(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5857072号
(P5857072)
(45)発行日　平成28年2月10日(2016.2.10)　(24)登録日　平成27年12月18日(2015.12.18)

(51)Int.Cl.　F I
G06F 9/45 (2006.01)　G06F 9/44 322F

請求項の数 47 (全 34 頁)

(21)出願番号	特願2013-550671 (P2013-550671)
(86)(22)出願日	平成24年1月24日 (2012.1.24)
(65)公表番号	特表2014-506693 (P2014-506693A)
(43)公表日	平成26年3月17日 (2014.3.17)
(86)国際出願番号	PCT/US2012/022441
(87)国際公開番号	WO2012/103148
(87)国際公開日	平成24年8月2日 (2012.8.2)
審査請求日	平成27年1月22日 (2015.1.22)
(31)優先権主張番号	61/436,051
(32)優先日	平成23年1月25日 (2011.1.25)
(33)優先権主張国	米国(US)

早期審査対象出願

(73)特許権者　595168543
マイクロン　テクノロジー，　インク.
アメリカ合衆国，アイダホ州　８３７１６－９６３２，　ボイズ，　サウス　フェデラル　ウェイ　８０００
(74)代理人　100106851
弁理士　野村　泰久
(74)代理人　100074099
弁理士　大菅　義之
(72)発明者　シュー，ジュンジュエン
アメリカ合衆国，カリフォルニア州　９５１３１，サン　ノゼ，クアドロス　レーン　２００８

最終頁に続く

(54)【発明の名称】オートマトンの入次数および／または出次数を制御するための量化子の展開

(57)【特許請求の範囲】
【請求項１】
量化子をオートマトンに変換することであって、変換することが、前記オートマトンの入次数を制御するように前記量化子を展開することを含む、量化子をオートマトンに変換することと、
前記オートマトンを、対象装置に対応する機械コードに変換することと
を含むコンピュータ実装方法。
【請求項２】
式を、前記量化子を含む、言語固有でない表現に解析すること、をさらに含む、請求項１に記載のコンピュータ実装方法。
【請求項３】
式を、言語固有でない表現に解析することが、正規表現を構文木に解析することを含む、請求項２に記載のコンピュータ実装方法。
【請求項４】
変換することが、前記量化子に対する繰返し式が空値可能である場合、前記量化子を、ゼロ回一致し得る前記繰返し式の空値不能形式を有する量化子として展開することを含む、請求項１に記載のコンピュータ実装方法。
【請求項５】
前記オートマトンを機械コードに変換することが、前記オートマトンを、対象装置をプログラムするように構成されたイメージに変換することを含む、請求項１に記載のコンピ

ュータ実装方法。
【請求項６】
前記イメージを公開すること、をさらに含む、請求項５に記載のコンピュータ実装方法。
【請求項７】
前記オートマトンの入次数を制御するように前記量化子を展開することが、入次数制限に応じて、前記量化子を展開することを含む、請求項１に記載のコンピュータ実装方法。
【請求項８】
前記入次数制限が、前記オートマトンの状態への遷移数に関する制限を含む、請求項７に記載のコンピュータ実装方法。
【請求項９】
前記オートマトンの入次数を制御するように前記量化子を展開することが、前記対象装置の入次数制約に基づいて展開することを含む、請求項１に記載のコンピュータ実装方法。
【請求項１０】
前記量化子が正規表現である、請求項１に記載のコンピュータ実装方法。
【請求項１１】
前記オートマトンの入次数を制御するように前記量化子を展開することが、閾値に基づいて前記オートマトンの入次数を制御するように前記量化子を展開することを含む、請求項１に記載のコンピュータ実装方法。
【請求項１２】
前記閾値が、前記対象装置のハードウェア制限に基づく、請求項１１に記載のコンピュータ実装方法。
【請求項１３】
命令を含むコンピュータ可読媒体であって、
前記命令が、前記コンピュータによって実行されるときに、前記コンピュータに：
量化子をオートマトンに変換することであって、変換することが、前記オートマトンの出次数を制御するように前記量化子を展開することを含む、量化子をオートマトンに変換することと、
前記オートマトンを、対象装置に対応する機械コードに変換することと
を含む動作を実行させる、
コンピュータ可読媒体。
【請求項１４】
展開することが、前記オートマトンの各状態に対する前記出次数を閾値数未満に制限することを含む、請求項１３に記載のコンピュータ可読媒体。
【請求項１５】
展開することが、前記オートマトンの各状態に対する出遷移を最小限にすることを含む、請求項１３に記載のコンピュータ可読媒体。
【請求項１６】
変換することが、前記量化子に対する繰返し式が空値可能である場合、前記量化子を、ゼロ回一致し得る前記繰返し式の空値不能形式を有する量化子として展開することを含む、請求項１３に記載のコンピュータ可読媒体。
【請求項１７】
変換することが、
前記量化子が単一のある個数のループと一致し得る場合、前記量化子を、連続的にリンクされた前記単一のある個数に等しい、いくつかの式を表す状態を有するオートマトンを形成するように展開することを含み、前記式が、前記量化子に対する繰返し式に対応する、請求項１３に記載のコンピュータ可読媒体。
【請求項１８】
前記量化子が複数のある個数のループと一致し得る場合、前記複数のある個数のループ

が、第１の数のループ、ｎ１、および第２の数のループ、ｎ２を含み、前記量化子を展開することが：
　前記量化子を、ｎ１－１個のループと一致し得る第１の量化子および、１～ｎ２－ｎ１＋１個までのループと一致し得る第２の量化子に分割することを含む、
請求項１３に記載のコンピュータ可読媒体。
【請求項１９】
　前記機械コードが、対象装置をプログラムするように構成されたイメージを含み、
　前記命令が、前記コンピュータに：
　前記イメージを対象装置上にロードすること
を含む動作を実行させる、請求項１３に記載のコンピュータ可読媒体。
【請求項２０】
　前記機械コードが、対象装置をプログラムするように構成されたイメージを含み、
　前記命令が、前記コンピュータに：
　前記イメージを、コンピュータ可読媒体上に格納すること
を含む動作を実行させる、請求項１３に記載のコンピュータ可読媒体。
【請求項２１】
　コンピュータであって、
　ソフトウェアがその上に格納されたメモリと、
　前記メモリに通信的に結合されたプロセッサであって、前記ソフトウェアが、前記プロセッサによって実行されるとき、前記プロセッサに：
　量化子をオートマトンに変換することであって、変換することが、前記量化子を、前記オートマトンの入次数または出次数を制御するように展開することを含む、量化子をオートマトンに変換することと、
　前記オートマトンを、対象装置に対応する機械コードに変換することと
をさせる、プロセッサと
を備えるコンピュータ。
【請求項２２】
　前記ソフトウェアが、前記プロセッサに、前記オートマトン内で入遷移を出遷移に対してトレードオフすることにより、前記入次数および出次数を制御させる、請求項２１に記載のコンピュータ。
【請求項２３】
　各状態に対する前記入遷移が、展開時に閾値数未満に制限される、請求項２２に記載のコンピュータ。
【請求項２４】
　前記入次数が、前記出次数の割合に制限される、請求項２２に記載のコンピュータ。
【請求項２５】
　前記オートマトンの状態に対する入遷移の閾値に達するまで、前記入次数が、前記出次数の前記割合に制限される、請求項２４に記載のコンピュータ。
【請求項２６】
　量化子をオートマトンに変換することであって、変換することが、前記オートマトンの入次数または出次数を制御するように前記量化子を展開することを含む、量化子をオートマトンに変換することと、
　前記オートマトンを、対象装置に対応する機械コードに変換することと
を行うように構成されたコンピュータ
を備えるシステム。
【請求項２７】
　前記機械コードが、対象装置をプログラムするように構成されたイメージを含み、
　前記コンピュータに通信的に結合され、かつ前記イメージを対象装置上にロードするように構成された装置を
さらに含む、請求項２６に記載のシステム。

【請求項２８】
前記オートマトンの入次数を制御するように前記量化子を展開することが、入次数制限に応じて前記量化子を展開することを含む、請求項２６に記載のシステム。
【請求項２９】
前記入次数制限が、前記オートマトンの状態への遷移数に関する制限を含む、請求項２８に記載のシステム。
【請求項３０】
前記オートマトンの入次数を制御するように前記量化子を展開することが、前記対象装置の入次数制約に基づいて展開することを含む、請求項２６に記載のシステム。
【請求項３１】
前記オートマトンの入次数を制御するように前記量化子を展開することが、閾値に基づいて前記オートマトンの入次数を制御するように前記量化子を展開することを含む、請求項２６に記載のシステム。
【請求項３２】
前記閾値が、前記対象装置のハードウェア制限に基づく、請求項３１に記載のシステム。
【請求項３３】
量化子をオートマトンに変換することであって、変換することが、前記オートマトンの出次数を制御するように前記量化子を展開することを含む、量化子をオートマトンに変換することと、
前記オートマトンを、対象装置に対応する機械コードに変換することと
を含むコンピュータ実装方法。
【請求項３４】
正規表現を構文木に解析すること、をさらに含む、請求項３３に記載のコンピュータ実装方法。
【請求項３５】
前記オートマトンを機械コードに変換することが、前記オートマトンを、対象装置をプログラムするように構成されたイメージに変換することを含む、請求項３３に記載のコンピュータ実装方法。
【請求項３６】
前記イメージを公開すること、をさらに含む、請求項３５に記載のコンピュータ実装方法。
【請求項３７】
前記オートマトンの出次数を制御するように前記量化子を展開することが、出次数制限に応じて前記量化子を展開することを含む、請求項３３に記載のコンピュータ実装方法。
【請求項３８】
前記出次数制限が、前記オートマトンの状態からの遷移数に関する制限を含む、請求項３７に記載のコンピュータ実装方法。
【請求項３９】
前記オートマトンの出次数を制御するように前記量化子を展開することが、前記対象装置の出次数制約に基づいて展開することを含む、請求項３３に記載のコンピュータ実装方法。
【請求項４０】
前記オートマトンの出次数を制御するように量化子を展開することが、閾値に基づいて前記オートマトンの出次数を制御するように前記量化子を展開することを含む、請求項３３に記載のコンピュータ実装方法。
【請求項４１】
前記閾値が、前記対象装置のハードウェア制限に基づく、請求項４０に記載のコンピュータ実装方法。
【請求項４２】

命令を含むコンピュータ可読媒体であって、
前記命令が、前記コンピュータによって実行されるときに、前記コンピュータに：
量化子をオートマトンに変換することであって、変換することが、前記オートマトンの入次数を制御するように前記量化子を展開することを含む、量化子をオートマトンに変換することと、
前記オートマトンを、対象装置に対応する機械コードに変換することと
を含む動作を実行させる、
コンピュータ可読媒体。
【請求項４３】
展開することが、前記オートマトンの各状態に対する入遷移を閾値数未満に制限することを含む、請求項４２に記載のコンピュータ可読媒体。
【請求項４４】
展開することが、前記オートマトンの各状態に対する入遷移を最小限にすることを含む、請求項４２に記載のコンピュータ可読媒体。
【請求項４５】
変換することが、前記量化子に対する繰返し式が空値可能である場合、前記量化子を、ゼロ回一致し得る前記繰返し式の空値不能形式を有する量化子として、展開することを含む、請求項４２に記載のコンピュータ可読媒体。
【請求項４６】
変換することが、
前記量化子が単一のある個数のループと一致し得る場合、連続的にリンクされた前記単一のある個数に等しい、いくつかの式を表す状態を有するオートマトンを形成するように前記量化子を展開することを含み、前記式が、前記量化子に対する繰返し式に対応する、請求項４２に記載のコンピュータ可読媒体。
【請求項４７】
前記量化子が複数のある個数のループと一致し得る場合、前記複数のある個数のループが、第１の数のループ、ｎ１、および第２の数のループ、ｎ２を含み、前記量化子を展開することが：
前記量化子を、ｎ１－１個のループと一致し得る第１の量化子および、１～ｎ２－ｎ１＋１個までのループと一致し得る第２の量化子に分割することを含む、
請求項４２に記載のコンピュータ可読媒体。

【発明の詳細な説明】
【技術分野】
【０００１】
〔優先権主張〕
本特許出願は、２０１１年１月２５日に出願された、「ＵＮＲＯＬＬＩＮＧ　ＱＵＡＮＴＩＦＩＣＡＴＩＯＮＳ　ＴＯ　ＣＯＮＴＲＯＬ　ＩＮ－ＤＥＧＲＥＥ　ＡＮＤ／ＯＲ　ＯＵＴ　ＤＥＧＲＥＥ　ＯＦ　ＡＵＴＯＭＡＴＯＮ」という名称の米国仮特許出願整理番号６１／４３６，０５１に対する優先権の利益を主張し、これによって、参照によりその全体として本明細書に組み込まれる。
【背景技術】
【０００２】
有限状態機械（ＦＳＭ）（有限状態オートマトン、オートマトン、または単に状態機械とも呼ばれる）は、状態、状態と動作との間の遷移の表現である。有限状態機械は、並列マシン用のデジタル論理、コンピュータプログラム、またはイメージを設計するために使用できる。有限状態機械は、有限数の状態、それら状態間の遷移、および出力から成る挙動のモデルである。有限状態機械は、グラフとして表すことができ、グラフのバーテックスが有限状態機械の状態に対応し、グラフのエッジが、有限状態機械に対する１つまたは複数の入力に起因して生じる状態間の遷移に対応する。有限状態機械は、確率的遷移、フ

ァジー状態、または他の奇妙な事柄（ｏｄｄｉｔｙ）も持つことができる。有限状態機械は、有限の内部メモリ、入力特徴、および随意の出力特徴を有する。出力を有する有限状態機械は、有限状態トランスデューサと呼ぶことができる。

