

クスを右クリックすると、新しいウィンドウが開き、そのノードで入力するデータを設定・修正することができる。

図4に「血圧測定、性、年齢」ノードのデータ入力画面を示す。左端のチェックボックスをチェックすると、該当する欄が入力可能となる。最初の状態ではチェックされていないため、データが存在せず、不明であることを示している。これにより、該当するデータが存在するか不明であるのかを明示することができる。ここでは性別、年齢、収縮期血圧、拡張期血圧を入力することができるが、そのいずれかが不明であれば、左端のチェックボックスをチェックしない状態にしておけば良い。データ入力が完了したならば、上にある「Apply」ボタンを押してウィンドウを閉じる。

高血圧の診断では、まず2次性高血圧の可能性を検討する。知識記述には2次性高血圧として、腎血管性高血圧、原発性アルドステロン症、Cushing症候群、褐色細胞腫、糖尿病性腎症などの疾患ごとに、その疾患を疑わせる症状を記述している。これらの症状を疾患ごとにチェックするのは、対話的インターフェイスとしては煩雑であるため、すべての症状を合わせたノードを作り、「高血圧原因疾患の症状」とした。データの有無に関しては、2次性高血圧の検討を行ったのであれば全ての症状に目を通したと考えられるため、全体に対してデータの有無を記録するのみとした。この様子を図5に示す。

「高血圧原因疾患の症状」において、2次性高血圧を疑わせる症状のいずれかがチェックされていると、Applyボタンを押してウィンドウが閉じた直後に、次のノードである「高血圧原因疾患の確認」の内部で、その症状に対応する疾患が自動的にセットされる。また、このノ

ードで高血圧の原因となる疾患が存在すれば2次性高血圧、存在しなければ本態性高血圧に分岐するが、すでに上記の症状のいずれかがチェックされていることから2次性高血圧の存在が疑われるため、2次性高血圧側のノードが選択可能のまま残り、本態性高血圧以下のノードは暗くなっている。これは現在の状態から考えて本態性高血圧以下の手順が選択される可能性はないことを示している。この状況を図6に示す。

2次性高血圧を疑わせる症状が存在するからと言って、単純に2次性高血圧であるとは言えない。これは人間が確認しなくてはならない判断である。それを確認するために「高血圧原因疾患の確認」ノードが存在する。これを右クリックしてウィンドウを開けると、直前のノードでセットされた症状から計算により自動的に、該当する疾患に対しチェックがされている。これを図7に示す。この疾患が実際に疑われるのであればチェックされたままに残し、そうでなければこれを削除する。この操作が人間による確認である。このように、診療ガイドラインの内容に従い、データの値に応じて手順の分岐を対話的に確認することができることを確認した。臨床の現場で実際に使用することも可能ではあるが、それとともに、このような症状があった場合にはどのような疾患が疑われるのか、といった事項を試してみることにも有用であることが確認された。

また、選択されないパスを消去するインターフェイスについても試作を行った。これを図8に示す。グラフ描画アルゴリズムでは、表示されるべきノードだけを整列するように再描画が行われる。そのため、必要のない部分が消去されたグラフは、それ以前のグラフよりも小さくなり、必要部分のみが表示されて簡潔であった。その反面、パスの一部が消去されてしまうと痕跡が全く残ら

ないため、どこにかつての分岐があったのかがわかりにくくなるという不都合も見出された。

C.1.3. リスク層別化：人間の判断を仰ぐ必要のない場合

本態性高血圧の場合には、血圧に基づいた脳心血管リスク層別化を行う。そのためには臓器障害（脳疾患や腎疾患）、心血管病（心疾患や血管疾患）、合併症（糖尿病、高脂血症、メタボリックシンドローム等）の有無に加え、危険因子として喫煙の有無や高齢であるか否か（男性なら60歳以上、女性なら65歳以上を高齢としている）、などの要因を使用する。収縮期血圧と拡張期血圧の値から得られた血圧分類と、ここで得られたデータから算出されたリスク層により、システムは脳心血管リスク層別化の値を計算する。この計算方法はルールとして知識に記述されている。これは必然のみからなるルールであり、しかも同値ルールであるため、人間の判断や確認を仰ぐ必要はない。従ってこのルールの詳細はインターフェイス画面には表示されず、結果のみが提示されることを確認した。

C.1.4. 慢性腎疾患の例：病名の階層関係

糖尿病や慢性腎疾患がある場合には、通常の高血圧治療とは別の治療に分岐する。そして糖尿病や慢性腎疾患の有無は、すでにリスク層別化の評価を行う際に取得しているため、ここでも自動的に関連するデータがセットすることができた。ただし、以前のデータ取得は腎疾患の有無を尋ねるものであり、その内容は蛋白尿・腎障害・腎不全であり、慢性腎疾患とは異なる。ここではこれらの腎疾患のいずれかが存在す

ば慢性腎疾患の可能性があると仮定することにより、データの自動設定を行った。

しかし、このような疾患名称あるいは疾患概念の階層関係を記述することは簡単ではない。さらにこの場合のように蛋白尿という用語は疾患名と言うよりむしろ検査結果であり、階層関係ではなく因果関係や解釈までもが含まれてしまうことになる。しかもこうした関係は、提示されるインターフェイスでは把握することができないため、ユーザーはデータの自動設定が何故行われたのが理解できず混乱を招く可能性があると思われた。

また、単に可能性があるという判断基準で記述すると、関連するデータが増大してしまい、確認画面で不要なチェックを消す手間が増えてしまう。本来は同じ様なチェックを重複して入力する手間を減らすことが目的であったことを考えると、この様なデータ自動設定はあまり適切ではないと思われた。

C.1.5. 介入結果の保存：生活指導や降圧薬処方

リスク層別化により、高リスク群では直ちに降圧薬治療が開始され、中等リスクや低リスクの場合には生活指導を行った後に3カ月もしくは1カ月後に再受診し、目標血圧値に達していなければ降圧薬治療を開始するが、ここで生活指導の内容をデータとして保存できるようにした。また降圧薬の処方においても、どのような薬剤を処方したのかをデータとして保存することができるようになった。

今回の開発範囲の中では、介入結果として保存したデータを活用する場面は存在しないが、糖尿病など他の診療ガイドラインにおける判断に必要とされる生活指導や処方薬剤などの介

入内容と関連付けることが可能となった。また、電子カルテやオーダーリングシステムとの連携において、介入とはオーダーと同等である。判断支援システムが直接オーダーを発行することはないが、診療システムで発行されたオーダーが判断支援システムに取得、活用されうることを意味するものであり、判断に必要なデータを収集するために有効な機能と考えられた。

C.1.6. 降圧薬処方推奨と禁忌：提示の工夫

「降圧薬処方」は、診療ガイドラインの手順においては単にひとつのノードに過ぎない。しかし合併症や既往症に応じて推奨される薬剤や禁忌薬剤などの記述は知識として重要であり、この知識をどのようにして手順グラフに反映させるかを検討した。こうした知識をルールとして記述し、グラフとして提示する際には許された薬剤のみを提示する、という単純な方法も考えられたが、それでは手順グラフを見ても理由が理解できないと考えられたため、少しでも処方に関する知識を提示できるように開発を行った。

