

(12) 发明专利申请

(10) 申请公布号 CN 103019830 A

(43) 申请公布日 2013. 04. 03

(21) 申请号 201210593588. 7

(22) 申请日 2012. 12. 31

(71) 申请人 深圳 TCL 新技术有限公司

地址 518052 广东省深圳市南山区中山园路
1001 号 TCL 国际 E 城科技大厦 D4 栋 7
楼

(72) 发明人 李宴儒

(74) 专利代理机构 深圳市世纪恒程知识产权代
理事务所 44287

代理人 胡海国

(51) Int. Cl.

G06F 9/46 (2006. 01)

G06F 11/07 (2006. 01)

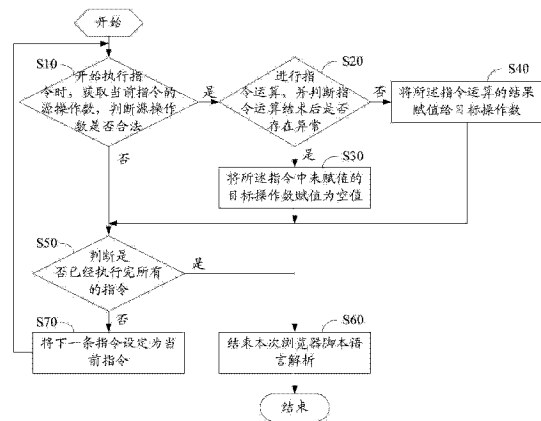
权利要求书 2 页 说明书 6 页 附图 4 页

(54) 发明名称

实现浏览器脚本语言执行容错的方法及装置

(57) 摘要

本发明公开了一种实现浏览器脚本语言执行容错的方法及装置,所述实现浏览器脚本语言执行容错的方法包括以下步骤:A. 开始执行指令时,获取当前指令的源操作数,判断源操作数是否合法;若是,则执行步骤B,否则执行步骤E;B. 进行指令运算,并判断指令运算结束后是否存在异常;若是,则执行步骤C,否则执行步骤D;C. 将所述指令中未赋值的目标操作数赋值为空值;并执行步骤E;D. 将所述指令运算的结果赋值给目标操作数;并执行步骤E;E. 判断是否已经执行完所有的指令;若是,则结束本次浏览器脚本语言解析,否则将下一条指令设定为当前指令,并返回执行步骤A。



1. 一种实现浏览器脚本语言执行容错的方法,其特征在于,包括以下步骤:

A,开始执行指令时,获取当前指令的源操作数,判断源操作数是否合法;若是,则执行步骤B,否则执行步骤E;

B,进行指令运算,并判断指令运算结束后是否存在异常;若是,则执行步骤C,否则执行步骤D;

C,将所述指令中未赋值的目标操作数赋值为空值;并执行步骤E;

D,将所述指令运算的结果赋值给目标操作数;并执行步骤E;

E,判断是否已经执行完所有的指令;若是,则结束本次浏览器脚本语言解析,否则将下一条指令设定为当前指令,并返回执行步骤A。

2. 如权利要求1所述的实现浏览器脚本语言执行容错的方法,其特征在于,在执行所述步骤A之前还包括:

将从服务器端下载含有脚本语言程序的文件解码为相应的脚本语言字符串;

根据词法和语法分析解码后的脚本语言字符串,生成字节码。

3. 如权利要求1所述的实现浏览器脚本语言执行容错的方法,其特征在于,所述步骤C具体包括:

C1,判断所述指令中是否包含目标操作数;若是,则执行步骤C2,否则执行步骤E;

C2,判断所述目标操作数是否被赋值;若是,则执行步骤E,否则执行步骤C3;

C3,将所述目标操作数赋值为空值,并执行步骤E。

4. 如权利要求1至3中任一项所述的实现浏览器脚本语言执行容错的方法,其特征在于,所述判断源操作数是否合法具体为:判断源操作数是否为空。

5. 如权利要求1至3中任一项所述的实现浏览器脚本语言执行容错的方法,其特征在于,所述指令运算产生的异常为:源操作数对应的属性或方法未找到,或者源操作数的类型或范围不符合指令的要求。

6. 一种实现浏览器脚本语言执行容错的装置,其特征在于,包括:

第一判断模块,用于在开始执行指令时,获取当前指令的源操作数,判断源操作数是否合法;

第二判断模块,用于在第一判断模块判断所述源操作数合法时,进行指令运算,并判断指令运算结束后是否存在异常;

