

基于 DFA 和笛卡尔集的数据录入机器人设计与实现

刘家兴¹, 刘金锋², 李捷²

(1. 河南省实验中学, 河南 郑州 450000;
2. 河南大学, 河南 开封 475000)

摘要: 信息化时代, 大量的数据处理、数据操作、数据再加工是生产、工作中必不可少的一项基本任务, 提高大量数据处理的自动化程度, 有助于提高社会生产力。在计算理论中, 确定有限状态自动机 (deterministic finite automation, DFA) 是一个能实现状态转移的自动机, 它能根据事先给定的转移函数自动转移到下一个状态, 而笛卡尔集 (Cartesian Set) 定义了两个集合的操作集。文中基于确定有限状态自动机 (DFA) 理论和笛卡尔集构建了一种数据自动处理机器人模型, 并以财政支付数据自动录人为切入点, 设计和实现了财政支付数据录入机器人。通过这种机器人, 2 个员工一天的录入工作量, 可以用 1 个小时来完成, 极大提高了工作效率; 同时, 人工录入的准确率很难达到 100%, 而数据自动录入机器人可以达到无差错录入。

关键词: 自动机理论; 笛卡尔集; 握手认证; 数据自动处理机器人

中图分类号: TP39

文献标识码: A

文章编号: 1673-629X(2019)12-0178-06

doi: 10.3969/j.issn.1673-629X.2019.12.032

Design and Implementation of Automatic Data Processing Robot Based on DFA and Cartesian Set

LIU Jia-xing¹, LIU Jin-feng², LI Jie²

(1. Henan Experimental High School, Zhengzhou 450000, China;
2. Henan University, Kaifeng 475000, China)

Abstract: A large number of data processing, data operation and data reprocessing are indispensable tasks in production and work with the development of information age. Enhancement of the degree of automatic data processing is conducive to improving social productivity. Deterministic finite automation (DFA) is an automaton that can realize state automatic transition according to the given transfer function, while Cartesian set defines the operation set of two sets in computational theory. In this paper, an automatic data processing robot model is constructed based on DFA theory and Cartesian set. Furthermore, a data automatic inputting robot for financial payment system is designed and implemented. Through this kind of robot, two employees can input the workload in one day, which can be completed in one hour, greatly improving the work efficiency. Meanwhile, the accuracy of manual input is difficult to reach 100%, while the robot input can achieve error-free input.

Key words: DFA; Cartesian set; handshake authentication; automatic data processing robot

0 引言

近年来, 科学技术的迅猛发展, 尤其是计算机技术的发展, 极大便利了人们的工作及生活。再加上人们工作的需求及生活水平的提升, 计算机用户数量呈全球性的爆炸式增长, 同时产生了大量数据信息有待处理。这就使得人们对计算机的计算能力需求不断提高^[1]。针对提高数据处理、数据操作、数据再加工的能

力的问题, 文中对数据自动化处理展开研究。

自动机理论是一种将离散数学系统的构造、作用和关系作为研究对象的数学理论。自动机是有限状态机 (FSM) 的数学模型, FSM 是给定符号输入, 依据转移函数“跳转”过一系列状态的一种机器, 自动机状态有开始状态和结束状态^[2]。DFA 在自动化、智能化领域有着广泛的应用。例如, 文献[3]提出的汉字自动

收稿日期: 2019-01-08

修回日期: 2019-05-09

网络出版时间: 2019-06-27

基金项目: 国家自然科学基金 (61170243)

作者简介: 刘家兴 (2001-), 男, 研究方向为集合论; 刘金锋, 硕士, 讲师, 研究方向为软件工程; 李捷, 博士, 教授, 研究方向为移动代理、计算机网络。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190627.1115.084.html>

