

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6692281号
(P6692281)

(45) 発行日 令和2年5月13日 (2020.5.13)

(24) 登録日 令和2年4月16日 (2020.4.16)

(51) Int.Cl.

F I

G 0 6 F 11/36 (2006.01)

G 0 6 F 11/36 1 8 4

請求項の数 13 (全 22 頁)

(21) 出願番号	特願2016-235350 (P2016-235350)	(73) 特許権者	000005108
(22) 出願日	平成28年12月2日 (2016.12.2)		株式会社日立製作所
(65) 公開番号	特開2018-92374 (P2018-92374A)		東京都千代田区丸の内一丁目6番6号
(43) 公開日	平成30年6月14日 (2018.6.14)	(74) 代理人	110000176
審査請求日	平成31年3月5日 (2019.3.5)		一色国際特許業務法人
		(72) 発明者	伊藤 信治
			東京都千代田区丸の内一丁目6番6号 株
			式会社日立製作所内
		(72) 発明者	佐藤 直人
			東京都千代田区丸の内一丁目6番6号 株
			式会社日立製作所内
		(72) 発明者	小川 秀人
			東京都千代田区丸の内一丁目6番6号 株
			式会社日立製作所内

最終頁に続く

(54) 【発明の名称】 テストケース生成装置、及びテストケース生成方法

(57) 【特許請求の範囲】

【請求項 1】

業務ルールの仕様を記述したルール仕様にに基づき生成されるプログラムのテストに用いるテストケースを生成するテストケース生成装置であって、

前記ルール仕様と、テストケースを生成する範囲に関する情報である範囲条件、及びテストケースの生成の方式を指定する情報である生成方式を含む、一つ以上のテストケース生成条件と、を記憶する情報記憶部と、

前記ルール仕様を表す論理式であるルール仕様論理式を生成するルール仕様論理式生成部と、

前記テストケース生成条件ごとに、テストケースを生成する範囲を表す論理式である範囲条件論理式を生成する、範囲条件論理式生成部と、

前記ルール仕様論理式と前記範囲条件論理式とを論理積によって結合した論理式であるテストケース生成用論理式を生成するテストケース生成用論理式生成部と、

前記テストケース生成用論理式に基づきテストケースを生成するテストケース生成部と、

を備え、

前記情報記憶部は、前記範囲条件として、前記ルール仕様の変更前と変更後の差分を示す情報が設定された前記テストケース生成条件を記憶し、

前記範囲条件論理式生成部は、前記ルール仕様の変更前と変更後の差分に基づき前記範囲条件論理式を生成する、

10

20

テストケース生成装置。

【請求項 2】

請求項 1 に記載のテストケース生成装置であって、

前記ルール仕様は、前記業務ルールを条件と結果を示す記述形式で記述した情報である複数のルールを含み、

前記情報記憶部は、前記範囲条件として、前記ルール仕様における前記ルールの入力項目の値に関する条件を示す情報が設定された前記テストケース生成条件を記憶し、

前記範囲条件論理式生成部は、前記ルールの入力項目の値に関する条件を示す情報に基づき前記範囲条件論理式を生成する、

テストケース生成装置。

10

【請求項 3】

請求項 1 又は 2 に記載のテストケース生成装置であって、

前記ルール仕様は、前記業務ルールを条件と結果を示す記述形式で記述した情報である複数のルールを含み、

前記生成方式は、

前記ルールの前記条件の組み合わせを網羅するように前記テストケースを生成する方式である条件網羅、

前記ルールのセットであるルールセットの間の前記ルールの組み合わせを網羅するように前記テストケースを生成する方式であるパス網羅、

前記ルールセットに含まれている前記ルールが少なくとも一回以上実行されるように前記テストケースを生成する方式であるルール網羅、

20

All Pairs 法を用いて 2 因子網羅となるように前記テストケースを生成する方式である All Pairs / 2 因子、及び、

All Pairs 法を用いて 3 因子網羅となるように前記テストケースを生成する方式である All Pairs / 3 因子、

のうちの少なくともいずれかである、

テストケース生成装置。

【請求項 4】

業務ルールの仕様を記述したルール仕様に基づき生成されるプログラムのテストに用いるテストケースを生成するテストケース生成装置であって、

30

前記ルール仕様と、テストケースを生成する範囲に関する情報である範囲条件、及びテストケースの生成の方式を指定する情報である生成方式を含む、一つ以上のテストケース生成条件と、を記憶する情報記憶部と、

前記ルール仕様を表す論理式であるルール仕様論理式を生成するルール仕様論理式生成部と、

前記テストケース生成条件ごとに、テストケースを生成する範囲を表す論理式である範囲条件論理式を生成する、範囲条件論理式生成部と、

前記ルール仕様論理式と前記範囲条件論理式とを論理積によって結合した論理式であるテストケース生成用論理式を生成するテストケース生成用論理式生成部と、

前記テストケース生成用論理式に基づきテストケースを生成するテストケース生成部と

40

を備え、

前記情報記憶部は、前記テストケース生成条件の夫々について設定された優先度を示す情報を記憶し、

前記テストケース生成部は、複数の前記テストケース生成条件の夫々の前記範囲条件論理式の間に重複があるか否かを判定し、重複がある場合、重複が無くなるように少なくともいずれかの前記範囲条件論理式を修正する、

テストケース生成装置。

【請求項 5】

請求項 4 に記載のテストケース生成装置であって、

50

前記テストケース生成部は、前記判定に際し、前記判定の対象となる複数の前記範囲条件論理式の論理積の充足可能性を調べ、充足可能な場合は前記複数の範囲条件論理式の間に重複があると判定し、充足不能な場合は前記複数の範囲条件論理式の間に重複が無いと判定する、

テストケース生成装置。

【請求項 6】

請求項 1 又は 2 に記載のテストケース生成装置であって、

前記テストケース生成部は、前記テストケース生成用論理式の充足可能性を判定し、充足可能な場合、充足可能となるような前記ルール仕様における入力項目の入力値の組み合わせを含むデータをテストケースとして生成する、

テストケース生成装置。

【請求項 7】

請求項 6 に記載のテストケース生成装置であって、

前記テストケース生成部は、前記テストケース生成用論理式の充足可能性を判定し、充足可能な場合、充足可能となるルール仕様における入力項目の入力値の組み合わせと前記組み合わせに対して期待される出力項目の出力値とを含むデータをテストケースとして生成する、

テストケース生成装置。

【請求項 8】

情報処理装置が、

業務ルールの仕様を記述したルール仕様に基づき生成されるプログラムのテストに用いるテストケースを生成する範囲に関する情報である範囲条件、及びテストケースの生成の方式を指定する情報である生成方式を含む、一つ以上のテストケース生成条件と、を記憶するステップ、

前記ルール仕様を表す論理式であるルール仕様論理式を生成するステップ、

前記テストケース生成条件ごとに、テストケースを生成する範囲を表す論理式である範囲条件論理式を生成するステップ、

前記ルール仕様論理式と前記範囲条件論理式とを論理積によって結合した論理式であるテストケース生成用論理式を生成するステップ、

前記テストケース生成用論理式に基づきテストケースを生成するステップ、

前記範囲条件として前記ルール仕様の変更前と変更後の差分を示す情報が設定された前記テストケース生成条件を記憶するステップ、及び、

前記ルール仕様の変更前と変更後の差分に基づき前記範囲条件論理式を生成するステップ、

を実行する、テストケース生成方法。

【請求項 9】

請求項 8 に記載のテストケース生成方法であって、

前記ルール仕様は、前記業務ルールを条件と結果を示す記述形式で記述した情報である複数のルールを含み、

前記情報処理装置が、

前記範囲条件として前記ルール仕様における前記ルールの入力項目の値に関する条件を示す情報が設定された前記テストケース生成条件を記憶するステップ、

前記ルールの入力項目の値に関する条件を示す情報に基づき前記範囲条件論理式を生成するステップ、

をさらに実行する、テストケース生成方法。

【請求項 10】

請求項 8 又は 9 に記載のテストケース生成方法であって、

前記ルール仕様は、前記業務ルールを条件と結果を示す記述形式で記述した情報である複数のルールを含み、

前記生成方式は、

10

20

30

40

50

前記ルールの前記条件の組み合わせを網羅するように前記テストケースを生成する方式である条件網羅、

前記ルールのセットであるルールセットの間の前記ルールの組み合わせを網羅するように前記テストケースを生成する方式であるパス網羅、

前記ルールセットに含まれている前記ルールが少なくとも一回以上実行されるように前記テストケースを生成する方式であるルール網羅、

All Pairs法を用いて2因子網羅となるように前記テストケースを生成する方式であるAll Pairs / 2因子、及び、

All Pairs法を用いて3因子網羅となるように前記テストケースを生成する方式であるAll Pairs / 3因子、

のうちの少なくともいずれかである、

テストケース生成方法。

【請求項11】

情報処理装置が、

業務ルールの仕様を記述したルール仕様に基づき生成されるプログラムのテストに用いるテストケースを生成する範囲に関する情報である範囲条件、及びテストケースの生成の方式を指定する情報である生成方式を含む、一つ以上のテストケース生成条件と、を記憶するステップ、