【０００３】

有限状態機械の用途には、コンピュータによる電子設計の自動化、通信プロトコル設計、生物学および人工知能研究、ならびに自然言語の文法を説明する言語学を含む。

【図面の簡単な説明】

【０００４】

【図１】本発明の様々な実施形態に従い、オートマトンを使用して、ソースコードを機械コードに変換できるコンパイラのための流れ図の一例を示す。

【図２】本発明の様々な実施形態に従い、コンパイラによって使用されるオートマトンの一例を示す。

【図３】本発明の様々な実施形態に従った、量化子を展開するための方法の一例を示す。

【図４Ａ】本発明の様々な実施形態に従って、展開された量化子のためのオートマトン例を示す。

【図４Ｂ】本発明の様々な実施形態に従って、展開された量化子のためのオートマトン例を示す。

【図４Ｃ】本発明の様々な実施形態に従って、展開された量化子のためのオートマトン例を示す。

【図５】本発明の様々な実施形態に従った、並列マシンの一例を示す。

【図６】本発明の様々な実施形態に従って、有限状態機械エンジンとして実装された図５の並列マシンの一例を示す。

【図７】本発明の様々な実施形態に従って、図６の有限状態機械エンジンのブロックの一例を示す。

【図８】本発明の様々な実施形態に従って、図７のブロックの行（ｒｏｗ）の一例を示す。

【図９】本発明の様々な実施形態に従って、図８の行の２つのグループの一例を示す。

【図１０】本発明の様々な実施形態に従って、コンパイラが正規表現を、図５の並列マシンをプログラムするように構成されたイメージに変換するための方法の一例を示す。

【図１１Ａ】本発明の様々な実施形態に従った、量化子のためのオートマトンに対応するグラフ例を示す。

【図１１Ｂ】本発明の様々な実施形態に従った、量化子のためのオートマトンに対応するグラフ例を示す。

【図１２】本発明の様々な実施形態に従った、量化子のためのオートマトンに対応する別のグラフ例を示す。

【図１３】本発明の様々な実施形態に従った、量化子のためのオートマトンに対応するさらに別のグラフ例を示す。

【図１４】本発明の様々な実施形態に従った、量化子のためのオートマトンに対応するさらにまた別のグラフ例を示す。

【図１５】本発明の様々な実施形態に従った、コンピュータの一例を示す。

【発明を実施するための形態】

【０００５】

以下の説明および図は、特定の実施形態を当業者が実施できるように十分に説明する。他の実施形態は、構造的、論理的、電気的、プロセス、および他の変更を包含し得る。いくつかの実施形態の部分および特徴は、他の実施形態に含まれ得るか、または他の実施形態のそれらと置き換えられ得る。請求項に規定される実施形態は、それらの請求項の全ての利用可能な均等物を包含する。

【０００６】

本文書では、特に、ソースコードを、有限状態機械の機械コード実装に変換するコンパイラを説明する。機械コードが、ソースコードによって記述された機能を対象装置上で実

装するように構成されているという点において、機械コードは、対象装置に対応できる。一例では、対象装置は並列マシンであり、機械コードは、その並列マシン用のイメージを含む。別の例では、対象装置は、フォンノイマン型アーキテクチャを有するコンピュータを含み、機械コードは、そのコンピュータ内のプロセッサによって実行するための命令を含む。

【０００７】

いかなる場合にも、コンパイラは、ソースコードを、そのソースコードによって記述された機能を実施する有限状態機械に変換する。ソースコードをコンパイルするプロセスにおいて、コンパイラは、ソースコードをオートマトンに変換する。オートマトンを使用すると、コンパイラは、機械コードで実装された、結果として得られる有限状態機械を最適化するために、ソースコード内の冗長性を識別し、結合できる。さらに、コンパイラは、結果として得られる機械コードの複雑性を減少させ、対象装置の動作効率を向上させるために、オートマトンを形成する際に、対象装置の態様または制限を考慮することができる。

【０００８】

図１は、コンパイラ例に対する流れ図１００を示す。コンパイラは、ソースコードを入力として受け取り、ソースコードによって記述された機能を対象装置上で実装するために、機械コードを生成する。一例では、対象装置は、図５～図９に関して後述するように、並列マシンを含む。並列マシンは、複数の状態のうちの１つに設定できる複数のプログラム可能要素を含む。並列マシンに対する機械コードは、１つまたは複数のプログラム可能要素の状態を設定するためのイメージを含む。別の例では、対象装置は、フォンノイマン型アーキテクチャを有するコンピュータである。コンピュータは、１つまたは複数のプロセッサによって実行するためにその上にソフトウェアを有する、１つまたは複数の記憶装置に結合された１つまたは複数のプロセッサを含む。フォンノイマン型アーキテクチャのための機械コードは、１つまたは複数のプロセッサによって実行するための命令を含む。フォンノイマン型アーキテクチャを有するコンピュータの例は、図１５に関して後述される。いかなる場合にも、コンパイラは、オートマトンを中間変換として使用することにより、機械コードを生成する。コンパイラは、オートマトンを、特に、結果として得られるＦＳＭを最適化するため、また、同様に、機械コードを最適化するために使用する。

【０００９】

一例では、ソースコードは、シンボルグループ内のシンボルのパターンを識別するための検索文字列を記述する。検索文字列を記述するため、ソースコードは、複数の正規表現（ｒｅｇｅｘ）を含むことができる。正規表現は、シンボル検索パターンを記述するための文字列であり得る。正規表現は、プログラミング言語、テキストエディタ、ネットワークセキュリティ、およびその他などの、様々なコンピュータドメインで幅広く使用されている。一例では、コンパイラによってサポートされる正規表現は、非構造化データを検索するための検索基準を含む。非構造化データは、自由形式のデータを含み得、データ内のワード（ｗｏｒｄ）に適用された指標付けがない。ワードは、データ内に、印刷可能および印字不能な、バイトの任意の組合せを含み得る。一例では、コンパイラは、Ｐｅｒｌ（例えば、Ｐｅｒｌ互換正規表現（ＰＣＲＥ））、ＰＨＰ、Ｊａｖａ（登録商標）、および．ＮＥＴ言語を含む、正規表現を実装するための、複数の異なるソースコード言語をサポートできる。

【００１０】

図１を再度参照すると、ブロック１０２で、コンパイラは、ソースコードを解析して、関係的に結合された演算子の配置を形成できる。ソースコードの解析により、ソースコードの一般的表現を作成できる。一例では、一般的表現は、ソースコード内の正規表現の、構文木として知られているツリーグラフの形でコード化された表現を含む。本明細書で説明する例は、そのアレンジメントを構文木（「抽象構文木」としても知られている）と呼ぶが、他の例では、具象構文木または他のアレンジメントも使用できる。

【００１１】

前述したように、コンパイラは、複数言語のソースコードをサポートできるので、解析は、その言語にかかわらず、ソースコードを、例えば、構文木などの、言語固有でない表現に変換する。従って、コンパイラによるさらなる処理（ブロック１０４、１０６、１０８、１１０）は、ソースコードの言語にかかわらず、共通の入力構造から動作できる。

【００１２】

構文木は、関係的に結合されている複数の演算子を含む。構文木は、複数の異なるタイプの演算子を含むことができ、そこでは、異なるタイプの演算子は、ソースコードによって実装された異なる機能に対応する。すなわち、異なる演算子は、ソースコード内の正規表現によって実装された異なる機能に対応できる。

【００１３】

ブロック１０４で、構文木はオートマトンに変換される。オートマトンは、ＦＳＭのソフトウェアモデルを含み、それに応じて、決定性または非決定性として分類できる。決定性オートマトンは、一時に、単一経路の実行を有するが、他方、非決定性オートマトンは、複数の並列経路の実行を有する。オートマトンは、複数の状態を含む。構文木をオートマトンに変換するために、構文木内の演算子および演算子間の関係は、状態間の遷移をもつ状態に変換される。

【００１４】

対象装置の要素を効果的に利用するために、コンパイラは、対象装置の要素の態様または制限に基づいて、オートマトンを形成できる。例えば、要素は、オートマトン内の所与の状態への、または所与の状態からの遷移の数に関して特定の制約を有し得る。他の例では、ハードの制約は存在しないかも知れないが、コンパイラは、機械コードを生成するため、および／または対象装置上での機械コードの動作効率を向上させるために、コンパイラによって要求される処理を単純化するため、所与の状態への、または所与の状態からの遷移の数を制限し得る。所与の状態への、または所与の状態からの遷移の制限に関するさらなる詳細は、図３および図４Ａ～図４Ｃに関して、以下で提供される。

【００１５】

ブロック１０６で、オートマトンが形成されると、オートマトンは、とりわけ、その複雑性およびサイズを縮小するように最適化される。オートマトンは、特に、同等な状態を結合することにより最適化できる。

【００１６】

ブロック１０８で、オートマトンが、対象装置用の機械コードに変換される。一例では、機械コードは、フォンノイマン型アーキテクチャのプロセッサ用の実行可能な命令を含む。ここで、機械コードは、実行可能プログラムを含むことができる。別の例では、機械コードは、並列マシン内のハードウェア要素をプログラムするように構成されたビットを含むことができる。ここで、機械コードは、並列マシン上にロードするためのイメージを含むことができる。

【００１７】

ブロック１１０で、機械コードは、コンパイラによって公開できる。一例では、機械コードは、機械コードをコンピュータ可読媒体に保存することによって公開できる。別の例では、機械コードは、機械コードを並列マシン上にロードするためのローダーなど、別の装置に機械コードを送信することによって公開できる。さらに別の例では、機械コードは、機械コードを並列マシン上にロードすることによって公開できる。さらにまた別の例では、機械コードは、機械コードをディスプレイ装置上に表示することにより公開できる。

【００１８】

一例では、コンパイラは、フォンノイマン型アーキテクチャを有するコンピュータ用の命令によって実装できる。これらの命令は、コンピュータ上のプロセッサにコンパイラの機能を実装させることができる。例えば、命令は、プロセッサによって実行される場合、プロセッサがアクセス可能なソースコード上のブロック１０２、１０４、１０６、１０８、および１１０に記述する動作をプロセッサに実行させることができる。フォンノイマン型アーキテクチャを有するコンピュータ例が図１５に示され、以下で説明される。

【００１９】
ソースコード内に記述できる正規表現の１つのタイプは、量化子を含む。量化子は、当技術分野で周知であり、繰返しパターンを記述するために使用される。一例として、“Ａ（Ｂ）｛ｎ１，ｎ２｝Ｃ”は、一般的な正規表現であり、ここで、Ａ、ＢおよびＣは部分式であり、また、“（Ｂ）｛ｎ１，ｎ２｝”は量化子を含む。本明細書では、大文字は、正規表現または正規表現の一部（例えば、部分式）を表すために使用される。二重引用符が、混乱を避けるために、正規表現または部分式の前後に追加され得る。その結果、式を記述する大文字は、複数の入力シンボルに対する検索文字列に対応できる。例えば、式“Ａ”は、入力文字列‘ａｂｂｃ’に対応できる。

【００２０】
さらに、本明細書では、式および部分式という用語が関係記述のみのために使用される（例えば、部分式は式の一部である）こと、ならびに式および部分式という用語は、いかなる特定の長さ、構文、または文字数に限定されるべきではないことが理解されるはずである。特に、ソースコードは、文字セット全体またはその任意の個々の部分が「式」と考えられる多数の文字（メタ文字および検索文字を含む）を含むことができる。例えば、以下の各々は、式と見なされ得る：“ａ（ｂｂ｜ｄ？）｛５，２０｝ｃ”、“（ｂ）｛０，１０｝”、“（ｂ｜ｄ）”、および“ｂ”。

【００２１】
量化子は、“（Ｂ）｛ｎ１，ｎ２｝”のような正規表現で表され、ここで、Ｂは、部分式であり、ｎ１およびｎ２は、先行する部分式が何回出現するのを許可されるかを指定する整数である。Ｂは、ｎ１およびｎ２によって指定された回数だけ繰り返される部分式なので、Ｂは、本明細書では、繰返し部分式（ｒｅｐｅａｔｅｄ　ｓｕｂ－ｅｘｐｒｅｓｓｉｏｎ）と呼ばれる。量化子“（Ｂ）｛ｎ１，ｎ２｝”と一致させるため、繰返し部分式Ｂは、ｎ１～ｎ２回、一致しなければならない。例えば、正規表現“（Ｂ）｛５，７）”は、部分式Ｂが５、６、または７回、一致することを必要とするであろう。正規表現“Ａ（Ｂ）｛ｎ１，ｎ２｝Ｃ”では、部分式Ａは、一致する場合、量化子に遷移するので、部分式Ａは本明細書では、駆動式（ｄｒｉｖｅ　ｅｘｐｒｅｓｓｉｏｎ）と呼ばれる。さらに、量化子のためのカウントの繰返しおよび増分を継続するため、量化子の繰返し部分式が連続して一致しなければならない。つまり、量化子の所与のループ中、繰返し部分式が一致しない場合、量化子が終了する。一例では、シンボル‘？’も量化子に対応し、ここで、‘？’に先行するシンボルは、１回またはゼロ回のどちらかに識別できる。

【００２２】
図２は、量化子を有する正規表現に対応するオートマトン例２００を示す。オートマトン２００は、正規表現“ａｂｂ？（ｂ｜ｃ）｛１，２｝”に対応する。オートマトン２００は、初期状態２０２および最終状態２１２、２０４を含む。最終状態２１２、２０４は、図２では、二重丸で識別される。初期状態２０２は、最初にアクティブにされ、入力シンボル‘ａ’で、状態２０６に遷移する。状態２０６は、入力シンボル‘ｂ’で、状態２０８および状態２１０の両方に遷移する。状態２０８は、入力シンボル‘ｂ’で状態２１０に遷移し、入力シンボル‘ｂ’または‘ｃ’のどちらかで、状態２１２に遷移する。同様に、オートマトン２００は、入力シンボル‘ｂ’または‘ｃ’のどちらかで、状態２１２から状態２０４に遷移する。状態２１２または状態２０４のどちらかのアクティブ化が、オートマトン２００が対応する正規表現“ａｂｂ？（ｂ｜ｃ）｛１，２｝”との一致を示すので、状態２１２および２０４は、最終状態として識別される。

【００２３】
図３は、構文木をオートマトンに変換するための方法３００の一例を示す。特に、方法３００は、量化子を展開することにより、量化子演算子（本明細書では、単に「量化子」とも呼ぶ）をオートマトンに変換するための手段を示す。量化子を展開することは、例えば、量化子を量化子構文で書き換えること、および非量化子構文をオートマトンに変換することを含み得る。例えば、“（ｂ｜ｃ）｛１，２｝”は“（ｂ｜ｃ）（ｂ｜ｃ）？”と書き換えることができる。展開の利点には、結果として得られるオートマトンが有向非巡