降圧薬処方では、併発疾患や合併症に対して推奨薬剤や禁忌薬剤が定まる。そこで2次性高血圧を来す様々な原因疾患の症状をまとめて提示したのと同じように、ここでも「併発疾患の確認」というノードにおいて、薬剤に推奨や禁忌として関連することのある疾患をまとめて提示した。ここで何らかの疾患がチェックされると、その疾患に該当する禁忌薬剤があれば、以下の適応薬剤ノードの中で、その薬剤は非選択となって暗く表示されることを確認した。

しかし推奨に関しても同様な選択/非選択を行ってしまうと、推奨でない薬剤が一切選択できなくなってしまう。そのため、推奨薬剤に関する

インタ-フェイスは実装しなかった。ただし、推奨薬剤をチェックボックスに表示させるにとどめるのであれば、提示することも可能と考えられた。

C.2. 診療ブロック

高血圧診療ガイドラインの手順内容は、診療ブロックとして診断、治療、経過観察の3つに分割された。診療手順のグラフ表示においては、これらを区別せずに全体の手順を表示することも、また個別に表示することもできるようにした。高血圧診療の治療ブロックだけを表示させた場合を図9に示す。

ところがこのように個別に表示させた場合、治療ブロックのグラフでは最初に「臓器障害の有無」による分岐があるが、このブロックには各種の臓器障害を入力するノードが存在しない。この「臓器障害の有無」は診断ブロックで入力されたはずのデータを使用しているため、以前に診断ブロックを使って該当するデータが入力されていれば、それに従った条件分岐が表示された。しかし診断ブロックを経過せず、最初から治療ブロックの手順グラフを参照した場合には、臓器障害に関するデータ項目が不定であるため、この分岐はいずれも選択可能な状態が表示されてしまった。また、すでに診断ブロックで該当データが入力されていた場合には、表示された経路を変更することができなかった。

そのため、条件判断に使われるデータを入力するノードがその診療ブロックの中に存在しない場合には、自動的に必要なデータに対する入力ノードを作ってデータ入力を促す必要がある。もしくは、該当する診療ブロック以前の診療ブロックで設定されたデータで、かつ該当する診療ブロック内には入力箇所が存在せず、そのブロック内部で条件判断に使用されているデータ項目

に関しては、そのリストを作成しておく。そしてその診療ブロックの最初のノードに進む前に、これらいずれ必要となるデータ群を入力するノードを作ることが必要と考えられた。

なお、診療経過に従えば、時間的には診断、治療、経過観察の順序になるため、診断なしにいきなり治療の判断を必要とすることはない。しかし、診療ガイドラインの中身がどの様になっているのかを知るために、ガイドラインの手順グラフを試しに見る様な場合には、時間的な順序は全く保証されない。従って、ひとつの診療ブロック単独に対して手順グラフを正しく機能させるためには、全体の手順グラフを単純に診療ブロックごとに分割するのではなく、上記のようなデータ群入力に関するノードの増設が必要であることが明らかとなった。

ところで、経過観察の診療ブロックについては、色々問題があることがわかった。本来であれば経過観察には、生活指導を行った後の血圧値の評価も、降圧薬投与後の血圧値の評価も含まれる。ところが本システムでは、降圧薬治療を行う場面を治療ブロックとしたため、生活指導後の血圧再評価は診断ブロックに含まれてしまった。また降圧薬投与を開始した後に、経過観察で血圧を再評価した上で、再度降圧薬投与の増量または減量を考慮することになるが、この「降圧薬投与の変更」は、本システムでは治療ブロックに属してしまっており、従って経過観察ブロックなるものは単に降圧薬処方ノードから血圧再評価ノードへの戻りエッジのみ、もしくはこれら一覽のノード間のループを示すことになった。つまり「全体の手順を sequential に分割」することができなかつたことになる。しかし経過観察診療ブロックには「降圧薬治療開始」という前提条件と「血圧コントロールを実現する」という目的が存在し、診療上も意義がある

ブロックであるため、他の診療ブロックとは独立させた。このことは、診療ブロックとは必ずしも sequential に切り分けられるものではなく、入れ子になることも考慮しなくてはならないことを意味するものと考えられた。

C.3. 手順グラフの検索モード

手順グラフの最初のノードである「検索」を右クリックすると図10のウィンドウが表示される。ここで検索モードをONにすれば、以下の各項目のチェック内容に従って、一括してデータ投入が行われ、手順グラフの描画が行われることを確認した。例えばCa拮抗薬であるカルブロックをチェックして検索を行うと、Ca拮抗薬処方が行われうるパスが表示され、それ以外のパスは非選択として暗く表示されることを確認した。もし手順グラフが表示された後に、各ノードのデータを修正したり新たに入力したりすることも可能であれば、さらに探索的に手順グラフを試すことが可能であったはずであるが、残念ながらそれを行うためには極めて複雑なプログラムを書かなくてはならず、実現できなかった。検索ノードで設定できる項目は薬剤であり、疾患名や症状はもちろんのこと、最も重要と思われる血圧の値を入力することができない。これは多くの項目を入力することができるようにすると、非常に煩雑になってしまい、使いにくくなることが予想されたためである。

むしろ、ひとつの診療ガイドラインの内部に限らず、複数のガイドラインに亘った検索を行えることが重要であると考えられたため、高血圧と胃潰瘍の両方のガイドラインの手順内容を検索対象として含めることを優先した。

ここで表示されている手順グラフは高血圧か胃潰瘍のいずれかひとつであるため、与えられた

データを満たすパスも、すでに表示されているグラフの上でしか観察することができない。しかし入力されたデータはシステムに保存されているので、単に表示する手順グラフを切り替えて再描画させることにより、求めるパスが表示されることを確認した。

検索モードの主たる目的は、降圧薬処方における禁忌の検出であったが、上記のように疾患名をデータとして入力していないため、それは実現できなかった。しかしデータ項目を増やし、適切なプログラミングを行うことにより、禁忌検出は可能となるはずであり、その場合にはすべての手順グラフが非選択として暗く表示されるはずである。ただし、そのような提示方法が禁忌の検出に適切であるとは思えなかった。

D. 考察

本研究では診療ガイドラインの内容である電子的知識表現を、手順グラフという形式で表示することにより視覚的に解りやすく表現し、グラフの形態を対話的なデータ入力によってダイナミックに変化させることを可能にした。これにより、診療における判断支援に利用するだけではなく、仮想的に様々なデータを入力することによってどのような帰結が得られるのかを試すことにより、診療ガイドラインの内容をより深く理解することにも利用することができた。ここでは開発過程や結果を通じて得られた様々な問題点について考察を行う。

D.1. 手順グラフの提示形態について

診療ガイドラインに即して診療手順をフローチャートのようにグラフとして表示すること自体は、全体像が容易に把握でき、しかも個々の条件判断に応じた経路が視覚化されているため、理解しやすいものであると言える。

