

一种通信有限状态机的被动测试及其错误诊断

赵保华^{1,2}, 张炜^{1,2}, 林华辉^{1,2}, 李培龙^{1,2}

(1. 中国科学技术大学计算机科学与技术系, 230027, 合肥; 2. 安徽省计算与通讯软件重点实验室, 230027, 合肥)

摘要: 针对现有的被动测试方法无法解决嵌入式通信系统的错误检测问题, 提出了一种被动测试错误检测方法. 该方法在外部通道放置观察者, 并使用了全局状态描述系统的当前可能状态, 利用了观察到的输入输出信息, 同时结合了模型转换推测出系统的状态变迁. 基于该检测方法又设计出一种层次化的错误诊断算法, 首先记录被动检测过程中系统经历的转换轨迹, 用变异分析方法生成候选诊断, 然后通过被动观察来降低错误集合的规模, 最后使用区分序列和交叉判定进行错误定位. 通过一个实际的 H. 245 通信协议的实验表明, 所提算法能够在有限观察步骤内检测出被测系统中的错误, 并能有效定位检测错误, 其回溯算法也可以显著缩短错误症状序列的长度.

关键词: 被动测试; 错误定位; 通信有限状态机

中图分类号: TN915.04 **文献标识码:** A **文章编号:** 0253-987X(2007)06-0640-05

Passive Testing and Fault Diagnosis for Communication Finite State Machines

Zhao Baohua^{1,2}, Zhang Wei^{1,2}, Lin Huahui^{1,2}, Li Peilong^{1,2}

(1. Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China;
2. Anhui Province Key Laboratory of Software in Computing and Communication, Hefei 230027, China)

Abstract: Focusing on the problem that the existing methods of passive testing can not accomplish the fault detection of embedded communication systems, a passive testing approach to fault detection is proposed, in which an observer is placed in external channel, global states are used to describe the possible states of the system, the observed input and output information is employed and combined with model conversion to presume the system state transitions. Based on the approach, a hierarchical method of fault diagnosis is further designed, in which the system conversion trace are recorded in passive testing, then mutant analysis method is utilized to generate fault candidates. The size of fault set can be reduced by passive observations, and distinguishing sequence and cross verification are used to locate the fault. Experiments of a real protocol H. 245 shows that the proposed approach can detect the fault in embedded communication systems in limited steps and can locate the system fault effectively. Its backtracing algorithm can reduce the length of the fault symptom sequence significantly.

Keywords: passive testing; fault location; communication finite state machine

协议一致性测试旨在检查协议实现是否同协议规范一致, 按照测试序列的生成方式划分, 可以分为主动测试和被动测试. 主动测试能够通过外部控制

被测系统的输入, 因而能保证较高的测试覆盖率, 状态跟踪和错误定位也比较容易, 但是它的测试序列长度同模型复杂程度有关, 在被测系统的状态和转

换较多时,生成测试序列就比较困难.被动测试通过被动观察获取输入输出序列,无须干扰网络设备的正常运行,适用于运行中的网络故障检查和定位.现有的一致性测试序列生成方法未能针对所有的异常,所以有一些错误仍无法被检测^[1],另外还存在一些特殊环境使得主动测试无法开展.

运用被动观察获取的系统输入输出信息可以检测、诊断系统错误,目前已有的若干解决方案包括基于有限状态机(FSM)模型的被动测试错误检测算法^[2],利用通信FSM模型对网络建模并基于该模型进行错误管理^[3],在扩展FSM模型上做被动测试错误检测以解决数据流错误检测问题^[4],使用前向后向轨迹追踪进行错误标识^[5]以及解决FSM模型的被动测试错误诊断问题^[6]等.这类被动测试模型均假设协议实体间的消息通道是可观察的,而在嵌入式通信系统中,协议实体间的通信常常是不可见的或是部分可见的,所以此类方法不适用于嵌入式系统.

为此,本文引入了一种适用于嵌入式通信系统的被动测试模型ECFSM,并提出了该模型下的被动测试错误检测算法及相应的错误诊断方法.