机模型能刻画出文字智能输入、词句输入,为汉字的智能处理提供数据模型;文献[4]利用自动机的状态转换表实现串的快速自动匹配;文献[5]利用层次式时间自动机模型自动转化为模型检测工具 UPPAAL 的输入;文献[6]以时间自动机为建模工具,分别为要检测和控制的物理实体以及不同种类的物联网服务独立建模,这些时间自动机形成了一个网络,刻画完整的物联网服务通信并发过程;针对系统多层嵌套式结构范畴模型的动态行为描述和分析问题,文献[7]提出一种基于端口自动机的行为表达方法;文献[8]将节点调度问题转化为集合覆盖问题后,利用学习自动机的特性自适应地更新所选取感知方向的概率;文献[9]将自动机模型与遗传算法相结合,使得遗传算法的每一代遗传因子自动产生。文中研究的是数据自动化、智能化处理问题,与自动机理论的本质特征切合,所以基于自动机理论构建了一种数据自动处理机器人模型。另外,设置了自动机的启动条件即握手认证,握手认证模型基于笛卡尔集。笛卡尔集是指在数学中,两个集合 X 和 Y 的笛卡尔集(Cartesian Set)表示为 $X \times Y$,第一个对象是 X 的成员,第二个对象是 Y 的所有可能有序对的其中一个成员^[10]。

依据数据自动处理机器人模型,文中设计和实现了某财政支付系统数据自动录入机器人。由于某财政支付系统和某财务系统是两套独立的软件系统,这就造成了在财政支付环节存在一个脱节,即财政系统无法有效获取并使用这些结算信息。目前采用人工方式实现结算信息逐条录入,工作效率低下,已不能满足未来工作的需要。而文中设计的财政支付数据自动录入机器人将会把财务系统中的结算信息搬运到财政支付系统,有效地解决这个“脱节”问题,使得结算支付信息得到充分利用,极大地提高工作效率,使得财政支付流程更顺畅。

1 机器人模型

1.1 数据自动操作模型

确定有限状态自动机^[11] Γ :由一个非空有限的状态集合 Ω ,一个输入字母表(非空有限的字符集合) Σ ,一个转移函数 $\delta: \Omega \times \Sigma \rightarrow \Omega$,一个开始状态 $s \in \Omega$,一个结束状态的集合 $F \subseteq \Omega$ 所组成的 5-元组,因此一个 DFA 可以写成这样的形式: $\Gamma = (\Omega, \Sigma, \delta, s, F)$ 。

数据自动操作模型构建步骤如下:

(1) 信息元集合 U 。

$$U = \{u_1, u_2, \dots, u_n\}, n \in N$$

(2) 集合 U 划分。

等价类的划分基于以下定理:设 R 是定义在集合

A 上的等价关系,那么 R 的等价类构成 S 的划分。

$$R = \{r_1, r_2, \dots, r_m\}, m \in N$$

$$S_i = \{x, y | x r_i y, x \in U, y \in U, r_i \in R\}, i \leq m, i \in N$$

(3) 信息序列集合 LQ。

信息序列为 M -元组记为 Q :

$$Q = \langle u_1, u_2, \dots, u_i, \dots, u_m \rangle$$

信息序列集合 LQ 为:

$$LQ = \{ \langle u_1, u_2, \dots, u_i, \dots, u_m \rangle | u_i \in S_i \}$$

(4) 信息序列状态集合。

信息元操作状态因子集合为:

$$A = \{a, -a\}$$

带操作的信息序列 QA 为:

$$QA = \langle u_1 \oplus a_j, u_2 \oplus a_j, \dots, u_i \oplus a_j, \dots, u_m \oplus a_j \rangle$$

带操作的信息序列集合 LQA 为:

$$LQA = \{ \langle u_1 \oplus a_j, u_2 \oplus a_j, \dots, u_i \oplus a_j, \dots, u_m \oplus a_j \rangle | u_i \in S_i, a_j \in A \}$$

可得: $|LQA| = 2^M$ 。

带依次操作的信息序列 QAS_k 为:

$$QAS_k = \langle u_1 \oplus a_j, u_2 \oplus a_j, \dots, u_k \oplus a_j, \dots, u_m \oplus a_j \rangle$$

$$a_j = \begin{cases} a, j < k \\ -a, j \geq k \end{cases}$$

带依次操作的信息序列集合 LQAS 为:

$$LQAS = \{QAS_k\}, k \leq M$$

可得: $|LQAS| = M$ 。

(5) 操作集合 POpt。

位置集合 P :

$$P = \{ \langle x, y \rangle | x, y \in N, x \leq W, y \leq H \}, W, H \text{ 为自然数常量;}$$

