert er *Lysosomen*, welche durch den Abbau komplexer Moleküle und der Abgabe deren Bestandteile an das Cytoplasma eine Art Zellverdauung realisieren. Zu den Organellen ohne Membran zählen die Ribosomen und das *Cytoskelett*. Letzteres ist ein dichtes Geflecht von Fasern, bei denen *Mikrofilamente*, *Mikrotubuli* und *Intermediär-Filamente* unterschieden werden. Dabei handelt es sich um aus Proteinen aufgebaute *Polymere*, welche für Formgebung, intrazelluläre Bewegungsabläufe und Fortbewegung der Zelle zuständig sind. Abbildung 2.1 zeigt die schematische Darstellung einer tierischen Eukaryotenzelle.

Pflanzliche Zellen sind ebenfalls Eukaryoten und besitzen mit den *Chloroplasten* zur Synthese von Glukose eine weitere Art von Organellen mit doppelter Membranhülle. Davon abgesehen unterscheiden sie sich nicht wesentlich von den tierischen Zellen.

In einer Zelle lassen sich im Wesentlichen vier Klassen von Makromolekülen benennen. *Nukleinsäuren* beinhalten die Erbinformation in Form von DNA und dienen umgesetzt zu RNA durch die in ihr codierten Aminosäuresequenzen der Proteinsynthese. *Kohlenhydrate* stehen hauptsächlich mit der Energiespeicherung und Strukturen auf der Zelloberfläche und außerhalb von Zellen in Zusammenhang. Weil diese zwei Makromolekülklassen in dieser Arbeit keine weitere Beachtung erfahren, werden wir auf deren genauere Vorstellung verzichten. Stattdessen konzentrieren wir uns auf die anderen zwei Vertreter: *Proteine* und *biologische Membranen*.

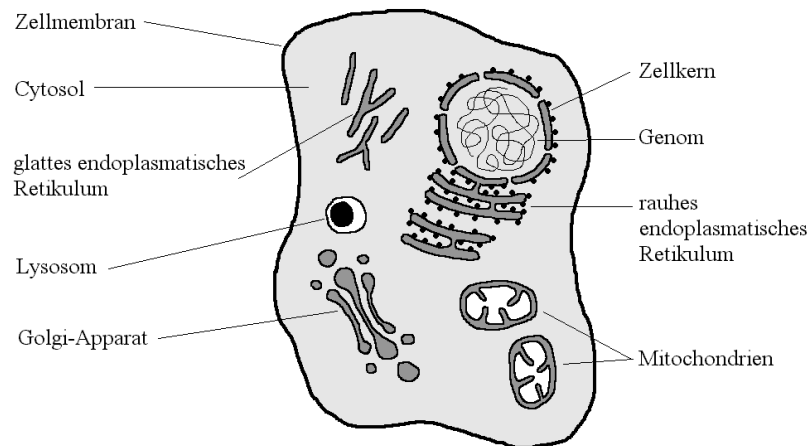


Abbildung 2.1: Schematische Darstellung einer Eukaryotenzelle (aus [49]).

2.1.1 Proteine

Die elementaren Bausteine der Proteine sind 21 Aminosäuren, welche sich zu langen, gefalteten Ketten zusammenlagern können. Daraus resultieren eine Vielzahl verschiedener Molekülstrukturen, wobei für deren Eigenschaften nicht allein die Anordnung der Aminosäuren, sondern auch die dreidimensionale Gestalt des Gesamtmoleküls ausschlaggebend sind. Insbesondere lassen sich Proteine durch ihre *Bindungsstellen* charakterisieren, an denen wiederum Proteine, aber auch andere Moleküle, andocken können. Eine solche *Proteininteraktion* ist jedoch nur mit an der Oberfläche befindlichen *aktiven* Bindungsstellen möglich, im Inneren der Molekülstruktur befindliche sind unzugänglich und damit *inaktiv*. Es kommt zu *Komplexierungen* oder *Dekomplexierungen*, wobei jede Proteininteraktion zum Aktivieren oder Deaktivieren entsprechender Bindungsstellen aufgrund von Änderungen der dreidimensionalen Gestalt führen kann. Dieser Vorgang wird auch *Konformationsänderung* genannt.

Eine weitere Interaktionsmöglichkeit entsteht durch spezielle Bindungsstellen zur *Phosphorylierung* und *Dephosphorylierung*. Hierbei führt eine meist durch ein Enzym katalysierte Anlagerung oder Abspaltung einer Phosphatgruppe zu einem Schaltmechanismus, welcher eine Änderung der Molekülstruktur und eine damit verbundene Änderung von aktiven und inaktiven Bindungsstellen nach sich zieht. Diesem Mechanismus kommt große Bedeutung bei der Energiespeicherung und dem Transport von Molekülen durch eine Membran hindurch zu.

Proteine haben verschiedenste Aufgaben in allen Bereichen der Zelle: sie vollführen katalytische Funktionen als Enzyme, wirken als Transkriptionsfaktoren bei der Proteinbiosynthese und dienen dem Stofftransport, um nur einige zu nennen. Eine entscheidende Rolle spielen sie auch bei Membranen, was im nächsten Abschnitt verdeutlicht werden soll.

2.1.2 Biologische Membranen

Biologische Membranen bestehen aus einer ca. 6 nm dicken semipermeablen Phospholipid-Doppelschicht, die ein Verhindern der Durchmischung des membranumschlossenen Inneren mit seiner Umgebung zum Ziel hat. Diese Abgrenzung realisiert separate

Funktionseinheiten, die von außen ungestört der Ausführung ihrer speziellen Aufgaben nachgehen können. Die strukturgebende Zellmembran hält die einzelnen Organellen zusammen und ermöglicht somit das Funktionieren der Zellmaschinerie in einem klar abgegrenzten Raum. Einzelne Organellen benötigen wiederum ihren eigenen Operationsbereich zur Durchführung spezieller Einzelaufgaben. Die eukaryotische Zelle besteht aus einem komplizierten System von ineinander geschachtelten, sich nicht überschneidenden Membranen.

Neben der Abgrenzung einzelner Funktionseinheiten sind verschiedene Mechanismen zum selektiven Austausch von Stoffen nötig, ohne den ein Überleben unmöglich wäre. Dazu zählen bei der Zellmembran das Aufnehmen von Zucker und Vitaminen sowie das Senden und Empfangen von Signalmolekülen zur interzellularen Kommunikation. Intrazellular sei beispielhaft auf die Lysosomen verwiesen, welche komplexe Moleküle zerlegen und wieder nach außen abgeben. Die Realisierung dieser Funktionen erfolgt durch verschiedenartige, die Membran durchdringende, *Transmembranproteine* und durch an jeweils einer Seite angelagerte *periphere Proteine*. Die Vielfalt existierender Proteinstrukturen und deren Interaktionsmöglichkeiten hat zur Entwicklung passgenauer Andockstellen, sogenannter *Rezeptoren*, für die Reaktion mit ausgewählten Molekülen der Umgebung geführt. Anhaftende Moleküle werden durch die Membran bewegt, wobei ein Transport auch gegen das Konzentrationsgefälle und damit unter Aufwendung von Energie möglich ist.

Eine weitere wesentliche Funktion von Membranen besteht in deren zahlreichen Umformungsmöglichkeiten. Proteine verschiedener Membranen und solche mit derselben Membran können durch Interaktion ein Aneinanderhaften einzelner Membranteile bewirken und damit eine Membranstrukturänderung initiieren. Diese kann zur Fusion zweier oder zum Aufspalten einer in zwei Membranen führen, aber auch zum Verschlingen oder Ausspeien einer Membran durch eine andere. Derartige Umformungen spielen eine große Rolle bei Wechselwirkungen mit komplexeren Strukturen. Dazu zählt die Aufnahme und Abgabe von Molekülen, die aufgrund ihrer Größe nicht transmembrally transportiert werden können. Ein weiteres Beispiel sind die *Makrophagen* des Immunsystems, welche sowohl einzelne Proteine als auch ganze Zellen verschlingen können.

2.2 Abstraktion von der Biologie

Strebt man ein Verständnis der höheren Organisationsprinzipien biologischer Prozesse an, sind geeignete Abstraktionen aufgrund der hohen Komplexität unabdingbar ([47]). Für die Modellierung intra- sowie interzellulärer Prozesse hat sich die Betrachtung von *abstrakten Maschinen* ([11]) im Sinne der Informatik als geeignet herausgestellt. Abstrakte Maschinen sind informationsverarbeitende Einheiten mit einer diskreten Menge von Zuständen und Operationen, mit denen sich diskrete Übergänge zwischen den Zuständen formulieren lassen. Jede der genannten vier Klassen von Makromolekülen in einer Zelle lässt sich als eine solche abstrakte Maschine auffassen. Zwei davon wollen wir etwas näher betrachten.

- In der *Protein-Maschine* wird auf die Betrachtung von dreidimensionalen Strukturen und chemischen Reaktionen verzichtet, die Beschreibung eines Proteins erfolgt lediglich durch die Angabe seiner inaktiven und aktiven Bindungs- und Phosphorylierungsstellen. Ein Zustand lässt sich formalisieren als Menge von derar-

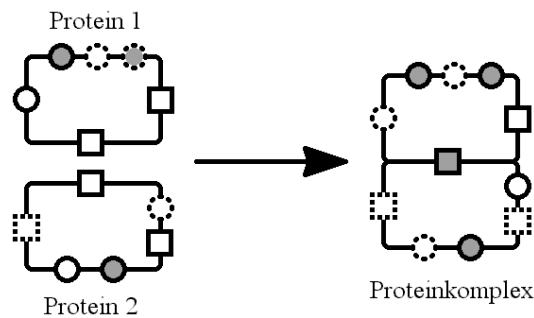


Abbildung 2.2: Komplexbildung zweier Proteine mit daraus resultierender Änderung der Zustände von Bindungs- und Phosphorylierungsstellen (aus [49]).

tig beschriebenen Proteinen. Man betrachtet vier Operationen: Komplexierungen bzw. Dekomplexierungen sind charakterisiert durch zwei interagierende Proteine unter Angabe der beteiligten Bindungsstellen und resultierender Änderungen in den Zuständen der Bindungsstellen. Phosphorylierungen bzw. Dephosphorylierungen bewirken eine Zustandsänderung an der jeweiligen Phosphorylierungsstelle.

Abbildung 2.2 demonstriert die grafische Notation der Protein-Maschine an einem Beispiel. Die Darstellung eines einzelnen Proteins erfolgt als Rechteck mit abgerundeten Ecken, seine vorhandenen aktiven Bindungsstellen werden durch auf der Rechteckskante befindliche Quadrate und seine aktiven Phosphorylierungsstellen entsprechend durch Kreise notiert. Gestrichelte Quadrate bzw. Kreise deuten auf inaktive, grau ausgefüllte hingegen auf in Verbindung stehende Bindungsstellen bzw. besetzte Phosphorylierungsstellen hin.

- In der *Membran-Maschine* werden ineinander verschachtelte Systeme von sich nicht überschneidenden Membranen als Zustände betrachtet und vier Operationen unterschieden, die in Abbildung 2.3 grafisch dargestellt sind: *Endozytose* bewirkt das Verschlingen eines nicht notwendig mit einer Membran umgebenen Systems P von einem System Q mit äußerer Membran durch das Umstülpen dieser Membran um P , gefolgt von einer Verschmelzung der umstülpenden Membranteile von Q . Hierbei entsteht eine neue Membran um P im Inneren von Q . Die Operation *Exozytose* kehrt den zuletzt beschriebenen Prozess entsprechend um. Desweiteren ist die Operation *Membranfusion* durch ein Verschmelzen zweier von äußeren Membranen umgebener Systeme unter Zusammenlegung der jeweiligen Inhalte charakterisiert. Die Umkehroperation wird mit *Membranspaltung* bezeichnet.

Die komplexen Organisationsmöglichkeiten einer Zelle basieren auf einer Vielzahl von Interaktionen zwischen den verschiedenen Funktionseinheiten: beispielsweise sind Proteine angewiesen auf bestimmte räumliche Unterteilungen zum korrekten Ablauf von Interaktionen, ebenso können Membraninteraktionen nicht ohne Rezeptor- und Signalproteine zur Steuerung der Abläufe stattfinden. Entsprechend gibt es auch in der Modellierung zahlreiche Abhängigkeiten zwischen den einzelnen abstrakten Maschinen zu beachten. Zur Reduzierung des Komplexitätsgrades kann es dennoch von Nutzen sein, sich zunächst auf eine abstrakte Maschine zu konzentrieren und lediglich wenige Me-

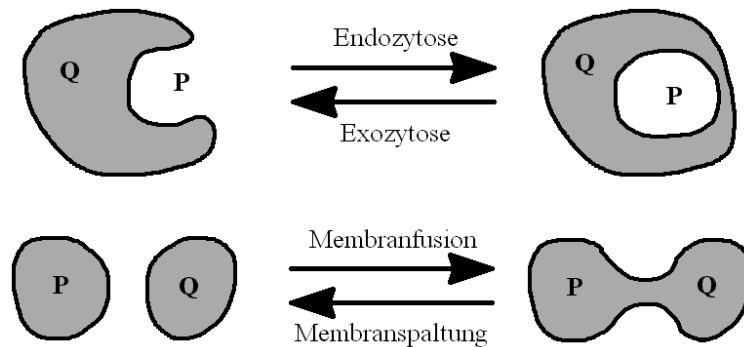


Abbildung 2.3: Die Operationen der Membranmaschine.

chanismen anderer Maschinen in das Konzept einzubinden. Dies ermöglicht direktere Umsetzungen und damit effizientere Implementierungen spezieller Teilprobleme. Wir gehen kurz auf einige Ansätze ein, die diese Strategie verfolgen.

2.2.1 Einige Modellierungsansätze

Im Folgenden wollen wir zunächst einige Ansätze nennen, denen zwar in dieser Arbeit keine weitere Bedeutung zukommt, die aufgrund ihrer thematischen Nähe aber dennoch Erwähnung finden sollen. *Kohn Molecular Interaction Maps* ([28]) ermöglichen die übersichtliche Darstellung von Proteininteraktionen in einer grafisch orientierten Sprache nahe dem Level chemischer Reaktionsgleichungen. Aufgrund ihrer statischen Notation eignen sie sich für die visuelle Beschreibung eines Systems, sind für die Simulation dynamischen Verhaltens jedoch schlecht einsetzbar. *Petri-Netze* verbinden formale Notation mit grafischer Anschaulichkeit. Obwohl sie ursprünglich nicht für Zwecke der Systembiologie entwickelt wurden, erfahren sie nun auch in diesem Feld in einigen Arbeiten Anwendung, als Beispiel sei [48] genannt. Auch Ansätze mit *Differentialgleichungen* werden verfolgt (beispielsweise in [1]), diese beschäftigen sich hauptsächlich mit der Beschreibung von Molekülanzahlen und -dichten und berücksichtigen vorwiegend kontinuierliche Modellierungsaspekte. Die drei genannten Ansätze können als Modellierungen der Protein-Maschine verstanden werden, welche jeweils einzelne Aspekte anderer Maschinen berücksichtigen.

Eine besondere Stellung nehmen die *P-Systeme* ([39]) ein. Ursprünglich vom Aufbau der Zelle inspiriert, haben sie sich als abstraktes Berechnungskonzept im Sinne der Sprachtheorie mehr und mehr von den Zielen der Systembiologie entfernt. Dennoch werden einige Ansätze diskutiert, die vorwiegend die Modellierung von Operationen auf systembiologisch nahem Level zum Ziel haben, als Beispiel sei [15] genannt. Darüberhinaus existieren Arbeiten (zum Beispiel [2] und [33]), welche sowohl Aspekte der P-Systeme als auch solche der im Folgenden vorgestellten Prozesskalküle für Modellierungen der Membran- und Protein-Maschine einsetzen.

2.3 Prozesskalküle

Die Grenzen sequentieller Berechnungsmodelle zur Beschreibung nebenläufiger Systeme führten in den 1970er Jahren zur Entwicklung der Prozesskalküle. Diese ermöglichen die

formale Modellierung von Systemverhalten basierend auf Interaktion, Kommunikation und Synchronisation einzelner *Prozesse*. Aufgrund des hohen Grades an Parallelität bei biologischen Abläufen haben sie sich auch im Bereich der Systembiologie etabliert ([42]). Der wohl bekannteste Ansatz ist der π -Kalkül ([36]), welcher sich durch seine sehr allgemeine Struktur zur Beschreibung aller abstrakten Maschinen und deren Interaktionen untereinander bedingt eignet. Eine umfassende Bewertung der Möglichkeiten und Grenzen dieses Kalküls und seiner Adaptionen für eine Anwendung in der Systembiologie liefert die Arbeit [49].

Eine Schwäche des π -Kalküls besteht in der schwierigen Modellierung natürlicher Trennschichten zwischen einzelnen Bereichen der biologischen Zelle. Die Positionierung verschiedener Strukturen ineinander ist nicht möglich, hierarchisch angeordnete Membranen verlangen deshalb ein Konzept zur ebenen Darstellung. Dies führt insbesondere bei großen Modellen zu einem Erscheinungsbild, welches die Zuordnung einzelner biologischer Einheiten nicht ohne größeren Aufwand gestattet. Modellierungen von Interaktionen zwischen Membranen und die damit verbundenen Änderungen der hierarchischen Struktur verlangen darüberhinaus komplexe Mechanismen zur Nachbildung der Restrukturierung beteiligter Kompartimentinhalte. Aufgrund der Bedeutung, die Membranen und deren Interaktionen bei der Organisation verschiedenster Zellprozesse zukommt, wurden einige Prozesskalküle entwickelt, die durch angepasste Strukturen die Modellierung dieser Aspekte erleichtern.

Eine direkte Umsetzung der Membran-Maschine stellt der von Luca Cardelli eingeführte *Brane-Kalkül* ([13]) dar. Dieser bietet spezielle Möglichkeiten zur Modellierung hierarchischer Membranstrukturen und die direkte Umsetzung deren gegenseitiger Beeinflussung und abstrahiert vollständig von den jeder Membraninteraktion zugrunde liegenden Proteininteraktionen. Die Untersuchung seiner Möglichkeiten zur Modellierung membranbezogener biologischer Prozesse steht im Fokus dieser Arbeit. Anhand eines konkreten Beispiels aus der Biologie demonstrieren wir die Vielzahl der durch den Brane-Kalkül darstellbaren Effekte. Darüberhinaus arbeiten wir Aspekte heraus, deren Modellierung eine Erweiterung insbesondere um Konzepte der Protein-Maschine erfordert. Wir demonstrieren anhand gewählter Proteininteraktionen, dass sich der von Vincent Danos und Cosimo Laneve entwickelte κ -Kalkül ([23]), ein speziell für die Beschreibung von Proteininteraktionen konstruierter Ansatz, für die Darstellung von Interaktionen von auf der Membran befindlichen (*membranständigen*) Proteinen als Auslöser von Wechselwirkungen zwischen Membranen gut eignet. Seine Erweiterung zum *bio κ -Kalkül* ([30]) integriert darüberhinaus Membranstrukturen und einige ihrer Interaktionsmöglichkeiten in den Kalkül. Ausgehend von unserer Beispielmmodellierung diskutieren wir einige im bio κ -Kalkül nicht modellierbare Aspekte und entwickeln schließlich den Protein-Brane-Kalkül, welcher die Modellierung von Proteininteraktionen sowohl im Zusammenhang mit Membraninteraktionen als auch losgelöst davon zur Darstellung von Signaltransduktionswegen in den Brane-Kalkül integriert und damit umfassendere Modellierungen unter Wahrung der Vorteile der ursprünglichen Ansätze ermöglicht.

Wir stellen zunächst in knapper Form zwei in der Systembiologie zur Anwendung kommende Prozesskalküle vor, die neben der Abgrenzung von Kompartimenten auch Möglichkeiten zu deren Strukturmodifikation geben und damit für Modellierungen der Membranmaschine geeignet sind. Beide lehnen sich durch Angabe einer induktiv defi-

nierten termbasierten Syntax mit struktureller Kongruenz und operationaler Semantik stark an die Theorie des π -Kalküls an. Wir verzichten jedoch an dieser Stelle auf formale Definitionen und verweisen auf Kapitel 3, welches sich unter anderem der ausführlichen Vorstellung des Brane-Kalküls widmet, dessen Syntax und Semantik ebenfalls im starken Zusammenhang zum π -Kalkül steht.

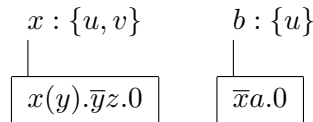
2.3.1 β -Binder

Die β -Binder ([44]) bauen auf einer reduzierten Variante des π -Kalküls auf, die wir im Folgenden in knapper und informeller Weise vorstellen. Grundvoraussetzung ist eine abzählbar unendliche Menge von *Namen*. Die einzelnen Bestandteile von Termen sind der *leere Prozess* 0 , welcher kein Verhalten aufweist, und zwei *Aktionspräfixe*: $\bar{x}y$ zum Senden eines Namens y über einen Kanal x (Kanäle sind ebenfalls Namen) und $x(z)$ zum Empfangen eines Namens über einen Kanal x , der alle Vorkommen von z im zugehörigen Teilprozess des Aktionspräfixes ersetzt. Terme lassen sich mit 0 beginnend aufbauen durch Anfügen von mit Punkten abgetrennten Aktionspräfixen und werden als *Prozesse* bezeichnet. Einen solchen Prozess beschreibt beispielsweise der Term $x(y).\bar{y}z.0$. Jeder Prozess lässt sich mittels \circ mit einem anderen Prozess verknüpfen und stellt wiederum einen Prozess dar. Von uns nicht näher betrachtete weitere Elemente sind die Deklaration privater Namen und die Replikation einzelner Prozesse. Die Kommunikation verschiedener mit \circ verknüpfter Prozesse verdeutlichen wir mit dem folgenden Ableitungsschritt:

$$x(y).\bar{y}z.0 \circ \bar{x}a.0 \longrightarrow \bar{a}z.0 \circ 0$$

Zwischen den beiden Teilprozessen $P = x(y).\bar{y}z.0$ und $Q = \bar{x}a.0$ kann eine Kommunikation stattfinden, Teilprozess Q sendet über den Kanal x den Namen a , welcher schließlich von P empfangen wird, weil dieser ebenfalls den Kanal x besitzt. Daraus resultiert innerhalb von P der Austausch von y mit dem empfangenen a . Die an der Kommunikation beteiligten Aktionspräfixe $x(y)$ und $\bar{x}a$ werden verbraucht, es entsteht der Prozess $R = \bar{a}z.0 \circ 0$. Ein Aktionspräfix kann nur dann ein Verhalten zeigen, wenn ihm kein anderes vorangestellt ist. Aktionspräfix $\bar{a}z$ in R könnte somit aktiv werden, $\bar{y}z$ in P hingegen noch nicht.

Die β -Binder erweitern diese reduzierte Variante des π -Kalküls um die Möglichkeit zur Einteilung verschiedener Prozesse in einzelne Kompartimente, die mithilfe getypter Schnittstellen miteinander kommunizieren können. Betrachten wir noch einmal die Prozesse P und Q , so können diese nun beispielsweise zwei verschiedenen Kompartimenten zugeteilt werden:



Jedes hier mit einem Rechteck dargestellte Kompartiment besitzt eine Schnittstelle x bzw. b , welche mit der Menge von Namen $\{u, v\}$ bzw. $\{u\}$ getypt ist, eine Einheit aus Schnittstelle und ihrer Typung wird als *Binder* bezeichnet. Eine Kommunikation zwischen P und Q kann nur dann stattfinden, wenn die Typmengen an den Schnittstellen

der zugehörigen Kompartimente nicht disjunkt sind. Dabei ist der konkrete Name der Schnittstelle nicht entscheidend, legt jedoch den Kanal innerhalb des Kompartiments fest, der zur Kommunikation nach außen fähig ist. Beispielsweise ist ein Ableitungsschritt der Art

$$\begin{array}{ccc}
 \begin{array}{c} x : \{u, v\} \\ | \\ \boxed{x(y).\bar{y}z.0} \end{array} & \begin{array}{c} b : \{u\} \\ | \\ \boxed{\bar{b}a.0} \end{array} & \longrightarrow & \begin{array}{c} x : \{u, v\} \\ | \\ \boxed{\bar{a}z.0} \end{array} & \begin{array}{c} b : \{u\} \\ | \\ \boxed{0} \end{array}
 \end{array}$$

möglich. Hierbei kommt es zur Kommunikation des Prozesses $P = x(y).\bar{y}z.0$ mit $Q' = \bar{b}a.0$, obwohl der sendende nicht mit dem empfangenden Kanal übereinstimmt. Die Termstruktur in den β -Bindern wird um drei weitere Präfixe ergänzt: $\exp(x, \Gamma)$ fügt dem umgebenden Kompartiment eine neue Schnittstelle mit dem Namen x und der Menge von Namen Γ hinzu, $\text{hid}(x)$ versteckt die Schnittstelle mit dem Namen x , $\text{unh}(x)$ macht die Schnittstelle x entsprechend sichtbar. Versteckte Schnittstellen sind für Interaktionen nicht mehr zugänglich und werden in runden Klammern dargestellt. Wir verdeutlichen die Anwendung am folgenden Beispiel, dabei steht R für einen beliebigen Prozess:

$$\begin{array}{ccc}
 \begin{array}{c} y : \{v\} \\ | \\ \boxed{\exp(x, \{u\}).\text{hid}(x).R} \end{array} & \longrightarrow & \begin{array}{c} y : \{v\} \quad x : \{u\} \\ | \quad | \\ \boxed{\text{hid}(x).R} \end{array} & \longrightarrow & \begin{array}{c} y : \{v\} \quad (x : \{u\}) \\ | \quad | \\ \boxed{R} \end{array}
 \end{array}$$

Die Interaktion verschiedener Kompartimente wird mittels zweier separat spezifizierter Funktionen f_{split} und f_{join} realisiert. Diese ermöglichen die Spaltung bzw. die Fusion von Kompartimenten und bieten ein hohes Maß an Freiheit in der Modellierung. Eine Fusion zweier Kompartimente kann nur dann stattfinden, wenn f_{split} diese für deren Binder und Inhalte explizit vorsieht. Die Binder des resultierenden Kompartiments können dabei völlig frei definiert werden. Analog lässt sich auch die Spaltung mittels f_{split} durch zahlreiche Restriktionsmöglichkeiten genau steuern.

Ineinander verschachtelte Strukturen sind auch mit den β -Bindern nicht darstellbar. Allerdings bietet die zusätzliche Kompartimentstruktur Möglichkeiten zu deren Nachempfindung mittels spezieller Typungen, einige Anwendungsbeispiele werden in [49] vorgestellt. Wir betrachten im Folgenden einen Prozesskalkül, der hierarchische Membrananordnungen auf direkte Weise umsetzt.

2.3.2 Bio-Ambients

Der *Ambienten-Kalkül* ([14]) ist ebenfalls aus dem π -Kalkül hervorgegangen. Sein Konzept der räumlichen Abgrenzung einzelner Einheiten, sogenannter *Ambienten*, ermöglicht bereits den Aufbau hierarchischer Strukturen und stellt einige Operationen zu deren Interaktion zur Verfügung. Die *Bio-Ambients* ([46]) stellen eine Adaption des Ambienten-Kalküls dar, die sich speziell auf die Modellierung systembiologisch relevanter Inhalte konzentriert.

Die Syntax der Bio-Ambients basiert auf einer leicht vereinfachten Variante der Syntax des π -Kalküls und erlaubt die hierarchische Abgrenzung verschiedener Funktionseinheiten. Zusätzlich zu den vom π -Kalkül bekannten Kommunikationspräfixen werden

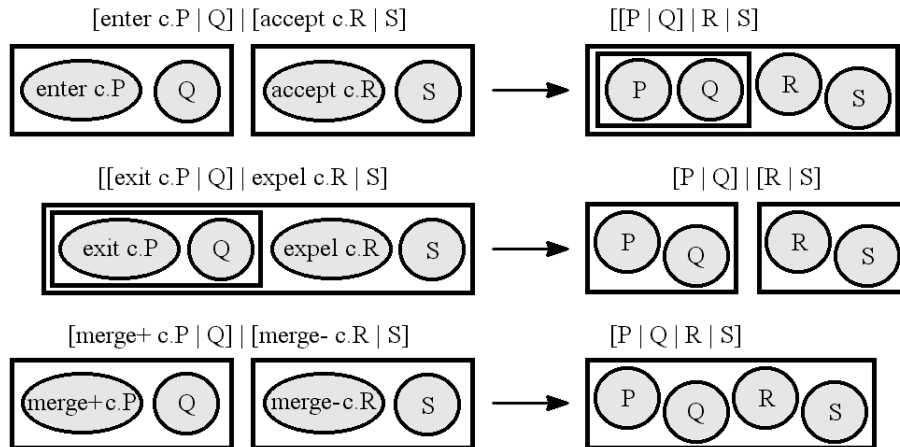


Abbildung 2.4: Operationen der Bio-Ambients, Grafik- und Termnotation (aus [49]).

weitere Präfixe zur Manipulation der Ambientenstruktur eingeführt, Abbildung 2.4 verdeutlicht die möglichen Interaktionen. Das Präfix *accept* innerhalb eines Ambienten gestattet zusammen mit dem Präfix *enter* innerhalb eines anderen das Eintreten des zweiten in den ersten, wie auch beim π -Kalkül werden die Präfixe nach erfolgter Interaktion verbraucht. Umgekehrt kann das Austreten mittels zweier Präfixe *exit* und *expel* realisiert werden, die Fusion zweier Ambienten erfolgt durch *merge+* und *merge-*. Im Gegensatz zu den β -Bindern gibt es keine direkte Möglichkeit zur Modellierung einer Membranspaltung.

Biologische Membranen stellen eine Abgrenzung ihres Inneren von ihrer Umgebung dar und bieten verschiedene Mechanismen zur Kommunikation durch Membranen hindurch. Die direkte Übermittlung eines Signals über mehrere Membrangrenzen hinweg ist jedoch nicht möglich. Die Bio-Ambients realisieren dies durch Einbettung zusätzlicher Informationen in die vom π -Kalkül bekannten Kommunikationspräfixe zur Festlegung bestimmter Arten von Kommunikation. So lässt sich mit *local* der Signalfluß innerhalb eines Ambienten angeben und mit *s2s* (sibling to sibling) die Kommunikation zweier Ambienten, die sich innerhalb desselben übergeordneten Ambienten befinden. Desweiteren ist das Senden von Informationen ausgehend von einem übergeordneten Ambient zu einem direkt eingebetteten Ambient mittels *p2c* (parent to child) und in umgekehrte Richtung mittels *c2p* (child to parent) möglich.

Die Bio-Ambients erlauben zwar die Darstellung der Endozytose mittels *enter/accept* und der Exozytose mittels *exit/expel*, bieten jedoch keine direkte Möglichkeit zur Modellierung der sich bei der Endozytose von der Membran des verschlingenden Systems abspaltenden neuen Membran und der damit verbundenen Bildung eines Vesikels um das verschlungene System, welches im Zusammenhang mit Zersetzungsprozessen bei der Modellierung von Membranvorgängen von Bedeutung sein kann. Auch die bei der Exozytose mit der Membran des ausspeienden Systems verschmelzende innere Membran kann beispielsweise zur Übertragung von Proteinen mit Rezeptorfunktion führen und deshalb einen relevanten Modellierungsaspekt darstellen. Wir betrachten im folgenden Kapitel einen Prozesskalkül, der diese Aspekte berücksichtigt und damit den speziellen Anforderungen der Membran-Maschine näher kommt.

Kapitel 3

Der Brane-Kalkül

Die Modellierung räumlich unterteilter Strukturen ist in dem sehr allgemein gehaltenen Ansatz des π -Kalküls nur mit großem Aufwand und einem erheblichen Verlust an Übersichtlichkeit realisierbar. Insbesondere zur Darstellung komplexer Membranstrukturen und deren Interaktionen scheint er somit schlecht geeignet. Luca Cardelli schlug 2004 einen weniger allgemeinen, dafür aber speziell auf die Modellierung der Membran-Maschine zugeschnittenen Prozesskalkül vor ([13]). Dieser als Brane-Kalkül bezeichnete Ansatz realisiert die strikte Trennung von Funktionalitäten und Inhalten verschiedener Membranen auf biologisch glaubwürdige Weise und stellt deren komplexe Interaktionsmöglichkeiten in den Mittelpunkt.

Anders als die β -Binder und die Bio-Ambients betrachtet der Brane-Kalkül Membranen nicht als passive Einheiten zur Abtrennung räumlicher Bereiche, sondern modelliert sie als aktive Elemente der Zellmaschinerie. Informationen zu Interaktionen befinden sich direkt auf den Membranen und ermöglichen damit gezieltere Verknüpfungen einzelner Funktionen mit den dafür verantwortlichen Bereichen. Diese Modellierung befindet sich sehr nah an der intuitiven Auffassung des biologischen Systems, weil Membranen auch in der Natur mithilfe der an ihnen angelagerten Proteine als aktive Initiatoren von Interaktionen verschiedener Art wirken. In diesem Kapitel stellen wir die Syntax und Semantik des Brane-Kalküls ausführlich vor, gehen kurz auf einige weitere Arbeiten zum Brane-Kalkül ein und demonstrieren seine Anwendbarkeit anhand einer umfassenden Modellierung der Effekte einer Zellinfektion durch HIV.

3.1 Syntax

Die durch keine Funktionalität charakterisierte Einheit und jedes Molekül wird im Brane-Kalkül als System bezeichnet. Darauf aufbauend ist ein von einer Membran umschlossenes System und eine Ansammlung von Systemen stets wiederum ein System. Jede Membran ist gekennzeichnet durch einen mit ihr verknüpften Prozess. Dieser bezeichnet entweder eine Aktion oder eine Zusammensetzung von hintereinander oder parallel ausführbaren Prozessen. Wir geben nun eine formale Definition der Syntax.

Definition 3.1.1 (Brane-Kalkül-Syntax) Die Menge der *Systeme* im Brane-Kalkül wird mit \mathcal{S}_b notiert und ist gegeben durch die folgende Grammatik:

$$S ::= \diamond \mid m \mid S \circ S \mid !S \mid \sigma(S).$$

Dabei steht m für beliebige Elemente aus der Menge \mathcal{M} der *Molekülnamen* und σ für beliebige *Prozesse* gemäß der Grammatik

$$\sigma ::= 0 \mid \sigma \mid \sigma \mid !\sigma \mid a.\sigma,$$

wobei a beliebige *Aktionen* gemäß den zunächst nicht näher spezifizierten Grammatiken a_{pep} , a_{mbd} und $a_{b\&r}$ bezeichnet. Die Menge der Prozesse notieren wir mit \mathcal{P}_b . \square

In dieser Arbeit werden wir Systeme im Sinne der obigen Definition mit lateinischen Großbuchstaben, im Fall von Molekülen oder Molekülkompositionen auch mit lateinischen Kleinbuchstaben, und Prozesse mit griechischen Kleinbuchstaben notieren. Im Folgenden erläutern wir die einzelnen Bestandteile der Syntax näher.

- \diamond bezeichnet das *leere System*. Dieses ist von keiner Membran umgeben und damit auch zu keiner Interaktion fähig.
- $m \in \mathcal{M}$ bezeichnet ein Molekül durch Angabe seines Namens.
- $S \circ T$ bezeichnet ein System, welches sich durch *System-Komposition* aus zwei parallel agierenden Systemen S und T zusammensetzt.
- $!S$ notiert mithilfe des *Replikationsoperators* “!” ein System, welches aus der Komposition unendlich vieler Systeme S besteht.
- $\sigma(S)$ kennzeichnet die Umrahmung eines Systems S mit einer durch spezielle Klammern ausgedrückten Membran, welche einen Prozess σ besitzt.
- 0 bezeichnet den *leeren Prozess*, eine mit 0 verknüpfte Membran zeigt also keinerlei Verhalten.
- $\sigma \mid \tau$ bezeichnet einen Prozess, welcher sich durch *Prozess-Komposition* aus den Prozessen σ und τ zusammensetzt. Diese können unabhängig voneinander gleichberechtigt zur Interaktion führen.
- $!\sigma$ bezeichnet eine Komposition aus beliebig vielen Prozessen σ .
- $a.\sigma$ bezeichnet einen Prozess, welcher sich nach Auslösen der Aktion a wie σ verhält. Es handelt sich um eine Hintereinanderausführung, σ kann also erst nach Auslösung von a aktiv werden.