处理模块,用于在第二判断模块判断指令运算存在异常时,将所述指令中未赋值的目标操作数赋值为空值;在第二判断模块判断指令运算未存在异常时,将所述指令运算的结果赋值给目标操作数;

第三判断模块,用于在第一判断模块判断所述源操作数不合法或者处理模块赋值操作完成时,判断是否已经执行完所有的指令;若是,则结束本次浏览器脚本语言解析,否则将下一条指令设定为当前指令,并由第一判断模块继续执行获取当前指令的源操作数,判断源操作数是否合法的操作。

7. 如权利要求6所述的实现浏览器脚本语言执行容错的装置,其特征在于,所述实现浏览器脚本语言执行容错的装置还包括:

解码模块,用于将从服务器端下载含有脚本语言程序的文件解码为相应的脚本语言字符串;

分析模块,用于根据词法和语法分析解码后的脚本语言字符串,并生成字节码。

8. 如权利要求 6 所述的实现浏览器脚本语言执行容错的装置,其特征在于,所述处理模块包括:

第一判断单元,用于判断所述指令中是否包含目标操作数;

第二判断单元,用于在第一判断单元判断指令中包含目标操作数时,判断所述目标操作数是否被赋值;

处理单元,用于在第二判断单元判断所述目标操作数未被赋值时,将所述目标操作数赋值为空值,并由所述第三判断模块继续执行判断是否已经执行完所有的指令的操作。

9. 如权利要求 6 至 8 中任一项所述的实现浏览器脚本语言执行容错的装置,其特征在于,所述第一判断模块用于判断源操作数是否为空。

10. 如权利要求 6 至 8 中任一项所述的实现浏览器脚本语言执行容错的装置,其特征在于,所述指令运算产生的异常为:源操作数对应的属性或方法未找到,或者源操作数的类型或范围不符合指令的要求。

实现浏览器脚本语言执行容错的方法及装置

技术领域

[0001] 本发明涉及浏览器技术领域,特别涉及一种实现浏览器脚本语言执行容错的方法及装置。

背景技术

[0002] 脚本语言是一种解释型语言(如 JavaScript),一般以文本形式保存,其源代码在发往客户端运行之前不需要经过编译,而是将文本格式字符代码发送给浏览器由浏览器解释运行。

[0003] 在脚本语言中,如果一条语句运行出现错误,那么下面的语句也无法继续运行。在用网页浏览器浏览网页时,通常出现浏览器提示网页错误而无法正常使用现象,这种现象大多数是由于浏览器对网页中脚本语言文件执行解析时出现错误导致的。对脚本语言文件执行解析时出现的错误包括较小、次要的语法错误以及严重的语法错误或者逻辑错误,其中较小或次要的语法错误是指出现错误的程序片段对网页的主要逻辑、主要功能不会产生重要影响(即该程序段的执行结果对整体程序影响不大或者没有影响)。因此为降低网页的出错率,需要在网页中出现较小或次要的脚本语言语法错误的情况下,仍然能够保证网页的正常浏览。

发明内容

[0004] 本发明的主要目的在于提供一种实现浏览器脚本语言执行容错的方法,旨在防止网页中出现的较小或次要的脚本语言语法错误而影响网页的正常浏览。

[0005] 为了实现发明目的,本发明提供一种实现浏览器脚本语言执行容错的方法,该实现浏览器脚本语言执行容错的方法包括以下步骤:

[0006] A,开始执行指令时,获取当前指令的源操作数,判断源操作数是否合法;若是,则执行步骤 B,否则执行步骤 E;

[0007] B,进行指令运算,并判断指令运算结束后是否存在异常;若是,则执行步骤 C,否则执行步骤 D;

[0008] C,将所述指令中未赋值的目标操作数赋值为空值;并执行步骤 E;

[0009] D,将所述指令运算的结果赋值给目标操作数;并执行步骤 E;

[0010] E,判断是否已经执行完所有的指令;若是,则结束本次浏览器脚本语言解析,否则将下一条指令设定为当前指令,并返回执行步骤 A。

[0011] 优选地,在执行所述步骤 A 之前还包括:

[0012] S60,将从服务器端下载含有脚本语言程序的文件解码为相应的脚本语言字符串;

[0013] S70,根据词法和语法分析解码后的脚本语言字符串,并生成字节码。

[0014] 优选地,所述步骤 C 具体包括:

[0015] C1,判断所述指令中是否包含目标操作数;若是,则执行步骤 C2,否则执行步骤 E;

[0016] C2,判断所述目标操作数是否被赋值;若是,则执行步骤 E,否则执行步骤 C3;

