Verification of One-Third Consensus Distributed Systems Algorithm

Suha Mutluergil Orhun[1], Hassan Salehe Matar[1], and Oznur Ozkasap[2]

[1] Koc University - MSRC Lab , Istanbul, TURKEY,
{smutluergil, hmatar}@ku.edu.tr,
[2] Koc University - NDSL Lab, Istanbul, TURKEY

Abstract. Consensus problem is a well-known problem in distributed computing setting. It requires that nodes in a distributed platform that communicate via message passing to agree upon a common value in the presence of (benign or malicious) faults. There are algorithms attacking the Consensus problem, however a few of them is proven to be correct. Even a fewer is verified via formal verification tool. If only the benign faults are considered, the *Heard-Of* (HO) model proposes an elegant, simple and unifying formal framework for representing distributed algorithms that are developed for the Consensus problem. In the HO model, algorithms contain communication-closed rounds. Each message belongs to a round and it can be sent or received only in this particular round. Communication faults can be formalized by *communication predicates* and can be integrated into the model directly. In this study, we formalize a well-known Consensus algorithm called *One-Third Rule* (OTR) algorithm with the HO model, implement it in the meta-language PROMELA and verify it using the model-checking tool SPIN.

1 Introduction

A distributed computing platform consists of separate computing nodes that cooperate to achieve a common goal. In this kind of a setting, the algorithms are desired to be fault-tolerant i.e., they should work properly in case of problems in some nodes or communication channels. Therefore, assumptions on the types of faults on the channels and nodes should be stated clearly while developing an algorithm.

Particularly, the Consensus problem in a distributed system can formalized as follows: It is assumed that every node in a distributed program initially proposes some value and requires that nodes eventually choose a common value among the proposed ones. The problem can be defined over various distributed settings. However, we can characterize settings around two central ideas:

- Degree of synchrony among the nodes,
- Notion of fault in the system.

We can analyse effects of these two factors on the problem separately. First, we consider effect of synchronization on the Consensus problem. It is shown in

[7] that the Consensus problem is unsolvable in fully asynchronous distributed programs if at least one node may fail by a crash. However, the problem is solvable and algorithms exist for synchronous or partly asynchronous distributed programs.

Second, we can assume different kind of faults on a distributed system. We can group them as *malicious* or *benign* faults. Benign faults prevent processes from receiving expected messages, but contents of the messages are not affected. We can count crash failures and some kind of transmission faults (like lost messages) as benign faults. In our study, we assume benign faults since most of the algorithms developed for HO model works correctly in case of benign faults.

On the other hand, malicious faults model more malfunctioning. A message sent by a process may not arrive to another process or it may arrive with a different content. We can categorize *Byzantine* type of failures as malicious faults. It is harder to solve the Consensus problem assuming malicious faults. However, there are algorithms to solve the problem in partially asynchronous systems in the presence of malicious faults [6].

Since assumptions on the system has a wide-range, there are abundant number of studies solving the problem in distributed settings which have subtle differences. Therefore, it is easy to make erroneous claims about what algorithms actually achieve and formal statements and proofs about algorithms appear crucial for comparing them. The HO model presented in [2] offers an elegant, simple and unifying framework for formalizing the the distributed algorithms in the presence of benign faults. In HO, every algorithm operates in rounds and each round is separated from the other in terms of communication messages. A message created in a round can be sent or received only in this round. Moreover, each round is divided into three stages. At the beginning of a round each process first sends messages to the other process, then waits for messages from the other process and finally performs local computations which changes its local state. In integrate faults on this model, HO model requires *communication predicates*. Communication predicates are logical formulas that restrict behaviour of communication operations. With this round based algorithmic structure and communication variables, we can model a great portion of distributed algorithms for the Consensus problem in presence of benign faults. Moreover, a recent study in [1] extends HO model so that it can also model malicious faults in the system.

For the sake of simplicity, we only consider original HO model and assume only benign faults throughout the paper. Even in this restriction, we can mention plenty of algorithms for the Consensus problem. Among them, we can name Uniform Voting Algorithm [2], DLS Algorithm [6], Paxos Algorithm [11], The Last Voting Algorithm [2] and One-Third Rule (OTR) Algorithm [2]. All of the mentioned algorithms are proven to be correct [3]. However, all of them except OTR Algorithm require a coordinator to work properly. Moreover, OTR does not require invariant communication predicates on HO sets i.e. $\forall r > 0 : \mathcal{C}$ holds at r where r is a round number and $\mathcal{C}$ is predicate over HO sets. Apart from that, OTR Algorithm is simple and easier to understand. Each round is identical to the others whereas algorithm's structure changes from round to round in other algorithms. Therefore, we picked OTR as the sample algorithm for our study.

