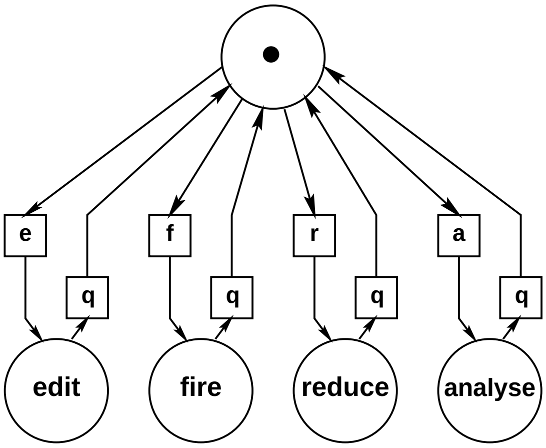
INA
Integrierter Netz Analysator
Version 1.7



Handbuch

Stephan Roch

Prof. Peter H. Starke

Humboldt–Universität zu Berlin
Institut für Informatik
Lehrstuhl für Automaten– und Systemtheorie
Axel Springer Straße 54a
10117 Berlin

Vorwort

Der Integrierte Netz Analysator INA ist ein Programm, mit dessen Hilfe Petri-Netze verschiedenster Typen unter verschiedenen Schaltregeln, insbesondere unter Zeitbewertung, hinsichtlich allgemeiner Eigenschaften untersucht werden können. Typische Eigenschaften, die durch Analyse nachgewiesen werden, sind Beschränktheit von Plätzen, Lebendigkeit von Transitionen, Erreichbarkeit bzw. Nicht-Erreichbarkeit von Markierungen bzw. Zuständen.

Mit der Modellierung eines (vorhandenen oder geplanten) Systems in der Sprache der Petri-Netze, d.h. dem Entwurf eines Petri-Netz-Modells, ist eine Abstraktion verbunden, bei der bestimmten Eigenschaften des Systems Eigenschaften des Netzes zugordnet sind, z.B. kann der Verklemmungsfreiheit des Systems die Nichterreichbarkeit einer toten Markierung entsprechen.

Der Lebenszyklus eines Netzmodells kann grob als

- Entwurf
- Analyse
- Interpretation
- Neuentwurf

skizziert werden. Dabei kommt der Interpretation der Analyseergebnisse eine nicht zu unterschätzende Bedeutung zu.

INA wurde von Prof. Dr. Peter H. Starke zur Analyse von Netzen entwickelt (es umfasst die in früheren Werkzeugen „Petri-Netz-Maschine“, „PAN“, „CPNA“ und „ATNA“ angebotenen Analyseverfahren und ist wesentlich erweitert worden). Mit diesem interaktiven, menügesteuerten Programm können Sie Netze editieren, reduzieren, ausführen und analysieren.

Dieses Handbuch beschreibt die Funktionalität der Version 1.7 von INA. Es wurde von Stephan Roch in Zusammenarbeit mit Prof. Peter H. Starke erstellt.

Folgende neue Funktionen werden hier erstmals beschrieben:

- sture Reduktion des Erreichbarkeitsgraphen
- Überdeckbarkeitsgraph nach Finkels Methode

Erweiterungen und Änderungen gibt es in der Menüführung und z.B. bei der Pfadberechnung und der Graphanalyse. Auch intern wurden Veränderungen vorgenommen.

Zum Verständnis des Programms ist eine gewisse Kenntnis der Netztheorie, mindestens ihrer grundlegenden Definitionen notwendig. Wir verweisen dazu auf das Buch *„Analyse von Petri–Netz–Modellen“* erschienen 1990 in der Reihe „Leitfäden und Monographien der Informatik“ beim B. G. Teubner Verlag in Stuttgart [Sta90]. Darin sind auch viele der in INA implementierten Algorithmen näher beschrieben.

Für Fehler in diesem Handbuch oder im Programm wird keine Haftung übernommen.

Berlin im April 1997

Peter H. Starke und Stephan Roch

Inhaltsverzeichnis

Anhang

Weitere Verzeichnisse

Übersichten und Abbildungen

Beispielnetze

1. Einleitung

1.1. Überblick

Dieses Handbuch ist in sechs Kapitel und einen Anhang eingeteilt. Die Kapitel enthalten Informationen, die in die Benutzung des Integrated Net Analyzers INA einführen und die angebotenen Funktionen vorstellen.

Kapitel 2 stellt die Installation, die Handhabung und den Dialog mit dem Programm, die Dateneingabe, verwendete Dateien, Extensionen und Datenformate, die Struktur einer INA–Session und mögliche Optionen vor.

Empfehlung: Bitte lesen Sie dieses Kapitel zuerst, dadurch können Ihnen viele Unannehmlichkeiten bei der Benutzung dieses Programms erspart bleiben.

Das Kapitel 3 beschreibt den Editor, das Kapitel 4 die Schaltausführung. Es enthält unter anderem eine Beschreibung der Zustandsdarstellung und des Schaltens unter Zeitbewertung.

Die Kapitel 5 und 6 führen in das Reduzieren und Analysieren von Netzen ein.

Im Anhang finden Sie Hinweise auf Literatur, die Definitionen und Implementationsdetails enthält, außerdem ein Stichwortverzeichnis und einen Glossar der im Programm vorkommenden Meldungen, Kommandos und Abfragen. Ebenso finden Sie dort weitere Informationen über dieses Handbuch und das Programm.

1.2. Darstellungsweise

Kommandos werden in folgender Form vorgestellt:

`<Taste>     Beispielkommando`

Ab hier startet die Erklärung des Kommandos, etwas eingerückt.

Dabei bezeichnet `<Taste>` die Tastenfolge, die am Terminal eingegeben werden muß, um den Menüpunkt `Beispielkommando` anzuwählen.

Einzelne Menüpunkte werden teilweise als Randbemerkungen notiert. `Beispielmenü`

Besondere Tasten und Zeichen

Die Taste `<cr>` steht für die Eingabe von `Return` bzw. `Enter`, `<esc>` für `Escape` und `<space>` für die Leertaste (zu den Tasten `<delete>` und `<backspace>` siehe die Bemerkungen auf Seite 4 im Kapitel 2.2).

Das Zeichen `oo` steht für ∞ und ist durch ein einzelnes `<o>` einzugegeben (siehe z.B. unter unendliche Kapazität im Kapitel 2.5.3 auf Seite 21).

Weitere Darstellungsweisen

Alle Beispieleingaben, Programmauszüge und Dateiausschnitte werden im gleichen Format wie das ursprüngliche gezeigt:

```
Der Text ist eingerueckt und in einer

etwas kleineren Fixschrift gesetzt.
```

Verweise auf das Literaturverzeichnis haben folgende Form, wobei nicht immer eine genauere Beschreibung der Verweisstelle möglich ist: [Buchkürzel, Verweisstelle (Seiten)].

Eine Beschreibung der an vielen Stellen verwendeten EBNF-Syntax finden Sie auf Seite 13 zu Beginn des Kapitels 2.3.4.

2. Grundlagen

In diesem Kapitel finden Sie alle wichtigen Informationen für den Umgang mit INA, angefangen von der Installation über generelle Regeln für den Dialog und die Struktur einer sogenannten INA–Session, eine Übersicht der verwendeten Dateien, Extensionen und Datenformate, bis zu den Optionen, die eingestellt werden können. Bitte lesen Sie dieses Kapitel sorgfältig vor der ersten Arbeit mit dem Programm durch.

2.1. Startvorbereitungen

In der Regel sollten Sie ein Verzeichnis auf Ihrer Festplatte einrichten und alle Dateien auf der INA–Diskette in dieses Verzeichnis kopieren. Die dafür nötigen Handhabungen entnehmen Sie bitte der Dokumentation Ihres Betriebssystems.

INA sucht alle Dateien in dem Verzeichnis, aus dem das Programm gestartet wurde, und schreibt seine Dateien wieder in dieses Verzeichnis. Sollten Sie gleichzeitig mit mehreren verschieden Netztypen arbeiten, ist es empfehlenswert, für jedes Netz oder jeden Netztyp ein eigenes Unterverzeichnis anzulegen. Da dadurch die aktuellen Optionen separat gespeichert werden, müssen Sie diese nicht jedesmal neu einstellen. Es ist dann ausreichend, das Verzeichnis zu wechseln und das Programm neu zu starten.

Das Verzeichnis, in dem die ausführbaren Dateien der INA–Diskette abgelegt wurden, sollte in den Suchpfad aufgenommen werden.

Installation unter DOS

Unter DOS können Sie INA selbstverständlich auch von Ihrer Diskette aus starten, wenn diese nicht schreibgeschützt ist. Sie riskieren aber einen Abbruch dadurch, daß die Diskette keinen freien Platz mehr enthält.

Wenn Sie die Programmbestandteile von INA im Verzeichnis `c:\ina\bin` abgelegt und ein Verzeichnis `c:\ina\pnt` angelegt haben, in dem sich zu analysierende Netze und Analysedaten befinden, müssen Sie den Suchpfad mit `PATH c:\dos; ... ;c:\ina\bin` setzen und können danach Netze im Verzeichnis `c:\ina\pnt` analysieren.

Installation unter UNIX

Wenn Sie die Programmbestandteile von INA im Verzeichnis `~/ina/bin` abgelegt und ein Verzeichnis `~/ina/pnt` angelegt haben, in dem sich zu analysierende Netze und Analysedaten befinden, müssen Sie den Suchpfad mit `setenv PATH "/usr/bin: ... :~/ina/bin"` (Beispiel für die C-Shell) setzen und können danach Netze im Verzeichnis `~/ina/pnt` analysieren.

2.2. Generelle Regeln für den Dialog mit INA

Bevor die von INA bereitgestellten Funktionen einzeln erläutert werden, finden Sie hier wichtige Hinweise über den Dialog und die Arbeit mit dem Programm.

Beantwortung von Abfragen

Da INA ein interaktives Programm ist, müssen Sie an vielen Stellen Fragen mit `<Y>` oder `<N>` beantworten. Standardantwort für alle diese Fragen ist immer `<N>`. Wenn Sie eine andere Taste als `<Y>` drücken, wird immer die Antwort `<N>` angenommen. Die Antwort `yes` kann durch Eingabe von `<y>` oder `<Y>` geschehen, INA unterscheidet hier nicht zwischen Groß- und Kleinbuchstaben.

Dateianforderungen

Wenn Sie aufgefordert werden, z.B. einen Namen für eine Datei oder einen Namen für einen Platz einzugeben, so schlägt INA oftmals einen Standartwert vor, der Cursor steht dann auf dem ersten Zeichen. Diese Vorgabe können Sie mit `<cr>` übernehmen. Wollen Sie dagegen einen anderen Namen eingeben, können Sie das durch einfaches Überschreiben erreichen.

Editieren der Eingabe

Leider ist es nicht möglich, die `<delete>`, die `<backspace>` und die Cursortasten zum Editieren zu benutzen, bei UNIX–Systemen kann dies sogar in einigen Fällen zum Abbruch des Programmes führen. Deshalb Vorsicht mit diesen Tasten! INA bietet stattdessen andere Möglichkeiten an, um das erwünschte zu erreichen:

Mit `<space>` können Sie den Cursor um eine Position nach rechts bewegen, ohne Zeichen zu überschreiben. Die Eingabe von `<?>` dagegen bewegt den Cursor um eine Position nach links. Sie können also `<?>` als Backspace benutzen, da Sie immer das Zeichen unter dem Cursor überschreiben können. Fordert INA gerade keinen Dateinamen an, können Sie mit `<#>` ein Leerzeichen eingeben.

Eingabe von Namen

Wenn Sie einen Namen eingegeben haben und `<cr>` drücken, wird INA alles vor der aktuellen Cursorposition als Eingabe übernehmen. `<esc>` akzeptiert dagegen die ganze Zeile als Eingabe, unabhängig von der aktuellen Cursorposition.

Bei der Eingabe von Dateinamen wird automatisch die passende Erweiterung angehängt, wenn die Eingabe keinen Punkt enthält.

Bitte benutzen Sie weder Pfadangaben, noch Sonderzeichen als Teil eines Dateinamens, denn dies kann zu einem Programmabbruch führen.

Fordert INA Daten an, z.B. einen Dateinamen oder eine Nummer eines Platzes und Sie geben nichtexistierende Daten ein, dann erhalten Sie die Meldung `not found!` und können die geforderten Daten erneut eingeben.

Umschaltung in den Terminalmodus

Bei Dateianforderungen ist es möglich durch Eingabe von `<esc>` in den Terminalmodus zu gelangen. Danach werden die benötigten Daten direkt über die Tastatur eingegeben. Dies ist insbesondere dann notwendig, wenn Sie keine Daten des angeforderten Types zur Verfügung haben (z.B. bei den ersten Arbeiten mit INA).

Speichern von Daten

Beim Speichern von Daten in ein `output file` geht INA durch Eingabe von `<esc>` ebenfalls in den Terminalmodus, d.h. Sie sehen die Ausgabe auf dem Bildschirm, teilweise wird die Ausgabe aber auch ganz übergangen. Auf das Speichern von Daten in einer schon existierenden Datei wird mit der Frage `overwrite?` reagiert, dabei wird die Datei durch Eingabe von `<Y>` überschrieben und durch `<N>` ein neuer Name abgefragt.

2.3. Dateien, Extensionen, Formate und die Netzeingabe

In diesem Abschnitts finden Sie einen Überblick über die Dateien, die von INA benutzt und erstellt werden, einen Liste der verwendeten Dateiextensionen, eine Vorstellung einiger Dateiformate und eine Beschreibung der Eingabe von Netzen.

2.3.1. Wichtige Dateien

`OPTIONS.ina` In dieser Datei speichert INA (bei regulärer Termination) alle Optionen ab, die Sie während Ihrer Sitzung eingestellt haben, z.B. den Netztyp, die Ausgabeformate, die Tiefe der zu erzeugenden Erreichbarkeitsgraphen usw. Zu Beginn der nächsten Sitzung werden diese Optionen automatisch wieder eingestellt. Siehe dazu auch das Kapitel 2.5 ab Seite 19, in dem die Optionen vorgestellt werden, die im Optionsmenü geändert werden können. Alle anderen Einstellungen können nur an bestimmten Stellen des Programms modifiziert werden; in diesem Handbuch finden Sie Beschreibungen in den entsprechenden Kapiteln und Abschnitten.

`COMMAND.ina` Diese Datei wird am Schluß einer Sitzung auf Anfrage erzeugt. In ihr werden alle während dieser Sitzung eingegebenen Kommandos abgespeichert. Wird nach einem Neustart in INA die Anfrage `Same procedure as last time?` mit `<Y>` beantwortet, so werden die gespeicherten Kommandos in der gegebenen Reihenfolge abgearbeitet, d.h. die Sitzung wird wiederholt. Wenn Sie währenddessen `<H>` wie `halt` eingegeben, geht INA wieder in den normalen Modus (Kommandos vom Terminal) über.

Im Kapitel 2.4.3 wird erläutert, wie Sie auf diese Art mehrere Netze nacheinander nach dem gleichen Schema analysieren können.

`SESSION.ina` Von INA werden alle während einer Sitzung erzielten Resultate (und Folgerungen) in diese Datei geschrieben. `SESSION.ina` wird bei Beginn der nächsten Sitzung überschrieben, Sie sollten diese Datei also vorher umbenennen, wenn sie noch von Bedeutung ist!

Weitere Bemerkungen zu dieser Datei finden Sie in den Kapiteln 2.4.1, 2.4.2 und 2.4.3.

Weitere Dateien, die sich nach einem Lauf von INA im aktuellen Verzeichnis befinden können:

`STATES.gra` Informationen über Erreichbarkeits-, Überdeckbarkeits oder Zustandsgraphen (wird zur Analyse des Graphen benötigt)

`INVARI.hlp` Hilfsdatei für Invariantenberechnung (bei der Berechnung von minimalen semi–positiven Invarianten)

`ININET.pnt` Hilfsdatei, die das Anfangsnetz enthält (wird vor der Durchführung von Reduktionsschritten gespeichert — siehe Seite 46 im Kapitel 5.2

2.3.2. Dateiextensionen

INA benutzt standardmäßig folgende Extensionen:

`.pnt` Netzbeschreibung eines Platz/Transitions–Netzes (siehe Kapitel 2.3.4 ab Seite 14)

`.cnt` Netzbeschreibung eines gefärbten Netzes (siehe Kapitel 2.3.4 ab Seite 15)

`.mar` Markierung eines Netzes

`.tim` Zeitintervalle der Transitionen bzw. Transitionsfarben

`.tmd` Schaltdauer der Transitionen bzw. Transitionsfarben

`.atm` Zeitintervalle der Bögen

`.pdl` Verweilzeiten der Marken auf den Plätzen bzw. Platzfarben

`.itm` Initiale Uhrenstellungen

`.pri` Prioritäten

`.cap` Kapazitäten

`.red` Reduktionsresultate

`.exc` Reduktionsausnahmen

`.res` Resultate allgemein

`.inv` Invarianten

`.hlp` Hilfsdatei mit unformatierte Invarianten

`.gra` Graphen

`.tre` Bäume

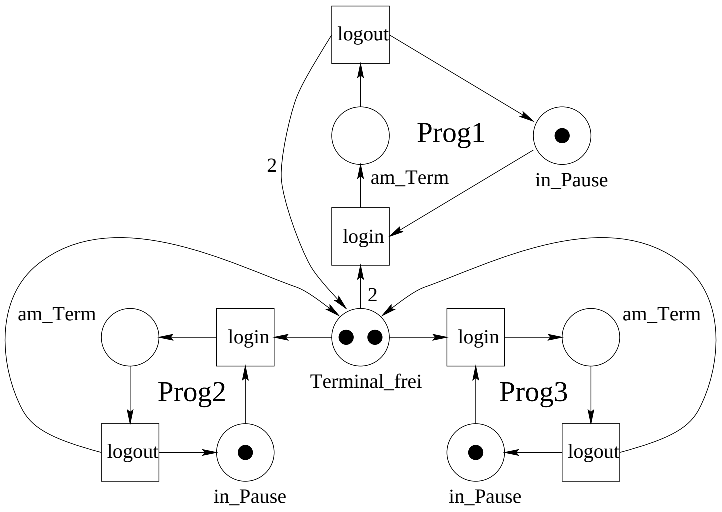
`.val` Bewertungen

2.3.3. Eingabe von Netzen

Netze können im Editor mit dem Kommando `<R><T>` (siehe Seite 26) eingegeben werden. Außerdem können Sie bei einer Dateianforderung eines Netzes mit `<esc>` in den Terminalmodus umschalten und das Netz, wie hier beschrieben, eingeben.

Ungefärbte Netze

Das ungefärbte Netz des Beispiel 1 modelliert die Teilung zweier Terminals durch drei Programmierer. Ein Programmierer ist entweder in der Pause oder bei der Arbeit an einem Terminal. Der Programmierer 1 benötigt zu seiner Arbeit gleichzeitig zwei Terminals. Die Transitionen modellieren das An– und Abmelden der Programmierer.



Beispiel 1: Drei Programmierer teilen sich zwei Terminals

Im folgenden wird die Eingabe dieses Netzes gezeigt:

Zuerst werden Netznummer (Standardwert: 0) und Netzname (maximal 16 Zeichen) erfragt. Es empfiehlt sich, diese beiden Rubriken auszufüllen, Netznummer und –name erscheinen an vielen Stellen im Protokoll und den gespeicherten Dateien und verhindern so Konfusionen.

Danach erscheint die Syntax für eine platzorientierte Eingabe, die wie folgt lautet:

```
Input syntax:
<nr>" " <tokens>" "[ <prelist> ] [ "," <postlist> ] "<cr>"
 <prelist> ::= { <nr> [ ":"<mult> ]" " }
<postlist> ::= { <nr> [ ":"<mult> ]" " }
P>
```

Als Eingabe wird jeweils eine Zeile erwartet, die einen Platz beschreibt. Eine solche Zeile beginnt mit der Nummer des Platzes, gefolgt von einem Leerzeichen und der Anzahl der Marken bei der Anfangsmarkierung. Soll der Platz Vortransitionen haben, so werden diese als durch Leerzeichen getrennte Elemente einer Liste in der Form `<Nummer>:<Vielfachheit>`

eingegeben. Die Angabe der einfachen Vielfachheit :<1> kann entfallen, da diese als Standard gesetzt ist. Die Liste der Vortransitionen kann selbstverständlich auch leer sein. Wenn der Platz Nachtransitionen haben soll, so müssen Sie analog eine durch <,> eingeleitete Liste eingeben. Der Abschluß einer Platzzeile erfolgt mit <cr>.

Für das Netz 1 der drei Programmierer, die sich zwei Terminals teilen, müssen Sie folgende Zeilen eingeben:

```
Input of Place/Transition Net nr: 1    Netname = 3_Prog_2Term

0 2 4:2 5 6, 1:2 2 3
1 0 1, 4
2 0 2, 5
3 0 3, 6
4 1 4, 1
5 1 5, 2
6 1 6, 3

Q
```

Die Zeile `0 2 4:2 5 6, 1:2 2 3` beschreibt den Platz 0, der initial zwei Marken enthält. Seine Vortransitionen sind die Transitionen 4, 5 und 6, dabei hat der Bogen von der Transition 4 zum Platz 0 die Vielfachheit zwei. Der Platz 0 hat die Nachtransitionen 1, 2 und 3, dabei hat der Bogen von dem Platz 0 zur Transition 1 ebenfalls die Vielfachheit zwei.

Beachten Sie bei der Eingabe die generellen Regeln, die im Kapitel 2.2 auf Seite 4 beschrieben wurden: Mit <?> können Sie ein Zeichen löschen. Zur Eingabe großer Transitionslisten empfiehlt es sich, nach Abschluß eines Listenelementes mit einem Kleiner–Zeichen < < >, gefolgt von <cr>, die Eingabe auf einer neuen Zeile fortzusetzen.

Wenn Sie alle Plätze durch ihre Nummer eingegeben haben, antworten Sie zum Beenden der Platzanforderung am P>–Prompt mit <Q>.

Als nächstes fordert INA für jeden von Ihnen eingegebenen Platz einen Namen an (maximal 16 Zeichen). Durch Eingabe von <cr> werden noch nicht belegte Stellen mit Leerzeichen besetzt. Sollte die Kapazitätsschaltregel ausgewählt sein (siehe Kapitel 2.5.3 auf Seite 21, fordert INA für jeden Platz zusätzlich die Angabe einer Kapazität. Standardwert ist <oo>, eine rechnerabhängige Konstante, die für eine unendliche Kapazität steht.

Anschließend werden in der selben Weise die Transitionen mit Namen versehen und eventuell als Fakten gekennzeichnet. Falls Sie mit Prioritäten arbeiten (siehe Kapitel 2.5.4 auf Seite 22), werden diese nach der Namenseingabe angefordert. Standardwert ist null.

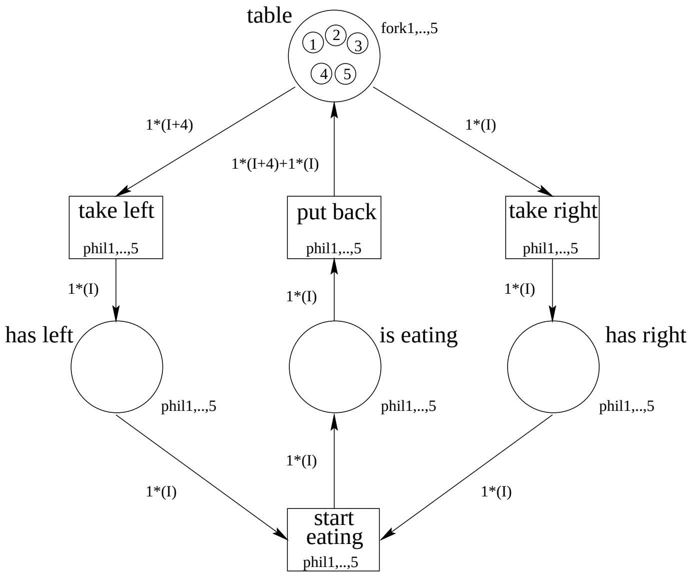
Falls Sie eine Zeitbewertung als Option gesetzt haben (siehe Kapitel 2.5.2 auf Seite 20), wird je nach Bewertungsart und bewertetem Netzelement eine initiale Uhrenstellung abgefragt (außer bei Zeitoption Dauer für Plätze).

Damit ist die Netzeingabe beendet und Sie können das Netz anschließend mit <W><F> auf eine Datei schreiben.

Gefärbte Netze

Das gefärbte Netz des Beispiels 2 modelliert das wohlbekannte Problem der Speisenden Philosophen: Fünf Philosophen sitzen um einen runden Tisch auf dem sich fünf Gabeln befinden.

Um vom Zustand „Denkend“ in den Zustand „Essend“ überzugehen, braucht ein Philosoph die Gabel zu seiner linken und rechten Seite. Nach dem Essen legt er beide Gabeln wieder ab und geht zurück in den Zustand „Denkend“.



Beispiel 2: Speisende Philosophen

An den Bögen des Netzes stehen die entsprechenden Farbinformationen.

Das Netz wird wie folgt eingegeben:

Nach dem Eingeben von Netznummer und –name werden zuerst alle Plätze definiert durch Angabe der Nummer, des Namens und der Farbmenge. Bei der Platzeingabe schlägt INA eine Nummer vor, diese können Sie entweder mit `<cr>` bestätigen oder durch eine Zahl überschreiben. Anschließend wird der Name des Platzes (maximal 16 Zeichen) sowie die Anzahl der Farben erfragt. Danach werden die Namen der Farben und die Anfangsmarkierung des Platzes erfragt, unter der Kapazitätsschaltregel auch die Kapazität.

Da sich die Namen der Farben eines Platzes häufig nur geringfügig unterscheiden, wird ab der zweiten Farbe der vorherige Farbname zum Editieren vorgelegt. Diese Vorgabe können Sie durch Überschreiben ändern, `<space>` bewegt den Cursor nach rechts, `<?>` nach links.

Für die Eingabe der Anfangsmarkierung sind zwei besondere Werte vorgesehen: Wenn Sie bei der ersten Farbe statt einer Markenzahl `<z>` (für `zero`) oder `<s>` (für `standard`) eingeben, so wird automatisch für alle Farben dieses Platzes die Markenzahl gleich null bzw. eins gesetzt.

Vielfach haben Netzknoten die gleiche Farbmengen. Nach der Eingabe des ersten Platzes erscheint daher nach dem Eingeben des Knotennamens ein Menü:

```
Colour set to be
read from the terminal................................R
copied from an existing node..........................C
```

```
constructed as square CxC of a known colour set.......S
constructed as product CxC1 of known colour sets......P
```

Sie können mit `<C>` die Farbmenge von einem bereits konstruierten Knoten kopieren, mit `<S>` als Cartesisches Quadrat $C \times C$ einer vorhandenen Farbmenge C oder mit `<P>` als Cartesisches Produkt $C \times C_1$ zweier vorhandener Farbmengen konstruieren bzw. mit `<R>` sie, wie bereits beschrieben, von Hand eingeben.

Die Platzdefinition wird durch `<Q>` auf die Anfrage einer Platznummer beendet.

Für das Netz 2 der fünf speisenden Philosophen sieht der Eingabedialog für die Netznummer, den Netznamen und die Plätze wie folgt aus (Darstellung teilweise gekürzt):

```
Input of Coloured Petri Net nr: 2    Netname = Dinner
Please, give first the places and the initial marking.

place nr.     1 name =     table
   number of colours: 5
 colour    1: name = fork1             m0: 1
 colour    2: name = fork2             m0: 1
 colour    3: name = fork3             m0: 1
 colour    4: name = fork4             m0: 1
 colour    5: name = fork5             m0: 1

place nr.     2 name =     has_left
Colour set to be
read from the terminal................................R
copied from an existing node..........................C
constructed as square CxC of a known colour set.......S
constructed as product CxC1 of known colour sets......P > R
   number of colours: 5
 colour    1: name = phil1             m0: 0
 colour    2: name = phil2             m0: 0
 colour    3: name = phil3             m0: 0
 colour    4: name = phil4             m0: 0
 colour    5: name = phil5             m0: 0

place nr.     3 name =     has_right
Colour set to be
......................................................
copied from an existing node..........................C > C
place nr. 2.has_left
 colour    1: phil1             m0: 0
 colour    2: phil2             m0: 0
 colour    3: phil3             m0: 0
 colour    4: phil4             m0: 0
 colour    5: phil5             m0: 0

place nr.     4 name =     is_eating
Colour set to be
......................................................
copied from an existing node..........................C > C
place nr. 2.has_left
 colour    1: phil1             m0: 0
 colour    2: phil2             m0: 0
 colour    3: phil3             m0: 0
 colour    4: phil4             m0: 0
 colour    5: phil5             m0: 0

place nr.     Q
```

Anschließend können Sie die Transitionen und Bögen eingeben. Für die Transitionen werden Nummer, Name, Farbmenge und Fakt–sein, gegebenenfalls auch die Prioritäten angefordert.

Danach werden zuerst die Eingangsbögen der Transitionen, anschließend die Ausgangsbögen eingegeben.

Für die Definition der Bögen gibt es drei Modi: `(C)opy, (E)xpression, by (H)and`. Mit `<H>` können Sie die Bögen per Hand eingeben. Für jede Transitionsfarbe ist die ihr durch die Funktion am Bogen zugeordnete Multimenge von Platzfarbnummern als formale Summe einzugeben. Der Kopiermodus `<C>` ist anzuwenden, wenn es bereits einen Bogen gibt, der wunschgemäß beschriftet ist. Sie geben in diesem Fall nur die Netzknoten an, die der vorhandene Bogen verbindet. Im Ausdruckmodus `<E>` können Sie vordefinierte Funktionen zur Bogenbeschriftung verwenden. Das Angebot hängt von der Anzahl der Farben der Transition und des Platzes ab. Die Identitätsfunktion `<I>` (für `identity`) wird immer angeboten.

In den folgenden Veranschaulichungen wird ein Bogen zwischen der Transition t mit f_t Farben und einem Platz p mit f_p Farben betrachtet. Der Ausdruck $3 * (I + 5) + 1 * (I + 7)$ als Bogenbeschriftung bewirkt, daß beim Schalten der Transition t in der Farbe j (mit $1 \leq j \leq f_t$) vier Marken durch diesen Bogen fließen, und zwar drei Marken mit der Platzfarbe $(I + 5)(j)$ und eine Marke der Farbe $(I+7)(j)$. Dabei ist bei $f_p = 1$ und beliebigem n stets $(I+n)(j) = 1$, und bei $f_p > 1$ ist $(I + n)(j) = ((j + n - 1) \bmod f_p) + 1$. Für $n = 0$ und $f_p = f_t$ ergibt sich also die identische Abbildung.