回グラフ（ＤＡＧ）であることを含み得、それは、対象装置内で実装するのをさらに容易にし得る。

【００２４】

方法３００は、量化子の繰返し部分式が空値可能（ｎｕｌｌａｂｌｅ）な場合に、量化子が空値不能形式に変換される、ブロック３０２から始まる。繰返し部分式が空値可能でない場合、方法は、量化子を変換することなく、３０４に進む。式が、空文字列によって一致され得る場合、式は空値可能である。例えば、式（ｂｃ｜）は、入力シンボル‘ｂｃ’または空文字列のどちらかによって一致され得る。その結果、式は空文字列によって一致できるので、式（ｂｃ｜）は空値可能である。空値可能な繰返し部分式に対応する量化子を、空値不能な繰返し部分式をもつ量化子に変換するために、繰返し部分式の空値可能な部分が取り除かれる。例えば、式（ｂｃ｜）は（ｂｃ）になる。さらに、量化子と一致させるループの数が、｛ｎ１，ｎ２｝から｛０，ｎ２｝に修正される。例えば、式“ＡＢ｛ｎ１，ｎ２｝Ｃ”が“ＡＢ´｛０，ｎ２｝Ｃ”に変換され、ここでＢ´はＢの空値不能な形式である。一例では、量化子“ａ（ｂｃ｜）｛３，５｝ｄ”は、“ａ（ｂｃ）｛０，５｝ｄ”に変換できる。

【００２５】

一例では、ブロック３０４で、量化子が展開されるかどうかが随意に判断される。前述したように、量化子を展開することは、非量化子構文をもつ量化子を書き換えること、および非量化子構文をオートマトンに変換することを含み得る。量化子のこの書き換えの結果は、結果として生じる展開された構文が、専用状態（例えば、カウンタ）ではなく、汎用状態（例えば、状態機械要素）で実装されることである。しかし、いくつかの状況では、量化子を専用要素（例えば、カウンタ）で実装すること、従って、量化子を展開しないことが望ましくあり得る。他の状況では、量化子は展開される。量化子が展開されない場合、方法３００は、その量化子に対して終了する。量化子が、少なくとも一部、展開できる場合、方法３００は、ブロック３０６に進む。以下の条件は、量化子が、いつ、おそらくはカウンタと共に実装できるか、また、いつ、カウンタで実装されず、展開されるかの例を示す。以下のある状況は、おそらくはカウンタで実装されているとして説明されているが、他の例では、これらの状況は展開によって実装され得ることが理解されるはずである。一般に、全ての量化子は、所望であれば、展開によって実装できる。

【００２６】

量化子を実装するためにカウンタが使用できるかどうかを判断する前に、£（Ｂ）が空文字を含む場合、“Ｂ｛ｎ１，ｎ２｝”の量化子が“Ｂ´｛０，ｎ２｝”として書き換えられ、ここで、Ｂ´は、Ｂの非空文字列バージョンであり、£（Ｂ´）＝£（Ｂ）－Φである。

例えば、“（ｂｃ｜）｛１０，２０｝”は“（ｂｃ）｛０，２０｝”と書き換えられるが、これは、これらの正規表現が全く同じデータを受け付けるからである。

【００２７】

１．（ｎ１＝０，ｎ２＝－１）の場合、量化子は展開によって実装される。ここで、カウンタは必要ない。

２．（ｎ１＝１，ｎ２＝－１）の場合、量化子は展開によって実装される。ここで、カウンタは必要ない。

３．（ｎ１＞１，ｎ２＝－１）の場合、Ｂ｛ｎ，－１｝がＢ｛ｎ１－１｝Ｂ＋と等しいので、量化子は２つの正規表現Ｂ｛ｎ１－１｝およびＢ＋に分割される。量化子Ｂ｛ｎ１－１｝は、次いで、おそらくはカウンタで実装でき、他方Ｂ＋は、展開によって実装される。

４．（ｎ１＝０，ｎ２＞０）の場合、（Ｂ｛１，ｎ２｝）？がＢ｛０，ｎ２｝と等しいので、量化子は（Ｂ｛１，ｎ２｝）？に修正される。空値不能なＢ｛１，ｎ２｝が、次いで、おそらくはカウンタで実装できる。

５．（ｎ１＞０，ｎ２＞０）の場合、量化子は、おそらくは、カウンタでＢ｛ｎ１，ｎ２｝として実装され得る。

要約として、おそらくは、修正することなく、カウンタで実装できる量化子は、Ｂ｛ｎ１，ｎ２｝として記述でき、ここで、Ｂは空値可能でなく、ｎ１＞０、ｎ２＞０、およびｎ１≦ｎ２である。

【００２８】

ブロック３０６で、単一のある個数のループと一致し得る量化子は、連続的にリンクされた、複数の繰返し部分式を有するオートマトンを形成するように展開される。単一のある個数のループを有する量化子は、ｎ１がｎ２に等しい量化子に対応する。例えば、量化子“Ｂ｛ｎ１｝”は、Ｂのｎ１個のコピーを有する、“ＢＢ．．．Ｂ”として展開できる。

【００２９】

ブロック３０８で、ｎ１がｎ２と等しくない場合、およびｎ１が１と等しく、かつｎ２が１より大きい場合、複数のある個数のループと一致できる量化子が展開される。ｎ１が１より大きい場合、量化子は、ｎ１－１個のループと一致できる第１の量化子、および１からｎ２－ｎ１＋１個のループと一致できる第２の量化子に分割される。例えば、量化子Ｂ｛ｎ１，ｎ２｝（ここで、ｎ１＞１、ｎ２＞１、およびｎ１＜ｎ２）は複数のある個数のループ、特に、ｎ１からｎ２個までのループと一致できる。この量化子、Ｂ｛ｎ１，ｎ２｝は、以下の量化子Ｂ｛ｎ１－１｝Ｂ｛１，ｎ２－ｎ１＋１｝に分割できる。第１の量化子は、ｎ１－１に等しい、いくつかのループに一致できる繰返し部分式Ｂである。この第１の量化子は、１からｎ２－ｎ１＋１までのいくつかのループによって一致できる繰返し部分式を有する第２の量化子と連結される。第１の量化子Ｂ｛ｎ１－１｝は、３０６で記述されるように展開される。

【００３０】

第２の量化子Ｂ｛１，ｎ２－ｎ１＋１｝は、結果として生じるオートマトンの入次数（ｉｎ－ｄｅｇｒｅｅ）および／または出次数（ｏｕｔ－ｄｅｇｒｅｅ）に基づいて展開できる。量化子の展開では、多数の入次数または多数の出次数を有する状態を作成できる。一例では、入次数はオートマトンの状態への遷移数に対応し、出次数はオートマトンの状態からの遷移数に対応する。その結果、第２の量化子は、第２の量化子をオートマトンに変換する際に、状態への遷移（入次数）または状態からの遷移（出次数）を制御するように展開できる。例えば、量化子は、各展開された状態の入次数を閾値数未満に制限するように展開できる。入次数の制限は、例えば、対象装置内の要素の態様および／または制限を考慮に入れるように実行できる。さらに、展開中に入次数を制限すると、コンパイラのために、以降の処理を削減できる。

【００３１】

一例では、量化子Ｂ｛１，ｎ２－ｎ１＋１｝を展開する際に、入次数と出次数との間のトレードオフとしてオートマトンが生成される。その結果、入次数を減らすと出次数が増加し得、また、出次数を減らすと入次数が増加し得る。一例では、量化子Ｂ｛１，ｎ２－ｎ１＋１｝のループ構造を展開するため、展開された状態への、または展開された状態からのどちらかの、いくつかの遷移が作成されて、オートマトンが連結されたＢの任意の文字列ｋ（ここで、１＜＝ｋ＜＝ｎ２－ｎ１＋１）を受け入れるようにする。展開された状態への遷移、または展開された状態からの遷移が作成されるかの制御は、オートマトンに対する入次数／出次数を制御するために使用できる。

【００３２】

方法３００は、単一の量化子に対応するものとして説明されているが、方法３００は、構文木内の複数の量化子に対して繰り返すことができ、結果として生じる別個のオートマトンが、次いで、さらに大きなオートマトンにリンクできる。

【００３３】

図４Ａは、式ＡＢ｛１，４｝が入次数を最小限にするように展開されるオートマトン４００の一例を示す。入次数を最小限にするように展開された量化子から生じたオートマトンは、本明細書では、散乱パターン（ｓｃａｔｔｅｒ　ｐａｔｔｅｒｎ）とも呼ばれる。式ＡＢ｛１，４｝の展開されたバージョンの散乱パターンは、式Ａ（（（（Ｂ？）Ｂ）？

Ｂ）？Ｂ）に直接対応し、それにオートマトン４００が対応する。オートマトン４００は、量化子Ｂ｛１，４｝に対する駆動状態４０２および、量化子の第１の状態４０４および量化子の最終状態４０８を含め、複数の展開された状態４０４～４０８を含む。一例では、式ＡおよびＢの各々は、図示されていない、もっと小さいオートマトンに対する複数の状態に対応できる。オートマトン４００の入次数を最小限にするために、量化子に対する遷移が、第１の状態４０４から他の展開された状態４０５～４０８への出遷移（ｏｕｔ－ｔｒａｎｓｉｔｉｏｎ）として割り当てられる。その結果、第１の状態４０４は、多数の出次数（４つの出遷移）を有し、また、全ての量化子状態４０４～４０８が少ない入次数（１つまたは２つの入遷移）を有する。

【００３４】

図４Ｂは、式ＡＢ｛１，４｝が出次数を最小限にするように展開されるオートマトン４１０の一例を示す。出次数を最小限にするように展開された量化子から生じたオートマトンは、本明細書では、マージパターン（ｍｅｒｇｅ　ｐａｔｔｅｒｎ）とも呼ばれる。式ＡＢ｛１，４｝の展開されたバージョンのマージパターンは、展開された式ＡＢ（Ｂ（Ｂ（Ｂ）？）？）？に直接対応する。オートマトン408は、状態４０２、４０４～４０８の間で異なる遷移をもつオートマトン４００と同じ状態４０２、４０４～４０８を含む。オートマトン４１０の出次数を最小限にするために、量化子に対する遷移が、量化子の最終状態４０８への入遷移（ｉｎ－ｔｒａｎｓｉｔｉｏｎ）として割り当てられる。入遷移は、展開された状態４０４～４０７の各々から来る。その結果、量化子状態４０４～４０８の全ては少ない出次数（１つまたは２つの出遷移）を有するが、量化子の最終状態４０８は多数の入次数（４つの入遷移）を有する。

【００３５】

一例では、量化子を有する式は、出次数または入次数のうちの１つを閾値未満に制限するように展開される。一例では、式ＡＢ｛１，ｎ１｝を、入次数を閾値に制限するように展開するために、量化子Ｂ｛１，ｎ１｝に対する閾値までのいくつかの遷移を、入遷移として、量化子Ｂ｛１，ｎ１｝の最終状態に割り当てることができ、また、他の遷移を、出遷移として、量化子Ｂ｛１，ｎ１｝の第１の状態に対して割り当てることができる。逆に、式ＡＢ｛１，ｎ１｝を、出次数を閾値に制限するように展開するために、量化子Ｂ｛１，ｎ１｝に対する閾値までのいくつかの遷移を、出遷移として、量化子に対する第１の状態に割り当てることができ、また、他の遷移を、出遷移として、量化子Ｂ｛１，ｎ１｝の最終状態に対して割り当てることができる。

【００３６】

図４Ｃは、式ＡＢ｛１，４｝が、任意の状態に対する入遷移を３以下に制限するように展開されるオートマトン４１２の別の例を示す。オートマトン４１２は、状態４０２、４０４～４０８の間で異なる遷移をもつオートマトン４００および４０８と同じ状態４０２、４０４～４０８を含む。一例では、オートマトン４１２の入次数を３以下の入遷移に制限するために、量化子に対する遷移が、最初は、量化子Ｂ｛１，４｝の最終状態４０８への入遷移として、制限の３に達するまで割り当てられ、また、他の遷移が、量化子状態４０４～４０８からの出遷移として割り当てられる。従って、量化子の最終状態４０８ならびに他の量化子状態４０４～４０７が、制限の３つ以下の入次数を有し、また、第１の状態４０４が３つの出次数を有する。

【００３７】

他の例では、式の入次数および出次数は、互いを特定の割合（例えば、１対１、２対１）に設定できる。さらに別の例では、式の入次数および出次数は、入遷移または出遷移のどちらかに対して閾値に達するまで、互いを特定の割合に設定でき、次いで、別の割合が使用できるか、または遷移の全てが、それぞれ入遷移または出遷移として割り当てられ得る。

【００３８】

〔実施形態例〕

図５～１５に関連した以下の説明は、並列マシン内に、制限された入次数および／また

は出次数をもつオートマトンを実装する実施形態例に関する。図５～図９に関連した説明は、量化子を実装するための並列マシン例に関し、また、図１０～図１４に関連した説明は、量化子を実装する並列マシンをプログラムするための機械コードを生成するコンパイラ例に関する。

【００３９】

図５は、データ分析用の階層構造を実装するために使用できる並列マシン５００の例を示す。並列マシン５００は、入力データを受信し、その入力データに基づいて出力を提供できる。並列マシン５００は、入力データを受信するためのデータ入力ポート５１０および出力を別の装置に提供するための出力ポート５１４を含むことができる。データ入力ポート５１０は、並列マシン５００に入力されるデータ用のインタフェースを提供する。

【００４０】

並列マシン５００は、汎用要素５０２および専用要素５１２を含む、複数のプログラム可能要素を含む。汎用要素５０２は、１つまたは複数の入力５０４および１つまたは複数の出力５０６を含み得る。汎用要素５０２は、複数の状態のうちの１つにプログラムされ得る。汎用要素５０２の状態は、汎用要素５０２が、所与の入力に基づいて、何の出力を提供するかを決定する。すなわち、汎用要素５０２の状態は、プログラム可能要素が所与の入力に基づいてどのように反応するかを決定する。データ入力ポート５１０に対するデータ入力は、汎用要素５０２にそれに対する処置を取らせるため、複数の汎用要素５０２に提供され得る。汎用要素５０２の例には、以下で詳細に説明する状態機械要素（ＳＭＥ）、および構成可能論理ブロックを含むことができる。一例では、ＳＭＥは、所与の入力がデータ入力ポート５１０で受信される場合に、特定の出力（例えば、高信号つまり「１」信号）を提供するため、所与の状態に設定され得る。所与の入力以外の入力がデータ入力ポート５１０で受信される場合、ＳＭＥは異なる出力（例えば、低信号つまり「０」信号）を提供し得る。一例では、構成可能論理ブロックは、データ入力ポート５１０で受信された入力に基づいて、ブール論理関数（例えば、ＡＮＤ、ＯＲ、ＮＯＲなど）を実行するように設定できる。