Although HO model provides a nice framework for the problem, it is not obvious how to realize this formal framework and use for automated proofs. In [3], proof assistant tool ISABELLE is used for representing HO models and conducting the proof. However, ISABELLE needs an abstract representation of the system in Linear Temporal Logic (LTL) formulae and requires frequent human intervention. Therefore, it is not a fully automated system.

On the other hand, model-checking tools are widely used for distributed program verification for a long time. They accept programs in a tool specific programming language and they can verify their properties via assertion of some formulae. In this study, we used model-checking tool SPIN to implement and verify our HO algorithm OTR. Although model-checking tools exist in a wide spectrum, we preferred the tool SPIN for two reasons. First, it is widely used and known to be stable. Second, it supports LTL formulae as verification conditions which are necessary for expressing our verification conditions.

In the following section, we will elaborate on related studies upon the subject. Then, we will explain HO model in detail including the OTR algorithm in Section 3. Then, we give formal details of SPIN in Section 4. After all these basics, we explain our implementation details in Section 5. We give experimental results in Section 6 and make conclusion remarks and mention future studies in Section 7.

2 Related Work

In many agreement problems, round-based message-passing systems are designed [4, 5, 16] if the system is synchronous. According to [2], first message-passing algorithm for asynchronous system is proposed in [6]. In [8], round-based model is extended to any type of systems and it is called Round-by-Round Fault Detector (RRFD). In this model, failures are handled by introducing the notion of *faulty component*. This model is not adequate for capturing real faults since only processes are responsible for the failures, never the links. For instance, real cause of the transmission fault which can be sender failure, receiver failure or channel failure may never be known in RRFD. Moreover, faulty component notion unnecessarily overloads system analysis with non-operational details. Therefore, a new model called *Transmission Faults* model is introduced in [15, 14]. In transmission fault model, cause of the problem is not important. A transmission failure may be due to channel, sender or receiver failure but this is not important in the context of transmission faults model. However, transmission faults model is developed for only synchronous systems.

In [2], they propose Heard-Of (HO) model which contains advantages of RRFD and transmission faults model but avoids its drawbacks. In HO model, synchrony degree and failure model is encapsulated in a high-level of abstraction and notion of faulty component is not relevant. It makes HO a desirable model for formal verification and model-checking purposes. In [1], HO model is extended to include systems with malicious faults. Moreover, they prove correctness of HO algorithms in a semi-automated manner using ISABELLE in [3]. However, proofs require high-level of mathematical abstraction of HO model and the algorithms and proof idea and direction should be shown to ISABELLE. Therefore, it can only be used to check correctness of an existing proof idea.

In order to perform proofs in a fully automated fashion, model-checking tools can be used. Although model checking has been widely practised, a little is done for applying it to verification distributed algorithms for consensus. A plausible reason for this is that these algorithms induce huge, often infinite state spaces, thereby severely limiting the usefulness of model-checking techniques. They also induce unbounded number of rounds which is hard to handle. In [10], a shared memory-based randomised consensus algorithm was verified. The authors separated the algorithm into a probabilistic and a non-probabilistic component. They applied standard probabilistic model-checking techniques to the probabilistic component. In the verification of the non-probabilistic part, of which state-space is infinite, they divided problem into small parts and used model-checking for verification. In [18], automatic verification is applied to Optimistically Terminating Consensus (OTC) Algorithm (a single-phase, coordinator-based algorithm). The main aim is to automatically discover one phase that can be extended to the full algorithm. Moreover, they do not prove termination condition for the algorithm and coordination selection is not handled. Our study differs from [18] that we verify entire algorithm with all turns considered and termination property satisfied. In [9], a synchronous consensus algorithm is analysed with a real-time model-checker for the case $n = 3$, where n is the number of processes.

Other examples that use formal verification methods other than model-checking can be stated as [17, 12, 13].