操作元集合 Opt 为:

$$Opt = \{Opt_1, Opt_2, \dots, Opt_i, \dots, Opt_N\}$$

$$POpt = \{p \wedge opt | p \in R, opt \in Opt\}$$

POpt 反映了对某位置的某种操作。

(6) 转移函数 δ 。

$$f(QA_k, \text{popt}) = \{ \langle u_1 \oplus a_j, u_2 \oplus a_j, \dots, u_k \oplus a_j, \dots, u_m \oplus a_j \rangle | a_k = -a, u_i \in S_i, a_j \in A \}, \text{popt} \subseteq POpt, \text{popt} \neq \emptyset$$

$f(QA_k, \text{popt})$ 定义了 Ω 集合的状态转移规则,读入操作 popt ,跳转的状态不唯一,所以该函数定义的是一种不确定的有穷状态自动机(NFA)。

$f(QAS_k, \text{popt})$ 定义了 Ω 集合的另一种状态转移规则,读入操作 popt ,跳转的状态唯一,所以该函数定义的是一种确定的有穷状态自动机(DFA)。

(7) 起始状态与终止状态。

$$s = \langle u_1 \oplus - a, u_2 \oplus - a, \dots, u_k \oplus - a, \dots, u_M \oplus - a \rangle$$

$$F = \langle u_1 \oplus a, u_2 \oplus a, \dots, u_k \oplus a, \dots, u_M \oplus a \rangle$$

综述,基于 FA 理论,文中建立的自动操作模型描述如下,非依次操作的自动机模型 AOM:

AOM = {LQA, POpt, f(QA_k, popt), s, F}, AOM 是 NFA;

依次操作的自动机模型 AOM-S:

AOM - S = {LQAS, POpt, f(QAS_k, popt), s, F}, AOM - S 是 DFA。

1.2 笛卡尔集握手认证模型

该模型的设计基于笛卡尔集理论,用于设计自动机的启动条件,握手成功自动机启动,模型构建步骤如下:

(1) 标识因子集合。

$$T = \{t_1, t_2, \dots, t_i, \dots, t_n\}$$

(2) 标识因子关系。

$$R = \{R_1, R_2\}$$

(3) 标识因子划分。

$$T_1 = \{x, y | xR_1y, x \in T, y \in T\}$$

$$T_2 = \{x, y | xR_2y, x \in T, y \in T\}$$

其中 T₁、T₂ 为不同类别的标识。

(4) 握手认证模型。

$$A = \{ \langle t, U, f \rangle \mid t \in T_1, U \in T_2 \}$$

$$B = \{ \langle s, V, g \rangle \mid s \in T_2, V \in T_1 \}$$

$$A \times B = \{ \langle a, b \rangle \mid a \in A, b \in B, f(b_1) \in U, g(a_1) \in V \}$$

其中 b₁、a₁ 分别为元素(三元组)中的第 1 元数据, A × B 即为可握手认证集合。

2 财政数据自动录入机器人设计

基于数据自动操作模型和笛卡尔集握手认证模型,设计财政数据自动录入机器人。该设计以一系列操作组合作为自动机模型的输入字母表,以数据操作后的状态为有限状态集合,开始状态定义为数据的原始状态,即未对数据进行任何操作,结束状态定义为数据的完全操作态,即对数据的所有操作均已完成。该设计中操作的数据对象为批量财政结算数据,包含账号名称、账户账号、开户行信息、摘要、金额等信息元;针对以上数据进行的操作为在某位置的单击、复制、粘贴等。初始状态为数据的原始状态,结束状态为数据的完成操作态。

2.1 财政数据自动录入机器人

财政数据录入机器人定义为: AOM - S = {LQAS, POpt, f(QAS_k, popt), s, F}, 其中, LQAS =

{QAS_k}, k ≤ M, M = 5; QAS_k = < u₁ ⊕ a_j, u₂ ⊕ a_j, ... , u_s ⊕ a_j >, 其中, u_i ∈ S_i, i ≤ M, a_j ∈ A