【００４１】

並列マシン５００は、プログラム（例えば、イメージ）を並列マシン５００上にロードするためのプログラミングインタフェース５１１も含むことができる。イメージは、汎用要素５０２の状態をプログラム（例えば、設定）できる。すなわち、イメージは、所与の入力に対して特定の方法で反応するように汎用要素５０２を構成できる。例えば、汎用要素５０２は、文字‘ａ’がデータ入力ポート５１０で受信される場合、高信号を出力するように設定できる。いくつかの例では、並列マシン５００は、汎用要素５０２の動作のタイミングを制御するためのクロック信号を使用できる。ある例では、並列マシン５００は、汎用要素５０２とやりとりするため、および専用機能を実行するために、専用要素５１２（例えば、ＲＡＭ、論理ゲート、カウンタ、ルックアップテーブルなど）を含むことができる。いくつかの実施形態では、データ入力ポート５１０で受信されるデータは、長い時間をかけて、もしくは一度にそろって受信されたデータの一定のセット、または長い時間をかけて受信されたデータのストリームを含むことができる。データは、並列マシン５００に結合された、データベース、センサー、ネットワークなどの任意のソースから受信され得るか、または生成され得る。

【００４２】

並列マシン５００は、並列マシン５００の異なる要素（例えば、汎用要素５０２、データ入力ポート５１０、出力ポート５１４、プログラミングインタフェース５１１、および専用要素５１２）を選択的に共に結合するための複数のプログラム可能スイッチ５０８も含む。その結果、並列マシン５００は、要素間で形成されたプログラム可能マトリックス（ｐｒｏｇｒａｍｍａｂｌｅ　ｍａｔｒｉｘ）を含む。一例では、汎用要素５０２の入力５０４、データ入力ポート５１０、プログラミングインタフェース５１１、または専用要素５１２が、１つまたは複数のプログラム可能スイッチ５０８を通して、汎用要素５０２の出力５０６、出力ポート５１４、プログラミングインタフェース５１１、または専用要

素５１２に結合できるように、プログラム可能スイッチ５０８が、２つ以上の要素を選択的に互いに結合できる。従って、要素間の信号のルーティングが、プログラム可能スイッチ５０８を設定することによって制御できる。図５は、所与の要素とプログラム可能スイッチ５０８との間の特定数の導体（例えば、ワイヤー）を示しているが、他の例では、異なる数の導体が使用できることが理解されるはずである。また、図５は、個別にプログラム可能スイッチ５０８に結合された各汎用要素５０２を示しているが、他の例では、複数の汎用要素５０２がグループ（例えば、図８に示すような、ブロック８０２）としてプログラム可能スイッチ５０８に結合できる。一例では、データ入力ポート５１０、データ出力ポート５１４、および／またはプログラミングインタフェース５１１はレジスタとして実装でき、レジスタへの書込みが、それぞれの要素へデータを提供するか、またはそれぞれの要素からデータを提供するようにする。

【００４３】

一例では、単一の並列マシン５００が物理デバイス上に実装されるが、他の例では、２つ以上の並列マシン５００が単一の物理デバイス（例えば、物理チップ）上に実装できる。一例では、複数の並列マシン５００の各々が、別個のデータ入力ポート５１０、別個の出力ポート５１４、別個のプログラミングインタフェース５１１、および汎用要素５０２の別個のセットを含むことができる。さらに、汎用要素５０２の各セットは、それらの対応する入力データポート５１０でデータに反応（例えば、高信号または低信号を出力）できる。例えば、第１の並列マシン５００に対応する汎用要素５０２の第１のセットは、第１の並列マシン５００に対応する第１のデータ入力ポート５１０でデータに反応できる。第２の並列マシン５００に対応する汎用要素５０２の第２のセットは、第２の並列マシン５００に対応する第２のデータ入力ポート５１０に反応できる。その結果、各並列マシン５００は、１セットの汎用要素５０２を含み、異なるセットの汎用要素５０２は、異なる入力データに反応できる。同様に、各並列マシン５００、および汎用要素５０２の各対応するセットは、別個の出力を提供できる。いくつかの例では、第２の並列マシン５００に対する入力データが、第１の並列マシン５００からの出力データを含むことができるように、第１の並列マシン５００からの出力ポート５１４が、第２の並列マシン５００の入力ポート５１０に結合できる。

【００４４】

一例では、並列マシン５００上にロードするためのイメージは、汎用要素５０２の状態を設定するため、プログラム可能スイッチ５０８をプログラミングするため、および並列マシン５００内で専用要素５１２を構成するための複数のビットの情報を含む。一例では、ある入力に基づいて所望の出力を提供するように並列マシン５００をプログラムするために、イメージが並列マシン５００上にロードできる。出力ポート５１４は、データ入力ポート５１０での汎用要素５０２のデータに対する反応に基づいて、並列マシン５００からの出力を提供できる。出力ポート５１４からの出力は、所与のパターンの一致を示す単一ビット、複数のパターンに対する一致および不一致を示す複数のビットを含むワード（ｗｏｒｄ）、ならびに所与の時に全てまたは特定の汎用要素５０２の状態に対応する状態ベクトルを含むことができる。

【００４５】

並列マシン５００に関する使用例には、パターン認識（例えば、音声認識、画像認識など）、信号処理、画像処理、コンピュータビジョン、暗号化、およびその他を含む。ある例では、並列マシン５００は、有限状態機械（ＦＳＭ）エンジン、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、およびそれらの変形を含むことができる。さらに、並列マシン５００は、コンピュータ、ポケットベル、携帯電話、電子手帳、携帯音楽プレーヤ、ネットワーク装置（例えば、ルータ、ファイアウォール、スイッチ、またはそれらの任意の組合せ）、制御回路、カメラなどの、大きな装置内のコンポーネントであり得る。

【００４６】

図６～図９は、本明細書で「ＦＳＭエンジン６００」と称する並列マシンの一例を示す。一例では、ＦＳＭエンジン６００は、有限状態機械のハードウェア実装を含む。その結

果、ＦＳＭエンジン６００は、ＦＳＭ内の複数の状態に対応する、複数の選択的に結合可能なハードウェア要素（例えば、プログラム可能要素）を実装する。ＦＳＭ内の状態と同様に、ハードウェア要素は、入力ストリームを分析し、その入力ストリームに基づいて下流のハードウェア要素をアクティブにできる。

【００４７】

ＦＳＭエンジン６００は、汎用要素および専用要素を含む、複数のプログラム可能要素を含む。汎用要素は、多数の異なる機能を実装するようにプログラムされ得る。これらの汎用要素は、行（ｒｏｗ）６０６（図７および図８に示す）およびブロック６０２（図６および図７に示す）に階層的に編成されているＳＭＥ ６０４、６０５（図９に示す）を含む。階層的に編成されたＳＭＥ ６０４、６０５間で信号をルーティングするために、ブロック間スイッチ６０３（図６および図７に示す）、ブロック内スイッチ６０８（図７および図８に示す）、ならびに行内（ｉｎｔｒａ－ｒｏｗ）スイッチ６１２（図８に示す）を含む、プログラム可能スイッチの階層が使用される。ＳＭＥ ６０４、６０５は、ＦＳＭエンジン６００によって実装されたＦＳＭの状態に対応できる。ＳＭＥ ６０４、６０５は、以下で説明するように、プログラム可能スイッチの使用によって共に結合できる。その結果、ＦＳＭは、状態の機能に対応するようにＳＭＥ ６０４、６０５をプログラミングすることにより、また、ＦＳＭ内の状態間の遷移に対応するため、ＳＭＥ ６０４、６０５を選択的に共に結合することにより、ＦＳＭエンジン６００上に実装できる。

【００４８】

図６は、ＦＳＭエンジン例６００の全体図を示す。ＦＳＭエンジン６００は、プログラム可能ブロック間スイッチ６０３と選択的に共に結合できる複数のブロック６０２を含む。さらに、ブロック６０２は、信号（例えば、データ）を受信するため、およびデータをブロック６０２に提供するために、入力ブロック６０９（例えば、データ入力ポート）に選択的に結合できる。ブロック６０２は、ブロック６０２から外部デバイス（例えば、別のＦＳＭエンジン６００）に信号を提供するために、出力ブロック６１３（例えば、出力ポート）にも選択的に結合できる。ＦＳＭエンジン６００は、プログラム（例えば、イメージ）をＦＳＭエンジン６００上にロードするために、プログラミングインタフェース６１１も含み得る。イメージは、ＳＭＥ ６０４、６０５の状態をプログラム（設定）できる。すなわち、イメージは、入力ブロック６０９で所与の入力に対して特定の方法で反応するように、ＳＭＥ ６０４、６０５を構成できる。例えば、ＳＭＥ ６０４は、入力ブロック６０９で文字‘ａ’が受信される場合、高信号を出力するように設定できる。

【００４９】

一例では、入力ブロック６０９、出力ブロック６１３、および／またはプログラミングインタフェース６１１はレジスタとして実装でき、そのレジスタへの書込みがそれぞれの要素に対して、またはそれぞれの要素から、データを提供できるようにする。その結果、プログラミングインタフェース６１１に対応するレジスタ内に格納されているイメージからのビットが、ＳＭＥ ６０４、６０５上にロードできる。図６は、ブロック６０２、入力ブロック６０９、出力ブロック６１３、およびブロック間スイッチ６０３の間に特定数の導体（例えば、ワイヤ、線）を示しているが、他の例では、もっと少ないかまたは多い導体が使用できることが理解されるはずである。

【００５０】

図７は、ブロック６０２の一例を示す。ブロック６０２は、プログラム可能ブロック内スイッチ６０８と選択的に共に結合できる複数の行６０６を含むことができる。その結果、行６０６は、ブロック間スイッチ６０３で、別のブロック６０２内の別の行６０６と選択的に結合できる。一例では、バッファ６０１は、ブロック間スイッチ６０３への／ブロック間スイッチ６０３からの信号のタイミングを制御するために含まれる。行６０６は、本明細書で２つのグループ（ＧＯＴ：ｇｒｏｕｐ ｏｆ ｔｏｗ）６１０として参照される要素のペアに編成された複数のＳＭＥ ６０４、６０５を含む。一例では、ブロック６０２は、１６個の行６０６を含む。

【００５１】

図８は、行６０６の一例を示す。ＧＯＴ ６１０は、プログラム可能行内スイッチ６１２によって、行６０６内の他のＧＯＴ ６１０および任意の他の要素６２４と選択的に結合できる。また、ＧＯＴ ６１０は、ブロック内スイッチ６０８で他の行６０６内の他のＧＯＴ ６１０と、またはブロック間スイッチ６０３で他のブロック６０２内の他のＧＯＴ ６１０とも結合できる。一例では、ＧＯＴ ６１０は、第１および第２の入力６１４、６１６ならびに出力６１８を有する。第１の入力６１４は、ＧＯＴ ６１０の第１のＳＭＥ ６０４と結合され、第２の入力６１４は、ＧＯＴ ６１０の第２のＳＭＥ ６０４と結合される。

【００５２】

一例では、行６０６は、第１および第２の複数の行相互接続導体６２０、６２２を含む。一例では、ＧＯＴ ６１０の入力６１４、６１６は、１つまたは複数の行相互接続導体６２０、６２２に結合でき、出力６１８は、１つの行相互接続導体６２０、６２２に結合できる。一例では、第１の複数の行相互接続導体６２０は、行６０６内の各ＧＯＴ ６１０の各ＳＭＥ ６０４に結合できる。第２の複数の行相互接続導体６２２は、行６０６内の各ＧＯＴ ６１０の１つのＳＭＥ ６０４に結合できるが、そのＧＯＴ ６１０の他のＳＭＥ ６０４には結合できない。一例では、第２の複数の行相互接続導体６２２の半分の第１の方が、行６０６内のＳＭＥ ６０４の半分の第１の方（各ＧＯＴ ６１０からの１つのＳＭＥ ６０４）と結合でき、第２の複数の行相互接続導体６２２の半分の第２の方が、行６０６内のＳＭＥ ６０４の半分の第２の方（各ＧＯＴ ６１０からのその他のＳＭＥ ６０４）と結合できる。第２の複数の行相互接続導体６２２とＳＭＥ ６０４、６０５との間の制限された接続性は、本明細書では「パリティ」と呼ばれる。

【００５３】

一例では、行６０６は、カウンタ、プログラム可能ブール論理要素、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、特定用途向け集積回路（ＡＳＩＣ）、プログラム可能プロセッサ（例えば、マイクロプロセッサ）、および他の要素などの、専用要素６２４も含むことができる。追加として、一例では、専用要素６２４は、異なる行６０６内では異なる。例えば、ブロック６０２内の４つの行６０６が、専用要素６２４としてブール論理を含むことができ、ブロック６０２内の他の８つの行６０６が、専用要素６２４としてカウンタを含むことができる。

【００５４】

一例では、専用要素６２４は、カウンタ（本明細書ではカウンタ６２４とも呼ばれる）を含む。一例では、カウンタ６２４は、１２ビットのプログラマブル減算カウンタを含む。１２ビットのプログラマブルカウンタ6２４は、カウント入力、リセット入力、およびゼロカウント出力を有する。カウント入力は、アサート時に、カウンタ６２４の値を１だけ減算する。リセット入力は、アサート時に、カウンタ６２４に、関連したレジスタから初期値をロードさせる。１２ビットのカウンタ６２４に対して、最大で１２ビット数が初期値としてロードできる。カウンタ６２４の値がゼロ（０）に減算されると、ゼロカウント出力がアサートされる。カウンタ６２４は、少なくとも２つのモード、パルスおよび保留（ｈｏｌｄ）を有する。カウンタ６２４がパルスモードに設定されている場合、カウンタ６２４がゼロに減算されると、第１のクロック周期中にゼロカウント出力がアサートされ、また、次のクロック周期では、たとえカウント入力がアサートされても、ゼロカウント出力はもはやアサートされない。この状態は、カウンタ６２４が、アサートされているリセット入力によってリセットされるまで、継続する。カウンタ６２４が保留モードに設定されている場合、カウンタ６２４がゼロに減算されると、第１のクロック周期中にゼロカウント出力がアサートされ、また、カウント入力がアサートされる場合、カウンタ６２４が、アサートされているリセット入力によってリセットされるまで、アサートされたままである。