1 ECFSM 模型

本文描述的被测系统是一个嵌入式通信系统,可以用ECFSM模型描述. ECFSM由多个确定有限状态机(DFSM)构成,每个DFSVM可以表示为 $M_i = (I, O, S, s_0, \delta, \lambda)$,其中 I, O, S 分别是输入符号集合、输出符号集合和系统状态集合, s_0 为系统初始状态, $\delta: S \times I \rightarrow S$ 是状态转移函数, $\lambda: S \times I \rightarrow O$ 是输出函数, i 表示系统中第 i 个DFSVM, $i \in [0, n]$, n 为系统中DFSVM的数量. M_0 为上下文状态机,其余的均为内嵌状态机.该模型满足以下条件.

(1)上下文状态机的外部端口与外界环境通信,外部环境输入符号集合为 E_I ,输出符号集合为 E_O ,上下文状态机可进行外部、内部转换.内嵌状态机只能进行内部转换,并可通过内部端口与上下文状态机通信.内嵌状态机的输入为 I_1^{0-i} ,它来自上下文状态机内部转换的输出;输出为 I_0^{i-0} ,可作为上下文状态机外部转换的输入.

(2)系统的输入和输出是同步的,因为状态机的内部转换将导致发生外部转换.

(3)假设被测系统最多有一个实现错误,且错误只可能是转移错误或输出错误^[7].

ECFSM被动测试框架如图1所示,其中IUT

是一个ECFSM,并假设 $n=2$ (为了便于讨论,以下均假设 $n=2$). M_0 和 M_1 的通信是不可观察的, M_1 是内嵌状态机,它的输入输出通道不可与外界交互,而 M_0 是上下文状态机,能够与外界环境通信,观察点选取在外部通道上. ECFSM模型与文献^[3]模型的区别在于:后者的DFSVM是对称的,DFSVM间的通信是外部可见的.

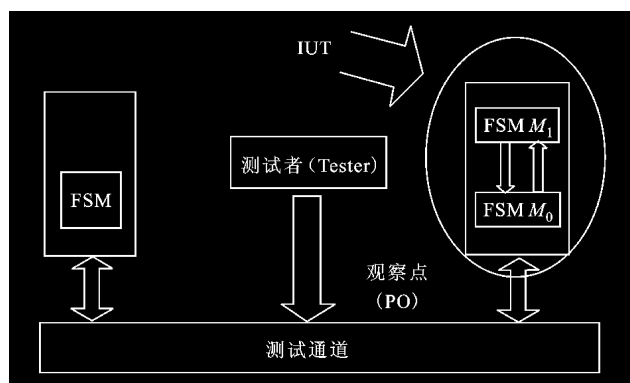


图1 ECFSM的被动测试框架

2 错误检测算法

当进行被动测试时,系统的状态是不确定的,但随着测试的不断进行,系统当前可能的状态数在不断减少,若系统只剩下惟一的可能状态,便完成了被动引导^[2]过程,而错误检测过程开始,如果系统可能的状态数变为0,则视为检测到错误.受到此思想的启发,鉴于模型中状态机间的通信是不可见的,本文检测算法选择全局状态来描述整个被测系统的状态.系统的全局状态记为 (s_j^0, s_j^1) , s_j^0, s_j^1 分别表示上下文状态机和嵌入式状态机的状态.测试一开始,系统的全局可能的状态为状态机状态集合的笛卡儿集.被测系统的输入输出全部通过外界环境获取,上下文状态机面临外界输入时分为下面3种情况完成触发.

(1) M_0 中存在以状态 s_j^0 为头状态, i_k 为输入的外部转换,转换输出为 o_k ,则 s_j^0 状态变迁, s_j^1 不变.

(2) M_0 中存在以状态 s_j^0 为头状态, i_k 为输入的内部转换,该情况下会导致部分观察缺失, s_j^0 的状态需要变迁2次,条件是 s_j^1 能够顺利变迁,变迁能够引发 s_j^0 的第2次变迁,且变迁的输出结果为 o_k .

(3)不满足情况(1)、(2)的情形,这时可以将该状态从 L 中删除.

不断地观察,当系统的全局可能的状态数减为0时,错误检测算法结束.一个ECFSM的状态转换

图如图2所示,并以此为例,观察序列 $x_2/y_2, x_1/y_1, x_1/y_2, x_1/y_2$, 若算法检测到错误,则称该序列为错误症状序列^[6]. 错误检测算法描述如下.