3 the HO model

As it is mentioned in the Introduction section, algorithms in HO model contains rounds. Each round is a *communication-closed layer* which means that a message sent can be received only in that round. Each phase contains three stages. First a process sends its messages to the other processes, then it waits for replies from the other processes and lastly it makes a local state transition. Algorithms should contain processes and messages from finite, non-empty sets Π and M, respectively. In some cases we may need to extend M to $M' = M \cup \{\perp\}$ where $\perp$ is a place holder indicating that no message has arrived, to be able to represent message content of a process.

With this rough description in mind, we can describe algorithms in HO model more formally in the following part.

3.1 Definitions

An HO model machine can be described as a duple $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ where $\mathcal{A}$ is an algorithm and $\mathcal{P}$ is a communication predicate. We will describe structure of an algorithm and a communication predicate in more detail in the following.

An algorithm consists of process specifications. For each $p \in \Pi$, there is an associated *process specification* tuple $Proc_p = (States_p, Init_p, S_p, T_p)$ where

- $States_p$ is a set of possible states (which will be explained more precisely later on) p can take,
- $Init_p \subseteq States_p$ is a non empty set representing *initial states* p can be on,
- $S_p : States_p \times \Pi \to M$ is the message sending function of p,
- $T_p : States_p \times M'^{\Pi} \to States_p$ is the next-state function of p.

Only vague term above is the $States_p$ field of the process specification. Basically, we can think of a process state as a function that maps local variables of p to values from an appropriate range. Each process has fixed variables x_p, $decide_p$ and $HOSet_p$. x_p represents current choice of p on the consensus object at a given round. $decide_p$ holds the value of the consensus object when p decides on it. Before that $decide_p$ is null. $HOSet_p \subseteq \Pi$ is a set of processes that p receives from other processes in a given round.

After formalizing an algorithm $\mathcal{A}$, we can describe the communication predicate $\mathcal{P}$. Communication predicates are defined over a notion called *heard-of collection* which is the collection of subsets of Π indexed by $\Pi \times \mathbb{N}^+$. We represent heard-of collections as $(HO(p,r))_{p \in \Pi, r>0}$ where r denotes the round number. Then, $\mathcal{P}$ is a boolean function over heard-of collections i.e.,

$$\mathcal{P} : (2^{\Pi})^{\Pi \times \mathbb{N}^+} \to \{false, true\}.$$

For HO model, we need to put some restrictions on $\mathcal{P}$. We use predicates whose truth value is invariant under time translation. Formally, $\mathcal{P}$ has same truth value for all $(HO(p, r+i))_{p \in \Pi, r>0}$ where i is any element of $\mathbb{N}^+$.

Let $\mathcal{C}$ be some condition over heard-of sets in a given round. We say that

$$\mathcal{P}_{(\mathcal{C})^\infty} :: \forall r > 0, \exists r_0 > r : \mathcal{C} \; holds \; at \; r_0$$

is also time invariant since it expresses that $\mathcal{C}$ holds infinitely often.

A useful concept that frequently occurs in predicates is the *kernel.* For a process p in a round r, we define kernel as

$$K(r) = \bigcap_{p \in \Pi} HO(p,r).$$

Similarly, we define global kernel of a computation as:

$$K = \bigcap_{r>0} K(r).$$

A round is said to be *nek* (non-empty kernel) if $K(r) \neq \emptyset$.

We may use more concepts that rely on intersection of heard-of sets of processes in rounds, in communication predicates. Two important concepts are *uniform* and *split* rounds. Round r is split when there exists two processes $p, q \in \Pi$ such that $HO(p,r) \cap HO(q,r))\emptyset$ and is uniform when for any two processes $p, q \in \Pi$, $HO(p,r) = HO(q,r)$.

3.2 Executions of HO Machines

As we defined an HO machine formally in the previous part, now we explain its executions and operational semantics.

Each process of an HO algorithm produces executions consisting of an infinite sequence of rounds. At any round r, first $p \in \Pi$ sends a message to all processes according to S_p. Then, it waits for a fixed amount of time for messages from other processes i.e., it updates $HOSet_p$ without violating $\mathcal{P}$. Last, It performs a state transition according to T_p function. Considering this structure, we can define two execution models:

- *Fine-grained executions:* We consider each of three stages as *atomic actions* that may change state of process p and channels incident to p. Then, an *event* is execution of a single action by a process. In this execution model, configuration σ' is a successor configuration of σ if there exists an event such that this event transforms state of a process and channels incident to it in σ to state of these process and channels in σ'. Then, a *fine-grained execution* is defined as a sequence of configurations $\sigma_0, \sigma_1, ...$ such that σ_0 is an initial configuration and for all $i \in N^+$ σ_{i+1} is a successor configuration of σ_i.
- *Coarse-grained executions:* Definition of a coarse-grained execution is the same as fine-grained execution except the notion of atomic actions. In coarse-grained executions each round is considered as a single atomic action i.e., a process $p \in \Pi$ changes state according to T_p and messages p receives from other processes in Π. Moreover, channels are assumed to be empty in this execution model.