【００５５】

図９は、ＧＯＴ ６１０の一例を示す。ＧＯＴ ６１０は、入力６１４、６１６を有し、かつそれらの出力６２６、６２８がＯＲゲート６３０に結合されている、第１および第２

のＳＭＥ　６０４、６０５を含む。出力６２６、６２８は、ＧＯＴ　６１０の共通の出力６１８を形成するため、ＯＲゲート６３０で共に論理ＯＲされる。一例では、第１および第２のＳＭＥ　６０４、６０５はパリティを示し、そこで、第１のＳＭＥ　６０４の入力６１４は、行相互接続導体６２２のいくつかに結合でき、第２のＳＭＥ　６０５の入力６１６は、他の行相互接続導体６２２に結合できる。一例では、ＧＯＴ　６１０内の２つのＳＭＥ　６０４、６０５は、第１のＳＭＥ　６０４の出力６２６を第２のＳＭＥ　６０５の入力６１６に結合するために、スイッチ６４０を設定することにより、カスケード接続できる。

【００５６】

一例では、状態機械要素６０４、６０５は、検出線６３４と平行に結合された、ダイナミックランダムアクセスメモリ（ＤＲＡＭ）でしばしば使用されるような、複数のメモリセル６３２を含む。１つのかかるメモリセル６３２は、高値または低値（例えば、１または０）のいずれかに対応するものなどの、データ状態に設定できるメモリセルを含む。メモリセル６３２の出力は、検出線６３４に結合され、メモリセル６３２への入力は、データストリーム線６３６上のデータに基づいて、信号を受信する。一例では、データストリーム線６３６上の入力が、メモリセル６３２の１つを選択するために復号される。選択されたメモリセル６３２は、その格納されたデータ状態を出力として検出線６３４上に提供する。例えば、データ入力ポート６０９で受信されたデータがデコーダ（図示せず）に提供でき、デコーダは、データストリーム線６３６のうちの１本を選択できる。一例では、デコーダは、ＡＳＣＩＩ文字を２５６ビットのうちの１つに変換できる。

【００５７】

メモリセル６３２は、それ故、メモリセル６３２が高値に設定され、かつ、データストリーム線６３６上のデータがメモリセル６３２に対応する場合、高信号を検出線６３４に出力する。データストリーム線６３６上のデータがメモリセル６３２に対応し、メモリセル６３２が低値に設定されている場合、メモリセル６３２は、低信号を検出線６３４に出力する。メモリセル６３２からの検出線６３４上への出力は、検出回路６３８によって検知される。一例では、入力線６１４、６１６上の信号は、それぞれの検出回路６３８をアクティブ状態または非アクティブ状態のいずれかに設定する。非アクティブ状態に設定されると、検出回路６３８は、それぞれの検出線６３４上の信号にかかわらず、低信号をそれぞれの出力６２６、６２８上に出力する。アクティブ状態に設定されると、高信号がそれぞれのＳＭＥ　６０４、６０５のメモリセル６３４の１つから検出される場合、検出回路６３８は、高信号をそれぞれの出力線６２６、６２８上に出力する。アクティブ状態にある場合、それぞれのＳＭＥ　６０４、６０５のメモリセル６３４の全てからの信号が低ければ、検出回路６３８は、低信号をそれぞれの出力線６２６、６２８上に出力する。

【００５８】

一例では、ＳＭＥ　６０４、６０５は、２５６のメモリセル６３２を含み、各メモリセル６３２は、異なるデータストリーム線６３６に結合される。それ故、ＳＭＥ　６０４、６０５は、選択された１つまたは複数のデータストリーム線６３６がその上に高信号を有する場合、高信号を出力するようにプログラムされ得る。例えば、ＳＭＥ　６０４は、第１のメモリセル６３２（例えば、ビット０）を高に設定し、他の全てのメモリセル６３２（例えば、ビット１～２５５）を低に設定できる。それぞれの検出回路６３８がアクティブ状態にあるとき、ＳＭＥ　６０４は、ビット０に対応するデータストリーム線６３６がその上に高信号を有する場合、高信号を出力６２６上に出力する。他の例では、複数のデータストリーム線６３６のうちの１本が、適切なメモリセル６３２を高値に設定することにより、その上に高信号を有する場合、ＳＭＥ　６０４は、高信号を出力するように設定できる。

【００５９】

一例では、メモリセル６３２は、関連したレジスタからビットを読み取ることにより、高値または低値に設定できる。その結果、ＳＭＥ　６０４は、コンパイラによって作成されたイメージをレジスタ内に格納し、レジスタ内のビットを関連したメモリセル６３２に

ロードすることにより、プログラムされ得る。一例では、コンパイラによって作成されたイメージは、高および低（例えば、１および０）ビットのバイナリイメージを含む。イメージは、ＳＭＥ ６０４、６０５をカスケード接続することにより、ＦＳＭエンジン６００をＦＳＭとして動作するようにプログラムできる。例えば、第１のＳＭＥ ６０４は、検出回路６３８をアクティブ状態に設定することにより、アクティブ状態に設定できる。第１のＳＭＥ ６０４は、ビット０に対応するデータストリーム線６３６がその上に高信号を有する場合、高信号を出力するように設定できる。第２のＳＭＥ ６０５は、最初は非アクティブ状態に設定できるが、アクティブなとき、ビット１に対応するデータストリーム線６３６がその上に高信号を有する場合、高信号を出力するように設定できる。第１のＳＭＥ ６０４および第２のＳＭＥ ６０５は、第１のＳＭＥ ６０４の出力６２６を第２のＳＭＥ ６０５の入力６１６に結合するように設定することにより、カスケード接続できる。従って、高信号が、ビット０に対応するデータストリーム線６３６上で検知される場合、第１のＳＭＥ ６０４は、高信号を出力６２６上に出力し、第２のＳＭＥ ６０５の検出回路６３８をアクティブ状態に設定する。高信号が、ビット１に対応するデータストリーム線６３６上で検知される場合、第２のＳＭＥ ６０５は、別のＳＭＥ ６０５をアクティブにするため、またはＦＳＭエンジン６００からの出力のため、高信号を出力６２８上に出力する。

【００６０】

図１０は、コンパイラがソースコードを、並列マシンをプログラムするように構成されたイメージに変換するための方法１０００の一例を示す。方法１０００は、ソースコードを構文木に解析すること（ブロック１００２）、構文木をオートマトンに変換すること（ブロック１００４）、オートマトンを最適化すること（ブロック１００６）、オートマトンをネットリストに変換すること（ブロック１００８）、ネットリストをハードウェア上に配置すること（ブロック１０１０）、ネットリストをルーティングすること（ブロック１０１２）、および結果として生じるイメージを公開すること（ブロック１０１４）を含む。

【００６１】

一例では、コンパイラは、ソフトウェア開発者がＦＳＭエンジン６００上でＦＳＭを実装するために、イメージを作成できるようにする、アプリケーションプログラミングインタフェース（ＡＰＩ）を含む。コンパイラは、正規表現の入力セットを、ＦＳＭエンジン６００をプログラムするように構成されているイメージに変換するための方法を提供する。コンパイラは、フォンノイマン型アーキテクチャを有するコンピュータ用の命令によって実装できる。これらの命令は、コンピュータ上のプロセッサにコンパイラの機能を実装させることができる。例えば、命令は、プロセッサによって実行される場合、プロセッサがアクセス可能なソースコード上のブロック１００２、１００４、１００６、１００８、１０１０、１０１２、および１０１４に記載する動作をプロセッサに実行させることができる。フォンノイマン型アーキテクチャを有するコンピュータ例が図１５に示され、以下で説明される。

【００６２】

ブロック１００２で、構文木を形成するためにソースコードが解析される。解析することで、図３に関して上述したように、ソースコードの一般的表現が作成される。さらに、解析することで、ＦＳＭエンジン６００によってサポートされている正規表現、およびサポートされていない正規表現を考慮に入れることができる。サポートされている正規表現は、適切な機械コード実装に変換できるが、サポートされていない正規表現は、例えば、エラーを生じ得るか、または、機能において、サポートされていない正規表現に近い、サポートされた機械コードに変換され得る。

【００６３】

ブロック１００４で、構文木は、オートマトンに変換される。図３に関連して前述したように、構文木の変換は、構文木を、複数の状態を含むオートマトンに変換する。一例では、オートマトンは、ＦＳＭエンジン６００のハードウェアに一部基づいて、変換され得

る。

【００６４】

一例では、オートマトンに対する入力シンボルは、アルファベット、０～９の数字、および他の印刷可能な文字のシンボルを含む。一例では、入力シンボルは、バイト値０～２５５で表される。一例では、オートマトンは、グラフのノードが状態の集合に対応する有向グラフとして表され得る。一例では、入力シンボルαを受けての状態ｐから状態ｑへの遷移、すなわち、δ（ｐ，α）は、ノードｐからノードｑへの有向接続によって示される。

【００６５】

一例では、オートマトンは、汎用状態ならびに専用状態を含む。汎用状態および専用状態は、コンパイラがそれに対して機械コードを生成している対象装置によってサポートされる汎用要素および専用要素に対応する。異なるタイプの対象装置は、異なるタイプの汎用要素ならびに１つまたは複数の異なるタイプの専用要素をサポートできる。汎用要素は、通常、幅広い範囲の機能を実装するために使用でき、他方、専用要素は、通常、もっと狭い範囲の機能を実装するために使用できる。しかし、一例では、専用要素は、例えば、その狭い範囲の機能内でより大きな効果を得ることができる。その結果、専用要素は、例えば、対象装置において特定の機能を実装するために必要なマシンサイクルまたはマシンリソースを削減するために使用できる。いくつかの例では、対象装置は、単に専用要素をサポートし、そこで、複数の異なるタイプの専用要素がサポートされる。

【００６６】

コンパイラがＦＳＭエンジン６００に対して機械コードを生成している例では、汎用状態はＳＭＥ　６０４、６０５に対応でき、汎用状態はそれに応じて本明細書では「ＳＭＥ状態」と呼ばれる。さらに、コンパイラがＦＳＭエンジン６００に対して機械コードを生成している場合、専用状態は、カウンタ６２４に対応でき、それに応じて本明細書では「カウンタ状態」と呼ばれる。一例では、オートマトン内のＳＭＥ状態は、ＳＭＥにマッピングしないオートマトンの開始状態を除いて、ＦＳＭエンジン６００内のＳＭＥ（例えば、ＳＭＥ　６０４、６０５）に１：１でマッピングする。カウンタ６２４は、カウンタ状態に１：１でマッピングすることもあれば、しないこともある。

【００６７】

一例では、入力シンボル範囲外の特別な遷移シンボルが、オートマトン内で使用され得る。これらの特別な遷移シンボルは、例えば、専用要素224の使用を可能にするために使用できる。さらに、特別な遷移シンボルは、入力シンボル以外の何かを受けて起こる遷移を提供するために使用できる。例えば、特別な遷移シンボルは、第２の状態および第３の状態の両方が有効な場合に、第１の状態が有効にされる（例えば、遷移される）ことを示し得る。その結果、第１の状態は、第２の状態および第３の状態の両方がアクティブにされる場合にアクティブにされ、また、第１の状態への遷移は、入力シンボルに直接依存しない。特に、第２の状態および第３の状態の両方が有効な場合に、第１の状態が有効にされることを示す特別な遷移シンボルは、例えば、専用要素224としてのブール論理によって実行されるブールＡＮＤ関数を表すために使用できる。一例では、特別な遷移シンボルは、カウンタ状態がゼロに達したこと、従って下流の状態への遷移を示すために使用できる。

【００６８】

一例では、オートマトンは、グルシコフの方法などの、標準的な方法の１つを使用して構築できる。一例では、オートマトンは、εのない（ε－ｆｒｅｅ）均質な（ｈｏｍｏｇｅｎｏｕｓ）オートマトンであり得る。構文木のオートマトンへの変換に関するさらなる詳細は、以下で、図３、図４Ａ～図４Ｃ、図１１Ａ、図１１Ｂ、図１２、図１３、および図１４に関連して提供される。

【００６９】

ブロック１００６で、構文木がオートマトンに変換されると、オートマトンが最適化される。オートマトンは、とりわけ、その複雑性およびサイズを縮小するように最適化され

得る。オートマトンは、重複する状態を結合することにより最適化できる。

【００７０】

ブロック１００８で、最適化されたオートマトンがネットリストに変換される。オートマトンをネットリストに変換することは、オートマトンの各状態をＦＳＭエンジン６００上のハードウェア要素（例えば、ＳＭＥ　６０４、６０５、専用要素６２４）のインスタンスにマッピングする。また、インスタンス間の接続が、ネットリストを作成するために決定される。

【００７１】

ブロック１０１０で、ネットリストの各インスタンスに対応する、対象装置の特定のハードウェア要素（例えば、ＳＭＥ　６０４、６０５、専用要素６２４）を選択するために、ネットリストが位置付けられる。一例では、位置付けられると、ＦＳＭエンジン６００に対する一般入力および出力制約に基づいて、各特定のハードウェア要素が選択される。

【００７２】

ブロック１０１２で、ネットリストによって記述された接続を達成するために、選択されたハードウェア要素を共に結合するため、位置付けられたネットリストが、プログラム可能スイッチ（例えば、ブロック間スイッチ６０３、ブロック内スイッチ６０８、および行内スイッチ６１２）に対する設定を決定するようにルーティングされる。一例では、プログラム可能スイッチに対する設定が、選択されたハードウェア要素を接続するために使用されるＦＳＭエンジン６００の特定の導体を決定することにより決定される。ルーティングは、ブロック１０１０で位置付けるハードウェア要素間の接続のさらに詳しい制限を考慮に入れることができる。その結果、ルーティングは、ＦＳＭエンジン６００上の導体の実際の制限を前提として、適切な接続を行うために、グローバルな位置付けによって決定された通りに、いくつかのハードウェア要素の位置付けを調整し得る。