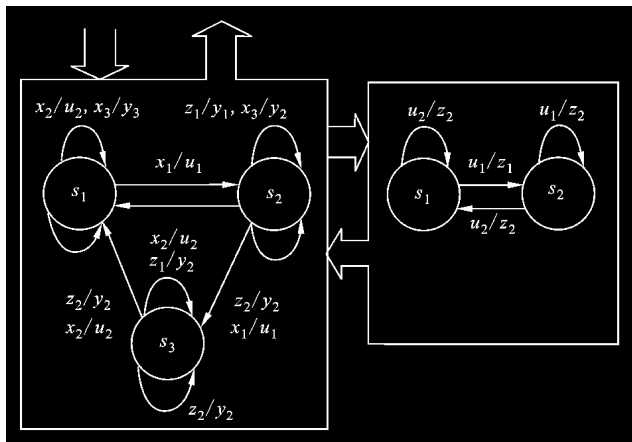


图2 一个 ECFSM 的状态转换图

算法 基于 ECFSM 的被动测试错误检测算法.

算法输入: $i_1/o_1, i_2/o_2, \dots$, ECFSM.

算法输出: 如果检测出错误, 返回 FAULT.

- (1) $L = S^0 \times S^1$
- (2) for ($k=1$ to ∞)
- (3) for ($j=1$ to n_{k-1})
- (4) if ($\lambda^0(s_j^0, i_k) \equiv o_k$)
- (5) $s_j^0 = \delta^0(s_j^0, i_k)$
- (6) else if ($\lambda^0(s_j^0, i_k) \in I_1^{0-1}$)
- (7) if ($\lambda^0(\delta^0(s_j^0, i_k), \lambda^1(s_j^1, \lambda^0(s_j^0, i_k))) \equiv o_k$)
- (8) $s_j^1 = \delta^1(s_j^1, \lambda^0(s_j^0, i_k))$
- (9) $s_j^0 = \delta^0(\delta^0(s_j^0, i_k), \lambda^1(s_j^1, \lambda^0(s_j^0, i_k)))$
- (10) else
- (11) remove (s_j^0, s_j^1)
- (12) else
- (13) remove (s_j^0, s_j^1)
- (14) endfor
- (15) if ($|L| \equiv 0$) return FAULT
- (16) endfor

3 错误诊断算法

3.1 初始诊断集合生成方法

假设错误检测算法在第 m 步检测到系统错误, 由于系统只有一个实现错误, 所以被测系统在 m 次转换过程中肯定经历了该错误, 错误的头状态只能在可能的状态集中, 而输入输出存在于症状序列中, 转换过程中的系统可能的状态信息可以通过被动检测过程获取, 再结合系统状态机模型和可能的错误

特征就可以获得候选诊断集合. 假设在第 k ($0 < k \leq m$) 步之前, 系统当前可能的状态集合为 $L^{k-1}, (s_j^0, s_j^1) \in L^{k-1}$.

情形 1 M_0 规范中存在以状态 s_j^0 为头状态、 i_k 为输入的外部转换.

当转换的输出为 o_k 时, 系统有可能包含状态 s_j^0 为头状态、 i_k/o_k 为输入输出的外部转换错误, 将 $s_j^0 \xrightarrow{i_k/o_k} s$ 作为转换错误添加到诊断集合中, 其中 $s \in S^0, s \neq \delta^0(s_j^0, i_k)$; 当转换的输出不为 o_k 时, 系统只可能包含以 s_j^0 为头状态、 i_k/o_k 为输入输出的输出错误, 将 $s_j^0 \xrightarrow{i_k/o_k} \delta^0(s_j^0, i_k)$ 作为输出错误添加到诊断集合中.

情形 2 M_0 规范中存在以状态 s_j^0 为头状态、 i_k 为输入的内部转换.

状态 s_j^0 会触发 3 个转换, 转换 1 的输入和转换 3 的输出是给定的, 3 个转换都有可能存在错误. 转换 1 存在于 M_0 之中, 对于输出错误, 输出需要遍历符号集合 I_1^{0-1} , 而对于转移错误, 结束状态需要遍历集合 S^0 ; 转换 2 发生在 M_1 之中, 可能的错误需要分别遍历 I_0^{1-0} 和 S^1 ; 转换 3 发生在 M_0 之中, 相应的转移错误需要遍历 S^0 . 该方法生成的可能错误须满足 $\lambda^0(\delta^0(s_j^0, i_k), \lambda^1(s_j^1, \lambda^0(s_j^0, i_k))) \equiv o_k$.