前記ルール仕様を表す論理式であるルール仕様論理式を生成するステップ、

前記テストケース生成条件ごとに、テストケースを生成する範囲を表す論理式である範囲条件論理式を生成するステップ、

前記ルール仕様論理式と前記範囲条件論理式とを論理積によって結合した論理式であるテストケース生成用論理式を生成するステップ、

前記テストケース生成用論理式に基づきテストケースを生成するステップ、

前記テストケース生成条件の夫々について設定された優先度を示す情報を記憶するステップ、及び、

複数の前記テストケース生成条件の夫々の前記範囲条件論理式の間に重複が有るか否かを判定し、重複が有る場合、重複が無くなるように少なくともいずれかの前記範囲条件論理式を修正するステップ、

を実行する、テストケース生成方法。

【請求項12】

請求項11に記載のテストケース生成方法であって、

前記情報処理装置が、

前記判定に際し、前記判定の対象となる複数の前記範囲条件論理式の論理積の充足可能性を調べ、充足可能な場合は前記複数の範囲条件論理式の間に重複が有ると判定し、充足不能な場合は前記複数の範囲条件論理式の間に重複が無いと判定するステップ、

をさらに実行する、テストケース生成方法。

【請求項13】

請求項8又は9に記載のテストケース生成方法であって、

前記情報処理装置が、

前記テストケース生成用論理式の充足可能性を判定し、充足可能な場合、充足可能となるような前記ルール仕様における入力項目の入力値の組み合わせを含むデータをテストケースとして生成するステップ、

をさらに実行する、テストケース生成方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、テストケース生成装置、及びテストケース生成方法に関する。

【背景技術】

【0002】

10

20

30

40

50

特許文献１には、「ルール群ネットワークがルートノード、１つ又は複数の中間ノード、及び１つ又は複数の端末ノードを含み、各ノードについて局所的な制限が個別に設定される。生成装置は、ルール群ネットワークを走査して、祖先ノードの局所的な制限と自分の局所的な制限とを組み合わせることで各ノードの大域的な制限を算出する走査部と、端末ノードの大域的な制限に基づいて、テストケースを生成するための事実情報を取得する事実情報取得部と、テストケースに含まれる必要がある事実群と各事実属性の値範囲とを含む事実情報に基づいてテストケースを生成するテストケース生成部とを含む。これにより、ビジネスルール管理システム（ＢＲＭＳ）においてテストケースを自動的に生成できる。」と記載されている。

【先行技術文献】

【特許文献】

【０００３】

【特許文献１】特開２０１５－２０４１１６号公報

【発明の概要】

【発明が解決しようとする課題】

【０００４】

業務（金融、保険、製造、流通、通信、教育、医療等）を支援するソフトウェア（以下、業務ソフトウェアと称する。）は、業務ルール（業務に係る商品やサービス等に関する法的な規制、業務上の規則、基準等。ビジネスルール、システムルール、チェックルール等とも称される。）に従ってその仕様が決定され設計がなされる。上記仕様はＩＦ－ＴＨ

【０００５】

特許文献１では、ルール仕様にに基づき一律にテストケースを生成している。しかし業務ソフトウェアの開発や保守の現場では、テストケースの生成に関し、例えば、ルール仕様の変更前と変更後の差分や過去に不具合が発生しているパターン等、プログラムの特定の記述部分や論理についてテストケースの数を増やして重点的にテストを行いたい、プログラムのルール仕様を変更していない記述部分や論理についてはテストケースの数を減らしたい、といった多様なニーズが存在し、こうした要求に効率よく対応する必要がある。

【０００６】

本発明はこうした背景に鑑みてなされたもので、テストケースの生成に関する多様なニーズに柔軟に対応することが可能な、テストケース生成装置、及びテストケース生成方法を提供することを目的とする。

【課題を解決するための手段】

【０００７】

本発明の一つは、業務ルールの仕様が記述したルール仕様にに基づき生成されるプログラムのテストに用いるテストケースを生成するテストケース生成装置であって、前記ルール仕様と、テストケースを生成する範囲に関する情報である範囲条件、及びテストケースの生成の方式を指定する情報である生成方式を含む、一つ以上のテストケース生成条件と、を記憶する情報記憶部と、前記ルール仕様を表す論理式であるルール仕様論理式を生成するルール仕様論理式生成部と、前記テストケース生成条件ごとに、テストケースを生成する範囲を表す論理式である範囲条件論理式を生成する、範囲条件論理式生成部と、前記ルール仕様論理式と前記範囲条件論理式とを論理積によって結合した論理式であるテストケース生成用論理式を生成するテストケース生成用論理式生成部と、前記テストケース生成用論理式に基づきテストケースを生成するテストケース生成部と、を備え、前記情報記憶部は、前記範囲条件として、前記ルール仕様の変更前と変更後の差分を示す情報が設定された前記テストケース生成条件を記憶し、前記範囲条件論理式生成部は、前記ルール仕様の変更前と変更後の差分に基づき前記範囲条件論理式を生成する。

【 0 0 0 8 】

その他、本願が開示する課題、及びその解決方法は、発明を実施するための形態の欄、及び図面により明らかにされる。

【発明の効果】

【 0 0 0 9 】

本発明の実施形態によれば、テストケースの生成に関する多様なニーズに柔軟に対応することができる。

【図面の簡単な説明】

【 0 0 1 0 】

【図 1】テストケース生成装置のハードウェア構成例を示す図である。

10

【図 2】テストケース生成装置が備える機能及びテストケース生成装置が記憶するデータを示す図である。

【図 3】ルール仕様の一例を示す図である。

【図 4】ルール仕様が変更される場合の一例を示す図である。

【図 5】テストケース生成条件情報の一例を示す図である。

【図 6】テストケースの一例を示す図である。

【図 7】メイン処理を説明するフローチャートである。

【図 8】生成方式として「条件網羅」が設定されているテストケース生成条件に基づくテストケース生成処理を説明するフローチャートである。

【図 9】生成方式として「パス網羅」が設定されているテストケース生成条件に基づくテストケース生成処理を説明するフローチャートである。

20

【発明を実施するための形態】

【 0 0 1 1 】

以下、図面を参照しつつ実施形態について説明する。尚、以下に説明する実施形態は特許請求の範囲にかかる発明を限定するものではない。また実施形態において説明されている諸要素及びその組み合わせの全てが発明の解決手段に必須であるとは限らない。

【 0 0 1 2 】

図 1 に、一実施形態として示すテストケース生成装置 10 のハードウェア構成を、図 2 に、テストケース生成装置 10 が備える機能及びテストケース生成装置 10 が記憶するデータを、夫々示している。

30

【 0 0 1 3 】

テストケース生成装置 10 は、業務（金融、保険、製造、流通、通信、教育、医療等）を支援するソフトウェア（以下、業務ソフトウェアと称する。）の開発や保守等に利用される情報処理装置（コンピュータ）である。テストケース生成装置 10 は、クラウドシステムにより提供されるクラウドサーバ等のように仮想的に実現されるものであってもよい。

【 0 0 1 4 】

業務ソフトウェアは、業務ルール（業務に係る商品やサービス等に関する法的な規制、業務上の規則、基準等。ビジネスルール、システムルール、チェックルール等とも称される。）に従ってその仕様が決定され設計がなされる。上記仕様は、いわゆる IF - THEN 形式で記述される情報（以下、ルールと称する。）の集まりであるルール仕様によって整理される。上記のルールは、例えば、「もし～ならば～である」といった自然文による記述、「IF A = 1 AND B = 2 THEN R = 1」というようにコンピュータにより解釈可能な所定の規則に従った記述、条件と結果の対応関係を表形式で定義した決定表による記述等、様々な形態で表現され得る。

40

【 0 0 1 5 】

業務ソフトウェアの開発や保守に際しては、上記ルール仕様を基に情報処理装置に実行させるプログラムが作成される。そしてルール仕様を基に作成したテストケースをプログラムに与えることにより、プログラムがルール仕様通りに作成されていることのテスト（検証）が行われる。

50

【 0 0 1 6 】

テストケース生成装置 1 0 は、ルール仕様にに基づき生成されるプログラムをテスト（検査）するためのテストケースを生成する。テストケースは、入力条件と、当該入力条件に対して業務ソフトウェアを実行した際に期待される結果（以下、期待結果と称する。）と、を含む。