Eine Definition von Präzedenzregeln für die einzelnen Operatoren ist sinnvoll, um durch geeignetes Einsparen von Klammern die Übersichtlichkeit zu erhöhen. Wir legen die folgende Reihenfolge fest: den größten Vorrang erhält die Aktionsverkettung “.”, mit jeweils niedrigerem Vorrang folgen der Replikationsoperator “!”, die Prozess-Komposition “|”, die implizit notierte Verknüpfung von Prozess und Membran (z.B. $\sigma(P)$) und die System-Komposition “ \circ ”.

Wir merken an, dass in der Definition der Brane-Kalkül-Syntax zwei verschiedene Replikationsoperatoren für Systeme und Prozesse notwendig wären. Aus Übersichtsgründen und weil eine Verwechslung aufgrund der Präzedenzregeln ausgeschlossen ist, verzichten wir jedoch auf diese Unterscheidung.

Beispiel 3.1.2 (Präzedenzregeln) Wir betrachten das System

$$R = !(\tau(P)) \circ !a.\sigma_1 | \sigma_2(Q).$$

Es setzt sich aus den zwei Teilsystemen $!(\tau(P))$ und $!a.\sigma_1 | \sigma_2(Q)$ zusammen. Das erste ist wiederum eine Komposition von unendlich vielen Systemen der Gestalt $\tau(P)$. Das zweite besitzt den Prozess $!a.\sigma_1 | \sigma_2$, welcher sich aus den Einzelprozessen $!a.\sigma_1$ und σ_2 zusammensetzt. Eine ausführlichere Klammerung von R führt zu dem folgenden zu R äquivalenten System

$$R' = (!(\tau(P))) \circ (((!(a.\sigma_1)) | \sigma_2)(Q)). \quad \square$$

Wir vereinbaren ein paar Kurzschreibweisen, um weiter die Übersichtlichkeit zu erhöhen. So schreiben wir a für $a.0$, (P) für $0(P)$ und $\sigma(\diamond)$ für $\sigma(\diamond)$.

3.2 Strukturelle Kongruenz

Die Brane-Kalkül-Syntax ermöglicht durch die exakte Beschreibung eines Systems die eindeutige Spezifikation dessen (an späterer Stelle definierten) Verhaltens. Darüber hinaus ist gewünscht, dass auch einige verschiedene Systeme das gleiche Verhalten zeigen. Beispielsweise erscheint bei der Komposition eine Gleichberechtigung der einzelnen Teilsysteme oder Teilprozesse sinnvoll, unterschiedliche Positionierung sollte also keinen Einfluss auf das Verhalten des Gesamtsystems haben. Ebensovienig wollen wir ein System mit untätigem Teilprozess von einem System unterscheiden, welches bis auf das Fehlen dieses untätigen Teilprozesses dem ersten gleicht. Wir führen eine Kongruenzbeziehung zwischen Systemen ein, um den nach unserer intuitiven Vorstellung gleichen Systemen später auch das gleiche Verhalten zuordnen können.

Definition 3.2.1 (Strukturelle Kongruenz) Sei $\equiv_{pr} \subseteq \mathcal{P}_b \times \mathcal{P}_b$ die kleinste Äquivalenzrelation mit den Eigenschaften (1') bis (11'). Die *Strukturelle Kongruenz* $\equiv \subseteq \mathcal{S}_b \times \mathcal{S}_b$ ist die kleinste Äquivalenzrelation, welche (1) bis (11) erfüllt.

- | | |
|---|---|
| <p>(1) $P \circ Q \equiv Q \circ P$</p> <p>(2) $P \circ (Q \circ R) \equiv (P \circ Q) \circ R$</p> <p>(3) $P \circ \diamond \equiv P$</p> <p>(4) $!\diamond \equiv \diamond$</p> <p>(5) $!(P \circ Q) \equiv !P \circ !Q$</p> <p>(6) $!!P \equiv !P$</p> <p>(7) $!P \equiv P \circ !P$</p> <p>(8) $0(\diamond) \equiv \diamond$</p> <p>(9) $P \equiv Q \Rightarrow P \circ R \equiv Q \circ R$</p> <p>(10) $P \equiv Q \Rightarrow !P \equiv !Q$</p> <p>(11) $P \equiv Q \wedge \sigma \equiv_{pr} \tau \Rightarrow \sigma(P) \equiv \tau(Q)$</p> | <p>(1') $\sigma \tau \equiv_{pr} \tau \sigma$</p> <p>(2') $\sigma (\tau \varrho) \equiv_{pr} (\sigma \tau) \varrho$</p> <p>(3') $\sigma 0 \equiv_{pr} \sigma$</p> <p>(4') $!0 \equiv_{pr} 0$</p> <p>(5') $!(\sigma \tau) \equiv_{pr} !\sigma !\tau$</p> <p>(6') $!!\sigma \equiv_{pr} !\sigma$</p> <p>(7') $!\sigma \equiv_{pr} \sigma !\sigma$</p> <p>(9') $\sigma \equiv_{pr} \tau \Rightarrow \sigma \varrho \equiv_{pr} \tau \varrho$</p> <p>(10') $\sigma \equiv_{pr} \tau \Rightarrow !\sigma \equiv_{pr} !\tau$</p> <p>(11') $\sigma \equiv_{pr} \tau \Rightarrow a.\sigma \equiv_{pr} a.\tau$</p> |
|---|---|
-

Die Regeln (1) bis (3) modellieren die Eigenschaften Kommutativität, Assoziativität und das neutrale Element des binären Operators “ \circ ”. (4) bis (7) legen die Funktionalität des Replikationsoperators fest, wobei (7) die eigentliche Replikation in Form von Erzeugung eines neuen Systems ermöglicht. (8) schließlich beseitigt eine Membran, die das Systemverhalten nicht beeinflusst.

Die Regeln (9) bis (11) erweitern \equiv zur Kongruenzrelation. Sie gewährleisten, dass auch komplexere Systeme als kongruent gelten, wenn deren Teilsysteme kongruent sind. Bei Regel (11) ist dabei insbesondere die Kongruenzrelation \equiv_{pr} für Prozesse zu beachten, die benötigt wird, um zwei Systeme mit Membranen als kongruent zu betrachten, wenn deren Membranzinhalt und deren Prozesse kongruent sind.

Die Regeln (1') bis (11') sind ähnlich zu (1) bis (11) und werden deshalb nicht näher erläutert.

Beispiel 3.2.2 (kongruente Terme) Wir betrachten das System S und eine Reihe von Umformungsschritten gemäß Definition 3.2.1. Die jeweils angewandte Regel wird hinter der entsprechenden Umformung notiert, dabei verzichten wir jedoch auf die Angabe von zusätzlich benötigten Regeln aus (9) bis (11).

$$\begin{aligned}
S &= \tau|!\sigma(!(!Q \circ 0(\diamond))) \\
&\equiv \tau|!\sigma(!(!Q \circ \diamond)) & (8) \\
&\equiv \tau|!\sigma(!!Q \circ !\diamond) & (5) \\
&\equiv \tau|!\sigma(!!Q \circ \diamond) & (4) \\
&\equiv \tau|!\sigma(!Q \circ \diamond) & (6) \\
&\equiv \tau|!\sigma(!Q) & (3) \\
&\equiv \tau|(\sigma|\sigma)(!Q) & (7') \\
&\equiv (\tau|\sigma)|!\sigma(!Q) & (2') \\
&\equiv (\sigma|\tau)|!\sigma(!Q) & (1')
\end{aligned}$$

Aufgrund der Transitivität von \equiv sind die Systeme aller Zeilen paarweise kongruent. \square

Jede Äquivalenzklasse von \equiv enthält nun genau die Systeme, die gemäß unserer intuitiven Vorstellung das gleiche Verhalten aufweisen sollen. Darüber hinaus wird die Semantik des Replikationsoperators definiert. Die Kongruenzrelation kapselt damit bereits einige wichtige semantische Elemente und ermöglicht eine übersichtliche Definition der Reduktionsrelation des Brane-Kalküls basierend auf wenigen Regeln.

3.3 Reduktionsrelation

Bevor wir zu den grundlegenden Regeln der Reduktionsrelation für die Modellierung von Membraninteraktionen kommen, wollen wir einige darauf aufbauende Regeln betrachten, welche die Reduktion eines Gesamtsystems gestatten, wenn lediglich für einen Teil des Systems eine Reduktionsregel vorliegt. Es wird also gewährleistet, dass eine Reduktion auch innerhalb eines sich dabei nicht verändernden Kontextes stattfinden kann. Eine weitere Regel stellt sicher, dass gemäß \equiv kongruente Systeme das gleiche Verhalten zeigen. Wir notieren jeweils oberhalb des horizontalen Striches die Bedingungen, die erfüllt sein müssen, damit die Reduktion darunter stattfinden kann.

Definition 3.3.1 (Reduktionsrelation \longrightarrow) Seien \longrightarrow_{pep} , \longrightarrow_{mbd} und $\longrightarrow_{b\&r}$ drei zunächst nicht näher spezifizierte Relationen. $\longrightarrow \subseteq \mathcal{S}_b \times \mathcal{S}_b$ ist die kleinste Relation mit $\longrightarrow_{pep} \cup \longrightarrow_{mbd} \cup \longrightarrow_{b\&r} \subseteq \longrightarrow$, die folgende Reduktionsregeln erfüllt:

$$\text{PAR} \frac{P \longrightarrow Q}{P \circ R \longrightarrow Q \circ R} \quad \text{BRANE} \frac{P \longrightarrow Q}{\sigma(P) \longrightarrow \sigma(Q)}$$

$$\text{STRUCT} \frac{P' \equiv P, \quad P \longrightarrow Q, \quad Q \equiv Q'}{P' \longrightarrow Q'}$$

Wir nennen \longrightarrow die *Reduktionsrelation* des Brane-Kalküls und ihre Elemente *Ableitungsschritte*. Eine endliche Folge $S_1 \longrightarrow S_2, \dots, S_{l-1} \longrightarrow S_l$ bezeichnen wir auch als *Reduktionsableitung* und notieren sie in der Form $S_1 \longrightarrow S_2 \longrightarrow \dots \longrightarrow S_{l-1} \longrightarrow S_l$. Mit \longrightarrow^* notieren wir die reflexive, transitive Hülle von \longrightarrow . \square

PAR: Diese Regel stellt sicher, dass es auch innerhalb einer System-Komposition zur Reduktion kommen kann.

BRANE: Ähnlich zu PAR ermöglicht diese Regel die Reduktion eines Systems, welches von einer Membran umschlossen ist.

STRUCT: Durch diese Regel wird die Einflechtung der strukturellen Kongruenz in die Reduktionsrelation realisiert. Sie stellt sicher, dass bezüglich \equiv kongruente Systeme das gleiche Ableitungsverhalten zeigen.

Beispiel 3.3.2 (kontextuelle Erweiterungen von \longrightarrow) Angenommen, ein System P kann in einem Schritt zu P' abgeleitet werden. Durch Anwendung der in Definition 3.3.1 angegebenen Regeln ergibt sich unter Nutzung von \equiv beispielsweise Folgendes.

$$\begin{aligned} & P \longrightarrow P' \\ \Rightarrow & P \circ Q \longrightarrow P' \circ Q && \text{(PAR)} \\ \Rightarrow & Q \circ P \longrightarrow Q \circ P' && \text{(STRUCT)} \\ \Rightarrow & \sigma(Q \circ P) \longrightarrow \sigma(Q \circ P') && \text{(BRANE)} \end{aligned}$$

Also kann auch $\sigma(Q \circ P)$ in einem Schritt zu $\sigma(Q \circ P')$ abgeleitet werden. \square

Die bisher vorgestellte Reduktionsrelation ist unvollständig, denn wir haben noch keine Regeln angegeben, die einen Ableitungsschritt ermöglichen, ohne wiederum einen Ableitungsschritt vorauszusetzen. In den folgenden Abschnitten spezifizieren wir die grundlegenden Relationen \longrightarrow_{pep} , \longrightarrow_{mbd} und $\longrightarrow_{b\&r}$, aus denen sich die Reduktionsrelation aufbaut.

3.3.1 Phago/Exo/Pino

In diesem Abschnitt widmen wir uns den Operationen Exozytose und Endozytose. Die Modellierung erfolgt mittels dreier Operationen. Während die Exozytose direkt als Operation *Exo* umgesetzt werden kann, wird die Endozytose in zwei Operationen aufgeteilt, *Phago* zum Verschlingen von Strukturen mit exakt einer äußeren Membran und

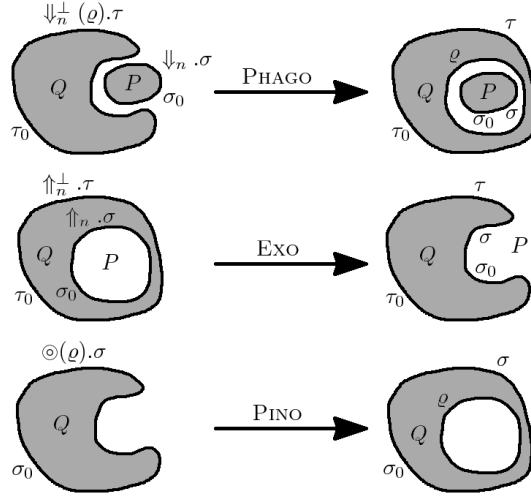


Abbildung 3.1: Die Reduktionsregeln PHAGO, EXO und PINO.

Pino zum Verschlingen von leeren Strukturen. Diese Trennung ermöglicht eine bessere Kontrolle der Modellierung bezüglich der Art der Strukturen, die bei der Endozytose aufgenommen werden.

Die Definition der Membranoperationen setzt zunächst die Spezifikation weiterer Komponenten der Brane-Kalkül-Syntax voraus. Wir werden hier nun die Aktionen angeben, die zur Modellierung von Phago, Exo und Pino benötigt werden. Diese sind einer der fehlenden Bestandteile der Brane-Kalkül-Syntax aus Definition 3.1.1.

Definition 3.3.3 (Aktionen a_{pep}) Sei $n \in \mathbb{N}$. In der Grammatik

$$a_{pep} ::= \downarrow_n \mid \downarrow_n^\perp(\sigma) \mid \uparrow_n \mid \uparrow_n^\perp \mid \odot(\sigma)$$

bezeichnen \downarrow_n bzw. $\downarrow_n^\perp(\sigma)$ die Aktion bzw. Ko-Aktion von Phago, \uparrow_n bzw. \uparrow_n^\perp die Aktion bzw. Ko-Aktion von Exo und $\odot(\sigma)$ die Aktion von Pino. \square

Die Operationen Phago bzw. Exo benötigen jeweils zu ihrer entsprechenden Aktion eine weitere Ko-Aktion zur Synchronisation. Diese kennzeichnet das mit ihr verknüpfte System als einen der Interaktionspartner, welcher später das die Aktion enthaltende System verschlingen bzw. auspeien kann. Indizes $n \in \mathbb{N}$ werden verwendet, um die Zugehörigkeit solcher Paare auszudrücken. Dabei verwenden wir o.B.d.A. auch beliebige Namen oder verzichten auf Indizes, wenn dadurch keine Mehrdeutigkeit entsteht.

Wir können nun die ersten grundlegenden Elemente der Reduktionsrelation aus Definition 3.3.1 angeben, die letztlich die Interaktion zwischen einzelnen Systemen des Brane-Kalküls gestatten. Eine grafische Repräsentation der folgenden Regeln ist in Abbildung 3.1 zu sehen.

Definition 3.3.4 (Relation \longrightarrow_{pep}) Sei $n \in \mathbb{N}$. \longrightarrow_{pep} ist die kleinste Relation, die folgende Reduktionsaxiome erfüllt:

$$\text{PHAGO} \frac{}{\downarrow_n.\sigma|\sigma_0(P) \circ \downarrow_n^\perp(\varrho).\tau|\tau_0(Q) \longrightarrow \tau|\tau_0(\varrho|\sigma|\sigma_0(P)) \circ Q}$$

$$\text{EXO} \frac{\uparrow_n^\perp . \tau | \tau_0 (\uparrow_n . \sigma | \sigma_0 (P) \circ Q) \longrightarrow P \circ \sigma | \sigma_0 | \tau | \tau_0 (Q)}{\uparrow_n^\perp . \tau | \tau_0 (\uparrow_n . \sigma | \sigma_0 (P) \circ Q) \longrightarrow P \circ \sigma | \sigma_0 | \tau | \tau_0 (Q)}$$

$$\text{PINO} \frac{\odot(\varrho) . \sigma | \sigma_0 (Q) \longrightarrow \sigma | \sigma_0 (\varrho(\diamond) \circ Q)}{\odot(\varrho) . \sigma | \sigma_0 (Q) \longrightarrow \sigma | \sigma_0 (\varrho(\diamond) \circ Q)} \quad \square$$

Allgemein lässt sich feststellen, dass bei jeder Interaktion die an ihr beteiligten Aktionen aus den jeweiligen Prozessen verschwinden und die übrig bleibenden Teilprozesse ihrer jeweiligen Membran zugeordnet bleiben. Wir betrachten die einzelnen Reduktionsregeln nun etwas genauer.

PHAGO: Diese Regel realisiert das Verschlingen eines Systems $\sigma | \sigma_0 (P)$. An der Ko-Aktion von Phago ist zusätzlich ein Prozess ϱ notiert, welcher der durch das Verschlingen neu entstehenden Membran zugeordnet wird. Damit liegt die Verantwortung bezüglich der Funktionalität der neu entstehenden Membran bei dem verschlingenden System. Dies entspricht dem Ablauf der Endozytose in der Natur, denn dort bildet sich die neue Membran durch Abspaltung von der Membran des verschlingenden Systems und kann somit einen Teil von deren funktionalen Einheiten übernehmen.

EXO: Diese Regel führt zum Ausspeien eines Systems P . Dabei verschwindet die P umgebende Membran. Deren verbleibender Prozess wird der Membran des ausspeienden Systems zugeordnet. Auch dies entspricht dem natürlichen Ablauf, denn umgekehrt zur Endozytose verschmilzt hier die innere Membran mit der äußeren und reichert sie dadurch um ihre funktionalen Einheiten an.

PINO: Ähnlich zu Phago ist an Pino ein Prozess ϱ notiert, welcher der neu entstehenden Membran zugeordnet wird. Im Unterschied zu den beiden anderen Operationen benötigt Pino jedoch keine Ko-Aktion, weil das daran beteiligte System lediglich ein leeres System verschlingt und sich deshalb mit keinem weiteren Interaktionspartner synchronisieren muss.

Wir sehen nun, warum die Endozytose in die zwei Operationen Phago und Pino aufgeteilt werden muss. Würde man jegliche Endozytose ähnlich zu Pino ohne Ko-Aktion modellieren, hätte man keine Kontrolle über die Art der Strukturen, die ein System zu verschlingen in der Lage sein soll. Jedes zur Phagozytose fähige System könnte also beliebige Systeme verschlingen. Würde man andererseits Endozytose ähnlich zu Phago mit Ko-Aktion modellieren, gäbe es keine Möglichkeit zur Modellierung des Verschlingens von leerem Inhalt, da diesem keine Aktion zugeordnet werden kann. Für ein besseres Verständnis des Zusammenspiels der drei Operationen folgt nun ein Beispiel.

Beispiel 3.3.5 (Reduktionsableitung) Wir betrachten eine Reduktionsableitung eines Systems S . Dabei notieren wir die beim jeweiligen Ableitungsschritt angewendete

Aktion über dem Relationssymbol.

$$\begin{aligned}
S &= \odot(\Downarrow_n^\perp(\varrho). \Uparrow_{n'}). \Uparrow_{n'}^\perp . \tau | \tau_0(\Downarrow_n . \sigma | \sigma_0(P) \circ Q) \\
&\xrightarrow{\odot} \Uparrow_{n'}^\perp . \tau | \tau_0(\Downarrow_n^\perp(\varrho). \Uparrow_{n'}(\diamond) \circ \Downarrow_n . \sigma | \sigma_0(P) \circ Q) \\
&\equiv \Uparrow_{n'}^\perp . \tau | \tau_0(\Downarrow_n . \sigma | \sigma_0(P) \circ \Downarrow_n^\perp(\varrho). \Uparrow_{n'}(\diamond) \circ Q) \\
&\xrightarrow{\Downarrow_n} \Uparrow_{n'}^\perp . \tau | \tau_0(\Uparrow_{n'}(\varrho(\sigma | \sigma_0(P))) \circ \diamond) \circ Q) \\
&\equiv \Uparrow_{n'}^\perp . \tau | \tau_0(\Uparrow_{n'}(\varrho(\sigma | \sigma_0(P)))) \circ Q) \\
&\xrightarrow{\Uparrow_{n'}} \varrho(\sigma | \sigma_0(P)) \circ \tau | \tau_0(Q)
\end{aligned}$$

Das im Gesamtsystem S enthaltene System $\sigma | \sigma_0(P)$ wird also mittels dreier Ableitungsschritte vom ebenfalls in S enthaltenen System Q unter Teilung der S umgebenden Membran abgespalten. \square

3.3.2 Mate/Bud/Drip

Neben der Exozytose und der Endozytose betrachtet die Membran-Maschine noch zwei weitere Operationen, Membranfusion und Membranspaltung. Während Membranfusion direkt mit einer Operation *Mate* umgesetzt wird, muss ähnlich zur Endozytose auch die Membranspaltung mittels zweier verschiedener Operationen modelliert werden, *Bud* zum Abspalten eines Inhalts mit exakt einer äußeren Membran und *Drip* zur Abspaltung eines leeren Inhalts.

Wir fügen nun der in Definition 3.1.1 gegebenen Brane-Kalkül-Syntax die Grammatik der zur Modellierung von Mate, Bud und Drip benötigten Aktionen hinzu.

Definition 3.3.6 (Aktionen a_{mbd}) Sei $n \in \mathbb{N}$. In der Grammatik

$$a_{mbd} ::= \mathfrak{mt}_n \mid \mathfrak{mt}_n^\perp \mid \mathfrak{bd}_n \mid \mathfrak{bd}_n^\perp(\sigma) \mid \mathfrak{dr}(\sigma)$$

bezeichnen \mathfrak{mt}_n bzw. \mathfrak{mt}_n^\perp die Aktion bzw. Ko-Aktion von Mate, \mathfrak{bd}_n bzw. $\mathfrak{bd}_n^\perp(\sigma)$ die Aktion bzw. Ko-Aktion von Bud und $\mathfrak{dr}(\sigma)$ die Aktion von Drip. \square

Wie schon in Definition 3.3.3 verwenden wir Aktionen und Ko-Aktionen zur Synchronisation und Indizes für deren eindeutige Zuordnung. Wir geben nun weitere Elemente der Reduktionsrelation von Definition 3.3.1 an, deren Regeln in Abbildung 3.2 auch grafisch dargestellt sind.

Definition 3.3.7 (Relation \longrightarrow_{mbd}) Sei $n \in \mathbb{N}$. \longrightarrow_{mbd} ist die kleinste Relation, die folgende Reduktionsaxiome erfüllt:

$$\text{MATE} \frac{}{\mathfrak{mt}_n . \sigma | \sigma_0(P) \circ \mathfrak{mt}_n^\perp . \tau | \tau_0(Q) \longrightarrow \sigma | \sigma_0 | \tau | \tau_0(P \circ Q)}$$

$$\text{BUD} \frac{}{\mathfrak{bd}_n^\perp(\varrho) . \tau | \tau_0(\mathfrak{bd}_n . \sigma | \sigma_0(P) \circ Q) \longrightarrow \varrho(\sigma | \sigma_0(P)) \circ \tau | \tau_0(Q)}$$

$$\text{DRIP} \frac{}{\mathfrak{dr}(\varrho) . \sigma | \sigma_0(Q) \longrightarrow \varrho(\diamond) \circ \sigma | \sigma_0(Q)} \quad \square$$

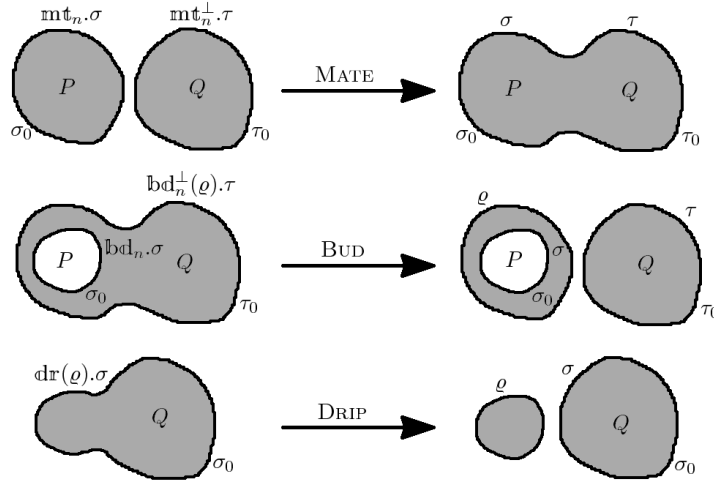


Abbildung 3.2: Die Reduktionsregeln MATE, BUD und DRIP.

MATE: Diese Regel führt zum Verschmelzen zweier Systeme, wobei auch hier gemäß dem natürlichen Ablauf der Membranverschmelzung die Prozesse beider Membranen auf die resultierende gemeinsame Membran übertragen werden.

BUD: Diese Regel führt zum Abspalten eines im Gesamtsystem enthaltenen Systems $\sigma|\sigma_0(P)$. Dabei wird die Membran des Gesamtsystems geteilt und der abgespaltene Teil mit dem Prozess ϱ versehen, dem natürlichen Ablauf entsprechend ordnet also auch hier das Gesamtsystem der neu entstehenden Membran seine Funktionalität zu.

DRIP: Ähnlich zu BUD kommt es zur Abspaltung einer Membran mit einem ihr zugeordneten Prozess ϱ , der abzuspaltende Inhalt ist jedoch leer und benötigt deshalb keine Ko-Aktion.

Wir stellen fest, dass das in Beispiel 3.3.5 abgeleitete System die Form der rechten Seite der Regel BUD aufweist. Substituieren wir im dortigen Anfangssystem S die Aktion \Downarrow_n mit bd_n und $\odot(\Downarrow_n^\perp(\varrho) \cdot \Uparrow_{n'}) \cdot \Uparrow_{n'}$ mit $\text{bd}_n^\perp(\varrho)$, stimmt dieses zudem mit der linken Seite der Regel BUD überein. Die Beispielableitung zeigt also, dass sich die Regel BUD auch als Kombination der Regeln PHAGO, EXO und PINO ausdrücken lässt. Bei den Regeln MATE und DRIP lassen sich ähnliche Zusammenhänge feststellen. Wir fassen zusammen.

Beobachtung 3.3.8 Mithilfe der Gleichsetzungen

$$\begin{aligned} \text{mt}_n &= \Downarrow_n \cdot \Uparrow_{n'} & \text{mt}_n^\perp &= \Downarrow_n^\perp (\Uparrow_{n'}^\perp \cdot \Uparrow_{n''}) \cdot \Uparrow_{n''} \\ \text{bd}_n &= \Downarrow_n & \text{bd}_n^\perp(\varrho) &= \odot(\Downarrow_n^\perp(\varrho) \cdot \Uparrow_{n'}) \cdot \Uparrow_{n'} \\ \text{dr}_n(\varrho) &= \odot(\odot(\varrho) \cdot \Uparrow_n) \cdot \Uparrow_n^\perp \end{aligned}$$

lassen sich die Regeln MATE, BUD und DRIP (unter zusätzlicher Einführung eines Indexes für die Aktion von DRIP) auch als Kombinationen der Regeln PHAGO, EXO und PINO ausdrücken. Die Erweiterung von \longrightarrow um die Reduktionsaxiome MATE, BUD

und DRIP führt also lediglich zu Reduktionsableitungen, die mithilfe der obengenannten Gleichsetzungen auch schon durch die Reduktionsaxiome PHAGO, EXO und PINO erreichbar sind. \square

Wir könnten auf die Definition 3.3.7 und die mit ihr zusammenhängenden Aktionen von Definition 3.3.6 verzichten, da diese die Ausdrucksstärke des Brane-Kalküls nicht erhöhen. Allerdings setzen sich Membranfusion und Membranspaltung in der Natur nicht aus Exozytose- und Endozytose-Schritten zusammen, sondern werden auf direktem Wege durchgeführt. Um biologisch plausibel zu bleiben, verwenden wir MATE, BUD und DRIP deshalb als eigenständige Reduktionsaxiome. Beobachtung 3.3.8 sollte also als rein theoretische Überlegung bezüglich der Ausdrucksstärke des Brane-Kalküls verstanden werden.

3.3.3 Bind and Release

Bisher haben wir uns auf Interaktionen von Membranen untereinander konzentriert. Darüber hinaus haben sie jedoch weitere Aufgaben. Eine wesentliche Funktion ist der Transport von Molekülen durch Membranen hindurch zur Realisierung des Stoffwechsels. Durch die Abgrenzung der Zelle in einzelne Bereiche beschleunigen sie aber auch chemische Reaktionen, da entsprechende Reaktionspartner räumlich nah zueinander gebracht werden und eine Störung der Abläufe durch unbeteiligte Moleküle verhindert wird. Wir lernen nun eine Operation kennen, die eine Modellierung der genannten Aufgaben gestattet. Weil sie den Fähigkeiten natürlicher Membranen zum Binden und Loslassen von Molekülen nachempfunden ist, wird sie *Bind and Release* genannt und im Folgenden durch B&R abgekürzt. Mit der folgenden Grammatik vervollständigen wir die in Definition 3.1.1 gegebene Syntax des Brane-Kalküls.

Definition 3.3.9 (Aktion $a_{b\&r}$) Die Aktion von B&R ist gegeben durch die Grammatik

$$a_{b\&r} ::= \mathbb{b}\mathbb{r} \left\langle \begin{array}{c} p(p) \\ p(p) \end{array} \right\rangle,$$

dabei bezeichnet p eine aus Molekülen $m \in \mathcal{M}$ bestehende *Molekül-Komposition* gemäß der Grammatik

$$p ::= \diamond \mid m \mid m \circ p. \quad \square$$

Die Aktion $\mathbb{b}\mathbb{r} \left\langle \begin{array}{c} p(q) \\ p'(q') \end{array} \right\rangle$ beinhaltet die Information, dass eine Molekülkomposition p an der Außenseite und eine Molekülkomposition q an der Innenseite einer Membran miteinander reagieren können, wobei die Komposition p' an der Außenseite und q' an der Innenseite entsteht. Man beachte, dass auch das leere System als Komposition von Molekülen zugelassen ist und damit das Verschwinden und Entstehen von Molekülen an beiden Seiten der Membran erlaubt. Wir geben nun die letzte fehlende Komponente der Reduktionsrelation von Definition 3.3.1 an und realisieren damit die eben formulierte Interaktion im Brane-Kalkül.

Definition 3.3.10 (Relation $\longrightarrow_{b\&r}$) $\longrightarrow_{b\&r}$ ist die kleinste Relation, die folgendes Reduktionsaxiom erfüllt:

$$\text{B\&R} \frac{}{p \circ \mathbb{b}\mathbb{r} \left\langle \begin{array}{c} p(q) \\ p'(q') \end{array} \right\rangle . \sigma | \sigma_0 (q \circ P) \longrightarrow p' \circ \sigma | \sigma_0 (q' \circ P)} \quad \square$$

Es folgen einige Beispiele zur Verdeutlichung der Funktionsweise und Anwendungsmöglichkeiten von B&R.

Beispiel 3.3.11 (Aktivierung) In realen Zellen wird die Veränderung eines Verhaltens oft durch das Einwirken bestimmter Stoffe erzielt. Beispielsweise manipulieren Enzyme durch kurzfristiges Andocken an membranständige Proteine die Rezeptoren einer Zelle und beeinflussen damit die Eigenschaften der Zellmembran, ohne ihre eigene Struktur zu ändern. Im Brane-Kalkül kann die Aktivierung einer Funktionalität a in einem System $\langle S \rangle$ durch eine sich nicht verändernde Molekül-Komposition p folgendermaßen modelliert werden:

$$\begin{aligned} & p \circ \mathbb{b}_R \left\langle \begin{array}{c} p(\diamond) \\ p(\diamond) \end{array} \right\rangle . a(\langle S \rangle) \\ \xrightarrow{\mathbb{b}_R} & p \circ a.(\langle S \rangle) \end{aligned}$$

Die Modellierung der Deaktivierung einer Aktion im Brane-Kalkül ist jedoch nur durch ihr Verbrauchen während der Ausführung, jedoch nicht über die Interaktion mit freien Molekül-Kompositionen modellierbar. \square

Beispiel 3.3.12 (Chemische Reaktionen) Angenommen, eine chemische Reaktion $m_1 + m_2 \longrightarrow m_3$ findet innerhalb einer Membran statt. Dies kann folgendermaßen modelliert werden:

$$\begin{aligned} & \mathbb{b}_R \left\langle \begin{array}{c} \diamond(m_1 \circ m_2) \\ \diamond(m_3) \end{array} \right\rangle (m_1 \circ m_2 \circ R) \\ \xrightarrow{\mathbb{b}_R} & (m_3 \circ R) \end{aligned}$$

Eine außerhalb der Membran auftretende mit m_1 reaktionsfähige Molekül-Komposition m_4 kann die Reaktion $m_1 + m_2 \longrightarrow m_3$ jedoch nicht beeinflussen, da sie durch die Membran räumlich getrennt ist. \square

Beispiel 3.3.13 (Degeneration und Synthese) Makrophagen sind dazu in der Lage, Proteinkomplexe aufzunehmen und mithilfe von intrazellulären Enzymen in ihre Bestandteile zu zerlegen. Dieser meist in Lysosomen ablaufende Zersetzungsprozess wird im Folgenden formalisiert:

$$\begin{aligned} & (p \circ !\mathbb{b}_R \left\langle \begin{array}{c} p(\diamond) \\ \diamond(\diamond) \end{array} \right\rangle (\parallel)) \\ \equiv & (p \circ \mathbb{b}_R \left\langle \begin{array}{c} p(\diamond) \\ \diamond(\diamond) \end{array} \right\rangle \mid !\mathbb{b}_R \left\langle \begin{array}{c} p(\diamond) \\ \diamond(\diamond) \end{array} \right\rangle (\parallel)) \\ \xrightarrow{\mathbb{b}_R} & (!\mathbb{b}_R \left\langle \begin{array}{c} p(\diamond) \\ \diamond(\diamond) \end{array} \right\rangle (\parallel)) \end{aligned}$$

In dieser Modellierung wird die Molekül-Komposition p mithilfe eines durch $!\mathbb{b}_R \left\langle \begin{array}{c} p(\diamond) \\ \diamond(\diamond) \end{array} \right\rangle (\parallel)$ modellierten Enzyms in nicht weiter betrachtete Bestandteile zerlegt. Das Enzym liegt nach der Reaktion in unveränderter Form vor.

Auf ähnliche Weise lässt sich auch die Proteinsynthese umsetzen. \square

Beispiel 3.3.14 (Transmembraner Molekültransport) Wir betrachten das System $S = !\mathbb{b}_R \left\langle \begin{smallmatrix} p(\diamond) \\ \diamond(p) \end{smallmatrix} \right\rangle (S')$, welches dazu in der Lage ist, Molekül-Kompositionen der Form p durch seine Membran nach innen zu transportieren. Trifft ein p an der äußeren Membran von S ein, ergibt sich Folgendes:

$$\begin{array}{c} p \circ !\mathbb{b}_R \left\langle \begin{smallmatrix} p(\diamond) \\ \diamond(p) \end{smallmatrix} \right\rangle (S') \\ \xrightarrow{\mathbb{b}_R} !\mathbb{b}_R \left\langle \begin{smallmatrix} p(\diamond) \\ \diamond(p) \end{smallmatrix} \right\rangle (p \circ S') \end{array}$$

Die Molekül-Komposition p wurde also aufgenommen, ohne die Struktur von S nachhaltig zu ändern. Der Transport einer Molekül-Komposition q von innen nach außen kann analog durch $!\mathbb{b}_R \left\langle \begin{smallmatrix} \diamond(q) \\ q(\diamond) \end{smallmatrix} \right\rangle (S')$ und die Möglichkeit beider Transportrichtungen durch $!\mathbb{b}_R \left\langle \begin{smallmatrix} p(\diamond) \\ \diamond(p) \end{smallmatrix} \right\rangle !\mathbb{b}_R \left\langle \begin{smallmatrix} \diamond(q) \\ q(\diamond) \end{smallmatrix} \right\rangle (S')$ modelliert werden. Für die Biologie ist auch das System $!\mathbb{b}_R \left\langle \begin{smallmatrix} p(q) \\ q(p) \end{smallmatrix} \right\rangle (S')$ relevant, welches das Einschleusen von Molekülen nur im Gegenzug zum gleichzeitigen Ausschleusen anderer Moleküle erlaubt.