以上面的错误症状序列为例, 按照初始诊断集合生成方法得到系统的错误诊断集合包含 $(s_3^0, z_2/y_2, s_1^0), (s_3^0, z_2/y_2, s_2^0), (s_1^0, x_2/u_1, s_1^0)$ 等 22 个元素, 该集合的大小与 L^k 的规模以及错误症状序列长度有关. 在观察被动检测的过程中, 如果观察时机选择不当, 会导致症状序列过长, 此时利用后向迹追踪可以缩短序列长度. 该方法假设最后一步系统的当前可能的状态为所有状态, 从后向前进行逆向推导, 直到状态集合为空. 这样, 可以得出上例第 1 个观察到的输入输出 x_2/y_2 是多余的, 剔除多余的输入输出, 则初始诊断集合只包含 13 个元素.

3.2 候选诊断集合生成方法

初始诊断集合是根据每一步系统可能的状态, 再结合可能错误的特征得出的, 并没有考虑整个测试序列, 所以还应对该集合作进一步筛选, 即如果系统引入某个错误能够解释症状序列, 则仅依赖当前的序列信息是无法排除该错误的, 还应实施候选诊断集合生成方法, 步骤如下.

步骤 1: 从初始诊断集合中选取一个元素, 并用该元素产生系统的变异.

步骤2: 调用2.1节的错误检测算法,此时算法的输入为 $i_k/o_k, i_{k+1}/o_{k+1}, \dots, i_m/o_m$ ($0 < k \leq m$), 变异后的 ECFSM, 其当前可能的状态集合为 $\{(s_j^0, s_j^1)\}$, 其中 $(s_j^0, s_j^1) \in L^{k-1}$. 如果算法检测出系统有错, 则该错误肯定不是系统实现错误, 可以将其从错误诊断集合中删除. 如果初始诊断集合的所有元素均测试完毕, 结束返回, 否则转到步骤(1).

经过上述步骤, 错误诊断集合仅包含 $(s_3^0, z_2/y_2), (s_2^0, z_1/y_1), (s_1^0, y_2/s_1^0)$ 等5个元素.

3.3 继续测试以区分错误

在获取症状序列之后, 被测系统仍然在运行, 经步骤1、步骤2之后, 如果诊断集合中的元素数大于1, 可以通过继续被动观察来区分错误. 随着检测的继续, 引入错误后的系统当前可能的状态数逐渐减少, 如果状态数减到0, 则该错误可以被排除. 在生成候选诊断集合之后, 假设继续观察到被动测试序列 $x_3/y_3, x_1/y_2, x_1/y_2, x_3/y_2$, 则可确定系统错误为 $(s_3^0, z_2/y_2, s_1^0)$. 为了能够快速定位错误, 可以将被测系统复位, 再设计主动测试用例进行错误区分, 方法描述如下.

(1) 将 ECFSM 合成为一个 DFSM^[8], 将 ECFSM 中的可能错误转换为 DFSM 中的错误.

(2) 将错误按输出错误和转换错误进行分类. 对于输出错误, 施加引导序列和测试输入, 如果存在输出错误, 则错误定位结束, 返回; 对于转换错误, 施加引导序列、测试输入和状态对区分序列^[9] (如自适应区分序列), 如果判定系统状态与规范状态有所不同, 则定位过程结束, 返回. 要求引导序列和区分序列不能存在不确定的转换, 而系统存在状态对区分序列. 如果条件不满足, 下转(3).

(3) 使用交叉判定法^[9]进行错误区分, 如果存在2个错误无法区分, 则说明存在等价错误.

单独使用交叉判定方法可以保证系统在不存在特征序列的情况下也能定位错误, 但该方法每次最多只能排除2个错误, 所以本文将区分序列和交叉判定结合使用.