[0017] C3,将所述目标操作数赋值为空值,并执行步骤 E。

[0018] 优选地,所述判断源操作数是否合法具体为:判断源操作数是否为空。

[0019] 优选地,所述指令运算产生的异常为:源操作数对应的属性或方法未找到,或者源操作数的类型或范围不符合指令的要求。

[0020] 本发明还提供一种实现浏览器脚本语言执行容错的装置,该实现浏览器脚本语言执行容错的装置包括:

[0021] 第一判断模块,用于在开始执行指令时,获取当前指令的源操作数,判断源操作数是否合法;

[0022] 第二判断模块,用于在第一判断模块判断所述源操作数合法时,进行指令运算,并判断指令运算结束后是否存在异常;

[0023] 处理模块,用于在第二判断模块判断指令运算存在异常时,将所述指令中未赋值的目标操作数赋值为空值;在第二判断模块判断指令运算未存在异常时,将所述指令运算的结果赋值给目标操作数;

[0024] 第三判断模块,用于在第一判断模块判断所述源操作数不合法或者处理模块赋值操作完成时,判断是否已经执行完所有的指令;若是,则结束本次浏览器脚本语言解析,否则将下一条指令设定为当前指令,并由第一判断模块继续执行获取当前指令的源操作数,判断源操作数是否合法的操作。

[0025] 优选地,所述实现浏览器脚本语言执行容错的装置还包括:

[0026] 解码模块,用于将从服务器端下载含有脚本语言程序的文件解码为相应的脚本语言字符串;

[0027] 分析模块,用于根据词法和语法分析解码后的脚本语言字符串,并生成字节码。

[0028] 优选地,所述处理模块包括:

[0029] 第一判断单元,用于判断所述指令中是否包含目标操作数;

[0030] 第二判断单元,用于在第一判断单元判断指令中包含目标操作数时,判断所述目标操作数是否被赋值;

[0031] 处理单元,用于在第二判断单元判断所述目标操作数未被赋值时,将所述目标操作数赋值为空值,并由所述第三判断模块继续执行判断是否已经执行完所有的指令的操作。

[0032] 优选地,所述第一判断模块用于判断源操作数是否为空。

[0033] 优选地,所述指令运算产生的异常为:源操作数对应的属性或方法未找到,或者源操作数的类型或范围不符合指令的要求。

[0034] 本发明通过获取指令的源操作数,首先判断该源操作数的合法性,并在源操作数不合法的情况下直接跳过当前指令直接进入下一条指令的执行,在源操作数合法的情况下,判断执行指令结束后是否存在异常,若存在异常,则将未赋值的目标操作数赋值为空值,其次抛出异常执行下一条指令;若不存在异常,则将指令运算后的结果赋值给目标操作数,其次执行下一条指令。因此本发明可有效防止浏览器解析网页中的脚本语言文件时出现的较小或次要的脚本语言语法错误而影响网页的正常浏览,从而提高了实用性。

附图说明

- [0035] 图 1 为本发明实现浏览器脚本语言执行容错的方法一实施例的流程示意图；
- [0036] 图 2 为本发明实现浏览器脚本语言执行容错的方法另一实施例的流程示意图；
- [0037] 图 3 为本发明实现浏览器脚本语言执行容错的方法又一实施例的流程示意图；
- [0038] 图 4 为本发明实现浏览器脚本语言执行容错的装置一实施例的结构示意图；
- [0039] 图 5 为图 4 中处理模块的结构示意图。
- [0040] 本发明目的的实现、功能特点及优点将结合实施例，参照附图做进一步说明。

具体实施方式

[0041] 应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0042] 参照图 1，图 1 为本发明实现浏览器脚本语言执行容错的方法一实施例的流程示意图。本实施例提供的实现浏览器脚本语言执行容错的方法包括以下步骤：

[0043] 步骤 S10，开始执行指令时，获取当前指令的源操作数，判断源操作数是否合法；若是，则执行步骤 S20，否则执行步骤 S50；

[0044] 本实施例中，在执行指令时，首先获取指令的源操作数，然后判断源操作数的合法性。具体地，本实施例中，可通过判断源操作数是否为空值，从而判断指令的源操作数是否合法。若源操作数为空值，则判定源操作数不合法；若源操作数不为空值，则判定源操作数合法。例如，可通过调用一空值判断函数判断源操作数的合法性。