具体的:

S₁: 账户名称;

S₂: 账户账号;

S₃: 开户行信息;

S₄: 摘要;

S₅: 金额;

A = {已处理, 未处理}。

且 QAS_k 满足 a_j = $\begin{cases} \text{已处理}, j < k \\ \text{未处理}, j \geq k \end{cases}$ 。

Popt = {p ∧ opt | p ∈ P, opt ∈ OPT}, 其中:

P = { < x, y > | x, y ∈ N, x ≤ W, y ≤ H }, W、H 为屏幕分辨率。

Opt = {复制, 单击, 粘贴}。

$$f(QAS_k, popt) = QAS_{k+1}, popt \subseteq POpt, popt \neq \emptyset。$$

s = < u₁ ⊕ 未处理, u₂ ⊕ 未处理, ... , u_i ⊕ 未处理, ... , u_s ⊕ 未处理 >, 其中 u_i ∈ S_i。

F = < u₁ ⊕ 已处理, u₂ ⊕ 已处理, ... , u_i ⊕ 已处理, ... , u_s ⊕ 已处理 >

其中 u_i ∈ S_i, S_i 中 i ≤ 5。

2.2 握手认证设计

启动财政数据自动录入机器人的程序定义为机器人唤醒系统,握手认证模型定义如下:

$$A \times B = \{ \langle a, b \rangle \mid a \in A, b \in B, f(b_1) \in U, g(a_1) \in V \}$$

$$A = \{ \langle t, U, f \rangle \mid t \in T_1, U \in T_2 \}$$

其中, T₁ 为机器人唤醒系统身份标识集; T₂ 为财政数据自动录入机器人身份标识集; A 为机器人唤醒系统集合; t 为机器人唤醒系统身份标识; f 为握手规则; U 为 f 规则下符合条件的财政数据自动录入机器人身份标识。

$$B = \{ \langle s, V, g \rangle \mid s \in T_2, V \in T_1 \}$$

其中, B 为财政数据自动录入机器人集合; s 为财政数据自动录入机器人身份标识; g 为握手规则; V 为 g 规则下符合条件的机器人唤醒系统身份标识。

f(b₁) ∈ U, g(a₁) ∈ V 为笛卡尔集计算规则。

3 方案实现

功能总体设计:插入财政支付数据录入机器人,主机机器人唤醒系统唤醒机器人程序,机器人启动,载入数据,完成财政支付信息的自动录入;拔出机器人,机器人进程随之关闭。

实现平台: .NET FrameWork 4.0^[12];

实现语言: C#^[13];