【 0 0 1 7 】

テストケース生成装置 1 0 は、テストケースを生成する範囲に関する条件である範囲条件と当該範囲条件の下で適用するテストケースの生成の方式とを含む情報であるテストケース生成条件をユーザ等から受け付け、ルール仕様と上記テストケース生成条件とに基づき、テストケース生成条件ごとに、テストケースを生成する範囲を表す論理式である範囲条件論理式を生成する。

10

【 0 0 1 8 】

テストケース生成条件は、当該テストケース生成条件の範囲条件論理式と他のテストケース生成条件の範囲条件論理式との間に重複が有る場合に何れのテストケース生成条件を優先させるかを示す情報（後述する優先度）を含む。テストケース生成装置 1 0 は、複数のテストケース生成条件の間の範囲条件論理式の重複の有無を調べ、重複が有る場合は重複が無くなるように範囲条件論理式を修正する。

【 0 0 1 9 】

テストケースの生成に際し、テストケース生成装置 1 0 は、ルール仕様を表す論理式であるルール仕様論理式を生成し、生成したルール仕様論理式と範囲条件論理式とを論理積で結合したテストケース生成用論理式を生成する。そしてテストケース生成装置 1 0 は、生成したテストケース生成用論理式の充足可能性を判定し、テストケース生成用論理式が充足可能と判定される場合、充足可能となるルール仕様の入力項目の値の組み合わせと出力項目の値とを含むデータをテストケースとして生成する。尚、充足可能性の詳細については後述する。

20

【 0 0 2 0 】

図 1 に示すように、テストケース生成装置 1 0 は、プロセッサ 1 1、主記憶装置 1 2、補助記憶装置 1 3、入力装置 1 4、出力装置 1 5、及び通信装置 1 6 を備える。これらは図示しないバス等の通信手段を介して互いに通信可能に接続されている。

【 0 0 2 1 】

プロセッサ 1 1 は、例えば、C P U（Central Processing Unit）や M P U（Micro Processing Unit）を用いて構成されている。プロセッサ 1 1 が、主記憶装置 1 2 に格納されているプログラムを読み出して実行することにより、テストケース生成装置 1 0 の様々な機能が実現される。主記憶装置 1 2 は、プログラムやデータを記憶する装置であり、例えば、R O M（Read Only Memory）、R A M（Random Access Memory）、不揮発性半導体メモリ（N V R A M（Non Volatile RAM））等である。

30

【 0 0 2 2 】

補助記憶装置 1 3 は、例えば、ハードディスクドライブ、S S D（Solid State Drive）、光学式記憶装置（C D（Compact Disc）、D V D（Digital Versatile Disc）等）、ストレージシステム、I C カード、S D メモリカードや光学式記録媒体等の記録媒体の読取／書込装置、クラウドサーバの記憶領域等である。補助記憶装置 1 3 に格納されているプログラムやデータは主記憶装置 1 2 に随時ロードされる。

40

【 0 0 2 3 】

入力装置 1 4 は、例えば、キーボード、マウス、タッチパネル、カードリーダ、音声入力装置等である。出力装置 1 5 は、ユーザに処理経過や処理結果等の各種情報を提供するユーザインタフェースであり、例えば、画面表示装置（液晶モニタ、L C D（Liquid Crystal Display）、グラフィックカード等）、音声出力装置（スピーカ等）、印字装置等である。尚、例えば、テストケース生成装置 1 0 が通信装置 1 6 を介して他の装置との間で情報の入力や出力を行う構成としてもよい。

【 0 0 2 4 】

50

通信装置 16 は、LAN やインターネット等の通信手段を介した他の装置との間の通信を実現する有線方式又は無線方式の通信インタフェースであり、例示すれば、NIC (Network Interface Card)、無線通信モジュール、USB (Universal Serial Interface) モジュール、シリアル通信モジュール等である。

【0025】

図2に示すように、テストケース生成装置10は、テストケース設定編集部110、ルール仕様論理式生成部111、範囲条件論理式生成部112、テストケース生成用論理式生成部113、テストケース生成部114、及び情報記憶部150の各機能(処理部、機能部)を備える。これらの機能は、例えば、プロセッサ11が、主記憶装置12や補助記憶装置13に格納されているプログラムを読み出して実行することにより実現される。またこれらの機能は、例えば、テストケース生成装置10が備えるハードウェア(ASIC (Application Specific Integrated Circuit) 等)によって実現される。

10

【0026】

情報記憶部150は、ルール仕様151、テストケース生成条件情報152、及びテストケース153を、補助記憶装置13に記憶する。これらの情報の詳細については後述する。情報記憶部150は、これらの情報以外にも、テストケース設定編集部110やテストケース生成部114が適宜参照もしくは生成する情報等を記憶する。情報記憶部150は、補助記憶装置13に記憶する情報を、例えば、ファイルシステムやDBMS (Database Management System) によって管理する。

【0027】

テストケース設定編集部110は、ユーザが入力装置14及び出力装置15を介してテストケース生成条件情報152等の各種設定(登録、変更、削除)を行う際のユーザインタフェースを提供する。

20

【0028】

ルール仕様論理式生成部111は、ルール仕様151を表す論理式であるルール仕様論理式を生成する。

【0029】

範囲条件論理式生成部112は、テストケース生成条件ごとに、テストケースを生成する範囲を表す論理式である範囲条件論理式を生成する。

【0030】

テストケース生成用論理式生成部113は、ルール仕様論理式と範囲条件論理式とを論理積によって結合した論理式であるテストケース生成用論理式を生成する。

30

【0031】

テストケース生成部114は、テストケース生成用論理式に基づきテストケースを生成する。尚、テストケース生成部114が生成したテストケースは情報記憶部150がテストケース153として補助記憶装置13に記憶する。

【0032】

図3に情報記憶部150が記憶するルール仕様151の一例を示している。同図に示すように、例示するルール仕様151は、識別子(以下、IDと表記することがある。)として、「RS1」が付与されたルール群(名称:ランク判定)と、識別子として「RS2」が付与されたルール群(名称:割引率判定)とを含む。以下、上記ルール群のことをルールセットと称する。また各ルールセットは、ルールセットRS1、ルールセットRS2等と表記して区別する。同図に示すように、ルールセットRS1は、夫々R1~R3の識別子が付された3つのルールを含む。ルールセットRS2は、夫々R1~R5の識別子が付された5つのルール(以下、各ルールをルールR1、ルールR2、・・・と表記する。)を含む。

40

【0033】

各ルールは、IF - THEN形式における記述のIF節に相当する条件部と、IF - THEN形式の記述におけるTHEN節に相当する結果部とを含む。例えば、ルールセットRS1のルールR1は「IF Amount < 20000 THEN Rank = Reg

50

ular」との内容であり、条件部は「Amount < 20000」、結果部は「Rank = Regular」である。

【0034】

以下の説明において、ルール条件部に記述されている変数のことを入力項目と称し、結果部に記述されている変数のことを出力項目と称する。また入力項目に設定される具体的な値のことを入力値と称し、出力項目に設定される具体的な値のことを出力値と称する。図3の例の場合、ルールセットRS1のルールR1には入力項目として「Amount」が、出力項目として「Rank」が記述されている。またルールセットRS1のルールR1の場合、入力項目「Amount」に設定される値が入力値であり、出力項目「Rank」に設定される値である「Regular」が出力値である。入力値や出力値は必ずしも数値であるとは限らず、文字や文字列等の数値以外で表されるものであってもよい。

10

【0035】

図3の例では、ルールセットRS1の出力項目である「Rank」は、ルールセットRS2の入力項目の一つとして用いられている。またルールセットRS2の入力項目のように、ルールセット及びルールの入力項目は一つとは限らない。入力項目が複数である場合、個々の入力項目の条件式は、論理演算子（AND、OR等）で結合される。例えば、図3のルールセットRS2では論理演算子「AND」で条件式が結合されている。

【0036】

複数種の出力項目に共通する（共用できる）ルールがある場合、一つのルールセットに複数種の出力項目をもたせることも可能である。しかしルールセットに複数種の出力項目をもたせると、例えば、業務規則等が変更された際のルールの変更が煩雑になる。そこでルールセットの意味の分かり易さやルールの変更の容易さ等の観点から、本実施形態では同じルールセットに含まれる各ルールの出力項目を同一としている。例えば、ルールセットRS1の各ルールの出力項目は同じ「Rank」としている。

20

【0037】

同じルールセットに含まれるルールの入力項目に設定される入力値は、いずれか一つのルールの条件部を真とする（成立させる）。例えば、ルールセットRS1の入力項目「Amount」に設定される入力値は、ルールR1～R3のいずれか一つの条件部を真とする。