[0045] 步骤 S20，进行指令运算，并判断指令运算结束后是否存在异常；若是，则执行步骤 S30，否则执行步骤 S40；

[0046] 步骤 S30，将所述指令中未赋值的目标操作数赋值为空值，并执行步骤 S50；

[0047] 步骤 S40，将所述指令运算的结果赋值给目标操作数，并执行步骤 S50；

[0048] 本实施例中，在确定指令的源操作数合法后，将进行指令运算，若运算出现错误，则会生成对应的异常，该异常包含错误信息，并被保存至一全局变量中。然后通过调用一异常检测函数查询异常信息，从而判定在进行指令运算的过程中是否出现错误。若存在异常，则将指令中未赋值的目标操作数赋值为空值；若不存在异常，则将指令运算的结果赋值给目标操作数。应当说明的是，本实施例中，上述指令运算产生的异常具体为在操作指令运算时，未找到源操作数对应的属性或方法，或者源操作数的类型或范围不符合指令的要求。

[0049] 步骤 S50，判断是否已经执行完所有的指令；若是，执行步骤 S60，否则执行步骤 S70；

[0050] 步骤 S60，结束本次浏览器脚本语言解析；

[0051] 步骤 S70，将下一条指令设定为当前指令，并返回执行步骤 S10。

[0052] 当执行完上述步骤 S30 或 S40 或者在步骤 S10 中判断源操作数本身不合法时，则表示已经执行完一条指令，此时需将下一条指令设定为当前指令，再次执行当前指令。当所有的指令均执行完成后，才完成一次浏览器脚本语言解析。

[0053] 本发明通过获取指令的源操作数，首先判断该源操作数的合法性，并在源操作数不合法的情况下直接跳过当前指令直接进入下一条指令的执行，在源操作数合法的情况下，判断执行指令结束后是否存在异常，若存在异常，则将未赋值的目标操作数赋值为空值，其次抛出异常执行下一条指令；若不存在异常，则将指令运算后的结果赋值给目标操作

数,其次执行下一条指令。因此本发明可有效防止浏览器解析网页中的脚本语言文件时出现的较小或次要的脚本语言语法错误而影响网页的正常浏览,从而提高了实用性。

[0054] 参照图 2,图 2 为本发明实现浏览器脚本语言执行容错的方法另一实施例的流程示意图。基于上述实施例,本实施例中,上述在执行上述步骤 S10 之前还包括:

[0055] S80,将从服务器端下载含有脚本语言的文件解码为相应的脚本语言字符串;

[0056] S90,根据词法和语法分析解码后的脚本语言字符串,生成字节码。

[0057] 应当说明的是,上述字节码是由操作码和操作数组成的指令序列;浏览器解析器将按顺序逐一执行字节码的每一条指令。

[0058] 参照图 3,图 3 为本发明实现浏览器脚本语言执行容错的方法又一实施例的流程示意图。基于上述实施例,本实施例中,上述步骤 S30 具体包括:

[0059] 步骤 S31,判断指令中是否包含目标操作数;若是,则执行步骤 S32,否则执行步骤 S50;

[0060] S32,判断目标操作数是否被赋值;若是,则执行步骤 S50,否则执行步骤 S33;

[0061] S33,将目标操作数赋值为空值,并执行步骤 S50。

[0062] 本实施例中,在指令运算结束后存在异常情况下,首先判断当前执行的执行中是否包含目标操作数,若不存在目标操作数,则判断是否已经执行完所有的指令,若是,则结束本次浏览器脚本语言解析;若否,则将下一条指令设定为当前指令,并返回执行步骤 S10。在存在目标操作数的情况下,判断目标操作数是否被赋值,若目标操作数已经被赋值,则判断是否已经执行完所有的指令,若是,则结束本次浏览器脚本语言解析;若否,则将下一条指令设定为当前指令,并返回执行步骤 S10;若目标操作数未被赋值,则将目标操作数赋值为空值,然后判断是否已经执行完所有的指令,若是,则结束本次浏览器脚本语言解析;若否,则将下一条指令设定为当前指令,并返回执行步骤 S10。

[0063] 本发明还提供一种实现浏览器脚本语言执行容错的装置,用于实现上述方法。参照图 4,图 4 为本发明实现浏览器脚本语言执行容错的装置一实施例的结构示意图。本实施例提供的实现浏览器脚本语言执行容错的装置包括:

[0064] 第一判断模块 100,用于在开始执行指令时,获取当前指令的源操作数,判断源操作数是否合法;