【００７３】

ネットリストが位置付けられルーティングされると、位置付けられルーティングされたネットリストは、ＦＳＭエンジン２００のプログラミング用に複数のビットに変換できる。複数のビットはここではイメージと呼ばれる。

【００７４】

ブロック１０１４で、イメージがコンパイラによって公開される。イメージは、ＦＳＭエンジン６００の特定のハードウェア要素および／またはプログラム可能スイッチをプログラミングするための複数のビットを含む。イメージが複数のビット（例えば、０および１）を含む実施形態では、イメージは、バイナリイメージと呼ぶことができる。ビットは、プログラム化されたＦＳＭエンジン６００が、ソースコードによって記述された機能を有するＦＳＭを実装するように、ＳＭＥ　６０４、６０５の状態、専用要素６２４、およびプログラム可能スイッチをプログラムするためにＦＳＭエンジン６００上にロードできる。位置付け（ブロック１０１０）およびルーティング（ブロック１０１２）は、特定のハードウェア要素をＦＳＭエンジン６００内の特定の位置で、オートマトン内の特定の状態にマッピングできる。その結果、イメージ内のビットは、特定のハードウェア要素および／またはプログラム可能スイッチを所望の機能を実装するようにプログラムできる。一例では、イメージは、機械コードをコンピュータ可読媒体に保存することにより公開できる。別の例では、イメージは、イメージをディスプレイ装置上に表示することにより公開できる。さらに別の例では、イメージは、イメージをＦＳＭエンジン６００上にロードするためのプログラミング装置などの、別の装置にイメージを送信することにより、公開できる。さらにまた別の例では、イメージは、イメージを並列マシン（例えば、ＦＳＭエンジン６００）上にロードすることにより、公開できる。

【００７５】

一例では、イメージは、イメージからのビット値をＳＭＥ　６０４、６０５、および他のハードウェア要素６２４に直接ロードするか、またはイメージを１つまたは複数のレジスタにロードし、次いで、レジスタからのビット値をＳＭＥ　６０４、６０５、および他のハードウェア要素６２４に書き込むかのいずれかにより、ＦＳＭエンジン６００上にロ

ードできる。一例では、ＦＳＭエンジン６００のハードウェア要素（例えば、ＳＭＥ ６０４、６０５、他の要素６２４、プログラム可能スイッチ６０３、６０８、６１２）が、プログラミング装置および／またはコンピュータが、イメージを１つまたは複数のメモリアドレスに書き込むことにより、イメージをＦＳＭエンジン６００上にロードできるように、メモリマップされる。

【００７６】

ここで、図３および図４Ａ～図４Ｃを再度参照すると、構文木をオートマトンに変換する際に、展開された量化子の入次数および／または出次数が制御できる。一例では、入次数および／または出次数は、閾値に基づいて制御される。閾値は、例えば、コンパイラがイメージを準備している並列マシン５００のハードウェア制限に基づくことができる。例えば、並列マシン５００の汎用要素５０２が最大１４の入遷移を有することができる場合、展開に対する閾値は、１４の入遷移以下に設定できる。別の例では、展開に対する閾値は、最大入遷移の一部が使用されるように、設定できる。例えば、最大１４の入遷移を有する汎用要素５０２に対して、閾値は、展開している式中ではない式からの入遷移のために、７つの入遷移に設定できる。

【００７７】

展開は、以下で説明するように、一般的なグラフ問題として定式化できる。図１１Ａ、図１１Ｂ、図１２、および図１３は、以下で議論する問題１に対する４つの解決策のグラフを示し、そこでは、入次数および出次数が制御される。異なる解決策は異なる最大長を提供するが、それは、入次数および出次数閾値によって決定される。その結果、所望の入次数および出次数に応じて、所与の解決策が選択され得る。結果として得られるオートマトンの入次数および／または出次数は、以下で議論する２つの問題に対してどの解決策が選択されるかの関数である。

【００７８】

問題１（入次数および出次数制約のある経路を拡張する）。長さｎの有向路ならびに２つの符号なし整数ｐ＞１およびｑ＞１を前提とする。ｎ＋１個のバーテックスをｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｎとして示す。経路の開始バーテックスがｖ０であり、終了バーテックスがｖｎである。ループが導入されないように、エッジをこの経路に追加し、任意のｋ（１＜＝ｋ＜＝ｎ）に対して、長さｋでｖ０からｖｎまでの有向路が存在し；また、全てのバーテックスの入次数は、ｐ以下でなければならず、かつ、全てのバーテックスの出次数はｑ以下でなければならない。

【００７９】

問題２（入次数および出次数制約のある経路を最大限にする）。無限有向路ならびに２つの符号なし整数ｐ＞１およびｑ＞１を前提とする。バーテックスをｖ０－＞ｖ１－＞ｖ２－＞．．．として示す。エッジをこの経路に追加し、ｖｍとして示される、終了バーテックスを見つけ、任意のｋ（１＜＝ｋ＜＝ｍ）に対して、長さｋでｖ０からｖｍまでの有向路が存在するようにし；全てのバーテックスの入次数は、ｐ以下でなければならず、かつ、全てのバーテックスの出次数はｑ以下でなければならない。グラフ内でループは許可されない。最適化の目標は、結果の経路の長さｍを最大限にすることである。

【００８０】

一例では、問題２が最初に解決される。最大限にされた経路長ｍがｎ以上の場合、問題１は容易に解決できる。図１１～図１３に関連して以下で説明する４つの解決策は、図４Ａ～図４Ｃに関連して説明した散乱パターンおよびマージパターンから導出される。最大長は、ｐおよびｑの値によって決まる。

【００８１】

グラフの用語では、散乱パターンは、ｑ個のバーテックスをもつ経路の拡張として説明できる。元の経路を含むエッジに加えて、開始バーテックス、ｖ０が他の全てのバーテックスを駆動する。ｖ０の出次数がｑ－１、ｖｑ－１が０、および他のバーテックスｖ１．．．ｖｑ－２が１である。ｖ０の入次数が０、ｖ１が１、およびｖ２．．．ｖｑ－１が２である。全てのｋ（１＜＝ｋ＜＝ｑ－１）に対して、長さｋのｖ０からｖｑ－１までの経

路がある。
・ｋ＝１：ｖ０－＞ｖｑ－１
・ｋ＝２：ｖ０－＞ｖｑ－２－＞ｖｑ－１
・ｋ＝３：ｖ０－＞ｖｑ－３－＞ｖｑ－２－＞ｖｑ－１
・．．．．．
・ｋ＝ｑ－１：ｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｑ－２－＞ｖｑ－１
【００８２】
同様に、グラフの用語では、マージパターンは、ｐ個のバーテックスをもつ経路として説明できる。元の経路を含むエッジに加えて、終了バーテックス、ｖｐ－１が他の全てのバーテックスによって駆動される。ｖｐ－１の入次数がｑ－１、ｖ０が０、および他のバーテックスｖ１．．．ｖｑ－２が１である。ｖｐ－１の出次数が０、ｖｐ－２が１、およびｖ０．．．ｖｑ－３が２である。全てのｋ（１＜＝ｋ＜＝ｐ－１）に対して、長さｋのｖ０からｖｐ－１までの経路がある。
・ｋ＝１：ｖ０－＞ｖｐ－１
・ｋ＝２：ｖ０－＞ｖ１－＞ｖｐ－１
・ｋ＝３：ｖ０－＞ｖ１－＞ｖ２－＞ｖｐ－１
・．．．．．
・ｋ＝ｑ－１：ｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｐ－２－＞ｖｐ－１
【００８３】
図１１Ａは、前述した問題２に対する第１の解決策を示す。構築方法は以下の通り。長さｑ－１の散乱パターンのコピーされたｐ－１を連結し、次いで、最後にもう１つバーテックス、ｖ（ｐ－１）（ｑ－１）＋１を追加する。散乱パターンの終了バーテックスおよび開始バーテックスは、連結中に、接合バーテックス（ｊｕｎｃｔｉｏｎ　ｖｅｒｔｅｘ）として示される、１つのバーテックスにマージされる。全ての接合バーテックスからのエッジを最後のバーテックス、ｖ（ｐ－１）（ｑ－１）＋１に接続する。実際には、接合バーテックスをｖ（ｐ－１）（ｑ－１）＋１に接続することは、マージパターンを接合バーテックスに適用することである。
【００８４】
ここで、バーテックス、ｖｋの出次数は、

ｑ	ｋ＝ｘ（ｑ－１），０＜＝ｘ＜＝ｐ－２
０	ｋ＝（ｐ－１）（ｑ－１）＋１
１	その他

である。
【００８５】
バーテックス、ｖｋの入次数は、

０	ｋ＝０
１	ｋ＝ｘ（ｑ－１）＋１，０＜＝ｘ＜＝ｐ－２
ｐ	ｋ＝（ｐ－１）（ｑ－１）＋１
２	その他

である。
【００８６】
全てのｋ（１＜＝ｋ＜＝ｑ）に対して、長さｋでｖ０からｖ（ｐ－１）（ｑ－１）＋１までの経路は、
・ｋ＝１：ｖ０－＞ｖ（ｐ－１）（ｑ－１）＋１
・ｋ＝２：ｖ０－＞ｖｑ－１－＞ｖ（ｐ－１）（ｑ－１）＋１
・ｋ＝３：ｖ０－＞ｖｑ－２－＞ｖｑ－１－＞ｖ（ｐ－１）（ｑ－１）＋１
・．．．．．
・ｋ＝ｑ：ｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｑ－１－＞ｖ（ｐ－１）（ｑ－１）＋１
【００８７】

全てのｋ（ｑ＋１＜＝ｋ＜＝２ｑ－１）に対して、長さｋでｖ０からｖ（ｐ－１）（ｑ－１）＋１までの経路は、
・ｋ＝ｑ＋１：ｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｑ－１－＞ｖ２ｑ－２－＞ｖ（ｐ－１）（ｑ－１）＋１
・ｋ＝ｑ＋２：ｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｑ－１－＞ｖ２ｑ－１－＞ｖ２ｑ－２－＞ｖ（ｐ－１）（ｑ－１）＋１
・ｋ＝ｑ＋３：ｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｑ－１－＞ｖ２ｑ－２－＞ｖｑ－１－＞ｖ２ｑ－２－＞ｖ（ｐ－１）（ｑ－１）＋１
・．．．．．
・ｋ＝２ｑ：ｖ０－＞ｖ１－＞ｖ２－＞．．．－＞ｖｑ－１－＞ｖｑ－＞ｖｑ＋１－＞．．．－＞ｖ２ｑ－２－＞ｖ（ｐ－１）（ｑ－１）＋１

【００８８】

その他のｋ（２ｑ＋１＜＝ｋ＜＝（ｐ－１）（ｑ－１）＋１）に対して、経路が同様に構築される。それ故、解決策１の達成された最大経路長は、（ｐ－１）（ｑ－１）＋１に等しい。

【００８９】

図１１Ｂは、前述した第１の解決策と似ている、問題２に対する第２の解決策を示す。この解決策では、マージパターンおよび散乱パターンの使用が、問題２に対する第１の解決策に関してスワップされる。基本的には、マージパターンが副経路に対して使用され、散乱パターンが接合バーテックスに対して適用される。入次数、出次数、および様々な長さの経路に関する詳細は、解決策１と同様である。解決策１の達成された最大経路長は、（ｐ－１）（ｑ－１）＋１に等しい。

【００９０】

図１２は、問題２に対する第３の解決策を示す。この解決策は、マージパターンを副経路および接合バーテックスの両方に適用することである。解決策１の達成された最大経路長は、ｐ（ｐ－１）＋１に等しい。

【００９１】

図１３は、問題２に対する第４の解決策を示す。この解決策は、散乱パターンを副経路および接合バーテックスの両方に適用することである。解決策１の達成された最大経路長は、ｑ（ｑ－１）＋１に等しい。

【００９２】

図１４は、問題１に対する解決策を示す。前述した４つの解決策から問題１の解決策を得ることは、簡単である。最大長がｎ（経路長）以上の任意の解決策を取ればよい。解決策１が取られ、ｎが（ｐ－２）（ｑ－１）＋１に等しいと仮定する。（ｐ－２）（ｑ－１）＋１より大きい指標をもつバーテックスを取り除き、次いで、接合バーテックスをｖ（ｐ－２）（ｑ－１）＋１に再接続する。結果として得られるグラフが、問題１に対する解決策である。同じ技法が問題２に対する４つの解決策すべてに当てはまる。

【００９３】

本明細書で説明する方法例は、少なくとも一部は、機械またはコンピュータ実装できる。いくつかの例は、上の例で説明した方法を実行する電子装置を構成するように動作可能な命令でコード化されたコンピュータ可読媒体または機械可読媒体を含み得る。かかる方法の実装は、マイクロコード、アセンブリ言語コード、高水準言語コード、または同様のものなどの、コードを含み得る。かかるコードは、様々な方法を実行するためのコンピュータ可読命令を含み得る。コードは、コンピュータプログラム製品の一部を形成し得る。さらに、コードは、実行中または他の時に、１つもしくは複数の揮発性または不揮発性のコンピュータ可読媒体上に有形的に格納され得る。これらのコンピュータ可読媒体は、ハードディスク、取り外し可能磁気ディスク、取り外し可能光ディスク（例えば、コンパクトディスクおよびデジタルビデオディスク）、磁気カセット、メモリカードまたはメモリスティック、ランダムアクセスメモリ（ＲＡＭ）、読取り専用メモリ（ＲＯＭ）、および同様のものを含み得るが、これらに限定されない。