Ist $f_t > f_p$ und f_t durch f_p ohne Rest teilbar, dann werden die Funktionen `<F>` (für `first component`) und `<S>` (für `second component`) angeboten. INA nimmt an, daß die Farbmenge der Transition ein Cartesisches Produkt von Farbmengen ist, bei dem die Farbmenge von p als erster (mit `<F>`) bzw. zweiter (mit `<S>`) Faktor auftritt. Für eine Transitionsfarbnummer j ist $F(j)$ bzw. $S(j)$ die Farbe der ersten bzw. zweiten Komponente der Farbe j: $F(j) = ((j - 1) \operatorname{div} (f_t \operatorname{div} f_p)) + 1)$ und $S(j) = ((j - 1) \bmod f_p) + 1$

Ist $f_t < f_p$ und f_p durch f_t ohne Rest teilbar, dann werden die Funktionen `<M>` (für `mine`) und `<H>` (für `his`) angeboten. INA nimmt an, daß die Farbmenge von p ein Cartesisches Produkt von Farbmengen ist, bei dem die Farbmenge von t als erster (mit `<M>`) bzw. als zweiter (mit `<H>`) Faktor auftritt. Für eine Transitionsfarbnummer ist $M(j)$ bzw. $H(j)$ die formale Summe der Platzfarben, deren erste bzw. zweite Komponente gleich j und deren zweite bzw. erste Komponente ungleich j ist.

Bei $f_p = f_t \cdot f_t$ wird zusätzlich die Funktion `<D>` (für `diagonal`) angeboten. INA nimmt an, daß die Farbmenge von p das Cartesische Quadrat der Farbmenge von t ist. Für eine Transitionsfarbnummer j ist $D(j)$ die Nummer des entsprechenden Diagonalelements.

Weitere Funktionen werden nicht vorgegeben.

Die Eingabe der Prä– bzw. Post–Bögen wird durch `<Q>` als Antwort auf die Anforderung einer Platznummer beendet.

Falls Sie eine Zeitbewertung als Option gesetzt haben (siehe Kapitel 2.5.2 auf Seite 20), wird je nach Bewertungsart und bewertetem Netzelement jeweils zusätzlich eine initiale Uhrenstellung abgefragt (außer bei Zeitoption Dauer für Plätze).

Für das Netz 2 der fünf speisenden Philosophen sieht der Eingabedialog für die Transitionen und Bögen wie folgt aus (wieder leicht gekürzt):

```
Give next the transitions and the arcs.

transition nr.    1  name =    take_left
Colour set to be
read from the terminal...............................R
copied from an existing node.........................C
constructed as square CxC of a known colour set.......S
constructed as product CxC1 of known colour sets......P > C
place nr. 2.has_left
 colour    1: phil1             fact? Y/N: N
 colour    2: phil2             fact? Y/N: N
 colour    3: phil3             fact? Y/N: N
 colour    4: phil4             fact? Y/N: N
 colour    5: phil5             fact? Y/N: N
Pre-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 1.take_left        REMOVES from
place nr.     1.table
functions:  (I)dentity,
expression: 1*(I+4)
place nr.     Q

Post-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 1.take_left        SENDS to
place nr.     2.has_left
functions:  (I)dentity,
expression: 1*(I)
place nr.     Q


transition nr.     2 name =    take_right
Colour set to be
.....................................................
copied from an existing node.........................C > C
............Colour set C to be taken from a place? Y/N N
transition nr. 1.take_left
 colour    1: phil1             fact? Y/N: N
 colour    2: phil2             fact? Y/N: N
 colour    3: phil3             fact? Y/N: N
 colour    4: phil4             fact? Y/N: N
 colour    5: phil5             fact? Y/N: N
Pre-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 2.take_right       REMOVES from
place nr.     1.table
functions:  (I)dentity,
expression: 1*(I)
place nr.     Q

Post-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 2.take_right       SENDS to
place nr.     3.has_right
functions:  (I)dentity,
expression: 1*(I)
place nr.     Q


transition nr.     3 name =    start_eat
Colour set to be
.....................................................
copied from an existing node.........................C > C
............Colour set C to be taken from a place? Y/N N
transition nr. 1.take_left
 colour    1: phil1             fact? Y/N: N
```

```
 colour    2: phil2             fact? Y/N: N
 colour    3: phil3             fact? Y/N: N
 colour    4: phil4             fact? Y/N: N
 colour    5: phil5             fact? Y/N: N
Pre-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 3.start_eat        REMOVES from
place nr.     2.has_left
functions:  (I)dentity,
expression: 1*(I)
place nr.     3.has_right
functions:  (I)dentity,
expression: 1*(I)
place nr.     Q

Post-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 3.start_eat        SENDS to
place nr.     4.is_eating
functions:  (I)dentity,
expression: 1*(I)
place nr.     Q


transition nr.     4 name =    put_back
Colour set to be
.......................................................
copied from an existing node.........................C > C
.............Colour set C to be taken from a place? Y/N N
transition nr. 1.take_left
 colour    1: phil1             fact? Y/N: N
 colour    2: phil2             fact? Y/N: N
 colour    3: phil3             fact? Y/N: N
 colour    4: phil4             fact? Y/N: N
 colour    5: phil5             fact? Y/N: N
Pre-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 4.put_back         REMOVES from
place nr.     4.is_eating
functions:  (I)dentity,
expression: 1*(I)
place nr.     Q

Post-arc input mode: (C)opy , (E)xpression, by (H)and > E
Firing the transition 4.put_back         SENDS to
place nr.     1.table
functions:  (I)dentity,
expression: 1*(I)+ 1*(I+4)
place nr.     Q


transition nr.     Q
```

2.3.4. Dateiformate

In diesem Abschnitt finden Sie Dateiformatbeschreibungen in erweiterter Backus-Nauer-Form (EBNF). Die Zeichen `[ ]` stehen für eine Option, `{ }` für eine Wiederholung, `|` für eine Alternative, `" "` für eine Zeichenkette, `< >` für eine syntaktische Einheit und `::=` ist das Ersetzungszeichen. Einzige Besonderheit ist die Zeichenkette `"<cr>"`, die für ein Zeilenendezeichen steht. Zeilenumbrüche in der Darstellung haben keine Bedeutung, sie dienen nur der Übersichtlichkeit.

Die Beschreibungen dienen als eine Schnittstelle, die es Ihnen ermöglicht, beliebige (grafische) Editoren anzuschließen, sofern Sie die Formate so gut kennen, daß Sie ein Programm zur Transformation in das INA–Dateiformat schreiben können.

Syntax der .pnt–Datei

Die `.pnt`–Netzdatei besteht aus drei Abschnitten, die durch das `@`–Zeichen getrennt sind. Die Abschnitte enthalten Informationen über die Netzstruktur, die Plätze bzw. die Transitionen. Namen bestehen in der Regel aus 16 Buchstaben, die gegebenenfalls mit Leerzeichen aufgefüllt werden. Erlaubt sind auch einige Sonderzeichen wie z.B. `"_"`. Die maximale Größe von Zahlen ist rechnerabhängig.

```
 <pnt-file> ::= <netheader> "<cr>"
                <netstruct> "<cr>"
                "@<cr>"
                <placedata> "<cr>"
                "@<cr>"
                <transdata> "<cr>"
                "@<cr>"

<netheader> ::= "P   M   PRE,POST  NETZ " <netid>

    <netid> ::= <netnr> [ ":" <netname> ]

    <netnr> ::= <number>

  <netname> ::= <name>

<netstruct> ::= <placedef>
                { "<cr>" <placedef> }

 <placedef> ::= { "  " <placenr> " " <tokens> "    " [ <prelist> ] [ ", " <postlist> ]

  <placenr> ::= <number>

   <tokens> ::= <number>

  <prelist> ::= { <transnr> [ ": " <arcmult> ] " " }

 <postlist> ::= { <transnr> [ ": " <arcmult> ] " " }

  <transnr> ::= <number>

  <arcmult> ::= <number>

<placedata> ::= "place nr.             name capacity time"
                { "<cr>       " <placenr> ": " <placename> "      " <capacity> <time> }

<placename> ::= <name>

 <capacity> ::= " " <nr> | "oo"

     <time> ::= <number>

<transdata> ::= "trans nr.             name priority time"
                { "<cr>       " <transnr> ": " <transname> "      " <priority>
                <time> [ " fact" ] }

<transname> ::= <name>
```

```
 <priority> ::= <number>
```

Beispiel für eine .pnt–Datei

Hier finden Sie die Datei des Netzes 1 „Drei Programmierer teilen sich zwei Terminals“:

```
P   M   PRE,POST  NETZ 1:3_Prog_2_Term
  0 2     4: 2 5 6, 1: 2 2 3
  1 0     1, 4
  2 0     2, 5
  3 0     3, 6
  4 1     4, 1
  5 1     5, 2
  6 1     6, 3
@
place nr.            name capacity time
       0: Terminal_frei          oo    0
       1: Prog1_am_Term          oo    0
       2: Prog2_am_Term          oo    0
       3: Prog3_am_Term          oo    0
       4: Prog1_in_Pause         oo    0
       5: Prog2_in_Pause         oo    0
       6: Prog3_in_Pause         oo    0
@
trans nr.            name priority time
       1: login_Prog1             0    0
       2: login_Prog2             0    0
       3: login_Prog3             0    0
       4: logout_Prog1            0    0
       5: logout_Prog2            0    0
       6: logout_Prog3            0    0
@
```

Syntax der .cnt–Datei

Die `.cnt`–Netzdatei besteht aus der entfalteten Netzstruktur, die die Platzfarben, deren Anfangsmarkierung und die Verbindungen mit den Transitionsfarben beschreibt und den Platz– bzw. Transitionsfarben. Bis zu diesem Punkt findet sich in der Syntax kein Unterschied zu einer `.pnt`–Datei. Es folgt jetzt in einem vierten Abschnitt die Faltungsinformation, d.h. die Information, welche gefärbten Plätze und Transitionen es gibt, und welche Farben diese haben.

```
 <cnt-file> ::= <pnt-file>
                <aggregat> "<cr>"
                "@<cr>"

 <aggregat> ::= "AGGREGATION:<cr>"
                <placeaggr> "<cr>"
                "@<cr>"
                <transaggr>

<placeaggr> ::= "places:<cr>"
                <placecol>
                { "<cr>" <placecol> }

 <placecol> ::= "    " <placenr>:<placename> " " <placenr> { " " <placenr> }
```

```
<transaggr> ::= "transitions:<cr>"
                <transcol>
                { "<cr> <transcol> }

 <transcol> ::= "    " <transnr>:<transname> " " <transnr> { " " <transnr> }
```

Die dargestellte Syntax ermöglicht eine Kompatibilität der Netzdateien; es ist also möglich, `.pnt`–Dateien zu falten und `.cnt`–Dateien zu entfalten, d.h. die Faltungsinformation zu ignorieren (siehe dazu Kapitel 2.5.1 auf Seite 20).

Beispiel für eine .cnt–Datei

Hier finden Sie die Datei des gefärbten Netzes des Beispiels 2: „Fünf speisende Philosophen“

```
P   M   PRE,POST  NETZ 2:Dinner
  1 1     16 17, 2 6
  2 1     17 18, 3 7
  3 1     18 19, 4 8
  4 1     19 20, 5 9
  5 1     16 20, 1 10
  6 0     1, 11
  7 0     2, 12
  8 0     3, 13
  9 0     4, 14
 10 0     5, 15
 11 0     6, 11
 12 0     7, 12
 13 0     8, 13
 14 0     9, 14
 15 0     10, 15
 16 0     11, 16
 17 0     12, 17
 18 0     13, 18
 19 0     14, 19
 20 0     15, 20
@
place nr.             name capacity time
       1: fork1                 oo    0
       2: fork2                 oo    0
       3: fork3                 oo    0
       4: fork4                 oo    0
       5: fork5                 oo    0
       6: phil1                 oo    0
       7: phil2                 oo    0
       8: phil3                 oo    0
       9: phil4                 oo    0
      10: phil5                 oo    0
      11: phil1                 oo    0
      12: phil2                 oo    0
      13: phil3                 oo    0
      14: phil4                 oo    0
      15: phil5                 oo    0
      16: phil1                 oo    0
      17: phil2                 oo    0
      18: phil3                 oo    0
      19: phil4                 oo    0
      20: phil5                 oo    0
@
```

```
trans nr.            name priority time
       1: phil1              0    0
       2: phil2              0    0
       3: phil3              0    0
       4: phil4              0    0
       5: phil5              0    0
       6: phil1              0    0
       7: phil2              0    0
       8: phil3              0    0
       9: phil4              0    0
      10: phil5              0    0
      11: phil1              0    0
      12: phil2              0    0
      13: phil3              0    0
      14: phil4              0    0
      15: phil5              0    0
      16: phil1              0    0
      17: phil2              0    0
      18: phil3              0    0
      19: phil4              0    0
      20: phil5              0    0
@
AGGREGATION:
places:
    1:table            1     2     3     4     5
    2:has_left         6     7     8     9    10
    3:has_right       11    12    13    14    15
    4:is_eating       16    17    18    19    20
@
transitions:
    1:take_left        1     2     3     4     5
    2:take_right       6     7     8     9    10
    3:start_eat       11    12    13    14    15
    4:put_back        16    17    18    19    20
@
```

2.4. Die Struktur einer INA–Session

2.4.1. Aufruf und Beenden von INA

Zum Starten des Programms gehen Sie in das Verzeichnis, in dem sich das zu analysierende Netz befindet und in dem die erstellten Dateien abgelegt werden sollen und geben `ina` am Eingabeprompt ein.

Sollte sich im Verzeichnis die Datei `OPTIONS.ina` befinden, so wird diese eingelesen. Die Datei `COMMAND.ina` dagegen wird nur abgearbeitet, wenn Sie die Frage `Same procedure as last time?` (erscheint in diesem Fall beim Start) mit `<Y>` beantworten. Die gespeicherten Kommandos der letzen Sitzung werden dann wiederholt, bis Sie `<H>` für `halt` eingeben, oder alle Aktionen ausgeführt worden sind.

Gibt es keine `COMMAND.ina` oder Sie haben die Abfrage mit `<N>` beantwortet, so erscheint das Hauptmenü auf dem Bildschirm und die eingestellten Optionen werden angezeigt:

```
>>>>>>>>>>>>  Welcome to the Integrated Net Analyzer! <<<<<<<<<<<<
Version 1.7;                            Peter Starke, Berlin 1996
```

```
Current net options are:
    token  type: black        (for Place/Transition nets)
    time option: no times
    firing rule: normal
    priorities : not to be used
    strategy   : single transitions
    line length: 80

Do You want to
    edit ? .....................................E
    fire ? .....................................F
    analyse ? ..................................A
    reduce ? ...................................R
    read the session report ? ..................S
    delete the session report ? ................D
    change options ? ...........................O
    quit ? .....................................Q
 choice >
```

Durch Tippen der angezeigten Buchstaben wird die entsprechende Funktion ausgewählt.

Zum Beenden eines Kommandos oder eines Menüs kann in den meisten Fällen `<Q>` eingegeben werden. An vielen Stellen können so auch laufende Berechnungen vorzeitig abgebrochen werden.

Im Hauptmenü wird durch Eingabe von `<Q>` das Programm beendet. Befindet sich im internen Speicher noch ein Netz, so wird die erste Frage von INA sein, ob Sie dieses noch speichern möchten oder ob es vergessen werden kann: `Forget the net?` Außerdem werden Sie gefragt, ob die eingegebenen Kommandos gespeichern werden sollen: `Save the commands to COMMAND.ina?`. Wenn ja, dann stehen diese für einen weiteren Durchlauf zur Verfügung. INA wird Sie noch fragen, ob Sie den aktuellen Session–Report umbenennen wollen, um ihn vor überschreiben zu schützen: `Rename SESSION.ina?` Bei Eingabe von `<Y>` wird ein `Session output file` angefordert, eine Datei, die noch nicht existieren darf.

2.4.2. Der Session–Report

Der Session–Report ist eine ASCII–Datei, die alle Analyseergebnisse und Folgerungen enthält. Im Hauptmenü werden zwei Menüpunkte angeboten, um einerseits den Session–Report anzuzeigen und andererseits zu löschen.

`<S>     read the session report`

Die Datei `SESSION.ina` wird seitenweise auf den Bildschirm geholt, geblättert wird durch Drücken einer beliebigen Taste. Beim Lesen der Datei wird der Seiten-Modus durch `<Q>` abgeschaltet (der Inhalt wird dann ohne anzuhalten ausgegeben), ein weiteres `<Q>` bricht das Lesen ab. Nach Beendigung des Lesens befinden Sie sich wieder im Hauptmenü.

`<D>     delete the session report`

Die Datei `SESSION.ina` wird gelöscht. Danach erscheint ebenfalls das Hauptmenü. Sollten Sie danach erneut `<S>` eingeben, so sigalisiert Ihnen INA, daß der Anfangsteil gelöscht worden ist: `INA session report (first part deleted)`.

Um den Session–Report während eine Analyse zu lesen, müssen Sie also immer zuerst ins Hauptmenü zurückgehen.

Durch Setzen von Schreiboptionen können Sie vermeiden, daß diese Datei zu umfangreich wird (z.B. vermeiden, daß alle erreichbaren Zustände protokolliert werden).

2.4.3. Analyse mehrerer Netze nach dem gleichen Schema

Durch Verwendung der Datei `COMMAND.ina` ist es möglich, mehrere Netze auf die gleiche Art zu analysieren. Dies soll an einem Beispiel verdeutlicht werden. Angenommen Sie möchten die Dateien `netz01.pnt`, `netz02.pnt` und `netz03.pnt` untersuchen. Dann können Sie folgende Schritte durchführen:

1. Umbenennung des ersten Netzes `netz01.pnt` in `current.pnt`.
2. Aufruf von INA und Analyse von `current.pnt` mit den gewünschten Kommandos.
3. Beenden der Session unter Speichern von `COMMAND.ina`. Eventuell dabei `SESSION.ina` umbenennen.
4. Umbenennung von `current.pnt` rückgängig machen (für das erste Netz `netz01.pnt`) und das nächste Netz (also `netz02.pnt`) in `current.pnt` umnennen.
5. Aufruf von INA und die Frage `Same procedure as last time?` mit `<Y>` beantworten. Das Netz `current.pnt` wird nun genauso wie das vorhergehende Netz analysiert, ohne daß ein neues Kommando eingegeben werden muß. Danach eventuell wieder `SESSION.ina` umbenennen.

Die beiden letzen Schritte können für alle Netze, die Sie untersuchen möchten, wiederholt werden (im Beispiel also noch `netz03.pnt`).

2.5. Optionen

Gleich nach dem Start werden im Hauptmenü die aktuellen Optionen angezeigt. Durch Eingabe von `<O>` können diese verändert werden. So informieren Sie INA, Netze welchen Typs Sie bearbeiten möchten.

Der Reihe nach werden nun die Markenart, die Zeitoption, die Schaltregel, die Benutzung von Prioritäten, die Schaltstrategie und die Zeilenlänge abgefragt. Optionen, die Sie bestehen lassen wollen, können Sie durch Eingabe von `<cr>` bestätigen. Sollte bei einem Punkt nur eine Auswahlmöglichkeit bestehen, so geht die Abfrage sofort zur nächsten Option weiter. Die Standardoptionen, die Sie nach einem Erststart des Programms vorfinden, sind:

```
token  type: black        (for Place/Transition nets)
time option: no times
firing rule: normal
priorities : not to be used
strategy   : single transitions
line length: 80
```

Alle Optionseinstellungen, die Sie vornehmen, werden in der Datei `OPTIONS.ina` gespeichert (siehe im Kapitel 2.3.1 auf Seite 5).

Im folgenden werden alle möglichen Einstellungen einzeln vorgestellt.

2.5.1. Markenarten

`token type`

Sie können mit dieser Option die Beschaffenheit der Marken bestimmen.

`<B>     black (indistinguishable) tokens`

Hiermit wählen Sie die Arbeit mit Platz/Transitionsnetzen an. Die Markierung des Netzes besteht aus schwarzen ununterscheidbaren Marken [Sta90, Definition 2.2 (23)].

`<C>     coloured tokens`

Mit dieser Einstellung arbeiten Sie mit gefärbten Netzen. Sie haben die Möglichkeit, mit farbigen, d.h. unterscheidbaren Marken zu arbeiten [Sta90, Kapitel 20 (222–235)]. Beachten Sie: Gefärbte Netze können mit INA nicht reduziert werden!

Sollten Sie die Markenart eines gefärbten Netzes auf ununterscheidbare Marken gesetzt haben, so fragt INA `Forget the colour structure?` und bei positiver Antwort werden die Farben gelöscht [Sta90, Definition 20.3 (226)]. Vorsicht: Dabei können Informationen über das Netz verloren gehen!

Ändern Sie dagegen die Markenart umgekehrt, erscheint die Frage, ob Sie das Netz falten möchten: `Fold the Net?` Bei Eingabe von `<Y>` müssen Sie angeben, wie INA dabei vorgehen soll. Der Tabelle können Sie die Auswahlmöglichkeiten entnehmen [Sta90, Definition 20.2 und Bemerkung (225)]:

Faltungsart	nötige Eingabe	Wirkung
`1-place;1-transition CPN`	`<A>`	maximale Faltung
`by hand`	`<B>`	nutzerbestimmte Faltung
`trivially coloured CPN`	`<C>`	keine Faltung

2.5.2. Zeitoption

`time option`

Hier können Sie zwischen verschiedenen zeitbewerteten Netzarten wählen [Sta90, Kapitel 17 (192–196)]:

`<N>     no times`

Mit dieser Auswahl findet keine Zeitbewertung statt. Die Abfrage geht beim nächsten Punkt weiter.

`<D>     durations`

Mit einer Dauer können Plätze `<P>` (als Verweildauer für Marken) und Transitionen `<T>` (als Schaltdauer) belegt werden [Sta90, Kapitel 18 (197–208)].

`<I>     intervals`

Mit einem Intervall können Transitionen `<T>` (als früheste und späteste Schaltzeit nach der Konzessionierung) oder Bögen `<A>` (als Durchlaßfenster) belegt werden [Sta90, Kapitel 19 (209–221)].

Haben Sie sich für eine Zeitbewertung entschieden (d.h. `<D>` oder `<I>` eingegeben), so werden Sie anschließend aufgefordert anzugeben, welche Netzelemente mit Zeiten bewertet werden sollen. Die dazu nötigen Eingaben können Sie noch einmal der Tabelle entnehmen. `clocked net elements`

Zeitoption	zeitbewertete Elemente	nötige Eingabe	Wirkung
Dauer	Plätze	`<P>`	Verweildauer
	Transitionen	`<T>`	Schaltdauer
Intervalle	Transitionen	`<T>`	eft/lft
	Bögen	`<A>`	Durchlaßfenster

Abhängig von der gewählten Zeitoption ändert sich die Zustandsdarstellung. Sie finden im Kapitel 4.2 auf Seite 33 einen Überblick mit Beispielen. Im Kapitel 4.4.2 auf Seite 37 finden Sie Bemerkungen zum Schaltverhalten zeitbewerteter Netze.

2.5.3. Schaltregel

Die Schaltregel gibt an, unter welchen Bedingungen eine Transition schalten darf und wie die Wirkung des Schaltens auf die Markierung ist. `firing rule`

In Abhängigkeit vom Netztyp und der Zeitbewertung stehen bis zu vier verschiedene Schaltregeln zur Auswahl:

`<N>     normal firing rule`

Bei der normalen Schaltregel hat eine Transition Konzession, wenn jeder Vorplatz mindestens so viele Marken enthält, wie die Vielfachheit des Bogens von diesem Vorplatz zur Transition beträgt. Ein Schalten der Transition bewirkt, daß von ihren Vorplätzen der Vielfachheit der Bögen entsprechende Markenzahlen abgezogen und auf ihre Nachplätze entsprechende Markenzahlen aufgebracht werden [Sta90, Definition 2.4 (25)].

Beachten Sie: Nur unter dieser Schaltregel ist Reduzieren möglich!

`<C>     normal firing rule with capacities`

Gegenüber der normalen Schaltregel ist die Schaltfähigkeit der Transitionen beim Schalten unter Berücksichtigung der Kapazitäten dahingehend eingeschränkt, daß durch das

Schalten auf den Plätzen vorgegebene Markenzahlen (Kapazitäten) niemals überschritten werden können. Dabei darf die Kapazität eines Platzes auch unendlich sein [Sta90, Definition 10.2 (102)].

Das Zeichen `oo` steht für eine unendliche Kapazität, wird am Terminal durch `<o>` eingegeben und entspricht einer rechnerabhängigen Konstanten.

Ein Platz, der schon bei der Anfangsmarkierung mehr Marken enthält als seine Kapazität ausmacht, wird als überladen (`overloaded`) gemeldet.

`<S>     safe firing rule`

Die sichere Schaltregel ist nur bei gewöhnlichen Netzen (d. h. alle Bögen haben die Vielfachheit 1) anwendbar. Hierbei hat eine Transition genau dann Konzession, wenn alle ihre Vorplätze markiert und keiner ihrer Nachplätze markiert ist. Plätze, die bei der Anfangsmarkierung mehr als eine Marke tragen, werden als überladen gemeldet.

`<V>     Valk's firing rule`

Bei einem selbstmodifizierenden Netz [Sta90, (106)] kann die Vielfachheit eines Bogens durch die aktuelle Markenzahl auf einem diesem Bogen zugeordneten Platz bestimmt sein. Ist diese Zahl Null, so ist der Bogen in diesem Moment praktisch nicht vorhanden.

Bei der Bestimmung der Vielfachheit eines Bogens bei der aktuellen Markierung m geht INA wie folgt vor: Ist die dem Bogen (bei der Netzeingabe) zugeordnete Zahl gleich 1, dann ist die Vielfachheit gleich 1; ist diese Zahl $k > 1$, dann wird geprüft, ob es einen Platz p mit der Nummer k gibt. In diesem Fall ist die Markierung $m(p)$ dieses Platzes die gesuchte Vielfachheit, sonst die Zahl k selbst.

2.5.4. Prioritäten

`obey priorities`

Hier können Sie entscheiden, ob beim Schalten Prioriäten beachtet (Eingabe von `<Y>`) oder übersehen werden sollen (Eingabe von `<N>`). Standardmäßig werden Prioritäten nicht berücksichtigt.

Prioritäten sind als nichtnegative ganze Zahlen den Transitionen zugeordnet. Die Prioritäten zu beachten, bedeutet daß im gegebenen Zustand nur die konzessionierten Transitionen schalten dürfen, die die höchste (d.h. größte) vorkommende Priorität haben [Sta90, (103)]. Die Anwendung von Prioritäten schränkt die Analysemöglichkeiten ein.

2.5.5. Schaltstrategie

`firing strategie`

INA kennt zwei Schaltstrategien. Je nach Einstellung der anderen Optionen sind aber Einschränkungen nötig, die automatisch berücksichtigt werden. Falls also im aktuellen Kontext nur eine Strategie möglich ist, wird INA diese anwählen und die Abfrage beim nächsten Punkt fortführen.

Hier nun die Auswahlmöglichkeiten:

`<T>     fire single transitions`

Dies ist die normale Schaltstrategie, die Transitionen einzeln schalten läßt.

`<S>     fire maximal steps`

Diese Strategie läßt immer eine größte Menge von nebenläufigen Transitionen schalten. Dadurch wird aber die Erreichbarkeit von Markierungen eingeschränkt.

2.5.6. Zeilenlänge

`line lenght`

Diese Angabe wird zur Formatierung der Ausgabedatei `SESSION.ina` benötigt und ist dadurch auch teilweise für den Bildschirm wirksam. Wenn Ihr Bildschirm oder Ihr Drucker mit mehr als 80 Spalten arbeitet, so ändern Sie den Wert entsprechend, so daß die Ausgaben (zum Beispiel der Zustandsgraph als Baum) besser lesbar werden (Maximalwert: 255 Spalten).

3. Editor

Der Editor ist ein alphanumerischer Editor, der alle Funktionen enthält, die zum Eingeben und Verändern eines Netzes, sowie zum Zusammenfügen von Netzen erforderlich sind. Die erstellten Netze können auf Dateien abgespeichert und von dort wieder eingelesen werden. Eine Beschreibung der Formate können Sie im Kapitel 2.3.4 ab Seite 13 finden.

Wenn Sie im Hauptmenü die Funktion `edit` durch Eingabe von `<E>` auswählen, erscheint ein den aktuellen Anfangsoptionen entsprechendes Editormenü. Die Abbildung zeigt das Menü bei Verwendung der Standardoptionen. Wenn Sie einen anderen Netztyp ausgewählt haben, werden zusätzliche Lösch- und Änderungsfunktionen angeboten. Sollte sich noch kein Netz im internen Speicher von INA befinden, ist das Funktionsangebot weiter eingeschränkt, es werden nur die dann sinnvollen Kommandos angezeigt.

```
Editor Menue:
Quit the editing process........................Q
Read a net from a File.........................RF
Read a net from the Terminal...................RT
Read a Node from the terminal..................RN
Read an Arc from the terminal..................RA
Write the net to a File........................WF
Write the net to the Terminal..................WT
Delete the net Totally.........................DT
Delete a Node..................................DN
Delete all Isolated nodes......................DI
Change Facts...................................CF
Change nUmbers.................................CU
Change Names...................................CN
Change Multiplicities..........................CM
Change the Token load (marking)................CT
Node Doubling..................................ND
Merge two nOdes................................MO
Merge two Nets ................................MN
Test Connectedness.............................TC
edit>
```

Die Kommandos können in drei Kategorien eingeteilt werden: Lese- und Schreibkommandos (beginnend mit `<R>` oder `<W>`), Lösch- und Änderungskommandos (beginnend mit `<D>` oder `<C>`) und Verschmelzungskommandos (beginnend mit `<M>`).

Im folgenden finden Sie eine Beschreibung aller im Editormenü möglichen Kommandos. Wichtig: Die Verfügbarkeit einiger Funktionen hängt von den aktuellen Netzoptionen ab!

Einige Funktionen haben nur einen Sinn, wenn INA ein Netz im internen Speicher hat. Sie erhalten sonst die Meldung `no net present`. Sollte das Netz nur aus Plätzen bzw. Transitionen bestehen, erhalten Sie entsprechende Meldungen, da das Netz unvollständig ist: `only places present` bzw. `only transitions present`.

Folgendes Kommando gehört zu keiner der oben genannten Kategorien, ist aber selbstverständlich verfügbar:

`<Q>      Quit the editing process`

Durch Eingabe von `<Q>` wird der Editor verlassen. Falls Sie ein unvollständiges Netz eingegeben haben, werden Sie zum Löschen aufgefordert: `Delete the fragmented net?`

3.1. Einlesen und Schreiben

INA bietet Kommandos zum Einlesen und Speichern von Netzen, Plätzen, Transitionen und Bögen:

`<R><F>      Read a net from a File`

Wenn INA schon ein Netz im internen Speicher hat, erscheint die Frage `Delete the old net?`. Da nur ein Netz im Speicher gehalten werden kann, kann bei Eingabe von `<N>` kein Netz eingelesen werden. Eingabe von `<Y>` bewirkt die Löschung des Netzes. Nun werden alle sich im aktuellen Verzeichnis befindenden Netzdateien angezeigt und INA fragt nach einem Dateinamen. Im Kapitel 2.2 finden Sie ab Seite 4 eine Beschreibung, wie Sie diesen eingeben können. Danach liest INA die angegebene Datei in den internen Speicher und geht in das Editormenü zurück. Bei der Dateianforderung ist es möglich durch Eingabe von `<esc>` in den Terminalmodus zu gelangen.