[0065] 本实施例中,在执行指令时,首先由第一判断模块 100 获取指令的源操作数,并判断源操作数的合法性。具体地,本实施例中,可通过判断源操作数是否为空值,从而判断指令的源操作数是否合法。若源操作数为空值,则判定源操作数不合法;若源操作数不为空值,则判定源操作数合法。例如,可通过调用一空置判断函数判断源操作数的合法性。

[0066] 第二判断模块 200,用于在第一判断模块 100 判断所述源操作数合法时,进行指令运算,并判断指令运算结束后是否存在异常;

[0067] 处理模块 300,用于在第二判断模块 200 判断指令运算存在异常时,将所述指令中未赋值的目標操作数赋值为空值;在第二判断模块 200 判断指令运算未存在异常时,将所述指令运算的结果赋值给目标操作数;

[0068] 第三判断模块 400,用于在第一判断模块 100 判断所述源操作数不合法或者处理模块 300 赋值操作完成时,判断是否已经执行完所有的指令;若是,则结束本次浏览器脚本语言解析,否则将下一条指令设定为当前指令,并由第一判断模块 100 继续执行获取当前

指令的源操作数,判断源操作数是否合法的操作。

[0069] 本实施例中,在第一判断模块 100 判定指令的源操作数合法后,将由第二判断模块 200 进行指令运算,若运算出现错误,则会生成对应的异常,该异常包含错误信息,并被保存至一全局变量中。然后通过调用一异常检测函数查询异常信息,从而判定在进行指令运算的过程中是否出现错误。若存在异常,则由处理模块 300 将指令中未赋值的目標操作数赋值为空值,并由第三判断模块 400 执行判断是否已经执行完所有的指令的操作;若不存在异常,则由处理模块 300 将指令运算的结果赋值给目标操作数,并由第三判断模块 400 执行判断是否已经执行完所有的指令的操作。应当说明的是,本实施例中,上述指令运算产生的异常具体为在操作指令运算时,未找到源操作数对应的属性或方法,或者源操作数的类型或范围不符合指令的要求。在第一判断模块 100 判定指令的源操作数不合法后,则由第三判断模块 400 执行判断是否已经执行完所有的指令的操作。

[0070] 本发明通过获取指令的源操作数,首先有第一判断模块 100 判断该源操作数的合法性,并在源操作数不合法的情况下直接跳过当前指令直接进入下一条指令的执行,在源操作数合法的情况下,由第二判断模块 200 判断执行指令结束后是否存在异常,若存在异常,则由处理模块 300 将未赋值的目標操作数赋值为空值,其次抛出异常执行下一条指令;若不存在异常,则由处理模块 300 将指令运算后的结果赋值给目标操作数,其次执行下一条指令。因此本发明可有效防止浏览器解析网页中的脚本语言文件时出现的较小或次要的脚本语言语法错误而影响网页的正常浏览,从而提高了实用性。

[0071] 进一步地,上述实现浏览器脚本语言执行容错的装置还包括:

[0072] 解码模块 500,用于将从服务器端下载含有脚本语言程序的文件解码为相应的脚本语言字符串;

[0073] 分析模块 600,用于根据词法和语法分析解码后的脚本语言字符串,生成字节码。

[0074] 应当说明的是,上述字节码是由操作码和操作数组成的指令序列;浏览器解析器将按顺序逐一执行字节码的每一条指令。当第一判断模块 100 获取到指令的源操作数后,第一判断模块 100 将判断所获取的源操作数的合法性。

[0075] 结合参照图 5,图 5 为图 4 中处理模块的结构示意图。基于上述实施例,本实施例中,上述处理模块 300 包括:

[0076] 第一判断单元 301,用于判断所述指令中是否包含目标操作数;

[0077] 第二判断单元 302,用于在第一判断单元 301 判断指令中包含目标操作数时,判断所述目标操作数是否被赋值;

[0078] 处理单元 303,用于在第二判断单元判断所述目标操作数未被赋值时,将所述目标操作数赋值为空值,并由所述第三判断模块 400 继续执行判断是否已经执行完所有的指令的操作。

[0079] 本实施例中,通过第二判断模块 200 判断在指令运算结束后存在异常情况下,首先由于第一判断单元 301 判断当前执行的执行中是否包含目标操作数,若不存在目标操作数,则由所述第三判断模块 400 继续执行判断是否已经执行完所有的指令的操作。在存在目标操作数的情况下,由第二判断单元 302 判断目标操作数是否被赋值,若目标操作数已经被赋值,则由所述第三判断模块 400 继续执行判断是否已经执行完所有的指令的操作;若目标操作数未被赋值,则由处理单元将目标操作数赋值为空值,然后由所述第三判断模