【００９４】
図１５は、フォンノイマン型アーキテクチャを有するコンピュータ１５００の例をおおまかに示す。本開示の内容を読んで理解すると、当業者であれば、ソフトウェアプログラム内に定義された機能を実行するために、ソフトウェアプログラムが、コンピュータベースシステム内のコンピュータ可読媒体から起動できる方法を理解するであろう。当業者は、さらに、本明細書で開示する方法を実装および実行するように設計された１つまたは複数のソフトウェアプログラムを作成するために採用できる様々なプログラム言語を理解するであろう。プログラムは、Ｊａｖａ、Ｃ＋＋、または１つもしくは複数の他の言語などのオブジェクト指向言語を使用して、オブジェクト指向フォーマットで構造化できる。代替として、プログラムは、アセンブリ、Ｃ、などの手続き型言語を使用して、手続き指向フォーマットで構造化できる。ソフトウェアコンポーネントは、リモートプロシージャコールまたはその他を含む、アプリケーションプログラムインタフェースまたはプロセス間通信技術など、当業者に周知の任意の数の機構を使用して、通信できる。様々な実施形態の技術は、任意の特定のプログラム言語または環境に制限されない。
【００９５】
このようにして、他の実施形態が実現できる。例えば、コンピュータ、メモリシステム、磁気ディスクもしくは光ディスク、何らかの他の記憶装置、または任意のタイプの電子装置もしくはシステムなどの、製品は、その上に格納された命令１５２４（例えば、コンピュータプログラム命令）を有するメモリ（例えば、取り外し可能記憶媒体、ならびに電気導体、光導体、または電磁導体を含む任意のメモリ）などの、コンピュータ可読媒体１５２２に結合された１つまたは複数のプロセッサ１５０２を含み得、それは、１つまたは複数のプロセッサ１５０２で実行される場合に、上の方法に関して説明した動作のいずれかを実行する結果となる。
【００９６】
コンピュータ１５００は、いくつかのコンポーネントに直接、および／またはバス１５０８を使用して、結合されたプロセッサ１５０２を有するコンピュータシステムの形をとることができる。かかるコンポーネントには、メインメモリ１５０４、スタティックまたは不揮発性メモリ１５０６、および大容量記憶１５１６を含み得る。プロセッサ１５０２に結合された他のコンポーネントには、ビデオディスプレイなどの出力装置１５１０、キーボードなどの入力装置１５１２、およびマウスなどのカーソル制御装置１５１４を含み得る。プロセッサ１５０２および他のコンポーネントをネットワーク１５２６に結合するためのネットワークインタフェース装置１５２０は、バス１５０８にも結合できる。命令１５２４は、いくつかの周知の転送プロトコル（例えば、ＨＴＴＰ）のうちの任意の１つを利用するネットワークインタフェース装置１５２０を経由し、ネットワーク１５２６を通して、さらに転送または受信され得る。バス１５０８に結合されたこれら要素のいずれかは、実現される特定の実施形態に応じて、存在しないか、単独で存在し得るか、または複数で存在し得る。
【００９７】
一例では、１つまたは複数のプロセッサ１５０２、メモリ１５０４、１５０６、または記憶装置１５１６は各々、コンピュータ１５００に、実行時に、本明細書に記載する方法のいずれか１つまたは複数を実行させ得る、命令１５２４を含むことができる。代替実施形態では、コンピュータ１５００は、スタンドアロン装置として動作するか、または他の装置に接続（例えば、ネットワーク接続）できる。ネットワーク化された環境では、コンピュータ１５００は、サーバークライアントネットワーク環境内のサーバーもしくはクライアント装置の資格で、または、ピアツーピア（もしくは分散）ネットワーク環境内のピア装置として、動作できる。コンピュータ１５００には、パーソナルコンピュータ（ＰＣ）、タブレットＰＣ、セットトップボックス（ＳＴＢ）、携帯情報端末（ＰＤＡ）、携帯電話、ウェブアプライアンス、ネットワークルーター、スイッチもしくはブリッジ、またはその装置によってとられる動作を指定する命令のセットを（連続して、もしくは他の方法で）実行可能な任意の装置を含み得る。さらに、単一のコンピュータ１５００のみが示

されているが、「コンピュータ」という用語は、本明細書で説明する方法のいずれか１つまたは複数を実行するための命令のセット（または複数のセット）を単独に、または一緒に実行する装置の任意の集合を含むようにも解釈されるものとする。

【００９８】

コンピュータ１５００は、１つまたは複数の通信プロトコル（例えば、ユニバーサルシリアルバス（ＵＳＢ）、ＩＥＥＥ　１３９４など）を使用して周辺機器と通信するための出力コントローラ１５２８も含むことができる。出力コントローラ１５２８は、例えば、コンピュータ１５００に通信的に結合されているプログラミング装置１５３０にイメージを提供できる。プログラミング装置１５３０は、並列マシン（例えば、並列マシン５００、ＦＳＭエンジン６００）をプログラムするように構成できる。他の例では、プログラミング装置１５３０は、コンピュータ１５００と統合されて、バス１５０８に結合できるか、またはネットワークインタフェース装置１５２０または別の装置を経由してコンピュータ１５００と通信できる。

【００９９】

コンピュータ可読媒体１５２４は単一の媒体として示されているが、「コンピュータ可読媒体」という用語は、命令１５２４の１つまたは複数のセットを格納する単一の媒体または複数の媒体（例えば、集中型もしくは分散型データベース、または関連するキャッシュおよびサーバー、ならびに／またはプロセッサ１５０２レジスタ、メモリ１５０４、１５０６、および記憶装置１５１６などの様々な記憶媒体）を含むように解釈されるべきである。「コンピュータ可読媒体」という用語は、コンピュータによって実行するための命令のセットを格納、コード化、または運搬可能であり、かつ、コンピュータに本発明の方法のいずれか１つまたは複数を実行させるか、または命令のかかるセットによって利用されるか、またはそれに関連したデータ構造を格納、コード化、または運搬可能である、任意の媒体を含むように解釈されるものとする。「コンピュータ可読媒体」という用語は、その結果、ソリッドステートメモリ、光媒体、および磁気媒体などの、有形的媒体を含むように解釈されるものとするが、それらに限定されない。

【０１００】

要約は、読者が技術的開示の本質および要点を確認できるようにする要約を要求する米国特許規則（３７　Ｃ．Ｆ．Ｒ）１．７２（ｂ）項に準拠するために提供されている。それは、請求項の範囲または意味を制限または解釈するために使用されないという理解で提示されている。以下の請求項は、これによって詳細な説明に組み込まれ、各請求項が別個の実施形態として権利を主張する。

【０１０１】

〔実施形態例〕

例１は、量化子をオートマトンに変換することを含む、コンピュータ実装方法を含み、変換することは、オートマトンの入次数を制御するように量化子を展開すること；および、オートマトンを、対象装置に対応する機械コードに変換することを含む。

【０１０２】

例２は、命令を含むコンピュータ可読媒体を含み、それは、コンピュータによって実行されるときに、そのコンピュータに、量化子をオートマトンに変換することであって、変換することが、オートマトンの出次数を制御するように量化子を展開することを含む、量化子をオートマトンに変換すること；およびオートマトンを、対象装置に対応する機械コードに変換すること、を含む動作を実行させる。

【０１０３】

例３は、ソフトウェアがその上に格納されたメモリ；およびそのメモリに通信的に結合されたプロセッサを含むコンピュータを含む。ソフトウェアは、プロセッサによって実行されるとき、そのプロセッサに：量化子をオートマトンに変換することであって、変換することが、オートマトンの入次数または出次数を制御するように量化子を展開することを含む、量化子をオートマトンに変換すること；およびオートマトンを、対象装置に対応する機械コードに変換すること、をさせる。

【０１０４】
例４は、量化子をオートマトンに変換することであって、変換することが、オートマトンの入次数または出次数を制御するように量化子を展開することを含む、量化子をオートマトンに変換すること；およびオートマトンを、対象装置に対応する機械コードに変換することを行うように構成されたコンピュータを含むシステムを含む。
【０１０５】
例５は、量化子をオートマトンに変換することであって、変換することが、オートマトンの出次数を制御するように量化子を展開することを含む、量化子をオートマトンに変換すること；およびオートマトンを、対象装置に対応する機械コードに変換することを含む、コンピュータ実装方法を含む。
【０１０６】
例６は、命令を含むコンピュータ可読媒体を含み、それは、コンピュータによって実行されるときに、そのコンピュータに、量化子をオートマトンに変換することであって、変換することが、オートマトンの入次数を制御するように量化子を展開することを含む、量化子をオートマトンに変換すること；およびオートマトンを、対象装置に対応する機械コードに変換すること、を含む動作を実行させる。
【０１０７】
例７では、例１～６のいずれかの主題が、式を、量化子を含む、言語固有でない表現に解析することを、を随意に含み得る。
【０１０８】
例８では、例１～７のいずれかの主題が、式を、言語固有でない表現に解析することが、正規表現を構文木に解析することを含むこと、を随意に含み得る。
【０１０９】
例９では、例１～８のいずれかの主題が、変換することが、量化子に対する繰返し式が空値可能である場合、量化子を、ゼロ回一致し得る繰返し式の空値不能形式を有する量化子として展開することを含むこと、を随意に含み得る。
【０１１０】
例１０では、例１～９のいずれかの主題が、オートマトンを機械コードに変換することが、オートマトンを、並列マシンをプログラムするように構成されたイメージに変換することを含むこと、を随意に含み得る。
【０１１１】
例１１では、例１～１０のいずれかの主題が、イメージを公開すること、を随意に含み得る。
【０１１２】
例１２では、例１～１１のいずれかの主題が、オートマトンの入次数を制御するように量化子を展開することが、入次数制限に応じて量化子を展開することを含むこと、を随意に含み得る。
【０１１３】
例１３では、例１～１２のいずれかの主題が、入次数制限が、オートマトンの状態への遷移数に関する制限を含むこと、を随意に含み得る。
【０１１４】
例１４では、例１～１３のいずれかの主題が、オートマトンの入次数を制御するように量化子を展開することが、対象装置の入次数制約に基づいて展開することを含むこと、を随意に含み得る。
【０１１５】
例１５では、例１～１４のいずれかの主題が、量化子が正規表現であること、を随意に含み得る。
【０１１６】
例１６では、例１～１５のいずれかの主題が、オートマトンの入次数を制御するように量化子を展開することが、閾値に基づいてオートマトンの入次数を制御するように量化子

を展開することを含むこと、を随意に含み得る。
【０１１７】
例１７では、例１～１６のいずれかの主題が、閾値が並列マシンのハードウェア制限に基づくこと、を随意に含み得る。
【０１１８】
例１８では、例１～１７のいずれかの主題が、展開することが、オートマトンの各状態に対する出遷移を閾値数未満に制限することを含むこと、を随意に含み得る。
【０１１９】
例１９では、例１～１８のいずれかの主題が、展開することが、オートマトンの各状態に対する出遷移を最小限にすることを含むこと、を随意に含み得る。
【０１２０】
例２０では、例１～１９のいずれかの主題が、変換することが、量化子に対する繰返し式が空値可能である場合、量化子を、ゼロ回一致し得る繰返し式の空値不能形式を有する量化子として展開することを含むこと、を随意に含み得る。
【０１２１】
例２１では、例１～２０のいずれかの主題が、変換することが、量化子が単一のある個数のループと一致し得る場合、量化子を、連続的にリンクされた単一のある個数に等しい、いくつかの式を表す状態を有するオートマトンを形成するように展開することを含み、その式が、量化子に対する繰返し式に対応すること、を随意に含み得る。
【０１２２】
例２２では、例１～２１のいずれかの主題が、量化子が複数のある個数のループと一致し得る場合、複数のある個数のループが、第１の数のループ、ｎ１、および第２の数のループ、ｎ２を含み、量化子を展開することが、量化子を、ｎ１－１個のループと一致し得る第１の量化子および、１～ｎ２－ｎ１＋１個までのループと一致し得る第２の量化子に分割することを含むこと、を随意に含み得る。
【０１２３】
例２３では、例１～２２のいずれかの主題が、イメージを並列マシン上にロードすること、を随意に含み得る。
【０１２４】
例２４では、例１～２３のいずれかの主題が、イメージを、コンピュータ可読媒体上に格納すること、を随意に含み得る。
【０１２５】
例２５では、例１～２４のいずれかの主題が、オートマトン内で入遷移を出遷移に対してトレードオフすることにより、入次数および出次数を制御すること、を随意に含み得る。
【０１２６】
例２６では、例１～２５のいずれかの主題が、各状態に対する入遷移が、展開時に閾値数未満に制限されること、を随意に含み得る。
【０１２７】
例２７では、例１～２６のいずれかの主題が、入次数が出次数の割合に制限されること、を随意に含み得る。
【０１２８】
例２８では、例１～２７のいずれかの主題が、オートマトンの状態に対する入遷移の閾値に達するまで、入次数が出次数の割合に制限されること、を随意に含み得る。
【０１２９】
例２９では、例１～２８のいずれかの主題が、機械コードが、並列マシンをプログラムするように構成されたイメージを含むこと、および、コンピュータに通信的に結合され、かつイメージを並列マシン上にロードするように構成された装置をさらに含むこと、を随意に含み得る。
【０１３０】

例３０では、例１～２９のいずれかの主題が、オートマトンの入次数を制御するように量化子を展開することが、入次数制限に応じて量化子を展開することを含むこと、を随意に含み得る。

【０１３１】

例３１では、例１～３０のいずれかの主題が、入次数制限が、オートマトンの状態への遷移数に関する制限を含むこと、を随意に含み得る。

【０１３２】

例３２では、例１～３１のいずれかの主題が、オートマトンの入次数を制御するように量化子を展開することが、対象装置の入次数制約に基づいて展開することを含むこと、を随意に含み得る。

【０１３３】

例３３では、例１～３２のいずれかの主題が、オートマトンの入次数を制御するように量化子を展開することが、閾値に基づいてオートマトンの入次数を制御するように量化子を展開することを含むこと、を随意に含み得る。

【０１３４】

例３４では、例１～３３のいずれかの主題が、閾値が並列マシンのハードウェア制限に基づくこと、を随意に含み得る。

【０１３５】

例３５では、例１～３４のいずれかの主題が、正規表現を構文木に解析すること、を随意に含み得る。

【０１３６】

例３６では、例１～３５のいずれかの主題が、オートマトンを機械コードに変換することが、オートマトンを、並列マシンをプログラムするように構成されたイメージに変換することを含むこと、を随意に含み得る。

【０１３７】

例３７では、例１～３６のいずれかの主題が、イメージを公開すること、を随意に含み得る。

【０１３８】

例３８では、例１～３７のいずれかの主題が、オートマトンの出次数を制御するように量化子を展開することが、出次数制限に応じて量化子を展開することを含むこと、を随意に含み得る。

【０１３９】

例３９では、例１～３８のいずれかの主題が、出次数制限が、オートマトンの状態からの遷移数に関する制限を含むこと、を随意に含み得る。

【０１４０】

例４０では、例１～３９のいずれかの主題が、オートマトンの出次数を制御するように量化子を展開することが、対象装置の出次数制約に基づいて展開することを含むこと、を随意に含み得る。