通常フローチャートでは、状態・作業・条件判断などのノードの形状を丸・四角・菱形などで表示している。ところが本システムでは、使用したグラフ描画ルーチンの性質から、個々のノードの形状を変更する機能を持っていなかったため、すべてのノードが四角で表示されてしまっている。しかし本システムは手順を示すものであるため、基本的に状態と分岐さえ見分けられれば良い。従って各ノードの形状自体は四角のみであっても、問題はないと考えられた。

しかしノードには内部にデータを持ちうるノードと、何もデータを持っていない状態表示のためだけのノードがある。またデータを持っているノード

ドについても、データが何もセットされていないのか、何らかのデータがセットされた状態であるのかは、ノードを開いてみないと解らない。この点は非常に不便を感じる部分であり、どのノードにデータ設定が可能であるのか、データ設定が可能なノードで何らかのデータがセットされているのかいないのか、が手順グラフを見ただけで解るような仕組みが望まれる。

また各ノードにはノードの名称が表示されているが、これは必ずしも簡潔に短い言葉で記述できるとは限らない。長い記述があるとノードの中に収まりきらず、読めなくなってしまう。従って、ノード名称は簡潔に済ませ、そのノードの説明が簡単な操作で出現するような仕組みが望まれる。

D.2. 対話的インターフェイス

診療ガイドラインに限らず、医療では数値によって正常や異常を判断することが多い。例えば「収縮期血圧が140以上160未満であれば軽症高血圧 (JSH2009ではI度高血圧) である」の140や160という値である。しかしながら注意すべきは、この閾値が元来、統計的に導出されたものであることである。その一方、個々の症例で得られた測定値は、その患者が有する真の平均値の周りに、統計的な揺らぎを伴って生じた結果のひとつに過ぎない。従って、この両者は単純に同列に扱って良いものではなく、厳密に言うならば比較することすらできないのである。比較するためには、閾値に対しては平均値が得られた状況を理解し、測定値に対してはランダムな揺らぎを伴ったものである上に何らかの要因が系統的な誤差として影響していないかを考慮しなくてはならないのである。実際にはこうした要因を全て考慮することが難しいのは確かだが、何も考えずに単純に閾値を適用

して判断して良いというものでは決していない。このことはしばしば忘れ去れており、定型的な医療、チャート式医療といった問題が生じる要因のひとつになっていると推測される。そして診療ガイドラインに対する批判の原因のひとつにもなっていると推測される。診療ガイドラインの手順は、しかもそれをフローチャート形式に提示したものは、まさに「チャート式」だからである。しかし診療ガイドラインの主たる目的は、進んだ医療を平均に戻すことではなく、間違っただけあるいは遅れた医療を平均にまで持ち上げることである。つまりその分野が専門ではない医療従事者にこそ有用なものなのである。従って詳細な内容を理解できていなくとも、解りやすいチャート式で手順を追うことによって、最良ではなくとも誤っていない医療を提供できることが重要であると考えられる。

そして単に手順を追うことから次の段階に進み、閾値を正しく比較できるようになるためには、やはり診療ガイドラインの内容を理解する必要がある。それには本研究で開発した対話的インターフェイスが助けになる。この対話的インターフェイスでは、データに様々な値を仮に設定することによって、条件判断で取り得る他の経路を色々と試すことができるからである。特に閾値を単純に適用した際に選択される分岐だけではなく、それ以外の分岐が選択された時にどのような経路を辿るのかを、リアルタイムに確認することができる点は重要と考えられる。

D.3. 関連性の範囲の問題、優先順位の問題、矛盾の問題

本システムは、あるノードでデータを設定すると、関連する別のデータが自動的に設定される機能を備えている。これは同一データであれば

全く問題ない、有用な機能であり、無駄な重複入力を削減する。そして同一ではなくとも、依存関係がある場合にも、無駄な入力を避けることができ有用であるように思える。しかしユーザーは、システムがどのような依存関係を根拠としてデータを自動的に設定しているのかを見ることができない。本システムでは依存関係には標準的と思われる医学知識が使われている。例えば「狭心症は虚血性心疾患である」という知識を使い、あるノードで狭心症が設定されれば、別のノードで虚血性心疾患が自動的に設定される。これはほぼ自明な例と思われるが、あるユーザーには自明であっても、もしかすると他のユーザーには不可解であるかもしれない。従って、何らかの理由を(求められたならば)提示する、説明する機能が必要であると考えられる。依存データの自動設定には、別の問題がある。上記の例で手順の上流で「狭心症」というデータ項目がチェックされ、自動的に下流のノードにある「虚血性心疾患」データが設定されたとする。ユーザーが順にノードを下っていき、「虚血性心疾患」データが設定されたノードに到達した場合、もしここでユーザーがこの「虚血性心疾患」の設定を削除した場合を考えてみる。虚血性心疾患ではないが、狭心症である、は明らかに矛盾である。では「狭心症」データの設定を自動的に削除すれば良いのであろうか。しかし狭心症は虚血性心疾患の一部であり、狭心症の他にも心筋梗塞など、他のデータ項目があるかもしれない。ではそれらも含めて全ての虚血性心疾患の下位階層に属する疾患名データの設定を削除すれば良いのだろうか。では家族歴の「心筋梗塞」が削除されないようにするにはどうしたら良いのか。

このように、一方向の依存関係が実現できても、逆方向は容易には実現できない。依存関係だ

けではなく、階層関係などのような明確な関係を記述しなくてはならないと同時に、データの自動設定を行う際の影響範囲を明確にしないことはならないことがわかる。前者については、現状では診療ガイドラインの手順記述を行うために、医学用語の階層関係までも記述しなくてはならないことを意味している。医学における基礎的なリソースとして医学用語の階層関係が別に記述され配布されていれば、それを利用することによって自動的に解決する問題であると言える。後者については、患者自身のデータと家族のデータを別の名称にすれば解決するが、一般的にはデータ依存の影響範囲を明確に規定しなくてはならないと言える。

D.4. 診療ブロック

D.4.1 前提条件の表現

全体の診療手順を幾つかに分割した診療ブロックにおいて、個々の診療ブロックを独立に使用する際には、上流の診療ブロックで設定されたデータ群に依存した条件判断が存在することに注意する必要がある。繰り返すと、このデータ群とは、この診療ブロック以前の診療ブロックで設定されたデータであり、かつこの診療ブロック内には入力箇所が存在せず、このブロック内部で条件判断に使用されているデータ項目の集合である。

このデータ群、データの名称だけではなく値の制約も存在する。制約を持たないデータもあるかもしれないが、制約を持つ場合は、そのデータの値が、この診療ブロックに到達する条件を満たしていなくてはならないのである。