Die hier genannten Mechanismen dienen sowohl der Modellierung passiver Diffusionen einzelner Moleküle durch Membranen hindurch in Richtung des durch Membranen gebildeten Konzentrationsgefälles, als auch der Modellierung von aktiven Molekülpumpen, welche den Molekültransport unter Aufwendung von Energie auch gegen das Konzentrationsgefälle realisieren. \square

Die obigen Beispiele verdeutlichen, dass B&R eine sehr allgemeine Operation ist, die eine Vielzahl von Modellierungsmöglichkeiten bietet. Während bei den vorgestellten Operationen aus den Definitionen 3.3.4 und 3.3.7 die Wahrung von Strukturprinzipien sehr bedeutsam ist, gibt es hingegen bei B&R bezogen auf die Entstehung und das Verschwinden von Molekülen keinerlei Einschränkung. Die Bewahrung von Masse- und Energie-Gesetzen ist also keineswegs gewährleistet und verlangt eine sorgfältige Verwendung von B&R bei der Modellierung.

3.4 Weitere Arbeiten zum Brane-Kalkül

Ein stark vereinfachtes Konzept zur Darstellung von *Molekülkomplexen* im Brane-Kalkül wird bereits bei seiner Einführung in [13] vorgestellt, dieses beschränkt sich auf einen weiteren Verknüpfungsoperator zur Kennzeichnung der Zusammengehörigkeit verschiedener Moleküle sowie eine Anpassung der Operation B&R zur Darstellung von Komplexierungen und Dekomplexierungen. Auch die Integration von Konzepten des π -Kalküls sowie der Bio-Ambients zur Erweiterung seiner Ausdruckstärke steht zur Diskussion. Wir gehen nun kurz auf einige weitere mit dem Brane-Kalkül im Zusammenhang stehende Arbeiten ein. Zunächst geben wir einen Überblick über Aussagen zu Eigenschaften einiger Varianten und stellen anschließend zwei weitere Prozesskalküle vor, welche die Konzepte des Brane-Kalküls verwenden.

3.4.1 Eigenschaften des Kalküls

Membraninteraktionen laufen in der Biologie auf lokaler Ebene ab, Verschmelzungen und Abspaltungen werden also nicht durch ganze Membranen beeinflusst, sondern basieren auf der Umstrukturierung einzelner Membransegmente. Derartige Prozesse lassen sich mit der in Abbildung 3.3 dargestellten Operation *switch* beschreiben. Luca Cardelli zeigt in [12] unter anderem, dass sämtliche Membraninteraktionen des Brane-Kalküls außer B&R unter lokaler Betrachtung auf *switch* zurückführbar sind und damit eine biologisch plausible Modellierung darstellen. Desweiteren geht er auf das durch *switch* bewahrte biologische Prinzip der *Bitonalität* ein, demgemäß Membranhälften in der hierarchischen Membranstruktur mittels Interaktionen stets nur zwei Ebenen hinauf oder hinab gelangen oder auf gleicher Ebene bleiben können und weist diese Eigenschaft für exakt die Operationen nach, die sich auf *switch* zurückführen lassen. Wir erinnern uns vergleichend an die Modellierungen der Endozytose und Exozytose in den Bio-Ambients, welche die Bitonalität nicht bewahren und sich dementsprechend auch nicht mit *switch* beschreiben lassen.

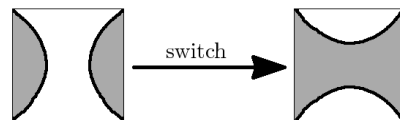


Abbildung 3.3: Die Operation *switch*.

Die Arbeit [7] betrachtet Eigenschaften zur Ausdrucksstärke einzelner Teile des Brane-Kalküls. Es wird gezeigt, dass bereits unter Weglassen der Reduktionsregeln PINO, MATE, BUD, DRIP und B&R die Turing-Mächtigkeit erreicht werden kann. Verzichtet man hingegen auf PHAGO, EXO, PINO und B&R, so ist diese Eigenschaft nicht gegeben. In einer Fortführung dieser Arbeit ([6]) wird die *Semantik der maximalen Parallelität* eingeführt, welche die Interaktion aller interaktionsfähigen Komponenten in jedem Ableitungsschritt erzwingt und dadurch auch in der zweiten Variante die Turing-Mächtigkeit erreicht. Weitere Eigenschaften von Varianten des Brane-Kalküls werden in [5] und [38] diskutiert, darüberhinaus existieren Ansätze zum Model Checking, beispielsweise in [34] und [35].

3.4.2 Verwandte Prozesskalküle

Der *Projektive Brane-Kalkül* ([25]) differenziert zwischen Außen- und Innenseiten von Membranen und ordnet ihnen *gerichtete* Aktionen zu. Diese Sichtweise ermöglicht die konzeptionelle Zusammenfassung einiger Operationen durch Verwendung gemeinsamer Aktionen. Exo und Mate lassen sich darstellen mittels einer Aktion f_n , welche die Fusion zweier Membranen bewirkt. Liegen zwei Membranen direkt nebeneinander und besitzen jeweils die Aktion f_n an ihrer Außenseite, so können sie miteinander fusionieren, der Ablauf ist analog zu Mate im Brane-Kalkül. Darüberhinaus ist aber jeweils auch deren Verschmelzung mit einer direkt übergeordneten Membran möglich, wenn diese die Aktion f_n an ihrer Innenseite trägt. In diesem Fall ist das Resultat vergleichbar mit Exo, der Inhalt der inneren Membran wird im Zuge der Verschmelzung ausgespien. Die nach der Interaktion verbleibenden gerichteten Prozesse der inneren Membran werden

in vertauschter Form der äußeren zugeordnet: der Prozess der ursprünglichen Außenseite der inneren Membran liegt dann auf der inneren Seite der äußeren Membran und umgekehrt. Dies entspricht dem natürlichen Ablauf der Exozytose, die ebenfalls zu einer Umstülpung der inneren Membran und damit zur Richtungsumkehr aller auf ihr befindlichen Bestandteile führt. Auf ähnliche Weise lassen sich sowohl Phago als auch Bud mit gemeinsamen Aktionen w und w^\perp (*wrap*) darstellen. Pino und Drip benötigen lediglich eine gemeinsame Aktion b (*bubble*), die an der Innenseite einer Membran notiert eine Membranblase innen (Pino), an der Außenseite notiert entsprechend eine Membranblase außen (Drip) produzieren kann. Der Projektive Brane-Kalkül befindet sich näher am Prinzip lokaler Operationen gemäß switch, abstrahiert also weniger stark von der biologischen Realität als der Brane-Kalkül.

In [16] wird ein Ansatz diskutiert, der sich primär auf die Modellierung von Molekülinteraktionen mithilfe von Prozessen im Stile des π -Kalküls konzentriert. Dabei wird jedem Molekül eine Menge von mit Vorzeichen versehenen *Bindungsstellen* zugeordnet, welche mithilfe spezieller Aktionen $offer(a^+)$, $offer(a^-)$, $retract(a^+)$ und $retract(a^-)$ um eine Bindungsstelle a^+ bzw. a^- erweitert bzw. gekürzt werden können. Zwei Moleküle gelten als miteinander verbunden, wenn eines von ihnen eine Bindungsstelle a^+ , das zweite entsprechend die zugehörige Bindungsstelle a^- besitzt. Die Menge der Bindungsstellen eines durch derartige Verbindungen geformten Molekülkomplexes ergibt sich schließlich aus der Vereinigung der Mengen von Bindungsstellen der zugehörigen Moleküle abzüglich der Bindungsstellen, die innerhalb dieser Vereinigungsmenge sowohl in der Form a^+ als auch in der Form a^- vorkommen. Miteinander verbundene Molekülkomplexe sind dazu in der Lage, durch die Verwendung ihres Verbindungsnamens als Kanal miteinander zu kommunizieren. Eine Erweiterung der Beschreibungsmöglichkeiten um Moleküle mit Inhalt integriert schließlich Membranen in den Kalkül und ermöglicht die Formulierung deren Interaktionen über der ursprünglich für Moleküle definierten Struktur. Darüberhinaus werden die gerichteten Membranoperationen des Projektiven Brane-Kalküls verwendet, die vom Brane-Kalkül bekannte Replikation und die umfassenden Möglichkeiten von B&R gibt es hingegen nicht.

3.5 Anwendungsbeispiel: HIV

Eine erste Modellierung im Brane-Kalkül hat Luca Cardelli bereits bei seiner Einführung in [13] mit der Infektion einer Zelle durch das Semliki-Forest-Virus gegeben. Seine Ausführungen beinhalten den Prozess vom Andocken eines Viruspartikels an eine Zelle bis hin zu ihrer Infektion und der damit verbundenen fortlaufenden Abkospung neuer Viruspartikel. Ein weiteres Beispiel liefern Nadia Busi und Claudio Zandron ([9]) mit der Aufnahme und zellularen Zersetzung eines cholesterintragenden Partikels in dessen Bestandteile. Zwei nicht unmittelbar mit der Biologie in Zusammenhang stehende Anwendungen lassen sich in [8] und [50] finden.

Wir modellieren in diesem Abschnitt eine Infektion durch HIV. Darüberhinaus demonstrieren wir die Ausdrucksstärke des Brane-Kalküls anhand der Modellierung einer Abwehrreaktion des Immunsystems und einiger Mechanismen des Virus, die zum erfolgreichen Unterwandern der Immunreaktion und seinem damit verbundenen Ausbreiten im infizierten Organismus führen, und liefern somit das bisher umfassendste Anwen-

dungsbeispiel im Brane-Kalkül.

3.5.1 Aufbau und Funktionsweise

Für ein besseres Verständnis der folgenden Modellierungen stellen wir zunächst die Funktionsweise des Immunsystems und die Mechanismen des HIV in knapper und vereinfachter Form vor. Für weiterführende Betrachtungen seien [29], [4], [37] und [41] empfohlen.

Aufbau des Virus

Das *humane Immundefizienzvirus* (HIV) gehört zur Familie der Retroviren und zur Unterfamilie der Lentiviren. Die kugelförmigen infektiösen Partikel haben einen Durchmesser von etwa 100 nm und sind von einer Hüllmembran umgeben, die in ihrer Struktur ähnlich zur Cytoplasmamembran einer Zelle ist. In dieser ist das *transmembrane virale Glycoprotein* gp41 verankert, welches wiederum mit dem *externen* viralen Glycoprotein gp120 assoziiert ist. Innerhalb der Hüllmembran befindet sich ein aus Proteinen aufgebautes *Capsid*, in dessen Inneren sich in Form zweier identischer Kopien einzelsträngiger RNA das Virusgenom befindet. Abbildung 3.4 zeigt unter Abschnitt 1 die schematische Darstellung eines HIV-Partikels.

Replikationszyklus

Der im Folgenden geschilderte Replikationszyklus des HIV ist in Abbildung 3.4 grafisch dargestellt. Wir erläutern die einzelnen Schritte und geben die zugehörigen Abschnitte der Abbildung in Klammern an.

Entscheidend für eine Infektion mit HIV ist das Vorhandensein des Glycoproteins CD4 auf der Membran der zu infizierenden Zelle, welches hauptsächlich bei den für die Immunabwehr zuständigen Makrophagen und Helfer-T-Zellen auftritt. Das Andocken des externen viralen Glycoproteins gp120 an das CD4 (2) induziert eine Konformationsänderung im gp120. Dadurch wird eine fusogene Domäne des transmembranen viralen Glycoproteins gp41 freigelegt, die zur Fusion von Virus- und Zellmembran führt und damit das Capsid ins Zellinnere befördert (3). Ein Triphosphat macht das Capsid durchlässig für die virale RNA, sie tritt in das Cytoplasma der Zelle ein (4). Dort vollführt das Enzym *Reverse Transkriptase* einen RNA-Translationsvorgang, der das Entstehen von doppelsträngiger viraler DNA zur Folge hat (5). Diese wird in den Zellkern transportiert und in das Genom der Zelle dauerhaft eingebaut (6). Aufgrund der veränderten Erbinformation produziert die Zellmaschinerie fortan virale RNA und virale Proteine (7, 8). Diese formieren sich zunächst zu unfertigen Viruspartikeln, welche nach der Abknospung von der Zellmembran (9) schließlich zu infektiösen Viruspartikeln heranreifen (1).

Abwehrreaktion des Immunsystems

Makrophagen sind nach ihrer Infektion durch HIV dazu in der Lage, in ihnen vorhandene virale Proteine zu zersetzen, an eigene MHC-Proteine zu binden und diese schließlich auf ihrer Zellmembran zu exponieren. Dadurch wird das Virus für das Immunsystem

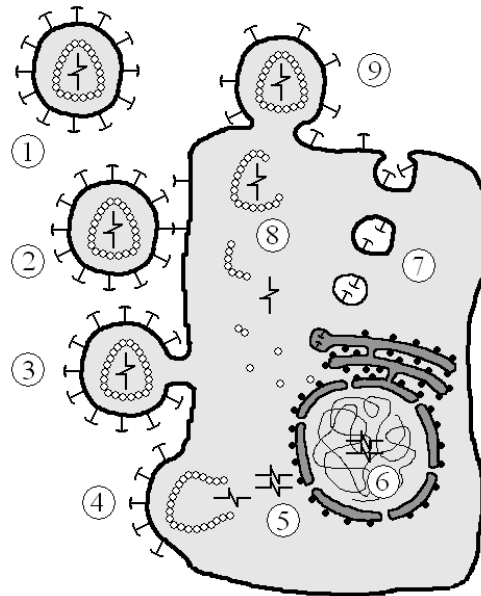


Abbildung 3.4: Schematische Darstellung des HIV-Replikationszyklus.

als Fremdkörper markiert. Ein weiterer Mechanismus zur Gewinnung solcher MHC-Protein-Komplexe ist das Verschlingen der Bruchstücke von Zellen, die im Zuge der Virusinfektion nach dem Abknospen mehrerer Viruspartikel abgestorben sind. Bei der zweiten Variante ist im Gegensatz zur ersten für das Exponieren des Komplexes keine direkte Infektion der Makrophagen nötig.

In der Gruppe der T-Zellen wird zwischen den CD4-positiven Helfer-T-Zellen und den CD8-positiven cytotoxischen T-Zellen unterschieden. Diese liegen zunächst in ruhender Form vor. Mit ihren T-Zell-Rezeptoren mit jeweils spezifischem Ende können sie an spezielle von Makrophagen exponierte MHC-Protein-Komplexe andocken. Einige T-Zellen besitzen den zum mit HIV-Fragmenten angereicherten MHC-Protein-Komplex passenden Rezeptor und werden durch das Andocken an solche Makrophagen aktiviert. Es folgt deren selektive Vermehrung. Aktivierte cytotoxische T-Zellen sind nun dazu in der Lage, die Zerstörung körpereigener virusbefallener Zellen einzuleiten. Helfer-T-Zellen hingegen exponieren nach einem Umwandlungsprozess wiederum Virusfragmente auf ihrem T-Zell-Rezeptor. B-Zellen tragen den Antikörper IgM auf ihrer Zellmembran, der ebenfalls ein spezifisches Ende besitzt. Solche mit zum Virusfragment passendem Ende docken an entsprechende Helfer-T-Zellen an. Dies bewirkt die Vermehrung der B-Zellen und darüberhinaus deren Heranreifen zu *Plasmazellen*, welche fortan an HIV angepasste Antikörper IgG synthetisieren und in freier Form an die Umgebung abgeben. Diese docken an membranständiges gp120 der Viren und befallenen Zellen an und verhindern damit deren Fusion mit CD4-positiven Zellen oder geben ein Signal zur Einleitung deren gezielten Zelltodes (*Apoptose*). Zusätzlich hemmen sie das Abknospen neuer Viruspartikel von befallenen Zellen.

Sowohl B-Zellen als auch T-Zellen können nach dem Auftreten eines Virus die Funktion als Gedächtniszellen durch dauerhaftes Exponieren derartiger Proteinkomplexe einnehmen und damit fortan die schnellere Immunreaktion unterstützen.

Pathomechanismen

Das Immunsystem kann der Ausbreitung von HIV auf Dauer nicht standhalten, im Laufe eines Zeitraumes von mehreren Jahren gewinnt das Virus im befallenen Organismus die Oberhand. Wir gehen kurz auf einige Mechanismen ein, die zum allmählichen Unterwandern der körpereigenen Abwehr führen.

Absterben infizierter Zellen: Die fortlaufende Abknospung von Viruspartikeln schwächt infizierte Zellen nachhaltig und führt nach einiger Zeit zu deren Absterben. Weil HIV die Zellen des Immunsystems befällt, wird dieses fortlaufend gestört.

Synzytien-Bildung: Infizierte Zellen exponieren gp120 an ihrer Oberfläche und können mit gesunden Zellen zu mehrkernigen Riesenzellen (*Synzytien*) fusionieren und damit die Funktion der gesunden Zellen stören. Eine infizierte Zelle kann bis zu 200 nicht infizierte Zellen binden, ein einziges infektiöses HIV-Partikel trägt damit maßgeblich zur Abnahme der Anzahl nicht infizierter Zellen bei.

Freies gp120: Infektiöse Viruspartikel sind dazu in der Lage, freies Glycoprotein gp120 abzusondern. Dieses lagert sich an den CD4-Rezeptoren gesunder Zellen an und macht sie zum Ziel von freien Antikörpern, nach deren Andocken der gezielte Zelltod eingeleitet wird.

Mutation: Aufgrund der hohen Fehlerrate des RNA-Translationsvorganges weichen die produzierten viralen Proteine verschiedener infizierter Zellen voneinander ab, wodurch das Virus ein starkes Mutationsverhalten aufweist. Dies äußert sich insbesondere in einer sich fortlaufend ändernden Struktur des viralen gp120 und resultiert in verändertem Andockverhalten. Das Immunsystem muss immer neue Antikörper und spezifische T-Zellen mit den dazu passenden Rezeptoren bereitstellen, um die neuen Varianten des Virus zu eliminieren.

Die geschilderten Mechanismen führen zu einem stetigen Abbau der für die Immunabwehr des Körpers zuständigen Zellen und zwingen die noch vorhandenen zur ständigen Anpassung an die sich neu entwickelnden Virusmutationen. Dadurch wird das Immunsystem dauerhaft geschwächt und unterliegt schließlich den Auswirkungen anderer Infektionen.

3.5.2 Modellierung

Zur Erhöhung der Übersichtlichkeit weichen wir von den bei der Einführung genannten Namenskonventionen im Brane-Kalkül zugunsten ausführlicherer Bezeichnungen ab. Desweiteren notieren wir einzelne Teilbereiche eines Systems abkürzend mit geeigneten Namen als Platzhaltern, wobei Namen von Teilsystemen mit großen Buchstaben und Namen von Teilprozessen mit kleinen Buchstaben beginnen. Zur optischen Abgrenzung von Platzhalten notieren wir Moleküle in kursiven Großbuchstaben. Jede Zelle und jedes Viruspartikel wird als eigenständiges System umgesetzt. Dabei konzentrieren wir uns stets nur auf die an den von uns betrachteten Abläufen beteiligten Organellen, verzichten also auf die Modellierung aller anderen.

Modellierung eines HIV-Partikels

Ein infektiöses HIV-Partikel wird durch das folgende System **Virus** angegeben:

$$\begin{aligned}
\mathbf{Virus} &:= m_v(\mathbf{Cap}_v) \\
m_v &:= !\mathbf{mt} \\
\mathbf{Cap}_v &:= m_c(\mathbf{RNA}_v) \\
m_c &:= \mathbf{br} \left\langle \begin{array}{l} TP(\mathbf{RNA}_v) \\ TP \circ \mathbf{RNA}_v(\diamond) \end{array} \right\rangle
\end{aligned}$$

Die viralen Glycoproteine gp120 und gp41 werden lediglich indirekt durch den Prozess \mathbf{mt} auf der Virusmembran (m_v) modelliert, welcher die Fähigkeit des Virus zur Fusion mit einer CD4-positiven Zelle charakterisiert. Die Aktion \mathbf{mt} benötigt jeweils eine in der entsprechenden interagierenden Zelle verankerte Ko-Aktion, die wir später definieren werden. Das aus Proteinen aufgebaute Capsid wird als Teilsystem \mathbf{Cap}_v mit eigener Membran modelliert, die bei Einwirken eines Triphosphats (TP) für die umschlossene virale RNA (\mathbf{RNA}_v) durchlässig wird.

Replikationszyklus

Mit dem folgenden System **TZelle** geben wir zunächst die Definition einer Helfer-T-Zelle und demonstrieren dann mithilfe einer Reduktionsableitung den Befall dieser Zelle durch ein infektiöses HIV-Partikel und das darauf folgende Abknospen neuer Viruspartikel.

$$\begin{aligned}
\mathbf{TZelle} &:= m_t(TP \circ \mathbf{Nuc}_t) \\
m_t &:= !\mathbf{mt}^\perp | !\uparrow^\perp \\
\mathbf{Nuc}_t &:= \mathbf{br} \left\langle \begin{array}{l} \mathbf{RNA}_v(\mathbf{DNA}) \\ \diamond(\mathbf{DNA}_v) \end{array} \right\rangle . \mathbf{protSyn}_v(\mathbf{DNA}) \\
\mathbf{protSyn}_v &:= !(\mathbf{dr}(\mathbf{memProt}_v) | \mathbf{dr}(\mathbf{capProt}_v)) \\
\mathbf{memProt}_v &:= \uparrow . \mathbf{bd}^\perp(m_v) \\
\mathbf{capProt}_v &:= \mathbf{bd} . \mathbf{br} \left\langle \begin{array}{l} \diamond(\diamond) \\ \diamond(\mathbf{RNA}_v) \end{array} \right\rangle . m_c
\end{aligned}$$

Das CD4-Protein ist indirekt durch den Prozess \mathbf{mt}^\perp auf der Membran der Helfer-T-Zelle (m_t) angegeben. $!\uparrow^\perp$ ist Teil der Zellmaschinerie und spezifiziert den möglichen Transport von Proteinen aus dem Inneren der Zelle an deren Membran. TP modelliert das für das Herauslösen der viralen DNA aus dem Capsid benötigte Triphosphat als freies Molekül. Der Zellkern und das ihn umschließende rauhe endoplasmatische Retikulum werden als Einheit durch \mathbf{Nuc}_t dargestellt. Die Aktion $\mathbf{br} \left\langle \begin{array}{l} \mathbf{RNA}_v(\mathbf{DNA}) \\ \diamond(\mathbf{DNA}_v) \end{array} \right\rangle$ gewährleistet die Einbettung viralen Genoms in die DNA der Zelle beim Auftreten von \mathbf{RNA}_v , wobei hier auf eine Modellierung der dafür notwendigen Translationsvorgänge mittels Reverser Transkriptase verzichtet wird. Nach erfolgter Einbettung ist \mathbf{Nuc}_t zur viralen Proteinsynthese fähig ($\mathbf{protSyn}_v$) mithilfe der Abschnürung einzelner Vesikel mit viralen Proteinen ($\mathbf{memProt}_v$, $\mathbf{capProt}_v$). Diese dienen der Bildung neuer Viruspartikel und deren Abknospung von der infizierten Zelle.

Die folgende Reduktionsableitung beschreibt den Ablauf der Infektion einer Helfer-T-Zelle durch HIV. Dabei werden die jeweils an einer Umformung beteiligten Systeme durch Unterstreichen gekennzeichnet.

$$\begin{aligned}
& \underline{\text{Virus}} \circ \underline{\text{TZelle}} \\
\equiv & \underline{\text{mt}|\text{m}_v(\text{Cap}_v)} \circ \underline{\text{mt}^\perp|\text{m}_t(\text{TP} \circ \text{Nuc}_t)} \\
\stackrel{\text{mt}}{\longrightarrow} & \text{m}_v|\text{m}_t(\underline{\text{Cap}_v} \circ \text{TP} \circ \text{Nuc}_t) \\
= & \text{m}_v|\text{m}_t(\underline{\text{br} \left\langle \begin{array}{c} \text{TP}(\text{RNA}_v) \\ \text{TP} \circ \text{RNA}_v(\diamond) \end{array} \right\rangle} (\text{RNA}_v) \circ \underline{\text{TP}} \circ \text{Nuc}_t) \\
\stackrel{\text{br}}{\longrightarrow} & \text{m}_v|\text{m}_t(\text{RNA}_v \circ \underline{\emptyset} \circ \text{TP} \circ \underline{\text{Nuc}_t}) \\
\equiv & \text{m}_v|\text{m}_t(\underline{\text{RNA}_v} \circ \text{TP} \circ \underline{\text{br} \left\langle \begin{array}{c} \text{RNA}_v(\text{DNA}) \\ \diamond(\text{DNA}_v) \end{array} \right\rangle} \cdot \text{protSyn}_v(\text{DNA})) \\
\stackrel{\text{br}}{\longrightarrow} & \text{m}_v|\text{m}_t(\text{TP} \circ \underline{\text{protSyn}_v(\text{DNA}_v)}) \\
\equiv & \text{m}_v|\text{m}_t(\text{TP} \circ \underline{\text{dr}(\text{memProt}_v)}|\text{dr}(\text{capProt}_v)|\text{protSyn}_v(\text{DNA}_v)) \\
\stackrel{\text{dr}}{\longrightarrow} & \text{m}_v|\underline{\text{m}_t}(\text{TP} \circ \underline{\text{memProt}_v}(\emptyset) \circ \text{dr}(\text{capProt}_v)|\text{protSyn}_v(\text{DNA}_v)) \\
\equiv & \underline{\text{m}_v|\uparrow^\perp|\text{m}_t(\text{TP} \circ \uparrow \cdot \text{bd}^\perp(\text{m}_v)(\emptyset) \circ \text{dr}(\text{capProt}_v)|\text{protSyn}_v(\text{DNA}_v))} \\
\stackrel{\uparrow}{\longrightarrow} & \text{m}_v|\text{m}_t|\text{bd}^\perp(\text{m}_v)(\text{TP} \circ \underline{\text{dr}(\text{capProt}_v)}|\text{protSyn}_v(\text{DNA}_v)) \\
\stackrel{\text{dr}}{\longrightarrow} & \text{m}_v|\text{m}_t|\text{bd}^\perp(\text{m}_v)(\text{TP} \circ \underline{\text{capProt}_v}(\emptyset) \circ \text{protSyn}_v(\text{DNA}_v)) \\
= & \text{m}_v|\text{m}_t|\text{bd}^\perp(\text{m}_v)(\text{TP} \circ \underline{\text{bd} \cdot \text{br} \left\langle \begin{array}{c} \diamond(\diamond) \\ \diamond(\text{RNA}_v) \end{array} \right\rangle} \cdot \text{m}_c(\emptyset) \circ \text{protSyn}_v(\text{DNA}_v)) \\
\stackrel{\text{bd}}{\longrightarrow} & \underline{\text{m}_v(\text{br} \left\langle \begin{array}{c} \diamond(\diamond) \\ \diamond(\text{RNA}_v) \end{array} \right\rangle} \cdot \text{m}_c(\emptyset)) \circ \text{m}_v|\text{m}_t(\text{TP} \circ \text{protSyn}_v(\text{DNA}_v)) \\
\stackrel{\text{br}}{\longrightarrow} & \underline{\text{m}_v(\text{m}_c(\text{RNA}_v))} \circ \text{m}_v|\text{m}_t(\text{TP} \circ \text{protSyn}_v(\text{DNA}_v)) \\
= & \text{Virus} \circ \text{m}_v|\text{m}_t(\text{TP} \circ \text{protSyn}_v(\text{DNA}_v))
\end{aligned}$$

Im ersten Ableitungsschritt verschmelzen Zell- und Virusmembran und Cap_v gelangt in das Cytoplasma der Helfer-T-Zelle. Durch das dort vorhandene TP wird RNA_v im zweiten Ableitungsschritt durch die Capsidmembran hindurch in das Cytoplasma befördert und bewirkt im dritten Ableitungsschritt die Änderung des Zellgenoms. Es

folgt die Produktion eines Vesikels, welches schließlich durch Exozytose die am Vesikel angelagerten viralen Proteine an die Zellmembran weitergibt. Danach wird ein Vesikel mit Capsid-Proteinen gebildet, welches sich unter Nutzung der vorher zur Zellmembran transportierten viralen Proteine von der Zellmembran abspaltet. Erst nach dieser Abknospung erfolgt im letzten Schritt die endgültige Reifung zum infektiösen Viruspartikel. Die infizierte Helfer-T-Zelle ist dazu in der Lage, kontinuierlich neue Viruspartikel zu produzieren.

Cytotoxische T-Zellen sind gegen eine Infektion durch HIV aufgrund des fehlenden CD4 auf ihrer Membran immun. Eine Modellierung dieses Aspekts kann problemlos erreicht werden, indem der das Viruseindringen verursachende Prozess $!m\mathfrak{t}^\perp$ nicht auf deren Membran angebracht wird.

Synzytienbildung

Die Modellierung berücksichtigt bereits die Bildung mehrkerniger Riesenzellen. Es wird eine infizierte Helfer-T-Zelle benötigt, die gemäß obiger Reduktionsableitung die Form $m_v|m_t(TP \circ \text{protSyn}_v(DNA_v))$ hat. Zusammen mit einer gesunden Helfer-T-Zelle ergibt sich Folgendes:

$$\begin{aligned}
& \underline{m_v|m_t(TP \circ \text{protSyn}_v(DNA_v))} \circ \underline{\text{TZelle}} \\
& \equiv \underline{m\mathfrak{t}|m_v|m_t(TP \circ \text{protSyn}_v(DNA_v))} \circ \underline{m\mathfrak{t}^\perp|m_t(TP \circ \text{Nuc}_t)} \\
& \xrightarrow{m\mathfrak{t}} m_v|\underline{m_t|m_t(TP \circ \text{protSyn}_v(DNA_v))} \circ TP \circ \text{Nuc}_t \\
& \equiv m_v|m_t(TP \circ TP \circ \text{Nuc}_t \circ \text{protSyn}_v(DNA_v))
\end{aligned}$$

Dieser Vorgang kann nun wiederholt ablaufen, denn die entstandene zweikernige Zelle besitzt wiederum den zur Verschmelzung mit weiteren gesunden Zellen benötigten Rezeptor $m\mathfrak{t}$. Mit dieser Modellierung können also beliebig viele nicht infizierte Helfer-T-Zellen mit einer infizierten Helfer-T-Zelle zu einem Synzytium verschmelzen.

Phagozytose viraler Bruchstücke durch Makrophagen

Eine infizierte Helfer-T-Zelle ist nur eine begrenzte Zeit lang dazu in der Lage, neue Viruspartikel zu produzieren. Langfristig unterliegt sie der Virusinfektion und stirbt ab. Es entstehen Fragmente der Zelle, die virale Bestandteile enthalten. Auch dieser Sachverhalt ist im Brane-Kalkül darstellbar. Wir verzichten jedoch auf die Modellierung und nehmen an, es seien bereits solche Fragmente vorhanden. Die darauf folgende Phagozytose dieser Reste durch eine Makrophage und deren Exposition mithilfe von MHC-Proteinen werden wir im Folgenden darstellen. Zunächst geben wir dafür die

Umsetzung einer Makrophage im Brane-Kalkül an.

$$\begin{aligned}
\text{Makrophage} &:= m_m(\text{Lysosom}_m) \\
m_m &:= !\downarrow_m^\perp(m\mathfrak{t}_l)|!\uparrow^\perp \\
\text{Lysosom}_m &:= m_l(\emptyset) \\
m_l &:= !m\mathfrak{t}_l^\perp.\text{dr}(\uparrow.\text{mhcPK})
\end{aligned}$$

Auf der Makrophagenmembran (m_m) befindet sich ein Rezeptor zur Phagozytose kleinerer Strukturen ($!\downarrow_m^\perp(m\mathfrak{t}_l)$) und, wie schon bei einer Helfer-T-Zelle, ein Teil der Zellmaschinerie ($!\uparrow^\perp$) zum Transport von Proteinen. Im Cytoplasma der Makrophage befindet sich ein Lysosom (Lysosom_m), welches der Aufnahme des phagozytierten Partikels dient. Nach dessen an dieser Stelle nicht modellierten weiteren Zersetzung kann ein MHC-Protein-Komplex, dargestellt durch den zunächst nicht näher spezifizierten Prozess mhcPK , auf der Makrophagenmembran exponiert werden.

Beispielhaft betrachten wir das Bruchstück ($m\text{emProt}_v(\emptyset)$) einer infizierten Helfer-T-Zelle und reichern es um einen Rezeptor \downarrow_m an, der es für die Makrophage als verschlingbar kennzeichnet. Die folgende Reduktionsableitung verdeutlicht den ablaufenden Vorgang.

$$\begin{aligned}
&\downarrow_m(m\text{emProt}_v(\emptyset)) \circ \underline{\text{Makrophage}} \\
\equiv &\quad \underline{\downarrow_m(m\text{emProt}_v(\emptyset))} \circ \underline{\downarrow_m^\perp(m\mathfrak{t}_l)|m_m(\text{Lysosom}_m)} \\
\stackrel{\downarrow_m}{\longrightarrow} &\quad m_m(m\mathfrak{t}_l((m\text{emProt}_v(\emptyset))) \circ \underline{\text{Lysosom}_m}) \\
\equiv &\quad m_m(\underline{m\mathfrak{t}_l((m\text{emProt}_v(\emptyset)))} \circ \underline{m\mathfrak{t}_l^\perp.\text{dr}(\uparrow.\text{mhcPK})|m_l(\emptyset)}) \\
\stackrel{m\mathfrak{t}_l}{\longrightarrow} &\quad m_m(\underline{\text{dr}(\uparrow.\text{mhcPK})|m_l((m\text{emProt}_v(\emptyset)))}) \\
\stackrel{\text{dr}}{\longrightarrow} &\quad \underline{m_m(\uparrow.\text{mhcPK}(\emptyset) \circ m_l((m\text{emProt}_v(\emptyset)))})} \\
\equiv &\quad \underline{\uparrow^\perp|m_m(\uparrow.\text{mhcPK}(\emptyset) \circ m_l((m\text{emProt}_v(\emptyset)))})} \\
\stackrel{\uparrow}{\longrightarrow} &\quad \text{mhcPK}|m_m(m_l((m\text{emProt}_v(\emptyset))))
\end{aligned}$$

Der erste Ableitungsschritt modelliert die Phagozytose des Virusproteins. Dabei entsteht ein das Viruspartikel umhüllendes *Endosom*, welches im zweiten Ableitungsschritt mit dem Lysosom verschmilzt. Das Lysosom sondert nun ein Vesikel ab, welches den MHC-Protein-Komplex auf seiner Membran trägt. Dieses gelangt schließlich im letzten Schritt an die Zelloberfläche.

Auch eine Makrophage besitzt das Glycoprotein CD4 auf seiner Membran und ist somit anfällig für eine Virusinfektion. Wie bei einer Helfer-T-Zelle könnte dies durch den Prozess $!m\mathfrak{t}^\perp$ modelliert werden, bei zusätzlichem Anreichern des Cytoplasmas

um TP und Nuc_t läuft der Infektionsprozess auf gleiche Weise ab. Das auch in diesem Fall stattfindende Exponieren eines MHC-Protein-Komplexes ließe sich in Nuc_t durch Erweiterung von protSyn_v auf $\text{dr}(\uparrow \cdot \text{mhcPK})|\text{protSyn}_v$ verankern. Wir verzichten an dieser Stelle auf eine Modellierung der genannten Verhaltensweisen bei einer Makrophage.

Aktivierung einer Helfer-T-Zelle durch eine Makrophage

Nach Exponieren des Komplexes aus HIV-Fragmenten und MHC-Proteinen auf der Makrophagenmembran binden einige Helfer-T-Zellen mit passendem Rezeptor temporär an diesen, um dadurch aktiviert zu werden und nach einigen Umwandlungsprozessen ihrerseits Virusfragmente an ihrem T-Zell-Rezeptor zu exponieren. Die Modellierung einer solchen temporären Verbindung zweier Zellen durch membranständige Proteine, die keine Veränderung der Membranstrukturen zur Folge hat, ist im Brane-Kalkül zunächst nicht vorgesehen. Allerdings kann dieser Effekt mit den vorhandenen Operationen nachgebildet werden. Hierzu wählen wir eine Phagozytose der Makrophage durch die Helfer-T-Zelle und ihre unmittelbar darauf folgende Exozytose und erweitern die Struktur der beiden Zellen derart, dass sie nach diesem Prozess in unveränderter Form vorliegen. Als Konsequenz von Phagozytose und Exozytose wird schließlich ein Prozess auf der Membran der Helfer-T-Zelle freigelegt, der das Exponieren von Virusfragmenten auf dem T-Zell-Rezeptor charakterisiert.