块 400 继续执行判断是否已经执行完所有的指令的操作。

[0080] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

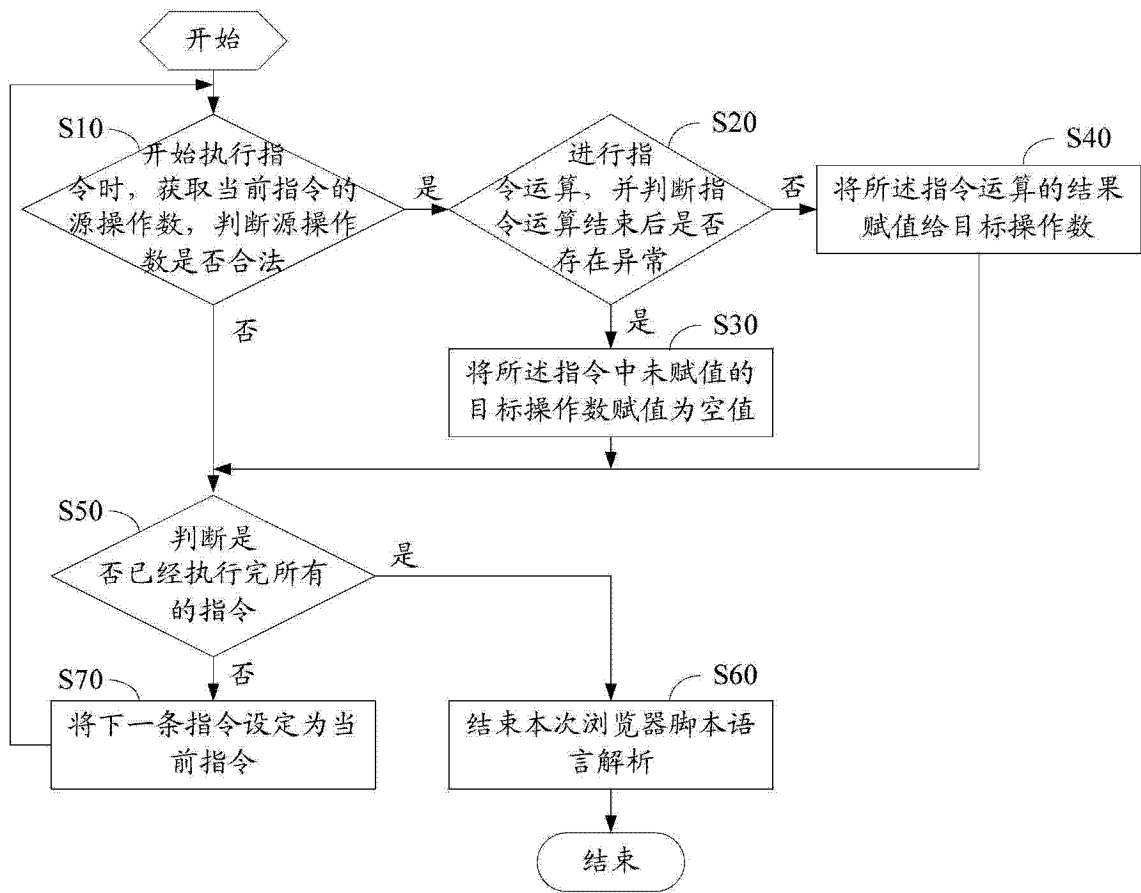


图 1

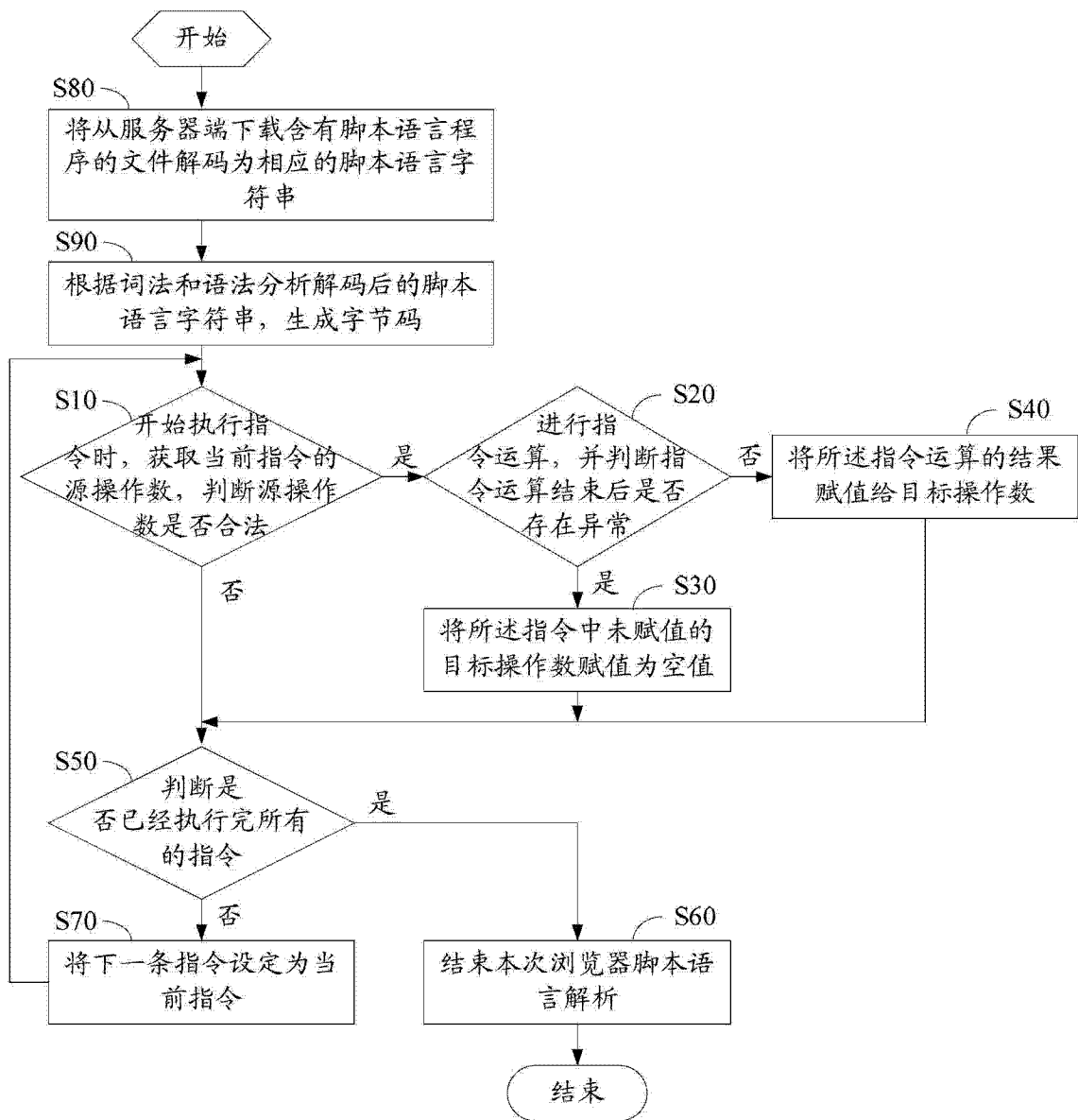


图 2

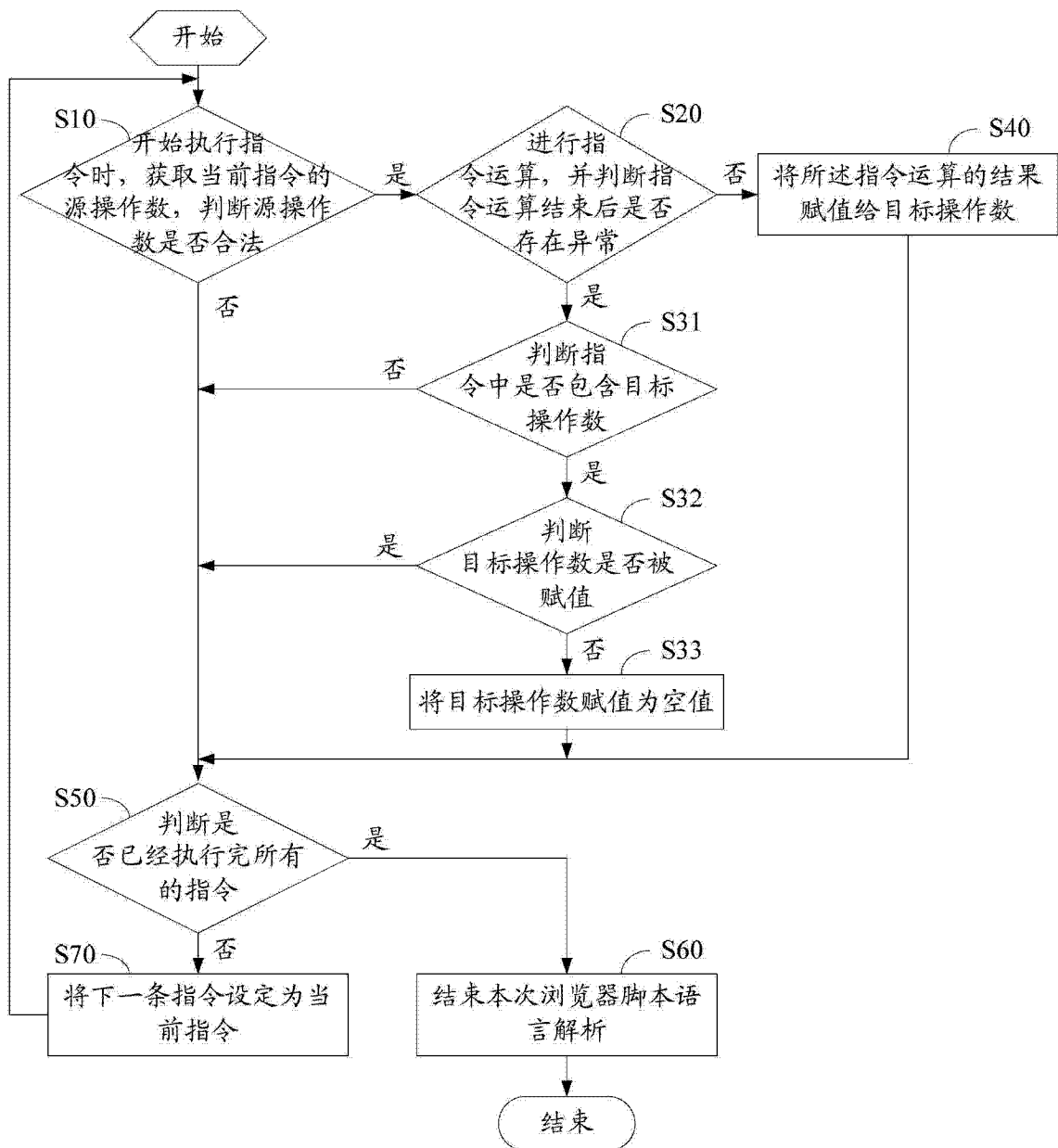