4 时间复杂度分析及实验结果分析

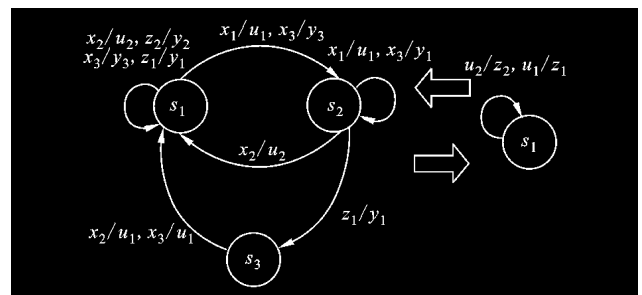
4.1 时间复杂度分析

假设被测系统在第 k 步的可能的状态数为 $|L^k|$, 则以其中1个元素为起始状态的最大可能的错误数为 $l = |I_1^{0-1}| + 2|S^0| + |I_0^{1-0}| + |S^1|$, 每一个错误最多需要检测的步骤数为 $m - k + n$, 其中症

状序列长度为 m , 继续向前观察的步骤数为 n , 系统的状态数为 p , 所以本文诊断算法总的时间复杂度为 $\sum_{k=0}^m (|L^k| \cdot l(m - k + n)) = O((m^2 + mn)lp)$.

4.2 实验及结果分析

H. 245 协议是 H. 323 协议族中用于媒体信道控制的协议, 主要完成多媒体通信中逻辑信道的建立、维护和释放. 在执行单向逻辑信道信令的过程中, 当 incoming LCSE 接收到 outgoing LCSE 的控制请求时, 需要与上层的 user 进行消息或原语交互, 交互过程是不可见的. 将 incoming LCSE (M_0)、上层 user (M_1) 作为被测系统进行测试, 这种 ECFSM 模型如图3所示. 依据该模型构造一个能生成观察序列的执行器, 当植入错误后, 获取被动观察序列, 同时运用本文算法对执行器进行被动测试, 重复测试1 000次, 结果如图4所示.



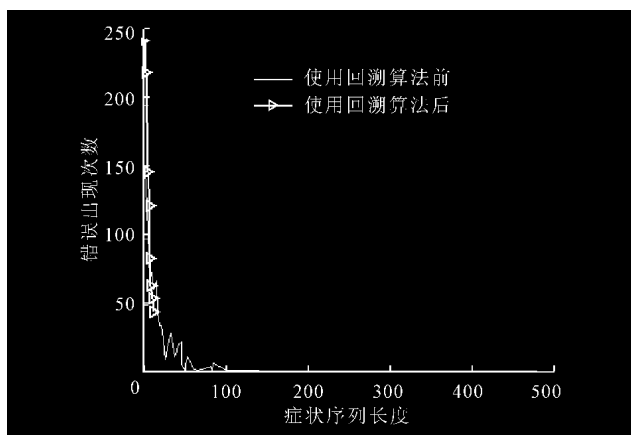
x_1 : 打开逻辑信道; x_2 : 关闭逻辑信道; x_3 : 无效请求; y_1 : 结束“打开逻辑信道”命令; y_2 : 结束“关闭逻辑信道”命令;
 y_3 : -; u_1 : 确定显示; u_2 : 释放显示; z_1 : 确定; z_2 : -

图3 单向逻辑信道信令过程的 ECFSM

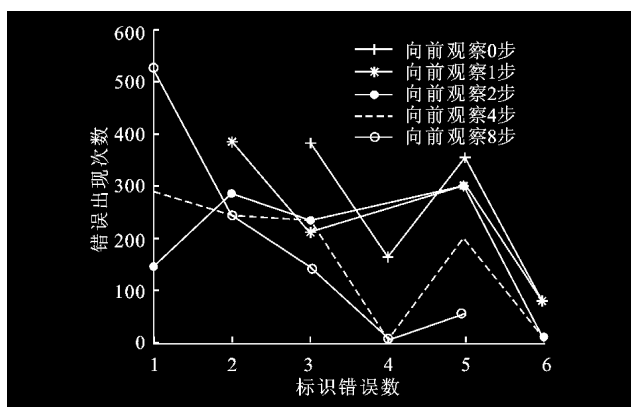
图4a为使用回溯前、后症状序列长度的比较. 从中可以看出, 错误检测算法 (不使用回溯) 可以在500步以内 (95%以上在100步以内) 检测出系统实现错误. 使用回溯可以把症状序列长度降低到15步以内.