ここで、この診療ブロックの前提条件について考えてみる。診療ブロックには前提条件と終了

条件(または目的)が存在し、いずれもテキストで記載されたものである。しかし前提条件を、その診療ブロックの入口に到達するための条件であると考えれば、それはこの診療ガイドライン全体の最上位であるスタートノードから始めて、この診療ブロックの最初のノードに至るパスの経路途中に存在する全ての分岐条件において、このパスを通りうるデータとそれらの値である、とすることができる。つまりテキストで記述された前提条件を、このようなデータ群とそれらの値として記述することが可能となる。そうすると、先ほどから問題になっている「この診療ブロックでデータ入力箇所を持たず、かつ条件判断に使用されるデータ群とその値」とは、この様にして(テキストではなく)データとその値の集合として与えられた前提条件に含まれると考えられる。すなわち、前提条件を「データ群とそれらの値」として記述しておくことができれば、この「診療ブロック内のデータ設定問題」は解決することになる。

同様に、この診療ブロックの終了条件も、テキストではなく「データ群とそれらの値」として記述することが可能である。しかしひとつの診療ブロックの目的あるいは意図なるものを、終了「条件」としてデータのみで記述したものと同値に扱えるものであろうか。確かに、この診療ブロックの最終ノードに到達する条件として、「データ群とそれらの値」を記述することは可能であるが、診療の意図に即して「成功」した結果、到達しえた最終ノードである、ということまでを記述できるとは限らない様に思える。また、この手順で記述された「成功」に至るパスの他にも、別の手順が存在するかもしれない、その場合には異なった終了条件が与えられる可能性がある。「成功」に至るために様々な手順があったとしても「意図」は変わらないはずであり、「デー

タ群とそれらの値」として与えられたものだけでは、意図を満たす条件あるいは解釈の一部にしかかなり得ないように思われる。

このように意図あるいは診療の目的というものは、手順あるいは分岐条件を記述するデータ群に比べて一段上の抽象的な概念であると考えることができる。そして前提条件についても、診療ブロックの入口にいる「状態」と考えると、意図と同じく抽象的な概念であると考えられる。これら抽象的なものを、手順や分岐条件などのデータ群と同列に考えることは、しばしば混乱を招き、注意を要すると考えられる。本システムにおいては、ある診療ブロックの入口に立つ状態や診療の意図というものを、一定の手順およびその上に成り立つ条件分岐を規定した上でならば、「データ群とそれらの値」として前提条件や終了条件として記述することができる、と言うに留めておくべきであろう。

D.4.2 経過観察ブロック

本システムでの経過観察ブロックには問題があり、診療ブロックとは必ずしも全体の診療手順の一部を切り出す様な形では定義することができないという結果が得られた。しかしノードだけではなく、時間的な変遷としてノードを通過するエッジまで含んで考えるならば、これを新たな診療ブロックとして捉えられる可能性がある。経過観察では、降圧薬処方ノードの後に血圧再評価を行って目標血圧以内であることを確認すること、が診療上の経過観察であり、ここには時間的な推移とともにノードを推移するループが存在する。従ってこのループを経過観察ブロックであると定義すれば良いと思われるからである。その際には、知識記述における「再診で測定された血圧値」は「前回の診察時に測定された

「血圧値」になる。本システムにおける知識記述では「前回」のような時間的な抽象的概念を許容していないため、初診時の血圧値と再診時の血圧値しか存在しないが、これを許すならば実装可能と考えられる。ただしその場合には、データの時間的な履歴を扱えるようにする必要があり、「再診時血圧値」という同一の名称に対して異なる日時の値を繰り返し利用できるようななくてはならない。

診療ブロックをこのように「ノードの変遷」として定義するならば、既存の sequential なノードの切り出しをも含むことになる。また診療上の様々な状態をよりきめ細かく記述することが可能となる。例えば糖尿病の診療における「生活指導で血糖コントロールが達せられずに薬剤治療でコントロールが達成された状態」などの複雑な状態、しかしながら臨床では別個のものとして捉えるべき状態、を定義することが可能となる。このように、単なるノードやその静的な連鎖を対象にするだけではなく、現実における時間的な推移までを含んだノードの変遷を診療ブロックとして捉えること、さらには診療ブロックとしてではなくひとつの状態として、もう一段階上のノードとして捉えることは、診療過程を記述する上で極めて有力な手法となる可能性があると考えられる。

D.5. 手順グラフの検索モード

本システムでは、検索モードの初期画面で薬剤のみに固定してデータを与えることにより、該当するノードとパスを検索した。多数のガイドラインに亘って出現する全ての項目に対してデータ入力を行うことは現実的ではないと思われたためである。しかし、上記の診療ブロックに関する考察をもとに、前提条件を「データ群とそ

れらの値」として記述したならば、診療ブロック内の任意のノードに至る条件も同様に「データ群とそれらの値」として記述することができる。これに注目すると、検索モードで与えるデータを、まず「症状名」、「疾患名」、「薬剤名」、「検査名」などの分類別に、階層的に用意し、その中のひとつだけにデータを入力させる。次にそのデータと値を満たすようなノードを検索し、今度はそれらの全ノードを満たしうるデータに対して、ダイナミックに2番目の検索項目を階層的に用意して入力させることが考えられる。ここで禁忌を検出するためには「データを満たすノード」とは別に「禁忌データを満たすノード」を積極的に検索する必要があり、そこには補集合の検索のような複雑な処理が必要となるが、一応可能ではあろう。このようにして、同様に3番目、4番目のデータ入力を行わせて検索対象を絞り込むことも考えられる。

ただし、このように設計した場合には、検索モードの結果として手順グラフを表示させることよりも、ダイナミックに検索データを提示すること、に意義があることになる。また、本システムにおける手順グラフは、論理的な帰結だけではなく、人間に確認を求めるときのノードが存在するため、検索の立場からすると冗長な構成になっている。すなわち手順グラフを表示することによる「検索」としての意義は薄く、診療ガイドラインを跨いでダイナミックにデータを検索して成立しうるノードや禁忌ノードを提示する、というインターフェイスの方が、発展性があると考えられる。そしてその際には、手順グラフを描画するために確認ノードなどを増設した知識記述ではなく、それらの基になっている診療ガイドラインの知識記述を対象とすべきと考えられる。

D.6. 共通の電子的知識表現から異なる応用層の抽出について

一般的に、対話的インターフェイスや動的なインターフェイスはコンピュータを利用する際の操作性向上に利するものであるが、これに対立するものがバッチ処理であり、定型的な処理をより少ない手数で大量かつ高速に実行する際に有用である。前年度に実施した禁忌検出のような処理は、このバッチ処理に該当するものであり、そこには禁忌であることの理由を詳しく理解するよりも、緊急性を重視する面があり、迅速に警告を発することが重要である。しかもその対象は特定の疾患に関するガイドライン内に留まらず、医療に関連する膨大な項目間の全てに亘って高速に検索することが望まれる。診療ガイドラインの応用としてはどちらも重要であると考えられるが、診療ガイドラインのテキストからこれら2つの異なるインターフェイスに対して別々に電子的知識表現を構築するアプローチは効率的ではない。ひとつの診療ガイドラインからひとつの電子的知識記述を作成した上で、応用層として対話的なインターフェイスも、バッチ的な禁忌検出に適する知識記述も、いずれをも抽出することが望まれるが、今後の課題としたい。