Wir spezifizieren nun den Rezeptor mhcPK aus der Definition einer Makrophage. Zusätzlich modifizieren wir die Helfer-T-Zelle, konzentrieren uns aber lediglich auf die veränderten Komponenten, alle anderen behalten die in der ursprünglichen Definition angegebene Struktur.

$$\begin{aligned} \text{mhcPK} &:= !\downarrow_c \\ \text{TZelle}' &:= m'_t(\text{TP} \circ \text{Nuc}_t) \\ m'_t &:= !\downarrow_c^\perp (\uparrow_c) \cdot \uparrow_c^\perp \cdot !\downarrow_d |m_t \end{aligned}$$

Die Helfer-T-Zelle behält ihre bisherigen in m_t angegebenen Membranfunktionen. Zusätzlich exponiert sie nun einen T-Zell-Rezeptor durch $\downarrow_c^\perp (\uparrow_c)$. Dieser ist für das Andocken einer Makrophage bestimmt und phagozytiert sie. Die darauf folgende Aktion \uparrow_c^\perp gewährleistet ihr sofortiges Wiederaustrreten. Erst nach Konsumieren dieser zwei Aktionen wird $!\downarrow_d$ freigelegt und stattet die Helfer-T-Zelle mit einem das Virusfragment präsentierenden T-Zell-Rezeptor aus, der das Andocken einer B-Zelle ermöglicht. Gemäß der im vorigen Abschnitt gezeigten Reduktionsableitung hat eine Virusfragment-exponierende Makrophage die Form $\text{mhcPK}|m_m(m_l(\text{memProt}_v(\emptyset)))$. Der beschriebene Vorgang des temporären Andockens lässt sich dann durch folgende Reduktionsableitung beschreiben.

$$\begin{aligned} &\underline{\text{mhcPK}|m_m(m_l(\text{memProt}_v(\emptyset)))} \circ \underline{\text{TZelle}'} \\ \equiv &\underline{\downarrow_c | \text{mhcPK}|m_m(m_l(\text{memProt}_v(\emptyset)))} \circ \underline{\downarrow_c^\perp (\uparrow_c) \cdot \uparrow_c^\perp \cdot !\downarrow_d | m'_t(\text{TP} \circ \text{Nuc}_t)} \end{aligned}$$

$$\xrightarrow{\downarrow_c} \underline{\uparrow_c^\perp \cdot ! \downarrow_d | m'_t(\uparrow_c(\text{mhcPK}|_{m_m}(\text{m}_l(\text{memProt}_v(\emptyset)))) \circ TP \circ \text{Nuc}_t)}$$

$$\xrightarrow{\uparrow_c} \text{mhcPK}|_{m_m}(\text{m}_l(\text{memProt}_v(\emptyset))) \circ ! \downarrow_d | m'_t(TP \circ \text{Nuc}_t)$$

Wir stellen fest, dass die Makrophage nach dem Prozess in unveränderter Form vorliegt und die resultierende Helfer-T-Zelle sich von der ursprünglichen lediglich durch den zusätzlichen Rezeptor $! \downarrow_d$ unterscheidet.

Dem Aspekt, dass nicht jede Helfer-T-Zelle den zum MHC-Protein-Komplex der Makrophage passenden Rezeptor besitzt, kann durch die gleichzeitige Verwendung von Systemen TZelle und TZelle' in der Modellierung Rechnung getragen werden. Strebt man eine umfassendere Modellierung mit mehreren Virenarten an, lassen sich weitere auf Makrophagen befindliche virenspezifische MHC-Protein-Komplexe durch zu $! \downarrow_c$ parallel verknüpfte Prozesse $! \downarrow_x$ modellieren, wobei x zur eindeutigen Benennung des speziellen Rezeptors dient. Entsprechend müssen dann auch die jeweiligen Helfer-T-Zellen mit dazu passenden Prozessen ausgestattet werden.

Nach der Aktivierung einer Helfer-T-Zelle erfolgt auch deren Vermehrung. Im Brane-Kalkül kann diese Selbstreplikation zwar mithilfe des Replikationsoperators erreicht werden, denn für beliebige Systeme $!S$ gilt $!S \equiv !!S \equiv !S \circ !S \equiv !S \circ !S$. Derartige Systeme $!S$ sind allerdings nicht zu Interaktionen fähig und deshalb für die Modellierung schlecht geeignet. Eine Möglichkeit zur Darstellung der Selbstreplikation in unserem Beispiel ist die Angabe eines zusätzlichen Systems, welches durch die Aktivierung einer Helfer-T-Zelle ebenfalls aktiviert wird und daraufhin Systeme TZelle' generiert. Weil diese Modellierung wenig anschaulich ist, verzichten wir auf ihre Umsetzung.

Bildung von Plasmazellen

Einige B-Zellen besitzen HIV-spezifische Antikörper IgM auf ihrer Membran, die an die von einigen Helfer-T-Zellen durch T-Zell-Rezeptoren exponierten HIV-Fragmente temporär binden können und dadurch aktiviert werden. Diesen Aspekt modellieren wir ähnlich zu obigem Beispiel durch Phagozytose einer aktivierten Helfer-T-Zelle seitens der B-Zelle und ihre unmittelbar darauf folgende Exozytose. Der dafür bei der Helfer-T-Zelle notwendige Rezeptor $! \downarrow_d$ wurde bereits in der Definition des Systems TZelle' verankert und durch temporäres Andocken einer Makrophage freigelegt.

Bei der B-Zelle kommt es nach der Aktivierung zu ihrer Vermehrung, erst daraufhin erfolgt die Entwicklung zu Plasmazellen. Wir bedienen uns an dieser Stelle eines Hilfsmechanismus: in unserer Modellierung ist die B-Zelle nach der Aktivierung dazu in der Lage, Plasmazellen in unbegrenzter Anzahl zu produzieren. Wir betrachten nun die Definition einer entsprechenden B-Zelle. Die Modellierung einer Plasmazelle erfolgt mit dem System PZelle , welches an späterer Stelle genauer charakterisiert wird.

$$\begin{aligned} \text{BZelle} &:= m_b(\text{PSyn}_b) \\ m_b &:= ! \downarrow_d^\perp (\uparrow_d) \cdot \uparrow_d^\perp \cdot ! \uparrow_p^\perp \\ \text{PSyn}_b &:= !(\uparrow_p(\text{PZelle})) \end{aligned}$$

Im Folgenden betrachten wir eine Reduktionsableitung der von einer aktivierten B-Zelle ausgehenden Plasmazellenbildung. Auslösender Schritt ist das temporäre Andocken einer gemäß obiger Ableitung durch $! \downarrow_d | m'_t(TP \circ \text{Nuc}_t)$ modellierten aktivierten

Helfer-T-Zelle. Dessen Ablauf ist ähnlich zum bereits dargestellten Andockprozess einer Makrophage an eine Helfer-T-Zelle, wir konzentrieren uns deshalb an dieser Stelle lediglich auf das resultierende System.

$$! \downarrow_d |m'_t(TP \circ Nuc_t) \circ BZelle \longrightarrow^* ! \downarrow_d |m'_t(TP \circ Nuc_t) \circ ! \uparrow_p^\perp |m_b(PSyn_b)$$

Die B-Zelle ist nun dazu in der Lage, fortlaufend P-Zellen zu produzieren.

$$\begin{aligned} & \underline{! \uparrow_p^\perp |m_b(PSyn_b)} \\ \equiv & \underline{\uparrow_p^\perp |! \uparrow_p^\perp |m_b(\uparrow_p(PZelle)) \circ PSyn_b)} \\ \xrightarrow{\uparrow_p} & PZelle \circ ! \uparrow_p^\perp |m_b(PSyn_b) \end{aligned}$$

Freie Proteine

Wir gehen abschließend auf einige durch freies gp120 und freie Antikörper IgG verursachte Effekte ein. Eine mögliche Umsetzung dieser Proteine im Brane-Kalkül wären Moleküle, welche mithilfe der Operation B&R mit Systemen interagieren könnten. Weil membranassoziierte Proteine bisher jedoch indirekt mit durch sie verursachten Membraninteraktionen modelliert wurden, würde dies zu zwei verschiedenen Repräsentationsformen von Proteinen innerhalb der gleichen Modellierung führen. Um eine solche Vermischung der Konzepte zu vermeiden, werden wir deshalb auch freie Proteine mithilfe von membranassoziierten Prozessen modellieren. Dadurch benötigen wir allerdings zusätzliche Hilfsmembranen ohne biologische Entsprechung, mit denen wir die Prozesse verknüpfen können.

Eine Plasmazelle ist aus einer aktivierten B-Zelle entstanden und dazu in der Lage, spezifische gegen das Hüllprotein gp120 wirkende Antikörper IgG abzugeben. Das zugehörige System geben wir im Folgenden an. Weil wir keine anderen Aspekte einer Plasmazelle modellieren, bleibt ihr Inhalt leer.

$$PZelle := !dr(\downarrow_a^\perp(0))\langle \rangle$$

Durch Anwendung der Operation Drip entsteht das System $\downarrow_a^\perp(0)\langle \rangle$, welches das freie IgG repräsentiert. Dieses muss dazu in der Lage sein, sich an membranständiges gp120 von Viren und befallenen Zellen zu heften und diese unschädlich zu machen. Dieser Aspekt muss noch modelliert werden und führt zu einer modifizierten Definition des Systems *Virus*. Im Zuge dieser Erweiterung stattdessen wir das Viruspartikel zusätzlich mit der Fähigkeit aus, freies gp120 abzusondern.

$$Virus' := \downarrow_a |!dr(m\mathfrak{t} | \downarrow_a)|m_v(Cap_v)$$

Ähnlich zur P-Zelle liefert die Anwendung der Operation Drip ein System $m\mathfrak{t} | \downarrow_a \langle \rangle$, welches das freie gp120 repräsentiert. Durch den Rezeptor \downarrow_a auf der Virusmembran wird es anfällig für den Antikörper IgG. Dieser macht das Viruspartikel durch Verschlingen komplett handlungsunfähig und modelliert damit dessen Eliminierung oder das Besetzen seiner zum Andocken benötigten Rezeptoren.

$$\underline{\downarrow_a |!dr(m\mathfrak{t} | \downarrow_a)|m_v(Cap_v)} \circ \underline{\downarrow_a^\perp(0)\langle \rangle} \xrightarrow{\downarrow_a} \langle \langle !dr(m\mathfrak{t} | \downarrow_a)|m_v(Cap_v) \rangle \rangle$$

Die durch Mate modellierte Membranfusion während der Infektion einer Zelle berücksichtigt die Weitergabe der viralen Rezeptoren, entsprechend wird auch der neu eingeführte Rezeptor \Downarrow_a auf die infizierte Zelle übertragen und macht sie ebenfalls für den Antikörper angreifbar. Zusätzlich kann auch freies gp120 durch IgG eliminiert werden.

$$\underline{\text{mt} \mid \Downarrow_a \langle \rangle} \circ \underline{\Downarrow_a^\perp (0) \langle \rangle} \xrightarrow{\Downarrow_a} \langle \langle \text{mt} \langle \rangle \rangle \rangle$$

Freies gp120 ist außerdem dazu in der Lage, an das CD4 von gesunden Helfer-T-Zellen anzudocken. Diese können daraufhin vom Antikörper befallen und ebenfalls handlungsunfähig gemacht werden. Die folgende Reduktionsableitung veranschaulicht diesen Effekt.

$$\begin{aligned} & \Downarrow_a^\perp (0) \langle \rangle \circ \underline{\text{mt} \mid \Downarrow_a \langle \rangle} \circ \underline{\text{TZelle}} \\ \xrightarrow{\text{mt}} & \underline{\Downarrow_a^\perp (0) \langle \rangle} \circ \underline{\Downarrow_a \mid \text{mt} \langle TP \circ \text{Nuc}_t \rangle} \\ \xrightarrow{\Downarrow_a} & \langle \langle \text{mt} \langle TP \circ \text{Nuc}_t \rangle \rangle \rangle \end{aligned}$$

Entscheidend für den Befall durch freies gp120 ist, ebenso wie für den Befall durch ein HIV-Partikel, die Präsenz der Aktion mt^\perp auf der Zellmembran. Diese Aktion entspricht dem Protein CD4 in der Natur, welches auch dort von freien gp120-Partikeln befallen werden kann. Die Modellierung setzt den Pathomechanismus also auf intuitive Weise um.

3.5.3 Diskussion

Die Modellierung der HIV-Infektion und der mit ihr verbundenen Seiteneffekte zeigt, dass der Brane-Kalkül als mächtiges Beschreibungs- und Simulationswerkzeug für das Zusammenspiel verschiedenster membranbasierter biologischer Einheiten verstanden werden kann. Obwohl er von den für Membranwechselwirkungen unbedingt notwendigen Proteininteraktionen abstrahiert, bietet er differenzierte Modellierungsmöglichkeiten ihrer Auswirkungen durch die Zuordnung spezieller membranassoziierter Prozesse. Seine übersichtliche Syntax vereint die direkte Darstellung hierarchisch aufgebauter Membranstrukturen mit der formalen Spezifikation ihrer Eigenschaften und qualifiziert ihn somit als Beschreibungssprache membranorientierter Systeme.

Allerdings verlangen einige Aspekte der Modellierung zusätzliche Hilfskonzepte, die in der Natur in dieser Form nicht auftreten. So haben die Membranen bei den freien Proteinen gp120 und IgG in der Natur keine Entsprechung, sie dienen lediglich als Träger der Prozesse, die für die Interaktionsmöglichkeiten der freien Proteine eingesetzt werden. Desweiteren sind die Modellierungen der Zellaktivierung durch temporäres Andocken erst mithilfe der zwei Hilfsaktionen Phagozytose und anschließender Exozytose realisierbar. Der modellierte Ablauf findet in der Natur auf diese Art nicht statt und führt somit zwischenzeitlich zu Systemzuständen, welche die Natur nicht korrekt abbilden. Auch eine Modellierung mithilfe der Operationen Mate und Bud hätte zu nicht plausiblen Systemzuständen geführt, zudem würde in diesem Fall eine zusätzliche Membran in einer der beiden Zellen benötigt. Eine Umsetzung durch Absondern und Aufnehmen von freien Molekülen über B&R führt zu ähnlichen Schwierigkeiten, zusätzlich wäre

hier der temporäre Kontakt von B-Zelle und Helfer-T-Zelle nicht darstellbar. Künstliche Strukturen sind für die Modellierung einiger Mechanismen also unabdingbar. Eine weitere Schwierigkeit ergibt sich bei der Modellierung eines blockierten Rezeptors durch sich anlagernde freie Moleküle, sodass dieser fortan nicht mehr zur Interaktion fähig ist. In einer direkten Umsetzung des Brane-Kalküls würde dies dem Deaktivieren einer Aktion entsprechen. Dies ist allerdings stets nur durch deren Verbrauchen erreichbar, bei mit dem Replikationsoperator versehenen Aktionen funktioniert auch das nicht. Erst der Hilfsmechanismus der Phagozytose schafft ein Umhüllen der entsprechenden Aktionen und damit deren symptomatische Deaktivierung, wie am Beispiel des Antikörpers IgG demonstriert wurde. Mit dieser Herangehensweise ergibt sich jedoch das Problem, dass stets das ganze die Aktion tragende System deaktiviert wird und keine Auswahl einer einzigen zu deaktivierenden Aktion möglich ist. Im Hinblick auf die Modellierung von *Inhibitoren* ([31]), welche die Infektion gesunder Zellen oder die Produktion neuer Viren hemmen und somit von wachsender Bedeutung für die Entwicklung von Medikamenten zur Behandlung der mit HIV in Zusammenhang stehenden AIDS-Erkrankung sind, kann die Deaktivierung einzelner Aktionen jedoch erforderlich sein.

Der Brane-Kalkül verlangt die explizite Modellierung jeder durch Proteininteraktionen verursachten Umstrukturierung von Membranen. Jede neu erlangte Erkenntnis über das Zusammenspiel mehrerer Proteine muss also mithilfe einer Brane-Kalkül-Aktion mit neuem Index direkt umgesetzt werden. Das schnell wachsende Wissen über die komplexen Umlagerungs- und Interaktionsmöglichkeiten von Proteinen legt die Überlegung nahe, nach allgemeineren Notationen für Membraninteraktionen zu suchen, welche zum Beispiel auf Basis von Proteininteraktionsdatenbanken selbstständig die neu erlangten Erkenntnisse umzusetzen imstande sind. Neben der Einsatzmöglichkeit des Brane-Kalküls als Beschreibungssprache könnte ein solcher Ansatz die Erforschung unbekannter Membran-Wechselwirkungen durch Simulation beschleunigen. Im Kapitel 5 werden wir eine Erweiterung des Brane-Kalküls entwickeln, welche unter Nutzung seiner Vorteile für die Modellierung von Membraninteraktionen seine Fähigkeiten der Darstellung von Proteininteraktionen ausbaut und durch eine intuitivere und wirklichkeitsnähere Abbildung biologischer Prozesse deren Simulationsmöglichkeiten erweitert.

Kapitel 4

Der κ -Kalkül

Sämtliche Zellvorgänge unterliegen dem massiven Einfluß von Proteinen. Diese fungieren als Rezeptoren, Katalysatoren, haben steuernde Funktionen, dienen dem Stofftransport und der Herstellung neuer Proteine. Wesentliches Organisationsprinzip ist die Bildung von Komplexen mit anderen Proteinen und der Zerfall derartiger Komplexe in einzelne Bestandteile. Die damit verbundenen Änderungen der räumlichen Struktur beeinflussen maßgeblich die Eigenschaften eines Proteins und damit seine Interaktionsmöglichkeiten mit anderen Proteinen.

Im Jahr 2004 haben Vincent Danos und Cosimo Laneve den κ -Kalkül vorgestellt ([23]). Als reine Umsetzung der Protein-Maschine abstrahiert er vollständig von dreidimensionalen Strukturen und modelliert Proteininteraktionen auf der Basis von Manipulationen abstrakter Bindungsstellen unter Angabe spezieller *Reaktionsregeln*. Er kann als direkte Fortführung von [24] und [22] verstanden werden, weitere Arbeiten sind beispielsweise in [17] und [51] zu finden. Eine Erweiterung um die in eukaryotischen Zellen unabdingbare Abgrenzung von Reaktionsräumen durch Membranen erfolgte durch Cosimo Laneve und Fabien Tarissan im Jahr 2007 ([30]). Aufbauend auf dem κ -Kalkül entstand der *bio* κ -Kalkül, welcher zusätzlich zu Kompartimenten auch die Interaktionsmöglichkeiten von membranständigen Proteinen und einige Konzepte der Membran-Maschine in den Kalkül integriert. In diesem Kapitel stellen wir den κ -Kalkül vor, demonstrieren seine Ausdrucksfähigkeit anhand einiger mit der HIV-Infektion in Zusammenhang stehender Proteininteraktionen und diskutieren seine Erweiterung hinsichtlich ihrer Anwendbarkeit für Modellierungen der Membran-Maschine.

4.1 Syntax

Auf der Abstraktionsstufe der Protein-Maschine werden bei einem Protein zwei Arten von Interaktionsstellen unterschieden: Bindungs- und Phosphorylierungsstellen. Während eine Bindungsstelle aktiv, inaktiv oder bereits mit der Bindungsstelle eines anderen Proteins verbunden sein kann und damit das Verbindungsglied bei der Komplexbildung darstellt, wirken Phosphorylierungsstellen als Schalter für das Freilegen oder Verstecken von Bindungsstellen. Der κ -Kalkül verzichtet auf diese Differenzierung und charakterisiert ein Protein allein durch seine Bindungsstellen. Wie sich später herausstellen wird, können die Wirkungen von Phosphorylierungen und Dephosphorylierungen dennoch modelliert werden.

Definition 4.1.1 (Bindungsstellen) Wir bezeichnen mit \mathcal{N} die abzählbar unendliche Menge der *Proteinennamen*.

Sei $A \in \mathcal{N}$. Die *Bindungsstellen von A* werden angegeben durch die Menge $S_A = \{1, \dots, \#(A)\}$, wobei die Funktion $\# : \mathcal{N} \rightarrow \mathbb{N}$ jedem A eine endliche Anzahl von Bindungsstellen $\#(A)$ eindeutig zuordnet. \square

Bei der Modellierung eines Proteins gehen wir also von einer konstanten Anzahl von Bindungsstellen aus. Jede Bindungsstelle bekommt zu ihrer Benennung innerhalb des Proteins eindeutig eine natürliche Zahl zugeordnet, wobei wir für n Bindungsstellen die Zahlen 1 bis n verwenden. Jede dieser Bindungsstellen befindet sich in einem der drei im Folgenden genannten Zustände, wobei wir in Klammern bereits eine später verwendete Kurzform für die Bezeichnung des jeweiligen Zustandes angeben.

- *aktiv* (v): Sie kann mit einer Bindungsstelle eines anderen Proteins interagieren.
- *inaktiv* (h): Sie ist unzugänglich für Interaktionen.
- *gebunden* (x): Sie ist bereits zu einer anderen Bindungsstelle über die Verbindung mit dem Namen x gebunden.

Mit der folgenden Definition formalisieren wir die Zuordnung von Zuständen zu den Bindungsstellen eines Proteinennamens.

Definition 4.1.2 (Interface) Wir bezeichnen mit \mathcal{E} die abzählbar unendliche Menge der *Verbindungsnamen*.

Sei $A \in \mathcal{N}$. Ein *Interface von A* ist eine partielle Funktion $\varphi_A : S_A \dashrightarrow \mathcal{E} \cup \{v, h\}$ mit $\{v, h\} \cap \mathcal{E} = \emptyset$. Dabei gilt für jedes $\varphi_A(i), \varphi_A(j) \in \mathcal{E}$ mit $i \neq j$, dass $\varphi_A(i) \neq \varphi_A(j)$. Die Menge aller Interfaces bezeichnen wir mit \mathcal{I} .

Sei $\text{def}(\varphi_A) = \{i \mid \varphi_A(i) \in \mathcal{E} \cup \{v, h\}\}$. Gilt $\text{def}(\varphi_A) = S_A$, sprechen wir auch von einem *totalen Interface*. Wir bezeichnen die Menge aller totalen Interfaces mit \mathcal{I}_t . \square

Die Symbole v bzw. h zur Kennzeichnung von aktiven bzw. inaktiven Bindungsstellen werden aus Gründen der Eindeutigkeit nicht als Verbindungsnamen verwendet. Zudem wird sich später als nützlich erweisen, dass ein Interface φ_A nicht notwendigerweise allen Bindungsstellen von A einen der Zustände v , h oder x zuordnen muss. Zusätzlich zur gegebenen Definition werden wir eine spezielle Form des Interfaces benötigen, die wir im Folgenden einführen.

Definition 4.1.3 (v - h -Interface) Ein Interface φ_A heißt *v - h -Interface*, wenn $\varphi_A(i) \in \{v, h\}$ für alle $i \in \text{def}(\varphi_A)$. Sei φ_A ein v - h -Interface. Das zu φ_A komplementäre v - h -Interface ist für alle $i \in \text{def}(\varphi_A)$ definiert als

$$\overline{\varphi_A}(i) = \begin{cases} h & \text{wenn } \varphi_A(i) = v \\ v & \text{sonst} \end{cases}$$

mit $\text{def}(\overline{\varphi_A}) = \text{def}(\varphi_A)$. Die Menge aller v - h -Interfaces bezeichnen wir mit \mathcal{I}_{vh} . \square

Vereinfachend notieren wir jedes Interface als Summation von um Zustandsinformationen angereicherte Bindungsstellen seines Definitionsbereiches. Dabei schreiben wir

eine aktive Bindungsstelle n als n , eine inaktive als \bar{n} und eine mit x gebundene als n^x . Sei beispielsweise $\varphi_A = [1 \mapsto v, 2 \mapsto h, 3 \mapsto x]$ ein Interface, dann schreiben wir $\varphi_A = 1 + \bar{2} + 3^x$. Die Reihenfolge ist nicht von Bedeutung, es gilt also unter anderem auch $\varphi_A = \bar{2} + 1 + 3^x$. Wir wenden uns nun der formalen Definition der Syntax des κ -Kalküls zu.

Definition 4.1.4 (κ -Kalkül-Syntax) Die Menge der *Konfigurationen* wird mit \mathcal{K} notiert und ist gegeben durch die folgende Grammatik:

$$K ::= \diamond \mid A(\varphi_A) \mid K \circ K.$$

Dabei gilt $A \in \mathcal{N}$ und $\varphi_A \in \mathcal{I}_t$. Wir nennen $A(\varphi_A)$ ein *Protein* und bezeichnen die Menge aller Proteine mit \mathcal{P} . \square

Eine Konfiguration bezeichnet eine Sequenz von Proteinen, dabei notieren wir mit \diamond die leere Konfiguration. Das in einem Protein $A(\varphi_A)$ angegebene Interface φ_A bezieht sich stets auf den ebenfalls im Protein angegebenen Proteinnamen A , wir verzichten deshalb im weiteren Verlauf auf den Index im Interface und schreiben vereinfachend φ . Man beachte, dass φ ein totales Interface ist. Ein Protein ist somit eindeutig charakterisiert durch einen Namen und die vollständige Beschreibung der Zustände seiner Bindungsstellen. $A_1(\varphi_1) \circ A_2(\varphi_2)$ bezeichnet die *Protein-Komposition* zweier Proteine $A_1(\varphi_1)$ und $A_2(\varphi_2)$. Diese ist jedoch nicht als Komplex im biologischen Sinne zu verstehen, sondern modelliert lediglich die Existenz beider Proteine in einem betrachteten biologischen Zusammenhang. Die Komplexbildung hingegen wird durch das Belegen von Bindungsstellen mit gemeinsamen Verbindungsnamen notiert. Zum besseren Verständnis der Struktur von Konfigurationen folgt nun ein Beispiel.

Beispiel 4.1.5 (Proteinkomplex) Wir betrachten die Konfiguration

$$K = A(1^x) \circ B(\bar{1} + 2) \circ C(1^y + 2^x + 3) \circ D(1^y + \bar{2}).$$

Hierbei ist A mit C über x verbunden, ebenso C mit D über y . Also bilden A , C und D einen Proteinkomplex. B hat keine Verbindung und ist somit nicht Teil eines Komplexes. \square

4.2 Semantik

Wir geben zunächst zwei Funktionen an, die wir im Folgenden wiederholt benötigen.

Definition 4.2.1 (*occ*) Die Funktion $occ : \mathcal{E} \times \mathcal{K} \rightarrow \mathbb{N}$ ist für alle $x \in \mathcal{E}$, $A(\varphi) \in \mathcal{P}$ und $K_1, K_2 \in \mathcal{K}$ induktiv definiert als

$$\begin{aligned} occ(x, \diamond) &= 0 \\ occ(x, A(\varphi)) &= \begin{cases} 1 & \text{wenn } \exists i \in \text{def}(\varphi). \varphi(i) = x \\ 0 & \text{sonst} \end{cases} \\ occ(x, K_1 \circ K_2) &= occ(x, K_1) + occ(x, K_2) \end{aligned} \quad \square$$

occ bestimmt die Anzahl der Vorkommen (occurrences) eines Verbindungsnamen x in einer Konfiguration K . Weil aufgrund der Definition eines Interfaces ein Verbindungsname in einem Protein höchstens einmal vorkommen kann, leistet die Definition das Gewünschte.

Definition 4.2.2 (*en, diff*) Die Funktion $en : \mathcal{K} \rightarrow 2^{\mathcal{E}}$ ist für alle $K \in \mathcal{K}$ definiert als

$$en(K) = \{x \mid occ(x, K) \geq 1\}.$$

Zusätzlich führen wir für alle $K, K', L \in \mathcal{K}$ abkürzend die Schreibweise $diff(K, K', L)$ ein mit

$$diff(K, K', L) \Leftrightarrow ((en(K') \setminus en(K)) \cup (en(K) \setminus en(K'))) \cap en(L) = \emptyset. \quad \square$$

Die Funktion en (edge names) baut auf occ auf und dient der Bestimmung aller Verbindungsnamen, die in einer Konfiguration K vorkommen. Mit $diff$ beschreiben wir die Aussage, dass Verbindungsnamen, die entweder nur in K oder nur in K' vorkommen, in L nicht vorhanden sind.

Innerhalb von Konfigurationen, die einen biologischen Sachverhalt repräsentieren, sollten Verbindungsnamen stets exakt zweimal oder gar nicht auftreten, denn die zugrunde liegende Intention ist die Beschreibung zweier Bindungsstellen verschiedener Proteine, welche sich zusammenlagern und dadurch einen Proteinkomplex formen. Eine an einen Verbindungsnamen gebundene Bindungsstelle, für die keine zweite Bindungsstelle mit demselben Verbindungsnamen existiert, kann also nicht auftreten. Das mehr als zweimalige Auftreten von Verbindungsnamen würde hingegen eine Bindung von mehr als zwei Proteinen an einer einzigen Bindungsstelle modellieren und kann aufgrund der geringen Auftrittswahrscheinlichkeit in der Natur vernachlässigt werden. Die folgende Definition formalisiert das Genannte.

Definition 4.2.3 (Wohlgeformtheit von Konfigurationen) Sei $K \in \mathcal{K}$. K heißt *geschlossen* oder *wohlgeformt*, wenn für alle $x \in \mathcal{E}$ $occ(x, K) = 0$ oder $occ(x, K) = 2$ gilt. \square

Gemäß obiger Definition werden in einer wohlgeformten Konfiguration verschiedene Verbindungen stets auch durch verschiedene Namen angegeben. Über diese eindeutige Zuordnung hinaus besteht jedoch keine Notwendigkeit zur Unterscheidung von Konfigurationen, die lediglich durch die verwendeten Verbindungsnamen voneinander abweichen. Wir führen die Relation $=_{\alpha}$ ein, die der Einbettung einer solchen inhaltlichen Gleichsetzung in den κ -Kalkül dient. Diese ist unter dem Namen α -Konversion auch aus anderen Prozesskalkülen bekannt.

Definition 4.2.4 (Substitution) Eine *Substitution* ist eine Funktion $\theta : \mathcal{E} \rightarrow \mathcal{E}$, welche nur in endlich vielen Stellen von der Identitätsfunktion abweicht. \square

Definition 4.2.5 ($=_{\alpha}$) Sei θ eine Substitution. Wir definieren die *Anwendung von θ auf Konfigurationen* für alle $A(\varphi) \in \mathcal{P}$ und $K_1, K_2 \in \mathcal{K}$ induktiv als

$$\begin{aligned} \diamond \theta &= \diamond \\ A(\varphi) \theta &= A(\varphi') && \text{mit } def(\varphi') = def(\varphi) \text{ und} \\ &&& \forall x \in def(\varphi). \varphi'(x) = \theta(\varphi(x)) \\ (K_1 \circ K_2) \theta &= (K_1 \theta) \circ (K_2 \theta) \end{aligned}$$

Jeder Substitution θ ordnen wir für jedes $K \in \mathcal{K}$ eine Substitution $\theta_K : \mathcal{E} \rightarrow \mathcal{E}$ zu, welche für alle $x \in \mathcal{E}$ definiert ist als

$$\theta_K(x) = \begin{cases} \theta(x) & \text{wenn } x \in \text{en}(K), \theta(x) \notin \text{en}(K) \text{ und } \forall y \in \text{en}(K) \setminus \{x\}. \theta(x) \neq \theta(y) \\ x & \text{sonst} \end{cases}$$

Sei schließlich $=_\alpha \subseteq \mathcal{K} \times \mathcal{K}$ die kleinste Äquivalenzrelation, die alle Paare (K, K') enthält, für die eine Substitution θ existiert mit $K \theta_K = K'$. \square

Jede Substitution θ_K beschränkt sich auf Namen von K und nimmt nur solche Umbenennungen gemäß θ vor, die nicht zur Zusammenlegung zweier verschiedener Namen von K zu einem Namen führen. Eine solche Zusammenlegung könnte direkt verursacht werden durch $\theta(x) = \theta(y)$ oder indirekt durch $\theta(x) = y$, wenn ein ebenfalls definiertes $\theta(y) = x$ aufgrund eines anderen Konfliktes nicht zur Anwendung kommt. Diese beiden Fälle werden in θ_K ausgeschlossen. Die Äquivalenzrelation $=_\alpha$ bewahrt also die Verbindungsnamenstruktur einzelner Konfigurationen und somit insbesondere deren Wohlgeformtheit bei der Umbenennung. Wir werden nun die Äquivalenzrelation und den Begriff der Wohlgeformtheit anhand eines Beispiels näher betrachten.

Beispiel 4.2.6 ($=_\alpha$, Wohlgeformtheit) Die Konfigurationen

$$K_1 = A(1^x) \circ B(1^y + 2^x) \circ C(1^y) \text{ und } K_2 = A(1^y) \circ B(1^z + 2^y) \circ C(1^z)$$

sind wohlgeformt. Unter Betrachtung der Substitutionen θ_1 und θ_2 mit $\theta_1(x) = a$, $\theta_1(y) = z$ und $\theta_2(a) = y$, die bezogen auf andere Werte der Identitätsabbildung entsprechen, ergibt sich $K_1 \theta_{1K_1} = K'$ mit $K' = A(1^a) \circ B(1^z + 2^a) \circ C(1^z)$ und $K' \theta_{2K'} = K_2$. Also gilt $K_1 =_\alpha K'$ und $K' =_\alpha K_2$, aufgrund der Transitivität von $=_\alpha$ folgt $K_1 =_\alpha K_2$. Sei

$$K_3 = A(1^x) \circ B(1^y + 2^x) \circ C(1^x),$$

dann gilt $K_1 \neq_\alpha K_3 \neq_\alpha K_2$. Weiterhin ist K_3 nicht wohlgeformt, denn $\text{occ}(x, K_3) = 3$ und $\text{occ}(y, K_3) = 1$. \square

Anders als beim Brane-Kalkül genügt es beim κ -Kalkül für die Modellierung des Verhaltens verschiedener Elemente eines Gesamtsystems nicht, eine Reduktionsrelation anzugeben, denn die Syntax liefert hier keine Kodierung der Interaktionsmöglichkeiten eines Systems, sondern lediglich den Zustand von Proteinen und Proteinkomplexen. Wir benötigen zusätzliche Informationen über die Art der Interaktionen, zu denen einzelne Proteine fähig sind. Diese werden in Form von *Reaktionsregeln* bereitgestellt, bei denen zwei Arten unterschieden werden. *Komplexierungsregeln* spezifizieren die Fähigkeit zweier Proteine, sich unter Ausbildung einer Verbindung zwischen zwei Bindungsstellen aneinanderzulagern, *Dekomplexierungsregeln* geben umgekehrt die Möglichkeit der Trennung derartiger Kontakte durch Zerstörung der entsprechenden Verbindung an. Im Folgenden betrachten wir die Bezeichnungen Reaktion und Interaktion im Zusammenhang mit Proteinen als äquivalent.

In der Biologie ist für das Zustandekommen einer Proteininteraktion die dreidimensionale Proteinstruktur der beteiligten Proteine und deren Beeinflussung während der Interaktion bedeutsam, die zur Aktivierung und Deaktivierung einiger Bindungsstellen führen kann. Informationen darüber müssen bei der Modellierung in der Reaktionsregel

verankert werden. Desweiteren wird jede Reaktionsregel später als Komplexierungsregel oder als Dekomplexierungsregel charakterisiert. Weil diese sich in ihrer syntaktischen Struktur jedoch nicht unterscheiden, verzichten wir in der formalen Definition zunächst auf die genannte Differenzierung.

Definition 4.2.7 (Reaktionsregel, LR_r) Eine *Reaktionsregel* r ist ein Tupel

$$r = ((A, i, \varphi_A, \varphi'_A), (B, j, \psi_B, \psi'_B)),$$

wobei $A, B \in \mathcal{N}$, $i \in S_A, j \in S_B, \varphi_A, \psi_B \in \mathcal{I}_{vh}$ und $\varphi'_A, \psi'_B \in \mathcal{I}$. Dabei gilt $\text{def}(\varphi_A) \cap \text{def}(\varphi'_A) = \emptyset = \text{def}(\psi_B) \cap \text{def}(\psi'_B)$. Desweiteren sind $\varphi_A(i), \varphi'_A(i), \psi_B(j)$ und $\psi'_B(j)$ undefiniert.