图4b为继续观察步骤数为0、1、2、4、8时, 各种可能的错误数所占比例的比较. 可以看出, 运用错误症状序列信息可以诊断错误, 错误集合的规模在3~6之间; 使用继续被动观察的方法可以减少错误集合规模, 并能最终定位错误, 观察步骤数为1时的错误集合的规模为2、3、5、6, 观察步骤数为2、4、8的错误定位率分别为15%、29%、53%.

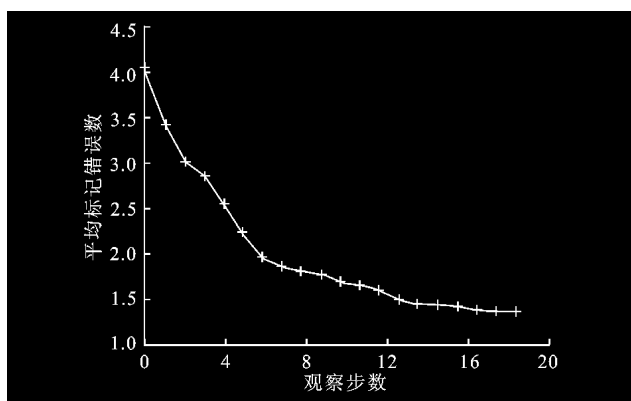
图4c为观察步骤数与平均错误集合规模的关系, 容易看出, 观察步骤越多, 错误集合规模越小. 在观察步骤数为1时的平均错误数由4.13降到3.47, 观察步骤数为6之后的平均错误数降低到



(a) 观察序列长度



(b) 错误数所占比例



(c) 观察步数与平均错误集合规模的关系

图4 本文算法的实验结果

1.99. 继续增加观察步骤数,发现错误集合规模的减少并不明显. 导致该问题的原因是:在观察若干步骤之后,如果不同错误导致系统停留在相同状态,就无法再通过辨别状态的方法来区分错误,系统必须再次经历该错误后才能加以区分. 所以,应使用主动测试加速完成此类错误的定位.

5 结束语

本文对通道不可观察的嵌入式通信系统进行被

动测试错误检测和错误诊断,并通过对一个嵌入式通信系统的实验结果验证了所提方法. 目前的被动测试错误诊断算法都没有考虑到数据流的诊断,能否将本文方法推广到扩展有限状态机模型上,能否应用到状态机判定上,这将是下一步的工作.

参考文献:

- [1] Chen Dongluo, Wu Jianping. An enhanced passive testing tool for network protocols [C]// Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing. Los Alamitos, USA: IEEE Computer Society, 2003: 513-516.
- [2] Lee D, Netravali A N, Sabnani K K, et al. Passive testing and applications to network management [C]// International Conference on Network Protocols. Los Alamitos, USA: IEEE Computer Society, 1997: 113-122.
- [3] Miller R E. Passive testing of networks using a CFSM specification [C]// Proceedings of the IEEE International Performance, Computing and Communications Conference. San Francisco, USA: IEEE Communication Society, 1998: 111-116.
- [4] Lee D, Chen Dongluo, Hao Ruibing, et al. Network protocol system monitoring — a formal approach with passive testing [J]. IEEE/ACM Transactions on Networkings, 2006, 14(2): 424-437.
- [5] Miller R E, Arisha K A. Fault identification in networks by passive testing [C]// 34th Annual Simulation Symposium. Los Alamitos, USA: IEEE Computer Society, 2001: 277-284.
- [6] 赵保华, 张炜. 基于有限状态机的被动测试错误诊断 [J]. 西安交通大学学报, 2006, 40(6): 626-629. Zhao Baohua, Zhang Wei. Passive test for diagnosis based on finite state machines [J]. Journal of Xi'an Jiaotong University, 2006, 40(6): 626-629.
- [7] Ghedamsi A, Bochmann G. Diagnostic tests for communicating finite state machines [C]// Proceedings of Phoenix Conference on Computers and Communications. Piscataway, USA: IEEE, 1993: 254-260.
- [8] El-Fakh K A. Protocol re-testing and diagnostic testing method [D]. Ottawa, Canada: University of Ottawa, 2002.
- [9] Lee D, Sabnani K. Reverse-engineering of communication protocols [C]// Proceedings of the International Conference on Network Protocols. Los Alamitos, USA: IEEE Computer Society, 1993: 208-216.

(编辑 苗凌)