【０１４１】

例４１では、例１～４０のいずれかの主題が、オートマトンの出次数を制御するように量化子を展開することが、閾値に基づいてオートマトンの出次数を制御するように量化子を展開することを含むこと、を随意に含み得る。

【０１４２】

例４２では、例１～４１のいずれかの主題が、閾値が並列マシンのハードウェア制限に基づくこと、を随意に含み得る。

【０１４３】

例４３では、例１～４２のいずれかの主題が、展開することが、オートマトンの各状態に対する入遷移を閾値数未満に制限することを含むこと、を随意に含み得る。

【０１４４】

例４４では、例１～４３のいずれかの主題が、展開することが、オートマトンの各状態

に対する入遷移を最小限にすることを含むこと、を随意に含み得る。

【０１４５】

例４５では、例１～４４のいずれかの主題が、変換することが、量化子に対する繰返し式が空値可能である場合、量化子を、ゼロ回一致し得る繰返し式の空値不能形式を有する量化子として、展開することを含むこと、を随意に含み得る。

【０１４６】

例４６では、例１～４５のいずれかの主題が、変換することが、量化子が単一のある個数のループと一致し得る場合、連続的にリンクされた単一のある個数に等しい、いくつかの式を表す状態を有するオートマトンを形成するように量化子を展開することを含み、その式が、量化子に対する繰返し式に対応すること、を随意に含み得る。

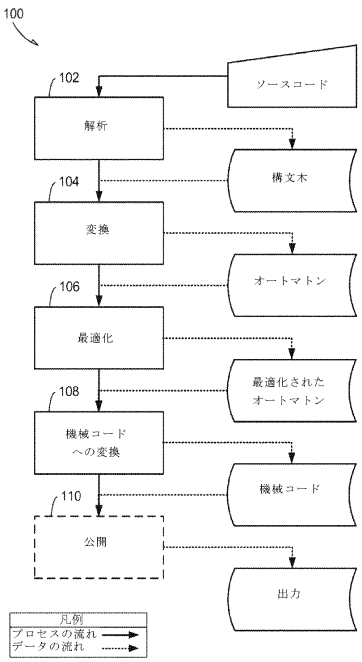
【０１４７】

例４７では、例１～４６のいずれかの主題が、量化子が複数のある個数のループと一致し得る場合、複数のある個数のループが、第１の数のループ、ｎ１、および第２の数のループ、ｎ２を含み、量化子を展開することが、量化子を、ｎ１－１個のループと一致し得る第１の量化子および、１～ｎ２－ｎ１＋１個までのループと一致し得る第２の量化子に分割することを含むこと、を随意に含み得る。

【０１４８】

例４８は、例１～４７のいずれかの主題を使用して生成されたイメージによってプログラム化された並列マシンを含む。

【図１】



【図２】

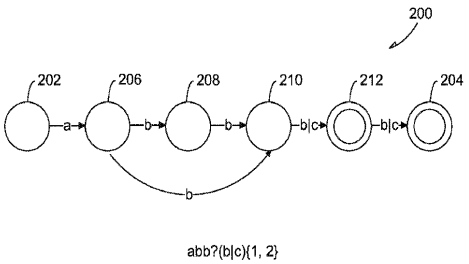
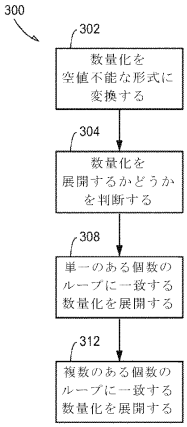


FIG. 2

【図３】



【図４Ａ】

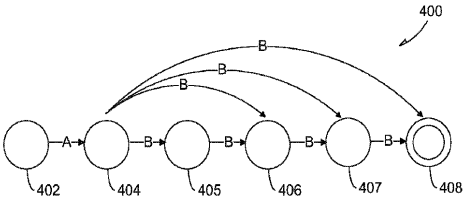


FIG. 4A

【図４Ｂ】

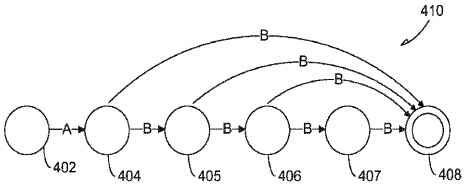


FIG. 4B

【図４Ｃ】

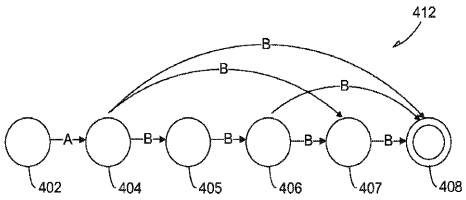
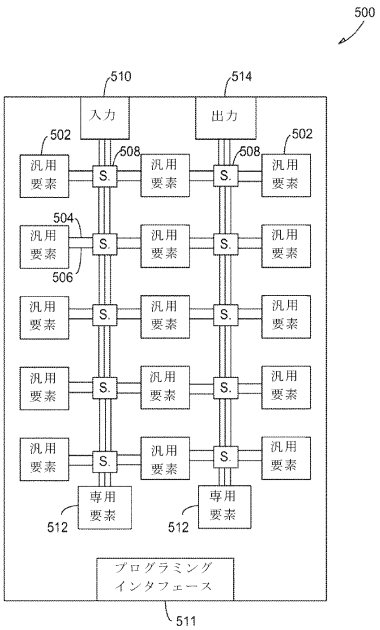
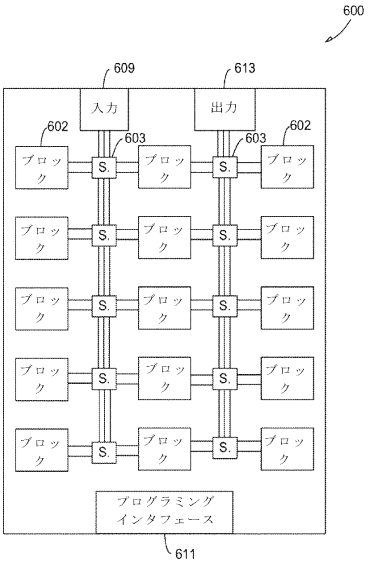


FIG. 4C

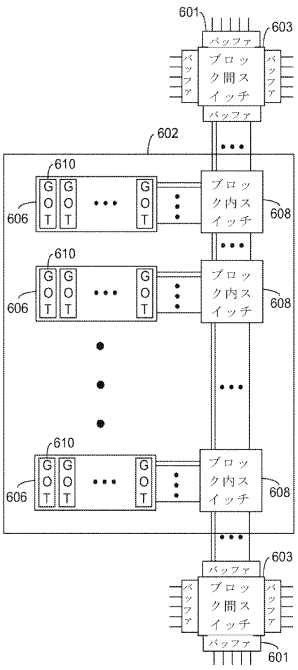
【図５】



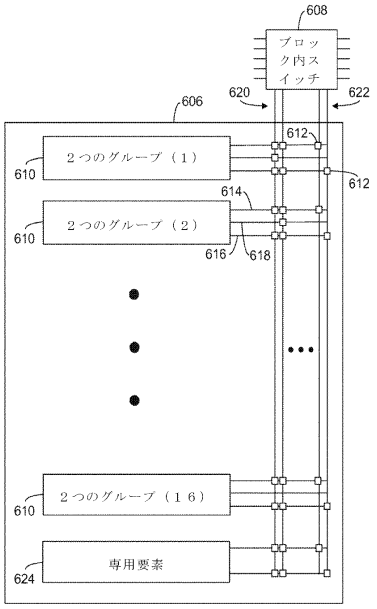
【図６】


600
609
入力
613
出力
602
603
ブロック
S.
プログラミング
インタフェース
611

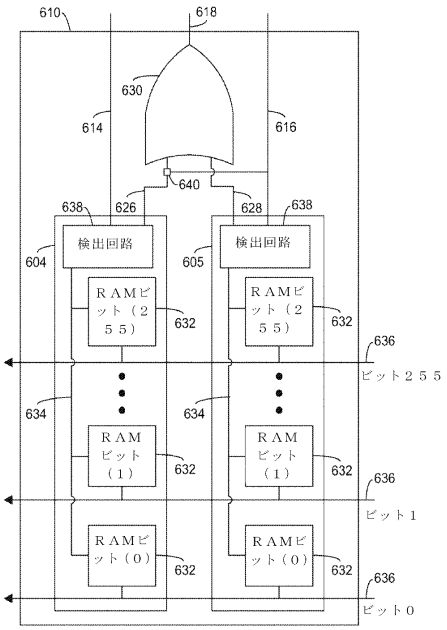
【図７】


601
603
バッファ
ブロック間スイッチ
602
610
606
GOT
ブロック内スイッチ
608

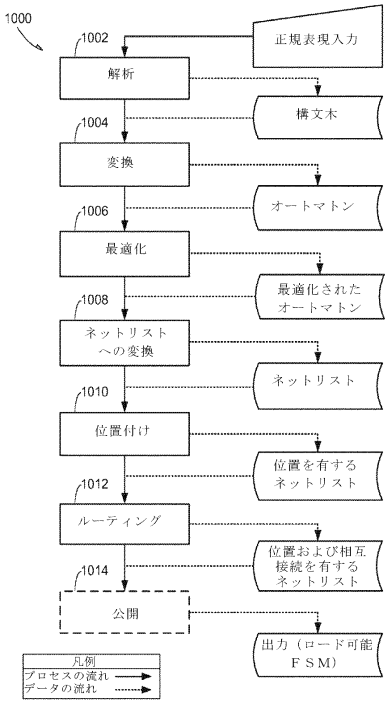
【図８】


608
ブロック内スイッチ
620
622
606
612
610
２つのグループ（１）
614
２つのグループ（２）
616
618
２つのグループ（１６）
624
専用要素

【図９】


610
618
630
614
616
640
638
626
628
検出回路
604
605
RAMビット（２５５）
632
636
ビット２５５
634
RAMビット（１）
ビット１
RAMビット（０）
ビット０

【図１０】



【図１１Ａ】

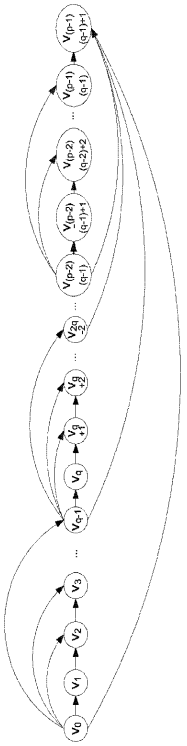


FIG. 11A

【図１１Ｂ】

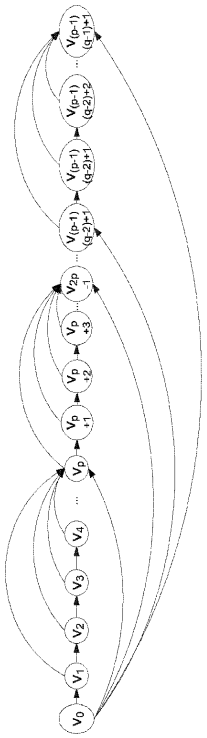


FIG. 11B

【図１２】

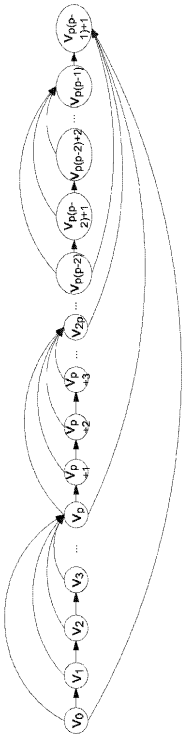


FIG. 12

【図１３】

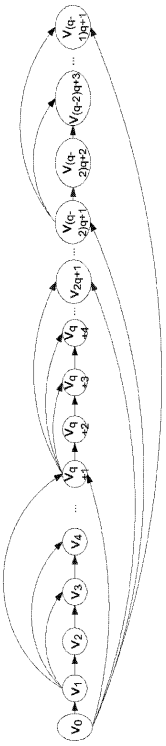


FIG. 13

【図１４】

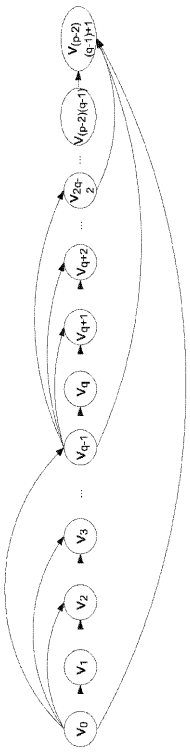
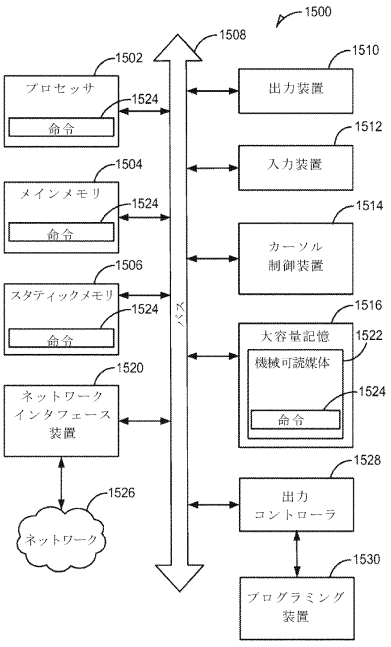


FIG. 14

【図１５】



フロントページの続き

(72)発明者　グレンデニング，ポール
アメリカ合衆国，カリフォルニア州　９４０６２，ウッドサイド，ブレイクウッド　ウェイ　３２３

審査官　坂庭　剛史

(56)参考文献　国際公開第２０１０／０１８７１０（WO，Ａ１）
米国特許出願公開第２０１０／０１８５６４７（ＵＳ，Ａ１）
特開２００９－０９３５９９（ＪＰ，Ａ）
特開２００５－３５３０６１（ＪＰ，Ａ）
新屋良磨、河野真治，動的なコード生成を用いた正規表現マッチャの実装，情報処理学会：シンポジウム＞プログラミング・シンポジウム＞冬＞５２回［ｏｎｌｉｎｅ］第５２回プログラミング・シンポジウム予稿集，一般社団法人情報処理学会，２０１１年　1月　7日，ｐ．１７３－１８０

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４５