运行环境:Windows XP、Win7 及以上。

财政支付数据录入机器人的设计包含两部分程序:主机唤醒程序和机器人程序。主机唤醒程序通过握手认证唤醒机器人。文中采用面向对象^[14]的思想对两部分程序进行实现。

3.1 主机唤醒程序

主机唤醒程序用于通过握手认证模型识别、启动财政支付数据录入机器人,工作流程如图 1 所示。

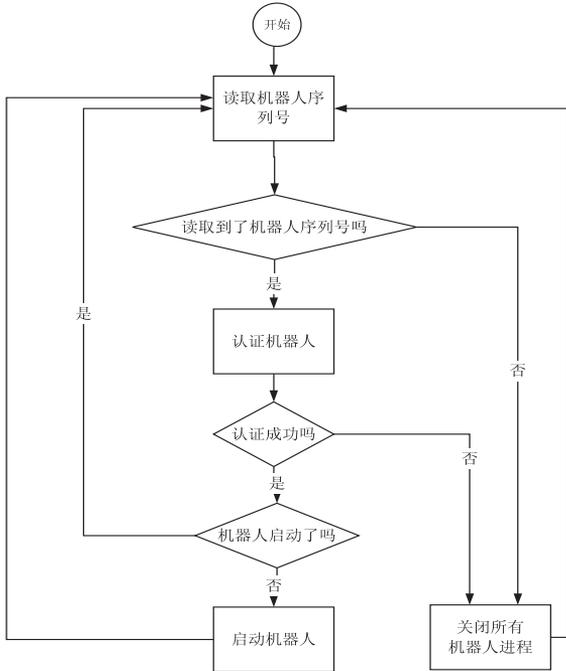


图 1 握手认证流程

根据工作流程设计类图,如图 2 所示。

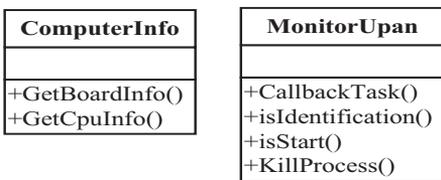


图 2 主机唤醒程序类图

图中,ComputerInfo 类用于获取机器人唤醒系统身份标识;MonitorUpan 类用于检测是否有机器人插入、与机器人握手认证、启动机器人。其中 isIdentification() 方法实现认证机器人功能; CallbackTask() 方法是后台线程^[15] 执行函数,根据认证结果启动或关闭机器人; isStart() 方法判断是否机器人启动; KillProcess() 方法实现关闭机器人功能。

核心代码如下:

```
public bool isIdentification()//认证
{
    bool flag=false;
    var scope=new ManagementScope();
    var query = new SelectQuery(" select * from win32_logi-
    caldisk");
```

```
using( var searcher = new ManagementObjectSearcher( scope,
query))
{
    //返回设备列表
    var collection = searcher. Get();
    foreach( var obj in collection)
    {
        foreach ( var property in obj. Properties)//读取机器人序列号
        {
            if(property. Value! = null)//读取到了吗
            {
                //认证机器人
                if( property. Name == " VolumeSerialNumber" && property.
                Value. ToString()= = "3C2382B3")
                {
                    flag = true;
                }
            }
        }
    }
    return flag;
}

public void CallbackTask( Object stateInfo)//机器人启动或
关闭
{
    while( true)
    {
        if(isIdentification())//如果认证成功
        {
            //启动机器人
            if(isStart()= =false)
            {
                ProcessStartInfo startInfo = new ProcessStartInfo( @" Robot.
                exe" );
                ComputerInfo cmp=new ComputerInfo();
                startInfo. Arguments=cmp. GetCpuInfo()+cmp. GetBoardInfo
                ();
                Process. Start( startInfo);
            }
        }
        else
        {
            //如果认证失败,关闭机器人
            KillProcess( " Robot" );
        }
    }
}
```

主机唤醒程序建立后台线程,后台线程执行 CallbackTask() 方法,即实现如下功能:与插入的机器人握手认证,认证成功,则启动机器人;如果认证不成功,则

关闭所有机器人进程。

3.2 机器人程序

财政支付数据自动录入机器人功能如下:读取数据,对于每一条数据的信息元,找到操作位置,完成相应操作,提交数据,修改数据状态为已处理,当所有数据的状态为已处理时,机器人停止。