【0038】

ルール仕様151は、業務ソフトウェアの開発や保守に際し適宜変更され得る。テストケース生成装置10は、ルール仕様151をバージョン別に管理している。テストケース生成装置10が記憶する各ルール仕様151には夫々のバージョンを示す情報（以下、バージョン情報と称する。）が付帯する。

30

【0039】

図4に、バージョン情報が「V1」のルール仕様151（以下、ルール仕様V1とも称する。）、及びルール仕様V1を更新したバージョン情報が「V2」のルール仕様151（以下、ルール仕様V2とも称する。）の組を例示している。同図においてルール仕様V2に下線を付して示した部分がルール仕様V1とルール仕様V2の差分である。

【0040】

本例では、ルール仕様V2に下線で示しているように、ルールセットRS1のルールR3の入力項目「Amount」に上限が追加され、ルールセットRS1の出力値に「Platinum」を出力する内容のルールR4が追加されている。またルールセットRS2のルールR1の条件部が変更され、ルールR4がルールセットRS2の入力項目として「Platinum」を用いる内容に変更され、「Rank」に「Platinum」を設定するルールR6が追加されている。

40

【0041】

図5は、情報記憶部150が記憶するテストケース生成条件情報152の一例である。同図に示すように、テストケース生成条件情報152は、テスト対象1521、テスト種別1522、条件ID1523、範囲条件1524（範囲種別15241、テスト範囲1

50

5 2 4 2を含む。) 、生成方式 1 5 2 5、境界値ケース要否 1 5 2 6、及び優先度 1 5 2 7の各項目を有する。

【 0 0 4 2 】

テスト対象 1 5 2 1には、テスト対象となるルールセットの識別子が設定される。例えば、ルール仕様 1 5 1が図 3に示す内容である場合、テスト対象 1 5 2 1には「 R S 1」や「 R S 2」が設定される。図 5の例ではテスト対象 1 5 2 1に「 R S 2」が設定されており、従って当該テストケース生成条件情報 1 5 2は識別子が「 R S 2」のルールセットに関するものということになる。

【 0 0 4 3 】

テスト種別 1 5 2 2には「単体」又は「結合」のいずれかが設定される。テスト種別 1 5 2 2に「単体」が設定されている場合、テストケース生成装置 1 0は、当該テストケース生成条件情報 1 5 2に基づき、ルールセットごとに独立したテストケースを生成する。例えば、図 3の例では、ルールセット R S 2の入力項目の 1 つにルールセット R S 1の出力項目「 R a n k」が含まれているが、テストケース生成装置 1 0は、テストケースの生成に際しルールセット R S 1を考慮せず、従ってこの場合、入力条件に含まれる入力項目は「 A g e」と「 R a n k」となる。

【 0 0 4 4 】

一方、テスト種別 1 5 2 2に「結合」が設定されている場合、テストケース生成装置 1 0は、同じルール仕様 1 5 1に属する複数のルールセットの関連を考慮したテストケースを生成する。例えば、図 3の例では、ルールセット R S 2の入力項目「 R a n k」の値は、ルールセット R S 1の入力項目である「 A m o u n t」によって一意に定まる。そのため、テストケースの生成に際し、テストケース生成装置 1 0は、入力条件から「 R a n k」を除外し、入力条件を「 A m o u n t」及び「 A g e」とする。

【 0 0 4 5 】

図 5に戻り、条件 I D 1 5 2 3、範囲条件 1 5 2 4（範囲種別 1 5 2 4 1、テスト範囲 1 5 2 4 2を含む。） 、生成方式 1 5 2 5、境界値ケース要否 1 5 2 6、及び優先度 1 5 2 7の項目からなる各行は、夫々、前述したテストケース生成条件に相当する。

【 0 0 4 6 】

上記項目のうち条件 I D 1 5 2 3には各テストケース生成条件の識別子が設定される。

【 0 0 4 7 】

範囲条件 1 5 2 4には、前述した範囲条件を指定する情報が設定される。範囲条件 1 5 2 4の範囲種別 1 5 2 4 1には、テストケース生成条件の種別を指定する情報が設定される。本例では、「仕様差分」、「入力条件」、「出力条件」、及び「 D e f a u l t」のいずれかが設定される。範囲条件 1 5 2 4のテスト範囲 1 5 2 4 2には、範囲条件に関するより具体的な内容が設定される。このテスト範囲 1 5 2 4 2に設定可能な内容は、範囲種別 1 5 2 4 1の設定内容によって異なる。以下具体例を示す。

【 0 0 4 8 】

例えば、範囲種別 1 5 2 4 1に「仕様差分」が設定されている場合、テスト範囲 1 5 2 4 2には「変更有」、「変更無」のいずれかを設定することができる。ここで「仕様差分」とは、ルール仕様 1 5 1の変更前と変更後の差分を意味する。テスト範囲 1 5 2 4 2に「変更有」が設定されている場合、テストケース生成装置 1 0は、例えば、ルール仕様 1 5 1の変更によってルール仕様 1 5 1に対する入力と出力の関係が変化した範囲に関する入力項目の値の組み合わせをテストケースとして生成する。またテスト範囲 1 5 2 4 2に「変更無」が設定されている場合、テストケース生成装置 1 0は、例えば、入力と出力の関係が変化していない範囲に関する入力項目の値の組み合わせをテストケースとして生成する。尚、本例では、変更前のルール仕様 1 5 1をルール仕様 V 2の 1 つ前のバージョンのルール仕様 V 1としているが、変更後のルール仕様と比較する変更前のルール仕様のバージョンをユーザ等が任意に指定できるようにしてもよい。

【 0 0 4 9 】

範囲種別 1 5 2 4 1に「入力条件」が設定されている場合、テスト範囲 1 5 2 4 2には

入力項目の値に関する条件を示す論理式を設定することができる。この場合、テストケース生成装置 10 は、入力項目の値がテスト範囲 15242 に設定されている論理式を満たすようなテストケースを生成する。

【0050】

範囲種別 15241 に「出力条件」が設定されている場合、テスト範囲 15242 には出力項目の値に関する条件を示す論理式を設定することができる。この場合、テストケース生成装置 10 は、出力項目の値がテスト範囲 15242 に設定された論理式を満たすようなテストケースを生成する。

【0051】

範囲種別 15241 に「Default」が設定されている場合、テスト範囲 15242 には「-」が設定される。この場合、テストケース生成装置 10 は、当該テストケース生成条件情報 152 における他のテストケース生成条件のいずれの条件も満たさない入力項目の値の組み合わせをテストケースとして生成する。

10

【0052】

その他、範囲種別 15241 及びテスト範囲 15242 として、例えば、ソースコードの差分、入力項目と出力項目の両方を使った条件等を設定可能としてもよい。その場合、ソースコードの差分は、例えば、記号実行を使うことにより特定することができる。

【0053】

生成方式 1525 には、テストケースの生成方式を指定する情報（以下、生成方式とも称する。）が設定される。本実施形態では、「条件網羅」、「パス網羅」、「ルール網羅」、「All Pairs / 2 因子」、「All Pairs / 3 因子」のいずれかが設定されるものとする。

20

【0054】

生成方式 1525 に「条件網羅」が設定されている場合、テストケース生成装置 10 は、ルールの条件部の最小構成要素である条件項（例えば図 3 における「Amount < 20000」、「Age ≤ 35」等）の True（真）/ False（偽）の組み合わせを網羅するようにテストケースを生成する。

【0055】

「パス網羅」は、テスト種別 1522 に「結合」が設定されているときにのみ生成方式 1525 に設定することができる。生成方式 1525 に「パス網羅」が設定されている場合、テストケース生成装置 10 は、ルール仕様 151 の各ルールセット間のルールの組み合わせを網羅するようにテストケースを生成する。例えば、図 3 の例では、テストケース生成装置 10 は、ルールセット RS1 のルール R1 とルールセット RS2 のルール R1（以下、{RS1 . R1、RS2 . R1} のように表記する。）、{RS1 . R1、RS2 . R2}、{RS1 . R1、RS2 . R3} 等の組み合わせを網羅するようにテストケースを生成する。

30

【0056】

尚、例えば、図 4 のルール仕様 V2 の例では、ルールセット RS1 に含まれるルール数が 4 つ、ルールセット RS2 に含まれるルール数が 6 つであり、この場合、テストケース生成装置 10 は、最大で $4 \times 6 = 24$ 個のテストケースを生成する。但し実際には実行不能なパスが存在するので、組み合わせの数は上記のような単純計算によって求められる値よりも少なくなる。

40

【0057】

生成方式 1525 に「ルール網羅」が設定されている場合、テストケース生成装置 10 は、テスト対象のルールセットに含まれているルールが少なくとも 1 回以上実行されるようにテストケースを生成する。