E. 結論

診療ガイドラインの手順を記述した電子的知識表現をもとにして、手順グラフという形式で表示することにより、ガイドラインに記された診療手順の内容を視覚的に解りやすく表現した。同時に対話的なデータ入力によってグラフの形態をダイナミックに変化させることを可能とした。これは診療における判断支援に利用するだけではなく、仮想的に様々なデータを入力することによってどのような帰結が得られるのかを試すことにより、診療ガイドラインの内容をより深く理解することにも利用することができた。最後にこのシステムの開発を通じて得られた知見について考察した。

F. 研究発表

1. 論文発表

2. 学会発表

- [1] 小野木雄三 廣瀬康行, 診療ガイドラインと診療スレッドの知識表現, 医療情報学, 28S: 1092-1097, 2008
- [2] 廣瀬康行 小野木雄三, 医療介入に関わる知識表現への診療スレッドの応用, 医療情報学, 28S: 1087-1091, 2008

G. 知的財産権の出願登録状況

1. 特許取得

なし

2. 実用新案登録

なし

3. その他

なし

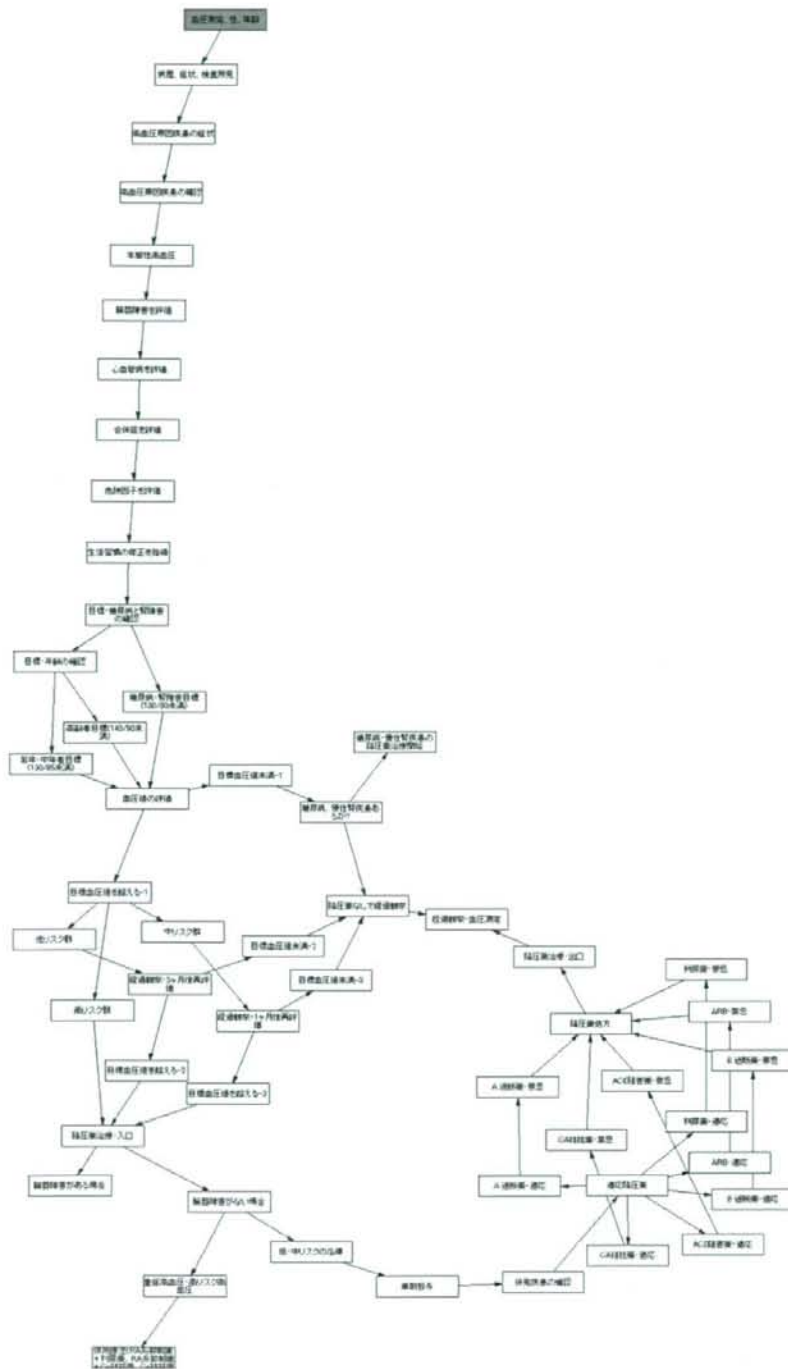


図 1a. Kamada-Kawai モデルによる描画 (高血圧診療ガイドラインから手順の一部を使用)

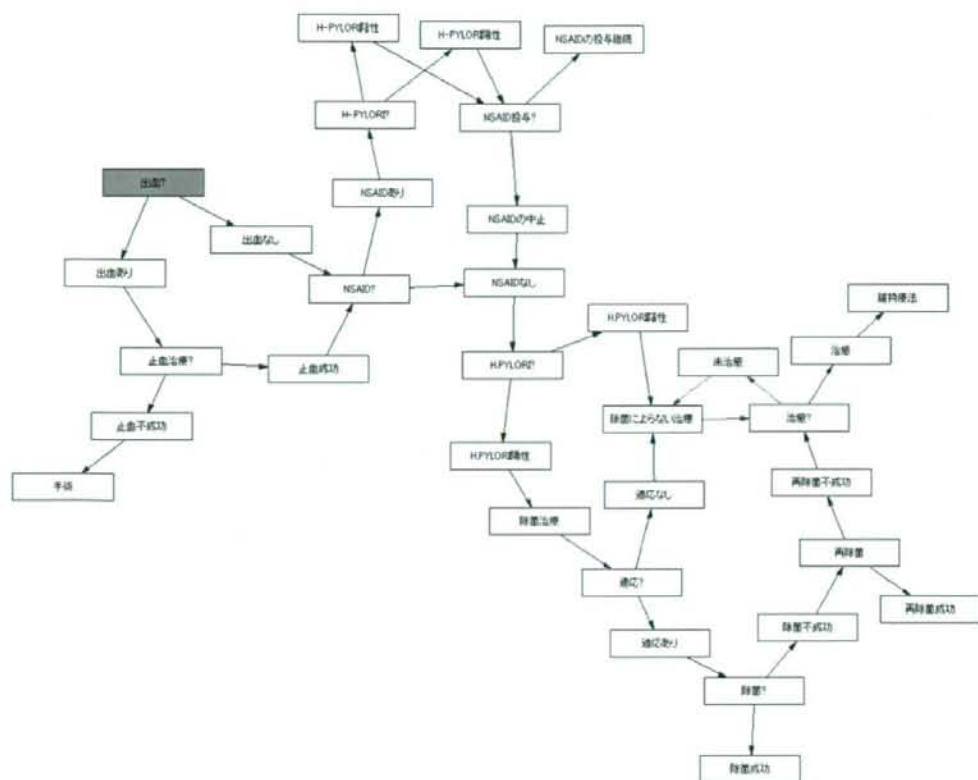


図 2a. Kamada-Kawai モデルによる描画 (胃潰瘍診療ガイドラインから手順の一部を使用)