机器人核心类图如图 3 所示。

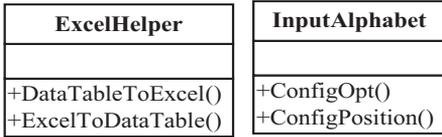


图 3 机器人核心类图

图中,ExcelHelper 类用于读取数据;InputAlphabet 类用于设置自动机的输入字母表,完成状态跳转。

机器人自动录入流程如图 4 所示。

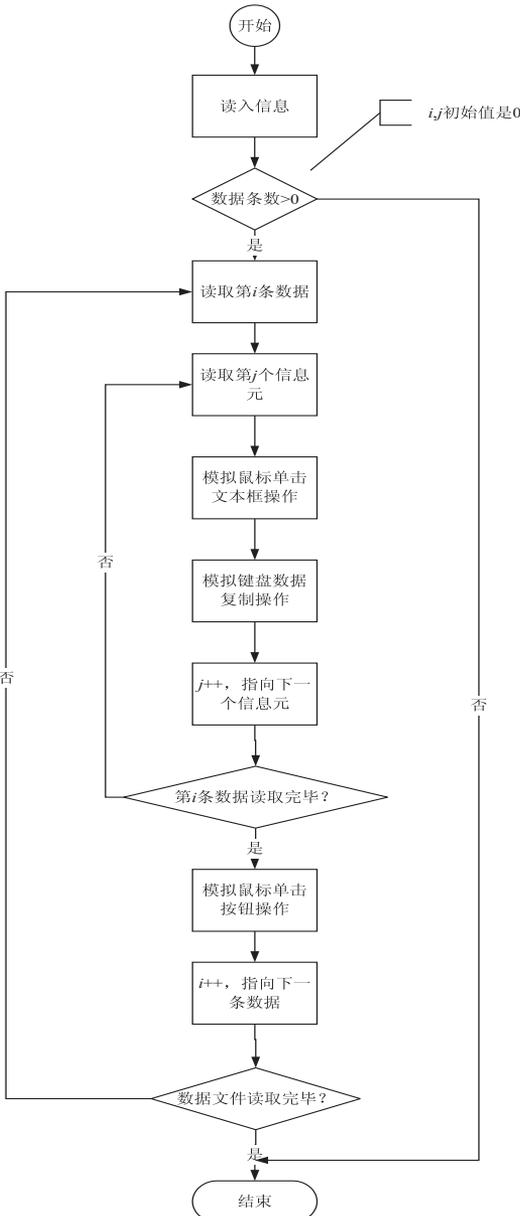


图 4 机器人自动录入流程

核心代码如下:

```

public void ConfigOpt()
{
    if (totalTable != null)
    {
        if (rowIndex < totalTable.Rows.Count)
        {
            label2.Text = rowIndex.ToString() + "/" + (totalTable.Rows.Count - 1);
            for (inti = 0; i < arr.Length; i++)
                // 选定第 i 个位置
                {
                    if (i == 2)
                    {
                        this.DoMouseDoubleClick(arr[i][0], arr[i][1]); // 双击操作
                        Thread.Sleep(50);
                    }
                    else
                    {
                        this.DoMouseClicked(arr[i][0], arr[i][1]); // 单击操作
                        Thread.Sleep(50);
                        keybd_event(17, 0, 0, 0);
                        Thread.Sleep(50);
                        keybd_event((byte)Keys.A, 0, 0, 0);
                        Thread.Sleep(50);
                        keybd_event(17, 0, 2, 0);
                        Thread.Sleep(50);
                        keybd_event((byte)Keys.A, 0, 2, 0);
                        // 全选操作
                        keybd_event(46, 0, 0, 0);
                        Thread.Sleep(50);
                        keybd_event(46, 0, 2, 0);
                        // 删除操作
                        Thread.Sleep(50);
                    }
                    try // 复制操作
                    {
                        Clipboard.Clear();
                        Clipboard.SetText(totalTable.Rows[rowIndex][i].ToString());
                        // 第 rowIndex 条数据的第 i 个信息元
                    } catch (Exception ee)
                    {
                    }
                    Application.DoEvents();
                    Clipboard.SetText(totalTable.Rows[rowIndex][i].ToString());
                }
                Thread.Sleep(50);
                keybd_event(17, 0, 0, 0);
                Thread.Sleep(50);
            }
        }
    }
}

```

```

keybd_event((byte)Keys.V,0,0,0);
Thread.Sleep(50);
keybd_event(17,0,2,0);
Thread.Sleep(50);
keybd_event((byte)Keys.V,0,2,0);
//粘贴操作
Thread.Sleep(100);
}
rowIndex++;
}
if(rowIndex == totalTable.Rows.Count)
{
timer1.Enabled = false;
rowIndex++;
}

```

4 实验对比

文中设计的财政支付机器人于 2018 年 11 月投入使用,2018 年 11 月之前采用人工录入,人工录入安排 2 人,各占用 1 台 PC,PC 配置如下:CPU 为 Intel(R) Core(TM) i3 CPU M350@2.27GHz;内存为 4 G;操作系统为 Win7 旗舰版(32 位)。