Wir definieren weiterhin für jede Reaktionsregel $r = ((A, i, \varphi_A, \varphi'_A), (B, j, \psi_B, \psi'_B))$ die Menge ihrer jeweils mit Zuordnungsindex versehenen rechten und linken Seite mit

$$LR_r := \{(A, i, \varphi_A, \varphi'_A)_1, (B, j, \psi_B, \psi'_B)_2\}. \quad \square$$

Eine Reaktionsregel bestimmt zwei Proteinnamen A und B als Reaktionspartner und legt die jeweiligen Bindungsstellen i und j von A und B fest, an denen die Komplexierung bzw. Dekomplexierung stattfinden soll. Durch φ_A und ψ_B werden Bindungsstellen angegeben, die vor der Reaktion auf ihren aktuellen Zustand geprüft werden und im Zuge der Reaktion ihren Zustand von aktiv zu inaktiv oder umgekehrt ändern. Entsprechend werden auch die durch φ'_A und ψ'_B angegebenen Bindungsstellen auf ihren Zustand geprüft, jedoch erfahren diese während der Reaktion keine Zustandsänderung. Alle nicht in φ_A oder φ'_A bzw. ψ_B oder ψ'_B vorkommenden Bindungsstellen sind für die Reaktion nicht von Bedeutung und bleiben unverändert. Die Proteinnamen A und B müssen nicht notwendig verschieden sein, eine Reaktionsregel berücksichtigt also auch die mögliche Interaktion von Proteinen mit anderen gleichartigen und damit gleich benannten Proteinen. Die Menge LR_r einer Reaktionsregel r ermöglicht den Zugriff auf einzelne Komponenten von r und wird an späterer Stelle benötigt.

Auch bei den Interfaces innerhalb einer Reaktionsregel verzichten wir fortan auf die Notation des zugehörigen Proteinnamens im Index, weil dieser eindeutig aus dem Kontext hervorgeht. Das folgende Beispiel dient der Verdeutlichung des Reaktionsvorganges, eine formale Definition folgt mit Einführung der Transitionsrelation des κ -Kalküls.

Beispiel 4.2.8 (Anwendung einer Reaktionsregel) Gegeben sei die Konfiguration

$$K = A(\bar{1} + 2^x + 3) \circ B(1^y + 2 + 3) \circ C(1^x + 2^y).$$

Bei Anwendung der Dekomplexierungsregel $r_d = ((A, 2, \bar{1} + 3, \emptyset), (C, 1, \emptyset, \emptyset))$ auf K ergibt sich die Konfiguration

$$K' = A(1 + 2 + \bar{3}) \circ B(1^y + 2 + 3) \circ C(1 + 2^y).$$

Betrachten wir nun die Komplexierungsregel $r_{c1} = ((A, 1, \bar{3}, 2), (B, 2, \emptyset, 1^x))$. Diese kann nicht auf K angewendet werden, denn die zur Komplexierung benötigte Bindungsstelle 1 von A ist inaktiv und deshalb nicht in der Lage, eine Verbindung einzugehen. Darüber hinaus ist die Bindungsstelle 3 von A aktiv, stimmt also nicht mit der in r_{c1} durch das v - h -Interface $\bar{3}$ geforderten Belegung überein. Auch die geforderte Nichtbelegung der

Bindungsstelle 2 von A ist nicht gewährleistet.

K' hingegen erfüllt alle genannten Bedingungen, r_{c1} kann auf K' angewendet werden. Es resultiert die Konfiguration

$$K'' = A(1^z + 2 + 3) \circ B(1^y + 2^z + 3) \circ C(1 + 2^y).$$

Wir stellen fest, dass die Bindungsstelle 2 in A während der letzten Reaktion zwar auf ihren Zustand geprüft wird, aber unverändert bleibt. Die Bindungsstelle 1 in B wird auf ihre Belegung mit einem Verbindungsnamen überprüft. Hierbei ist nicht entscheidend, dass der in der Reaktionsregel verwendete Name von dem in B verwendeten Namen abweicht.

Betrachten wir abschließend die Komplexierungsregel $r_{c2} = ((A, 3, \emptyset, 1^x), (B, 3, \emptyset, 2^x))$. Diese kann auf K'' angewendet werden. Auch hier ist nicht entscheidend, dass bei der Verbindungsprüfung in der Reaktionsregel der Verbindungsname x , in der Konfiguration hingegen der Name z steht. Es wird jedoch überprüft, ob beide Proteine denselben Namen verwenden, weil auch in der Reaktionsregel derselbe Name angegeben ist. \square

Wie schon im Brane-Kalkül benötigen wir auch im κ -Kalkül eine strukturelle Kongruenz, um Konfigurationen später das gleiche Verhalten zuordnen zu können, wenn sie sich lediglich durch die Anordnung ihrer Proteine unterscheiden.

Definition 4.2.9 (Strukturelle Kongruenz von Konfigurationen) Die *Strukturelle Kongruenz* $\equiv \subseteq \mathcal{K} \times \mathcal{K}$ ist die kleinste Äquivalenzrelation, welche (1) bis (4) erfüllt.

- (1) $K \circ L \equiv L \circ K$
- (2) $K \circ (L \circ M) \equiv (K \circ L) \circ M$
- (3) $K \circ \diamond \equiv K$
- (4) $K \equiv K' \Rightarrow K \circ L \equiv K' \circ L$ \square

Die Transitionsrelation des κ -Kalküls setzt die Spezifikation zweier Mengen von Komplexierungs- (\mathfrak{C}) und Dekomplexierungsregeln (\mathfrak{D}) voraus, welche die Interaktionsmöglichkeiten der Proteine charakterisieren. Unter Voraussetzung zweier konkreter Mengen \mathfrak{C} und \mathfrak{D} können wir nun die Transitionsrelation des κ -Kalküls definieren.

Definition 4.2.10 (Transitionsrelation $\longrightarrow_{\mathcal{R}}$) Sei $\mathcal{R} = (\mathfrak{C}, \mathfrak{D})$ mit zwei Mengen \mathfrak{C} und \mathfrak{D} von Reaktionsregeln. Wir bezeichnen mit \mathcal{L} die Menge der *Label* $\mathcal{L} = \{\tau\} \cup \{r_n^x \mid r \in \mathfrak{C} \cup \mathfrak{D}, x \in \mathcal{E}, n \in \{1, 2\}\}$ und setzen $\mu \in \mathcal{L}$ beliebig. $\longrightarrow_{\mathcal{R}} \subseteq \mathcal{K} \times \mathcal{L} \times \mathcal{K}$ ist die kleinste Relation, die folgende Transitionsregeln erfüllt, wobei wir für $(K, \mu, K') \in \longrightarrow_{\mathcal{R}}$ vereinfachend $K \xrightarrow{\mu} K'$ schreiben:

$$\text{COM} \frac{r \in \mathfrak{C} \quad (A, i, \varphi, \varphi')_n \in LR_r \quad x \in \mathcal{E} \setminus \text{en}(A(i + \varphi + \varphi' + \varphi_0))}{A(i + \varphi + \varphi' + \varphi_0) \xrightarrow{r_n^x} A(i^x + \bar{\varphi} + \varphi' + \varphi_0)}$$

$$\text{DEC} \frac{r \in \mathfrak{D} \quad (A, i, \varphi, \varphi')_n \in LR_r}{A(i^x + \varphi + \varphi' + \varphi_0) \xrightarrow{r_n^x} A(i + \bar{\varphi} + \varphi' + \varphi_0)}$$

$$\begin{array}{l} \text{REACT} \frac{K \xrightarrow{r_1^x} K' \quad L \xrightarrow{r_2^x} L'}{K \circ L \xrightarrow{\tau} K' \circ L'} \quad \text{CONV} \frac{K' =_\alpha K \quad K \xrightarrow{\tau} L \quad L =_\alpha L'}{K' \xrightarrow{\tau} L'} \\ \text{PAR} \frac{K \xrightarrow{\mu} K' \quad \text{diff}(K, K', L)}{K \circ L \xrightarrow{\mu} K' \circ L} \quad \text{STRUCT} \frac{K' \equiv K \quad K \xrightarrow{\mu} L \quad L \equiv L'}{K' \xrightarrow{\mu} L'} \end{array}$$

Wir nennen $\longrightarrow_{\mathcal{R}}$ die *Transitionsrelation* des κ -Kalküls und ihre Elemente *Ableitungsschritte*. Eine endliche Folge $K_1 \xrightarrow{\mu_1} K_2 \xrightarrow{\mu_2} \dots \xrightarrow{\mu_{l-1}} K_l$ bezeichnen wir als *Transitionsableitung*. \square

COM: Für jede Komplexierungsregel r stellt COM sicher, dass beide an r beteiligten Proteine ihre jeweils an der Reaktion beteiligte aktive Bindungsstelle i an einen beliebigen Verbindungsnamen x knüpfen können. Dabei wird überprüft, dass x nicht schon an einer anderen Bindungsstelle des Proteins verwendet wird, was für die Wahrung der Wohlgeformtheit notwendig ist. Dass beide Proteine den gleichen Verbindungsnamen verwenden, wird in der Transitionsregel REACT gewährleistet.

Weiterhin wird sichergestellt, dass die Komplexierung nur bei Vorhandensein der durch das Interface φ geforderten Bindungsstellenbelegungen des Proteins stattfinden kann, im Zuge der Reaktion erfahren die entsprechenden Bindungsstellen Zustandsänderungen von aktiv zu inaktiv oder umgekehrt. Die durch das Interface φ' angegebenen Bindungsstellenbelegungen müssen ebenfalls gewährleistet sein, allerdings bleiben diese während der Reaktion unverändert. Das Interface φ_0 ergibt sich wegen $\longrightarrow_{\mathcal{R}} \subseteq \mathcal{K} \times \mathcal{L} \times \mathcal{K}$ als beliebige Ergänzung von $i + \varphi + \varphi'$ zu einem totalen Interface $i + \varphi + \varphi' + \varphi_0$ des Proteinnamens A gemäß Definition 4.1.4. Die in φ_0 angegebenen Bindungsstellen bleiben während der Reaktion ebenfalls unverändert.

Das Label r_n^x markiert die zur Reaktion führende Reaktionsregel r , den Namen x der entstehenden Verbindung und die verwendete Reaktionsregelseite n . Diese Informationen werden in der Transitionsregel REACT benötigt. Einen Ableitungsschritt mit einem Label dieser Art nennen wir *Teilreaktion*. Wir stellen fest, dass eine Teilreaktion die Wohlgeformtheit seines Proteins nicht bewahrt.

DEC: Die Funktionsweise ist ähnlich zur Regel COM. Im Unterschied zu dieser betrachtet DEC jedoch eine Dekomplexierungsregel r . Die Überprüfung des Vorhandenseins von x an anderer Stelle im Protein entfällt, da keine Verbindung erzeugt, sondern eine gelöscht wird. Auch Ableitungsschritte dieser Art werden als Teilreaktionen bezeichnet.

REACT: Hier werden die an einer Reaktion r beteiligten Proteine, die ihre Komplexierung bzw. Dekomplexierung über COM bzw. DEC durchführen können, zusammengeführt. Die Label r_1^x und r_2^x stellen sicher, dass eine Reaktion nur stattfinden kann, wenn die entsprechenden Teilreaktionen aus derselben Reaktionsregel r hervorgegangen sind. Außerdem wird gewährleistet, dass die Verbindungsentstehung

oder -löschung bei beiden Teilen auf den gleichen Namen x bezogen ist. Die Angabe von 1 und 2 zur Kennzeichnung von linker und rechter Regelseite verhindert eine unbeabsichtigte, durch zwei linke oder zwei rechte Seiten einer Reaktionsregel verursachte Reaktion.

Bei Anwendung von REACT werden die ursprünglichen Label r_1^x und r_2^x der Teilreaktionen in der Gesamtreaktion zu τ modifiziert. Dieser sogenannte *stille Übergang* kennzeichnet die Abgeschlossenheit der Reaktion und macht die zugehörigen Elemente für andere Reaktionen unbeobachtbar.

CONV: An dieser Stelle wird die inhaltliche Gleichstellung zweier sich nur durch die Namen ihrer Verbindungen unterscheidender Konfigurationen in den κ -Kalkül eingebettet. Wesentlich hierfür ist allerdings die Beschränkung auf das Label τ , innerhalb von Teilreaktionen kann also keine Umbenennung von Namen stattfinden. Diese Beschränkung stellt sicher, dass bei der Zusammenführung von Komplezierungen in REACT die Teilreaktionen trotz gleich gekennzeichneter Labels keine durch vorherige Umbenennung erlangten verschiedenen Verbindungsamen verwenden. Weiterhin gewährleistet die Beschränkung die korrekte Funktionsweise des mittels geeigneter Reaktionsregeln formulierbaren Tests, ob beide Reaktionspartner über eine gemeinsame, also mit gleichem Namen bezeichnete, Verbindung verknüpft sind.

STRUCT: Die Strukturelle Kongruenz wird in die Transitionsrelation eingebettet und erzielt dadurch die Unabhängigkeit des Verhaltens einer Konfiguration von der Anordnung ihrer Proteine. Das Label μ kann zu einem beliebigen r_n^x oder τ instanziiert werden und berücksichtigt somit sowohl Teilreaktionen als auch stille Übergänge.

PAR: Diese Transitionsregel stellt sicher, dass eine Proteininteraktion auch innerhalb eines Kontextes stattfinden kann. Dabei muss zur Wahrung der Wohlgeformtheit darauf geachtet werden, dass der beim Übergang $K \xrightarrow{\mu} K'$ entstehende oder verschwindende Verbindungsname nicht auch im angefügten Kontext L verwendet wird. Auch hier wird das Label μ als Platzhalter für konkrete Label verwendet.

Beispiel 4.2.11 (Herleitung eines Ableitungsschrittes) Wir betrachten $\mathcal{R} = (\emptyset, \{r_d\})$ mit der Dekomplexierungsregel $r_d = ((A, 2, \bar{1} + 3, \emptyset), (C, 1, \emptyset, \emptyset))$ aus Beispiel 4.2.8. Es gilt $LR_{r_d} = \{(A, 2, \bar{1} + 3, \emptyset)_1, (C, 1, \emptyset, \emptyset)_2\}$. Folgendes lässt sich ableiten:

$$\begin{aligned}
& r_d \in \mathfrak{D}, (A, 2, \bar{1} + 3, \emptyset)_1 \in LR_{r_d}, (C, 1, \emptyset, \emptyset)_2 \in LR_{r_d} \\
\Rightarrow & A(\bar{1} + 2^x + 3) \xrightarrow{r_{d1}^x} A(1 + 2 + \bar{3}), C(1^x + 2^y) \xrightarrow{r_{d2}^x} C(1 + 2^y) \quad (\text{DEC, DEC}) \\
\Rightarrow & A(\bar{1} + 2^x + 3) \circ C(1^x + 2^y) \xrightarrow{\tau} A(1 + 2 + \bar{3}) \circ C(1 + 2^y) \quad (\text{REACT}) \\
\Rightarrow & A(\bar{1} + 2^x + 3) \circ B(1^y + 2 + 3) \circ C(1^x + 2^y) \\
& \xrightarrow{\tau} A(1 + 2 + \bar{3}) \circ B(1^y + 2 + 3) \circ C(1 + 2^y) \quad (\text{PAR, STRUCT})
\end{aligned}$$

Damit ist gezeigt, dass der Ableitungsschritt $K \xrightarrow{\tau} K'$ von Beispiel 4.2.8 existiert. \square

Wir weisen nun auf eine Eigenschaft der Transitionsrelation hin, die für die Modellierung biologischer Sachverhalte bedeutsam ist.

Beobachtung 4.2.12 Wenn K wohlgeformt ist und $K \xrightarrow{\tau} K'$, so ist auch K' wohlgeformt. \square

Wie bereits weiter oben motiviert wurde, können biologische Systemzustände stets durch wohlgeformte Konfigurationen repräsentiert werden. Die Transitionsrelation $\xrightarrow{\tau}$ bewahrt diese Eigenschaft und gibt uns damit die Gewissheit, mit jeder auf sie beschränkten Transitionsableitung eine biologisch sinnvolle Konfiguration zu erreichen, wenn wir von einer solchen ausgegangen sind.

4.2.1 Weitere Betrachtungen

Wir gehen nun kurz auf einige weitere bedeutsame Aspekte des κ -Kalküls ein.

Vereinfachung der Transitionsrelation

Die separate Behandlung von linken und rechten Reaktionsseiten bei COM und DEC ist nicht notwendig und lässt die Transitionsrelation unnötig kompliziert erscheinen. In der Tat können wir sogar auf die Regel REACT verzichten, wenn wir stattdessen COM modifizieren zu

$$\text{COM}' \frac{((A, i, \varphi, \varphi'), (B, j, \psi, \psi')) \in \mathfrak{C} \quad x \in \mathcal{E} \setminus \text{en}(A(i + \varphi + \varphi'_0) \circ B(j + \psi + \psi'_0))}{A(i + \varphi + \varphi'_0) \circ B(j + \psi + \psi'_0) \longrightarrow A(i^x + \bar{\varphi} + \varphi'_0) \circ B(j^x + \bar{\psi} + \psi'_0)}$$

mit $\varphi'_0 = \varphi' + \varphi_0$ und $\psi'_0 = \psi' + \psi_0$

und DEC entsprechend zu DEC'. Dadurch wären auch Label unnötig, denn es gäbe keine Teilreaktionen und deshalb keine Notwendigkeit zur Unterscheidung einzelner Arten von Reaktionen. Wir wählen dennoch die ausführlichere Definition, da diese im Hinblick auf die Erweiterung des κ -Kalküls um Konzepte der Membranmaschine von Bedeutung sein wird.

Phosphorylierung und Dephosphorylierung

Der κ -Kalkül verzichtet auf die Modellierung von Phosphorylierungsstellen. Wie wir aber schon am Anfang des Kapitels angedeutet haben, kann deren Funktionalität dennoch modelliert werden. Dies geschieht über Enzyme, die bei nahezu jeder Phosphorylierung oder Dephosphorylierung auftreten und durch das temporäre Andocken an das Protein die Reaktion beschleunigen. Da es sich bei Enzymen auch um Proteine handelt, können diese problemlos in das Modell integriert und über sie die Phosphorylierung oder Dephosphorylierung modelliert werden. Die mit der Aktivierung oder Deaktivierung einer Phosphorylierungsstelle verbundene Änderung des Zustandes verschiedener Bindungsstellen am Protein kann dann mithilfe einer Komplexierungsreaktion von Protein und Enzym und der unmittelbar darauf folgenden Dekomplexierungsreaktion dargestellt werden. Wir demonstrieren den Vorgang an dem folgenden Beispiel.

Beispiel 4.2.13 (Phosphorylierung) Sei $A(\bar{1} + 2)$ ein Protein, welches durch Aktivierung einer (nicht modellierten) Phosphorylierungsstelle den Zustand seiner Bindungsstelle 1 von inaktiv auf aktiv ändern kann. $E(1)$ sei ein Enzym, welches diesen Vorgang beschleunigt. Wir definieren die Regelmenge $\mathcal{R} = \mathfrak{C} \cup \mathfrak{D}$ mit

$$\mathfrak{C} = \{((A, 2, \bar{1}, \emptyset), (E, 1, \emptyset, \emptyset))\} \quad \text{und} \quad \mathfrak{D} = \{((A, 2, \emptyset, \emptyset), (E, 1, \emptyset, \emptyset))\}$$

und ermöglichen damit die folgende Transitionsableitung.

$$\begin{aligned} & A(\bar{1} + 2) \circ E(1) \\ & \xrightarrow{\tau} A(1 + 2^x) \circ E(1^x) \\ & \xrightarrow{\tau} A(1 + 2) \circ E(1) \end{aligned}$$

Die Endkonfiguration unterscheidet sich von der Anfangskonfiguration nur in der aktivierten Bindungsstelle 1 von A , es wurde also lediglich die gewünschte Aktivierung erreicht. Das Enzym $E(1)$ liegt, wie in der Natur auch, nach der Reaktion unverändert vor. \square

Abweichungen vom ursprünglichen κ -Kalkül

Der κ -Kalkül in seiner ursprünglichen in [23] eingeführten Version verwendet eine semantische Struktur, die von der hier vorgestellten abweicht. Weil das Ziel der Einführung des κ -Kalküls in dieser Arbeit jedoch das Einführen zum Aspekten der Membran-Maschine integrierenden $\text{bio}\kappa$ -Kalkül ist, haben wir die Semantik bereits im Vorfeld an diesen angepasst. Inhaltlich lassen sich zwei wesentliche Unterschiede feststellen:

- *Proteindegeneration und -synthese:* In der Semantik des κ -Kalküls von [23] werden Reaktionen der Form $A(\varphi) \circ B(\psi) \xrightarrow{\mu} A(\varphi)$ und $A(\varphi) \xrightarrow{\mu} A(\varphi) \circ B(\psi)$ zugelassen, durch das Löschen und Hinzufügen von Proteinen in Konfigurationen wird also mehr Ausdrucksstärke erreicht. Jedoch kann jeder Zerfall von Proteinen in Bausteine auf Niveau von Aminosäuren und auch die Erzeugung von Proteinen bei Bereitstellung der dazu benötigten Aminosäuren auch mit der hier vorgestellten Semantik modelliert werden. Der Vorteil der zusätzlichen Operationen liegt in der Fähigkeit zur Modellierung des Proteinzerfalls in atomare Bausteine, die nicht weiter modelliert werden und damit aus der Konfiguration verschwinden, und der einfacheren Modellierung von Proteinsynthese ohne die eigentlich notwendige zusätzliche Bereitstellung von Teilbausteinen.
- *kontextabhängige Reaktionen:* Gemäß den Modellierungen des $\text{bio}\kappa$ -Kalküls ([30]) sind die in dieser Arbeit vorgestellten Reaktionen des κ -Kalküls binär, es werden also stets genau zwei Proteine angegeben, die miteinander reagieren. Diese Interaktionen sind per Definition unabhängig von ihrem Kontext. Dadurch ist es nicht möglich, Reaktionen zu modellieren, welche die Anwesenheit von selbst nicht an der Reaktion teilnehmenden Proteinen voraussetzen. Beispielsweise würde ein Ableitungsschritt $A(1) \circ B(1) \circ C(\varphi) \xrightarrow{\tau} A(1^x) \circ B(1^x) \circ C(\varphi)$ stets auch ohne die Anwesenheit von $C(\varphi)$ stattfinden können. Für die von uns vorgestellten biologischen Sachverhalte sind derartige Kontexte nicht notwendig. Allgemein scheint diese Vereinfachung für die Modellierbarkeit biologischer Sachverhalte keine Einschränkung darzustellen, da Proteininteraktionen auf lokalen Vorgängen an Bindungsstellen basieren und sich die beteiligten Molekülstrukturen nur in unmittelbarer Nähe beeinflussen können. Wir verzichten an dieser Stelle jedoch auf eine genauere Untersuchung dieser Fragestellung.
Der κ -Kalkül in [23] ist nicht auf binäre Reaktionen beschränkt, jedoch wird eine Erweiterung diskutiert, welche seine Konzepte mithilfe von binären Reaktionen

umsetzt. Diese als $m\kappa$ -Kalkül bezeichnete Variante wurde im Hinblick auf eine Transformation des κ -Kalküls in den π -Kalkül entwickelt und orientiert sich deshalb weniger stark an biologischen Konzepten.

Allgemein lässt sich die angesprochene Thematik auf das eigenständige Problem der *Selbstassemblierung* zurückführen. Die auf Graphen basierenden Betrachtungen von [26] beispielsweise gehen darauf ein, in welchem Maße bestimmte Eigenschaften und Funktionen komplexer Strukturen allein durch lokale Interaktionen einzelner Komponenten erreichbar sind. Diese Fragestellung ist nicht nur in der Systembiologie, sondern unter anderem auch in der Nanotechnologie und Robotik von Bedeutung.

4.3 Anwendungsbeispiel: HIV-Proteininteraktionen

Mit dem im Folgenden vorgestellten Anwendungsbeispiel knüpfen wir an die Ausführungen von Abschnitt 3.5.2 an. Im Mittelpunkt stehen nun die im Brane-Kalkül lediglich indirekt modellierten Proteininteraktionen, welche den Ablauf der vorgestellten Prozesse erst ermöglichen.

4.3.1 Modellierung

Die mit dem HIV im Zusammenhang stehenden Proteininteraktionen besitzen ein enormes Ausmaß an Komplexität und sind darüberhinaus an vielen Stellen nicht bis ins letzte Detail erforscht und verstanden. Die folgenden Ausführungen können somit nur exemplarisch und stark vereinfachend auf einige der stattfindenden Mechanismen eingehen.

Andockprozess von HIV-Partikeln an CD4-positive Zellen

Der an der Oberfläche von Viruspartikeln befindliche Komplex aus den Glycoproteinen gp120 und gp41 ist entscheidend für das Andocken einzelner Viruspartikel an Zellen, welche den CD4-Rezeptor auf ihrer Membran tragen. Die Interaktion von gp120 und CD4 bewirkt eine Konformationsänderung im gp120, welche zur Freilegung einer fusogenen Domäne im gp41 führt, die schließlich die Fusion von Virushülle und Zellmembran zur Folge hat.

Die Modellierung des gp120-gp41-Komplexes kann folgendermaßen aussehen:

$$GP_{120}(1^x + 2 + \bar{3} + 4) \circ GP_{41}(1^x + 2 + \bar{3}).$$

Hierbei repräsentiert x an den jeweiligen Bindungsstellen 1 die Verbindung der beiden Glycoproteine, die Funktion der anderen Bindungsstellen wird erst an späterer Stelle deutlich. Modellieren wir nun CD4 mit $CD_4(1)$ und betrachten die Komplexierungsregel

$$r_1 = ((GP_{120}, 2, \bar{3}, 4), (CD_4, 1, \emptyset, \emptyset)),$$

so ergibt sich für den ersten Andockprozess die Transitionsableitung

$$\begin{aligned} & GP_{120}(1^x + 2 + \bar{3} + 4) \circ GP_{41}(1^x + 2 + \bar{3}) \circ CD_4(1) \\ \xrightarrow{\tau} & GP_{120}(1^x + 2^y + 3 + 4) \circ GP_{41}(1^x + 2 + \bar{3}) \circ CD_4(1^y) \end{aligned}$$

Die Konformationsänderung im gp120 wird durch Aktivierung seiner Bindungsstelle 3 dargestellt. Unter Einführung der Komplexierungsregel

$$r_2 = ((GP_{120}, 3, \emptyset, 1^x), (GP_{41}, 2, \bar{3}, 1^x))$$

kann anschließend die Aktivierung der fusogenen Domäne im gp41 durch Aktivierung seiner Bindungsstelle 3 modelliert werden:

$$\begin{aligned} & GP_{120}(1^x + 2^y + 3 + 4) \circ GP_{41}(1^x + 2 + \bar{3}) \circ CD_4(1^y) \\ \xrightarrow{\tau} & GP_{120}(1^x + 2^y + 3^z + 4) \circ GP_{41}(1^x + 2^z + 3) \circ CD_4(1^y) \end{aligned}$$

Entscheidend für den korrekten Ablauf der letzten Interaktion ist die vorherige Überprüfung auf eine Verbindung zwischen den Proteinen gp120 und gp41, da anderenfalls in der Modellierung die nicht beabsichtigte Interaktion eines beliebigen gp120-Moleküls mit einem beliebigen gp41-Molekül die Folge sein könnte.

Freie Proteine

Die vorgestellte Modellierung deckt bereits den Fall der Anlagerung von freiem gp120 an den CD4-Rezeptor ab, der einzige Unterschied zu membranständigem gp120 besteht in der Nichtbindung an ein gp41-Molekül. Unter erneuter Betrachtung der Komplexierungsregel r_1 resultiert deshalb auch die Transitionsableitung

$$GP_{120}(1 + 2 + \bar{3} + 4) \circ CD_4(1) \xrightarrow{\tau} GP_{120}(1 + 2^y + 3 + 4) \circ CD_4(1^y)$$

Die Anlagerung von freiem gp120 macht die Zelle anfällig für IgG-Antikörper. Diesem Effekt kann durch die Komplexierungsregel

$$r_3 = ((IgG, 1, \emptyset, \emptyset), (GP_{120}, 4, \emptyset, \emptyset))$$

Rechnung getragen werden. Ausgehend vom Resultat der letzten Transitionsableitung können wir Folgendes ableiten:

$$\begin{aligned} & IgG(1) \circ GP_{120}(1 + 2^y + 3 + 4) \circ CD_4(1^y) \\ \xrightarrow{\tau} & IgG(1^z) \circ GP_{120}(1 + 2^y + 3 + 4^z) \circ CD_4(1^y) \end{aligned}$$

IgG-Antikörper sind darüber hinaus in der Lage, sowohl an frei bewegliches gp120 als auch an virusmembranständiges gp120 zu binden. Beide Effekte werden bereits durch die Komplexierungsregel r_3 abgedeckt, es ergeben sich die Transitionsableitungen

$$IgG(1) \circ GP_{120}(1 + 2 + \bar{3} + 4) \xrightarrow{\tau} IgG(1^z) \circ GP_{120}(1 + 2 + \bar{3} + 4^z)$$

und

$$\begin{aligned} & IgG(1) \circ GP_{120}(1^x + 2 + \bar{3} + 4) \circ GP_{41}(1^x + 2 + \bar{3}) \\ \xrightarrow{\tau} & IgG(1^z) \circ GP_{120}(1^x + 2 + \bar{3} + 4^z) \circ GP_{41}(1^x + 2 + \bar{3}) \end{aligned}$$

Nun wird deutlich, warum mit r_1 die Bindungsstelle 4 in A überprüft werden muss. Sollte sich ein gp120-Molekül bereits in Verbindung mit einem IgG-Antikörper befinden, ist es zu keiner Interaktion mit membranständigem CD4 mehr fähig, in der Modellierung findet entsprechend die mit r_1 in Verbindung stehende Reaktion bei belegter Bindungsstelle 4 nicht statt.

4.3.2 Diskussion

Die Modellierungen im κ -Kalkül konzentrieren sich ausschließlich auf Proteininteraktionen und bieten eine gute Möglichkeit zu ihrer formalen und übersichtlichen Beschreibung. Dies zeichnet ihn insbesondere zur Darstellung und Simulation von Signaltransduktionswegen aus, einige Ansätze hierzu werden beispielsweise in [18], [19], [20] und [21] diskutiert. Betrachtungen dieser Art sind allerdings völlig losgelöst von ihrem Kontext und gehen somit in keiner Weise auf die Bedeutungen ein, die Proteininteraktionen innerhalb ihrer Umgebung und insbesondere im Zusammenhang mit Membranen haben. Beispielsweise ist in unserer Anwendung bedeutsam, dass der IgG-Antikörper die Funktion des gp120 blockieren und damit eine Membranfusion verhindern kann. Während das Belegen einer Bindungsstelle im gp120 das Blockieren gut modelliert, gibt es jedoch keine Möglichkeit zur Darstellung der damit verbundenen Auswirkung. Bei der Betrachtung komplexerer Systeme lässt der Ansatz zudem die vielfach essentielle räumlichen Abgrenzung verschiedener Reaktionsräume vermissen. So besteht beispielsweise keine Möglichkeit, eine Unterscheidung zwischen freiem und membranständigem gp120 vorzunehmen.

Erst in Kombination mit anderen Konzepten kann der κ -Kalkül zu einem umfassenden Modellierungswerkzeug werden und seine Stärken voll entfalten. Eine Entwicklung in diese Richtung stellt die von Cosimo Laneve und Fabien Tarissan entwickelte Erweiterung mit dem Namen $\text{bio}\kappa$ -Kalkül dar. Neben Konzepten der räumlichen Abgrenzung durch Membranstrukturen bettet sie auch einige Membranoperationen in den κ -Kalkül ein.

4.4 Integration von Konzepten der Membran-Maschine

Wir stellen nun die von Cosimo Laneve und Fabien Tarissan entwickelte Erweiterung des κ -Kalküls zum $\text{bio}\kappa$ -Kalkül vor. Dabei gehen wir in mehreren Schritten vor und reichen jeweils die Transitionsrelation $\longrightarrow_{\mathcal{R}}$ um die für die besprochenen Erweiterungen benötigten Regeln an.

4.4.1 Erweiterung um Membranen

Im ersten Schritt führen wir das Konzept der räumlichen Abgrenzung durch Membranen ein. Hierfür benötigen wir zunächst eine erweiterte Syntax. Diese wird im Folgenden angegeben und stellt bereits die endgültige Syntax des $\text{bio}\kappa$ -Kalküls dar.

Definition 4.4.1 ($\text{bio}\kappa$ -Kalkül-Syntax) Die Menge der *Systeme* im $\text{bio}\kappa$ -Kalkül wird mit \mathcal{S}_{κ} notiert und ist gegeben durch die folgende Grammatik:

$$S ::= K \mid [K]_{\mathbf{M}}(S) \mid S \circ S.$$

Dabei steht K für beliebige Konfigurationen gemäß der in Definition 4.1.4 gegebenen Grammatik und \mathbf{M} für beliebige Elemente aus der Menge \mathcal{B} der *Membrannamen*. \square

Jedes $[K]_{\mathbf{M}}(S)$ repräsentiert ein Membransystem. Hierbei werden dessen membranständige Proteine innerhalb der eckigen Klammern als Konfiguration K notiert und ihr Inhalt wie schon beim Brane-Kalkül in speziellen runden Klammern. Auch an dieser

Stelle ist durch die Struktur der Syntax eine beliebige Schachtelung derartiger Membransysteme möglich. Der Membrannamen M dient ihrer eindeutigen Bezeichnung und wird für spätere Membraninteraktionen benötigt. Wir vereinbaren zunächst ein paar notationelle Vereinfachungen. So verzichten wir auf die Angabe von Membrannamen M , wenn diese nicht benötigt werden, und schreiben $[K](\emptyset)$ für $[K](\diamond)$.

Bevor wir die Transitionsrelation des κ -Kalküls um die Möglichkeit von Proteininteraktionen innerhalb von Systemen anreichern, müssen wir auch einige zugrunde liegende Definitionen anpassen. Aufgrund des Zusammenhangs $\mathcal{K} \subseteq \mathcal{S}_\kappa$ können wir dabei auf den bekannten Definitionen aufbauen und behalten die eingeführten Bezeichnungen bei. Zunächst erweitern wir die Strukturelle Kongruenz.

Definition 4.4.2 (Strukturelle Kongruenz von Systemen) Die *Strukturelle Kongruenz* $\equiv \subseteq \mathcal{S}_\kappa \times \mathcal{S}_\kappa$ ist die kleinste Äquivalenzrelation, welche die Regeln (1) bis (4) aus Definition 4.2.9 bezogen auf den auf Systeme erweiterten Definitionsbereich erfüllt und zusätzlich folgender Regel genügt:

$$(5) \quad K \in \mathcal{K} \wedge K \equiv K' \wedge S \equiv S' \Rightarrow [K]_M(S) \equiv [K']_M(S') \quad \square$$

Auch die Funktion *occ* zur Bestimmung der Anzahl von verwendeten Verbindungsnamen in Konfigurationen muss auf Systeme erweitert werden. Dies kann unter Verwendung von Definition 4.2.1 durch Erweitern des Definitionsbereiches und Hinzufügen der Zeile

$$occ(x, [K]_M(S)) = occ(x, K) + occ(x, S)$$

geschehen. Die auf Systeme angepassten Definitionen von *en* und *diff* ergeben sich dann aus der modifizierten Definition der ihnen zugrunde liegenden Funktion *occ*. Die Anpassung von $=_\alpha$ lässt sich leicht entwickeln. Mit der folgenden Erweiterung der Transitionsrelation ermöglichen wir Proteininteraktionen im Zusammenhang mit Membranen und verwenden weiterhin dasselbe Relationssymbol.

Definition 4.4.3 (Erweiterung von $\longrightarrow_{\mathcal{R}}$) $\longrightarrow_{\mathcal{R}} \subseteq \mathcal{S}_\kappa \times \mathcal{L} \times \mathcal{S}_\kappa$ ist die kleinste Relation, die alle Transitionsregeln aus Definition 4.2.10 bezogen auf den auf Systeme erweiterten Definitionsbereich erfüllt und zusätzlich folgenden Transitionsregeln genügt:

$$\text{MEM} \quad \frac{K \xrightarrow{\mu} K' \quad diff(K, K', S)}{[K]_M(S) \xrightarrow{\mu} [K']_M(S)} \quad \text{BRANE} \quad \frac{S \xrightarrow{\tau} S' \quad diff(S, S', K)}{[K]_M(S) \xrightarrow{\tau} [K]_M(S')}$$

$$\text{MREACT} \quad \frac{K \xrightarrow{r_m^x} K' \quad S \xrightarrow{r_n^x} S' \quad m \neq n}{[K]_M(S) \xrightarrow{\tau} [K']_M(S')} \quad \square$$

MEM: Für den Fall $\mu = \tau$ stellt diese Transitionsregel sicher, dass Interaktionen von Proteinen auch auf einer Membran stattfinden können. Dabei wird zur Wahrung der Wohlgeformtheit überprüft, ob der bei der Reaktion entstehende oder verschwindende Verbindungsname nicht auch im Kontext vorkommt.