【0058】

生成方式 1525 に「All Pairs / 2 因子」が設定されている場合、テストケース生成装置 10 は、All Pairs 法を用いて 2 因子網羅となるようにテストケースを生成する。

50

【 0 0 5 9 】

生成方式 1 5 2 5 に「 A l l P a i r s / 3 因子」が設定されている場合、テストケース生成装置 1 0 は、 A l l P a i r s 法を用いて 3 因子網羅となるようにテストケースを生成する。

【 0 0 6 0 】

境界値ケース要否 1 5 2 6 には、テストケース生成装置 1 0 がテストケースを生成する際に境界値に関するテストケースを生成するか否かを表すフラグ（「要」、「不要」、「 - 」）が設定される。本例では、生成方式 1 5 2 5 に、「条件網羅」、「 A l l P a i r s / 2 因子」、「 A l l P a i r s / 3 因子」が設定されている場合にのみ、境界値ケース要否 1 5 2 6 に「要」又は「不要」を設定可能であり、それ以外の場合は「 - 」が設定される。境界値に関するテストケースの具体例については後述する。

10

【 0 0 6 1 】

優先度 1 5 2 7 には、複数のテストケース生成条件の夫々に基づく範囲条件論理式の間に重複が有る場合、即ち、あるテストケース生成条件の範囲種別 1 5 2 4 1 とテスト範囲 1 5 2 4 2 とによって定まる入力値の範囲（以下、入力空間とも称する。）が他のテストケース生成条件の入力空間と重複する場合、いずれのテストケース生成条件を優先するかを決定するための値（以下、優先度と称する。）が設定される。尚、本例では、優先度が大きいテストケース生成条件の生成方式 1 5 2 5 が優先されるものとする。また複数のテストケース生成条件の優先度 1 5 2 7 に同じ優先度が設定されている場合、例えば、重複する部分について、双方のテストケース生成条件の生成方式 1 5 2 5 に設定されている生成方式によってテストケースを生成する。

20

【 0 0 6 2 】

図 6 に、テストケース生成装置 1 0 が図 4 に示すルール仕様 1 5 1、及び図 5 に示すテストケース生成条件情報 1 5 2 に基づき生成したテストケースの例を示す。同図における各行は、夫々、テストケース生成装置 1 0 が生成する一つのテストケースに相当する。同図に示すように、テストケース 1 5 3 は、テストケース I D 1 5 3 1、条件 I D 1 5 3 2、入力条件 1 5 3 3、及び期待結果 1 5 3 4 の各項目を有する。テストケース I D 1 5 3 1 には、テストケース 1 5 3 の識別子が設定される。

【 0 0 6 3 】

条件 I D 1 5 3 2 には、当該テストケース 1 5 3 が、テストケース生成条件情報 1 5 2 のいずれのテストケース生成条件に基づき生成したテストケースであるかを識別する識別子が設定される。本例では、テストケース生成条件情報 1 5 2 の条件 I D 1 5 2 3 に設定される識別子が設定されている。

30

【 0 0 6 4 】

入力条件 1 5 3 3 には、ルール仕様 1 5 1 の入力項目の入力値が設定される。図 5 のテストケース生成条件情報 1 5 2 では、テスト対象 1 5 2 1 に「 R S 2 」が設定されており、テスト種別 1 5 2 2 に「結合」が設定されているので、テストケース生成装置 1 0 は、入力項目「 A g e 」及び「 A m o u n t 」の夫々の入力値を生成して入力条件 1 5 3 3 に設定する。

【 0 0 6 5 】

期待結果 1 5 3 4 には、ルール仕様 1 5 1 が入力条件を満たす場合のルールセットの出力項目の出力値が設定される。本例では、出力項目は「 D i s c o u n t 」であり、テストケース生成装置 1 0 は、入力条件 1 5 3 3 の「 A g e 」、「 A m o u n t 」の夫々に設定した入力値に基づき「 D i s c o u n t 」の出力値を求めて期待結果 1 5 3 4 に設定する。

40

【 0 0 6 6 】

図 7 は、テストケースの生成に際してテストケース生成装置 1 0 が行う処理（以下、メイン処理 S 7 0 0 と称する。）を説明するフローチャートである。以下、同図とともにメイン処理 S 7 0 0 について説明する。尚、テストケース生成部 1 1 4 は、例えば、ユーザからの指示を契機としてメイン処理 S 7 0 0 を開始する。またテストケース生成部 1 1 4

50

は、例えば、バッチ処理等により自動的に処理を実行する。

【0067】

同図に示すように、まずテストケース生成装置10は、情報記憶部150が記憶しているテストケース生成条件情報152を読み込む(S711)。またテストケース生成装置10は、テストケース生成に必要なルール仕様151を読み込む(S712)。例えば、テストケース生成条件情報152が図5の内容である場合、テスト対象1521がルールセットRS2であり、テスト種別1522に「結合」が設定されているので、テストケース生成装置10は、図4の最新バージョンのルール仕様151であるルール仕様V2の内容(ルールセットRS1及びルールセットRS2)を読み込む。また図5のテストケース生成条件情報152では、範囲種別15241に「仕様差分」が設定されているので、

10

【0068】

続いて、テストケース生成装置10は、S711で読み込んだテストケース生成条件情報152、及びS712で読み込んだルール仕様151に基づき、テストケース生成条件情報152の各テストケース生成条件について前述した範囲条件論理式を生成する(S713)。

【0069】

例えば、図5のテストケース生成条件情報152における条件ID1523が「C1」のテストケース生成条件の場合、範囲種別15241に「仕様差分」が設定され、テスト範囲15242に「変更有」が設定されているので、テストケース生成装置10は「DiscountV1 DiscountV2」という範囲条件論理式を生成する。ここで「DiscountV1」は、ルール仕様V1のルールセットRS2の出力項目、「DiscountV2」は、ルール仕様V2のルールセットRS2の出力項目である。上記の範囲条件論理式「DiscountV1 DiscountV2」は、ルール仕様151のルール仕様151の変更前と変更後とで出力値が異なることを表す。尚、同じテストケース生成条件でテスト範囲15242に「変更無」が設定されている場合、範囲条件論理式は「DiscountV1 = DiscountV2」となる。

20

【0070】

また例えば、図5のテストケース生成条件情報152における条件ID1523が「C2」のテストケース生成条件の場合、範囲種別15241に「入力条件」が設定され、テスト範囲15242に「Age > 60」が設定されているので、テストケース生成装置10は、「Age > 60」という範囲条件論理式を生成する。

30

【0071】

また例えば、図5のテストケース生成条件情報152における条件ID1523が「C3」のテストケース生成条件の場合、範囲種別15241に「Default」が設定されているので、テストケース生成装置10は、条件ID1523が「C1」のテストケース生成条件の範囲条件論理式と、条件ID1523が「C2」のテストケース生成条件の範囲条件論理式とを除いた範囲に相当する範囲条件論理式「 \neg (範囲条件論理式C1 範囲条件論理式C2)」を生成する。尚、「 \vee 」は論理和を表し、「 \neg 」は論理否定を表す。

40

【0072】

図7に戻り、続いてテストケース生成装置10は、S713で生成した複数の範囲条件論理式の間に重複が有るか否か(重複する入力空間が存在するか否か)を判定する(S714)。ここでテストケース生成装置10は、複数の範囲条件論理式の間に重複が有るか否かを、例えば、テストケース生成条件のうち範囲種別が「Default」であるテストケース生成条件を除く任意の2つのテストケース生成条件について生成した範囲条件論理式の論理積の充足可能性を判定することにより行う。具体的には、テストケース生成装置10は、上記判定により充足可能と判定した場合は重複が有ると判定し、上記判定により充足不能と判定した場合は重複が無いと判定する。例えば、図5のテストケース生成条

50

件情報 1 5 2 の場合、テストケース生成装置 1 0 は、条件 I D 1 5 2 3 が「C 1」のテストケース生成条件の範囲条件論理式と、条件 I D 1 5 2 3 が「C 2」のテストケース生成条件の範囲条件論理式の論理積の充足可能性を判定する。

【 0 0 7 3 】

ここで充足可能性とは、与えられた論理式を T R U E（真）にする変数の値の組み合わせが存在するか否かを表す性質をいう。充足可能性は、単純には、全ての変数の値の組み合わせをチェックすることにより判定することができる。また D P L L（Davis-Putnam-Logemann-Loveland）アルゴリズム等、変数の値の組み合わせではなく論理式の構造に着目して効率よく充足可能性を判定するツールを用いて充足可能性を判定することもできる。例えば、S M T（Satisfiability Modulo Theories）ソルバを用いて充足可能性を判定することができる。