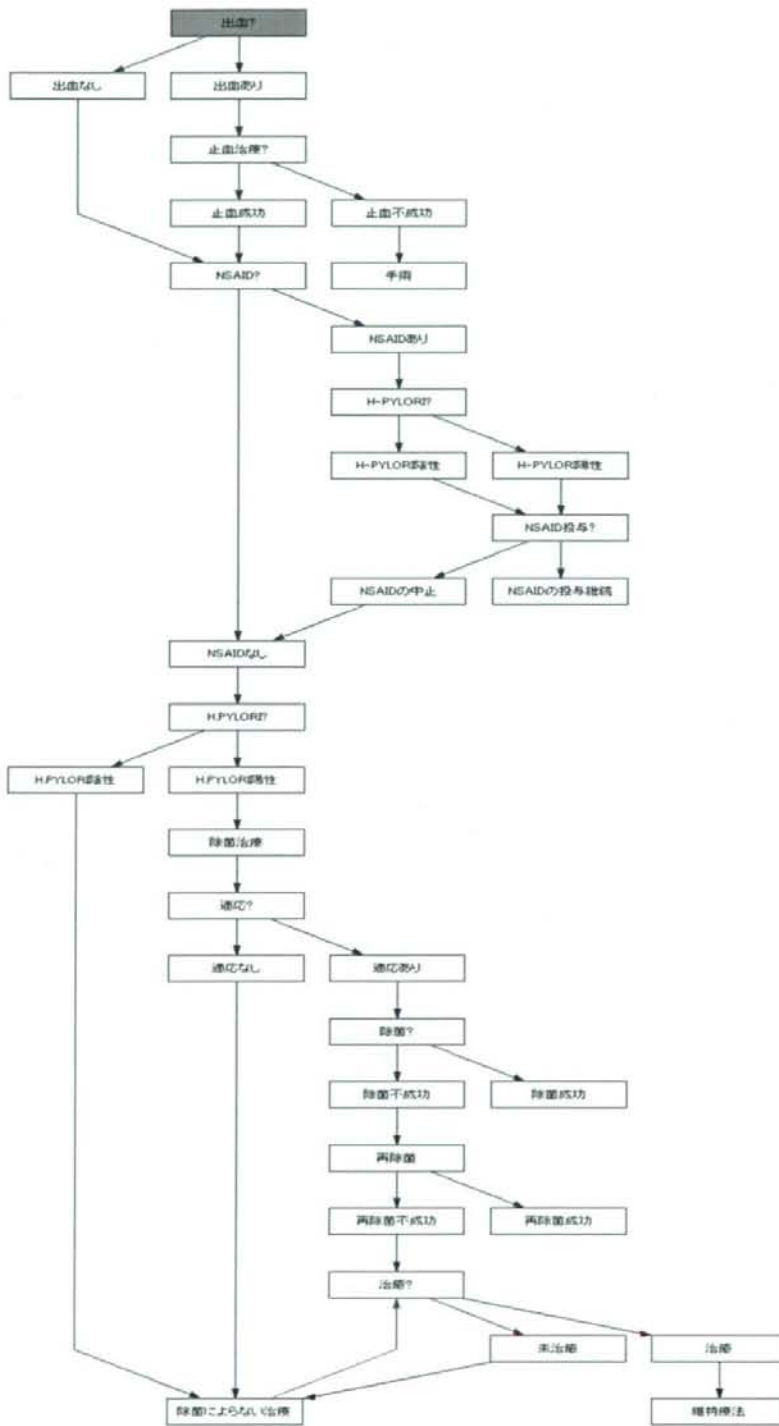


図 2b. Sugiyama モデルによる描画 (胃潰瘍診療ガイドラインから手順の一部を使用)