`<R><T>      Read a net from the Terminal`

Mit diesem Kommando ist es möglich, Netze direkt am Terminal einzugeben. Wenn INA schon ein Netz im internen Speicher hat, so wird analog wie im vorher beschriebenen Kommando reagiert.

Eine Beschreibung der Eingabe finden Sie im Kapitel 2.3.3 ab Seite 7.

`<R><N>      Read a Node from the terminal`

Mit diesem Kommando können Sie Knoten in ein (eventuell leeres) Netz eingeben. Zuerst werden Sie gefragt, ob Sie einen Platz oder eine Transition zufügen möchten. Danach erscheint die Eingabesyntax, die im Falle einer einzufügenden Transition wie folgt lautet:

```
Input syntax:
<nr>" "[ <prelist> ] [ "," <postlist> ] "<cr>"
 <prelist> ::= { <nr> [ ":"<mult> ]" " }
<postlist> ::= { <nr> [ ":"<mult> ]" " }
T>
```

Bei einem einzufügenden Platz erscheint die Syntax, die Sie bei der Beschreibung der Eingabe einer Platzzeile im Kapitel 2.3.3 auf Seite 7 finden. Dort finden Sie auch weitere Hinweise für die Platzeingabe.

Die Eingabe einer Transition erfolgt ganz analog, es wird nur keine Anfangsmarkierung erfragt und statt Listen von Transitionen werden Listen von Plätzen angefordert.

Existiert der definierte Knoten bereits, wird die alte Definition ersetzt. Wenn Sie einen nicht existierenden Knoten eingeben, wird dieser ohne Namen in das Netz eingefügt. Ebenso verhält es sich mit in der Definition vorkommenden neuen Knoten, die von INA automatisch eingefügt werden.

Bei dem Zufügen eines Knotens in ein gefärbtes Netz erfolgt die Eingabe analog zu der Beschreibung im Kapitel 2.3.3 ab Seite 8. Bitte beachten Sie: Bei einer einzufügenden Transition müssen alle Plätze existieren. INA gibt folgende Warnung aus: `Please, make sure that all places exist to which the new transition will be connected.`

`<R><A>      Read an Arc from the terminal`

Mit diesem Kommando können Sie einen neuen Bogen eingeben, löschen oder die Vielfachheit ändern. Als erstes werden die anliegende Transition und der anliegende Platz angefordert, dann die Vielfachheit. Als letztes müssen Sie die Richtung des Bogens angeben: `Place to Transition?` Die Antwort `<Y>` führt zu einem Bogen von einem Platz zu einer Transition, die Antwort `<N>` zu einem Bogen von einer Transition zum Platz. Ein bestehender Bogen wird überschrieben, mit der Vielfachheit 0 können Sie ihn löschen. Bei einem gefärbten Netz erfolgt die Eingabe eines Bogens wie bei der Netzeingabe (siehe Seite 11 im Kapitel 2.3.3).

`<W><F>      Write the net to a File`

Mit diesem Kommando können Sie das Netz im internen Speicher auf eine Datei ablegen. Durch Eingabe von `<Y>` auf die Frage `Select places` können Sie einzelne Plätze angeben, die ein zu speicherndes Teilnetz erzeugen. Nach der Eingabe eines Dateinamens wird das Netz bzw. das Teilnetz auf diese Datei geschrieben.

Durch die Auswahl einer Platzmenge haben Sie die Möglichkeit, das dadurch erzeugte Teilnetz später separat zu analysieren.

`<W><T>      Write the net to the Terminal`

Mit diesem Kommando können Sie verschiedene Informationen über das Netz im internen Speicher auf den Bildschirm ausgeben:

```
Information on the net ('N'), the places ('P'),
      the transitions ('T')  or quit    ('Q')?
```

Anhand des Netzes 1 werden die verschiedenen Ausgaben dieser Kommandos gezeigt.

Das Unterkommando `<N>` zeigt die Netznummer und den Netznamen an und gibt einen Überblick über die Anzahlen der verwendeten Plätze und Transitionen und die Intervalle, in denen die Knotennummern liegen:

```
Netnr:   1    Netname: 3_Prog_2_Term
Places:                7   [0,6]
Transitions:           6   [1,6]
```

Mit dem Unterkommando <P> werden die Plätze des aktuellen Netzes ausgegeben. Sie haben die Möglichkeit, bestimmte Plätze auszuwählen. Angezeigt werden: Platznummer, Platzname, Markierung, Namen und Nummern der Vortransitionen mit Vielfachheiten und analoges für die Nachtransitionen:

```
place nr.      0.Terminal_frei      load:  2
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
logout_Prog1              4:   2               1:   2   login_Prog1
logout_Prog2              5                    2        login_Prog2
logout_Prog3              6                    3        login_Prog3

place nr.      1.Prog1_am_Term      load:  0
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
login_Prog1               1                    4        logout_Prog1

place nr.      2.Prog2_am_Term      load:  0
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
login_Prog2               2                    5        logout_Prog2

place nr.      3.Prog3_am_Term      load:  0
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
login_Prog3               3                    6        logout_Prog3

place nr.      4.Prog1_in_Pause     load:  1
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
logout_Prog1              4                    1        login_Prog1

place nr.      5.Prog2_in_Pause     load:  1
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
logout_Prog2              5                    2        login_Prog2

place nr.      6.Prog3_in_Pause     load:  1
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
logout_Prog3              6                    3        login_Prog3
```

Mit dem Unterkommando <T> werden analog zur vorherigen Beschreibung die Transitionen ausgegeben:

```
transition nr.      1.login_Prog1
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
Terminal_frei             0:   2               1        Prog1_am_Term
Prog1_in_Pause            4

transition nr.      2.login_Prog2
--------------------------------------------------------------------------
PREname                  nr: mlt              nr: mlt   POSTname
Terminal_frei             0                    2        Prog2_am_Term
Prog2_in_Pause            5

transition nr.      3.login_Prog3
--------------------------------------------------------------------------
```

```
PREname               nr: mlt              nr: mlt   POSTname
Terminal_frei          0                    3        Prog3_am_Term
Prog3_in_Pause         6

transition nr.      4.logout_Prog1
-------------------------------------------------------------------------
PREname               nr: mlt              nr: mlt   POSTname
Prog1_am_Term          1                    0:   2   Terminal_frei
                                            4        Prog1_in_Pause

transition nr.      5.logout_Prog2
-------------------------------------------------------------------------
PREname               nr: mlt              nr: mlt   POSTname
Prog2_am_Term          2                    0        Terminal_frei
                                            5        Prog2_in_Pause

transition nr.      6.logout_Prog3
-------------------------------------------------------------------------
PREname               nr: mlt              nr: mlt   POSTname
Prog3_am_Term          3                    0        Terminal_frei
                                            6        Prog3_in_Pause
```

Mit `<Q>` beenden Sie das Besichtigen des Netzes.

3.2. Löschen und Ändern

INA bietet Kommandos zum Löschen von Netzen und Netzelementen. Teilweise sind diese Kommandos nur für gefärbte Netze verfügbar.

`<D><T>     Delete the net Totally`

Das gesamte Netz wird ohne weitere Rückfrage vollständig aus dem Speicher entfernt.

`<D><N>     Delete a Node`

Mit diesem Kommando können Sie einen Platz oder eine Transition des aktuellen Netzes löschen. Wenn Sie die Abfrage `delete a place` mit `<N>` beantworten, können Sie die Nummer der zu löschenden Transition angeben, ansonsten die eines Platzes. Der ausgewählte Knoten und alle anliegenden Kanten werden ohne weitere Rückfrage entfernt.

`<D><C>     Delete a Colour`

Bei einem gefärbten Netz können Sie mit diesem Kommando eine Farbe eines Netzknotens löschen, eventuell auch die Diagonale (siehe Seite 11).

`<D><I>     Delete all Isolated nodes/colours`

Mit diesem Kommando können Sie unbenutze (isolierte) Knoten bzw. Farben löschen, die beim Editieren eines Netzes entstanden sind. Da solche Netze von INA nicht analysiert werden können, empfiehlt es sich, daß Sie dieses Kommando vor dem Abspeichern bzw. vor dem Ende des Editierens ausführen.

`<D><A>     Delete all input-- or output--Arcs`

Mit diesem Kommando können Sie bei einem gefärbten Netz alle Eingangs– bzw. Ausgangsbögen eines Netzknotens löschen.

`<C><F>     Change Facts`

Nach Eingabe dieses Kommandos sehen Sie zuerst die aktuelle Faktenmenge (`currently the set of facts is`) in Form einer Liste der als Fakten spezifizierten Transitionen.

Falls diese Menge nicht leer ist, können Sie durch Beantwortung der Frage `Delete facts?` mit `<Y>` einzelne Transitionen aus dieser Menge entfernen: `Please, give the transitions not to remain facts`. Mit `<Q>` können Sie das Entfernen beenden, die jetzt aktuelle Faktenmenge wird ausgegeben.

Als nächstes können Sie durch Beantwortung der Frage `Add facts` mit `<Y>` einzelne Transitionen in die Faktenmenge aufnehmen: `Please, give the transition colours to become facts`. Mit `<Q>` können Sie das Einfügen beenden, die Faktenmenge wird erneut ausgegeben und in den Session–Report geschrieben.

Für gefärbte Netze können analog Farben von Transitionen als Fakten bestimmt werden.

`<C><P>     Change Priorities`

Dieses Kommando ist nur aktiv, wenn Sie die Arbeit mit Prioritäten als Option gesetzt haben (siehe Kapitel 2.5.4 auf Seite 22). Die aktuellen Prioritäten werden angezeigt. Sie können entweder eine Datei mit der Endung `.pri` einlesen, die die neuen Prioritäten enthält oder durch Eingabe von `<esc>` diese einzeln überschreiben. Mit `<cr>` können Sie alte Werte übernehmen, die Sie versehentlich zum Ändern ausgewählt haben.

`<C><C>     Change Capacities`

Dieses Kommando ist nur aktiv, wenn Sie die Kapazitätsschaltregel als Option gesetzt haben (siehe Kapitel 2.5.3 auf Seite 21). Die aktuellen Kapazitäten werden angezeigt. Sie können entweder eine Datei mit der Endung `.cap` einlesen, die die neuen Kapazitäten enthält oder durch Eingabe von `<esc>` diese einzeln überschreiben. Mit `<O>` können Sie eine unendliche Kapazität `oo` angeben. Mit `<cr>` können Sie alte Werte übernehmen, die Sie versehentlich zum Ändern ausgewählt haben.

`<C><I>     Change Initial times`

Dieses Kommando ist nur aktiv, wenn Sie eine Zeitbewertung, die nicht Dauer für Plätze ist, als Option gesetzt haben (siehe Kapitel 2.5.2 auf Seite 20). Die aktuellen initialen Zeiten werden angezeigt. Sie können entweder eine Datei mit der Endung `.itm` einlesen, die die neuen initialen Zeiten enthält oder durch Eingabe von `<esc>` diese einzeln überschreiben. Mit `<cr>` können Sie alte Werte übernehmen, die Sie versehentlich zum Ändern ausgewählt haben.

`<C><U>     Change nUmbers`

Mit diesem Kommando können Sie die Nummern der Netzelemente und des Netzes ändern. Folgendes Menü wird angeboten:

```
Do you want to
change the net number..............................N
change a place number..............................P
add a span to all place numbers....................A
renumber the places starting with 1................R
change a transition number.........................T
add a span to all transition numbers...............S
renumber the transitions starting with 1...........F
```

Sie können die Nummern einzelner Plätze bzw. Transitionen ändern, aber auch eine Spanne zu allen Platz– bzw. Transitionsnummern addieren. Diese Operation dient dazu, Platz– bzw. Transitionsmengen disjunkt zu den Platz– bzw. Transitionsmengen von Netzen zu machen, mit denen das aktuelle Netz verschmolzen werden soll. Eine fortlaufende (dichte) Nummerierung wird ebenfalls angeboten, die für die Effektivität einiger Algorithmen von Vorteil ist.

`<C><N>      Change Names`

Dieses Kommando dient zum Eingaben und Ändern von Namen.

```
Do You want to change the name of
the net....................................N
a place....................................P
a transition...............................T
a place colour.............................C
a transition colour........................F
```

`<C><M>      Change Multiplicities`

Dieses Kommando arbeitet genauso wie das Kommando `<R><A>` (siehe Kapitel 3.1 auf Seite 27).

`<C><T>      Change the Token load (marking)`

Mit diesem Kommando können Sie die Anfangsmarkierung ändern. Die aktuelle Anfangsmarkierung wird angezeigt. Sie können entweder eine Datei mit der Endung `.mar` einlesen, die die neue Anfangsmarkierung enthält oder durch Eingabe von `<esc>` diese einzeln überschreiben. Mit `<cr>` können Sie alte Werte übernehmen, die Sie versehentlich zum Ändern ausgewählt haben.

3.3. Verschmelzen

`<N><D>      Node Doubling`

Dieses Kommando verdoppelt einen Knoten, d.h. es führt für einen gegebenen Knoten einen identischen in das Netz ein. Dies kann z.B. für eine Verfeinerung des Netzes nützlich sein. Als erstes müssen Sie die zu verdoppelnde Knotenart eingeben: `Double a place (P), a transition (T) or quit (Q)`. Nach Eingabe von `<P>` bzw. `<T>` können Sie die Nummer des zu verdoppelnden Knotens eingeben. Danach wird eine neue Knotennummer abgefragt: `Duplicate node number`. Wenn Sie die Nummer eines existierenden Knotens eingeben, so meldet INA `is not free` und Sie können eine andere Nummer wählen.

Unabhängig von der Auswahl wird der Namen des verdoppelten Knotens gebildet aus der Kennzeichnung `duplicate` gefolgt von der ursprünglichen Knotennummer. Mit `<Q>` beenden Sie dieses Kommando.

`<M><O>      Merge two nOdes`

Dieses Kommando verschmilzt zwei Knoten. Als erstes müssen Sie eingeben, welche Knotenart Sie verschmelzen möchten: `Merge two places?` Nach Eingabe von `<Y>` können Sie die Nummern der zu verschmelzenden Plätze eingeben. Der Platz mit der höheren Nummer wird gelöscht, die Markierungen werden addiert und die Bögen umgesetzt. Analoges können Sie durch Eingabe von `<N>` auf obige Frage auch für Transitionen durchführen.

`<M><N>      Merge two Nets`

Mit diesem Kommando können Sie das Netz im internen Speicher von INA mit einem Netz verschmelzen, das in einer Datei abgelegt ist. Durch die Abfrage `Merge with a second net at the common places` können Sie die Art der Knoten bestimmen, an der die Netze verschmolzen werden sollen.

`<T><C>      Test Connectedness`

Es wird getestet, ob das aktuelle Netz zusammenhängend ist (siehe Kapitel 6.2.1 auf Seite 57). Gegebenenfalls können Sie die Zusammenhangskomponenten einzeln auf Dateien schreiben und später einzeln analysieren. Für einige Analysealgorithmen ist der Zusammenhang des Netzes eine Voraussetzung [Sta90, Definition 14.1 und Bemerkung (148)].

4. Schalten

Über eine nutzergeführte Ausführung des Netzes, d.h. Wanderungen durch den Erreichbarkeitsgraphen, können Sie sich eine Vorstellung des Netzverhaltens verschaffen und Knoten im Erreichbarkeitsgraphen näher untersuchen, an denen Unerwartetes geschieht (z.B. Konzessionierung eines Fakts oder Erreichbarkeit von toten Zuständen).

4.1. Grundlagen

Die Ausführung eines Netzes wird im Hauptmenü unter `<F>` angeboten. Falls noch kein Netz eingelesen wurde, erfragt INA zunächst ein Netz und anschließend gegebenenfalls die Zeitinformationen. Als nächstes können Sie die Schreiboptionen ändern und so Transitions– bzw. Farbnamen ausgeben lassen. Danach wird ein Menü angezeigt, das die verfügbaren Funktionen auflistet:

```
<R>: Reset to the initial state            <B>: Back to the previous state
<W>: Write the actual marking to a file
<S>: Show the stubborn set used by the stubborn reduced reachability graph
<A>: All minimal stubborn sets to be shown            <Q>: Quit computation
```

Vor jedem Schaltvorgang wird dieses Menü nocheinmal in einer verkürzten Form ausgegeben:

```
<r>eset | <b>ack | <w>rite | <s>tubborn set | <a>ll stubborn sets | <q>uit
```

4.2. Zustandsdarstellung

In den beiden folgenden Abschnitten finden Sie die für das ganze Programm gültige Beschreibung der Zustandsdarstellung von INA.

4.2.1. Zustandsdarstellung ohne Zeitbewertung

Wenn Sie ohne Zeitbewertung arbeiten (siehe Kapitel 2.5.2 auf Seite 20), so besteht die Zustandsinformation nur aus der jeweils aktuellen Markierung. Die Markierung wird in folgender Vektorform dargestellt:

```
PL:  1  1  1  1  1  2  2  2  2  2  3  3  3  3  3  4  4  4  4  4
MA:  1  0  1  1  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
```

In der Kopfzeile stehen die Platznummern (bei einem gefärbten Netz jede so oft, wie der Platz Farben hat), darunter die zugehörige Markenanzahlen. Das Beispiel zeigt eine Markierung des Netzes 2, bei der auf dem Platz 2 genau eine Marke, und zwar der Farbe 3, liegt.

4.2.2. Zustandsdarstellung unter Zeitbewertung

Falls Sie unter Zeitbewertung arbeiten, besteht die Zustandsinformation neben der Markierung noch aus einem die aktuelle Uhrenstellung beschreibenden Vektor.

In den folgenden Beispielen werden diese Vektoren für ein Platz/Transitionsnetz mit sechs Plätzen $(0,\ldots,5)$ und fünf Transitionen $(1,\ldots,5)$ aufgeführt.

Zeitoption Dauer für Plätze — Verweildauer

Die Uhren werden den Plätzen zugeordnet und zeigen die Zeit an, die noch vergehen muß, damit Marken ihre Verweilzeit abgesessen haben.

```
PL: 0 1 2 3 4 5
MA: 2 0 0 1 0 0
 2: 0 0 0 0 1 0
 4: 0 0 0 0 0 1
```

Bei diesem Zustand liegen zwei gültige Marke auf dem Platz 0, auf dem Platz 3 liegt eine gültige Marke. Die Plätze 4 und 5 enthalten je eine noch ungültige Marke, die dort noch 2 bzw. 4 Zeiteinheiten verweilen müssen.

Zeitoption Dauer für Transitionen — Schaltdauer

Die Uhren werden den Transitionen zugeordnet und zeigen die Zeit an, die noch vergehen muß, damit die Transition das Schalten beendet, d.h. Marken auf ihre Nachplätze liefert.

```
PL: 0 1 2 3 4 5
MA: 1 1 0 0 1 1
TR: 1 2 3 4 5
TC: 0 1 0 3 0
```

Bei diesem Zustand zeigt die Uhr der Transition 2 an, daß das Schalten dieser Transition noch einen Takt dauern wird. Entsprechend dauert das Schalten der Transition 4 noch drei Zeiteinheiten.

Zeitoption Intervalle für Transitionen — eft/lft

Die Uhren werden den Transitionen zugeordnet und zeigen die Zeit an, die seit der Konzessionierung vergangen sind.

```
PL: 0 1 2 3 4 5
MA: 0 1 1 0 0 1
TR: 1 2 3 4 5
TC: - 2 - 1 -
```

Bei diesem Zustand sind die Uhren der Transitionen 1, 3 und 5 abgestellt, weil nicht genügend Marken vorhanden sind, um eine dieser Transitionen zu schalten. Die Uhren der Transitionen 2 und 4 zeigen 2 bzw. 1 an, d.h. diese Transitionen sind auf Marken bezogen schaltfähig und zwar seit zwei bzw. einem Takt. Ob sie schalten können, hängt nur davon ab, ob ihre frühste Schaltzeit eft bereits erreicht ist.

Zeitoption Intervalle für Bögen — Durchlaßfenster

Die Uhren werden den Plätzen zugeordnet und zeigen die Zeit an, die die jüngste Marke auf diesem Platz bereits wartet.

```
PL: 0 1 2 3 4 5
MA: 1 0 1 1 0 1
PC: 0 0 5 6 0 0
```

Bei diesem Zustand sind die Marken auf den Plätzen 0 und 5 „frisch", d.h. mit diesem Zustand entstanden, während die Marken auf den Plätzen 2 und 3 bereits fünf bzw. sechs Takte darauf warten, daß sich ein Bogenfenster für sie öffnet.

4.3. Hilfsfunktionen

Dieser Abschnitt beschreibt einige Hilfsfunktionen.

`<Q>      Quit`

Mit dieser Taste können Sie das Schalten beenden und ins Hauptmenü zurückkehren, die ursprüngliche Markierung des Netzes wird automatisch wiederhergestellt. Längerdauernde Berechnungen können ebenso abgebrochen werden.

`<W>      Write the actual marking to a file`

Diese Funktion bietet Ihnen die Möglichkeit, die gerade erreichte Markierung in eine Datei zu speichern, um z.B. später einen kürzesten Pfad zu berechnen. Nach der Eingabe eines Dateinamens wird die aktuelle Markierung mit der Extension `.mar` geschrieben.

Die beiden folgenden Funktionen werden nur angeboten, wenn eine sture Reduktion (siehe Seite 66 im Kapitel 6.4.2) möglich ist. Zu Details der Implementation siehe [Pog94, Berechnung sturer Mengen].

`<S>      Show the stubborn set used by the stubborn red. reachab. graph`

Mit dieser Funktion können Sie sich die sture Menge anzeigen lassen, die bei der sturen Reduktion des Erreichbarkeitsgraphen von INA ausgewählt wird.

`<A>      All minimal stubborn sets to be shown`

Mit dieser Funktion werden alle minimalen sturen Mengen ausgegeben.

4.4. Schaltfunktionen

Dieser Abschnitt beschreibt die eigentliche Funktionalität der Schaltausführung.

4.4.1. Allgemeines

In jedem Zustand wird zuerst die aktuelle Markierung, gegebenenfalls die Liste konzessionierter Fakten sowie je nach Schaltstrategie die Liste der konzessionierten Transitionen bzw. Schritte ausgegeben:

```
 PL: 1
 MA: 1
enabled facts: 1,
enabled: 1.Fakt, 2.NoFakt,
Transition to fire >
```

Die Auswahl einer Transition bzw. eines Schrittes der angezeigten Schaltliste erfolgt durch Eingabe seiner Nummer. Die ausgewählte Transition bzw. der ausgewählte Schritt wird ausgeführt und der Folgezustand dargestellt. Bei gefärbten Netzen müssen Sie zusätzlich zur Transitionsnummer noch mit einem Semikolon abgetrennt die Schaltfarbe angeben. Wenn keine Transition Konzession hat meldet INA: `no transition has concession`. Falls Sie die Nummer einer nicht konzessionierten Transition bzw. eines Schrittes eingeben, so wird das Schalten mit der Meldung `no concession` abgelehnt.

`<cr>      fire the first element`

Durch Eingabe von `<cr>` wird immer das erste Element der angezeigten Schaltliste ausgewählt und geschaltet.

`<B>      Back to the previous state`

Mit der Taste `<B>` können Sie wiederholt einen Schaltvorgang rückgängig machen und so andere Auswahlen treffen.

`<R>      Reset to the initial state`

Mit dieser Funktion wird der Anfangszustand wiederhergestellt. Sie können so einen anderen Weg durch den Erreichbarkeitsgraphen inspizieren. Am Ende des Schaltens müssen Sie diese Funktion nicht aufrufen, da dies automatisch geschieht und so das Netz in den Anfangszustand zurückgesetzt wird.

4.4.2. Schalten unter Zeitbewertung

In den folgenden Unterabschnitten finden Sie die für das ganze Programm gültige Beschreibung des Schaltens unter Zeitbewertung.

Zeitoption Dauer für Plätze — Verweildauer

Netze, bei denen jedem Platz eine ganzzahlige Verweildauer zugeordnet ist, werden in maximalen Schritten ausgeführt. Jede Marke ist beim Eintreffen auf einem Platz zunächst ungültig, kann also zum Schalten nicht verwendet werden, und wird erst gültig, wenn die Verweilzeit dieses Platzes abgelaufen ist. Die Marken der Anfangsmarkierung sind beim Anfangszustand sämtlich gültig.

In solchen Netzen können transiente Zustände auftreten, das sind Zustände, bei denen durch die gültigen Marken keine Transition konzessioniert ist, aus denen aber durch Warten ein Zustand entsteht, bei dem genug gültige Marken vorhanden sind, um wenigstens eine Transition zu konzessionieren.

INA zeigt transiente Zustände nicht an, sondern setzt die Uhren automatisch weiter bis zum nächsten stabilen Zustand, in dem ein Schaltvorgang möglich ist: `Transitional period from previous state = 2` z.B. zeigt an, daß zwei Takte seit dem letzten Zustand vergangen sind, in denen keine Transition schalten konnte.

Wenn eine Transition schaltfähig ist, dann muß sie auch schalten. Diese Festlegung führt dazu, daß in jedem stabilen Zustand eine maximale Menge nebenläufig konzessionierter Transitionen, ein maximaler Schritt, ausgeführt wird. Sie können auswählen, welcher Schritt geschaltet werden soll.

Zeitoption Dauer für Transitionen — Schaltdauer

Transitionen, denen eine positive ganze Zahl als Schaltdauer zugeordnet ist, müssen schalten, sobald genügend Marken verfügbar sind. Wenn eine solche Transition t mit der Dauer d im Zeitpunkt τ schaltet, so werden die Eingabemarken im Zeitpunkt τ von ihren Vorplätzen abgezogen und ihre Ausgabemarken im Zeitpunkt $\tau + d$ auf ihren Nachplätzen verfügbar sein.

Eine Transition, die im Schalten begriffen ist, d.h. die Eingabemarken abgezogen, aber die Ausgabemarken noch nicht abgeliefert hat, kann in dieser Periode keinen zweiten Schaltvorgang beginnen, auch wenn noch genügend Marken auf ihren Vorplätzen liegen. Dementsprechend werden solche Netze in maximalen Schritten ausgeführt.

Auch bei solchen Schaltdauer–Netzen können transiente Zustände auftreten, d.h. Zustände, bei denen keine Transition schalten kann, aus denen aber durch Zeitverlauf ein Zustand entstehen, bei dem Aktionen möglich sind. Hier kann sogar der Anfangszustand transient sein, wenn die initiale Uhrenstellung nicht identisch Null ist, d.h. einige Transitionen sind mitten beim Schalten.

Wenn der Anfangszustand transient ist, zeigt INA den nächsten stabilen Zustand an: `Initial state is transient. Stabilisation time = 3` ist ein Beispiel für die Stabilisierung des Anfangszustandes. Im übrigen werden transiente Zustände durch Weiterstellen der Uhren und die damit verbundene Änderung der Markierung stabilisiert.

Zeitoption Intervalle für Transitionen — eft/lft

Jeder Transition ist ein ganzzahliges Intervall eft, lft zugeordnet. Die Uhr der Transition wird in Gang gesetzt in dem Moment, in welchem ihre Vorplätze ausreichend markiert sind. Die Transition darf schalten, sobald die von ihrer Uhr angezeigte Zeit mindestens gleich eft ist, sie muß schalten, wenn sie gleich lft ist, sofern sie nicht durch das Schalten anderer Transitionen die Konzession verliert. Das Schalten selbst kostet keine Zeit.

Wenn zwei Transitionen einen Vorplatz teilen, d.h. in einem statischen Konflikt stehen, und die eine der beiden schaltet, so wird die Uhr der anderen Transition abgestellt, wenn sie durch diesen Schaltvorgang die Konzession verliert, und sie wird auf Null zurückgestellt, wenn sie dabei Konzession behält.

Transiente Zustände sind Zustände, bei denen keine Transition schalten kann, aus denen aber durch Zeitverlauf ein Zustand entstehen kann, bei dem eine Transition schaltfähig ist. INA stabilisiert transiente Zustände automatisch.

Es gibt Zustände bei denen Transitionen schalten können, aber keine Transition schalten muß, weil ihre späteste Schaltzeit lft noch nicht erreicht ist. Das Weiterstellen aller Uhren um eine Einheit wird von INA durch eine Pseudotransition mit dem Namen `timetick` modelliert. Diese kann mit `<t>` ausgewählt werden. Die Markierung bleibt beim Schalten von `timetick` natürlich ungeändert.

Zeitoption Intervalle für Bögen — Durchlaßfenster

Jedem Bogen, der von einem Platz zu einer Transition führt, ist ein Zeitfenster eft, lft zugeordnet. Auf jedem Platz steht eine Uhr, die jeweils die Wartezeit der jüngsten Marke auf diesem Platz anzeigt. Diese Uhr wird also auf Null gestellt in dem Moment, wo eine Marke auf diesem Platz eintrifft, aber auch in dem Moment, wo eine Marke von diesem Platz abgezogen wird. Eine Transition kann schalten, wenn alle ihre Vorplätze ausreichend markiert sind und die Uhr jedes Vorplatzes eine Zeit anzeigt, die im Intervall eft, lft liegt, das dem Bogen von diesem Vorplatz zur Transition zugeordnet ist. Wenn eine Transition schalten kann, dann muß sie schalten; dementsprechend werden solche Netze in Schritten ausgeführt.Das Schalten der Transitionen erfolgt ohne Zeitverbrauch. INA stabiliert transiente Zustände automatisch.

4.5. Beispiel

Anhand des entfalteten Netzes des gefärbten Netzes 2 der speisenden Philosophen wird im folgenden ein Beispiel für das Schalten gezeigt. Sie sehen Bildschirmanzeigen und Ausschnitte aus der Session–Datei.