Für den Fall $\mu = r_n^x$ beschreibt sie eine Teilreaktion eines Proteins auf der Membran unter Beibehaltung des Labels. Zusammen mit der Transitionsregel REACT

aus Definition 4.2.10 ermöglicht dies die Interaktion mit einem Protein außerhalb der Membran, welches entweder frei oder ebenfalls membranständig auftritt. Wir setzen zusätzlich $K \in \mathcal{K}$ voraus, um bei der Einbettung von K in die Membran innerhalb der in Definition 4.4.1 gegebenen Struktur von Systemen zu bleiben, über der die Transitionsrelation $\longrightarrow_{\mathcal{R}}$ definiert ist. Wir verzichten hier und auch bei allen weiteren Transitionsregeln auf eine explizite Angabe derartiger Bedingungen.

BRANE: Die abgeschlossene Reaktion eines Systems S wird auch innerhalb einer Membran M ermöglicht. Wir stellen fest, dass die Transitionsregel das Label τ verwendet, somit keine Teilreaktionen einbezieht und damit die Interaktion von freien Proteinen unterbindet, die sich innerhalb verschiedener Membranen befinden. Auch hier wird das Auftreten des entstehenden oder verschwindenden Verbindungsnamen im Kontext ausgeschlossen und damit die Wahrung von Wohlgeformtheit sichergestellt.

MREACT: Ähnlich zur Transitionsregel **REACT** aus Definition 4.2.10 werden zwei Teilreaktionen zusammengeführt, jedoch befinden sich die zugehörigen Reaktionspartner hier nicht in unabhängiger Komposition zueinander, sondern sind ineinander verschachtelt. Dies macht im Gegensatz zu den Labels r_1^x und r_2^x in **REACT** die allgemeineren Label r_m^x und r_n^x mit $m, n \in \{1, 2\}$ und $m \neq n$ erforderlich, um unabhängig von der Reihenfolge der Reaktionspartner in der Regelspezifikation zu bleiben.

MREACT ermöglicht die Interaktion von membranständigen Proteinen auf einer äußeren Membran mit solchen, die sich frei in dieser oder auf einer von dieser Membran umgebenen inneren Membran befinden.

Beispiel 4.4.4 (Herleitung von Ableitungsschritten) Wir betrachten die Komplexierungsregel $r = ((A, 1, \emptyset, \emptyset), (B, 1, \emptyset, \emptyset))$. Die durch zweimaliges Anwenden der Transitionsregel **COM** erlangten Teilreaktionen

$$A(1) \xrightarrow{r_1^x} A(1^x), B(1) \xrightarrow{r_2^x} B(1^x)$$

referenzieren wir im Folgenden mit **TR**. Bezeichnet A ein freies und B ein auf einer Membran befindliches Protein, so kann A sowohl aus dem Inneren dieser Membran mit B interagieren durch

$$\text{TR} \Rightarrow [B(1)] \langle A(1) \rangle \xrightarrow{\tau} [B(1^x)] \langle A(1^x) \rangle \quad (\text{MREACT})$$

als auch von außen durch

$$\begin{aligned} \text{TR} &\Rightarrow A(1) \xrightarrow{r_1^x} A(1^x), [B(1)] \langle \rangle \xrightarrow{r_2^x} [B(1^x)] \langle \rangle && (\text{MEM}) \\ &\Rightarrow A(1) \circ [B(1)] \langle \rangle \xrightarrow{\tau} A(1^x) \circ [B(1^x)] \langle \rangle && (\text{REACT}) \end{aligned}$$

Bezeichnen A und B auf verschiedenen Membranen befindliche Proteine, finden Interaktionen bei Nebeneinanderliegen dieser Membranen statt durch

$$\begin{aligned} \text{TR} &\Rightarrow [A(1)] \langle \rangle \xrightarrow{r_1^x} [A(1^x)] \langle \rangle, [B(1)] \langle \rangle \xrightarrow{r_2^x} [B(1^x)] \langle \rangle && (\text{MEM, MEM}) \\ &\Rightarrow [A(1)] \langle \rangle \circ [B(1)] \langle \rangle \xrightarrow{\tau} [A(1^x)] \langle \rangle \circ [B(1^x)] \langle \rangle && (\text{REACT}) \end{aligned}$$

und bei direktem Ineinanderliegen durch

$$\begin{aligned} \text{TR} &\Rightarrow A(1) \xrightarrow{r_1^x} A(1^x), [B(1)] \llbracket \Rightarrow [B(1^x)] \llbracket && (\text{MEM}) \\ &\Rightarrow [A(1)] \llbracket [B(1)] \llbracket \xrightarrow{\tau} [A(1^x)] \llbracket [B(1^x)] \llbracket && (\text{MREACT}) \end{aligned}$$

Interaktionen von Proteinen, die mindestens eine Membran zwischen sich haben, sind hingegen nicht möglich. \square

Es wird nun deutlich, warum in der Definition der Transitionsregeln COM und DEC eine Aufteilung in Teilreaktionen sinnvoll ist. Verzichtet man auf diese Differenzierung, benötigt man eine zusätzliche Transitionsregel für jede der in Beispiel 4.4.4 gezeigten Interaktionen.

Die Wohlgeformtheit eines Terms des $\text{bio}\kappa$ -Kalküls hängt nicht mehr allein vom paarweisen Auftreten von Verbindungsnamen ab, sondern muss auch noch berücksichtigen, dass Proteine nicht über Membrangrenzen hinweg miteinander verbunden sein können. Demzufolge besitzen wohlgeformte Systeme der Art $[K] \llbracket T \rrbracket$ sämtliche Verbindungsnamen, die in T nur einmal vorkommen, genau einmal auch in K . Wir geben zunächst die Funktion de zur Bestimmung aller genau einmal vorkommenden Verbindungsnamen (dangling edges) an und erweitern danach den Begriff der Wohlgeformtheit auf Systeme.

Definition 4.4.5 (de) Die Funktion $de : \mathcal{S}_\kappa \rightarrow 2^\mathcal{E}$ ist für alle $S \in \mathcal{S}_\kappa$ definiert als

$$de(S) = \{x \mid occ(x, S) = 1\}. \quad \square$$

Definition 4.4.6 (Wohlgeformtheit von Systemen) Sei $S \in \mathcal{S}_\kappa$. S heißt *geschlossen*, wenn für alle $x \in \mathcal{E}$ $occ(x, S) = 0$ oder $occ(x, S) = 2$ gilt. S heißt *geordnet*, wenn $ord(S)$ mithilfe der folgenden Regeln abgeleitet werden kann:

$$\begin{aligned} &ord(\diamond) \\ &ord(A(\varphi)) \\ ord(S) \wedge ord(T) &\Rightarrow ord(S \circ T) \\ ord(S) \wedge de(S) \subseteq de(K) &\Rightarrow ord([K] \llbracket S \rrbracket) \end{aligned}$$

S heißt *wohlgeformt*, wenn es geschlossen und geordnet ist. \square

4.4.2 Membranfusion und -aktivierung

Abgesehen von der räumlichen Trennung einzelner Proteine spielen Membranstrukturen bisher keine Rolle bei der Interaktion. Die folgende Erweiterung ermöglicht die Angabe von zur Fusion fähigen Membranen und realisiert damit die Operationen Exozytose und Membranfusion der Membranmaschine. Grundlage der Erweiterung ist die Einführung von *Fusionsregeln*.

Definition 4.4.7 (Fusionsregel) Eine *Fusionsregel* r ist ein Tupel

$$r = (M_1, M_2, N), \text{ wobei } M_1, M_2, N \in \mathcal{B}. \quad \square$$

Eine Fusionsregel spezifiziert unter Angabe der Namen M_1 und M_2 zwei Membranen, die miteinander verschmelzen können zu einer neuen Membran mit Namen N . Die zugehörigen Transitionsregeln werden im Folgenden angegeben.

Definition 4.4.8 (Erweiterung von $\longrightarrow_{\mathcal{R}}$) Wir erweitern das Tupel von Regelmengen zu $\mathcal{R} = (\mathcal{C}, \mathcal{D}, \mathfrak{F})$, wobei \mathfrak{F} eine Menge von Fusionsregeln angibt. Die Transitionsrelation $\longrightarrow_{\mathcal{R}}$ erfülle nun auch folgende Transitionsregeln:

$$\text{MATE} \frac{(M_1, M_2, N) \in \mathfrak{F}}{[K_1]_{M_1}(S_1) \circ [K_2]_{M_2}(S_2) \xrightarrow{\tau} [K_1 \circ K_2]_N(S_1 \circ S_2)}$$

$$\text{EXO} \frac{(M_1, M_2, N) \in \mathfrak{F}}{[K_1]_{M_1}([K_2]_{M_2}(S) \circ T) \xrightarrow{\tau} S \circ [K_1 \circ K_2]_N(T)} \quad \square$$

MATE: Diese Transition ermöglicht die Fusion zweier nebeneinanderliegender Membranen M_1 und M_2 . Im Zuge der Reaktion werden sowohl die jeweils membranständigen Proteine, dargestellt als Konfigurationen K_1 und K_2 , als auch deren Inhalte S_1 und S_2 den entsprechenden Bereichen der neu entstehenden Membran mit dem Namen N zugeordnet.

EXO: Die Funktionsweise ist analog zu MATE, allerdings fusionieren hier zwei Membranen, die direkt ineinander liegen statt nebeneinander. Auf gleiche Weise werden die membranständigen Proteine zusammengelegt, der Inhalt S der inneren Membran verlässt jedoch die Membranen.

Oftmals spielt die Interaktion von Proteinen eine Rolle, bevor zwei Membranen miteinander interagieren. Wir erinnern uns an den Andockprozess eines HIV-Partikels an eine Helfer-T-Zelle, welcher durch das Aktivieren der fusogenen Domäne des gp41 initiiert wird. Wir stellen im Folgenden ein Konzept vor, welches diese Membranaktivierung realisiert.

Definition 4.4.9 (Aktivierungsregel) Eine *Aktivierungsregel* r ist ein Tupel

$$r = (r', i, M, N), \text{ wobei } r' \in \mathcal{C} \cup \mathcal{D}, i \in \{1, 2\} \text{ und } M, N \in \mathcal{B}. \quad \square$$

Jede Aktivierungsregel bezieht sich auf eine Komplexierungs- oder Dekomplexierungsregel r' und spezifiziert zusätzlich die Änderung des Membrannamens von M zu N im Zuge der Teilreaktion der durch i angegebenen Regelseite in r' .

Definition 4.4.10 (Erweiterung von $\longrightarrow_{\mathcal{R}}$) Wir erweitern das Tupel von Regelmengen zu $\mathcal{R} = (\mathcal{C}, \mathcal{D}, \mathfrak{F}, \mathfrak{A})$, wobei \mathfrak{A} eine Menge von Aktivierungsregeln angibt. Die Transitionsrelation $\longrightarrow_{\mathcal{R}}$ erfülle nun auch folgende Transitionsregeln:

$$\text{MEM-A} \frac{(r, i, M, N) \in \mathfrak{A} \quad K \xrightarrow{r'_i} K' \quad \text{diff}(K, K', S)}{[K]_M(S) \xrightarrow{r'_i} [K']_N(S)}$$

$$\text{MREACT-A} \frac{(r, m, M, N) \in \mathfrak{A} \quad K \xrightarrow{r'_m} K' \quad S \xrightarrow{r'_n} S' \quad m \neq n}{[K]_M(S) \xrightarrow{\tau} [K']_N(S')} \quad \square$$

Die Transitionsregeln MEM-A bzw. MREACT-A entsprechen im Wesentlichen den Transitionsregeln MEM bzw. MREACT, betten aber zusätzlich die Aktivierung von Membranen gemäß angegebener Aktivierungsregeln in \mathfrak{A} ein. Wir sehen, dass auch hier das Konzept der Teilreaktionen zur Einsparung einiger Transitionsregeln beiträgt. Mit der Einführung der beiden Transitionsregeln MEM-A und MREACT-A kommt es zu einer Unstimmigkeit: trotz Angabe einer Aktivierungsregel kann die zugehörige Komplexierung oder Dekomplexierung auch ohne Änderung des Membrannamens stattfinden aufgrund der Transitionsregeln MEM bzw. MREACT, dies muss durch zusätzliche Bedingungen in deren Spezifikation verhindert werden. Wir verzichten an dieser Stelle auf eine formale Anpassung und widmen uns einem Beispiel zur Verdeutlichung der Funktionsweise von Membranfusion und -aktivierung.

Beispiel 4.4.11 (Aktivierung mit anschließender Fusion) Wir betrachten $\mathcal{R} = (\{r_c\}, \emptyset, \{r_f\}, \{r_a\})$ mit

$$\begin{aligned} r_c &= ((A, 1, \emptyset, \emptyset), (B, 1, \emptyset, \emptyset)) \\ r_f &= (M_a, N, M_a) \\ r_a &= (r_c, 2, M, M_a) \end{aligned}$$

und das System S mit

$$S = [A(1)]_N(T) \circ [B(1)]_M([A(1)]_N(T)).$$

Die Aktivierungsregel r_a nimmt Bezug auf die Komplexierungsregel r_c , speziell auf den durch den Index 2 angegebenen Reaktionspartner B von r_c . Eine Aktivierung der Membran M zu M_a erfolgt also durch Änderung von B im Zuge der Komplexierung mit einem Protein A . Die Reaktion kann mit dem Vorkommen A innerhalb (durch MREACT-A) oder außerhalb des Teilsystems $[B(1)]_M([A(1)]_N(T))$ (durch MEM-A und REACT) stattfinden. Letzteres führt zu einem System

$$S' = [A(1^x)]_N(T) \circ [B(1^x)]_{M_a}([A(1)]_N(T))$$

und ermöglicht aufgrund der Fusionsregel r_f die Fusion der beiden Teilsysteme $[A(1^x)]_N(T)$ und $[B(1^x)]_{M_a}([A(1)]_N(T))$ (durch MATE) oder die Fusion der ineinanderliegenden Membranen von $[B(1^x)]_{M_a}([A(1)]_N(T))$ (durch EXO). Wiederum letzteres führt zu dem System

$$S'' = [A(1^x)]_N(T) \circ T \circ [B(1^x) \circ A(1)]_{M_a}(\emptyset). \quad \square$$

4.4.3 Transmembraner Transport und Phagozytose

Vergleichbar zur B&R-Operation im Brane-Kalkül gibt es auch im bio κ -Kalkül eine Möglichkeit zur Spezifikation von transmembranem Molekültransport. Im Unterschied dazu konzentriert sich dieser allerdings auf einzelne Proteine und verbindet spezielle Proteininteraktionen und Veränderungen der Membraneigenschaften mit der Interaktion. Wie schon im vorigen Abschnitt geben wir bei der formalen Definition zunächst den benötigten Regeltyp und anschließend die Erweiterung der Transitionsrelation an.

Definition 4.4.12 (Transportregel) Eine *Transportregel* r ist ein Tupel

$$r = ((A, i, \varphi, \varphi'), (B, j, \psi, \psi'), \mathbf{M}, \mathbf{N}),$$

wobei $((A, i, \varphi, \varphi'), (B, j, \psi, \psi'))$ eine Reaktionsregel gemäß Definition 4.2.7 bezeichnet mit der Einschränkung $\varphi' \in \mathcal{I}_{vh}$. Desweiteren gilt $\mathbf{M}, \mathbf{N} \in \mathcal{B}$. \square

A bezeichnet ein frei vorkommendes Protein. Mithilfe der nachfolgend angegebenen Transitionsregel wird gefordert, dass A vor dem Transport eine Verbindung zu dem Protein B der zu durchquerenden Membran \mathbf{M} besitzt, welche nach dem Transport gelöst wird. Diese Modellierung liegt nahe, weil membranständige Proteine maßgeblich am Molekültransport durch Membranen hindurch beteiligt sind. Die Einschränkung von φ' auf v - h -Interfaces erlaubt jedoch keine Überprüfung auf weitere vorhandene Kanten im Protein A . Solche dürfen auch nicht vorhanden sein, denn ein freies Protein kann nur dann eine Membran passieren, wenn es abgesehen von membranständigen Proteinen mit keinem anderen Protein verbunden ist. Diese Einschränkung ist notwendig, um die Wohlgeformtheit zu gewährleisten: die Verbindung von A mit einem anderen Protein C hätte nach dem Transport von A die in der Natur nicht mögliche Verbindung von A und C durch die Membran hindurch zur Folge. Im Zuge des Vorganges ändert sich der Name der transportierenden Membran \mathbf{M} zu \mathbf{N} und verhindert dadurch das unbegrenzte Eindringen von Proteinen durch wiederholtes Anwenden der gleichen Transportregel. Die folgende Transitionsregel formalisiert das Genannte.

Definition 4.4.13 (Erweiterung von $\longrightarrow_{\mathcal{R}}$) Wir erweitern das Tupel von Regelmengen zu $\mathcal{R} = (\mathcal{C}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \mathcal{T})$, wobei \mathcal{T} eine Menge von Transportregeln angibt. Die Transitionsrelation $\longrightarrow_{\mathcal{R}}$ erfülle nun auch folgende Transitionsregel:

$$\text{TRSP} \frac{((A, i, \varphi, \varphi'), (B, j, \psi, \psi'), \mathbf{M}, \mathbf{N}) \in \mathcal{T} \quad \varphi_0 \in \mathcal{I}_{vh}}{P_A \circ [P_B \circ K]_{\mathbf{M}}(S) \xrightarrow{\tau} [P'_B \circ K]_{\mathbf{N}}(P'_A \circ S)}$$

$$\text{mit } P_A = A(i^x + \varphi + \varphi' + \varphi_0), P_B = B(j^x + \psi + \psi' + \psi_0)$$

$$\text{und } P'_A = A(i + \bar{\varphi} + \varphi' + \varphi_0), P'_B = B(j + \bar{\psi} + \psi' + \psi_0) \quad \square$$

Wir betrachten nun eine Umsetzung der Phagozytose im bio κ -Kalkül. Bei dieser Membraninteraktion trennt sich eine neue Membran von einer vorhandenen ab und übernimmt einen Teil der mit ihr assoziierten Proteine. Dies kann realisiert werden durch explizite Angabe der übergehenden Proteine bei der Spezifikation von *Phagoregeln*, im Zuge der Reaktion werden schließlich genau diese Proteine von der ursprünglichen Membran entfernt und der neu entstehenden zugeordnet. Ähnlich zum Molekültransport wird auch hier die Verbindung von zwei jeweils auf einer Membran der beteiligten Systeme befindlichen Proteinen gefordert und im Zuge der Reaktion gelöst. Desweiteren bietet sich wiederum die zusätzliche Kontrolle der Interaktion durch Änderung von Membrannamen.

Definition 4.4.14 (Phagoregel) Eine *Phagoregel* r ist ein Tupel

$$r = ((A, i, \varphi, \varphi'), (B, j, \psi, \psi'), \mathbf{M}, \mathbf{N}, \mathbf{N}', \mathbf{N}'', K),$$

wobei $((A, i, \varphi, \varphi'), (B, j, \psi, \psi'))$ eine Reaktionsregel gemäß Definition 4.2.7 bezeichnet. Desweiteren gilt $\mathbf{M}, \mathbf{N}, \mathbf{N}', \mathbf{N}'' \in \mathcal{B}$ und $K \in \mathcal{K}$ mit $de(K) = \emptyset$. \square

An dieser Stelle bezeichnet A ein auf der Membran M des zu verschlingenden Systems befindliches Protein, B entsprechend ein der Membran N des verschlingenden Systems zugeordnetes Protein. Während der Reaktion ändert die verschlingende Membran ihren Namen zu N' , die neu entstehende wird mit N'' benannt und erhält die Proteine K , welche von der verschlingenden Membran N abgespalten werden. Die Forderung $de(K) = \emptyset$ stellt sicher, dass diese mit keinen Proteinen außerhalb der Membran in Verbindung stehen.

Die Definition der Transitionsregel für die Phagozytose benötigt noch einige zusätzliche Notationen. So ist es vor dem Abspalten einer Konfiguration K von einer Konfiguration K' notwendig zu überprüfen, ob K vollständig in K' enthalten ist. Trifft dies zu, so notieren wir $K \leq K'$. Desweiteren schreiben wir $K' \setminus K$ für die nach der Abspaltung übriggebliebenen Proteine. Die folgende Definition führt diese Schreibweisen formal ein.

Definition 4.4.15 (\leq, \setminus, \cap) Wir definieren $\leq \subseteq \mathcal{K} \times \mathcal{K}$ als

$$\leq := \{(K, K') \mid \exists K'' \in \mathcal{K} : K \circ K'' \equiv K'\}.$$

Desweiteren sei $- : \mathcal{K} \times \mathcal{P} \rightarrow \mathcal{K}$ für alle $K \in \mathcal{K}$ und $A(\varphi) \in \mathcal{P}$ definiert als

$$K - A(\varphi) = \begin{cases} \diamond & \text{für } K = A(\varphi) \\ K_e & \text{für } K = A(\varphi) \circ K_e \\ K_b \circ K_e & \text{für } K = K_b \circ A(\varphi) \circ K_e \text{ mit } A(\varphi) \not\leq K_b \\ K_b & \text{für } K = K_b \circ A(\varphi) \text{ mit } A(\varphi) \not\leq K_b \\ K & \text{sonst} \end{cases}$$

Darauf aufbauend definieren wir $\setminus : \mathcal{K} \times \mathcal{K} \rightarrow \mathcal{K}$ für alle $K, L \in \mathcal{K}$ und $A(\varphi) \in \mathcal{P}$ induktiv als

$$\begin{aligned} K \setminus \diamond &= K \\ K \setminus A(\varphi) &= K - A(\varphi) \\ K \setminus (A(\varphi) \circ L) &= (K - A(\varphi)) \setminus L \end{aligned}$$

und verwenden für alle $A, B \in \mathcal{K}$ die abkürzende Schreibweise

$$A \cap B := (A \circ B) \setminus ((A \setminus B) \circ (B \setminus A)). \quad \square$$

Definition 4.4.16 (Erweiterung von $\longrightarrow_{\mathcal{R}}$) Wir erweitern das Tupel von Regelmengen zu $\mathcal{R} = (\mathcal{C}, \mathcal{D}, \mathfrak{F}, \mathfrak{A}, \mathfrak{T}, \mathfrak{P})$, wobei \mathfrak{P} eine Menge von Phagoregeln angibt. Die Transitionsrelation $\longrightarrow_{\mathcal{R}}$ erfülle nun auch folgende Transitionsregel:

$$\text{PHAGO} \frac{((A, i, \varphi, \varphi'), (B, j, \psi, \psi'), M, N, N', N'', K) \in \mathfrak{P} \quad K \leq K'_B}{[K_A]_M(S) \circ [K_B]_N(T) \xrightarrow{\tau} [K'_B \setminus K]_{N'}([K]_{N''}([K'_A]_M(S)) \circ T)}$$

$$\text{mit } K_A = A(i^x + \varphi + \varphi' + \varphi_0) \circ K_1, K_B = B(j^x + \psi + \psi' + \psi_0) \circ K_2$$

$$\text{und } K'_A = A(i + \bar{\varphi} + \varphi' + \varphi_0) \circ K_1, K'_B = B(j + \bar{\psi} + \psi' + \psi_0) \circ K_2 \quad \square$$

An dieser Stelle sei angemerkt, dass die Phagozytose im bio κ -Kalkül die Wohlgeformtheit nicht bewahrt. Es wird keine Einschränkung bezüglich der in K_A möglichen Verbindungen vorgenommen, beispielsweise ist eine Verbindung zu einem externen Protein möglich und resultiert nach der Interaktion in einer Proteinverbindung durch zwei Membranen hindurch. Die Verwendung der Phagozytose in der Modellierung verlangt also eine gewisse Vorsicht, um biologisch plausible Strukturen nicht zu zerstören. Auch diesen Abschnitt wollen wir nun mit einem Beispiel abschließen.

Beispiel 4.4.17 (Transport mit anschließender Phagozytose) Wir betrachten $\mathcal{R} = (\{r_c\}, \emptyset, \emptyset, \emptyset, \{r_t\}, \{r_p\})$ mit

$$\begin{aligned} r_c &= ((B, 2, \emptyset, \emptyset), (C, 1, \emptyset, \emptyset)) \\ r_t &= ((A, 1, \emptyset, \emptyset), (B, 1, \bar{2}, \emptyset), M, M) \\ r_p &= ((B, 2, \emptyset, \emptyset), (C, 1, \emptyset, \emptyset), M, N, N, N', C(1)) \end{aligned}$$

und das System S mit

$$S = A(1^x) \circ [B(1^x + \bar{2})]_{\mathbb{M}} \circ [C(1)]_{\mathbb{N}}.$$

Die bestehende Verbindung x zwischen A und B erfüllt die Bedingung der Transportregel r_t , Protein $A(1^x)$ kann die Membran M passieren, es resultiert das System

$$S' = [B(1 + 2)]_{\mathbb{M}} \circ [C(1)]_{\mathbb{N}}.$$

Der Transport hat zur Aktivierung der Bindungsstelle 2 in B geführt und ermöglicht die Komplexierung von B und C mittels Reaktionsregel r_c , es entsteht das System

$$S'' = [B(1 + 2^y)]_{\mathbb{M}} \circ [C(1^y)]_{\mathbb{N}},$$

welches schließlich den Ausgangspunkt für die Anwendung der Phagoregel r_p bildet und sich nach deren Anwendung zu

$$S''' = [\diamond]_{\mathbb{N}} \circ [C(1)]_{\mathbb{N}'} \circ [B(1 + 2)]_{\mathbb{M}} \circ [A(1)]_{\mathbb{M}}$$

entwickelt. □

4.4.4 Diskussion

Der bio κ -Kalkül stellt eine bedeutende Erweiterung der Modellierungsmöglichkeiten des κ -Kalküls dar. Die Einführung von Membranen erlaubt die Beschreibung von Proteinen in ihrem natürlichen Kontext und die Betrachtung von Interaktionen, bei denen die Abgrenzung von Reaktionsräumen von Bedeutung ist. Neben ihrer Funktion als Schicht zur Trennung von inneren und äußeren Bereichen besitzt jede Membran im bio κ -Kalkül zusätzlich einen Bereich zur Beschreibung membranständiger Proteine. Diese abgegrenzte Struktur der Syntax empfindet die biologische Membran als Träger zahlreicher Proteine und deren Interaktionen auf natürliche Weise nach. Die Transitionsregeln liefern ein Regelwerk für mögliche Interaktionen sowohl innerhalb von Reaktionsräumen als auch über Reaktionsräume hinweg und sichern ihre biologische Plausibilität ab. Neben ihren passiven Einflüssen durch Abgrenzungen werden

Membranen zusätzlich als aktive Funktionseinheiten in den Kalkül integriert. Dabei stehen neben Wechselwirkungen mit Proteinen in Form von Membranaktivierungen und transmembranem Proteintransport auch die Operationen Phago, Exo und Mate verschiedener Membranen untereinander zur Verfügung.

Bezogen auf Membraninteraktionen lässt der $\text{bio}\kappa$ -Kalkül im Vergleich zum Brane-Kalkül jedoch noch einige Aspekte vermissen. Die fehlende Operation Pino lässt sich aus Phago ableiten, weiterhin können Bud und Drip schließlich durch Kombination der Operationen Phago, Exo und Pino in den Kalkül integriert werden. Auch eine weitere Regel für den transmembranen Transport nach außen ist leicht zu entwickeln. Schwieriger gestaltet sich allerdings die Degeneration und Synthese komplexer Strukturen. Der Replikationsoperator im Brane-Kalkül erlaubt die Erzeugung von Systemen und ist beispielsweise für die Modellierung der Vermehrung von Zellen von Bedeutung, wir erinnern uns an die Plasmazellenbildung im Zusammenhang mit der Immunabwehr gegen das HIV. Darüberhinaus stellt B&R eine wichtige Operation zur Formulierung von nicht unmittelbar mit Membranen in Zusammenhang stehenden Aspekten dar. Die Möglichkeit zur beliebigen Degeneration und Synthese von Molekülen erlaubt Modellierungen auf verschiedenen Abstraktionslevels, als Beispiel sei hier auf die Einbettung viralen Genoms in den Zellkern verwiesen, die wir in unserer Modellierung stark vereinfacht dargestellt haben. Weitere Vorteile liefert die lokale Anwendbarkeit von Replikation und B&R im Brane-Kalkül, denn seine syntaktische Struktur bietet ein hohes Maß an Kontrolle bezüglich der Position, an der Material synthetisiert oder degeneriert werden soll. Eine Umsetzung im $\text{bio}\kappa$ -Kalkül müsste globale Regeln für derartige Interaktionen angeben und könnte lediglich bestimmte Proteinstrukturen oder Membrannamen zur räumlichen Positionierung nutzen. Proteine allein scheinen dafür nicht geeignet, weil deren Wechselwirkungen stark kontextabhängig sind und in der Natur insbesondere die Erzeugung neuer Membransysteme stets durch Membranen erfolgt. Im Zusammenhang mit Membrannamen und deren Umbenennung zur Kontrolle einzelner Mechanismen gilt es bei der Modellierung zahlreiche überlagernde Effekte zu beachten, denn auch die Transitionsregeln MATE, EXO, MEM-A, MREACT-A, TRSP und PHAGO greifen zur Regulierung auf das Konzept der Umbenennung zurück. Desweiteren ist das Konzept verschiedener globaler Regelarten mit voneinander abweichenden Strukturen zur Beschreibung sämtlicher Membranprozesse bei größeren Systemen nur schwer zu überblicken. Während der $\text{bio}\kappa$ -Kalkül also umfassende Mittel zur Modellierung von Proteininteraktionen bietet, benötigen seine Möglichkeiten zur Darstellung von Membraninteraktionen und der Degeneration und Synthese biologischen Materials noch einige Anpassungen.

Kapitel 5

Der Protein-Brane-Kalkül

Mit der Modellierung der HIV-Infektion und ihrer Effekte haben wir die Ausdrucksstärke des Brane-Kalküls demonstriert. Dabei sind wir auf einige Aspekte aufmerksam geworden, die eine Erweiterung der Darstellungsmöglichkeiten sinnvoll erscheinen lassen. Beispielsweise ist die Modellierung eines temporären Andockens zweier Zellen zur Übermittlung eines Signals nur über einen zwischenzeitlichen Systemzustand möglich, der in der Natur in diesem Zusammenhang nicht auftritt. Die Blockierung einzelner Rezeptoren lässt sich hingegen gar nicht formulieren. Weil Proteine die steuernden Elemente sämtlicher mit Membranen in Verbindung stehender Prozesse sind, liegt eine Integration deren Wechselwirkungen in den Kalkül für eine natürlichere Darstellung der angesprochenen Aspekte nahe.

Die Modellierung ausgewählter Proteininteraktionen im Zusammenhang mit der HIV-Infektion hat gezeigt, dass der κ -Kalkül eine geeignete Möglichkeit zur Darstellung membranbezogener Proteininteraktionen darstellt. Seine Erweiterung zum $\text{bio}\kappa$ -Kalkül widmet sich zusätzlich verschiedenen Membranaspekten, lässt jedoch einige für die HIV-Modellierungen notwendige Elemente des Brane-Kalküls vermissen. Die Konstruktion eines vom Brane-Kalkül ausgehenden und Teile des $\text{bio}\kappa$ -Kalküls integrierenden Ansatzes verspricht hingegen die Nutzung der umfassenden Modellierungsmöglichkeiten von Proteininteraktionen innerhalb von Membranstrukturen unter Beibehaltung aller ursprünglichen Modellierungsaspekte des Brane-Kalküls.

In diesem Kapitel werden wir den auf dem Brane-Kalkül basierenden Protein-Brane-Kalkül entwickeln, welcher Proteininteraktionen im Stile des $\text{bio}\kappa$ -Kalküls integriert. Abschließend demonstrieren wir seine im Vergleich zum Brane-Kalkül erweiterte Ausdrucksstärke wiederum am HIV-Beispiel. Während wir alle vom Brane-Kalkül bekannten Strukturen aufgrund notwendiger Anpassungen neu definieren, bauen wir auf einigen in Kapitel 4 eingeführten Definitionen auf.

5.1 Syntax

Zunächst benötigen wir, wie auch bei den anderen Ansätzen, eine Termstruktur im Protein-Brane-Kalkül. Diese baut auf der in Definition 3.1.1 gegebenen Syntax des Brane-Kalküls auf und reichert sie unter Nutzung der in Definition 4.1.4 eingeführten Konfigurationen um eine Notation zur Darstellung von Proteinstrukturen an. Zusätzlich verwendet sie eine spezielle Art von Konfigurationen, die wir im Folgenden angeben.

Definition 5.1.1 (Partielle Konfigurationen) Die Menge der *partiellen Konfigurationen* wird mit \mathcal{K}_p notiert und ist gegeben durch die folgende Grammatik:

$$K_p ::= \diamond \mid A(\varphi_A) \mid K_p \circ K_p.$$

Dabei gilt $A \in \mathcal{N}$ und $\varphi_A \in \mathcal{I}$. Wir nennen $A(\varphi_A)$ ein *partielles Protein*. \square

Im Unterschied zu Definition 4.1.4 sind die hier verwendeten Interfaces nicht zwingend total. Die Definition gestattet somit die Angabe von unvollständig definierten Proteinen, die nicht in jeder Bindungsstelle einen eindeutig zugewiesenen Zustand besitzen.

Definition 5.1.2 (Protein-Brane-Kalkül-Syntax) Die Menge der *Systeme* im Protein-Brane-Kalkül wird mit \mathcal{S}_{pb} notiert und ist gegeben durch die folgende Grammatik:

$$S ::= K \mid S \circ S \mid !S \mid \sigma[K](S).$$

Dabei steht K für beliebige Konfigurationen gemäß der in Definition 4.1.4 gegebenen Grammatik und σ für beliebige *Prozesse* gemäß der Grammatik

$$\sigma ::= 0 \mid \sigma \mid \sigma \mid !\sigma \mid a.\sigma$$

mit beliebigen Aktionen a gemäß den Grammatiken a_{pep} , a_{mbd} und $a_{b\&r}$ mit

$$a_{pep} ::= \Downarrow_{K_p, n} \mid \Downarrow_{K_p, n}^\perp(\sigma[K]) \mid \Uparrow_{K_p, n} \mid \Uparrow_{K_p, n}^\perp \mid \odot_{K_p, K}(\sigma[K]),$$

$$a_{mbd} ::= \mathfrak{m}\mathfrak{t}_{K_p, n} \mid \mathfrak{m}\mathfrak{t}_{K_p, n}^\perp \mid \mathfrak{b}\mathfrak{d}_{K_p, n} \mid \mathfrak{b}\mathfrak{d}_{K_p, n}^\perp(\sigma[K]) \mid \mathfrak{d}\mathfrak{r}_{K_p}(\sigma[K]),$$

$$a_{b\&r} ::= \mathfrak{b}\mathfrak{r} \left\langle \begin{array}{c} K(K) \\ K(K) \end{array} \right\rangle_{K_p}.$$

Weiterhin steht K_p für beliebige partielle Konfigurationen gemäß der in Definition 5.1.1 gegebenen Grammatik. Die Menge der Prozesse wird mit \mathcal{P}_{pb} notiert. \square

Konfigurationen K stehen für eine Ansammlung von Proteinen und stellen bereits ein System dar. Membranen werden im Stile des bio κ -Kalküls um einen zusätzlichen durch eckige Klammern gekennzeichneten Bereich erweitert, welcher membranständige Proteine in Form von Konfigurationen darstellt. Die Struktur von Prozessen bleibt im Vergleich zum Brane-Kalkül unverändert, jedoch erfahren die zugrunde liegenden Aktionen und Ko-Aktionen einige Anpassungen. Ihr Index wird um eine partielle Konfiguration erweitert, welche Belegungen einzelner Bindungsstellen ausgewählter Proteine beschreibt. Mithilfe der später definierten Transitionsrelation kann eine Interaktion schließlich nur dann stattfinden, wenn auf der zugehörigen Membran der Aktion diese Auswahl von Proteinen mit exakt den geforderten Bindungsstellenbelegungen neben möglichen weiteren Proteinen existiert. Desweiteren wird die vom Brane-Kalkül bekannte Zuordnung von Prozessen zu neu entstehenden Membranen bei den Operationen Phago, Pino, Bud und Drip auf die nun vorhandenen membranständigen Proteine angepasst, die entsprechenden Ko-Aktionen bzw. Aktionen tragen hinter dem zuzuordnenden Prozess auch die zuzuordnenden membranständigen Proteine in eckigen Klammern. Die Aktion von Pino erhält darüberhinaus noch eine Konfiguration als Index und

spezifiziert damit einen Proteinkomplex, der durch die zugehörige Interaktion statt des vormals leeren Systems aufgenommen werden kann.