10

【 0 0 7 4 】

S 7 1 4 において範囲条件論理式に重複があると判定した場合（S 7 1 4：Y E S）、テストケース生成装置 1 0 は、テストケース生成条件の夫々の優先度 1 5 2 7 に基づき、重複が無くなるように少なくともいずれかの範囲条件論理式を修正する（S 7 1 5）。例えば、図 5 のテストケース生成条件情報 1 5 2 の場合、条件 I D 1 5 2 3 が「C 1」のテストケース生成条件の範囲条件論理式と、条件 I D 1 5 2 3 が「C 2」のテストケース生成条件の範囲条件論理式との間に重複があり、条件 I D 1 5 2 3 が「C 1」のテストケース生成条件の優先度が条件 I D 1 5 2 3 が「C 2」のテストケース生成条件の優先度よりも高い。この場合、テストケース生成装置 1 0 は、条件 I D 1 5 2 3 が「C 1」のテストケース生成条件の範囲条件論理式はそのままとし、条件 I D 1 5 2 3 が「C 2」のテストケース生成条件の範囲条件論理式を「範囲条件論理式 C 2 \rightarrow 範囲条件論理式 C 1」と修正する。このように「 \rightarrow 範囲条件論理式 C 1」を論理積で結合することで、範囲条件論理式間で重複が無いようにすることができる。また優先度によっていずれのテストケース生成条件を優先するかを選択可能にしたことで、優先すべきテストケース生成条件に対応するテストケースの生成を優先させることができる。

20

【 0 0 7 5 】

S 7 1 4 においてテストケース生成装置 1 0 が範囲条件論理式に重複が無いと判定した場合（S 7 1 4：N O）、処理は S 7 1 6 に進む。

【 0 0 7 6 】

30

図 7 に戻り、S 7 1 6 では、テストケース生成装置 1 0 は、S 7 1 2 で読み込んだルール仕様 1 5 1 を表す論理式（以下、ルール仕様論理式と称する。）を生成する。具体的には、テストケース生成装置 1 0 は、テスト対象のルールセットと、テスト対象のルールセットの入力になっている全てのルールセットの各論理式を論理積（「 \wedge 」）で結合することにより、ルール仕様論理式を生成する。

【 0 0 7 7 】

例えば、図 4 の場合、テストケース生成装置 1 0 は、最新バージョンのルール仕様 1 5 1 であるルール仕様 V 2 の内容（ルールセット R S 1 及びルールセット R S 2）に基づきルール仕様論理式を生成する。また範囲種別 1 5 2 4 1 に「仕様差分」が設定されているテストケース生成条件がテストケース生成条件情報 1 5 2 に存在する場合、テストケース生成装置 1 0 は、他のバージョンのルール仕様 1 5 1 に基づくルール仕様論理式も生成する。図 5 の場合、範囲種別 1 5 2 4 1 に「仕様差分」が設定されているテストケース生成条件がテストケース生成条件情報 1 5 2 に存在するので、テストケース生成装置 1 0 は、図 4 のルール仕様 V 1 の内容（ルールセット R S 1 及びルールセット R S 2）に基づくルール仕様論理式も生成する。

40

【 0 0 7 8 】

例えば、ルール仕様 V 1 のルールセット R S 1 を表す論理式を R S 1 V 2 と表し、ルール仕様 V 2 のルールセット R S 1 を表す論理式を R S 1 V 2 と表した場合、テストケース生成装置 1 0 は「R S 1 V 2 \wedge R S 2 V 2」というルール仕様論理式を生成する。例えば、図 4 の場合、R S 1 V 1 は「(A m o u n t < 2 0 0 0 0 R a n k V 1 = R e g u l

50

ar) (20000 ≤ Amount < 50000 RankV1 = Silver)
(Amount ≥ 50000 RankV1 = Gold)」となる。尚、上記の論理式
における「RankV1」のように、出力項目にはバージョンを識別する文字列（本例で
は「V1」）を付与している。範囲種別15241に「仕様差分」が設定されているテスト
ケース生成条件をテストケース生成条件情報152が含む場合、テストケース生成装置
10は、変更前のルール仕様151のルール仕様論理式も生成する。図4の場合、テスト
ケース生成装置10は「RS1V1 RS2V1」というルール仕様論理式を生成する。

【0079】

図7に戻り、以上の処理の後、テストケース生成装置10は、テストケース生成条件情
報152からテストケース生成条件を一つ選択し（S717）、テストケース生成条件情
報152の各テストケース生成条件についてテストケースを生成する（S718）。この
処理（以下、テストケース生成処理S718と称する。）の具体的な内容については後述
する。テストケース生成条件情報152の全てのテストケース生成条件についてテスト
ケースを生成すると（S719：NO）、メイン処理S700は終了する。

【0080】

尚、テストケース生成条件の内容によっては入力値の組み合わせが存在しない場合もあ
るが、入力値の組み合わせが存在するか否かは、例えば、S715等に後続して範囲条件
論理式の充足可能性を判定することにより確認することができる。即ち、範囲条件論理式
が充足可能な場合はテストケース生成条件を満たす入力値の組み合わせが存在し、範囲条
件論理式が充足不能な場合は入力値の組み合わせが存在しないと判定することができる。
そのような判定を行うことで、ユーザは無駄なテストケース生成条件が含まれていないか
を容易に知ることができる。

【0081】

図8は、図7のテストケース生成処理S718を説明するフローチャートであり、生成
方式1525に「条件網羅」が設定されているテストケース生成条件に基づくテストケー
ス生成処理S718を説明するフローチャートである。以下、同図とともに説明する。

【0082】

同図に示すように、まずテストケース生成装置10は、変更後のルール仕様151から
条件項を抽出する（S811）。尚、テストケース生成装置10は、図7のS712で読
み込んだルール仕様151のルールセットに含まれている条件項のうち、他のルールセッ
トの出力項目になっていない入力項目を含む条件項のみを抽出する。またテストケー
ス生成装置10は、同一の条件項が重複しないように条件項を抽出する。図4の例の場合、テ
ストケース生成装置10は、例えば、「Amount < 20000」、「20000 ≤
Amount」、「Amount < 50000」、「50000 ≤ Amount」、「
Amount < 100000」、「Amount ≥ 100000」、「Age ≤ 35
」、「35 < Age」、「Age < 60」、「Age ≥ 60」を条件項として抽出する。
尚、「Rank = Regular」、「Rank = Silver」など「Rank」に
関する条件項は、ルールセットRS1の出力項目に関する条件項であるため、テストケー
ス生成装置10はこれらについては抽出しない。

【0083】

続いて、テストケース生成装置10は、境界値ケース要否1526が「要」のテストケー
ス生成条件について、ルール仕様151から境界値条件を抽出する（S812）。テスト
ケース生成装置10は、例えば、S811で抽出した条件項について不等号演算子を等
号演算子（=）に置き換えることにより境界値条件を抽出する。図4の例の場合、テスト
ケース生成装置10は、例えば、条件項「Amount < 20000」から「Amount
= 20000」を、条件項「Amount < 50000」から「Amount = 500
00」を、条件項「Amount < 100000」から「Amount = 100000」
を、条件項「Age ≤ 35」から「Age = 35」を、条件項「Age < 60」から「
Age = 60」を、夫々境界値条件として抽出する。

【0084】

10

20

30

40

50

続いて、テストケース生成装置10は、テストケースを生成するための論理式であるテストケース生成用論理式を生成する(S813)。具体的には、テストケース生成装置10は、ルール仕様論理式と範囲条件論理式とを論理積で結合した論理式をテストケース生成用論理式として生成する。例えば、図4のルール仕様151に対して図5の条件ID1523が「C1」のテストケース生成条件に基づくテストケース生成用論理式は「(RS1V1 RS2V1) (RS1V2 RS2V2) (DiscountV1 DiscountV2)」となる。

【0085】

続いて、テストケース生成装置10は、生成したテストケース生成用論理式の充足可能性を判定する(S814)。テストケース生成装置10が充足可能と判定した場合(S814: YES)、処理はS815に進む。テストケース生成装置10が充足不能と判定した場合(S814: NO)、テストケース生成処理S718は終了し、その後、処理は図7のS718に進む。

【0086】

続いて、テストケース生成装置10はモデルを取得する(S815)。ここでモデルとは、与えられた論理式をTRUE(真)(充足可能)にする入力値と出力値の組み合わせのサンプルである。テストケース生成装置10は、例えば、SMTソルバ(SMT solver)の機能を利用してモデルを取得する。

【0087】

続いて、テストケース生成装置10は、S815で取得したモデルに基づき、各入力項目と出力項目の値を特定することによりテストケースを生成する(S816)。

【0088】

続いて、テストケース生成装置10は、同一のテストケースが生成されないような論理式である条件網羅制約式を生成し、生成した条件網羅制約式とテストケース生成用論理式とを論理積で結合した論理式を、新たなテストケース生成用論理式として生成する(S817)。