Zu Beginn wird der Anfangszustand und die Liste der darin konzessionierten Transitionen ausgegeben:

```
Fire
Current name options are:
     transition names to be written

 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
enabled:  1.phil1_take_left,  2.phil2_take_left,  3.phil3_take_left,
          4.phil4_take_left,  5.phil5_take_left,  6.phil1_take_right,
          7.phil2_take_right,  8.phil3_take_right,  9.phil4_take_right,
         10.phil5_take_right,
Transition to fire >
```

Durch Hintereinandereingabe von `<1>`, `<6>`, `<11>` und `<16>` können Sie einen kompletten Durchlauf des ersten Philosophen mit Aufnahme der linken und rechten Gabel, Essen und Abgabe der Gabeln ausführen. Danach befindet sich das Netz wieder im Anfangszustand.

```
Transition to fire > 1

fired transition:  1.phil1_take_left


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  1  1  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
enabled:  2.phil2_take_left,  3.phil3_take_left,  4.phil4_take_left,
          5.phil5_take_left,  6.phil1_take_right,  7.phil2_take_right,
          8.phil3_take_right,  9.phil4_take_right,
Transition to fire > 6

fired transition:  6.phil1_take_right


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  1  1  1  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0
enabled:  3.phil3_take_left,  4.phil4_take_left,  5.phil5_take_left,
          7.phil2_take_right,  8.phil3_take_right,  9.phil4_take_right,
         11.phil1_start_eat,
Transition to fire > 11

fired transition: 11.phil1_start_eat


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  1  1  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
enabled:  3.phil3_take_left,  4.phil4_take_left,  5.phil5_take_left,
          7.phil2_take_right,  8.phil3_take_right,  9.phil4_take_right,
         16.phil1_put_back,
Transition to fire > 16

fired transition: 16.phil1_put_back


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Ausgehend von dieser Situation wird durch wiederholte Eingabe von `<cr>` immer die erste Transition geschaltet, alle Philosophen nehmen die linke Gabel auf.

```
 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
enabled:  1.phil1_take_left,  2.phil2_take_left,  3.phil3_take_left,
          4.phil4_take_left,  5.phil5_take_left,  6.phil1_take_right,
          7.phil2_take_right,  8.phil3_take_right,  9.phil4_take_right,
```

```
        10.phil5_take_right,
Transition to fire >

fired transition:  1.phil1_take_left


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  1  1  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
enabled:  2.phil2_take_left,  3.phil3_take_left,  4.phil4_take_left,
          5.phil5_take_left,  6.phil1_take_right,  7.phil2_take_right,
          8.phil3_take_right,  9.phil4_take_right,
Transition to fire >

fired transition:  2.phil2_take_left


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  1  1  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
enabled:  3.phil3_take_left,  4.phil4_take_left,  5.phil5_take_left,
          7.phil2_take_right,  8.phil3_take_right,  9.phil4_take_right,
Transition to fire >

fired transition:  3.phil3_take_left


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  0  1  1  0  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0
enabled:  4.phil4_take_left,  5.phil5_take_left,  8.phil3_take_right,
          9.phil4_take_right,
Transition to fire >

fired transition:  4.phil4_take_left


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  0  0  1  0  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0
enabled:  5.phil5_take_left,  9.phil4_take_right,
Transition to fire >

fired transition:  5.phil5_take_left


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  0  0  0  0  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0
 no transition has concession!

Transition to fire >
```

Anhand der vorstehenden Meldung sehen Sie, daß keine Transition mehr Konzession hat; das gesamte Netz befindet sich in einem toten Zustand.

In dieser Situation können Sie z.B. die aktuelle Markierung mit `<W>` auf eine Datei speichern und danach mit `<B>` einen Schritt zurückgehen.

```
Transition to fire >   W

Marking output file > dinner.mar

Marking written to dinner.mar
```

```
Transition to fire >   B

previous state


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  0  0  1  0  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0
enabled:  5.phil5_take_left,  9.phil4_take_right,
```

Nun können Sie die andere Transition auswählen und dadurch den vierten Philosophen die fehlende Gabel aufnehmen lassen. Durch Eingabe von <cr> kann er mit Essen beginnen und danach beide Gabeln wieder abgeben.

```
Transition to fire > 9

fired transition:  9.phil4_take_right


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  0  0  0  0  1  1  1  1  0  0  0  0  1  0  0  0  0  0  0
enabled: 14.phil4_start_eat,
Transition to fire >

fired transition: 14.phil4_start_eat


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  0  0  0  0  1  1  1  0  0  0  0  0  0  0  0  0  0  1  0
enabled: 19.phil4_put_back,
Transition to fire >

fired transition: 19.phil4_put_back

 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  0  0  1  1  0  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0
enabled:  4.phil4_take_left,  5.phil5_take_left,  8.phil3_take_right,
          9.phil4_take_right,
Transition to fire >
```

Nun könnten die Philosophen drei und fünf eine weitere Gabel aufnehmen. Natürlich könnte auch der vierte Philosoph gleich wieder Appetit verspüren und unfairerweise die Gabeln immer wieder aufnehmen, die anderen Philosophen würden mit der Zeit verhungern.

Mit <R> können Sie das Spiel beenden und in den Anfangszustand zurückkehren.

```
Transition to fire >  R

reset to the initial state


 PL:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 MA:  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
enabled:  1.phil1_take_left,  2.phil2_take_left,  3.phil3_take_left,
          4.phil4_take_left,  5.phil5_take_left,  6.phil1_take_right,
          7.phil2_take_right,  8.phil3_take_right,  9.phil4_take_right,
         10.phil5_take_right,
Transition to fire >
```

Sie können sich außerdem mit `<S>` jederzeit die von INA bei der sturen Reduktion benutzte sture Menge ansehen. Bei der Anfangsmarkierung sieht die sture Menge wie folgt aus:

```
Transition to fire >  S

Stubborn set used by stubborn set method is:
       1.phil1_take_left, 10.phil5_take_right,
```

Mit `<A>` werden alle minimalen sturen Mengen ausgegeben, ein Zähler zeigt den Fortgang der Berechnung:

```
Transition to fire >  A
                      Candidate nr. 32
All minimal stubborn sets:
   1:  5.phil5_take_left,  9.phil4_take_right,
   2:  4.phil4_take_left,  8.phil3_take_right,
   3:  3.phil3_take_left,  7.phil2_take_right,
   4:  2.phil2_take_left,  6.phil1_take_right,
   5:  1.phil1_take_left, 10.phil5_take_right,
```

Mit `<Q>` beenden Sie den Schaltmodus und kehren in das Hauptmenü zurück. Die Anfangsmarkierung wird automatisch wiederhergestellt.

5. Reduktion

Bei einem großen Netz kann es sein, daß es unmöglich ist, den gesamten Zustandsgraphen zu berechnen. Um diesen zu verkleinern, können Sie das Netz schrittweise reduzieren. Ein Reduktionsschritt besteht dabei im Ersetzen eines Unternetzes durch ein anderes Unternetz (im allgemeinen mit weniger Plätzen) in der Weise, daß die Netzeigenschaften Lebendigkeit, Nicht–Lebendigkeit, Beschränktheit und Unbeschränktheit erhalten bleiben.

5.1. Grundlagen

Die Reduktion von Netzen wird im Hauptmenü unter `<R>` angeboten, allerdings nur für ungefärbte Platz/Transitions-Netze ohne Zeitbewertung und ohne Nutzung von Prioriäten, weil nur in diesem Fall die implementierten Reduktionsschritte Sinn haben, d.h. aus Eigenschaften des Reduktionsresultates auf Eigenschaften des ursprünglichen Netzes geschlossen werden kann.

Zuerst wird folgender Hinweis und das Reduktionsmenü ausgegeben:

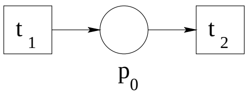
```
Reduction:
Transitions without pre-places are live and will be deleted together with their
post-places. If such transition has excepted post-places, it will be excepted.
Isolated transitions will be deleted even if they are excepted from reduction!
Reduction menue:
For:                            please, type:
    Quit .....................................Q
    Program a sequence of reduction steps.......P
    change eXceptions...........................X
    connectedness Test..........................T
    save the actual net.........................N
    Raise node numbers with exceptions..........R
    terminal Output of the current net..........O
    reset to the Initial net....................I
    Squeeze the node numbers....................S
    Fusion of congruent nodes...................F
    Merging of equivalent places................M
    ELIMINATION of F(pF)={p}-places.............A
    ELIMINATION of (Fp)F={p}-places.............B
    ELIMINATION of looping places...............C
    DELETION of looping transitions.............U
    DELETION of single output transitions.......V
    DELETION of PTP-sequences...................W
 choice>
```

Für die Reduktion können Sie mit der Funktion `<X>` (siehe Seite 45) Ausnahmen vereinbaren, d.h. Mengen von Plätzen und Transitionen, die bei der Reduktion nicht gestrichen werden dürfen.

Enthält das Netz Fakten so gibt INA vor dem Beginn der Reduktion folgenden Hinweis: `Facts should be deleted or excepted from reduction`. Bestehende Fakten werden automatisch bei der Reduktion ausgenommen. Sie können die Abläufe beschleunigen, wenn Sie alle Fakten vor der Reduktion löschen, indem Sie die Abfrage `Delete all facts?` mit `<Y>` beantworten.

INA zeigt im Reduktionsmenü an, daß Transitionen ohne Vorplätze zusammen mit ihren Nachplätzen gelöscht werden, sofern sie nicht selbst oder einer ihrer Nachplätze mit der Funktion `<X>` von der Reduktion ausgenommen sind. Transitionen ohne Vorplätze sind lebendig und ihre Nachplätze unbeschränkt [Sta90, Regel 1 (85)].

Wenn Sie ein Netz reduzieren, das Transitionen ohne Vorplätze enthält, dann werden diese erst gelöscht, wenn Sie die Kommandos `<A>`, `<B>`, `<V>` oder `<W>` benutzen. Die Untersuchung auf Transitionen ohne Vorplätze wird dabei nur ausgeführt, wenn das Netz neu eingelesen, gerade zurückgesetzt oder erfolgreich reduziert wurde. Bitte beachten Sie: Führen Sie eines der aufgeführten Kommandos aus, bevor Sie eine andere Reduktionsprozedur starten. Sollte nämlich die Anwendung der anderen Prozedur nicht möglich sein, so nimmt INA an, daß das Netz nicht geändert wurde und beim nächsten Durchlauf wird deshalb (irrtümlicherweise) nicht neu auf Transitionen ohne Vorplätze untersucht.



Beispiel 3: Ein Netz mit einer Transition ohne Vorplatz

Das einfache Netz des Beispiels 3 wird durch einmalige Anwendung eines der vorstehenden Kommandos vollständig gelöscht:

```
There are transitions without pre-place, the net is not bounded.
 1 live transition(s) deleted
 1 unbounded place(s) deleted
There are transitions without pre-place, the net is not bounded.
 1 live transition(s) deleted
The net is live and totally unbounded.
```

Im Session–Report wird die gleiche Situation beschrieben durch folgendes Protokoll:

```
deleted (live) transitions:
       1,
deleted (unbounded) places:
       0,
deleted (live) transitions:
       2,
deleted (unbounded) places:
      empty
The net is live and totally unbounded.
```

Falls das aktuelle Netz den Voraussetzungen der von Ihnen aufgerufenen Reduktionsprozedur nicht genügt, kehrt INA mit der Meldung `No application possible` ins Reduktionsmenü zurück. Laufende Reduktionen können Sie jederzeit mit `<Q>` abbrechen. Bereits begonnene Schritte werden zu Ende geführt und die Meldung `Stopped` ausgegeben.

INA zeigt den Fortgang der Reduktion am Bildschirm und im Session–Report an. Plätze bzw. Transitionen werden dabei nicht mit ihren Namen, sondern nur mit einem vorangestellten `p` bzw. `t` und ihrer Nummer ausgegeben.

Bei der Reduktion entstehende isolierte Knoten werden auf jeden Fall gelöscht, auch wenn sie vorher ausgenommen wurden.

Bitte beachten Sie: Im reduzierten Netz sind oftmals die Namen von Plätzen und Transitionen gelöscht und Werte wie Kapazitäten und Prioritäten teilweise nicht übernommen. Falls Sie ein reduziertes Netz weiter analysieren möchten, dann überprüfen Sie es vorher im Editor.

5.2. Hilfsfunktionen

Bei der Reduktion werden Sie durch folgende Kommandos unterstützt:

`<P>     Program a sequence of reduction steps`

Mit diesem Kommando können Sie eine Folge von Reduktionsschritten programmieren, so daß diese automatisch nacheinander ausgeführt werden. Sie geben eine Folge von Buchstaben ein und beenden dies durch Eingabe von `<Q>`: `Give a sequence of letters finnished by Q`. Jedem Buchstaben ist ein Reduktionsschritt zugeordnet. Mögliche Eingaben sind daher `<I>`, `<S>`, `<F>`, `<M>`, `<A>`, `<B>`, `<C>`, `<U>`, `<V>`, `<W>` und `<Q>`. Nachdem Sie die Programmierung beendet haben, zeigt INA das Programm und wartet auf Bestätigung der Abarbeitung: `Execute?`. Falls Sie die Frage mit `<Y>` beantworten, werden alle eingegebenen Prozeduren nacheinander ausgeführt. Danach befinden Sie sich wieder im Reduktionsmenü.

Wenn Sie dieses Kommando erneut aufrufen (ohne vorherige Rückkehr ins Hauptmenü), zeigt INA das zuletzt eingegebene Programm und ermöglicht die erneute Ausführung.

`<X>     change eXceptions`

Mit diesem Kommando können Sie Plätze und Transitionen des aktuellen Netzes bestimmen, die von der Reduktion ausgeschlossen werden sollen. Die Ausnahmen können in eine Datei mit der Extension `.exc` gespeichert werden. INA zeigt zuerst alle vorhandenen Dateien an. Sie können eine Datei auswählen und einlesen lassen oder durch `<esc>` die Ausnahmen direkt eingeben.

Falls Sie das Kommando erneut aufrufen (INA hat ausgenommene Plätze und Transitionen im Speicher) müssen Sie zuerst die Frage `Delete exceptions? (P)laces, (T)ransitions, (N)o` beantworten. Nach der Eingabe von `<P>` oder `<T>` sehen Sie die bis jetzt ausgenommenen Plätze oder Transitionen mit ihren Nummern. Nun können Sie nacheinander die zu löschenden Ausnahmen angeben oder durch Eingabe von `<A>` alle löschen lassen. Sollten Sie dagegegen die Frage mit `<N>` beantworten, so können Sie eine weitere Datei einlesen oder weitere Ausnahmen direkt eingeben. Beachten Sie: Die bestehenden Ausnahmen bleiben erhalten.

<T> connectedness Test

Mit diesem Kommando kann das aktuelle Netz auf Zusammenhang getestet werden. Es arbeitet genauso wie das Kommando <T><C> im Editor (siehe Seite 32 im Kapitel 3.3).

<N> save the actual net

Dieses Kommando dient zum Speichern des aktuellen Netzes mit einer neuen Netznummer. Sie werden gebeten, eine neue Netznummer und den Namen der Ausgabedatei anzugeben. Der Standardwert der Extension ist .red, da Sie ein Reduktionsresultat speichern.

<R> Raise node numbers with exceptions

Dieses Kommando erhöht je nach Wahl die Platz– oder Transitionsnummern um eine anzugebende Spanne. Die ausgenommenen Knoten bleiben dabei unberücksichtigt. Siehe auch die Beschreibung des Editorkommandos <C><U> (Seite 30 im Kapitel 3.2).

<O> terminal Output of the current net

Dieses Kommando gibt Informationen über das aktuelle Netz auf dem Bildschirm aus. Es ist identisch mit dem Kommando <W><T> im Editor (siehe Seite 27 im Kapitel 3.1).

<I> reset to the Initial net

Mit diesem Kommando können Sie das ursprüngliche Netz, welches zu Beginn der Reduktion in der Datei ININET.pnt abgelegt wurde, erneut einlesen, um andere Reduktionsschritte oder eine andere Reihenfolge auszuprobieren. Eventuell gelöschte Fakten müssen erneut gelöscht werden: Delete all facts? Die bisher eingegebenen Ausnahmen bleiben unverändert und müssen deshalb gegebenenfalls überprüft werden: Exceptions ok after reset?.

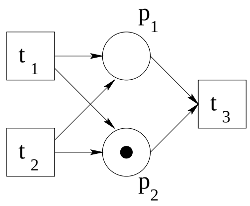
<S> Squeeze the node numbers

Dieses Kommando arbeitet genauso wie die Unterkommandos <R> und <F> im Editorkommando <C><U> (siehe Seite 30 im Kapitel 3.2) und dient zum Verdichten der Knotennummern. Die Plätze und/oder Transitionen werden mit 1 beginnend fortlaufend neu numeriert. Es empfielt sich, diese Funktion am Ende der Reduktion aufzurufen, denn für viele Analyseprozeduren ist es bezüglich Laufzeit und Speicherplatzausnutzung günstig, wenn die Knotennummern „dicht“ liegen.

5.3. Reduktionsprozeduren

<F> Fusion of congruent nodes

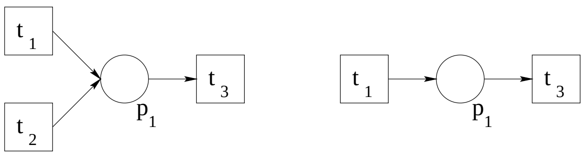
INA sucht nach jeweils zwei Plätzen oder zwei Transitionen, die mit allen übrigen Knoten des Netzes in der gleichen Weise verbunden sind, sogenannte parallele Knoten [Sta90, Definition 9.3 (87)].



Beispiel 4: Ein Netz mit zwei parallelen Plätzen und Transitionen

Abbildung 4 zeigt ein Netz mit zwei parallelen Plätzen (p1 und p2) und zwei parallelen Transitionen (t1 und t2).

Sie müssen die zu untersuchende Knotenart direkt angeben: Check places (P), transitions (T) or exit (E) ? Nach der Eingabe von <P> bzw. <T> für die Suche nach Plätzen bzw. Transitionen meldet INA entweder, daß keine parallelen Knoten gefunden (No application possible!) oder wieviele Knoten gelöscht wurden (nodes deleted). Normalerweise werden immer die Knoten mit der höheren Nummer gelöscht. Bei parallelen Plätzen wird dagegen der Platz mit der kleineren Nummer gestrichen, falls er mehr Marken trägt [Sta90, Regel 3 (88)].



Beispiel 5: Reduktionen des Beispielnetzes 4 bezüglich paralleler Knoten

Das Netz des Beispiels 4 wird durch Eingabe von <P> zum linken Netz des Beispiels 5. Da der Platz p_2 eine größere Nummer als p_1 hat und p_1 nicht mehr Marken als p_2 enthält, wird der Platz p_2 gelöscht. Wird weiter mit <T> reduziert, so ergibt sich das rechte Netz des Beispiels 5. Die Transition t_2 wird gelöscht, da sie eine größere Nummer als die Transition t_1 hat.

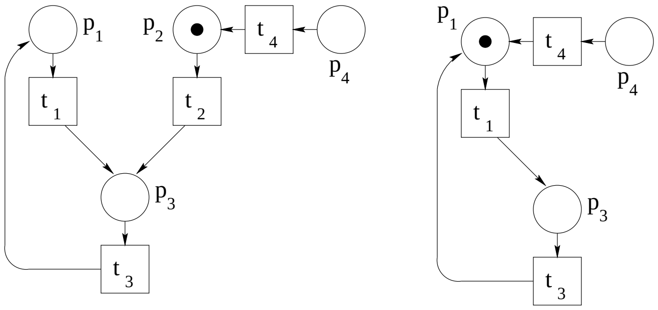
Im Session–Report finden Sie folgendes Protokoll:

```
Fuse:
p2 deleted
 1 place(s) deleted
t2 deleted
 1 transition(s) deleted
```

<M> Merging of equivalent places

Dieses Kommando verschmilzt äquivalente Plätze des Netzes. Zwei Plätze p_1 und p_2 sind äquivalent, wenn

- beide Plätze je genau eine Nachtransition t_1 bzw. t_2 besitzen,
- die Vielfachheit der Bögen von p_1 zu t_1 bzw. von p_2 zu t_2 gleich 1 ist und
- die Transitionen t_1 bzw. t_2 mit allen von p_1 und p_2 verschiedenen Plätzen auf die gleiche Weise verbunden sind [Sta90, Definition 9.4 (88)].



Beispiel 6: Ein Netz mit äquivalenten Plätzen und deren Reduktion

Im Netz des Beispiels 6 sind die Plätze p_1 und p_2 äquivalent.

Falls INA zwei äquivalente Plätze p_1 und p_2 findet, so werden der Platz p_2 (mit einer höheren Nummer als p_1) sowie seine Nachtransition t_2 gestrichen, die auf p_2 befindlichen Marken zusätzlich auf p_1 gelegt und die nach p_2 führenden Bögen nach p_1 geführt [Sta90, Regel 4 (89)].

Das rechte Netz des Beispiels 6 zeigt die Reduktion des linken Netzes. Der Platz p_2 und seine Nachtransition t_2 werden gelöscht, die Marke von p_2 wird auf p_1 gelegt und der Bogen von t_4 zu p_2 wird nach p_1 geführt.

Im Session–Report finden Sie folgendes Protokoll:

```
Merge equivalent places:
Transition  2 deleted, Place 2 deleted
```

<A> ELIMINATION of $F(pF) = \{p\}$-places

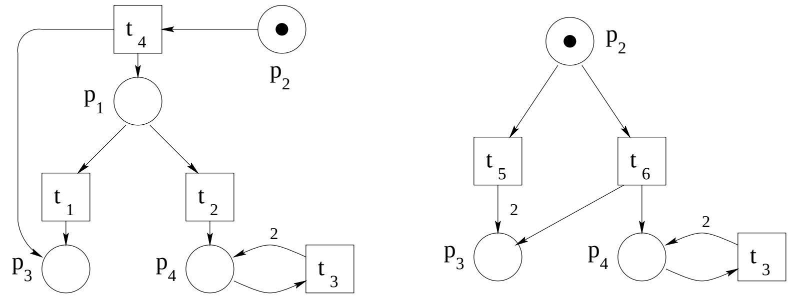
Dieses Kommando verschmilzt Vor– mit Nachtransitionen. Hierbei bezeichnet pF alle Nachtransitionen des Platzes p und $F(pF)$ alle Vorplätze der Nachtransitionen von p. $F(pF) = \{p\}$ sind die Plätze p, die die einzigen Vorplätze ihrer Nachtransitionen sind.

Gesucht wird ein Platz mit folgenden Eigenschaften [Sta90, Vorauss. Regel 5 (90)]:

- p hat $k > 0$ Vortransitionen $t_1, \ldots, t_k$
- p hat $n > 0$ Nachtransitionen $t'_1, \ldots, t'_n$
- keine Transition ist zugleich Vor– und Nachtransition von p
- wenigstens eine Nachtransition von p hat einen Nachplatz
- p ist der einzige Vorplatz seiner Nachtransitionen
- alle bei p entspringende Bögen haben die gleiche Vielfachheit v
-
  - falls p genau eine Nachtransition hat ($n = 1$), dann haben alle bei p einmündenden Bögen eine durch v teilbare Vielfachheit
  - falls p dagegen mehr als eine Nachtransition hat ($n > 1$), dann haben alle bei p einmündenden Bögen die Vielfachheit v

Der Platz p_1 des linken Netzes des Beispiels 7 erfüllt die obigen Bedingungen.

Falls INA einen Platz findet, der allen Bedingungen genügt, so werden eventuell auf p liegende Marken weggeschaltet, bis weniger als v Marken auf p sind. Dann werden die



Beispiel 7: Ein mit <A> reduzierbares Netz und das Ergebnis

Vortransitionen $t_1, \ldots, t_k$, der Platz p selbst und die Nachtransitionen $t'_1, \ldots, t'_n$ gestrichen, dafür werden $k \cdot n$ neue Transitionen $t_{i,j}$ so eingeführt, daß ein Schalten von $t_{i,j}$ die gleiche Wirkung wie ein Schalten von t_i gefolgt von x-maligem Schalten von t'_j, wobei x die Vielfachheit des Bogens von t_i nach p dividiert durch v ist [Sta90, Regel 5 (90)]. INA meldet nur die Anzahl der gelöschten Plätze.

Im Beispiel 7 ergibt sich das rechte Netz. Der Platz p_1 und die Transitionen t_1, t_2 und t_4 werden gelöscht und durch die Transitionen t_5 und t_6 ersetzt. Das Schalten von t_5 im reduzierten Netz hat die gleiche Wirkung wie das Schalten von t_4 und t_1 im alten Netz, genauso verhält es sich mit dem Schalten von t_6 und dem Schalten von t_4 und t_2.

Im Session–Report finden Sie folgendes Protokoll:

```
place elimination A:
New transition 5 replaces transitions 4 * 1
New transition 6 replaces transitions 4 * 2
Place 1 deleted
```

<B> ELIMINATION of $(Fp)F = \{p\}$-places

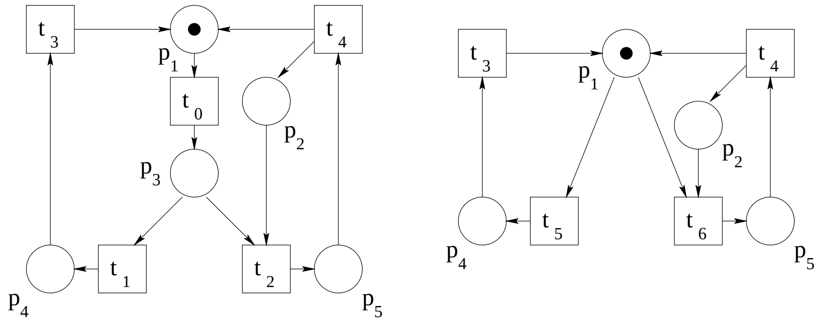
Dieses Kommando verschmilzt eine Vor– mit Nachtransitionen. Hierbei bezeichnet Fp alle Vortransitionen des Platzes p und $(Fp)F$ alle Nachplätze der Vortransitionen von p. $(Fp)F = \{p\}$ sind die Plätze p, die die einzigen Nachplätze ihrer Vortransitionen sind.

Gesucht wird ein Platz mit folgenden Eigenschaften [Sta90, Vorauss. Regel 6 (92)]:

- p hat genau eine Vortransitionen t_0 und p ist der einzige Nachplatz dieser Transition.
- p hat $n > 0$ Nachtransitionen $t'_1, \ldots, t'_n$
- keine Transition ist zugleich Vor– und Nachtransition von p
- t_0 ist die einzige Nachtransition ihrer Vorplätze
- alle bei p einmündenden oder entspringenden Bögen haben die gleiche Vielfachheit v und auf p liegen weniger als v Marken

Der Platz p_3 des linken Netzes des Beispiels 8 erfüllt die obigen Bedingungen.

Falls INA einen Platz findet, der allen Bedingungen genügt, so werden die Vortransition t_0, der Platz p selbst und die Nachtransitionen $t'_1, \ldots, t'_n$ gestrichen, dafür werden n neue Transitionen $t_1, \ldots, t_n$ so eingeführt, daß ein Schalten von t_j die gleiche Wirkung wie ein Schalten von t_0 gefolgt von einem Schalten von t'_j [Sta90, Regel 6 (92)]. INA meldet nur die Anzahl der gelöschten Plätze.



Beispiel 8: Ein mit <B> reduzierbares Netz und das Ergebnis

Im Beispiel 8 ergibt sich das rechte Netz. Der Platz p_3 und die Transitionen t_0, t_1 und t_2 werden gelöscht und durch zwei Transitionen t_5 und t_6 ersetzt. Das Schalten von t_5 im reduzierten Netz hat die gleiche Wirkung wie das Schalten von t_0 und t_1 im alten Netz, genauso verhält es sich mit dem Schalten von t_6 und dem Schalten von t_0 und t_2.

Im Session–Report finden Sie folgendes Protokoll:

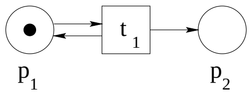
```
place elimination B:
Transition 5 replaces transitions 1 * 0
Transition 6 replaces transitions 2 * 0
Place 3 deleted
```

`<C>     ELIMINATION of looping places`

Dieses Kommando sucht nach beschränkten und unbeschränkten Plätzen, die folgende Bedingung erfüllen: [Sta90, Vorauss. Regel 7 (95) mit Ergänzungen]

- jede Transition t zieht höchstens so viele Marken von p ab, wie sie auf p aufbringt
- p ist ausreichend markiert

Der Platz p wird gestrichen, falls dadurch keine isolierte Transitionen entstehen. INA meldet die Anzahl der gelöschten Plätze.



Beispiel 9: Ein Netz mit einem zu reduzierenden Laufplatz

Im Netz des Beispiels 9 erfüllt der Platz p_1 die Bedingung und wird deshalb gelöscht.

Zu Beginn der Reduktion mit diesem Kommando stellt INA die Frage `Shall we delete only bounded places?` Falls Sie mit `<Y>` antworten, werden nur beschränkte Plätze gelöscht, sonst auch unbeschränkte. Ein Platz p ist unbeschränkt, wenn ein Schalten einer Transition t mehr Marken auf p bringt, als sie von p abzieht und die Transition t lebendig ist, d.h. p ausreichend markiert ist.

Falls im Netz des Beispiels 9 die Vielfachheit des Bogens von t_1 nach p_1 um eins auf zwei erhöht wird, dann ist der Platz p_1 unbeschränkt und wird deshalb nur gelöscht, wenn Sie die Abfrage mit `<N>` beantworten. Im Session–Report finden Sie folgendes Protokoll:

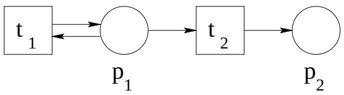
```
place elimination C:
only bounded places to be deleted
place elimination C:
bounded and unbounded places to be deleted
p1 deleted, was unbounded, if the net is live.
```

`<U>     DELETION of looping transitions`

Dieses Kommando löscht alle Transitionen t, die folgende Bedingungen erfüllt [Sta90, Regel 8 (95)]:

- das Schalten von t ändert die Markierung nicht
- es existiert eine Transition t', die zu ihrer Konzessionierung auf jedem Vorplatz von t mindestens so viele Marken wie t anfordert

Diese Situation wird als Schleife bezeichnet. Falls INA eine solche Transition findet, wird diese gelöscht. INA meldet die Anzahl der gelöschten Transitionen.



Beispiel 10: Ein Netz mit einer zu reduzierenden Schleife

Ein einfaches Beispiel für eine Schleife zeigt das Netz 10. Das Schalten der Transition t_1 ändert die Markierung des Netzes nicht. Da die Transition t_2 mindestens so viele Marken von p_1 abzieht wie t_1, wird t_1 gelöscht.

Im Session–Report finden Sie das folgende Protokoll:

```
transition elimination U:
transition 1 deleted
```

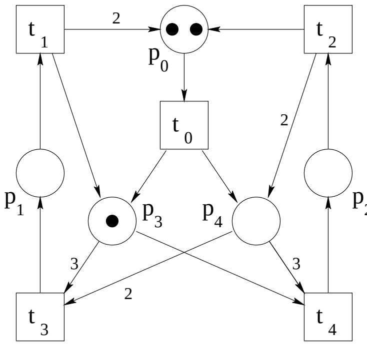
`<V>     DELETION of single output transitions`

Dieses Kommando verschmilzt einen Vor– mit den Nachplätzen. Gesucht wird eine Transitionen mit folgenden Eigenschaften [Sta90, Vorauss. Regel 9 (95)]:

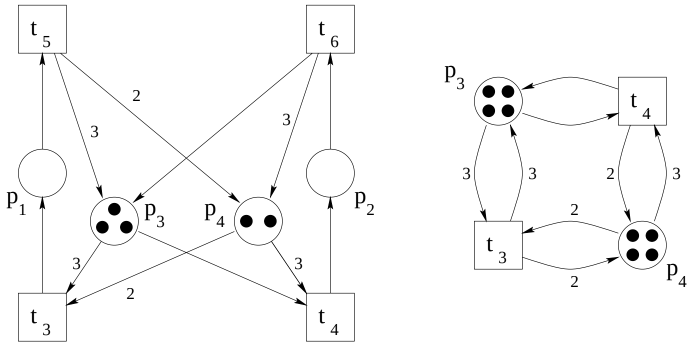
- t hat genau einen Vorplatz p
- die Vielfachheit des Bogens von p nach t ist eins
- p hat $k > 0$ Vortransitionen $t_1, \ldots, t_k$
- p ist kein Nachplatz von t, aber t hat Nachplätze

Die Transition t_0 des Netzes des Beispiels 11 erfüllt die obigen Bedingungen.

Falls INA eine Transition findet, die allen Bedingungen genügt, so wird t solange geschaltet, bis p keine Marke mehr trägt. Dann wird die Transition t, der Platz p und die Vortransitionen $t_1, \ldots, t_k$ gestrichen, dafür werden k neue Transitionen $t_j^\star$ so eingeführt,



Beispiel 11: Ein mit `<V>` reduzierbares Netz



Beispiel 12: Reduktionen des Beispielnetzes 11

daß ein Schalten von $t_j^\star$ die gleiche Wirkung wie ein Schalten von t_j gefolgt von x-maligem Schalten von $t_j^\star$, wobei x die Vielfachheit des Bogens von t_j nach p ist [Sta90, Regel 9 (95)]. INA meldet nur die Anzahl der gelöschten Transitionen.

Im Beispiel 11 ergibt sich durch einmalige Anwendung der Regel das linke Netz des Beispiels 12. Zuerst wird die Transition t_0 geschaltet, bis die zwei Marken von p_0 verbraucht sind. Dadurch liegen auf p_3 drei und auf p_4 zwei Marken. Als nächstes werden die Transition p_0, der Platz p_0 und die Transitionen t_1 und t_2 gelöscht und durch die Transitionen t_5 und t_6 ersetzt. Das Schalten von t_5 im reduzierten Netz hat die gleiche Wirkung wie das Schalten von t_1 und das zweimalige Schalten von t_0 im alten Netz, genauso verhält es sich mit dem Schalten von t_6 und dem Schalten von t_2 und t_0.

Insgesamt wird die Regel dreimal angewendet, es ergibt sich daß rechte Netz des Beispiels 12. Im Session–Report finden Sie folgendes Protokoll:

```
transition elimination V:
t0,p0 deleted.
t1,p1 deleted.
t2,p2 deleted.
```

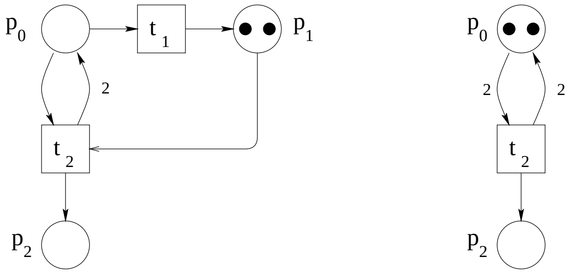
`<W>     DELETION of PTP-sequences`

INA sucht eine Transition t und zwei Plätze p und q, die folgende Bedingungen genügen:

- p ist der einzige Vorplatz, q der einzige Nachplatz von t, wobei $p \neq q$
- t ist die einzige Vortransition von q und q besitzt Nachtransitionen
- die Vielfachheit des Bogens von p nach t und von t nach q ist gleich eins

In solch einer Situation werden die Plätze p und q verschmolzen und die Transition t gelöscht.

Diese Transformation kann beschränkte bzw. nichtlebendige Netze in unbeschränkte bzw. lebendige Netze transformieren; ist also nur eingeschränkt anwendbar. Allerdings können Sie aus der Beschränkheit bzw. Nichtlebendigkeit des Reduktionsresultates auf die Beschränkheit bzw. Nichtlebendigkeit des ursprünglichen Netzes schließen. INA gibt die Warnung `Non-liveness and boundedness are not preserved under this rule` aus und bittet Sie um Bestätigung der Ausführung: `Execute?` Falls Sie diese Frage mit `<N>` beantworten, wird das Kommando nicht ausgeführt.



Beispiel 13: Ein mit `<W>` reduzierbares Netz und das Ergebnis

Das linke Netz des Beispiels 13 ist tot, da keine Transition schalten kann. Nach Reduktion mit `<W>` ergibt sich durch Löschen der Transition t_1 und Verschmelzen der Plätze p_0 und p_1 zu p_0 das rechte Netz. Dieses Netz ist nun lebendig, da die Transition t_2 immer schalten kann. Ein totes Netz ist also in ein lebendiges Netz transformiert worden!

INA meldet die Anzahl der gelöschten Transitionen und schreibt in den Session–Report eine erneute Warnung aus:

```
transition elimination W:
Boundedness and non-liveness are not preserved!

Nodes deleted:
p1, t1;
```

6. Analyse

Drei unterschiedliche Vorgehensweisen bei der Analyse eines Netzes können unterschieden werden: Erreichbarkeitsanalyse, Invariantenanalyse und Strukturanalyse.

Als Erreichbarkeitsanalyse wird jedes Vorgehen bezeichnet, das auf einer (mindestens teilweisen) Konstruktion des Zustands- bzw. Erreichbarkeitsgraphen des Netzes beruht. Dazu gehört die Pfadberechnung und die Berechnung und Analyse des Zustandsgraphen, wobei dieser symmetrisch bzw. stur reduziert berechnet werden kann. Für unbeschränkte Netze kann der Überdeckbarkeitsgraph berechnet werden.

Zur Invariantenanalyse gehören alle Verfahren, die auf die Lösung aus der Inzidenzmatrix des Netzes abgeleiteter linearer Gleichungssysteme abzielen.

Die Strukturanalyse versucht Aussagen über dynamische Eigenschaften wie Lebendigkeit auf der Basis von strukturellen Eigenschaften des Netzes, wie z.B. der Deadlock-Falle-Eigenschaft zu gewinnen.

6.1. Grundlagen

In den Analyseteil ist ein Expertsystem integriert, das in der Theorie nachgewiesene Zusammenhänge zwischen Netzeigenschaften berücksichtigt und aus den für das aktuelle Netz bereits festgestellten Resultaten Schlußfolgerungen zieht. Auf diese Weise kann unter Umständen aus einer elementaren Eigenschaft wie der, daß keine Transition beim Schalten die Gesamtzahl der Marken im Netz verändert (das Netz ist dann konservativ, bzw. subkonservativ, wenn sich die Markenzahl höchstens verringern kann) auf eine solche dynamische Eigenschaft wie Beschränktheit geschlossen werden. Andererseits werden durch das Expertsystem überflüssige Analysen vermieden, denn es hat z.B. nicht viel Sinn ein inhomogenes Netz daraufhin zu untersuchen, ob es die Deadlock-Falle-Eigenschaft hat.

Bitte beachten Sie, daß INA nach dem Ende einer Berechnung benutzten Speicher zurückgibt. Angekündigt wird dies mit der Meldung `returning heap`. Es kann vorkommen, daß dies nach speicherintensiven Prozeduren einige Sekunden dauert. Bitte warten Sie deshalb, bis der Speicher wieder freigegeben ist und die Analyse fortgesetzt werden kann.

6.2. Analysestatus

Der Fortschritt der Analyse wird in zwei Statuszeilen angezeigt.

```
ORD HOM NBM PUR CSV SCF CON SC  Ft0 tF0 Fp0 pF0 MG  SM  FC  EFC ES
 Y   Y   Y   N   N   N   Y   N   Y   N   N   N   N   N   N   N   Y
DTP SMC SMD SMA CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&S WL  CL
 ?   N   ?   ?   N   ?   N   N   ?   ?   N   ?   ?   ?   ?   N   ?   ?
```

Aufgeführt werden mögliche Eigenschaften des betrachteten Netzes, dabei bedeutet das jeweils darunter stehende Symbol, ob die Eigenschaft erfüllt (`Y`), nicht erfüllt (`N`) oder noch keine Entscheidung getroffen werden konnte (`?`). Die erste Zeile enthält elementare Netzeigenschaften, die bei der Voranalyse durch Betrachtung der Struktur des Netzes gefunden werden, die zweite dagegen weitere dynamische, d.h. von der Anfangsmarkierung abhängige Eigenschaften, die eventuell nach weiteren Analysen geklärt werden können. Natürlich ist die Anzeige auch von den eingestellten Netzoptionen und bereits gefundenen Ergebnissen abhängig, den einige Eigenschaften haben nur unter bestimmten Bedingungen einen Sinn. Das obige Beispiel zeigt alle Möglichkeiten.

6.2.1. Elementare Netzeigenschaften

`ORD` **Gewöhnlichkeit** `ordinary`

Ein Petri-Netz wird gewöhnlich genannt, wenn die Vielfachheit jedes Bogens gleich eins ist [Sta90, Definition 2.3 (24)]. Ein gefärbtes Netz ist gewöhnlich, wenn seine Entfaltung ein gewöhnliches Netz ist.

`HOM` **Homogenität** `homogenous`

Ein Netz wird homogen genannt, wenn für jeden Platz alle abgehenden Bögen die gleiche Vielfachheit haben [Sta90, Definition 14.4 (157)].

`NBM` **Nichtblockierende Vielfachheit** `nonblocking multiplicity`

Die Vielfachheit der Bögen eines Netzes wird als nichtblockierend bezeichnet, wenn bei jedem Platz das Minimum der Vielfachheiten der einmündenden Bögen nicht kleiner ist als das Maximum der Vielfachheiten der abgehenden Bögen [Sta90, Definition 15.3 (164)].

`PUR` **Reinheit** `pure`

INA prüft, ob in dem aktuellen Netz Transitionen vorkommen, bei denen ein Vorplatz zugleich Nachplatz ist. Wenn das der Fall ist, ist das Netz nicht rein, d.h. nicht schleifenfrei [Sta90, Bemerkung vor Abb. 3.1 (32)]. Solche Netze sind unter der sicheren Schaltregel nicht lebendig.

`CSV` **Konservativität** `conservativ`

Ein Netz heißt konservativ, wenn alle Transitionen genauso viele Marken auf ihre Nachplätze legen, wie sie von ihren Vorplätzen abziehen. In einem konservativen Netz ändert sich deshalb die Gesamtzahl der Marken durch Schalten einer beliebigen Transition nicht. Die Markenanzahl ist also eine Invariante. Ein Netz heißt subkonservativ, wenn alle Transitionen höchstens so viele Marken auf ihre Nachplätze legen, wie sie von ihren Vorplätzen abziehen, die Gesamtzahl der Marken kann sich also nicht erhöhen.

`SCF` Statische Konfliktfreiheit `static conflict free`

Wenn zwei Transitionen einen gemeinsamen Vorplatz haben (sie teilen sich diesen), dann stehen die beiden Transitionen in einem statischen Konflikt um die Marken auf diesem Vorplatz. Das Netz ist dann nicht statisch konfliktfrei [Sta90, Definition 3.4 (35)].

`CON` Zusammenhang `connected`

Ein Netz ist zusammenhängend, wenn von jedem Knoten zu jedem anderen Knoten ein ungerichteter Weg im Netz existiert [Sta90, Definition 14.1 (148)]. Die Richtung der Bögen wird bei dieser Untersuchung also ignoriert.

`SC` Starker Zusammenhang `strongly connected`

Wenn ein Netz zusammenhängend ist, wird für jeden Knoten geprüft, ob auch ein gerichteter Weg zu jedem anderen Knoten existiert, d.h. die Richtung der Bögen wird zusätzlich beachtet [Sta90, Definition 14.1 (148)].

Einseitige Knoten

`Ft0` Transition ohne Vorplatz `transition without preplace`

Ein Netz hat $Ft0$–Transitionen, wenn es Transitionen ohne einen Vorplatz gibt, d.h. $Ft = \emptyset$.

`tF0` Transition ohne Nachplatz `transition without postplace`

Ein Netz hat tF0–Transitionen, wenn es Transitionen ohne einen Nachplatz gibt, d.h. $tF = \emptyset$.

`Fp0` Platz ohne Vortransition `place without pretransition`

Ein Netz hat Fp0–Plätze, wenn es Plätze ohne eine Vortransition gibt, d.h. $Fp = \emptyset$.

`pF0` Platz ohne Nachtransition `place without posttransition`

Ein Netz hat pF0–Plätze, wenn es Plätze ohne eine Nachtransition gibt, d.h. $pF = \emptyset$.

Strukturen

`MG` Synchronisationsgraph `marked graph`

Ein Netz ist ein Synchronisationsgraph, wenn jeder Platz genau eine Vortransition und genau eine Nachtransition hat. Dabei werden Bogenvielfachheiten ignoriert [Sta90, Definition 14.3.1 (152)].

SM Zustandsmaschine `state machine`

Ein Netz ist eine Zustandsmaschine, wenn jede Transition genau einen Vorplatz und genau einen Nachplatz hat. Dabei werden Bogenvielfachheiten ignoriert [Sta90, Definition 14.3.2 (152)].

FC Free–Choice–Netz `free choice`

Ein Netz ist free–choice, wenn jeder geteilte Platz der einzige Vorplatz seiner Nachtransitionen ist [Sta90, Definition 14.3.3 (152)].

EFC Extended–Free–Choice–Netz `extended free choice`

Ein Netz ist extended–free–choice, wenn die Nachtransitionen geteilter Plätze dieselben Vorplätze haben [Sta90, Definition 14.3.4 (152)].

ES Extended–Simple–Netz `extended simple`

Ein Netz ist extended–simple, wenn gilt: Haben zwei Plätze eine gemeinsame Nachtransition, dann sind alle Nachtransitionen des einen Platzes auch Nachtransitionen des anderen Platzes, oder umgekehrt, d.h. einer der beiden Plätze kann auch noch andere Nachtransitionen haben [Sta90, Definition 14.3.5 (153)].

6.2.2. Weitere Netzeigenschaften

DTP Deadlock–Falle–Eigenschaft `deadlock-trap-property`

Ein Netz erfüllt die Deadlock–Falle–Eigenschaft, wenn die maximale Falle in jedem minimalen Deadlock ausreichend markiert ist [Sta90, Kapitel 15 (162–175)]. Eine Falle ist eine Menge von Plätzen, die nicht sauber werden kann, wenn sie Marken enthält, weil jede Transition, die Marken von einem Platz dieser Menge entfernt, einen Nachplatz in dieser Menge hat, also Marken in diese Menge zurückgibt. Die leere Menge ist also eine Falle. Eine Falle ist maximal, wenn sie nicht eine echte Teilmenge einer Falle ist. Ein Deadlock ist eine nichtleere Menge von Plätzen, die nicht wieder markiert werden kann, wenn sie einmal sauber ist, weil jede Transition, die Marken auf einen Platz aus dieser Menge schalten würde, einen Vorplatz in dieser Menge hat (also nicht schalten kann). Ein Deadlock ist minimal, wenn er keinen Deadlock echt enthält. Eine Platzmenge ist ausreichend markiert, wenn sie einen Platz enthält, der genügend Marken enthält, um alle seine Nachtransitionen zu konzessionieren.

Weiteres finden Sie im Kapitel 6.6 ab Seite 90.

SMC Zustandsmaschinenüberdeckbarkeit `state machine coverable`

Eine Komponente eines Netzes ist eine Platzmenge, die Deadlock und Falle zugleich ist. Durch diese Platzmenge wird ein Teilnetz mit diesen Plätzen und den anhängenden Transitionen bestimmt. Ist eine Reihe von Komponenten gegeben, die als Teilnetze Zustandsmaschinen bestimmen und ihre Vereinigung ist gleich der Platzmenge des Netzes, so wird dieses Netz zustandsmaschinenüberdeckbar genannt [Sta90, Definition 16.2.2 (180)].

Weiteres finden Sie im Kapitel 6.6 ab Seite 90.

`SMD` **Zustandsmaschinendekomponierbarkeit** `state machine decomposable`

Ist eine Reihe von Komponenten gegeben, die als Teilnetze stark zusammenhängende Zustandsmaschinen bestimmen und ihre Vereinigung ist gleich der Platzmenge des Netzes, so wird dieses Netz zustandsmaschinendekomponierbar genannt [Sta90, Definition 16.2.3. (180)].

Weiteres finden Sie im Kapitel 6.6 ab Seite 90.

`SMA` **Zustandsmaschinenauswählbarkeit** `state machine allocatable`

In einem zustandsmaschinendekomponierbaren Netz hat jede Transition einen Vorplatz; eine Vorplatzauswahl ist eine Abbildung, die jeder Transition einen ihrer Vorplätze zuordnet. Eine Komponente heißt ausgewählt bei einer Vorplatzauswahl, wenn diese allen Transitionen, die einen Vorplatz in der Komponente haben, einen Vorplatz aus der Komponente zuordnet. Das Netz wird zustandsmaschinenauswählbar genannt, wenn jede Vorplatzauswahl eine stark zusammenhängende Zustandsmaschinenkomponente auswählt [Sta90, Definition 16.3 (182)]

Weiteres finden Sie im Kapitel 6.6 ab Seite 90.

`CPI` **Platzinvariantenüberdeckbarkeit** `covered by place invariants`

Ein Netz ist mit Platzinvarianten überdeckt, wenn es eine P–Invariante gibt, die jedem Platz einen positiven Wert zuweist [Sta90, Definition 11.4 (114)].

`CTI` **Transitionsinvariantenüberdeckbarkeit** `covered by transition invariants`

Ein Netz ist mit Transitionsinvarianten überdeckt, wenn es eine T–Invariante gibt, die jeder Transition einen positiven Wert zuweist [Sta90, Definition 11.3 (112)].

`B` **Beschränktheit** `bounded`

Ein Netz ist beschränkt, wenn es eine Zahl k so gibt, daß bei keiner erreichbaren Markierung auf einem Platz mehr als k Marken liegen [Sta90, Definition 14.1 (38)].

`SB` **Strukturelle Beschränktheit** `structurally bounded`

Ein Netz ist strukturell beschränkt, wenn es bei jeder Anfangsmarkierung beschränkt ist [Sta90, Bemerkung (116)].

`REV` **Reversibilität** `reversible`

Ein Netz ist reversibel (rücksetzbar), wenn von jedem erreichbaren Zustand der Anfangszustand erreicht werden kann [Sta90, Definition 6.2 (59)].

Eine Beschreibung des Rücksetzbarkeitstests finden Sie im Kapitel 6.4.4 auf Seite 77.

`DSt` **Erreichbarkeit eines toten Zustandes** `dead state reachable`

Ein toter Zustand ist erreichbar, wenn ein Zustand erreichbar ist, in dem keine Transition mehr schalten kann [Sta90, Definition 6.1.1 (54)].

BSt Fakt hat Konzession `bad state reachable`

Ein „schlechter“ Zustand ist erreichbar, wenn ein Zustand erreichbar ist, in dem ein Fakt [Sta90, Definition 6.1.2 (54)] Konzession hat. Da ein Fakt eine Transition ist, die im Anfangszustand tot ist und diese Transition bei Eingabe des Netzes entsprechend gekennzeichnet wurde, liegt ein Modellierungsfehler vor (eine Transition, die nach der Modellierung nie schalten dürfte, kann in einem Zustand schalten).

DTr Tote Transition bei Anfangsmarkierung `dead transition at initial marking`

Hier wird angezeigt, daß das Netz tote Transitionen bei der Anfangsmarkierung hat, d.h. diese Transitionen sind Fakten.

DCF Dynamische Konfliktfreiheit `dynamically conflictfree`

Ein Netz heißt dynamisch konfliktfrei, wenn kein Zustand erreichbar ist, bei dem zwei Transitionen schalten können und die eine durch das Schalten der anderen die Konzession verliert [Sta90, Definition 3.4.3 (35)].

L Lebendigkeit `live`

Ein Netz ist lebendig, wenn alle seine Transitionen bei der Anfangsmarkierung lebendig sind, d.h. kein Zustand erreichbar ist, bei dem eine Transition tot ist. Fakten werden dabei ignoriert [Sta90, Definition 6.1.5 (55)].

Eine Beschreibung des Lebendigkeitstests finden Sie im Kapitel 6.4.4 auf Seite 77.

LV Lebendigkeit (unter Ignorierung toter Trans.) `live if dead trans. ign.`

Ein Netz ist lebendig unter Ignorierung toter Transitionen, wenn alle seine Transitionen, die nicht schon bei der Anfangsmarkierung tot sind, lebendig sind. Die dabei ignorierten Transitionen können als nicht gekennzeichnete Fakten betrachtet werden.

L&S Lebendig und Sicher `live and safe`

Ein Netz ist lebendig und sicher, wenn es lebendig ist und bei keiner erreichbaren Markierung mehr als eine Marke auf einem Platz liegt [Sta90, Definition 4.1 (39)].

Lebendigkeit für gefärbte Netze

WL Schwache Lebendigkeit `weakly live`

Ein gefärbtes Netz ist schwach lebendig, wenn alle Transitionen schwach lebendig sind, d.h. alle Transitionen sind in der Anfangsmarkierung bei einer Farbe lebendig [Sta90, Definition 20.6.1 (228)] (siehe auch Seite 77 im Kapitel 6.4.4).

CL Kollektive Lebendigkeit `collectively live`

Ein gefärbtes Netz ist kollektiv lebendig, wenn alle Transitionen kollektiv lebendig sind. Eine Transition ist kollektiv lebendig, wenn es für jeden erreichbaren Zustand eine Farbe gibt, so daß die Transition in dieser Farbe in einem von dieser Markierung erreichbaren Zustand schalten kann [Sta90, Definition 20.6.2 (228)] (siehe auch Seite 77 im Kapitel 6.4.4). Insbesondere ist jede schwach lebendige Transition auch kollektiv lebendig [Sta90, Folgerung 20.3.2 (229)].

6.3. Analysemenü und Voranalyse

Im Analysemenü werden in Abhängigkeit vom Netztyp unterschiedliche Analyseprozeduren angeboten. Nicht für jeden Netztyp sind alle Verfahren bereits theoretisch begründet bzw. überhaupt sinnvoll. Netze unter Zeitbewertung sind z.B. mit der TURING-MASCHINE gleichwertig, was die Existenz von Algorithmen zur Entscheidung z.B. der Beschränkheit ausschließt. Es werden für jeden Netztyp nur die Verfahren angeboten, die zu Aussagen über die wichtigen dynamischen Eigenschaften führen können. Die Rückkehr ins Hauptmenü erfolgt durch `<Q>`.

```
Analysis menue:
Compute a shortest path from the initial state to a target marking.........P
Compute a reachability graph...............................................R
Compute a coverability graph to decide boundedness and coverability.........G
Analyse the previously computed graph......................................A

Compute a base for all S/T-invariants [non-reachability test]..............I
Compute a base for all semipositive S/T-[sub/sur]-invariants................S
Format lines written to INVARI.HLP earlier.................................F
Test place- or transition-vectors for invariant properties..................T

Compute the symmetries of the net..........................................Y

Check the deadlock-trap-, SMD-, SMA-properties [boundedness, liveness]......D
Compute all components [check SMD-,SMA-properties]..........................C
Decide state-machine coverability [for boundedness].........................M

choice >
```

Voranalyse

Bevor das Analysemenü angezeigt wird, führt INA eine Voranalyse des Netzes durch. Dabei wird auch geprüft, welche Funktionen überhaupt verfügbar sind.

Untersucht werden strukturelle Eigenschaften (siehe Seite 56 bis 58). Zu Beginn können Sie Schreiboptionenen setzen. Dazu gehört unter anderem die Ausgabe der statischen Konflikte: `Print the static conflicts?` Mit `<Y>` werden diese ausgegeben.

Beispiel

Für das Netz 1 der drei Programmierer, die sich zwei Terminals teilen, ergibt die Voranalyse folgende Ergebnisse:

```
Information on elementary structural properties:
The net has no bad reachable states.
The net is not statically conflict-free.
The net is pure.
The net is not ordinary.
```

```
The net is not conservative.
The net is not subconservative.
The net is not a state machine.
The net is not free choice.
The net is not extended free choice.
The net is extended simple.
The net is not safe.
The net is not live and safe.
The net is marked.
The net is not marked with exactly one token.
The net is not a marked graph.
The net is not homogenous.
The net has not a non-blocking multiplicity.
The net has no nonempty clean trap.
The net has no transitions without pre-place.
The net has no transitions without post-place.
The net has no places without pre-transition.
The net has no places without post-transition.
Maximal in/out-degree:  3
The net is connected.
The net is strongly connected.
```

INA kann daraus schon einige Eigenschaften folgern:

```
ORD HOM NBM PUR CSV SCF CON SC  FtO tFO FpO pFO MG  SM  FC  EFC ES
 N   N   N   Y   N   N   Y   Y   N   N   N   N   N   N   N   N   Y
DTP CPI CTI  B   SB  REV DSt BSt DTr DCF  L   LV L&S
 ?   ?   ?   ?   ?   ?   ?   N   ?   ?   ?   ?   N
```

Die statischen Konflikte des Netzes sind:

```
Static conflicts:
transition 1.login_Prog1 is in conflict with:
       2.login_Prog2,  3.login_Prog3,
transition 2.login_Prog2 is in conflict with:
       1.login_Prog1,  3.login_Prog3,
transition 3.login_Prog3 is in conflict with:
       1.login_Prog1,  2.login_Prog2,
```

6.4. Erreichbarkeitsanalyse

Die Berechnung des Zustandsgraphen eines Netzes gehört zu den wirksamsten Analysemethoden, die nur dadurch eingeschränkt ist, daß unscheinbare beschränkte Netze sehr viele Zustände haben können. INA verfügt über eine Methodik, die beste Speicherausnutzung bei schnellem Zugriff verspricht.

6.4.1. Pfadberechnung

Diese Funktion beruht auf der teilweisen Berechnung des Zustandsgraphen und gehört deshalb zur Erreichbarkeitsanalyse. Eine Markierung wird auf Erreichbarkeit getestet und eventuell ein Pfad ausgegeben. Der Weg ist ein kürzester in bezug auf die Zahl der Schaltvorgänge (nicht notwendig ein schnellster Weg). Dabei reicht eine Teilmarkierung aus. Die Markierungen der restlichen Plätze werden als nicht definiert betrachtet, d.h. als `don't care` Bedingungen behandelt.

Beachten Sie bei Netzen mit Zeitbewertung, daß transiente Zustände nicht betrachtet werden. Wenn Ihre Markierung nur in einem transienten Zustand vorkommt, wird INA sie nicht finden und sie als unerreichbar qualifizieren, sofern der gesamte Zustandsgraph konstruiert werden kann.

`<P>`	`Compute a shortest path from the initial state to a target marking`

Die zu untersuchende Markierung kann von einer Datei eingelesen werden oder (durch Eingabe von `<esc>`) auch direkt eingegeben werden. Bei einem zeitbewerteten Netz wird danach die Zeitbewertung abgefragt.

Nach dem Eingeben bzw. Einlesen der Zielmarkierung beginnt INA damit, den Zustandsgraphen des Netzes in der Breite zuerst aufzubauen. Die Anzahl der berechneten Zustände wird angezeigt: `States generated`.

Trifft INA auf eine Markierung, die mit der Zielmarkierung auf den definierten Plätzen übereinstimmt, so bricht die Untersuchung ab und der Weg zu dem betreffenden Zustand wird in die Session-Datei geschrieben. Die Zielmarkierung wird auf dem Bildschirm als erreichbar gemeldet: `The marking is reachable`.

Wenn die Zielmarkierung nicht erreichbar ist, gibt es zwei Möglichkeiten: Kann der gesamte Zustandsgraph aufgebaut und gespeichert werden, dann wird die Markierung als unerreichbar gemeldet (`The marking is not reachable`), anderenfalls (bei unbeschränktem Netz immer) bricht INA mit der Meldung `Nodeoverflow` ab und zeigt an, daß keine Entscheidung getroffen werden kann: `No decision possible`.

Der Berechnungsprozeß kann jederzeit durch Eingeben von `<Q>` abgebrochen werden.

Eventuell gefundene „schlechte" bzw. tote Zustände werden für die weitere Analyse berücksichtigt.

Ergebnisanzeige

Am Beispiel des Netzes 1 wird die Berechnung eines Pfades gezeigt. Dabei bedeutet die verwendete Teilmarkierung, daß die Programmierer zwei und drei am Terminal arbeiten.

```
Computation of a shortest path.
target submarking:
   2:1;   3:1;

The marking is reachable.
Trace of length 2
State nr: 1
 PL: 0 1 2 3 4 5 6
 MA: 2 0 0 0 1 1 1

   n1====2.login_Prog2     ==> n3
State nr: 3
 PL: 0 1 2 3 4 5 6
 MA: 1 0 1 0 1 0 1
```

```
   n3====3.login_Prog3     ==> n5
State nr: 5
 PL: 0 1 2 3 4 5 6
 MA: 0 0 1 1 1 0 0


States generated: 6
```

In diesem Beispiel ist zweimaliges Schalten nötig, um von der Anfangsmarkierung zur Zielmarkierung zu gelangen. In der Anfangsmarkierung n1 wird der Zustand n3 durch das Schalten der Transition 2 und in diesem Zustand durch das Schalten der Transition 3 die Zielmarkierung im Zustand n5 erreicht. Einfach gesagt: Falls die beiden Programmierer zwei und drei in der Anfangssituation das Login ausführen, wird die gegebene Zielmarkierung erreicht.

6.4.2. Reduktionen des Zustandsgraphen

Zusätzlich zu einer guten Speicherausnutzung bietet INA optional Möglichkeiten, den Zustandsgraphen schon während der Berechnung zu reduzieren. Ziel der Reduktion ist es, geeignete Teilgraphen zu finden, die bestimmte Eigenschaften des Erreichbarkeitsgraphen repräsentieren. Dabei ist es möglich, daß Knoten dieser reduzierten Graphen als Vertreter für ganze Klassen von Knoten des Erreichbarkeitsgraphen stehen, oder daß nur ein Teil der möglichen Zustandsübergänge betrachtet wird und somit einige Bögen und Knoten des Erreichbarkeitsgraphen nicht berechnet werden. Desweiteren gibt es Darstellungen für die unendlichen Zustandsgraphen unbeschränkter Netze.

Als Beispiel für die Zustandsgraphreduktionen dient das Netz 2 der speisenden Philosophen. Der Zustandsgraph hat 393 Knoten.

Symmetrische Reduktion

Oftmals treten in einem Netzmodell Symmetrien auf, die bei der Analyse ausgenutzt werden können.

Bei der Berechnung eines Zustands– bzw. Überdeckbarkeitsgraphen, können alle Zustände identifiziert werden, die durch eine Symmetrieabbildung ineinander überführt werden. Die Zahl der zu berechnenden Zuständen wird dadurch unter Umständen erheblich reduziert, so daß die Gesamtberechnung (einschließlich der der Symmetrien) schneller sein kann als die des kompletten Zustandsgraphen. Der Nachteil des symmetrisch reduzierten Zustandsgraphen besteht darin, daß er im allgemeinen nur schwächere Aussagen hinsichtlich der Lebendigkeit von Transitionen liefert.

Die Symmetrien werden entweder separat mit der Funktion `<Y>`, oder automatisch (falls dies nicht bereits erledigt wurde) vor der Erzeugung eines symmetrisch reduzierten Graphen berechnet.