图 3

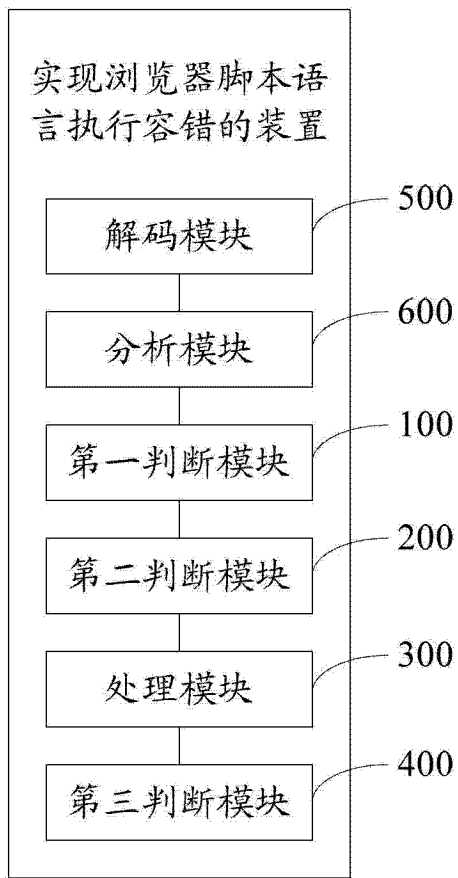


图 4

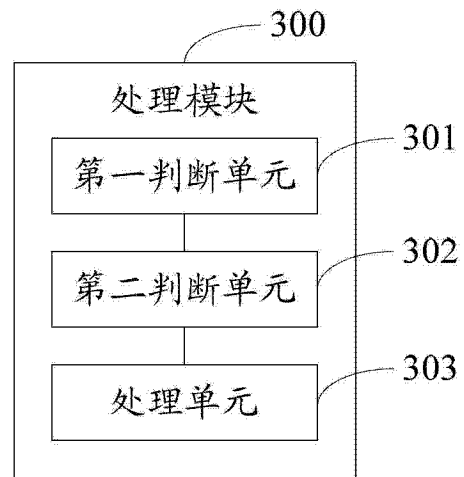


图 5