【0089】

具体的には、テストケース生成装置10は、取得したモデルから条件項及び境界値条件の真偽値を特定し、特定した真偽値の組み合わせが成立しないような条件網羅制約式を生成する。例えば、真(True)となった条件項または境界値条件をTermT__i、偽(False)となった条件項または境界値条件をTermF__jと表すと、テストケース生成装置10は「 $\neg \text{TermT_1} \neg \text{TermT_2} \cdots \text{TermF_1} \text{TermF_2} \cdots$ 」という条件網羅制約式を生成する。

【0090】

例えば、図6のテストケースID1531が「T101」の例では、「Age = 60」、「Amount = 100000」であることから、「 $20000 \leq \text{Amount}$ 」、「 $50000 \leq \text{Amount}$ 」、「 $\text{Amount} \geq 100000$ 」、「Amount = 100000」、「 $35 < \text{Age}$ 」、「 $\text{Age} \geq 60$ 」、「Age = 60」が真(True)となり、「Amount < 20000」、「Amount < 50000」、「Amount < 100000」、「Amount = 20000」、「Amount = 50000」、「Age <= 35」、「Age < 60」、「Age = 35」が偽(False)となる。この場合、テストケース生成装置10は、「 $\neg (20000 \leq \text{Amount}) \neg (50000 \leq \text{Amount}) \neg (\text{Amount} \geq 100000) \neg (\text{Amount} = 100000) \neg (35 < \text{Age}) \neg (\text{Age} \geq 60) \neg (\text{Age} = 60) \text{Amount} < 20000 \text{Amount} < 50000 \text{Amount} < 100000 \text{Amount} = 20000 \text{Amount} = 50000 \text{Age} <= 35 \text{Age} < 60 \text{Age} = 35$ 」という条件網羅制約式を生成する。

【0091】

S817の処理の後、処理はS814に戻り、テストケース生成装置10は、条件網羅制約式を用いて新たに生成したテストケース生成用論理式の充足可能性を判定する。そし

10

20

30

40

50

てテストケース生成装置 10 は、S 8 1 4 で充足可能と判定される間、S 8 1 4 ~ S 8 1 7 の処理を繰り返し実行する。S 8 1 4 でテストケース生成装置 10 が充足不能と判定すると、処理は図 7 の S 7 1 8 に進む。

【0092】

このように条件網羅制約式を用いることで、重複したテストケースの生成を効率よく排除することができ、テストケースを効率よく生成することができる。

【0093】

図 9 は、図 7 のテストケース生成処理 S 7 1 8 を説明するフローチャートであり、生成方式 1 5 2 5 に「パス網羅」が設定されているテストケース生成条件に基づくテストケース生成処理 S 7 1 8 を説明するフローチャートである。以下、同図とともに説明する。

10

【0094】

まずテストケース生成装置 10 は、図 8 の S 8 1 3 と同様の方法でテストケース生成用論理式を生成する (S 9 1 1)。

【0095】

続いて、テストケース生成装置 10 は、生成したテストケース生成用論理式の充足可能性を判定する (S 9 1 2)。テストケース生成装置 10 がテストケース生成用論理式を充足可能と判定した場合 (S 9 1 2 : YES)、処理は S 9 1 3 に進む。テストケース生成装置 10 がテストケース生成用論理式を充足不能と判定した場合 (S 9 1 2 : NO)、テストケース生成処理 S 7 1 8 は終了する。その後、処理は図 7 の S 7 1 8 に進む。

【0096】

20

続いて、テストケース生成装置 10 は、図 8 の S 8 1 5 と同様の方法でモデルを取得し (S 9 1 3)、取得したモデルに基づき、各入力項目と出力項目の値を特定することによりテストケースを生成する (S 9 1 4)。

【0097】

続いて、テストケース生成装置 10 は、同一のテストケースが生成されないような論理式であるパス網羅制約式を生成し、当該パス網羅制約式とテストケース生成用論理式とを論理積で結合した論理式を、新たなテストケース生成用論理式として生成する (S 9 1 5)。ここでテストケース生成装置 10 は、モデルから各ルールの条件部の真偽値を特定し、真となるルールの条件部の組み合わせが同時に真とならないようなパス網羅制約式を生成する。具体的には、テストケース生成装置 10 は、ルールセット毎に真となる条件部を特定し、これらの条件部を論理積で結合した論理式の否定をパス網羅制約式とする。尚、同一のルールセット内で複数の条件部が真となる場合は同一のルールセット内の真となった条件部を論理和で結合する。

30

【0098】

例えば、図 6 のテストケース ID 1 5 3 1 が「T 2 0 1」の例では、R S 1 R 3 C、R S 2 R 4 C、及び R S 2 R 5 C が真となる。尚、R S 1 R 3 C は、ルールセット R S 1 の ID が R 3 の条件部を意味する。R S 2 R 4 C と R S 2 R 5 C も同様の意味である。本例ではルールセット R S 2 の 2 つの条件部が真となるため、パス網羅制約式は「 $\neg (R S 1 R 3 C \vee (R S 2 R 4 \vee R S 2 R 5))$ 」となる。

【0099】

40

S 9 1 5 の処理の後、処理は S 9 1 2 に戻り、テストケース生成装置 10 は、新たなテストケース生成用論理式について充足可能性を判定する。そしてテストケース生成装置 10 は、S 9 1 2 で充足可能と判定される間、S 9 1 2 ~ S 9 1 5 の処理を繰り返し実行する。S 9 1 2 でテストケース生成装置 10 が充足不能と判定すると、処理は図 7 の S 7 1 8 に進む。

【0100】

このようにパス網羅制約式を用いた場合は重複したテストケースの生成を効率よく排除することができ、生成方式 1 5 2 5 に「パス網羅」が指定されている場合のテストケースを効率よく生成することができる。

【0101】

50

尚、生成方式 1 5 2 5 に「ルール網羅」が指定されている場合のテストケースについても「パス網羅」と同様な手法によりテストケースを効率よく生成することができる。尚、この場合の制約式（以下、ルール網羅制約式と称する。）は、モデルから各ルールの条件部の真偽値を特定し、真となるルールの条件部が偽となるように、当該真となるルールの条件部の否定を論理積で結合することにより生成することができる。

【 0 1 0 2 】

また生成方式 1 5 2 5 に「A l l P a i r s / 2 因子」や「A l l P a i r s / 3 因子」が指定されている場合には、P I C T (Pairwise Independent Combinatorial Testing tool) などオープンソースソフトウェアを利用することで容易にテストケースを生成することができる。

10

【 0 1 0 3 】

以上に説明したように、テストケース生成装置 1 0 は、ルール仕様論理式と、範囲条件及び生成方式を含むテストケース生成条件に基づく範囲条件論理式とを論理積によって結合した論理式であるテストケース生成用論理式を生成し、テストケース生成用論理式に基づきテストケースを生成するので、テストケースの生成に関する多様なニーズに柔軟に対応することができる。

【 0 1 0 4 】

またテストケース生成装置 1 0 は、テストケース生成条件の範囲種別 1 5 2 4 1 に「仕様差分」が設定されている場合、ルール仕様の変更前と変更後の差分に基づき範囲条件論理式を生成するので、ルール仕様の変更前と変更後の差分をテストするためのテストケースを効率よく生成することができる。

20

【 0 1 0 5 】

またテストケース生成装置 1 0 は、テストケース生成条件の範囲種別 1 5 2 4 1 に「入力条件」が設定されている場合、テスト範囲 1 5 2 4 2 に設定されている条件に基づく範囲条件論理式を生成するので、ニーズに対応したテストケースを効率よく生成することができる。

【 0 1 0 6 】

またテストケース生成装置 1 0 は、範囲条件論理式の間重複が有る場合は重複が無くなるように範囲条件論理式を修正するので、重複したテストケースの生成を防いで効率よくテストケースを生成することができる。

30

【 0 1 0 7 】

以上、本発明について実施の形態に基づき具体的に説明したが、本発明は上記の実施の形態に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。例えば、上記の実施の形態は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、上記実施形態の構成の一部について、他の構成の追加・削除・置換をすることが可能である。

【 0 1 0 8 】

また上記の各構成、機能部、処理部、処理手段等は、それらの一部または全部を、例えば、集積回路で設計する等によりハードウェアで実現してもよい。また上記の各構成、機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行することによりソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリやハードディスク、SSD (Solid State Drive) 等の記録装置、またはICカード、SDカード、DVD等の記録媒体に置くことができる。