In der Erweiterung sind Moleküle m aus einer Menge \mathcal{M} von Molekülnamen nicht länger vorhanden. Dies stellt jedoch keine Einschränkung dar, wenn wir uns in der Modellierung von der biologischen Bedeutung des Wortes Protein ein Stück entfernen: Moleküle des Brane-Kalküls lassen sich dann als Proteine ohne Bindungsstellen auch im Protein-Brane-Kalkül darstellen. Diese Sichtweise ermöglicht eine wesentlich geringere Anzahl von Transitionsregeln. Sie verlangt lediglich die Anpassung der Operation B&R auf Konfigurationen statt Molekül-Kompositionen und integriert damit bereits den transmembranen Proteintransport in den Kalkül.

Wir behalten die in Kapitel 3 festgelegten Präzedenzregeln der Operatoren bei, mit absteigender Reihenfolge verwenden wir also “.”, “!” , “|” , “ $\sigma[K](P)$ ”, “ \circ ”. Desweiteren schreiben wir für Proteine, die keine Bindungsstellen besitzen, vereinfachend A statt $A()$. Sind (partielle) Konfigurationen an den Indizes leer, verzichten wir auf deren Angabe. Ebenso verzichten wir auf die Angabe von mit eckigen Klammern gekennzeichneten Bereichen, wenn deren Inhalt die leere Konfiguration ist. Wir schreiben also beispielsweise $\Downarrow_n^\perp(\sigma[K])$ statt $\Downarrow_{\diamond,n}^\perp(\sigma[K])$, $\sigma(P)$ statt $\sigma[\diamond](P)$ und $\Downarrow_{K,n}^\perp(\sigma)$ statt $\Downarrow_{K,n}^\perp(\sigma[\diamond])$. Darüberhinaus verwenden wir weiterhin die Kurzschreibweisen des Brane-Kalküls.

5.2 Semantik

Bevor wir die Transitionsrelation betrachten, müssen einige aus Kapitel 4 bekannte Definitionen auf Systeme des Protein-Brane-Kalküls angepasst werden. Wir geben diese im Folgenden unter Beibehaltung der verwendeten Bezeichnungen vollständig an. Zunächst betrachten wir die Modifikation der Funktion occ .

Definition 5.2.1 (occ) Die Funktion $occ : \mathcal{E} \times \mathcal{S}_{pb} \rightarrow \mathbb{N}$ ist für alle $x \in \mathcal{E}$, $A(\varphi) \in \mathcal{P}$, $K \in \mathcal{K}$ und $S, S_1, S_2 \in \mathcal{S}_{pb}$ induktiv definiert als

$$\begin{aligned} occ(x, \diamond) &= 0 \\ occ(x, A(\varphi)) &= \begin{cases} 1 & \text{wenn } \exists i \in \text{def}(\varphi). \varphi(i) = x \\ 0 & \text{sonst} \end{cases} \\ occ(x, S_1 \circ S_2) &= occ(x, S_1) + occ(x, S_2) \\ occ(x, !S) &= occ(x, S) \\ occ(x, \sigma[K](S)) &= occ(x, K) + occ(x, S) \end{aligned} \quad \square$$

Wir stellen fest, dass als Indizes an Aktionen notierte partielle Konfigurationen nicht zu den Vorkommen von Verbindungsnamen eines Systems beitragen. Sie dienen lediglich der Angabe von Proteinen, die zum Stattfinden einer Interaktion auf der entsprechenden Membran vorhanden sein müssen und sind deshalb im System selbst nicht zwingend vorhanden. Selbiges gilt auch für Konfigurationen, die an den Aktionen von Pino und B&R sowie an membranerzeugenden Aktionen mit eckigen Klammern hinter den zuzuordnenden Membranprozessen notiert sind. Entsprechend haben solche Strukturen auch keinen Einfluß auf Funktionen, denen die Definition von occ zugrunde liegt.

Definition 5.2.2 (*en, diff, de*) Wir definieren $en : \mathcal{S}_{pb} \rightarrow 2^{\mathcal{E}}$ für alle $S \in \mathcal{S}_{pb}$ mit

$$en(S) = \{x \mid occ(x, S) \geq 1\}$$

und für alle $S, S', T \in \mathcal{S}_{pb}$ die abkürzende Schreibweise

$$diff(S, S', T) :\Leftrightarrow ((en(S') \setminus en(S)) \cup (en(S) \setminus en(S'))) \cap en(T) = \emptyset.$$

Desweiteren definieren wir die Funktion $de : \mathcal{S}_{pb} \rightarrow 2^{\mathcal{E}}$ für alle $S \in \mathcal{S}_{pb}$ mit

$$de(S) := \{x \mid occ(x, S) = 1\}. \quad \square$$

Unter Verwendung der Funktionen occ und de ergibt sich die Geschlossenheit (closeness) und Wohlgeformtheit eines Systems im Protein-Brane-Kalkül wie folgt.

Definition 5.2.3 (*cl, Wohlgeformtheit von Systemen*) Sei $S \in \mathcal{S}_{pb}$. S heißt *geschlossen*, notiert $cl(S)$, wenn für alle $x \in \mathcal{E}$ $occ(x, S) = 0$ oder $occ(x, S) = 2$ gilt. S heißt *geordnet*, wenn $ord(S)$ mithilfe der folgenden Regeln abgeleitet werden kann:

$$\begin{array}{ll} & ord(\diamond) \\ & ord(A(\varphi)) \\ ord(S) \wedge ord(T) & \Rightarrow ord(S \circ T) \\ ord(S) & \Rightarrow ord(!S) \\ ord(S) \wedge de(S) \subseteq de(K) & \Rightarrow ord(\sigma[K](S)) \end{array}$$

S heißt *wohlgeformt*, wenn es geschlossen und geordnet ist. □

Die Unabhängigkeit von verwendeten Verbindungsnamen innerhalb der später angegebenen Transitionsrelation des Protein-Brane-Kalküls benötigt wie schon im κ -Kalkül die Spezifikation einer Äquivalenzrelation, die wir im Folgenden angeben.

Definition 5.2.4 ($=_{\alpha}$) Sei θ eine Substitution gemäß Definition 4.2.4. Wir definieren die *Anwendung von θ auf Systeme* für alle $A(\varphi) \in \mathcal{P}$, $K \in \mathcal{K}$ und $S_1, S_2 \in \mathcal{S}_{pb}$ induktiv als

$$\begin{array}{ll} \diamond \theta & = \diamond \\ A(\varphi) \theta & = A(\varphi') \quad \text{mit } def(\varphi') = def(\varphi) \text{ und} \\ & \quad \forall x \in def(\varphi). \varphi'(x) = \theta(\varphi(x)) \\ (S_1 \circ S_2) \theta & = (S_1 \theta) \circ (S_2 \theta) \\ (!S) \theta & = !(S \theta) \\ (\sigma[K](S)) \theta & = \sigma[K \theta](S \theta) \end{array}$$

Jeder Substitution θ ordnen wir für jedes $S \in \mathcal{S}_{pb}$ eine Substitution $\theta_S : \mathcal{E} \rightarrow \mathcal{E}$ zu, welche für alle $x \in \mathcal{E}$ definiert ist als

$$\theta_S(x) = \begin{cases} \theta(x) & \text{wenn } x \in en(S), \theta(x) \notin en(S) \text{ und } \forall y \in en(S) \setminus \{x\}. \theta(x) \neq \theta(y) \\ x & \text{sonst} \end{cases}$$

Sei schließlich $=_{\alpha} \subseteq \mathcal{S}_{pb} \times \mathcal{S}_{pb}$ die kleinste Äquivalenzrelation, die alle Paare (S, S') enthält, für die eine Substitution θ existiert mit $S \theta_S = S'$. □

Die Strukturelle Kongruenz orientiert sich an der des Brane-Kalküls in Definition 3.2.1, benötigt aufgrund der erweiterten syntaktischen Struktur jedoch einige Anpassungen.

Definition 5.2.5 (Strukturelle Kongruenz) Sei $\equiv_p \subseteq \mathcal{P}_{pb} \times \mathcal{P}_{pb}$ die kleinste Äquivalenzrelation mit den Eigenschaften (1') bis (11'). Die *Strukturelle Kongruenz* $\equiv \subseteq \mathcal{S}_{pb} \times \mathcal{S}_{pb}$ ist die kleinste Äquivalenzrelation, welche (1) bis (11) erfüllt.

$$\begin{array}{ll}
(1) & S \circ T \equiv T \circ S \\
(2) & S \circ (T \circ U) \equiv (S \circ T) \circ U \\
(3) & S \circ \diamond \equiv S \\
(4) & !\diamond \equiv \diamond \\
(5) & !(S \circ T) \equiv !S \circ !T \\
(6) & !!S \equiv !S \\
(8) & 0\diamond \equiv \diamond \\
(9) & S \equiv T \Rightarrow S \circ U \equiv T \circ U \\
(10) & S \equiv T \Rightarrow !S \equiv !T \\
(11) & S \equiv T \wedge \sigma \equiv_p \tau \wedge K, L \in \mathcal{K} \wedge K \equiv L \Rightarrow \sigma[K](\diamond S) \equiv \tau[L](\diamond T) \quad \square
\end{array}
\qquad
\begin{array}{ll}
(1') & \sigma|\tau \equiv_p \tau|\sigma \\
(2') & \sigma|(\tau|\varrho) \equiv_p (\sigma|\tau)|\varrho \\
(3') & \sigma|0 \equiv_p \sigma \\
(4') & !0 \equiv_p 0 \\
(5') & !(\sigma|\tau) \equiv_p !\sigma|!\tau \\
(6') & !!\sigma \equiv_p !\sigma \\
(7') & !\sigma \equiv_p \sigma|!\sigma \\
(9') & \sigma \equiv_p \tau \Rightarrow \sigma|\varrho \equiv_p \tau|\varrho \\
(10') & \sigma \equiv_p \tau \Rightarrow !\sigma \equiv_p !\tau \\
(11') & \sigma \equiv_p \tau \Rightarrow a.\sigma \equiv_p a.\tau
\end{array}$$

Regel (7) der Strukturellen Kongruenz des Brane-Kalküls existiert im Protein-Brane-Kalkül nicht. Stattdessen verlagert sich die Funktionalität des Replikationsoperators in die später angegebene Transitionsrelation und ermöglicht dadurch die Kontrolle über Verbindungsnamen von neu entstehenden Systemen. Regel (8) wird auf die erweiterte syntaktische Struktur im Protein-Brane-Kalkül angepasst. Ähnlich geschieht dies auch in Regel (11) zur Gewährleistung der beliebigen Anordnung von Proteinen innerhalb membranständiger Konfigurationen. Wir stellen fest, dass Konfigurationen keine eigene Kongruenzrelation benötigen, weil sie eine Teilmenge der Systeme darstellen.

Proteine sind auch im Protein-Brane-Kalkül dazu in der Lage, mithilfe zusätzlich spezifizierter Reaktionsregeln miteinander zu interagieren. Anders als im $\text{bio}\kappa$ -Kalkül beschränken wir uns jedoch auf zwei Regelmengen \mathfrak{C} und \mathfrak{D} von Komplexierungs- und Dekomplexierungsregeln, deren in Definition 4.2.7 gegebene Struktur wir weiterhin verwenden. Auch den Ausdruck LR_r für die mit Indizes versehene Menge aus linker und rechter Regelseite einer beliebigen Reaktionsregel $r \in \mathfrak{C} \cup \mathfrak{D}$ und die in Definition 4.4.15 angegebenen Notationen \leq , \setminus und \cap über Konfigurationen behalten wir bei. Aufbauend auf \leq benötigen wir zusätzlich eine weitere Relation.

Definition 5.2.6 (\leq_p) Wir definieren $\leq_p \subseteq \mathcal{K}_p \times \mathcal{K}$ als

$$\begin{aligned}
\leq_p := & \{ (A_1(\varphi_1) \circ \dots \circ A_n(\varphi_n), K) \mid \\
& \exists \varphi'_1, \dots, \varphi'_n. A(\varphi_1 + \varphi'_1) \circ \dots \circ A(\varphi_n + \varphi'_n) \leq K \}. \quad \square
\end{aligned}$$

Mithilfe des Ausdruckes $K_p \leq_p K$ können wir die Bedingung formulieren, dass die in K_p angegebenen partiellen Proteine unter Erfüllung der angegebenen Bindungsstellenzustände in einer vollständig definierten Form in K vorhanden sein müssen. Dies ermöglicht die Überprüfung der als Indizes an den Aktionen bzw. Ko-Aktionen geforderten Kontexte zum Stattfinden von Interaktionen. Nachdem wir nun alle notwendigen Bestandteile betrachtet haben, können wir die Transitionsrelation des Protein-Brane-Kalküls definieren.

Definition 5.2.7 (Transitionsrelation $\longrightarrow_{\mathcal{R}}$) Sei $\mathcal{R} = (\mathfrak{C}, \mathfrak{D})$ mit zwei Mengen \mathfrak{C} und \mathfrak{D} von Reaktionsregeln. Es gelte $\mathcal{L} = \{\tau\} \cup \{r_n^x \mid r \in \mathfrak{C} \cup \mathfrak{D}, x \in \mathcal{E}, n \in \{1, 2\}\}$ und $\mu \in \mathcal{L}$ beliebig. $\longrightarrow_{\mathcal{R}} \subseteq \mathcal{S}_{pb} \times \mathcal{L} \times \mathcal{S}_{pb}$ ist die kleinste Relation, die folgende Transitionsregeln erfüllt (notiert $K \xrightarrow{\mu} K'$ statt $(K, \mu, K') \in \longrightarrow_{\mathcal{R}}$):

$$\text{PHAGO} \frac{K_p \leq_p K \quad L_p \leq_p L \quad L' \leq L \quad C_1 \quad C_2}{\Downarrow_{K_p, n} \cdot \sigma | \sigma_0 [K] (\Downarrow P) \circ \Downarrow_{L_p, n}^\perp (\varrho [L']) \cdot \tau | \tau_0 [L] (\Downarrow Q) \xrightarrow{\tau} RHS}$$

$$\text{mit } C_1 = de(K) \subseteq de(P \circ L'), \quad C_2 = de(L') \subseteq de((L \setminus L') \circ Q \circ K)$$

$$\text{und } RHS = \tau | \tau_0 [L \setminus L'] (\varrho [L'] (\sigma | \sigma_0 [K] (\Downarrow P))) \circ Q$$

$$\text{EXO} \frac{K_p \leq_p K \quad L_p \leq_p L}{\Uparrow_{K_p, n}^\perp \cdot \tau | \tau_0 [K] (\Uparrow_{L_p, n} \cdot \sigma | \sigma_0 [L] (\Downarrow P) \circ Q) \xrightarrow{\tau} P \circ \sigma | \sigma_0 \tau | \tau_0 [K \circ L] (\Downarrow Q)}$$

$$\text{PINO} \frac{L_p \leq_p L \quad L' \leq L \quad de(P) \subseteq de(L') \quad de(L') \subseteq de((L \setminus L') \circ Q \circ P)}{P \circ \odot_{L_p, P} (\varrho [L']) \cdot \sigma | \sigma_0 [L] (\Downarrow Q) \xrightarrow{\tau} \sigma | \sigma_0 [L \setminus L'] (\varrho [L'] (\Downarrow P) \circ Q)}$$

$$\text{MATE} \frac{K_p \leq_p K \quad L_p \leq_p L}{\text{mt}_{K_p, n} \cdot \sigma | \sigma_0 [K] (\Downarrow P) \circ \text{mt}_{L_p, n}^\perp \cdot \tau | \tau_0 [L] (\Downarrow Q) \xrightarrow{\tau} \sigma | \sigma_0 \tau | \tau_0 [K \circ L] (\Downarrow P \circ Q)}$$

$$\text{BUD} \frac{K_p \leq_p K \quad L_p \leq_p L \quad K' \leq K \quad C_1 \quad C_2}{\text{bd}_{K_p, n}^\perp (\varrho [K']) \cdot \tau | \tau_0 [K] (\text{bd}_{L_p, n} \cdot \sigma | \sigma_0 [L] (\Downarrow P) \circ Q) \xrightarrow{\tau} RHS}$$

$$\text{mit } C_1 = de(Q) \cap de(L \circ K') = \emptyset, \quad C_2 = de(L) \cap de(K \setminus K') = \emptyset$$

$$\text{und } RHS = \varrho [K'] (\sigma | \sigma_0 [L] (\Downarrow P)) \circ \tau | \tau_0 [K \setminus K'] (\Downarrow Q)$$

$$\text{DRIP} \frac{K_p \leq_p K \quad K' \leq K \quad de(K') \cap de(Q) = \emptyset}{\text{dr}_{K_p} (\varrho [K']) \cdot \sigma | \sigma_0 [K] (\Downarrow Q) \xrightarrow{\tau} \varrho [K'] (\Downarrow \diamond) \circ \sigma | \sigma_0 [K \setminus K'] (\Downarrow Q)}$$

$$\text{B\&R} \frac{K_p \leq_p K \quad C_1 \quad C_2 \quad cl(Deg_a) \quad cl(Deg_i) \quad cl(Syn_a) \quad cl(Syn_i) \quad C_3 \quad C_4}{L \circ \text{br} \left\langle \frac{L \langle M \rangle}{L' \langle M' \rangle} \right\rangle_{K_p} \cdot \sigma | \sigma_0 [K] \langle M \circ P \rangle \xrightarrow{\tau} L' \circ \sigma | \sigma_0 [K] \langle M' \circ P \rangle}$$

$$\text{mit } C_1 = de(Trsp_{ai}) \subseteq de(K), \quad C_2 = de(Trsp_{ia}) \subseteq de(K),$$

$$Deg_a = (L \setminus L') \setminus Trsp_{ai}, \quad Deg_i = (M \setminus M') \setminus Trsp_{ia},$$

$$Syn_a = (L' \setminus L) \setminus Trsp_{ia}, \quad Syn_i = (M' \setminus M) \setminus Trsp_{ai},$$

$$Trsp_{ai} = (L \setminus L') \cap (M' \setminus M), \quad Trsp_{ia} = (M \setminus M') \cap (L' \setminus L)$$

$$C_3 = en(Syn_a) \cap en((L' \setminus Syn_a) \circ K \circ M' \circ P) = \emptyset,$$

$$C_4 = en(Syn_i) \cap en(L' \circ K \circ (M' \setminus Syn_i) \circ P) = \emptyset$$

$$\text{REP} \frac{cl(S) \quad S =_\alpha S' \quad en(S) \cap en(S') = \emptyset}{!S \xrightarrow{\tau} S' \circ !S}$$

$$\text{COM} \frac{r \in \mathfrak{C} \quad (A, i, \varphi, \varphi')_n \in LR_r \quad x \in \mathcal{E} \setminus en(A(i + \varphi + \varphi' + \varphi_0))}{A(i + \varphi + \varphi' + \varphi_0) \xrightarrow{r_n^x} A(i^x + \bar{\varphi} + \varphi' + \varphi_0)}$$

$$\text{DEC} \frac{r \in \mathfrak{D} \quad (A, i, \varphi, \varphi')_n \in LR_r}{A(i^x + \varphi + \varphi' + \varphi_0) \xrightarrow{r_n^x} A(i + \bar{\varphi} + \varphi' + \varphi_0)}$$

$$\text{REACT} \frac{P \xrightarrow{r_1^x} P' \quad Q \xrightarrow{r_2^x} Q'}{P \circ Q \xrightarrow{\tau} P' \circ Q'} \quad \text{CONV} \frac{P' =_\alpha P \quad P \xrightarrow{\tau} Q \quad Q =_\alpha Q'}{P' \xrightarrow{\tau} Q'}$$

$$\text{PAR} \frac{P \xrightarrow{\mu} P' \quad diff(P, P', Q)}{P \circ Q \xrightarrow{\mu} P' \circ Q} \quad \text{STRUCT} \frac{P' \equiv P \quad P \xrightarrow{\mu} Q \quad Q \equiv Q'}{P' \xrightarrow{\mu} Q'}$$

$$\text{MEM} \frac{K \xrightarrow{\mu} K' \quad diff(K, K', S)}{\sigma[K] \langle S \rangle \xrightarrow{\mu} \sigma[K'] \langle S \rangle} \quad \text{BRANE} \frac{S \xrightarrow{\tau} S' \quad diff(S, S', K)}{\sigma[K] \langle S \rangle \xrightarrow{\tau} \sigma[K] \langle S' \rangle}$$

$$\text{MREACT} \frac{K \xrightarrow{r_m^x} K' \quad S \xrightarrow{r_n^x} S' \quad m \neq n}{\sigma[K] \langle S \rangle \xrightarrow{\tau} \sigma[K'] \langle S' \rangle}$$

Wir nennen $\longrightarrow_{\mathcal{R}}$ die *Transitionsrelation* des Protein-Brane-Kalküls und ihre Elemente *Ableitungsschritte*. Eine endliche Folge $S_1 \xrightarrow{\mu_1} S_2 \xrightarrow{\mu_2} \dots \xrightarrow{\mu_{l-1}} S_l$ bezeichnen wir als *Transitionsableitung*. \square

PHAGO: Die Bedingung $K_p \leq_p K$ fordert, dass alle an der Aktion von Phago mittels K_p angegebenen partiellen Proteine in einer totalen Form auf der äußeren Membran des zugehörigen Systems vorkommen, innerhalb von K müssen dabei alle in K_p formulierten Bindungsstellenbelegungen erfüllt sein. Bedingung $L_p \leq_p L$ fordert dies analog für die zugehörige Ko-Aktion. Jede Transitionsregel, die das Konsumieren von Aktionen zur Folge hat, ist mit derartigen Bedingungen versehen.

L' bezeichnet den Teil der Proteine, die im Zuge der Interaktion Phago von der verschlingenden Membran abgespalten und der neu entstehenden zugeordnet werden. Es muss mittels $L' \leq L$ überprüft werden, ob die abzuspaltenden Proteine auf der entsprechenden Membran und damit innerhalb von L vorhanden sind. Die Bedingungen C_1 und C_2 sichern die Wahrung der Wohlgeformtheit des Gesamtsystems. C_1 fordert, dass membranständige Proteine K des zu verschlingenden Systems lediglich Verbindungen zu dessen Systeminneren P und zu Proteinen L' besitzen, die sich zur neuen Membran hin abspalten. Jede andere Verbindung von K zu Proteinen hätte nach der Interaktion Phago das Weiterbestehen dieser Verbindung über Membrangrenzen hinweg zur Folge und ist biologisch nicht möglich. C_2 stellt eine ähnliche Forderung: weil Proteine in L' während der Interaktion nach innen gelangen, dürfen sie keine Verbindung zu Proteinen außerhalb des phagozytierenden Systems besitzen. Sind alle Bedingungen erfüllt, läuft die Interaktion Phago unter zusätzlicher Beachtung der Proteinabspaltung L' von L auf die vom Brane-Kalkül gewohnte Weise ab.

EXO: Der Ablauf ist analog zu EXO im Brane-Kalkül, zusätzlich werden durch $K_p \leq_p K$ und $L_p \leq_p L$ einige Bindungsstellen überprüft und die membranständigen Proteine der miteinander verschmelzenden Membranen zusammengelegt.

PINO: Pino ist nun dazu in der Lage, ganze Proteinkomplexe P zu verschlingen. Auch hier erfolgt mittels Bedingung $L' \leq L$ zunächst die Überprüfung, ob die der abzuspaltenden Membran zugeordneten Proteine L' vollständig auf der ursprünglichen Membran, also innerhalb der Konfiguration L , vorhanden sind. Mithilfe der Bedingung $de(P) \subseteq de(L')$ wird sichergestellt, dass der zu verschlingende Komplex P lediglich mit Proteinen der neu entstehenden Membran verbunden ist. Die Bedingung $de(L') \subseteq de((L \setminus L') \circ Q \circ P)$ fordert, dass die Proteine L' lediglich mit membranständigen Proteinen der alten Membran $L \setminus L'$, Proteinen im Systeminneren Q und mit solchen innerhalb von P verbunden sein können. Jede andere Verbindung verletzt auch hier die Wohlgeformtheit nach Ablauf der Interaktion. Sind alle Bedingungen erfüllt, kann die Konfiguration P in das Innere des Membransystems aufgenommen werden. Auch der Basisfall von Pino, welcher das leere System verschlingt, ist weiterhin modellierbar mittels $P = \diamond$ und der Transitionsregel STRUCT. Damit stellt das Verschlingen von Konfigurationen P im Vergleich zum Brane-Kalkül eine echte Erweiterung dar.

MATE: Ähnlich zu EXO werden die membranständigen Proteine der miteinander verschmelzenden Membranen zusammengelegt, der weitere Ablauf ist analog zu MATE im Brane-Kalkül.

BUD: Im Zuge der Abspaltung trennen sich die Inhalte Q und $\sigma|\sigma_0[L](P)$ des betrach-

teten Gesamtsystems voneinander. C_1 fordert deshalb, dass von Q keine Verbindungen zu L und K' vorhanden sind, denn diese bestünden nach der Interaktion durch Membranen hindurch. Bedingung C_2 verhindert Verbindungen seitens der membranständigen Proteine der inneren Membran zu den nach der Abspaltung nicht mehr präsenten membranständigen Proteinen $K \setminus K'$ der äußeren Membran.

DRIP: Die Proteine K' sind nach der Abspaltung durch eine Membran von Q getrennt, sie dürfen deshalb auch vorher nicht mit Q verbunden sein, dies stellt die Bedingung $de(K') \cap de(Q) = \emptyset$ sicher.

B&R: Mit $Trsp_{ai}$ bezeichnen wir alle Proteine, die durch Anwendung von B&R auf der Außenseite der Membran verschwinden ($L \setminus L'$) und auf der Innenseite entstehen ($M' \setminus M$), also insgesamt betrachtet von der Außenseite der Membran zur Innenseite transportiert werden. Analog verwenden wir $Trsp_{ia}$ für den Transport von innen nach außen. Solche Proteine dürfen zur Wahrung der Wohlgeformtheit lediglich Verbindungen zu den membranständigen Proteinen K der transportierenden Membran besitzen, die Bedingungen C_1 und C_2 sichern dies ab.

B&R ist sehr allgemein gehalten und gestattet auch das beliebige Synthetisieren und Degenerieren von Proteinkomplexen, allerdings sind auch hier einige Einschränkungen zur Wahrung der Wohlgeformtheit notwendig. Jeder nach der Anwendung von B&R degenerierte (Deg_a, Deg_i) oder synthetisierte (Syn_a, Syn_i) Proteinkomplex an jeder Seite der Membran (a für außen, i für innen) darf keine Verbindungen über sich selbst hinaus besitzen. Dies wird durch die Bedingungen $cl(Deg_a), cl(Deg_i), cl(Syn_a)$ und $cl(Syn_i)$ gewährleistet. Weiterhin fordert Bedingung C_3 , dass synthetisierte Proteinkomplexe an der Außenseite der Membran keine Verbindungsnamen verwenden, die auch in den übrigen Proteinkomplexen vorhanden sind. Die Bedingung C_4 fordert dies analog für synthetisierte Proteinkomplexe an der Innenseite.

B&R stellt eine echte Erweiterung der entsprechenden Reduktionsregel im Brane-Kalkül dar, denn das beliebige Entstehen und Verschwinden von Molekülen (hier als Proteine ohne Bindungsstellen dargestellt) ist nach wie vor möglich: betrachten wir Konfigurationen, in denen keine Verbindungsnamen vorkommen, sind sämtliche Bedingungen von vornherein erfüllt und gestatten damit die uneingeschränkte Anwendung von B&R.

REP: Diese Transitionsregel ersetzt die Regel (7) der Strukturellen Kongruenz des Brane-Kalküls (Definition 3.2.1). Die Replikation eines Systems S ist nur dann möglich, wenn es keine aus ihm herausragenden Verbindungen besitzt, dies kann durch $cl(S)$ gefordert werden. Darüberhinaus darf ein neu entstehendes System S' nicht die gleichen Verbindungsnamen verwenden wie S , es muss also zuvor durch $S =_{\alpha} S'$ eine Substitution stattfinden. Bedingung $en(S) \cap en(S') = \emptyset$ gewährleistet, dass dabei alle Namen umbenannt werden.

COM, ..., MREACT: Diese Transitionsregeln sind in ihrer Funktionsweise analog zu den in Kapitel 4 vorgestellten Transitionsregeln mit gleichem Namen.

Die im Protein-Brane-Kalkül eingeführten Indizes an Aktionen und Ko-Aktionen zur Überprüfung von Protein-Konstellationen vor dem Ablauf von Interaktionen bie-

ten durch die Verwendung partieller Konfigurationen die Möglichkeit zur Angabe unvollständig definierter Bindungsstellenbelegungen. Diese im Vergleich zu Konfigurationen größere Freiheit bei der Modellierung ist notwendig. Einige Membraninteraktionen hängen möglicherweise lediglich von einzelnen Bindungsstellen bestimmter Proteine ab, bei einer Modellierung mit Konfigurationen müsste ihnen jeweils eine Aktion für jede Kombination vollständiger Belegungen zugeordnet werden, die zur Interaktion führt. In der Anwendung würde dann jedoch nur eine dieser Aktionen konsumiert. Alle nicht verwendeten blieben zurück, obwohl die Interaktion möglicherweise nicht wiederholt stattfinden darf. Im Folgenden demonstrieren wir einige weitere Aspekte der Transitionsrelation am Beispiel von B&R.

Beispiel 5.2.8 (Anwendung von Transitionsregeln) Wir betrachten das System

$$S = D(1^z) \circ \text{br} \left\langle \begin{array}{c} D(1^z)(\langle E \rangle) \\ F(1^a) \circ G(1^a)(\langle D(1^z) \rangle) \end{array} \right\rangle_{A(1^x)} [A(1^y) \circ C(1^z)](\langle E \circ B(1^y) \rangle).$$

Für einen Ableitungsschritt gemäß Transitionsregel B&R muss zunächst $K_p \leq_p K$ gelten, bezogen auf S also $A(1^x) \leq_p A(1^y) \circ C(1^z)$. Dies ist nicht der Fall. Unter Verwendung der Substitution θ mit $\theta(y) = x$ und $\theta(z) = z$ für alle $z \neq y$ gilt $S =_\alpha S_{y \rightarrow x}$ mit

$$S_{y \rightarrow x} = D(1^z) \circ \text{br} \left\langle \begin{array}{c} D(1^z)(\langle E \rangle) \\ F(1^a) \circ G(1^a)(\langle D(1^z) \rangle) \end{array} \right\rangle_{A(1^x)} [A(1^x) \circ C(1^z)](\langle E \circ B(1^x) \rangle),$$

$K_p \leq_p K$ ist in der Instanz $A(1^x) \leq_p A(1^x) \circ C(1^z)$ erfüllt. Weiterhin gilt $\text{Trsp}_{ai} = D(1^z)$. Mit C_1 in B&R wird bezogen auf $S_{y \rightarrow x}$ gefordert, dass $de(D(1^z)) \subseteq de(A(1^x) \circ C(1^z))$. Diese Bedingung ist ebenfalls erfüllt. Da $\text{Trsp}_{ia} = \diamond$, gilt auch C_2 . Mit $\text{Deg}_a = \text{Syn}_i = \diamond$ folgt $cl(\text{Deg}_a)$ und $cl(\text{Syn}_i)$, wegen $\text{Deg}_i = E$ und $\text{Syn}_a = F(1^a) \circ G(1^a)$ gilt auch $cl(\text{Deg}_i)$ und $cl(\text{Syn}_a)$. Bedingung C_3 konkretisiert sich zu $en(F(1^a) \circ G(1^a)) \cap en(\diamond \circ (A(1^x) \circ C(1^z))) \circ D(1^z) \circ B(1^x) = \emptyset$ und gilt ebenfalls. Mit $\text{Syn}_i = \diamond$ folgt auch C_4 . Damit sind alle für die Operation B&R notwendigen Bedingungen erfüllt, $S_{y \rightarrow x}$ kann sich entwickeln zu

$$S'_{y \rightarrow x} = F(1^a) \circ G(1^a) \circ [A(1^x) \circ C(1^z)](\langle D(1^z) \circ B(1^x) \rangle).$$

Unter Verwendung der Substitution θ' mit $\theta'(x) = y$ und $\theta'(z) = z$ für alle $z \neq x$ gilt $S'_{y \rightarrow x} =_\alpha S'$ mit

$$S' = F(1^a) \circ G(1^a) \circ [A(1^y) \circ C(1^z)](\langle D(1^z) \circ B(1^y) \rangle).$$

Mit Transitionsregel CONV kann schließlich von $S_{y \rightarrow x} \xrightarrow{\tau} S'_{y \rightarrow x}$ auf $S \xrightarrow{\tau} S'$ geschlossen werden. Die Prüfung auf geforderte Proteine hängt also nicht von den in den Indizes verwendeten Verbindungsnamen ab, paarweise verwendete Verbindungsnamen werden jedoch stets auf paarweises Auftreten geprüft. \square

5.3 Modifikationen des Kalküls

Wir diskutieren an dieser Stelle einige Aspekte, die sich gut in den Protein-Brane-Kalkül integrieren lassen und somit Vorschläge zur Anpassung oder dem weiteren Ausbau seiner Modellierungsmöglichkeiten geben.

- Neben der Degeneration von Proteinkomplexen kann auch die Modellierung der *Degeneration von Membransystemen* von Bedeutung sein. Beispielsweise leiten HIV-spezifische freie Antikörper IgG nach Andocken an durch freies gp120 besetzte Zellen deren gezielten Zelltod ein. Eine mögliche Umsetzung ist die Einführung einer weiteren Aktion dg_{K_p} zusammen mit einer Transitionsregel, welche den Übergang $\text{dg}_{K_p}.\sigma|\sigma_0[K](P) \xrightarrow{\tau} \diamond$ realisiert. Dabei dürfen Proteine in K nur Verbindungen untereinander und zu P besitzen.
- Der Protein-Brane-Kalkül ist nicht dazu in der Lage, die Degeneration und Synthese von membranständigen Proteinen, deren Ablösen von einer Membran sowie das Anlagern freier Proteine an eine Membran darzustellen. Integrieren lässt sich dies durch die *Erweiterung von B&R* zu $\text{br} \left\langle \frac{L[K](M)}{L'[K'](M')} \right\rangle$. Notwendig hierfür ist die genaue Überprüfung der Verbindungen der sechs angegebenen Konfigurationen sowohl untereinander als auch mit ihrer Umgebung, um die Wahrung der Wohlgeformtheit nach Ablauf der Interaktion zu gewährleisten.
- Die Replikation eines Systems S ist durch *Modifikation der Reaktionsregel REP* auch ohne die Forderung $\text{cl}(S)$ realisierbar. Ausgehend von $!S$ kann die Erzeugung von Systemen S' gestattet werden, die sich über die Verwendung von nicht in S vorkommenden Verbindungsnamen hinaus dadurch von S unterscheiden, dass alle nur einmal in den Proteinen von S auftretenden Verbindungsnamen durch aktive oder inaktive Belegungen der entsprechenden Bindungsstellen ersetzt sind.
- Eine *Projektive Variante* des Protein-Brane-Kalküls zur Integration gerichteter Operationen im Stile des Projektiven Brane-Kalküls ([25]) ist denkbar. Damit zusammenhängend scheint die Unterscheidung zwischen membranständigen Proteinen der Außen- und Innenseiten von Membranen sowie transmembranen Proteinen sinnvoll, welche einige Anpassungen der Transitionsrelation benötigt.

5.4 Anwendungsbeispiel: HIV mit Proteininteraktionen

Mit den folgenden Betrachtungen führen wir die in den Abschnitten 3.5.2 und 4.3.1 vorgestellten Aspekte der HIV-Infektion zusammen und stellen damit das Zusammenwirken von Membran- und Proteininteraktionen in den Mittelpunkt. Darüberhinaus demonstrieren wir die erweiterte Ausdruckstärke des Protein-Brane-Kalküls anhand der Modellierung temporärer Andockprozesse und blockierter Rezeptoren, deren Umsetzung im Brane-Kalkül nur schwer bzw. gar nicht möglich ist. Abschließend diskutieren wir die erzielten Erweiterungen der Modellierbarkeit sowohl im Vergleich zum Brane- als auch zum $\text{bio}\kappa$ -Kalkül und zeigen Möglichkeiten zur weiteren Arbeit auf.

5.4.1 Modellierung

Wir passen zunächst die Komponenten der Modellierungen aus Abschnitt 3.5.2 an den Protein-Brane-Kalkül an und erläutern anschließend einige Transitionsableitungen.