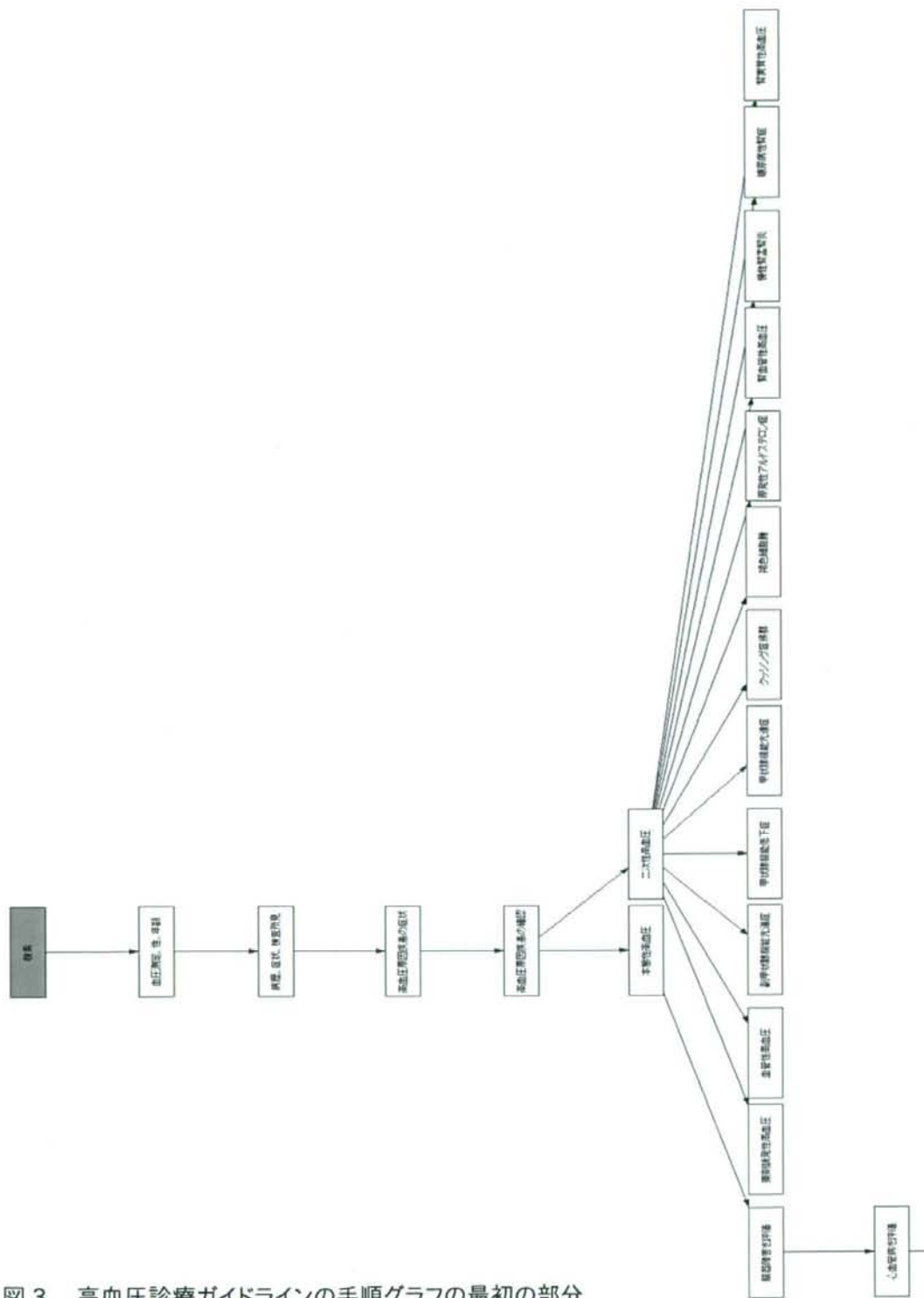


図 3. 高血圧診療ガイドラインの手順グラフの最初の部分

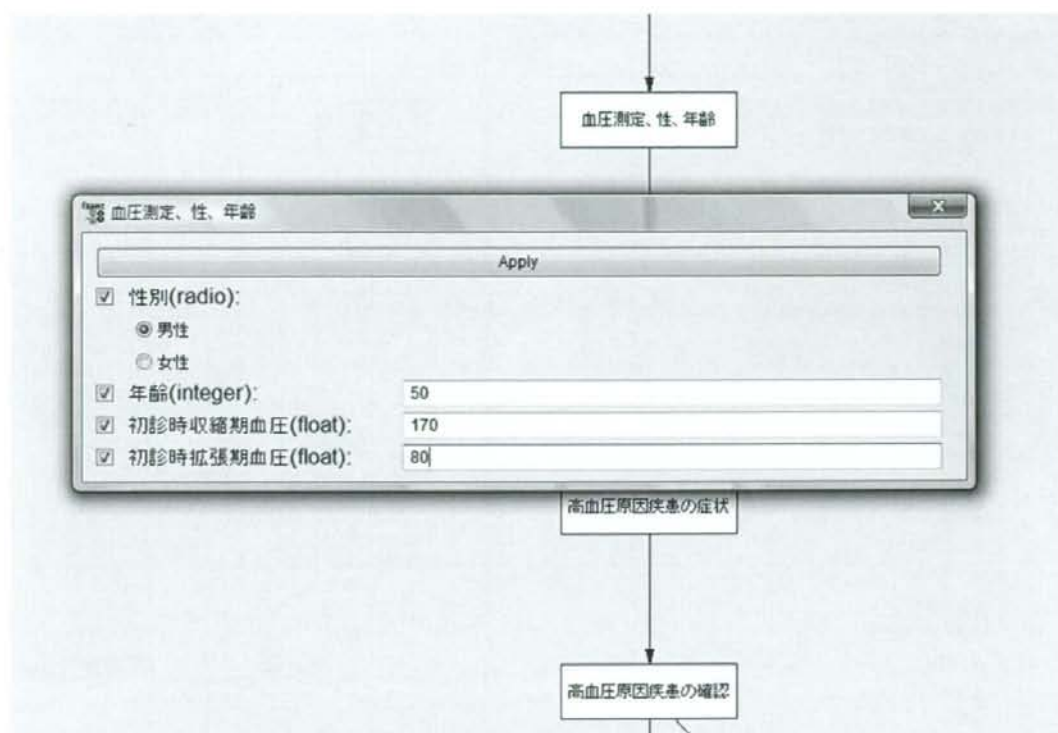


図 4. [血圧測定、性、年齢]ノードのデータ入力画面。
各項目左端のチェックボックスがチェックされていると、項目の値が存在することを示し、
チェックされていない場合は値が不明であることを示す。