40

【 0 1 0 9 】

また上記の各図において、制御線や情報線は説明上必要と考えられるものを示しており、必ずしも実装上の全ての制御線や情報線を示しているとは限らない。例えば、実際にはほとんど全ての構成が相互に接続されていると考えてもよい。

【 0 1 1 0 】

また以上に説明したテストケース生成装置 1 0 の各種機能部、各種処理部、各種データ

50

ベースの配置形態は一例に過ぎない。各種機能部、各種処理部、各種データベースの配置形態は、テストケース生成装置 10 が備えるハードウェアやソフトウェアの性能、処理効率、通信効率等の観点から最適な配置形態に変更し得る。

【0111】

また前述した各種データベースの構成（スキーマ（Schema）等）は、リソースの効率的な利用、処理効率向上、アクセス効率向上、検索効率向上等の観点から柔軟に変更し得る。

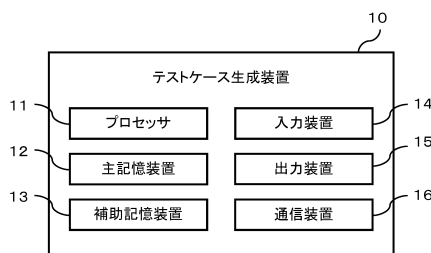
【符号の説明】

【0112】

10 テストケース生成装置、110 テストケース設定編集部、111 ルール仕様論理式生成部、112 範囲条件論理式生成部、113 テストケース生成用論理式生成部、114 テストケース生成部、150 情報記憶部、151 ルール仕様、152 テストケース生成条件情報、1521 テスト対象、1522 テスト種別、1524 範囲条件、15241 範囲種別、15242 テスト範囲、1525 生成方式、1526 境界値ケース要否、1527 優先度、153 テストケース、RS1, RS2 ルールセット、S700 メイン処理、S718 テストケース生成処理

10

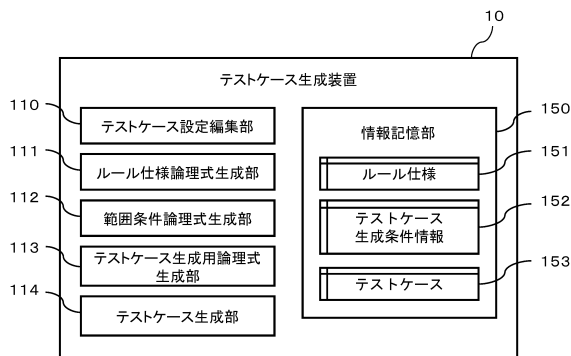
【図 1】



【図 3】

ルール仕様		
RS1: ランク判定		
ID	条件部	結果部
R1	Amount < 20000	Rank = Regular
R2	20000 <= Amount < 50000	Rank = Silver
R3	Amount >= 50000	Rank = Gold
RS2: 割引率判定		
ID	条件部	結果部
R1	Age<=35 AND Rank<>Gold	Discount = 5
R2	35<Age<60 AND Rank=Regular	Discount = 10
R3	35<Age<60 AND Rank=Silver	Discount = 15
R4	Age>=60	Discount = 20
R5	Rank = Gold	Discount = 20

【図 2】



【図 4】

バージョン: V1

RS1: ランク判定

ID	条件部	結果部
R1	Amount < 20000	Rank = Regular
R2	20000 <= Amount < 50000	Rank = Silver
R3	Amount >= 50000	Rank = Gold

RS2: 割引率判定

ID	条件部	結果部
R1	Age <= 35 AND Rank < Gold	Discount = 5
R2	35 < Age < 60 AND Rank = Regular	Discount = 10
R3	35 < Age < 60 AND Rank = Silver	Discount = 15
R4	Age >= 60	Discount = 20
R5	Rank = Gold	Discount = 20

変更

バージョン: V2

RS1: ランク判定

ID	条件部	結果部
R1	Amount < 20000	Rank = Regular
R2	20000 <= Amount < 50000	Rank = Silver
R3	50000 <= Amount < 100000	Rank = Gold
R4	Amount >= 100000	Rank = Platinum

RS2: 割引率判定

ID	条件部	結果部
R1	Age <= 35 AND (Rank = Regular OR Rank = Silver)	Discount = 5
R2	35 < Age < 60 AND Rank = Regular	Discount = 10
R3	35 < Age < 60 AND Rank = Silver	Discount = 15
R4	Age >= 60 AND Rank < Platinum	Discount = 20
R5	Rank = Gold	Discount = 20
R6	Rank = Platinum	Discount = 25

【図 5】

テストケース生成条件情報 152

1521	1522	テスト対象	RS2
		テスト種別	結合

	1523	15241	1524	15242	1525	1526	1527
条件ID	範囲条件		生成方式	境界値ケース		優先度	
	範囲種別	テスト範囲		要否			
C1	仕様差分	変更有	条件網羅	要	100		
C2	入力条件	Age > 60	パス網羅	—	80		
C3	Default	—	ルール網羅	—	—		

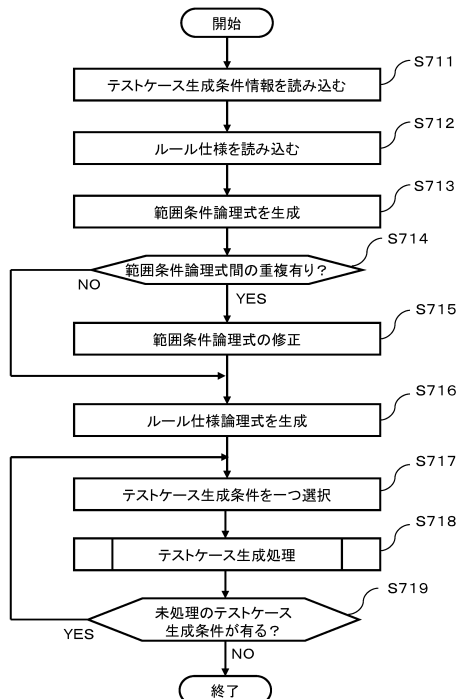
【図 6】

テストケース 153

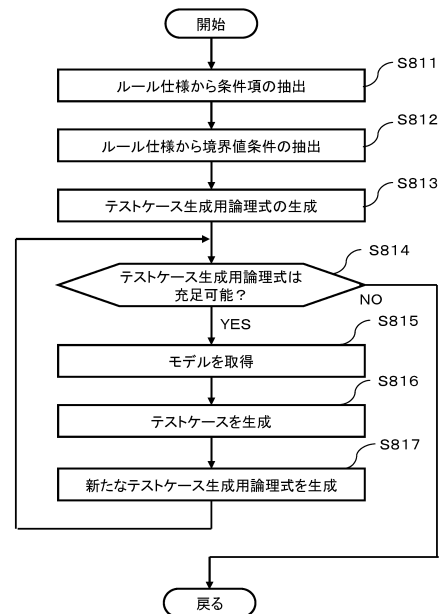
テストケースID	条件ID	入力条件		期待結果
		Age	Amount	
T101	C1	60	100000	25
T102	C1	61	100001	25
T103	C1	60	100001	25
T104	C1	61	100000	25
T105	C1	59	100001	25
T106	C1	59	100000	25
T107	C1	34	100001	25
T108	C1	34	100000	25
T109	C1	35	100000	25
T110	C1	35	100001	25
T201	C2	61	99999	20
T202	C2	61	19999	20
T203	C2	61	20000	20
T301	C3	0	19999	5
T302	C3	60	50000	20
T303	C3	59	49999	15
T304	C3	59	19999	10

【図 7】

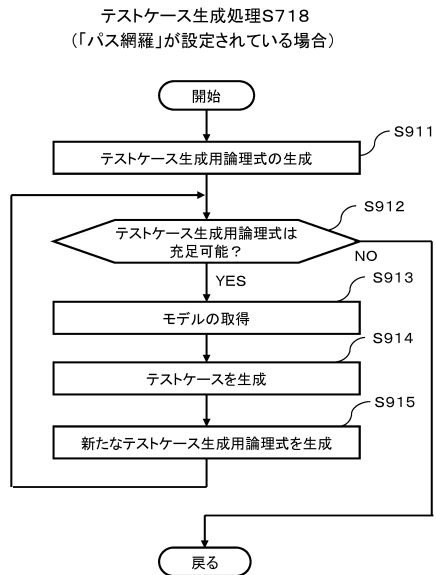
メイン処理 S700



【図 8】

テストケース生成処理 S718
(「条件網羅」が設定されている場合)

【図 9】



フロントページの続き

- (72)発明者 明神 智之
東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
- (72)発明者 松尾 努
東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
- (72)発明者 平田 久也
東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内

審査官 杉浦 孝光

- (56)参考文献 特開2013-148968(JP,A)
特開2016-057969(JP,A)
特開2015-204065(JP,A)
特開2014-186407(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 11/36