```
<Y>     Compute the symmetries of the net
```

INA berechnet die Symmetriegruppe des aktuellen Netzes unter Berücksichtigung eventueller Zeitbewertungen.

Zuerst müssen Sie angeben, ob der Anfangszustand als symmetrisch betrachtet oder ob Fixpunkte für Plätze oder Transitionen gesetzt werden sollen: `Initial state to be symmetric` bzw. `do you want to set fixpoints?` Teilweise werden so andere oder gar keine Symmetrien gefunden und die Berechnung abgebrochen: `Identity is the only symmetry`. Falls dagegen Symmetrien gefunden wurden, so können Sie diese mit `Write the symmetries to the session file` in den Session–Report schreiben, andernfalls werden nur die Äquivalenzklassen geschrieben.

Sollte in einem Teil der Berechnung ein Speicherüberlauf vorkommen, so meldet INA: `Stopped by heap overflow danger`. Die Untersuchung kann abbrechen, weil es mehr Symmetrien gibt, als der Speicher fassen kann: `Sorry, no space for computation`. Die Berechnung kann aber auch mit `<Q>` beendet werden, wenn Sie glauben, über eine hinreichend große Untergruppe der Symmetriegruppe zu verfügen. INA rechnet auf jeden Fall solange, bis eine vollständige Untergruppe berechnet ist.

Bei der Ausgabe der Symmetrien können Sie erkennen, wie die Zeilen berechnet wurden: Einträge, die mit einem Stern (*) versehen sind, sind durch Verkettung aus den Generatoren (Einträge ohne *) entstanden. Die erste Zeile entspricht immer der Identität.

Neben den Symmetrien werden die Zerlegungen der Platz– und Transitionsmengen in Äquivalenzklassen berechnet.

Die Funktion `<Y>` kann auch benutzt werden, um eine einmal berechnete Symmetriegruppe erneut berechnen zu lassen. Dazu müssen Sie nur die Frage `compute the symmetries once again` positiv beantworten. Sie können dabei beispielsweise neue Fixpunkte setzen oder die Anfangsmarkierung berücksichtigen.

Für das Netz 1 sieht die Berechnung wie folgt aus:

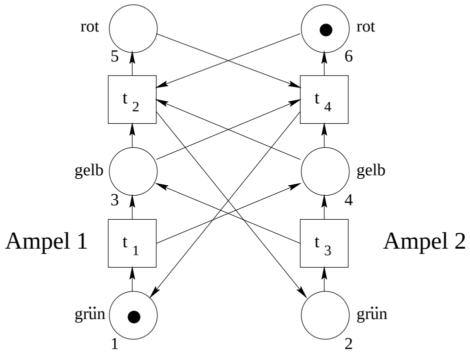
```
Computation of the symmetries:
    1:  0  1  2  3  4  5  6  *  1  2  3  4  5  6
    2:  0  1  3  2  4  6  5  *  1  3  2  4  6  5
    3:  0  2  1  3  5  4  6  *  2  1  3  5  4  6
*   4:  0  2  3  1  5  6  4  *  2  3  1  5  6  4
*   5:  0  3  1  2  6  4  5  *  3  1  2  6  4  5
*   6:  0  3  2  1  6  5  4  *  3  2  1  6  5  4

Classes of equivalent places:
   1:  4.Prog1_in_Pause,  5.Prog2_in_Pause,  6.Prog3_in_Pause,
   2:  1.Prog1_am_Term,  2.Prog2_am_Term,  3.Prog3_am_Term,
   3:  0.Terminal_frei,

Classes of equivalent transitions:
   1:  4.logout_Prog1,  5.logout_Prog2,  6.logout_Prog3,
   2:  1.login_Prog1,  2.login_Prog2,  3.login_Prog3,
```

Zuerst wird die Symmetriegruppe aufgeführt, dann die Äquivalenzklassen: Die Plätze 1, 2 und 3 sind zueinander äquivalent, genauso die Plätze 4, 5 und 6. Platz 0 bildet bezüglich der Zerlegung eine Einermenge. Für die Transitionen gilt, daß zum einen 1, 2 und 3 zum anderen 4, 5 und 6 zueinander äquivalent sind.

Das Netz 14 ist (im Gegensatz zum Netz 1) nur ohne Berücksichtigung der Anfangsmarkierung symmetrisch. Es modelliert zwei Ampeln: Die Plätze entsprechen den Farben der beiden Ampeln (es handelt sich aber nicht um ein gefärbtes Netz), die Transitionen sorgen für die notwendige Verkehrssicherheit.



Beispiel 14: Zwei Ampeln

Ein Ablauf der Symmetrieberechnung dieses Netzes kann wie folgt aussehen:

```
Computing the symmetries
Initial state to be symmetric? Y
Do You want to set fixpoints? N
Identity is the only symmetry!

Compute the symmetries once again? Y
Initial state to be symmetric? N
Do You want to set fixpoints? N

Classes of equivalent places:
   1:  5.rot_eins,  6.rot_zwei,
   2:  3.gelb_eins,  4.gelb_zwei,
   3:  1.gruen_eins,  2.gruen_zwei,

Classes of equivalent transitions:
   1:  2,  4,
   2:  1,  3,
```

Da die Philosophen des Netzes 2 ein symmetrisches Verhalten haben, lohnt dort die Berechnung der Symmetriegruppe: Der symmetrisch reduzierte Erreichbarkeitsgraph hat 47 Knoten.

Zu Details siehe [Sta90, Kapitel 7 (62–73)], [Sta91] und [SS91].

Sture Reduktion

Sture Reduktion ist eine weitere Möglichkeit, die Knotenanzahl des Erreichbarkeitsgraphen zu veringern. Sie kann mit der symmetrischen Reduktion kombiniert werden, ist aber nicht für alle Netztypen verfügbar.

Durch eine Freiheit der Reihenfolge des Schaltens von Transitionen, wie sie durch Nebenläufigkeit impliziert wird, kommt es zu einer Erhöhung der Anzahl erreichbarer Zustände. Bei der Untersuchung von zwei Pfaden zu einem Zustand ist eventuell erkennbar, daß diese sich nur in der Reihenfolge unterscheiden, in der unabhängige, d.h. nebenläufige Transitionen geschaltet werden. Der Zustandsgraph enthält alle Zwischenzustände, obwohl diese teilweise redundant sind. Sture Reduktion hat das Ziel, einen möglichst kleinen Ausschnitt aus dem Zustandsgraphen zu konstruieren. In einer gegebenen Markierung wird nur eine Teilmenge von Transitionen geschaltet. Diese Teilmenge wird so gewählt, daß diese von anderen Transitionen möglichst wenig beeinflußt wird, oder nur in bestimmter, vorher bekannter Art. Diese Vorgehensweise beruht wesentlich auf dem Lokalitätsprinzip, sowie auf der Nebenläufigkeit. Bei der Berechnung reicht es im allgemeinen nicht aus, nur die konzessionierten Transitionen zu betrachten, sondern teilweise auch Transitionen ohne Konzession. Die entstehende Menge erhielt aufgrund ihrer schweren Beeinflußbarkeit den Namen *sture Menge*.

Die wichtigste Eigenschaften, die noch gefolgert werden können, ist die Existenz toter Markierungen und unendlicher Pfade.

Im Netz 2 können Philosophen nebenläufig handeln. Der stur reduzierte Erreichbarkeitsgraph hat 183 Knoten. Unter Berücksichtigung der schon zuvor erwähnten Symmetrie des Netzes ergibt sich mit sturer und symmetrischer Reduktion ein Erreichbarkeitsgraph, der sogar nur aus 28 Knoten besteht. Die symmetrische Reduktion ist sozusagen orthogonal zur sturen, es ergibt sich unter Umständen eine insgesamt drastische Reduktion des Zustandsgraphen, beim Beispielnetz 2 um den Faktor zehn.

Zu Definitionen und Details der Implementation siehe die Diplomarbeit [Pog94].

Überdeckbarkeitsgrahen

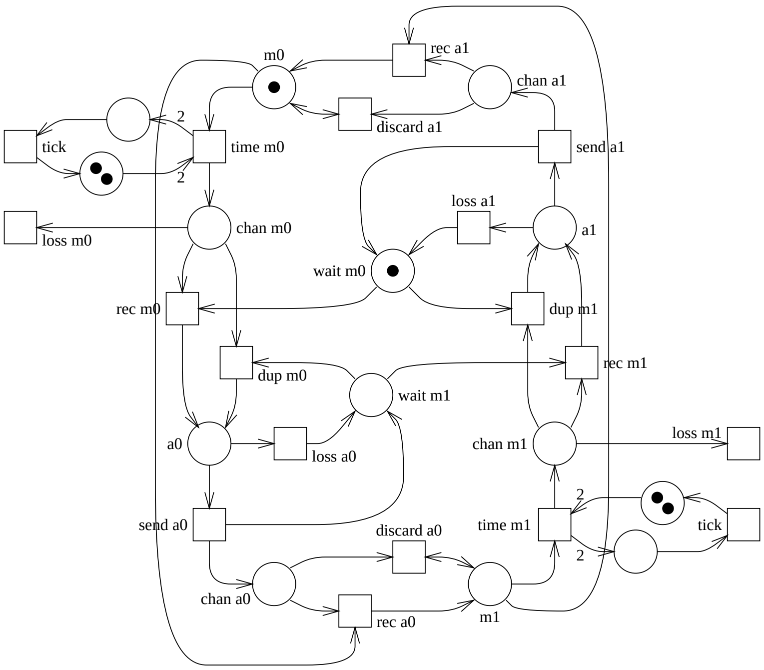
Überdeckbarkeitsgraphen bieten bei unbeschränkten Netzen eine vollständige Übersicht über die Beschränktheit von Plätzen und die Überdeckbarkeit von Markierungen. Ihre Berechnung ist nur für normale Netze ohne Zeitbewertung und Prioritäten unter der normalen Schaltregel möglich und wird daher anderweitig nicht angeboten. INA berechnet den Graphen nach dem Algorithmus von Karp und Miller. Falls das Netz beschränkt ist, dann entspricht der Überdeckbarkeitsgraph dem normalen Zustandsgraphen.

Zu Details siehe [Sta90, Kapitel 5 (43–53)].

Finkels Methode zur Berechnung des minimalen Überdeckbarkeitsgraphen

Finkel betrachtet in [Fin89] Erreichbarkeitsgraphen und Überdeckbarkeitsgraphen als Spezialfälle einer allgemeineren Struktur, sogenannter ω–Zustandsgraphen. Unter allen diesen Zustandsgraphen gibt es einen eindeutig bestimmten, der minimal bezüglich Knotenzahl ist. INA bietet für Netze ohne Zeitbewertung und Prioritäten unter der normalen Schaltregel die Möglichkeit, den Überdeckbarkeitsgraphen nach Finkels Methode zu berechnen und so während des Aufbaus die Knoten– und Kantenzahl zu reduzieren. Für unbeschränkte Netze ergibt sich ein Graph, der weniger Knoten als der klassische Überdeckbarkeitsgraph hat. Für beschränkte Netze bewegt sich der zeitliche Mehraufwand in vertretbaren Grenzen, obwohl keine wirkliche Reduktion erreicht wird. Die Kenntnis des minimalen ω–Zustandsgraphen genügt, um Eigenschaften wie Beschränktheit des Netzes, Überdeckbarkeit von Markierungen oder die Existenz unendlicher Pfade im Erreichbarkeitsgraph zu entscheiden.

Das Netz 15 wird gewöhnlich unter der Schrittschaltregel betrachtet und ist dann beschränkt. Wird dieses Netz ohne Schrittschaltregel analysiert, so modelliert es das `alternating bit protocol` nicht mehr korrekt. Da es dann unbeschränkt ist, wird es hier trotzdem als Beispiel für eine drastische Reduktion des Zustandsgraphen vorgestellt. Die Berechnung des klassischen Überdeckbarkeitsgraphen benötigt mehrere Tage; der Graph hat dann 435316 Knoten. Nach



Beispiel 15: Das alternating bit protocol

FINKELS Methode ergibt sich ein Graph mit 72 Knoten, die Berechnung dauert nur wenige Sekunden.

Zu Definitionen und Details der Implementation siehe die Diplomarbeit [Lü95].

6.4.3. Graphberechnung

In diesem Abschnitt finden Sie eine Beschreibung der Berechnung eines Zustands- bzw. Überdeckbarkeitsgraphen, beginnend mit einer Vorstellung der Optionen, über den allgemeinen Ablauf bis zur Form der Ergebnisausgabe. Am Ende finden Sie eine Beschreibung des optionalen Tests auf Erreichbarkeit bzw. Überdeckbarkeit einer Markierung.

INA bietet folgende Funktionen zur Graphberechnung an:

`<R>      Compute a reachability graph`

Dieses Kommando baut den Zustandsgraphen auf, wobei dieser symmetrisch bzw. stur reduziert berechnet werden kann. Im Anschluß kann optional ein Test auf Erreichbarkeit bzw. Überdeckbarkeit einer Markierung durchgeführt werden.

`<G>      Compute a coverability graph`

Dieses Kommando baut den Überdeckbarkeitsgraphen auf, wobei dieser symmetrisch reduziert bzw. nach Finkels Methode berechnet werden kann. Im Anschluß kann optional ein Test auf Erreichbarkeit bzw. Überdeckbarkeit einer Markierung durchgeführt werden.

Alle Graphberechnungen erstellen die Datei STATES.gra, die als Schnittstelle für die weitere Analyse (siehe Kapitel 6.4.4 ab Seite 72) dient. In dieser Datei wird die Information über die Struktur des Graphen abgelegt. Einige Folgerungen werden unmittelbar im Anschluß an die Berechnung der Zustände gezogen.

Optionen

INA zeigt zu Beginn der Berechnung die Optionen an, unter denen diese durchgeführt werden soll:

```
Current analysis options are:
     no depth restriction ..........................................D
     transition names not to be written .............................W
     place names not to be written ..................................V
     do not print all states .......................................M
     print the dead states..........................................O
     print the bad states...........................................B
     to the session file ...........................................S
     no reachability / coverability test ............................T
```

Sie können folgende Einstellungen vornehmen:

Berechnungsarten

Mit der Berechnungsart können Sie die zuvor vorgestellten Graphreduktionen anwählen. Mit <Y> bzw. <U> können Sie eine symmetrische bzw. sture Reduktion des Graphen einstellen, falls dies für das aktuelle Netz möglich ist. Im Falle eines Erreichbarkeitsgraphen können beide Möglichkeiten (falls Sie angeboten werden) beliebig kombiniert werden, für Überdeckbarkeitsgraphen steht nur die symmetrische Reduktion zur Verfügung. Falls die Symmetriegruppe nicht schon vorher berechnet worden ist, geschieht dies unmittelbar nach der Optionseinstellung. Siehe dazu die Beschreibung des Kommandos <Y> auf Seite 64 im Kapitel 6.4.2.

symmetrical/stubborn reduction

Mit <R> können Sie für Überdeckbarkeitsgraphen zusätzlich eine Reduktion nach der Methode von Finkel auswählen.

reduction by Finkels method

Tiefenbeschränkung

Mit der Option <D> kann die Tiefe festgelegt werden, bis zu der der Zustandsgraph berechnet werden soll. Durch Eingabe der Tiefe 0 wird die Tiefenbeschränkung aufgehoben.

depth restriction

Schreiboptionen

Sie können mit <W>,<V>,<C> und <V> angeben, ob die Platz- bzw. Transitionsnamen (und Farben bei gefärbten Netzen) bei der Ergebnisausgabe aufgeführt werden sollen oder nicht.

write options

Durch Ändern der Option <M> kann festgelegt werden, ob nur tote (<O>) und „schlechte“ (<B>) oder alle Zustände ausgegeben werden, durch <S>, in welche Datei dies geschehen soll: Zur Auswahl steht der Session–Report und eine separate Datei mit der Endung .res, die angefordert wird, sobald die Liste der Zustände berechnet wurde.

Erreichbarkeits– bzw. Überdeckbarkeitstest

`reachability/coverability test`

Durch setzen der Option <T> kann im Anschluß an die Berechnung auf Grund des aufgebauten Graphen eine Markierung auf Erreichbarkeit oder Überdeckbarkeit getestet werden (siehe dazu die folgende Beschreibung).

Ablauf der Berechnung

Nach dem Einstellen der Optionen wird der Graph aufgebaut. Der laufende Fortschritt wird durch die Anzahl der erzeugten Zustände angezeigt. Die Datei `STATES.gra` wird während der Berechnung des Graphen geschrieben und dient zur späteren Analyse.

Die laufende Berechnung kann jederzeit durch Eingabe von <Q> abgebrochen werden. Im Fall, daß aus Platzgründen nicht der gesamte Zustandsgraph erzeugt werden kann, bricht INA mit der Meldung `Nodeoverflow` ab. Die Berechnung eines Zustandsgraphen wird unterbrochen, wenn INA entdeckt, daß das aktuelle Netz unbeschränkt ist (bei unbeschränkter Tiefe). In beiden Fällen ist der berechnete Graph nicht vollständig: `computed graph is not complete`.

Falls Sie eine symmetrische Reduktion angefordert haben und zuvor die Symmetrien nicht vollständig berechnet wurden (`symmetries have not been computed completely`), weil Sie die Berechnung unterbrochen haben oder der Speicherplatz nicht ausgereicht hat, müssen Sie den weiteren Fortlauf bestätigen: `Execute nevertheless?`

Bei der Berechnung des Überdeckbarkeitsgraphen wird im unbeschränkten Fall nicht abgebrochen, sondern die Markierung an den entspechenden Stellen `oo` (ω) gesetzt. INA meldet zu Beginn, welche Konstante die höchste Markenzahl eines beschränkten Platzes sein kann: `It is assumed that all bounded places are 4294836224-bounded`. Diese Konstante ist abhängig von der Implementation und ist auf jeden Fall kleiner als der interne Wert von `oo` (ω — der angegebene Wert entspricht dem der UNIX–Implementation).

Im beschränkten Fall unterscheiden sich die berechneten Graphen nicht, der Algorithmus von Karp und Miller berechnet dann den Zustandsgraphen.

Falls Sie bei der Berechnung des Überdeckbarkeitsgraphen die Methode von Finkel ausgewählt haben, so unterscheidet sich die Bildschirmausgabe während der Berechnung. INA zählt die Anzahl der berechneten Bögen, erst am Ende wird die Zahl generierter Knoten ausgegeben. Die Ausgabe der Zustandsliste ist verkürzt, die Datei `STATES.gra` wird aber analog zu allen anderen Graphberechnungsalgorithmen aufgebaut.

Sollten Sie eingestellt haben, daß die Zustände in eine separate Datei gespeichert werden sollen, so wird am Ende einer Graphberechnung ein `state output file` mit der Standarderweiterung `.res` angefordert. Durch Eingabe von <esc> können Sie dies umgehen, die Informationen der Berechnung werden auf dem Bildschirm angezeigt.

Ergebnisanzeige

`needed capacities`

Als erstes wird Ihnen ein Überblick über die benutzen Kapazitäten gegeben: Für jeden Platz wird angezeigt, wieviele Marken auf ihm während der Berechnung maximal gelegen haben. Bei unvollständiger Zustandsberechnung sind entsprechend nur die bis dahin berechneten Markierungen berücksichtigt (`capacities needed so far`).

`states`

In der Ausgabe (entweder im Session–Report, der separaten Datei oder auf dem Bildschirm) wird jedem Zustand eine laufende Nummer zugeordnet (die Reihenfolge entspricht der Berechnung). Ausgehend vom Zustand 1, dem Anfangszustand (`initial`), wird für jeden Zustand angezeigt, aus welchem Zustand er durch Schalten welcher Transition entstanden ist und wie die Markierung in diesem Zustand aussieht. Eine Beschreibung der Zustandsdarstellung finden Sie im Kapitel 4.2 ab Seite 33.

Für das Netz 1 sieht die Zustandsliste wie folgt aus:

```
State nr: 1 initial
 PL: 0 1 2 3 4 5 6
 MA: 2 0 0 0 1 1 1

State nr: 2  generated from 1  by 1
 PL: 0 1 2 3 4 5 6
 MA: 0 1 0 0 0 1 1

State nr: 3  generated from 1  by 2
 PL: 0 1 2 3 4 5 6
 MA: 1 0 1 0 1 0 1

State nr: 4  generated from 3  by 3
 PL: 0 1 2 3 4 5 6
 MA: 0 0 1 1 1 0 0

State nr: 5  generated from 4  by 5
 PL: 0 1 2 3 4 5 6
 MA: 1 0 0 1 1 1 0
```

Im Beispiel bedeutet `generated from 4 by 5`, daß Zustand 5 aus Zustand 4 durch Schalten der Transition 5 entstanden ist.

Bei der Schaltstrategie maximale Schritte wird zuerst eine Schrittliste ausgegeben, statt Transitionsnummern werden dann Schrittnummern als Übergänge ausgegeben. `steplist`

Erreichbarkeits–/Überdeckbarkeitstest

Falls Sie die entsprechende Option gesetzt haben, können Sie direkt nach der Berechnung der Zustände Erreichbarkeits- bzw. Überdeckbarkeitstests durchführen und (nicht notwendig kürzeste) Wege von der Anfangsmarkierung zu einer Zielmarkierung abfragen. Sollte zu diesem Zeitpunkt schon der Speicher übergelaufen sein, so bricht INA die weitere Untersuchung mit `no space for reachability test` ab. `reachability/coverability test`

INA fragt zuerst nach der Nummer der Markierung, die diese bei dem Aufbau des Zustandsgraphen erhalten hat. Durch Eingabe von `<N>` können Sie selbstverständlich eine Markierung von Datei einlesen oder direkt eingeben. Beachten Sie: Unter Zeitbewertung werden transiente Zustände nicht betrachtet.

Wenn die Markierung nicht erreichbar ist, meldet INA dies und bietet einen Überdeckbarkeittest an. Sollte auch dieser negativ ausfallen, so wird die Markierung auch als nicht überdeckbar gemeldet. Sonst wird eine überdeckende Markierung ausgegeben, durch Beantwortung der Frage `next?` mit `<Y>` können Sie weitere anzeigen, bis INA meldet, daß keine mehr existiert: `no more covering marking found`. Unter Zeitbewertung wird die Markierung bei der Ausgabe als Komponente eines Zustandes bezeichnet.

Falls Sie eine symmetrische Reduktion des Graphen als Option gesetzt haben, so müssen Sie beachten, daß eventuell schwächere Ergebnisse erreicht werden: Es kann vorkommen, daß nur eine äquivalente Markierung gefunden wird: `by symmetry the marking is reachable`. Sollte die Anfangsmarkierung nicht symmetrisch sein, kann im allgemeinen Fall keine Erreichbarkeit entschieden werden: `no conclusion on reachability possible` Für den Überdeckbarkeittest gilt entsprechendes, es kann dann nur eine äquivalente überdeckene Markierung gefunden werden: `no conclusion, but equivalent covering marking found`.

`backtracking` Sollte eine Markierung erreichbar bzw. überdeckbar sein, dann können Sie sich einen Weg zu dieser Markierung berechnen und eventuell in eine separate Datei schreiben lassen. Falls Sie die Frage `exit?` verneinen, können Sie sich weitere Wege anzeigen lassen.

Wollen Sie eine weitere Markierung testen, so beantworten Sie die Abfrage `once more?` mit `<Y>`, sonst erscheint das Analysemenü auf dem Bildschirm.

6.4.4. Graphanalyse

Im folgenden finden Sie eine Beschreibung der weiteren Analyse der zuvor berechneten Graphen. Auf der Basis der Datei `STATES.gra` untersucht INA Eigenschaften, die eine Kenntnis des Inhalts der Zustände (Markierungen und ggf. Uhrenstellungen) nicht erfordern.

`<A>`	`Analyse the previously computed graph`

Dieses Kommando führt die weitere Analyse der zuvor mit den Kommandos `<R>` bzw. `<G>` berechneten Graphen durch.

Ablauf der Berechnung

Zuerst werden die aktuellen Optionen angezeigt, die Sie wieder mit `<Y>` ändern können. Angeboten wird ein Satz von Funktionen, die im folgenden genauer beschrieben werden. Die von Ihnen ausgewählten Funktionen werden in der Reihenfolge nacheinander ausgeführt, wie sie im Optionsmenü angezeigt werden:

```
Current graph analysis options are:
     transition names not to be written .............................W
     draw the graph as a tree .......................................G
     to the session file ............................................S
     output of the graph matrix .....................................X
     computation of the dynamic conflicts ...........................Y
     no computation of the distances ................................D
     no computation of the circuits .................................Z
     computation of terminal SC-components, resetability ............R
     liveness test ..................................................L
```

Nachdem Sie die Optionseinstellung mit `<N>` beendet haben, öffnet INA die Datei `STATES.gra`. Falls diese Datei nicht existiert, meldet INA `not found` und fordert eine Datei mit der Endung `.gra` an: `graph input file`. Diese Datei kann eine von Ihnen umgenannte Datei `STATES.gra` sein, die die Graphstrukturinformationen des aktuellen Netzes enthält. Sie können diese Datei nicht direkt am Terminal eingeben. Sollte die geöffnete Datei nicht mit dem aktuellen Netz übereinstimmen, oder nicht unter den gleichen Optionen (Markenart, Zeitoption, Schaltregel, Benutzung von Prioritäten, Schaltstrategie) erzeugt worden sein, meldet INA: `wrong file` und `graph analysis is impossible` und kehrt ins Analysemenü zurück.

Optionen

Standardmäßig wird der Zustandsgraph als Baum in den Session–Report gezeichnet und die Graphmatrix, die dynamischen Konflikte, und die terminalen starken Zusammenhangskomponenten ausgegeben und die Reversibilität und Lebendigkeit entschieden.

Hier nun eine Beschreibung aller Möglichkeiten:

Einfache Schreiboptionen

Mit `<W>` werden Transitionsnamen und bei gefärbten Netzen mit `<C>` Transitionsfarben in den Session–Report geschrieben oder nicht. `write options`

Baumdarstellung des Zustandsgraphen

Mit `<G>` bestimmen Sie, ob der Graph als Baum gezeichnet werden soll oder nicht. Falls Sie diese Option ausgewählt haben, können Sie zudem mit `<S>` bestimmen, ob dies in den Session–Report oder in eine seperate Datei mit der Endung `.tre` geschehen soll. `graph as tree`

Der Baum zeigt die Zustandsübergänge des Zustandsgraphen, wobei keine Bögen zu bereits früher erreichten Zuständen gezeichnet werden. Ein erneutes Erreichen eines Zustandes wird durch `*` angezeigt. Weiter werden tote Zustände durch `d`, „schlechte" Zustände (Zustände, in denen Fakten konzessioniert sind, siehe Kapitel 6.2.2 auf Seite 60) durch `b` und mit `c` werden Beschneidungen des Graphen durch eine Tiefenbeschränkung (siehe Kapitel 6.4.3 auf Seite 69) angezeigt. Bei Netzen mit Zeitbewertung wird durch `tt` ein Verstreichen von Zeit angezeigt.

Folgende Abbildung zeigt die Baumdarstellung des Zustandsgraphen des Netzes 1:

```
State tree of net 1
 ..d : dead state;  ..* : state reached earlier
 ..b : bad state, the tree has been cut at this node
 ..c : cut by depth restriction

s1
|==1=>s2
|     |==4=>s1*
|==2=>s3
|     |==3=>s4
|     |     |==5=>s5
<1<<<<<<<<<<<<<<<<<
s5
|==2=>s4*
|==6=>s1*
>>>>>>>>>>>>>>>>>>0>
|     |     |==6=>s3*
|     |==5=>s1*
|==3=>s5*
```

Das Schalten der Transition 1 führt vom Anfangszustand $s1$ in den Zustand $s2$ (die Zustandsliste des Netzes finden Sie auf Seite 71), schalten der Transition 4 führt zurück in den Anfangszustand, angezeigt durch `s1*`.

Falls die Breite des Ausgabemediums nicht ausreichend ist (siehe Kapitel 2.5.6 auf Seite 23), werden die Zeilen beim Erreichen des Randes umgebrochen. Durch `<Zahl<<<<<<<` wird eine neue Ebene angedeutet, die Zahl bezeichnet dabei die Ebene, auf der sich das Zeichnen befindet. Gestartet wird immer auf der Ebene 0. Durch `>>>>>>>Zahl>` wird eine Rückkehr auf die Ebene Zahl angedeutet.

Im obigen Beispiel wurde die Zeilenlänge herabgesetzt: Bei Erreichen des Zustandes $s5$ wird das Zeichnen auf Ebene 1 fortgesetzt, Transition 2 und 6 geschaltet und dann wieder zurück auf Ebene 0 gesprungen und dort von $s4$ aus weitergezeichnet. Vollständig entfaltet sieht der Baum an dieser Stelle wie folgt aus:

```
|       |       |==5=>s5
|       |       |       |==2=>s4*
|       |       |       |==6=>s1*
|       |       |==6=>s3*
```

Analog könnten auch weitere Ebenen entfaltet werden.

Sollte die Ausgabe in eine separate Datei umgeleitet worden sein, wird diese nach dem Schreiben geschlossen. Alle weiteren Graphanalysen werden nun wieder in den Session–Report geschrieben.

Schreiben der Schrittschaltliste

`steplist`

Eine erneute Ausgabe der Schrittliste (siehe Seite 71) kann bei der Schaltstrategie maximale Schritte an dieser Stelle mit `<M>` angefordert werden.

Matrixdarstellung des Zustandsgraphen

`graph matrix`

Mit `<X>` bestimmen Sie, ob die Matrix des Zustandsgraphen in den Session–Report ausgegeben werden soll, oder nicht. In dieser Matrix sind die Zustandsnummern die Zeilen und die Transitions– oder Schrittnummern die Spalten. Der Eintrag der i-ten Zeile und j-ten Spalte enthält den Zustand, der erreicht wird, wenn die Transition j im Zustand i geschaltet wird, ein Punkteintrag bedeutet, daß die Transition j im Zustand i keine Konzession hat:

```
Graph matrix:
=============
     | 1 2 3 4 5 6
-----|------------
    1| 2 3 5 . . .
    2| . . . 1 . .
    3| . . 4 . 1 .
    4| . . . . 5 3
    5| . 4 . . . 1
```

Für das Netz 1 zeigt die Matrix, daß im Zustand 1 durch Schalten der Transition 3 der Zustand 5 erreicht wird, die Punkte in der zweiten Zeile, daß nur die Transition 4 Konzession im Zustand 2 hat.

Bei Netzen mit Zeitbewertung wird zusätzlich eine Matrix der Zeiten für die Übergänge zwischen den Zuständen ausgegeben.

Berechnung dynamischer Konflikte

Mit `<Y>` bestimmen Sie, ob INA den Zustandsraum auf in einem dynamischen Konflikt (siehe im Kapitel 6.2.2 auf Seite 60) stehende Transitionen untersuchen soll oder nicht. Zwei in einem Zustand i in dynamischen Konflikt stehende Transitionen j und k werden in folgender Form in den Session–Report geschrieben: i : j # k. `dynamic conflicts`

Für das Netz 1 wird nur eine Zeile ausgegeben: `1: 1 # 2; 1 # 3;` d.h. im Anfangszustand stehen die Transitionen 1 und 2 bzw. 1 und 3 in einem dynamischen Konflikt.

Beachten Sie: Bei einer Reduktion des Zustandsgraphen ist eine Berechnung dynamischer Konflikte nicht möglich. INA meldet: `Sorry, not possible for symmetrically or stubborn reduced graphs` Für Netze, die unter der Schrittschaltregel betrachtet werden, erscheinen folgende Hinweise: `Every two steps enabled at the same state are in conflict` und `Please, confront the graph matrix.` Am Ende der Berechnung wird noch der maximale Ausgangsgrad der Graphknoten ausgegeben. Für das Netz 1 meldet INA: `maximal outdegree of a graph node = 3`.

Berechnung der Distanzen

Mit `<D>` können minimale und maximale Distanzen zwischen Knoten des Zustandsgraphen berechnet werden. Falls Sie die Abfrage `compute minimal distances` mit `<Y>` beantwortet haben, werden die minimalen Entfernungen des am `source node` Prompt eingegebenen Knotens des Graphen berechnet. Die dabei maximal gefundene Entfernung wird ausgegeben, ebenso eine Auflistung nicht erreichbarer Knoten. Mit der Abfrage `print all the distances` werden alle Entfernungen des eingegegebenen Knotens zu allen andereren Knoten aufgelistet. Für das Netz 2 wird z.B. die minimale Entfernung vom Anfangszustand zum Zustand `322` (alle Philosophen haben die rechte Gabel) als `distance(1,322)=5` ausgegeben. Die maximal gefundene Entfernung vom Anfangszustand ist sieben: `maximal computed distance = 7 to node nr. 33`. Eventuell werden nur obere Schranken gefunden. `distances`

Falls Sie die oben erwähnte Abfrage mit `N` beantworten, berechnet INA die maximalen Entfernungen. Zu Beginn wird folgende Warnung ausgegeben:

```
Computation of a maximal (loop-free) distance is very hard, if the graph is
not circuit-free. If a graph with circuits is assumed to be circuit-free, then
only a lower bound for the maximal distance is (faster) computed.
```

Sollten Sie bei einem nicht kreisfreien Graphen nur an einer unteren Schranke interessiert sein, beantworten Sie die Abfrage `assume the graph to be circuit-free` mit `<Y>`. Die Berechnung kann sonst sehr aufwendig sein und nicht in der vorhandenen Zeit zu einem Ergebnis kommen. Sie können die Knotennummern zweier Knoten des Zustandsgraphen bestimmen, zwischen denen die maximale (kreisfreie) Entfernung bestimmt werden soll. Die Anzahl gefundener Pfade wird ausgegeben, die laufende Berechnung kann jederzeit mit `<Q>` abgebrochen werden. Als Ergebnis wird am Ende entweder eine untere Schranke oder die maximale Entfernung ausgegeben, falls die Berechnung diese gefunden hat.

Bitte beachten Sie: Ohne Zeitbewertung wird bei beiden Verfahren die Anzahl geschalteter Transitionen bzw. Schritte gezählt. Bei einer Zeitbewertung haben Sie die Möglichkeit durch Beantwortung der Frage `count times` mit `<Y>` stattdessen die dabei vergangene Zeit betrachten zu lassen.

Berechnung der Kreise

circuits

Mit <Z> können Sie entscheiden, ob Kreise im Zustandsgraphen berechnet und bewertet werden sollen.

Zu Beginn der Kreisberechnung erscheinen folgende Optionen:

```
Work with transition values ..................................V
Print only selected circuits .................................A
Print circuits with minimal length ...........................B
Print circuits with maximal length ...........................C
Print circuits with minimal value ............................D
Print circuits with maximal value ............................E
Print circuits with best efficiency ..........................F
Print circuits with least efficiency .........................G
Print only circuits containing state nr.     0 ..............H
Print not only the first circuit of the selected type ........I
```

Mit <A> können Sie bestimmen, ob alle Kreise ausgegeben werden sollen. Falls Sie diese Option ausgewählt haben, steht nur noch die Option <V> zur Verfügung, sonst können Sie mit <B> bzw. <C> Kreise mit minimaler bzw. maximaler Länge ausgeben lassen. Die Länge ergibt sich aus der Anzahl geschalteter Transitionen bzw. Schritte.

Mit <V> können Sie eine Arbeit mit Bewertungen der Transitionen anfordern. Danach stehen weitere Optionen zur Verfügung: Mit <D> bzw. <E> können Sie Kreise mit minimaler bzw. maximaler Bewertung ausgeben lassen. Die Bewertung eines Kreises ist die Summe der Bewertungen der Transitionen des Kreises. Die Effizienz ergibt sich aus dem Quotienten der Bewertung und der Länge des Kreises. Mit <F> bzw. <G> können Sie Kreise mit minimaler bzw. maximaler Effizienz auswählen.

Mit <H> können Sie Kreise auswählen lassen, die einen bestimmten Zustand enthalten. Durch Eingabe einer Nummer eines Zustandes des Graphen werden nur Kreise ausgegeben, die diesen Zustand enthalten. Der Zustand 0 ist nie im Graph enthalten und dient zur Ausgabe aller Kreise unter Berücksichtigung der restlichen Optionen.

Durch Auswahl von <I> wird nur jeweils der erste Kreis der ausgewählten Typen ausgegeben.

Nach der Optionseinstellung beginnt die Berechnung. Falls Sie eine Arbeit mit Transitionsbewertungen angefordert haben, wird eine Datei mit der Endung .val am values input file Prompt angefordert. Nach Drücken von <esc> können Sie diese Datei auch direkt am Terminal eingeben. In diesem Fall werden die Transitionsbewertungen abgefragt. Nach Eingabe einer Transitionsnummer wird deren Namen angezeigt und die Bewertung (Kosten) angefordert. Nachdem Sie die Eingabe einer Zahl mit <Q> beendet haben, werden die Bewertungen der nicht spezifizierten Transitionen auf Null gesetzt. Sie haben die Möglichkeit, die eingegebene Bewertungen in eine Datei zu schreiben.

Der Fortschritt der Berechnung wird durch einen Zähler berechneter Kreise angezeigt. Selbstverständlich kann jederzeit mit <Q> abgebrochen werden.

Die Kreise werden in den Session–Report geschrieben, dabei wird jeder Kreis mit umgekehrter Orientierung ausgegeben. Mit Nx<-ty--Nz wird angedeutet, daß die Transition mit der Nummer y Teil eines Kreises ist und das Schalten der Transition y vom Zustand z zum Zustand y führt. Die Nummern der Zustände beziehen sich auf die zuvor berechnete Zustandsliste.

Ein Beispiel für das Netz 14 sieht so aus:

```
Circuit nr. 1:
N1<-t4--N4<-t3--N3<-t2--N2<-t1--N1
Corresponding T-invariant:
TR: 1 2 3 4
  : 1 1 1 1
Livelocked transitions: none
Circuit length = 4
```

Der Kreis besteht aus den Transitionen $t1$, $t2$, $t3$ und $t4$ und führt vom Anfangszustand zurück in den selben.

Zusätzlich ausgegeben werden zugehörige Transitionsinvarianten. Diese entsprechen realisierbaren T–Invarianten (siehe Seite 78 im Kapitel 6.5).

Weiterhin werden Transitionen ausgegeben, die sich in einem Livelock befinden. Eine Transition t befindet sich in einem Livelock, wenn es einen Kreis im Zustandsgraphen gibt, indem t nicht schaltet [Sta90, Kapitel 12 (123–133)]. Das zugehörige Netz kann also fortwährend schalten, ohne daß die lebendige Transition t geschaltet wird, t „verhungert“. Besonders deutlich wird dies in den Kreisen des Erreichbarkeitsgraphen des Netzes 2 der speisenden Philosophen, in dem es viele Transitionen gibt, die sich in einem Livelock befinden können.

Terminale starke Zusammenhangskomponenten und Rücksetztbarkeit

`terminal SC-components, resetability`

Mit `<R>` werden die terminalen starken Zusammenhangskomponenten berechnet. Dies sind die starken Zusammenhangskomponenten des Zustandsgraphen, aus denen keine Bögen herausführen. Sie werden in Form eines Intervalls ausgegeben. Die Knoten des Erreichbarkeitsgraphen, die in einem der ausgegebenen Intervalle liegen, sind jeweils durch gerichtete Wege voneinander erreichbar. Tote Zustände bilden immer ein Intervall mit gleicher Ober– und Untergrenze. Für das Netz 2 werden folgende terminale starke Zusammenhangskomponenten berechnet: `1: [6..6]` und `2: [322..322]`.

Falls der gesamte eingelesene Graph stark zusammenhängend ist, die Berechnung zuvor nicht abgebrochen wurde und Sie keine sture Reduktion angefordert haben, kann INA daraus folgern, daß der Graph rücksetzbar (siehe im Kapitel 6.2.2 auf Seite 59) ist.

Bitte beachten Sie: Überdeckbarkeitsgraphen sind nicht rücksetzbar [Sta90, Abbildung 6.3 (60)]. INA gibt eine entsprechende Warnung aus.

Für Netze unter Zeitbewertung wird gegebenenfalls eine Liste der Time–Deadlocks ausgegeben.

Lebendigkeitstest

`liveness test`

Mit `<L>` kann eine Lebendigkeitsanalyse durchgeführt werden. Diese funktioniert nur für vollständig berechnete Zustandsgraphen ohne sture Reduktion. Für unter der maximalen Schrittschaltregel ausgeführte Netze kann die Lebendigkeit von Schritten untersucht werden: `Test liveness of steps?`. Falls das Netz Transitionen enthält, die im Anfangszustand tot sind (siehe Seite 60 im Kapitel 6.2.2), dann bezieht sich die Lebendigkeitsanalyse nur auf Transitionen, die nicht tot sind: `Liveness analysis refers to the net where all dead transitions are removed (considered as facts).` Das Netz kann lebendig sein, wenn alle toten Transitionen als Fakten betrachtet werden, die Eigenenschaft `LV` (Lebendigkeit unter Ignorierung toter Transitionen) wird dann entsprechend gesetzt (siehe auch Seite 60 im Kapitel 6.2.2).

Weitere abgeschwächte Lebendigkeitsbegriffe werden gegebenenfalls auf dem Bildschirm erklärt: Zum einen die schwache Lebendigkeit (Seite 60 im Kapitel 6.2.2: `A transition is weakly live IFF it has a live colour`), zum anderen die kollektive Lebendigkeit (Seite 60 im Kapitel 6.2.2: `A net is collectively live IFF every transition is collectively live at the initial state. A transition t is collectively live IFF for every reachable state m there exists a colour c of t and a state m' reachable from m such that t has concession in colour c at m'`).

Eventuell werden Transitionsmengen ausgegeben, die nicht schwach lebendig sind, bzw. die kollektiv lebendig sind (`the following transitions are not weakly live` bzw. `the set of collectively live transitions`).

6.5. Invariantenanalyse

Die Platz– bzw. Transitionsinvarianten eines Netzes sind die ganzzahligen Lösungen des homogenen linearen Gleichungssystems $x \cdot C = 0$ bzw. $C \cdot y = 0$, wobei C die Inzidenzmatrix des Netzes ist, und werden auch kurz P– bzw. T–Invarianten genannt [Sta90, Kapitel 11 (110–122)]. Die Komponenten von P–Invarianten werden als Gewichte des jeweiligen Platzes aufgefaßt. Die gewichte Markenanzahl ist invariant gegenüber Schaltvorgängen. Die Komponenten von T–Invarianten können als Schaltzahlen der betreffenden Transition betrachtet werden (negative Werte entsprechen dem Rückwärtsschalten). Das Schalten aller Transitionen mit einer der Schaltzahl entsprechenden Häufigkeit führt zur selben Markierung, wie vor dem Schalten.
Jedes beschränkte und lebendige Netz besitzt eine T–Invariante, die in allen Komponenten positiv ist, das Netz ist mit T–Invarianten überdeckbar.

6.5.1. Invariantenberechnung

In diesem Abschnitt finden Sie eine Beschreibung des Analysemenükommandos `<I>`, welches eine Invariantenbasis berechnet und einen schnellen Nichterreichbarkeitstests anbietet. Desweiteren wird die Ausgabe der Inzidenzmatrix und der Invarianten vorgestellt.

In allen Beispielen wird auf das Netz 14 zurückgegriffen, das die Steuerung zweier Ampeln modelliert.

Inzidenzmatrix

`incidence matrix`

Bei der Berechnung von P–Invarianten kann die transponierte, bei T–Invarianten die Inzidenzmatrix selbst ausgegeben werden. Die Matrix zeigt, wie die Transitionen und Plätze verbunden sind [Sta90, Definition 11.1 (110)]. Plätze werden in den Zeilen, die Transitionen in den Spalten aufgeführt (für die transponierte Matrix gilt umgekehrtes).

Für das Netz 14 sieht die transponierte Inzidenzmatrix wie folgt aus:

```
Input matrix (transposed incidence matrix):

              1  2  3  4  5  6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    1 |     -1  0  1  1  0  0
    2 |      0  1 -1 -1  1 -1
    3 |      0 -1  1  1  0  0
    4 |      1  0 -1 -1 -1  1
```

Der Eintrag -1 in der ersten Spalte der ersten Zeile bedeutet, daß die Transition t_1 vom Platz $\text{grün}_{\text{eins}}$ eine Marke abzieht, der Eintrag 1 der sechsten Spalte der vierten Zeile dagegen, daß das Schalten der Transition t_4 eine Marke auf den Platz rot_{zwei} aufbringt.

Falls in der Matrix Einträge mit `00` statt mit 0 versehen sind, so sind diese Einträge durch die Unreinheit des Netzes entstanden (siehe Seite 56 im Kapitel 6.2.1).

Ausgabeformate

Invarianten können in verschiedenen Formen ausgegeben werden, mit der Option `<F>` der Invariantenkommandos wird ein Format ausgewählt. `output format`

Als Beispiel sehen Sie P–Invarianten des Netzes 14.

`<T>     complete table`

Die Basis wird in Form einer Tabelle geschrieben, wobei die Zeilen die Nummern der Invarianten, die Spalten dagegen die Nummern der Plätze bzw. Transitionen sind.

```
Nr.         1  2  3  4  5  6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    1 |     1  1  1  0  0  0 +
    2 |     1  1  0  1  0  0 +
    3 |     0  0  0  0  1  1 +
```

Invarianten, die Sie in diesem Format auf eine separate Datei geschrieben haben, können Sie später mit dem Kommando `<T>` testen (siehe Seite 86). Sie werden deshalb in eine Datei mit der Extension `.inv` gespeichert. Alle anderen Formate ermöglichen keinen späteren Test und werden deshalb in eine Datei mit der Extension `.res` geschrieben.

`<Z>     complete table, zero's not printed`

Die Basis wird ebenfalls in Form einer Tabelle geschrieben, aber statt Nullen werden Leerzeichen ausgedruckt.

```
Nr.         1  2  3  4  5  6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    1 |     1  1  1          +
    2 |     1  1     1       +
    3 |                 1  1 +
```

`<E>     print non-zero entries only`

Die Basisvektoren werden als Liste ihrer von Null verschiedenen Komponenten in der Form `<Knotennummer>:<Wert>` ausgegeben.

```
1 |     1: 1,  2: 1,  3: 1 +
2 |     1: 1,  2: 1,  4: 1 +
3 |     5: 1,  6: 1 +
```

```
<#>     print non-zero entries with names
```

Die Basisvektoren werden ebenfalls als Liste ihrer von Null verschiedenen Komponenten ausgegeben, aber mit ihren Namen versehen.

```
1 |     1.gruen_eins      : 1,
  |     2.gruen_zwei      : 1,
  |     3.gelb_eins       : 1 +
2 |     1.gruen_eins      : 1,
  |     2.gruen_zwei      : 1,
  |     4.gelb_zwei       : 1 +
3 |     5.rot_eins        : 1,
  |     6.rot_zwei        : 1 +
```

Bitte beachten Sie: Semipositive Invarianten werden bei der Ausgabe durch ein + gekennzeichnet (siehe auch Seite 82).

Basisberechnung

Die P– bzw. T–Invarianten bilden einen linearen Raum, der gegenüber (komponentenweiser) Addition, Multiplikation mit einer ganzen Zahl und Division durch den größten gemeinsamen Teiler aller Komponenten abgeschlossen ist. Dieser Raum wird durch endlich viele Basisvektoren aufgespannt, enthält aber im allgemeinen unendlich viele Invarianten. Mit dem Kommando `<I>` wird eine Basis berechnet.

```
<I>     Compute a base for all S/T-invariants
```

Zuerst können Sie die angezeigten Optionen ändern:

```
computation of place-invariants.................................P
output (transposed) incidence matrix............................O
output format = complete table..................................F
```

Mit der Option `<P>` wählen Sie zwischen der Berechnung von Platz– und Transitionsinvarianten, mit `<O>` können Sie bestimmen, ob die (transponierte) Inzidenzmatrix ausgegeben werden soll und mit `<F>` bestimmen Sie die Form, in der Invarianten ausgegeben werden sollen.

Danach berechnet dieses Kommando eine Basis für den Raum aller Invarianten des gewählten Typs. INA meldet, ob Invarianten gefunden wurden, und zieht eventuell daraus weitere Folgerungen. Das Programm entscheidet, ob das Netz mit Invarianten des gewählten Typs überdeckt ist, d.h. ob eine in allen Komponenten positive Invariante existiert. Falls diese tatsächlich berechnet wurde, wird sie auch ausgegeben.

Falls P–Invarianten gefunden wurden, wird ein schneller Nichterreichbarkeitstest angeboten: `Test a marking for non-reachability?` Eine Beschreibung dieses Tests finden Sie auf Seite 81.

Am Ende werden folgende Zeilen auf dem Bildschirm ausgegeben: `You may display the results by typing 'F <cr> Y <esc> '` und `results by other format (f), exit (e).`

Mit <E> kehren Sie ins Analysemenü zurück, mit <F> werden erneut die vier möglichen Ausgabeformate gezeigt. Falls Sie sich die Invarianten auf dem Bildschirm ausgeben lassen möchten, so drücken Sie jetzt einfach <cr>. Beantworten Sie die Frage `Open a separate output file` mit <Y> und geben danach <esc> ein, so werden die berechneten Invarianten wie gewünscht auf dem Bildschirm ausgegeben. Falls Sie dagegen die Ausgabe in eine separate Datei umleiten wollen, so geben Sie einen Dateinamen ein, die Endung `.inv` bzw. `.res` wird automatisch angehängt. Falls gewünscht, kann selbstverständlich auch das Ausgabeformat vorher geändert werden. Die Abfrage eines Nichterreichbarkeitstests erscheint erneut, danach ebenso die eben vorgestellten Zeilen. Falls Sie die Ergebnisse in eine separate Datei geschrieben haben und erneut <F> eingeben, wird zuerst die Frage `Close separate output file ?` gestellt. Falls Sie mit <Y> antworten, wird die Datei geschlossen und Sie können eine andere Ausgabedatei öffnen, sonst wird die Ausgabe an die bereits geöffnete Datei angehängt.

Beispiel für eine Invarianteninterpretation

Die oben aufgeführte Invariantenbasis für das Netz 14 und dessen Anfangsmarkierung besagt, daß

1. die Summe der Marken der Plätze $\text{grün}_{\text{eins}}$, $\text{grün}_{\text{zwei}}$ und $\text{gelb}_{\text{eins}}$ immer gleich eins ist,
2. die Summe der Marken der Plätze $\text{grün}_{\text{eins}}$, $\text{grün}_{\text{zwei}}$ und $\text{gelb}_{\text{zwei}}$ ebenfalls immer gleich eins ist,
3. die Summe der Marken der Plätze rot_{eins} und rot_{zwei} ebenfalls immer gleich eins ist

und somit

- genau eine Ampel rot zeigt,
- genau eine Ampel grün zeigt,
- eine Ampel nur gelb zeigt, wenn keine der beiden Ampeln auf grün geschaltet ist, d.h. eine Ampel zeigt genau dann gelb, wenn die andere Ampel auf rot–gelb geschaltet ist.

Als T–Invariantenbasis ergibt sich $1 \cdot t_1, 1 \cdot t_2, 1 \cdot t_3, 1 \cdot t_4$, d.h. wenn alle vier Transitionen geschaltet haben, ist die Anfangsmarkierung wiederhergestellt.

Für das Netz 14 können durch die Invariantenanalyse die Eigenschaften `CPI`, `CTI`, `SB` und `B` gefolgert werden.

Schneller Nichterreichbarkeitstest

Falls die Berechnung von P–Invarianten zu einem Ergebnis geführt hat, wird Ihnen ein schneller Nichterreichbarkeitstest angeboten. Mit den berechneten Invarianten kann INA dies einfach überprüfen, da folgende Gleichung gilt: Für die Anfangsmarkierung m_0, eine Invariante x und eine Markierung m ist immer $m_0 \cdot x = m \cdot x$, falls m von m_0 erreichbar ist. Für eine angeforderte Markierung wird diese Gleichung für alle berechneten Basisinvarianten getestet, eventuell werden die Nummern der die Gleichung verletzenden P–Invarianten ausgegeben.

`non-reachability test`

Anhand des Netzes 14 wird im folgenden ein solcher Nichterreichbarkeitstest gezeigt, der die zuvor berechneten P–Invarianten als Ausgangsbasis benutzt:

```
Non-reachability test of the marking m:
  p :  1  2  3  4  5  6
----------------------
m(p):  1  1  .  .  .  .

The marking is not reachable from m0.
invariant nr.  3
invariant nr.  2
invariant nr.  1

Non-reachability test of the marking m:
  p :  1  2  3  4  5  6
----------------------
m(p):  1  .  .  .  1  .

No conclusions possible.

Non-reachability test of the marking m:
  p :  1  2  3  4  5  6
----------------------
m(p):  .  .  1  1  .  .

The marking is not reachable from m0.
invariant nr.  3
```

Wie Sie sehen können, ist der Zustand „beide Ampeln zeigen grün“ nicht erreichbar, er verstößt gegen alle drei berechneten P–Invarianten. Weiterhin kann nicht entschieden werden, ob die Markierung „Ampel 1 zeigt gleichzeitig grün und rot“ nicht erreicht werden kann. Die Markierung „beide Ampeln zeigen gelb“ verstößt gegen die Invariante mit der Nummer drei, die besagt, daß genau eine Ampel rot zeigen muß, und ist deshalb nicht erreichbar.

6.5.2. Semipositive Invarianten

In diesem Abschnitt finden Sie eine Beschreibung der Berechnung einer Basis semipositiver Invarianten mit dem Analysekommando `<S>` und die Formatierung der berechneten Basis mit `<F>`. Beachten Sie auch die Bemerkungen zur Komplexität und die zusätzlich zur Verfügung stehenden Ausgabeformate.

Semipositive (Sub/Sur)–Invarianten

Für viele Anwendungen von Invarianten sind nur die semipositiven (echten) [Sta90, Definition 11.3/4 (112,114)], d.h. Invarianten, die in allen Komponenten ≥ 0 sind, interessant. Ein Platz ist z.B. beschränkt, wenn sein Gewicht in einer semipositiven P–Invariante positiv, d.h. > 0 ist [Sta90, Satz 11.9.1 (116)].

Neben semipositiven Invarianten können auch semipositive Sub– bzw. Sur–Invarianten berechnet werden: Beispielsweise ist ein Vektor y eine semipositive Sur–T–Invariante, wenn $C \cdot y \geq 0$. Ein Netz, das eine solche nichttriviale Invariante besitzt, kann nicht bei jeder Anfangsmarkierung beschränkt (strukturell beschränkt – siehe Kapitel 6.2.2) sein, weil y der Zählvektor eines Wortes ist, das beim Schalten aus der Markierung m eine Markierung $m' = m + C \cdot y$ erzeugt, die größer als m ist.

Bemerkungen zur Komplexität

Das Problem, eine Basis oder ein Erzeugendensystem für den Halbraum aller semipositiven Invarianten zu berechnen, ist von wesentlich größerer Komplexität als das Problem, eine Basis für alle Invarianten zu bestimmen. Es gibt Netze, die exponentiell viele minimale Invarianten haben. Daher sei Ihnen empfohlen, auch wenn Sie sich nur für semipositive Invarianten interessieren, zunächst mit `<I>` eine Basis für alle Invarianten zu berechnen, beim Ausschreiben der Basis werden die semipositiven Invarianten durch ein `+` gekennzeichnet. Unter Umständen genügen die in dieser Weise gefundenen semipositiven Invarianten bereits, um die beabsichtigten Schlüsse zu ziehen.

Interaktive Berechnung

Aufgrund der Kompliziertheit des Problems wird es INA in vielen Fällen nicht möglich sein, in der zur Verfügung stehenden Zeit und mit dem verfügbaren Speicherplatz ein vollständiges Erzeugendensystem zu berechnen. Im Modus „interaktiv“ können Sie den Fortgang der Berechnung durch Angabe der als nächstes zu bearbeitenden Spalte der Matrix steuern und so partielle Lösungen erhalten. Sollten Sie sich bei der Berechnung von P–Invarianten für bestimmte Plätze interessieren, so können Sie dies erreichen, indem Sie die Nummern der Vor– und Nachtransitionen dieser Plätze bevorzugen. Dadurch erhalten Sie unter Umständen wenigstens einen Teil der Platzinvarianten, die diese Plätze berühren, bevor Sie die Untersuchung mit `<Q>` aus Zeitgründen stoppen. `interactive/ automatic mode`

Ausgabeformate

Auch für die Ausgabe semipositiver Invarianten stehen verschiedene Formate zur Verfügung. Neben den bereits in 6.5.1 vorgestellten Formaten sind dies (als Beispiele wieder P–Invarianten des Netzes 14): `output format`

`<S>     print supports`

Dieses Format zeigt die Invarianten in Form ihres Supports (Trägers), d.h. über die Nummern der positiven Komponenten des Vektors [Sta90, Definition 11.5 (119)].

```
1 |     1,  2,  3
2 |     1,  2,  4
3 |     5,  6
```

`<N>     print supports with names`

Es wird ebenfalls der Support ausgegeben, aber die Komponenten werden mit ihren Namen versehen.

```
1 |     1.gruen_eins      ,
  |     2.gruen_zwei      ,
  |     3.gelb_eins
2 |     1.gruen_eins      ,
  |     2.gruen_zwei      ,
  |     4.gelb_zwei
3 |     5.rot_eins        ,
  |     6.rot_zwei
```

Basisberechnung semipositiver (Sub/Sur)–Invarianten

Dieses Kommando berechnet die unformatierte Invariantenbasis.

```
<S>     Compute a base for all semipositive S/T-[sub/sur]-invariants
```

Zuerst können Sie die angezeigten Optionen ändern:

```
computation of place-invariants.................................P
no computation of subinvariants.................................B
no computation of surinvariants.................................R
covered sub/surinvariants not to be deleted.....................D
run automatically...............................................I
output format = complete table..................................F
```

Mit der Option `<P>` wählen Sie zwischen der Berechnung von Platz– und Transitionsinvarianten, mit `<I>` wählen Sie den interaktiven bzw. automatischen Modus und mit `<F>` bestimmen Sie wieder die Form, in der Invarianten ausgegeben werden sollen. Außerdem können Sie durch Eingabe von `<B>` bzw. `<R>` bestimmen, ob Sub– bzw. Sur–Invarianten berechnet werden sollen. Beides kann nicht gleichzeitig ausgewählt werden. Mit der Option `<D>` können Sie weiterhin bestimmen, ob überdeckte Sur–/Sub–Invarianten während der Berechnung gelöscht werden sollen.

Falls INA zuvor berechnete Symmetrien im Speicher hat, können diese benutzt werden: `use symmetries?` Nach der Optionseinstellung beginnt die Berechnung. Im interaktiven Modus erhalten Sie einen Überblick über die positiven und negativen Einträge in den Spalten der aktuellen Matrix und die zu erwartende Änderung. Bei der Berechnung von T–Invarianten entsprechen die Spalten den Plätzen, bei P–Invarianten den Transitionen. INA erwartet die Eingabe einer Spaltennummer. Mit `<P>` oder `<cr>` kann die vorgeschlagene Spalte (`proposal for next to process`) ausgewählt, mit `<A>` in den automatischen Modus gewechselt und mit `<Q>` die Berechnung unterbrochen werden. Im automatischen Modus können Sie jederzeit durch Eingabe von `<I>` in den interaktiven Modus zuruckkehren.

Die berechneten Invarianten, die im allgemeinen kein minimales System bilden, werden automatisch in die Datei `INVARI.hlp` geschrieben. Auf dem Ausgabemedium muß (besonders unter DOS) ausreichend Speicherplatz zur Verfügung stehen.

Die Berechnung eines minimalen Erzeugendensystems und die Formatierung der Invarianten wird mit der Funktion `<F>` vorgenommen, die nach dem Ende der Berechnung automatisch von INA gerufen wird. Eine Fortführung der Beschreibung der Invariantenberechnung finden Sie deshalb im nächsten Abschnitt.

Formatierung von Invarianten

Diese Prozedur dient zur Ausgabeaufbereitung, insbesondere reduziert sie das zuvor mit `<S>` berechnete System von Vektoren.