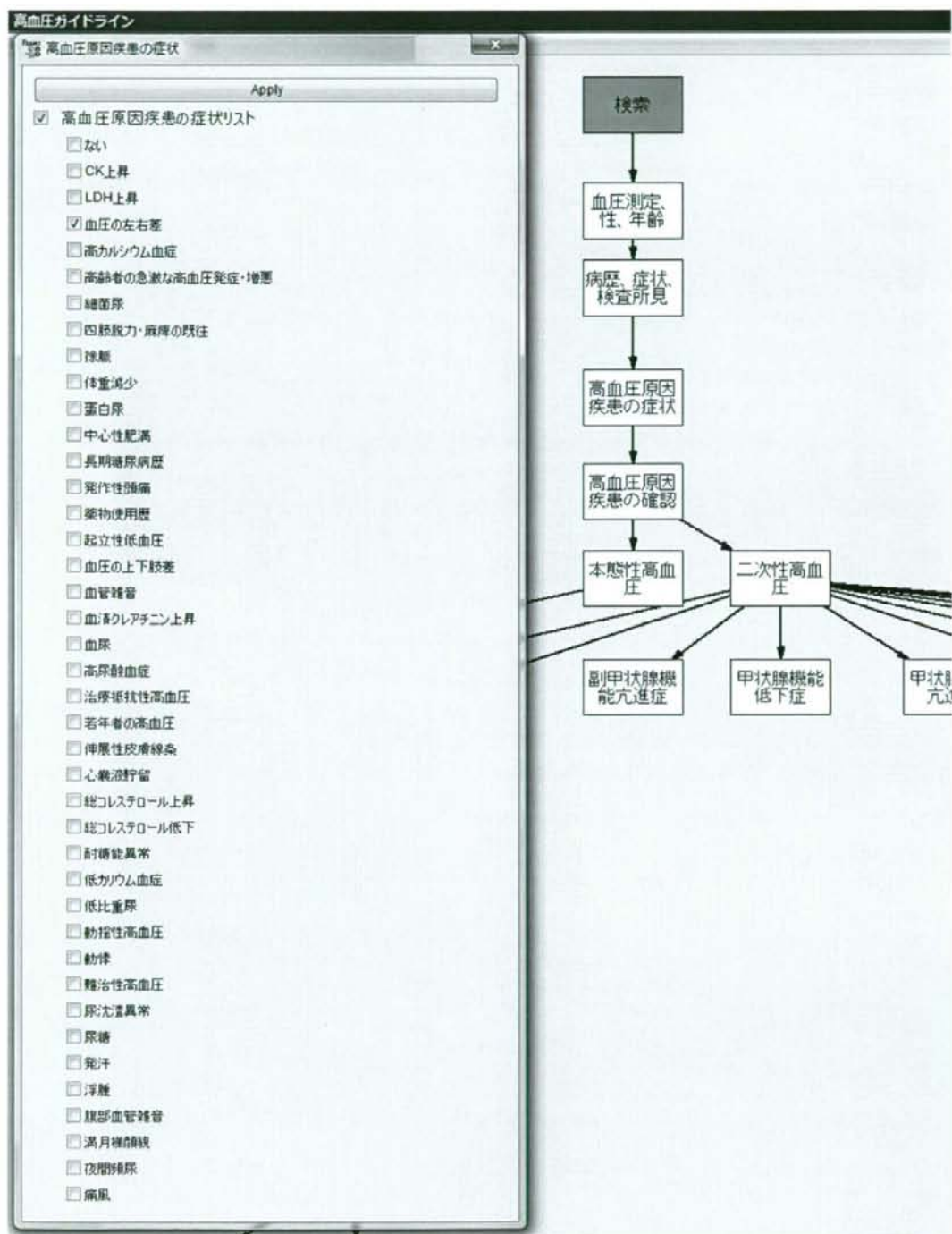


図 5. 2 次性高血圧を疑わせる症状の一括入力画面。これらの症状の取得は、同じ機会に行われると考えられるため、値が存在するか否かは 1 カ所のみとした。

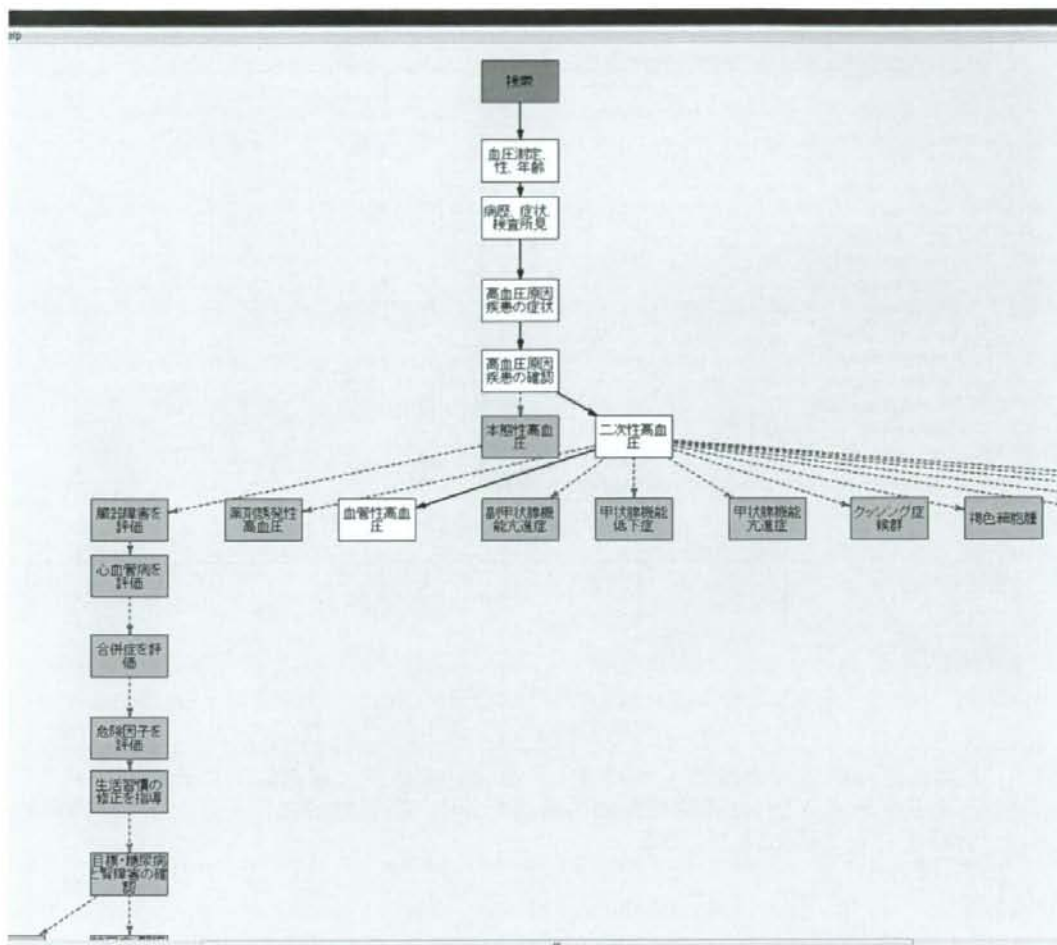


図 6. 高血圧原因疾患の症状ノードの 2 次性高血圧を疑わせる症状の中から、「血圧の左右差」を選択した結果、次の高血圧原因疾患の確認ノードの中にある「血管性高血圧」が自動的にチェックされ、手順グラフもこの図の様に書き換えられる。血管性高血圧は 2 次性高血圧なので、本態性高血圧であることはあり得ないとされ、暗くなって選択できない。また 2 次性高血圧の中でも他の原因疾患は選択できなくなっている。

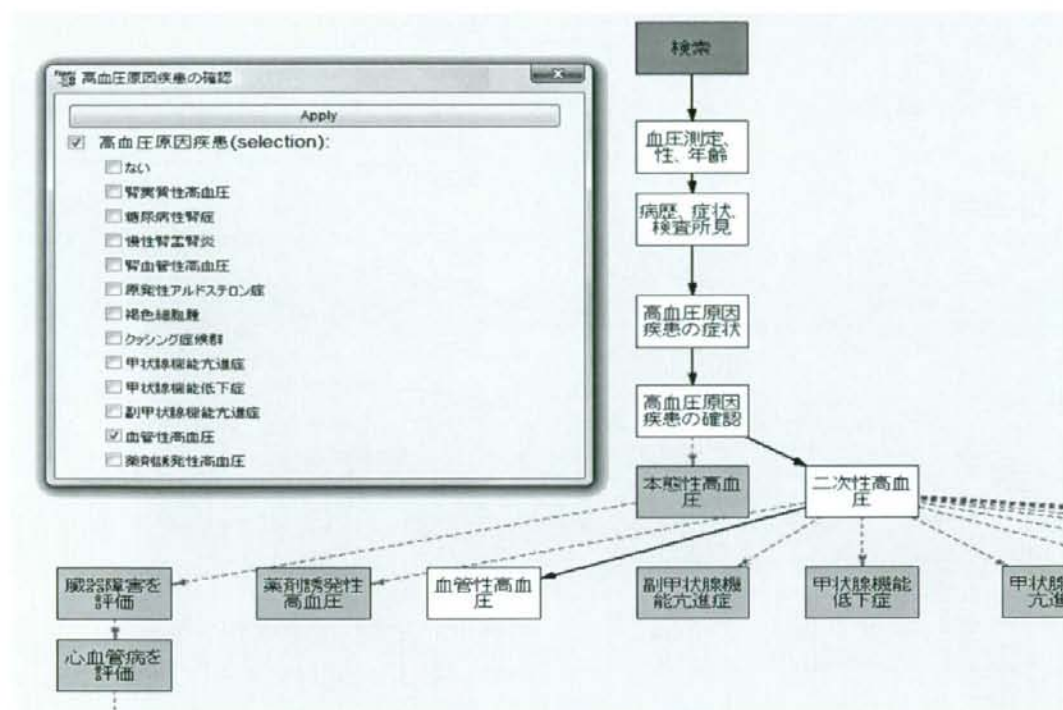


図 7. 「高血圧原因疾患の確認」ノードで、「血管性高血圧」が自動的にチェックされている様子を示す。これは確認のための画面であり、血管性高血圧を疑うか否かの最終判断を人間が行うことができる。