HIV-Partikel

Die Modellierung eines HIV-Partikels ist im Abschnitt 3.5.2 zunächst mit dem System **Virus** erfolgt. An späterer Stelle haben wir **Virus'** eingeführt, um zusätzlich das Andocken freier IgG-Antikörper zu ermöglichen. Die folgenden Betrachtungen konzentrieren sich nun zusätzlich auch auf membranständige Proteine. Wir geben die vollständige Modellierung eines HIV-Partikels im Protein-Brane-Kalkül und gehen anschließend auf die einzelnen Bestandteile ein. Die Namen einzelner Komponenten behalten wir bei, auch wenn sich ihr Inhalt ändert.

$$\begin{aligned}
\mathbf{Virus} &:= m_v[PK_v](\mathbf{Cap}_v) \\
m_v &:= !\mathfrak{m}\mathfrak{t}_{PK_v^a} | !\mathfrak{b}\mathfrak{r} \left\langle \begin{array}{c} \diamond(\diamond) \\ GP_{120(1+2+\bar{3}+4)}(\diamond) \end{array} \right\rangle \\
PK_v &:= GP_{120}(1^x + 2 + \bar{3} + 4) \circ GP_{41}(1^x + 2 + \bar{3}) \\
PK_v^a &:= GP_{120}(1^x + 2^y + 4) \circ GP_{41}(1^x + 3) \\
\mathbf{Cap}_v &:= m_c(\mathbf{RNA}_v) \\
m_c &:= \mathfrak{b}\mathfrak{r} \left\langle \begin{array}{c} TP(\mathbf{RNA}_v) \\ TP \circ \mathbf{RNA}_v(\diamond) \end{array} \right\rangle
\end{aligned}$$

Die Virusmembran ist mit einer Konfiguration PK_v versehen und modelliert die membranständigen Proteine gp120 und gp41. Diese sind über ihre jeweilige Bindungsstelle 1 miteinander zu einem Komplex verbunden. Die für die Fusion mit einer Helfer-T-Zellmembran verantwortliche Aktion $\mathfrak{m}\mathfrak{t}$ ist nun mit der Konfiguration PK_v^a versehen und ermöglicht $\mathfrak{M}\mathfrak{a}\mathfrak{t}\mathfrak{e}$ erst, wenn die membranständigen Proteine gp120 und gp41 die in ihr angegebenen Zustände der Bindungsstellen besitzen. Es gilt $PK_v^a \not\leq_p PK_v$, im initialen Zustand kann $\mathfrak{M}\mathfrak{a}\mathfrak{t}\mathfrak{e}$ also nicht zur Anwendung kommen. Der Prozess $!\mathfrak{b}\mathfrak{r} \left\langle \begin{array}{c} \diamond(\diamond) \\ GP_{120(1+2+\bar{3}+4)}(\diamond) \end{array} \right\rangle$ auf der Virusmembran modelliert die Eigenschaft des Virus, freie gp120-Partikel zu produzieren und an seine Umgebung abzusondern. Das Virusinnere \mathbf{Cap}_v hat sich im Vergleich zur ursprünglichen Definition nicht verändert.

Helfer-T-Zellen

Auch die Modellierung von Helfer-T-Zellen benötigt einige Anpassungen. Im Unterschied zu den Ausführungen in Abschnitt 3.5.2 verzichten wir hier auf die Umsetzung der Aktivierung von Helfer-T-Zellen durch Makrophagen, weil diese ähnlich zur später dargestellten Aktivierung einer B-Zelle abläuft. Wir betrachten deshalb lediglich solche Helfer-T-Zellen, die bereits in aktivierter Form vorliegen. Trotz einiger Übereinstimmungen mit der ursprünglichen Modellierung geben wir das System **TZelle** zunächst erneut vollständig an und betrachten anschließend die einzelnen Komponenten genauer.

$$\begin{aligned}
\mathbf{TZelle} &:= m_t[CD_4(1) \circ \mathit{Frg}_v(1)](TP \circ \mathbf{Nuc}_t) \\
m_t &:= !\mathfrak{m}\mathfrak{t}_{CD_4(1^y)} | !\uparrow^\perp \\
\mathbf{Nuc}_t &:= \mathfrak{b}\mathfrak{r} \left\langle \begin{array}{c} \mathbf{RNA}_v(\mathbf{DNA}) \\ \diamond(\mathbf{DNA}_v) \end{array} \right\rangle . \mathit{protSyn}_v(\mathbf{DNA}) \\
\mathit{protSyn}_v &:= !(\mathfrak{d}\mathfrak{r}(\mathit{memProt}_v[PK_v]) | \mathfrak{d}\mathfrak{r}(\mathit{capProt}_v)) \\
\mathit{memProt}_v &:= \uparrow . \mathfrak{b}\mathfrak{d}^\perp(m_v[PK_v])
\end{aligned}$$

$$\text{capProt}_v := \text{bd.lbr} \left\langle \begin{array}{c} \diamond(\diamond) \\ \diamond(\text{RNA}_v) \end{array} \right\rangle .\text{m}_c$$

Auf der Zellmembran befindet sich das membranständige Glycoprotein CD4. Darüberhinaus modelliert $\text{Frg}_v(1)$ ein virales Fragment, welches mithilfe eines (nicht explizit modellierten) T-Zell-Rezeptors aufgrund vorheriger Aktivierung durch eine Makrophage an der Membranoberfläche exponiert wird. Die Ko-Aktion mt^\perp liefert das Gegenstück zu mt für die Fusion mit der Virusmembran und wird um $\text{CD}_4(1^y)$ erweitert, damit ist auch die Helfer-T-Zelle in ihrem initialen Zustand nicht zur Fusion in der Lage. Weitere Neuerungen bestehen innerhalb von protSyn_v und memProt_v . Die nach der Einlagerung des Virusgenoms in den Zellkern entstehenden Vesikel $\text{memProt}_v[\text{PK}_v](\emptyset)$ besitzen nun auch den viralen Proteinkomplex PK_v auf ihrer Membran, welcher nach erfolgter Exozytose schließlich auf die Helfer-T-Zellmembran gelangt. Innerhalb von memProt_v wird angegeben, dass PK_v im Zuge des Abspaltens des Virus durch Bud auf die Virusmembran übergeht.

B-Zellen und Plasmazellen

Einige B-Zellen besitzen HIV-spezifische Antikörper IgM auf ihrer Membran und sind dadurch dazu in der Lage, an das mittels T-Zell-Rezeptor exponierte virale Fragment von aktivierten Helfer-T-Zellen für kurze Zeit zu binden, um aktiviert zu werden. Während wir die Möglichkeit zum temporären Andocken im Brane-Kalkül mit einem Prozess $!\downarrow_d^\perp$ (\uparrow_d). \uparrow_d^\perp modelliert haben, können wir im Protein-Brane-Kalkül den Antikörper IgM explizit auf der äußeren Membran der B-Zelle verankern. Das System BZelle hat dann folgende Struktur.

$$\begin{aligned} \text{BZelle} &:= \text{m}_b[\text{IgM}(1 + \bar{2})](\text{PSyn}_b) \\ \text{m}_b &:= !\uparrow_{\text{IgM}(1+2)}^\perp \\ \text{PSyn}_b &:= !(\uparrow(\text{PZelle})) \end{aligned}$$

Der Mechanismus zur Bildung von Plasmazellen ist unverändert, die mit ihm in Verbindung stehende Aktion $\uparrow_{\text{IgM}(1+2)}^\perp$ auf der P-Zellmembran verlangt jedoch zuvor die sichtbaren und freien Bindungsstellen 1 und 2 im IgM, die Plasmazellenproduktion kann somit wie schon in der Brane-Kalkül-Modellierung erst nach Aktivierung der B-Zelle stattfinden.

Plasmazellen sind dazu in der Lage, freie Antikörper IgG abzusondern. Auch diese werden nun nicht mehr durch an Membranen gebundene Prozesse modelliert, sondern direkt durch Proteine. Das System PZelle kann dann wie folgt dargestellt werden.

$$\text{PZelle} := !\text{br} \left\langle \begin{array}{c} \diamond(\diamond) \\ \text{IgG}(1)(\diamond) \end{array} \right\rangle (\emptyset)$$

Wir haben nun alle für unsere folgenden Betrachtungen benötigten Systeme eingeführt und können deren Zusammenspiel genauer betrachten.

Proteingesteuerte Fusion von Virus- und Helfer-T-Zellmembran

Zur Modellierung der proteingesteuerten Fusion von Virus und Helfer-T-Zelle benötigen wir zusätzliche Informationen über mögliche Proteininteraktionen. Diese wurden bereits

im Abschnitt 4.3.1 in Form der Komplexierungsregeln r_1 und r_2 spezifiziert, die wir im Folgenden nochmals angeben.

$$\begin{aligned} r_1 &= ((GP_{120}, 2, \bar{3}, 4), (CD_4, 1, \emptyset, \emptyset)) \\ r_2 &= ((GP_{120}, 3, \emptyset, 1^x), (GP_{41}, 2, \bar{3}, 1^x)) \end{aligned}$$

Wir können nun die Fusion von Virus- und Helfer-T-Zellmembran unter Beachtung von Proteininteraktionen simulieren. Dabei notieren wir statt τ als Label am Symbol der Transitionsrelation die jeweilig angewendete Reaktionsregel oder Aktion.

$$\begin{aligned} &\underline{\text{Virus}} \circ \underline{\text{TZelle}} \\ \equiv & \text{mt}_{PK_v^a} | \text{m}_v [\underline{GP_{120}(1^x + 2 + \bar{3} + 4)} \circ \underline{GP_{41}(1^x + 2 + \bar{3})}] (\text{Cap}_v) \circ \\ & \text{mt}_{CD_4(1^y)}^\perp | \text{m}_t [\underline{CD_4(1)} \circ \text{Frg}_v(1)] (\text{TP} \circ \text{Nuc}_t) \\ \xrightarrow{r_1} & \text{mt}_{PK_v^a} | \text{m}_v [\underline{GP_{120}(1^x + 2^y + 3 + 4)} \circ \underline{GP_{41}(1^x + 2 + \bar{3})}] (\text{Cap}_v) \circ \\ & \text{mt}_{CD_4(1^y)}^\perp | \text{m}_t [\underline{CD_4(1^y)} \circ \text{Frg}_v(1)] (\text{TP} \circ \text{Nuc}_t) \\ \xrightarrow{r_2} & \text{mt}_{PK_v^a} | \text{m}_v [\underline{GP_{120}(1^x + 2^y + 3^z + 4)} \circ \underline{GP_{41}(1^x + 2^z + 3)}] (\text{Cap}_v) \circ \\ & \text{mt}_{CD_4(1^y)}^\perp | \text{m}_t [\underline{CD_4(1^y)} \circ \text{Frg}_v(1)] (\text{TP} \circ \text{Nuc}_t) \\ \xrightarrow{\text{mt}} & \text{m}_v | \text{m}_t [\underline{GP_{120}(1^x + 2^y + 3^z + 4)} \circ \underline{GP_{41}(1^x + 2^z + 3)} \circ \underline{CD_4(1^y)} \circ \text{Frg}_v(1)] \\ & (\text{Cap}_v \circ \text{TP} \circ \text{Nuc}_t) \end{aligned}$$

Die hier dargestellten Proteininteraktionen entsprechen denen aus Abschnitt 4.3.1. Entscheidend ist, dass sie auch zwischen membranständigen Proteinen verschiedener Membransysteme stattfinden können. Zur Fusion von Virus- und Helfer-T-Zellmembran kann es erst dann kommen, wenn die durch seine Bindungsstelle 3 modellierte fusogene Domäne im gp41 sichtbar wird. Obwohl diese Bindungsstelle den eigentlichen Auslöser für Mate darstellt, ist die partielle Konfiguration $PK_v^a = GP_{120}(1^x + 2^y + 4) \circ GP_{41}(1^x + 3)$ als Index an mt sowie $CD_4(1^y)$ an mt^\perp erforderlich. Bindungsstelle 4 muss frei im gp120 vorliegen, um eine Reaktion bei Bindung an den freien Antikörper IgG zu verhindern. Der Verzicht auf die Forderungen 1^x und 2^y im gp120 gestattet die Fusion eines Viruspartikels und einer Helfer-T-Zelle, die nicht miteinander über gp120 und CD4 verbunden sind, denn die Verbindung y im CD4 kann in diesem Fall auch zu einem gp120-Protein auf einem beliebigen anderen Viruspartikel bestehen. Das gleiche Problem entsteht durch Weglassen der Bedingung $CD_4(1^y)$. Die vollständige Angabe von PK_v^a und $CD_4(1^y)$ ermöglicht hingegen die Überprüfung auf Verwendung derselben Verbindungsnamen sowohl zwischen gp120 und CD4 als auch zwischen gp120 und gp41 und gewährleistet dadurch die korrekte Zuordnung der miteinander zur Fusion fähigen Systeme. Die nach Mate ablaufenden Schritte der Virusinfektion sind analog zu den entsprechenden Schritten im Brane-Kalkül, wir verzichten deshalb auf ihre Darstellung.

Aktivierung von B-Zellen zur Plasmazellen-Bildung

Auch das temporäre Andocken von B-Zellen an aktivierte Helfer-T-Zellen zur Initiierung der Plasmazellenbildung werden wir nun mittels Proteininteraktionen beschreiben. Dafür benötigen wir zwei weitere Reaktionsregeln. Weil wir später auch die aus Abschnitt 4.3.1 bekannte Reaktionsregel r_3 verwenden, benennen wir diese mit r_4 und r_5 . Bei r_4 handelt es sich um eine Komplexierungsregel, bei r_5 hingegen um eine Dekomplexierungsregel.

$$\begin{aligned} r_4 &= ((Frg_v, 1, \emptyset, \emptyset), (IgM, 1, \bar{2}, \emptyset)) \\ r_5 &= ((Frg_v, 1, \emptyset, \emptyset), (IgM, 1, \emptyset, \emptyset)) \end{aligned}$$

Betrachten wir nun eine aktivierte Helfer-T-Zelle und eine B-Zelle, ergibt sich folgende Transitionsableitung.

$$\begin{aligned} &\underline{\text{TZelle}} \circ \underline{\text{BZelle}} \\ \equiv & \text{m}_t[CD_4(1) \circ \underline{Frg_v(1)}](\uparrow(TP \circ Nuc_t)) \circ \\ & \uparrow_{IgM(1+2)}^\perp |m_b[\underline{IgM(1 + \bar{2})}](\uparrow(\underline{\text{PZelle}}) \circ \text{PSyn}_b) \\ \xrightarrow{r_4} & \text{m}_t[CD_4(1) \circ \underline{Frg_v(1^x)}](\uparrow(TP \circ Nuc_t)) \circ \\ & \uparrow_{IgM(1+2)}^\perp |m_b[\underline{IgM(1^x + 2)}](\uparrow(\underline{\text{PZelle}}) \circ \text{PSyn}_b) \\ \xrightarrow{r_5} & \text{m}_t[CD_4(1) \circ \underline{Frg_v(1)}](\uparrow(TP \circ Nuc_t)) \circ \\ & \uparrow_{IgM(1+2)}^\perp |m_b[\underline{IgM(1 + 2)}](\uparrow(\underline{\text{PZelle}}) \circ \text{PSyn}_b) \\ \xrightarrow{\uparrow} & \text{m}_t[CD_4(1) \circ \underline{Frg_v(1)}](\uparrow(TP \circ Nuc_t)) \circ \\ & \underline{\text{PZelle}} \circ m_b[\underline{IgM(1 + 2)}](\text{PSyn}_b) \end{aligned}$$

Die Erzeugung einer Plasmazelle wird erst durch Freilegen der Bindungsstelle 2 im IgM möglich und kann anschließend beliebig oft stattfinden.

Freie Proteine und die Blockierung von Rezeptoren

Systeme vom Typ **Virus** sind dazu in der Lage, mittels B&R das freie Protein $GP_{120}(1+2+\bar{3}+4)$ an die Umgebung abzusondern. Auf gleiche Weise können Systeme vom Typ **PZelle** das Protein $IgG(1)$ abgeben, welches dazu in der Lage ist, an gp120 zu binden und damit dessen Interaktionen zu verhindern. Die von Membranen losgelösten Proteininteraktionen haben wir bereits im Abschnitt 4.3.1 dargestellt mithilfe der Komplexierungsregel

$$r_3 = ((IgG, 1, \emptyset, \emptyset), (GP_{120}, 4, \emptyset, \emptyset)).$$

Wir sind nun auch dazu in der Lage, sie in ihrem Membrankontext zu betrachten. Die Bindung von IgG an virusmembranständiges gp120 lässt sich modellieren mit

$$\begin{aligned} & \underline{IgG(1)} \circ m_v[\underline{GP_{120}(1^x + 2 + \bar{3} + 4)} \circ GP_{41}(1^x + 2 + \bar{3})](\text{Cap}_v) \\ \xrightarrow{r_3} & \underline{IgG(1^y)} \circ m_v[\underline{GP_{120}(1^x + 2 + \bar{3} + 4^y)} \circ GP_{41}(1^x + 2 + \bar{3})](\text{Cap}_v) \end{aligned}$$

die Bindung an gp120, welches sich im ersten Schritt in freier Form an das CD4 gesunder Helfer-T-Zellen anlagert, entsprechend mit

$$\begin{aligned} & IgG(1) \circ \underline{GP_{120}(1 + 2 + \bar{3} + 4)} \circ m_t[\underline{CD_4(1)} \circ Frq_v(1)](TP \circ Nuc_t) \\ \xrightarrow{r_1} & \underline{IgG(1)} \circ \underline{GP_{120}(1 + 2^y + 3 + 4)} \circ m_t[CD_4(1^y) \circ Frq_v(1)](TP \circ Nuc_t) \\ \xrightarrow{r_3} & IgG(1^x) \circ GP_{120}(1 + 2^y + 3 + 4^x) \circ m_t[CD_4(1^y) \circ Frq_v(1)](TP \circ Nuc_t) \end{aligned}$$

Wir stellen fest, dass nach Anlagerung von $IgG(1)$ an den Virusproteinkomplex keine Fusion des betrachteten Viruspartikels mit einer Helfer-T-Zelle mehr möglich ist, weil die Bindungsstelle 4 des GP_{120} belegt ist und somit der für die Fusion notwendige Kontext PK_y^a nicht erfüllt werden kann, der Rezeptor gp120 ist also blockiert. Die Anlagerung freier Antikörper IgG hat schließlich weitere Auswirkungen, auf die wir in dieser Arbeit nicht näher eingehen.

5.4.2 Diskussion

Der Protein-Brane-Kalkül stellt eine echte Erweiterung des Brane-Kalküls dar, jeder Term des Brane-Kalküls lässt sich mithilfe der eingeführten verkürzenden Schreibweisen auch als Term des Protein-Brane-Kalküls auffassen. Die zusätzlichen Bedingungen an den erweiterten Transitionsregeln PHAGO, EXO, PINO, MATE, BUD, DRIP und B&R sind bei derartigen Termen immer erfüllt und erlauben die gleichen Ableitungsschritte wie im Brane-Kalkül. Erweiternd integriert der Protein-Brane-Kalkül vollständig das Konzept expliziter Proteininteraktionen im Stile des $bio\kappa$ -Kalküls und ermöglicht beispielsweise die Darstellung von Signaltransduktionswegen über Membrangrenzen hinweg. Pino kann nun auch Proteinkomplexe verschlingen und bietet damit die Möglichkeit zur Modellierung des biologisch relevanten Prozesses der Endozytose membranloser Makromoleküle. Weiterhin lässt sich mittels Proteininteraktionen das temporäre Andocken von Zellen ohne biologisch nicht plausible Zwischenstände darstellen. Auch die Blockierung einzelner Rezeptoren ist realisierbar und gestattet beispielsweise die Modellierung von Inhibitoren, welche als mögliche Medikation bei der durch HIV verursachten AIDS-Erkrankung in Erprobung sind.

Auch im Vergleich zum $bio\kappa$ -Kalkül eröffnet der Protein-Brane-Kalkül zusätzliche Möglichkeiten. Die umfassenderen Mittel zur Modellierung von Membraninteraktionen im Brane-Kalkül sind vollständig integriert und an die durch die Erweiterung um Proteine entstehenden Effekte angepasst. Das Konzept der Membrannamen wird nicht benötigt, stattdessen können Aktionen mit partiellen Konfigurationen zur Forderung entsprechender Bindungsstellenbelegungen der membranständigen Proteine versehen werden, welche Voraussetzung für das Stattfinden von Interaktionen sind. Diese Struktur bietet ein hohes Maß an Kontrolle auf der Ebene einzelner Interaktionen. Die vom Brane-Kalkül bekannten Möglichkeiten zur Degeneration und Synthese biologischen Materials mittels Replikationsoperator und Operation B&R sind nun auch auf proteinbeinhaltende Strukturen anwendbar. Der Transport von Proteinkomplexen durch Membranen hindurch kann als Spezialfall mittels B&R modelliert werden und verallgemeinert damit den Transport einzelner Proteine im $bio\kappa$ -Kalkül.

Die Mutation von HIV-Partikeln ist hingegen auch im Protein-Brane-Kalkül nicht darstellbar. Eine Modellierung der Erzeugung fehlerhafter mRNA-Sequenzen und deren Vererbung in nachfolgende Virusgenerationen verlangt die Integration von Konzepten

der *Gen-Maschine*. Verzichtet man auf diese Betrachtungen und setzt nach der Genexpression an, werden zumindest Möglichkeiten zur zufälligen Manipulation vorhandener Proteinstrukturen und deren Erzeugungsmaschinerie benötigt. Derartige Erweiterungen sind nicht ohne weiteres in den Kalkül integrierbar und lassen Möglichkeiten zur Fortentwicklung. Ein weiteres Ziel stellt die Integration quantitativer Aspekte dar, wie sie beispielsweise im Stochastischen π -Kalkül ([45]) zu finden sind. Hiermit könnten Interaktionen mittels hinzugefügter Parameter an Aktionen mit einem zeitlichen Verhalten verknüpft werden und schließlich die Modellierung des konkurrierenden Verhaltens einzelner Komponenten eines biologischen Systems ermöglichen. Auch ein Konzept zur Integration privater Namen im Stile des π -Kalküls ist denkbar, um die Flexibilität bei der Zusammensetzung verschiedener Systeme zu erhöhen.

Kapitel 6

Bewertung und Ausblick

Der Brane-Kalkül stellt innerhalb der in der Systembiologie relevanten Prozesskalküle einen Hauptvertreter für Modellierungen der Membranmaschine dar. Während die β -Binder hierarchische Membranstrukturen lediglich indirekt zulassen und in den Bio-Ambients die Modellierung von Endozytose und Exozytose zur Verletzung biologischer Strukturprinzipien führt, steht im Brane-Kalkül die Wahrung der Bitonalität (siehe Abschnitt 3.4.1) und damit die detailgetreue Darstellung membranbezogener Prozesse im Vordergrund. Mit einer ausführlichen Beispielmotellierung zu Effekten der HIV-Infektion haben wir in Abschnitt 3.5.2 gezeigt, dass er für die Darstellung verschiedenster mit Membranen in Zusammenhang stehender Mechanismen geeignet ist.

Während bezogen auf andere funktionale Einheiten der biologischen Zelle einige Ansätze zu prozesskalkülgestützten Simulationen existieren (beispielsweise [19], [21] und [45]), lässt sich über Beispielmotellierungen hinaus jedoch keine Anwendung des Brane-Kalküls finden. Stattdessen gibt es zahlreiche theoretische Arbeiten, die wichtigsten haben wir in Abschnitt 3.4 vorgestellt. Auch bei den im Bereich der formalen Sprachtheorie angesiedelten P-Systemen sind Entwicklungen zur Integration der Konzepte des Brane-Kalküls feststellbar, zum Einstieg in dieses Feld sei [15] empfohlen. Eine mögliche Begründung für fehlende Simulationsansätze biologischer Prozesse mit dem Brane-Kalkül ist sein hoher Spezialisierungsgrad: obwohl Membranen eine große Rolle bei zahlreichen Zellprozessen spielen, scheint deren von anderen Aspekten losgelöste Betrachtung keine interessanten Ergebnisse zu versprechen. Damit in Zusammenhang stehen die sehr speziellen Operationen, welche zur Darstellung einiger Effekte schlecht oder gar nicht geeignet sind. Zur Steigerung seiner Attraktivität ist insbesondere die Anreicherung um Proteine vielversprechend, weil diese maßgeblich an Wechselwirkungen von Membranen beteiligt sind und selbst wiederum die Abgrenzung durch Membranen für korrekte Funktionsabläufe benötigen. Der bereits existierende vom κ -Kalkül ausgehende Ansatz des $\text{bio}\kappa$ -Kalküls eignet sich für umfangreiche Modellierungen der Protein-Maschine, lässt jedoch bezüglich Membraninteraktionen einige vorteilhafte Konzepte des Brane-Kalküls vermissen.

Die Anreicherung des Brane-Kalküls um Konzepte des $\text{bio}\kappa$ -Kalküls verspricht hingegen die Vereinigung der Vorteile beider zugrunde liegender Ansätze. Der in dieser Arbeit entwickelte Protein-Brane-Kalkül integriert sowohl die Aspekte der Membranals auch die der Protein-Maschine vollständig und ermöglicht neben der auf Protein-Konstellationen basierenden Modellierung von Membraninteraktionen die Darstellung

von Signaltransduktionswegen unter Berücksichtigung räumlicher Kompartimente. Trotz seiner veränderten Komponenten bewahrt er die strukturierte Repräsentation von Membranhierarchien des Brane-Kalküls und die umfassenden Proteininteraktionsmöglichkeiten des bio κ -Kalküls. Anhand des HIV-Beispiels haben wir in Abschnitt 5.4.1 die Ausdrucksstärke unseres Ansatzes demonstriert.

Für die detailliertere Modellierung systembiologischer Zusammenhänge besteht zunächst die Notwendigkeit zu Erweiterungen um quantitative Elemente, wie sie für andere Prozesskalküle beispielsweise in den Arbeiten [43], [27] und [3] zu finden sind. Der nächste Schritt ist eine Implementierung des Kalküls, um anschließend in intensiver Kooperation mit den Fachrichtungen der Biologie und Medizin anhand konkreter Fragestellungen Simulationen und Auswertungen der Ergebnisse vornehmen zu können. Einige Vorschläge zur Modifikation und Erweiterung der Modellierungsmöglichkeiten des Protein-Brane-Kalküls haben wir in Abschnitt 5.3 gegeben, desweiteren ist die Entwicklung eines Ansatzes zum Model Checking für die automatische Verifikation von Eigenschaften modellierter biologischer Systeme interessant.

Prozesskalküle erfahren im Bereich der Systembiologie im Vergleich zu etablierten Ansätzen wie den Petri-Netzen und Differentialgleichungen erst seit kurzer Zeit Verwendung. Ihre Vorteile liegen in der Flexibilität bei der Anpassung und Erweiterung von Modellierungen, welche nicht zuletzt auf ihrer komponentenbasierten Struktur und deren klar definierten Kommunikationsschnittstellen beruht. Darüberhinaus eignen sich ihre formalen Spezifikationen für exakte Beschreibungen und computergestützte Simulationen. Aufgrund der Spezifik biologischer Systeme werden jedoch zahlreiche Anpassungen benötigt, die Entwicklung repräsentativer und dennoch praktisch umsetzbarer Modelle verlangt geeignete Abstraktionsniveaus zur erfolgreichen Bewältigung der enormen Komplexität und stellt eine beständige Herausforderung dar. Eine Gemeinsamkeit derzeitiger auf Prozesskalkülen basierender systembiologischer Modellierungen besteht im Verzicht auf die Zuordnung räumlicher Koordinaten zu einzelnen Objekten, die Umsetzung eines Ort-Begriffes ist lediglich indirekt durch Kompartimente und in stochastischen Ansätzen durch die Modifikation einzelner Interaktionswahrscheinlichkeiten modellierbar. Angesichts des meist lokal gesteuerten Ablaufes biologischer Prozesse stellt sich die Frage, ob diese Abstraktion in jedem Fall geeignet ist. Die Anwendung von Prozesskalkülen in der Systembiologie steht erst am Anfang, Fragestellungen dieser Art können daher zum jetzigen Zeitpunkt nicht sicher beantwortet werden. Die bisher erzielten Ergebnisse lassen bezüglich ihrer Anwendbarkeit in diesem Feld jedoch auf ein hohes Potential schließen.

Literaturverzeichnis

- [1] R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. J. Pappas, H. Rubin, and J. Schug. *Hybrid Modeling and Simulation of Biomolecular Networks*. Proc. 4th International Workshop on Hybrid Systems: Computation and Control (LNCS 2034), 2001.
- [2] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, A. Troina. *A Calculus of Looping Sequences for Modelling Microbiological Systems*. Fundamenta Informaticae 72, 2006.
- [3] L. Brodo, P. Degano, C. Priami. *A Stochastic Semantics for BioAmbients*. Proc. 9th International Conference on Parallel Computing Technologies (LNCS 4671), 2007.
- [4] G.-R. Burmester, A. Pezzutto. *Taschenatlas der Immunologie*. Thieme, 1998.
- [5] N. Busi. *Deciding Behavioural Properties in Brane Calculi*. Proc. Computational Methods in Systems Biology 2006 (LNBI 4210), 2006.
- [6] N. Busi. *On the computational power of the Mate/Bud/Drip Brane Calculus: Interleaving vs. Maximal Parallelism*. Proc. 6th Workshop on Membrane Computing (LNCS 3850), 2006.
- [7] N. Busi, R. Gorrieri. *On the Computational Power of Brane Calculi*. Transactions on Computational Systems Biology VI (LNBI 4220), 2006.
- [8] N. Busi, M. A. Gutiérrez-Naranjo. *A Case Study in (Mem)Brane Computation: Generating Squares of Natural Numbers*. Proc. 4th Brainstorming Week on Membrane Computing I, 2006.
- [9] N. Busi, C. Zandron. *Modeling and analysis of biological processes by mem(brane) calculi and systems*. Proc. Winter Simulation Conference, 2006.
- [10] N. Cannata, F. Corradini, E. Merelli, A. Omicini, A. Ricci. *An Agent-Oriented Conceptual Framework for Systems Biology*. Transactions on Computational Systems Biology III (LNBI 3737), 2005.
- [11] L. Cardelli. *Abstract Machines of Systems Biology*. Transactions on Computational Systems Biology III (LNBI 3737), 2005.
- [12] L. Cardelli. *Bitonal Membrane Systems, Interactions of Biological Membranes*. To appear in Theoretical Computer Science, 2008.
- [13] L. Cardelli. *Brane Calculi, Interactions of Biological Membranes*. Proc. Computational Methods in Systems Biology 2004 (LNBI 3082), 2005.

- [14] L. Cardelli, A. D. Gordon. *Mobile Ambients*. Formal Methods for Distributed Processing, A Survey of Object-Oriented Approaches. H. Bowman, J. Derrick (Hrsg.), Cambridge University Press, 2001.
- [15] L. Cardelli, G. Paun. *An Universality Result for a (mem)brane calculus based on mate/drip operations*. International Journal of Foundations of Computer Science 17(1), 2005.
- [16] L. Cardelli, S. Pradalier. *Where membranes meet complexes*. Proc. Concurrent Models in Molecular Biology (Bioconcur'05), 2005.
- [17] P.-L. Curien, V. Danos, J. Krivine, M. Zhang. *Computational self-assembly*. To appear, 2007.
- [18] V. Danos. *Agile modelling of cellular signalling*. Proc. International Conference of Computational Methods in Sciences and Engineering, 2007.
- [19] V. Danos, J. Feret, W. Fontana, R. Harmer, J. Krivine. *Rule-based modelling of cellular signalling*. Proc. 18th International Conference on Concurrency Theory, 2007.
- [20] V. Danos, J. Feret, W. Fontana, J. Krivine. *Abstract interpretation of cellular signalling networks*. To appear in Proc. 9th International Conference on Verification, Model Checking and Abstract Interpretation, 2008.
- [21] V. Danos, J. Feret, W. Fontana, J. Krivine. *Scalable simulation of cellular signaling networks*. Proc. 5th Asian Symposium on Programming Languages and Systems, 2007.
- [22] V. Danos, C. Laneve. *Core formal molecular biology*. Proc. 12th European Symposium on Programming (LNCS 2618), Springer, 2003.
- [23] V. Danos, C. Laneve. *Formal Molecular Biology*. Theoretical Computer Science 325(1), 2004.
- [24] V. Danos, C. Laneve. *Graphs for Core Molecular Biology*. Proc. First International Workshop on Computational Methods in Systems Biology (LNCS 2602), 2003.
- [25] V. Danos, S. Pradalier. *Projective Brane Calculus*. Proc. Computational Methods in Systems Biology 2004 (LNBI 3082), 2005.
- [26] V. Danos, F. Tarissan. *Self Assembling Graphs*. Natural Computing: an international journal 6(3), 2007.
- [27] P. Degano, D. Prandi, C. Priami, P. Quaglia. *Beta-binders for biological quantitative experiments*. Electronic Notes in Theoretical Computer Science 164(3), 2006.
- [28] K. W. Kohn. *Molecular Interaction Map of the Mammalian Cell Cycle Control and DNA Repair Systems*. Molecular Biology of the Cell 10, 1999.
- [29] J. Koolman, K.-H. Röhm. *Taschenatlas der Biochemie. (2. Auflage)* Thieme, 1998.

- [30] C. Laneve, F. Tarissan. *A simple calculus for proteins and cells*. Electronic Notes in Theoretical Computer Science 171, 2007.
- [31] S. Liu, H. Lu, J. Niu, Y. Xu, S. Wu, S. Jiang. *Different from the HIV Fusion Inhibitor C34, the Anti-HIV Drug Fuzeon (T-20) Inhibits HIV-1 Entry by Targeting Multiple Sites in gp41 and gp120*. JBC Papers in Press, 2005.
- [32] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, J. Darnell. *Molecular Cell Biology. (Fourth Edition)* W. H. Freeman and Company, 2000.
- [33] R. Mardare, A. Ihekwaba. *A Calculus for Modelling, Simulating and Analysing Compartmentalized Biological Systems*. International Conference of Computational Methods in Sciences and Engineering, 2007.
- [34] M. Miculan, G. Bacci. *Modal Logics for Brane Calculus*. Proc. Computational Methods in Systems Biology 2006 (LNBI 4210), 2006.
- [35] M. Miculan, G. Bacci. *Undecidability of Model checking in Brane Logic*. Elsevier Science B. V., 2007.
- [36] R. Milner. *Communicating and Mobile Systems: the π -Calculus*. Cambridge University Press, 1999.
- [37] S. Modrow, D. Falke. *Molekulare Virologie*. Spektrum, 1997.
- [38] A. Obtulowicz. *Mathematical (denotational) semantics of some reducts of Ambient Calculus and Brane Calculi*. Proc. 7th Workshop on Membrane Computing (LNCS 4361), 2006.
- [39] G. Paun. *Computing with Membranes*. Journal of Computer and System Sciences 61(1), 2000.
- [40] H. Plattner, J. Hentschel. *Taschenlehrbuch Zellbiologie. (3. Auflage)* Thieme, 2006.
- [41] T. Porstmann. *Virusdiagnostik*. Blackwell Wissenschaft, 1996.
- [42] D. Prandi, C. Priami, P. Quaglia. *Process Calculi in a biological context*. Bulletin of the EATCS 85, 2005.
- [43] C. Priami. *Stochastic π -Calculus*. The Computer Journal 6, 1995.
- [44] C. Priami, P. Quaglia. *Beta Binders for Biological Interactions*. Proc. Computational Methods in Systems Biology 2004 (LNBI 3082), 2005.
- [45] C. Priami, A. Regev, E. Shapiro, W. Silverman. *Application of a stochastic name-passing calculus to representation and simulation of molecular processes*. Information Processing Letters 80, 2001.
- [46] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, E. Shapiro *BioAmbients: An abstraction for biological compartments*. Theoretical Computer Science, Special Issue on Computational Methods in Systems Biology 325(1), 2004.

- [47] A. Regev, E. Shapiro. *Cells as Computation, Cellular Abstractions*. Nature Vol 419, 2002.
- [48] A. Sackmann, M. Heiner, I. Koch. *Application of Petri net based analysis to signal transduction pathways*. BMC Bioinformatics 7, 2006.
- [49] F. Stenger. *Adaptionen von Prozesskalkülen für eine Anwendung in der Systembiologie*. Diplomarbeit, Technische Universität Dresden, 2007.
- [50] A. Vitale, G. Mauri. *Communication via Mobile Vesicles in Brane Calculi*. Electronic Notes in Theoretical Computer Science 171(1), 2007.
- [51] X. Xu, X. Dong, Y. Fu. *A Model in κ for DNA Addition*. Electronic Notes in Theoretical Computer Science 171, 2007.