```
<F>     Format lines written to INVARI.hlp earlier
```

Der Aufruf muß unter den gleichen Optionen wie die Erzeugung der Datei `INVARI.hlp` erfolgen. Diese Datei wird von INA automatisch geöffnet und eingelesen, initiale Reduktionsoptionen werden entsprechend der Art der Vektoren automatisch gesetzt.

Zu Beginn fragt INA, ob eine gewisse Zahl von Vektoren in der Datei übergangen werden sollen: `skip a certain number of lines?` Das kann dann sinnvoll sein, wenn die Datei mehr Vektoren enthält, als in den Speicher passen. In diesem Fall kann die Formatierung abschnittweise vorgenommen werden. Nach der Eingabe von `<Y>` können Sie die Anzahl der zu übergehenden Zeilen bestimmen.

Falls die Datei `INVARI.hlp` nicht existiert, meldet INA `not found` und fordert eine Datei mit der Endung `.hlp` an: `lines input file`. Diese Datei kann eine von Ihnen umgenannte Datei `INVARI.hlp` sein, die die unformatierten Invarianten des aktuellen Netzes enthält. Sie können diese Datei nicht direkt am Terminal eingeben.

Sollte die geöffnete Datei nicht mit dem aktuellen Netz übereinstimmen, so meldet INA: `INVARI.hlp and actual net do not match`. INA kehrt in diesem Fall in das Analysemenü zurück. Gleiches geschieht, wenn die geöffnete Datei überhaupt keine Invariantendatei ist: `wrong file`.

Folgende Optionen können Sie für die Formatierung setzen:

```
Delete repeated occurrences of the same support.................O
Do not delete covered sub/sur/invariants........................C
Delete strict invariants covering sub/sur/invariants............I
Reset to default reductions.....................................R
```

Beim Öffnen der Datei werden Voreinstellungen gesetzt, die gewährleisten, daß keine Informationen verloren gehen. Als Warnung gibt INA deshalb `NOTE that information may be lost by non-default reductions` aus, mit `<R>` können Sie nach Änderungen die Voreinstellung wiederherstellen. Mit `<O>` können Sie einstellen, ob mehrfache Vorkommen des gleichen Supports gelöscht werden sollen, oder nicht. Mit `<C>` bestimmen Sie über die Löschung überdeckter Sur–/Sub–/Invarianten, mit `<I>` über das Löschen von strikten Invarianten, die Sur–/Sub–/Invarianten überdecken.

Nach der Optionseinstellung zeigt INA den Fortgang der Formatierung an, indem die eingelesenen Zeilen gezählt werden: `reading line`. Falls eine bestimmte Anzahl von Zeilen übergangen werden sollte, so werden diese nicht mitgezählt. Das Einlesen können Sie jederzeit mit `<Q>` abbrechen.

Nach eventuellen Warnungen, wie z.B. `results may be incomplete` oder `covered solutions may be lost (option "c ")`, werden einige erste Folgerungen gezogen und die Ergebnisse in den Session–Report geschrieben: `writing solutions`. Das Schreiben können Sie ebenfalls mit `<Q>` abbrechen. Danach erscheint wieder folgende Zeile: `results by other format (f) or exit (e)`. Falls Sie `<E>` eingeben, kehrt INA in das Analysemenü zuruck, mit `<F>` können Sie die Formatierung mit anderen Optionen erneut durchführen oder das Ausgabeformat ändern. Ebenso können Sie eine Ausgabedatei bestimmen oder die Ausgabe mit `<esc>` auf den Bildschirm umleiten. Im Format `complete table` in eine `.inv` Datei geschriebene Vektoren können später mit `<T>` getestet werden (siehe Seite 79 und 86).

6.5.3. Invariantentest

Das Kommando `<T>` dient zum Testen von Vektoren und besteht aus mehreren Unterkommandos.

<T> Test place- or transition-vectors for invariant properties

Nach der Abfrage, ob P– bzw. T–Vektoren getestet werden sollen, berechnet INA die (transponierte) Inzidenzmatrix des Netzes. Anschließend erscheint folgendes Menü:

```
Please, select a sequence of functions
(files:      o, o1, o2, o3, o4, c, c1, c2)
(read/write: r, r1, w, k)
(results:    p, b, b1, q, q1)
(special:    f, m, s)
(exit:       e )
       --- h  for help  ---   <del> = <?>   :
```

Sie können nun eine Folge von Bearbeitungsschritten angeben, die mit <cr> abzuschließen ist und danach von INA abgearbeitet wird. Selbstverständlich können Sie auch nur ein einzelnes Kommando abarbeiten lassen, Sie müssen dann nur nach der Eingabe des entsprechenden Buchstabens ebenfalls <cr> drücken, INA kehrt nach erfolgreicher Arbeit zurück in das eben gezeigte Menü. Eine Beschreibung der einzelnen Unterkommandos finden Sie in den folgenden Abschnitten.

Mit dem Unterkommando <S> (siehe Seite 89) innerhalb einer Folge von Bearbeitungsschritten können Sie die bis dahin abgearbeite Folge wiederholen lassen und so z.B. mehrere Vektoren nacheinander auf die gleiche Art testen.

Dateien öffnen

<O> open files (input and output)

Mit diesem Unterkommando bestimmen Sie die Ein– und Ausgabedatei.

Als Eingabedatei können Sie mit den Analysemenükommandos <I> bzw. <S> berechnete Invarianten, die Sie im Format `complete table` (siehe Seite 79) auf eine Datei mit der Extension `.inv` geschrieben haben, bestimmen. Warnung: Versuchen Sie nicht, Invariantendateien, die Sie mit den hier beschriebenen Unterkommandos geschrieben haben, später wieder einzulesen, INA bricht in diesem Fall ab.

Beim Öffnen der Datei wird ein Konsistenztest durchgeführt: INA prüft, ob die in der geöffneten Datei gespeicherten Vektoren zu dem Netz im internen Speicher passen. Dies geschieht mit Hilfe der Kopfzeilen, die automatisch beim Speichern von Vektorendateien im Format `complete table` (siehe Seite 79) mit den Analysemenükommandos <I> bzw. <S> geschrieben wurden. Sollte der Konsistenztest negativ verlaufen, gibt INA entsprechende Hinweise und schließt die Datei.

Normalerweise werden alle berechneten Informationen in den Session–Report geschrieben. Durch Angabe einer separaten Datei mit der Extension `.res` können Sie diese Ausgaben umleiten. Selbstverständlich können Sie mit <esc> auch eine Ausgabe auf den Bildschirm anfordern.

<O><1> open input file only

Wie <O>, aber es wird nur die Eingabedatei geöffnet.

<O><2> open output file only

Wie <O>, aber es wird nur die Ausgabedatei geöffnet.

<O><3>   open input file with restricted consistency test

Dieses Kommando eröffnet nur die Eingabedatei, der Konsistenztest ist aber dahingehend abgeschwächt, daß nur die Anzahl der Plätze bzw. Transitionen mit der Länge der Kopfzeile verglichen wird.

<O><4>   open input file without consistency test

Mit diesem Kommando können Sie eine Datei ohne Kopfzeilen öffnen. Führen Sie es aus, wenn INA beim Öffnen einer Datei mit einem anderen Kommando entsprechende Hinweise gibt.

Dateien schließen

Die Eingabedatei wird automatisch geschlossen, wenn Fehler auftreten, das Ende der Datei erreicht wird, oder (zusammen mit der Ausgabedatei) wenn der Invariantentest beendet wird.

<C>   close files (input and output)

Mit diesem Kommando können Sie die Ein– und Ausgabedatei von Hand schließen.

<C><1>   close input file only

Wie <C>, aber es wird nur die Eingabedatei geschlossen.

<C><2>   close output file only

Wie <C>, aber es wird nur die Ausgabedatei geschlossen.

Leseoperationen

<R>   read vector

Falls Sie vorher mit einem <O>–Kommando eine Eingabedatei geöffnet haben, liest dieses Kommando einen Vektor von dieser Datei, sonst können Sie direkt einen Vektor eingeben: Please, give a vector. Für die Terminaleingabe sind zwei Modi vorgesehen: jump to next non-zero component mit <J> und sequential mode mit <S>. Im sequentiellen Modus fordert INA die Vektorkomponenten des Vektors sukzessive an. Im springenden Modus können Sie zuerst eine Knotennummer und dann den Wert der entsprechenden Komponente angeben. Alle übersprungenen Komponenten werden gleich null gesetzt, ein zurücksetzen ist nicht möglich. Die Abfrage startet im sequentiellen Modus und kann jederzeit durch Eingabe von <J> bzw. <S> anstelle der erwarteten Zahl im jeweils anderen Modus fortgesetzt werden. Mit <N> können Sie am Ende alle verbleibenden Komponenten gleich Null setzen: set remaining components to zero. Mit <Q> können Sie die Eingabe abbrechen, der Vektor ist unter Umständen dann unvollständig.

<R><1>   read vector without terminal protocol

Beim Einlesen von Datei mit <R> wird der eingelesene Vektor auf dem Bildschirm protokolliert. Mit dem Kommando <R><1> unterbleibt diese Protokollierung. Bei Eingabe der Komponenten am Terminal ändert sich nichts.

Schreiboperationen

`<W>     write vector`

Falls Sie mit einem `<O>`–Kommando eine Ausgabedatei geöffnet haben, schreibt dieses Kommando einen Vektor. Warnung: Bitte versuchen sie nicht, einen gespeicherten Vektor später wieder einzulesen, INA bricht in diesem Fall ab.

`<K>     write headlines for vector`

Mit diesem Kommando können Sie eine Kopfzeile in eine geöffnete Ausgabedatei schreiben. Rufen Sie dieses Kommando einmal auf, bevor Sie mit dem Schreiben von Vektoren beginnen.

`<F>     change output format`

Mit diesem Kommando ändern Sie das Ausgabeformat der Invarianten: Es stehen die gleichen Möglichkeiten zur Verfügung, die schon in den Abschnitten 6.5.1 und 6.5.2 auf den Seiten 79 bzw. 83 beschrieben wurden.

Testfunktionen

Als Beispiel für den Test einer Invariante wird auf die bereits im Abschnitt 6.5.1 auf Seite 81 vorgestellte Invariante „eine Ampel zeigt genau dann gelb, wenn die andere Ampel auf rot–gelb geschaltet ist“ zurückgegriffen. Ein entsprechende Vektor ist `0 0 1 -1 1 1`, er wird im folgenden getestet.

`<P>     give properties of the vector`

Dieses Kommando prüft den gerade eingegebenen bzw. zuletzt eingelesenen Vektor, ob er semipositiv ist, und ob er Invariante, Sub–Invariante bzw. Sur–Invariante ist. Das Zeichen `+` steht für ja, `-` für nein.

Der Vektor `0 0 1 -1 1 1` hat folgende Eigenschaften:

```
The place vector under test is not semipositive.
It has the following properties:

place invariant:        +
place subinvariant:     +
place surinvariant:     +
```

`<B>     give results of equations`

Mit diesem Kommando werden die Gleichungsresultat für den Vektor angezeigt.

```
equation for transition  1.              gives result:     0
equation for transition  2.              gives result:     0
equation for transition  3.              gives result:     0
equation for transition  4.              gives result:     0
```

<B><1> give equations with non-zero results

Dieses Kommando gibt einen Überblick über die Transitionen (im Fall von P–Vektoren) bzw. Plätze (im Fall von T–Vektoren), deren zugehörige Gleichung bei dem aktuellen Vektor nicht erfüllt ist, d.h. nicht Null ergibt.

<Q> show equations

Dieses Kommando arbeitet genauso wie das Unterkommando <B>, es werden aber zusätzlich die Einzelheiten der Berechnung ausgegeben.

Folgendes Beispiel zeigt einen Teil der Ausgabe:

```
equation for transition  4.                :
  place             equation * vector              sum
   1    |          1   *    0 =     0   ==>       0
   2    |          0   *    0 =     0   ==>       0
   3    |         -1   *    1 =    -1   ==>      -1
   4    |         -1   *   -1 =     1   ==>       0
   5    |         -1   *    1 =    -1   ==>      -1
   6    |          1   *    1 =     1   ==>       0
```

<Q><1> show single equation

Mit diesem Kommando können Sie sich eine einzelne Gleichung anzeigen lassen. Sie werden nach einer Transition (im Fall von P–Vektoren) bzw. einem Platz (im Fall von T–Vektoren) gefragt, die zugehörige Gleichung wird analog zum vorherigen Kommando ausgegeben.

Weitere Funktionen

<M> show (transposed) incidence matrix

Dieses Kommando zeigt die (transponierte) Inzidenzmatrix.

<S> repeat function sequence

Mit diesem Unterkommando können Sie innerhalb einer Folge von Bearbeitungsschritten die bis dahin abgearbeite Folge wiederholen lassen. Wird in der gerade aktuellen Folge <S> erreicht, so fragt INA: Repeat the function sequence?. Antworten Sie mit <Y>, so wird die Folge erneut von Anfang an ausgeführt.

Auf diese Art können Sie z.B. mehrere Vektoren nacheinander auf die gleiche Art testen: Zuerst öffnen Sie die Ein– und Ausgabedatei mit <O>, dann geben Sie als eine Sequenz <R>, dann die entsprechenden Testkommandos, dann das Wiederholungszeichen <S> und zum Schluß <C> zum Schließen der Dateien ein. Die Testkommandos werden nun solange auf die nacheinander eingelesene Vektoren ausgeführt, bis Sie <N> eingeben, danach werden die Dateien geschlossen

<H> help

Mit diesem Unterkommando erhalten Sie eine kurze Übersicht aller verfügbaren Funktionen.

`<E>      exit`

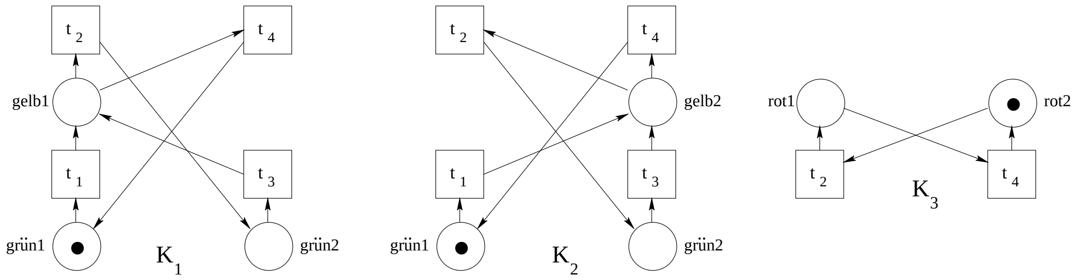
Dieses Kommando beendet den Invariantentest. INA kehrt in das Analysemenü zurück. Eventuell geöffnete Dateien werden zuvor geschlossen.

6.6. Strukturanalyse

Die Funktionen zur strukturellen Analyse werden nur in den Fällen angeboten, in denen die Petri–Netz–Theorie es gestattet, aus erkannten Strukturen auf dynamische Eigenschaften des Netzes zu schließen. Als Grundvoraussetzung müssen die Optionen wie folgt gesetzt sein: Keine Zeitbewertung, normaler Schaltmodus, keine Prioritäten und Schalten einzelner Transitionen. Weitere Bedingungen für die Verfügbarkeit der Kommandos ergeben sich durch die bereits durchgeführte Voranalyse des aktuellen Netzes.

In allen Beispielen dieses Abschnittes wird auf das Netz 14 zurückgegriffen, das die Steuerung zweier Ampeln modelliert.

Es hat folgende stark zusammenhängende Komponenten, die Zustandsmaschinen sind:



Beispiel 16: Drei stark zusammenhängende Zustandsmaschinen–Komponenten des Netzes 14

Die Plätze der Komponenten entsprechen auch den minimalen Deadlocks und den zugehörigen maximalen Fallen, die Deadlock–Falle–Eigenschaft ist also erfüllt.

Das Netz ist nicht zustandsmaschinenauswählbar, da eine Vorplatzauswahl [Sta90, Definition 16.3 (182)] existiert, die keine stark zusammenhängende Zustandsmaschinen–Komponente auswählt. Die oben abgebildeten Komponenten $K_1 = \{\text{grün}_{\text{eins}}, \text{grün}_{\text{zwei}}, \text{gelb}_{\text{eins}}\}$, $K_2 = \{\text{grün}_{\text{eins}}, \text{grün}_{\text{zwei}}, \text{gelb}_{\text{zwei}}\}$ und $K_3 = \{\text{rot}_{\text{eins}}, \text{rot}_{\text{zwei}}\}$ haben die Nachtransitionen $T_1 = \{t_1, t_2, t_3, t_4\}$, $T_2 = \{t_1, t_2, t_3, t_4\}$ bzw. $T_3 = \{t_2, t_4\}$. Die Vorplatzauswahl α mit $\alpha(t_1) = \text{grün}_{\text{eins}}$, $\alpha(t_2) = \text{gelb}_{\text{zwei}}$, $\alpha(t_3) = \text{grün}_{\text{zwei}}$ und $\alpha(t_4) = \text{gelb}_{\text{eins}}$ wählt keine Komponente aus, da $t_2 \in T_1$, aber $\alpha(t_2) = \text{gelb}_{\text{zwei}} \notin K_1$, $t_4 \in T_2$, aber $\alpha(t_4) = \text{gelb}_{\text{eins}} \notin K_2$, bzw. $t_2 \in T_3$, aber $\alpha(t_2) = \text{gelb}_{\text{zwei}} \notin K_3$.

Die verwendeten Begriffe und Abkürzungen können Sie im Index bzw. im Kapitel 6.2 ab Seite 56 nachschlagen.

Den aufgeführten Ausschnitten aus dem Session–Report können Sie neben der eigentlichen Berechnung auch die jeweils gefolgerten Netzeigenschaften entnehmen.

<D> Check the deadlock-trap-, SMD-, SMA-properties

Dieses Kommando entscheidet die Deadlock–Falle–Eigenschaft (`DTP`), sowie für gewöhnliche Netze, ob das Netz in stark zusammenhängende Zustandsmaschinenkomponenten dekomponierbar (`SMD`) und ob es Zustandsmaschinen–auswählbar (`SMA`) ist.

Nach der Abfrage der Schreiboptionen berechnet INA zunächst die Liste der bezüglich Mengeninklusion minimalen Deadlocks. Deadlocks, die bei der Anfangsmarkierung nicht ausreichend markiert sind, werden als sauber (`clean`) bzw. nicht ausreichend markiert (`not sufficiently marked`) gemeldet. Die maximale Falle in einem solchen Deadlock kann nicht ausreichend markiert sein, Transitionen, die Vorplätze in diesem Deadlock haben, sind tot.

Anschließend werden die zugehörigen maximalen Fallen berechnet, die Falle mit der Nummer i ist dabei die maximale Falle im Deadlock Nummer i. Auch hier werden Fallen, insbesondere die leere Menge, die keine oder zuwenig Marken haben, als sauber bzw. nicht ausreichend markiert gemeldet.

Falls dies für keine Falle zutrifft, ist die Deadlock–Falle–Eigenschaft erfüllt und deshalb keine tote Markierung erreichbar [Sta90, Satz 15.5 (165)].

Wenn das Netz eine nichtblockierende Vielfachheit (`NBM`) hat, homogen (`HOM`) und extended–simple (`ES`) ist, dann ist es lebendig [Sta90, Satz 15.6 (165)].

Ist eine der Fallen nicht ausreichend markiert, dann ist die Deadlock–Falle–Eigenschaft nicht erfüllt. Für ein extended–free–choice Netz (`EFC`) folgt daraus die Nichtlebendigkeit [Sta90, Satz 15.10 (173)].

Für gewöhnliche Netze (`ORD`) berechnet INA anschließend alle minimalen stark zusammenhängenden Zustandsmaschinenkomponenten, daneben weitere minimale Komponenten, um die Eigenschaft Zustandsmaschinendekomponierbarkeit (`SMD`) zu testen. In diesem Fall ist das Netz strukturell beschränkt (`SB`). Komponenten, die bei der Anfangsmarkierung sauber sind, werden gemeldet.

Wenn das Netz `SMD` und zusammenhängend (`CON`) ist, so prüft INA die Zustandsmaschinenauswählbarkeit (`SMA`). Wenn diese Eigenschaft nicht gegeben ist, so finden Sie im Session–Report eine Vorplatzauswahl als Liste der Form `<transnr> ==> <vorplatznr>`, die keine `SCSM`–Komponente auswählt.

Für das Netz des Beispiels 14 ergibt sich folgende Berechnung:

```
Structural liveness properties of net nr. 14.Zwei_Ampeln     :
processed candidates: 13
There are  3 minimal deadlocks:
-------------------------------
   1: 5.rot_eins, 6.rot_zwei,
   2: 1.gruen_eins, 2.gruen_zwei, 4.gelb_zwei,
   3: 1.gruen_eins, 2.gruen_zwei, 3.gelb_eins,

Corresponding maximal traps:
----------------------------
   1: 5.rot_eins, 6.rot_zwei,
   2: 1.gruen_eins, 2.gruen_zwei, 4.gelb_zwei,
   3: 1.gruen_eins, 2.gruen_zwei, 3.gelb_eins,
The deadlock-trap-property is valid.
The net has no dead reachable states.
The net is live.
The net has no dead transitions at the initial marking.
The net is live, if dead transitions are ignored.
```

```
SM-allocatability of net nr.  0:
********************************
Some minimal place sets Q with QF=FQ, but all SCSM's:

   1: 5.rot_eins, 6.rot_zwei, SCSM
   2: 1.gruen_eins, 2.gruen_zwei, 4.gelb_zwei, SCSM
   3: 1.gruen_eins, 2.gruen_zwei, 3.gelb_eins, SCSM
 3 minimal component(s) found.

The net is state-machine decomposable (SMD).
The net is coverable by state-machines (SMC).
The net is covered by semipositive P-invariants.
The net is structurally bounded.
The net is bounded.
The net is covered by semipositive T-invariants.
There are no proper semipositive T-surinvariants.

The net is not state-machine allocatable (SMA).
A pre-place allocation allocating no SCSM:
   1==>   1,   2==>   4,   3==>   2,   4==>   3,
The net is covered by 1-token SCSM's.

ORD HOM NBM PUR CSV SCF CON SC  Ft0 tF0 Fp0 pF0 MG  SM  FC  EFC ES
 Y   Y   Y   Y   N   N   Y   Y   N   N   N   N   N   N   N   N   Y
DTP SMC SMD SMA CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&S
 Y   Y   Y   N   Y   Y   Y   Y   ?   N   N   N   ?   Y   Y   ?
```

Die Zeile `1==> 1, 2==> 4, 3==> 2, 4==> 3` beschreibt die zu Beginn angegebene Vorplatzauswahl α.

Da die Suche der minimalen Deadlocks erheblichen Rechenaufwand bedeutet, können Sie die laufende Berechnung jederzeit mit `<Q>` unterbrechen.

Eventuell vorher berechnete Symmetrien verkürzen die Suche: Für das Netz 14 ergibt sich mit der in 6.4.2 berechneten Symmetriegruppe, daß statt 13 nur noch 7 Kandidaten untersucht werden müssen.

`<C>      Compute all components`

Mit diesem Kommando werden alle Komponenten des aktuellen Netzes berechnet. Als erstes können Sie Schreiboptionen setzen, danach startet die Berechnung. Die Anzahl der schon berechneten Komponenten wird dabei ausgegeben. Für jede Komponente wird getestet, ob sie eine Zustandsmaschine (`SM` und stark zusammenhängend (`SC`) ist. Am Ende werden die Eigenschaften `SMD` und `SMA` überprüft und entsprechende Folgerungen gezogen.

Für das Netz des Beispiels 14 finden Sie im Session–Report folgende Untersuchungsergebnisse:

```
candidates: 117
 All components of net nr. 14.Zwei_Ampeln     :

   1:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins,  4.gelb_zwei,  5.rot_eins,
       6.rot_zwei,
   2:  5.rot_eins,  6.rot_zwei, SM SC
   3:  1.gruen_eins,  2.gruen_zwei,  4.gelb_zwei,  5.rot_eins,  6.rot_zwei,
   4:  1.gruen_eins,  2.gruen_zwei,  4.gelb_zwei,  6.rot_zwei,
   5:  1.gruen_eins,  2.gruen_zwei,  4.gelb_zwei, SM SC
   6:  1.gruen_eins,  2.gruen_zwei,  4.gelb_zwei,  5.rot_eins,
   7:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins,  5.rot_eins,  6.rot_zwei,
```

```
   8:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins,  6.rot_zwei,
   9:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins, SM SC
  10:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins,  5.rot_eins,
  11:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins,  4.gelb_zwei,  6.rot_zwei,
  12:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins,  4.gelb_zwei,
  13:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins,  4.gelb_zwei,  5.rot_eins,

The net is state-machine decomposable (SMD).
The net is coverable by state-machines (SMC).
The net is covered by semipositive P-invariants.
The net is structurally bounded.
The net is bounded.
There are no proper semipositive T-surinvariants.

The net is not state-machine allocatable (SMA).
A pre-place allocation allocating no SCSM:
   1==>   1,   2==>   4,   3==>   2,   4==>   3,

ORD HOM NBM PUR CSV SCF CON SC  Ft0 tF0 Fp0 pF0 MG  SM  FC  EFC ES
 Y   Y   Y   Y   N   N   Y   Y   N   N   N   N   N   N   N   N   Y
DTP SMC SMD SMA CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&S
 ?   Y   Y   N   Y   ?   Y   Y   ?   ?   N   ?   ?   ?   ?   ?
```

Da die Berechnung sehr aufwendig ist, kann sie jederzeit mit `<Q>` abgebrochen werden.

`<M>` Decide state-machine coverability

Dieses Kommando entscheidet für gewöhnliche Netze, ob sie mit (nicht notwendigerweise stark zusammenhängenden) Komponenten, die Zustandsmaschinen sind, überdeckt werden kann.

Zuerst können Sie wieder Schreiboptionen ändern. Danach berechnet INAalle minimalen Komponenten [Sta90, Definition 16.1 (177)], dabei wird die Anzahl der bereits berechneten Komponenten ausgegeben. Sie können die Untersuchung mit `<Q>` jederzeit abbrechen.

Alle berechneten Komponenten werden daraufhin untersucht, ob sie Zustandsmaschinen [Sta90, Definition 14.3 (152)] sind. Im Session–Report werden diese als `SM` gekennzeichnet. Komponenten, die bei der Anfangsmarkierung nicht markiert sind, werden als sauber (`clean`) gemeldet.

Wenn die Platzmenge des Netzes eine Vereinigung von Platzmengen minimaler Zustandsmaschinen–Komponenten ist, dann ist das Netz `SMC` und folglich beschränkt [Sta90, Folgerung 16.4].

Für das Netz des Beispiels 14 sieht die Ausgabe wie folgt aus:

```
SM-coverability of net nr. 14.Zwei_Ampeln     :

Minimal place sets Q with QF=FQ:

   1:  5.rot_eins,  6.rot_zwei, SM
   2:  1.gruen_eins,  2.gruen_zwei,  3.gelb_eins, SM
   3:  1.gruen_eins,  2.gruen_zwei,  4.gelb_zwei, SM
 3 minimal component(s) found.

candidates created: 29
The net is coverable by state-machines (SMC).
The net is covered by semipositive P-invariants.
The net is structurally bounded.
The net is bounded.
```

```
There are no proper semipositive T-surinvariants.

ORD HOM NBM PUR CSV SCF CON SC  Ft0 tF0 Fp0 pF0 MG  SM  FC  EFC ES
 Y   Y   Y   Y   N   N   Y   Y   N   N   N   N   N   N   N   N   Y
DTP SMC SMD SMA CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&S
 ?   Y   ?   ?   Y   ?   Y   Y   ?   ?   N   ?   ?   ?   ?   ?
```

Falls das Netz nicht mit Zustandsmaschinen überdeckt werden kann, so finden Sie im Session–Report eine Übersicht der Plätze, die in Zustandsmaschinen–Komponenten liegen und deshalb beschränkt sind.

Wenn INA bereits bekannt ist, daß das Netz mit Zustandsmaschinen überdeckbar ist (`the net is known to be state machine coverable`), lautet die erste Abfrage: `Compute the SM-components?` Falls Sie diese Abfrage mit `<Y>` beantworten, führt INA die Berechnung trotzdem durch.

Auch hier kann jederzeit die Berechnung mit `<Q>` abgebrochen werden.

Anhang

Literaturverzeichnis

[Sta90] STARKE, Peter H.: *Analyse von Petri–Netz–Modellen*. Stuttgart : B. G. Teubner, 1990 (Leitfäden und Monographien der Informatik)

[Sta91] STARKE, Peter H.: Reachability Analysis of Petri Nets Using Symmetries. In: *J. of Math. Modelling & Simulation in Systems Analysis* 8 (1991), S. 293–304

[SS91] SCHMIDT, Karsten ; STARKE, Peter H.: An Algorithm to Compute the Symmetries of Petri Nets. In: *Petri Net Newsletter* 40 (1991), S. 25–30

[Fin89] FINKEL, Alain: A Minimal Coverability Graph for Petri Nets / Université Paris 11. 1989 (501). – Forschungsbericht

[Lü95] LÜTTGE, Karsten: *Zustandsgraphen von PETRI–Netzen*, Humboldt–Universität zu Berlin, Institut für Informatik, Diplomarbeit, 1995

[Pog94] POGRELL, Lutz: *Theoretische Grundlagen und Implementation der Methode der sturen Mengen*, Humboldt–Universität zu Berlin, Institut für Informatik, Diplomarbeit, 1994

Index

Glossar

Über dieses Handbuch

Dieses Handbuch beschreibt die Version 1.7 von INA. Es wurde von Stephan Roch in Zusammenarbeit mit Prof. Peter H. Starke mit dem Textsatzsystem LaTeX2e und dem KOMA–Script–Package (von Frank Neukamm und Markus Kohm) bzw. dem Programm LaTeX2HTML (von Nikos Drakos nikos@cbl.leeds.ac.uk) erstellt. Es basiert auf ausführlichen Tests, sorgfältiger Quelltextanalyse und persönlichen Gesprächen mit Prof. Starke.

Über Internet bietet die WWW–Seite

`http://www.informatik.hu-berlin.de/lehrstuehle/automaten/ina/`

Zugriff auf dieses Handbuch in den Formaten TeX–DVI und Postscript. Ebenso ist dort eine mit üblichen Browsern betrachtbare HTML–Version verfügbar. Genauso können Sie dort alle Beispielnetze dieses Scriptes abrufen.

Für Fragen und Anregungen zu diesem Handbuch wenden Sie sich bitte an Stephan Roch (roch@informatik.hu-berlin.de).

Eine kostenlose Demoversion des Programms mit eingeschränkter Funktionalität steht für DOS–Plattformen mit anonymen FTP auf `ftp.informatik.hu-berlin.de` im Verzeichnis `/pub/local/petritool` zur Verfügung (dort kann auch dieses Handbuch abgerufen werden).

Zu Informationen für eine uneingeschränkte Vollversion wenden Sie sich bitte an Prof. Starke (starke@informatik.hu-berlin.de). Zur Zeit werden DOS– und Unix–Plattformen (unter Sun–OS) unterstützt. Preise und Konditionen werden auf Anfrage genannt.

Für Fehler in diesem Handbuch oder im Programm wird keine Haftung übernommen.

Kontaktadresse:

Humboldt–Universität zu Berlin
Institut für Informatik
Lehrstuhl für Automaten– und Systemtheorie
Prof. Peter. H. Starke
Axel Springer Straße 54a
10117 Berlin
Telefon: 030 / 20 181 - 285
Fax: 030 / 20 181 - 287
Email: starke@informatik.hu-berlin.de