现有方式为机器人录入,1 台机器人录入需要一台 PC 支撑,PC 配置同上。表 1 是 2017 年度和 2018 年度 12 月份数据对比。

表 1 数据对比

日期	录入数据量(笔)	录入方式	录入合计用时	录入效率	准确率/%
2017 年 12 月 18 日	450	人工录入 2 人	8 小时 * 75% * 2 人	0.59 笔/分钟	98.6
2017 年 12 月 19 日	509	人工录入 2 人	8 小时 * 75% * 2 人 * 1.5 天	0.47 笔/分钟	98
2017 年 12 月 20 日	566	人工录入 2 人	8 小时 * 75% * 2 人 * 1.5 天	0.52 笔/分钟	98.2
2018 年 12 月 18 日	427	机器人 1 个	72 分钟	6 笔/分钟	100
2018 年 12 月 19 日	526	机器人 1 个	88 分钟	6 笔/分钟	100
2018 年 12 月 20 日	571	机器人 1 个	96 分钟	6 笔/分钟	100

注:数据源自某大学财务系统;8 小时 * 75% 为人工录入有效用时;准确率来自对账后的数据统计。

通过以上数据对比发现,人工录入速率约为 2 分钟一条数据,机器人录入速率约为一分钟 6 笔,机器人录入效率远远大于人工录入效率;人工录入准确率很难达到 100%,而机器人录入可以达到无差错录入。

5 结束语

文中基于确定有限状态自动机(DFA)理论和笛卡尔集构建了一种数据自动处理机器人模型,依据该模型,设计了财政支付数据录入机器人。基于.NET Framework 4.0,对财政支付数据录入机器人进行了实现。通过这种机器人,2 个员工一天的录入工作量,可以用 1 个小时来完成,极大提高了工作效率。

参考文献:

- [1] 梁凌宇. 大数据时代计算机信息处理技术[J]. 信息与电脑,2016(18):139-140.
- [2] 蒋宗礼,姜守旭. 形式语言与自动机理论[M]. 北京:清华大学出版社,2013.
- [3] 蔡增玉,刘书如,张建伟,等. 汉字模糊有穷自动机的研究[J]. 计算机技术与发展,2008,18(3):89-91.
- [4] 陈倩. 一种基于有限自动机的快速串匹配算法[J]. 计算机技术与发展,2009,19(1):131-133.
- [5] 周宇,胡军,葛季栋. 一种层次式时间自动机模型检测方法[J]. 计算机应用与软件,2012,29(11):48-51.

- [6] 李力行,金芝,李戈. 基于时间自动机的物联网服务建模和验证[J]. 计算机学报,2011,34(8):1365-1377.
- [7] 薛岗,张云春,刘笛,等. 面向嵌入式系统结构的端口自动机行为表达方法[J]. 计算机应用,2016,36(7):2016-2020.
- [8] 李明,胡江平. 基于学习自动机的异构有向传感器节点调度算法[J]. 计算机工程,2018,44(8):199-203.
- [9] ZHANG Yueqin, CHEN Shiyong, LIU Jinfeng, et al. The application of genetic algorithm in vehicle routing problem [C]//International symposium on electronic commerce and security. Guangzhou, China: IEEE, 2008.
- [10] 张清华,蒲兴成,尹邦勇,等. 离散数学及其应用[M]. 北京:清华大学出版社,2016.
- [11] 吴哲辉. 形式语言与自动机理论[M]. 北京:机械工业出版社,2007.
- [12] GOLDSHTEIN S. NET 性能优化[M]. 姚琪琳,刘夏,译. 北京:人民邮电出版社,2018.
- [13] MICHAELIS M, LIPPERT E. C#本质论[M]. 周靖,庞燕,译. 北京:人民邮电出版社,2017.
- [14] BOOCH G, MAKSIMCHUK R A, ENGEL M W, et al. 面向对象分析与设计[M]. 王海鹏,潘加宇,译. 第 3 版. 北京:人民邮电出版社,2009.
- [15] 马骏. C#网络应用编程[M]. 第 3 版. 北京:人民邮电出版社,2014.