Coarse-grained model have few advantages upon fine-grained model. First, coarse-grained model is simpler and has smaller state space. Second, in coarse-grained model, we do not need to represent state of channels in executions since they are always assumed to be empty. Since coarse-grained model presents advantage of representing each round as an atomic action, we want to use it instead of fine-grained model. In fact, we are allowed for such a choice since Theorem 1 in [1], for each fine-grained execution η of an HO algorithm , there exists a coarse-grained execution ζ such that $\eta \approx \zeta$, where $\approx$ represents *stuttering equivalence*, and vice-versa.

3.3 One-Third Rule (OTR) Algorithm

4 SPIN

5 Discussion

6 Results

7 Conclusions

8 Recommendations

9 References

10 Appendices

In this section, we will consider the case when the Hamiltonian $H(x)$ is autonomous. For the sake of simplicity, we shall also assume that it is C^1.

We shall first consider the question of nontriviality, within the general framework of (A_∞, B_∞)-subquadratic Hamiltonians. In the second subsection, we shall look into the special case when H is $(0, b_\infty)$-subquadratic, and we shall try to derive additional information.

The General Case: Nontriviality. We assume that H is (A_∞, B_∞)-subquadratic at infinity, for some constant symmetric matrices A_∞ and B_∞, with $B_\infty - A_\infty$ positive definite. Set:

$$\gamma :=\text{smallest eigenvalue of}\quad B_\infty - A_\infty \tag{1}$$

$$\lambda :=\text{largest negative eigenvalue of}\quad J\frac{d}{dt} + A_\infty \ . \tag{2}$$

Theorem 1 tells us that if $\lambda + \gamma < 0$, the boundary-value problem:

$$\begin{aligned} \dot{x} &= JH'(x) \\ x(0) &= x(T) \end{aligned} \tag{3}$$

has at least one solution $\overline{x}$, which is found by minimizing the dual action functional:

$$\psi(u) = \int_o^T \left[\frac{1}{2}\left(\varLambda_o^{-1}u, u\right) + N^*(-u)\right] dt \tag{4}$$

on the range of $\varLambda$, which is a subspace $R(\varLambda)_L^2$ with finite codimension. Here

$$N(x) := H(x) - \frac{1}{2}\left(A_\infty x, x\right) \tag{5}$$

is a convex function, and

$$N(x) \le \frac{1}{2}\left(\left(B_\infty - A_\infty\right)x, x\right) + c \quad \forall x \ . \tag{6}$$

Proposition 1. *Assume $H'(0)=0$ and $H(0)=0$. Set:*

$$\delta := \liminf_{x\to 0} 2N(x)\left\|x\right\|^{-2} \ . \tag{7}$$

If $\gamma < -\lambda < \delta$, the solution $\overline{u}$ is non-zero:

$$\overline{x}(t) \neq 0 \quad \forall t \ . \tag{8}$$

Proof. Condition (7) means that, for every $\delta' > \delta$, there is some $\varepsilon > 0$ such that

$$\left\|x\right\| \leq \varepsilon \Rightarrow N(x) \leq \frac{\delta'}{2}\left\|x\right\|^2 \ . \tag{9}$$

It is an exercise in convex analysis, into which we shall not go, to show that this implies that there is an $\eta > 0$ such that

$$f\left\|x\right\| \leq \eta \Rightarrow N^*(y) \leq \frac{1}{2\delta'}\left\|y\right\|^2 \ . \tag{10}$$

Fig. 1. This is the caption of the figure displaying a white eagle and a white horse on a snow field

Since u_1 is a smooth function, we will have $\left\|hu_1\right\|_\infty \leq \eta$ for h small enough, and inequality (10) will hold, yielding thereby:

$$\psi(hu_1) \leq \frac{h^2}{2}\frac{1}{\lambda}\left\|u_1\right\|_2^2 + \frac{h^2}{2}\frac{1}{\delta'}\left\|u_1\right\|^2 \ . \tag{11}$$

If we choose δ' close enough to δ, the quantity $\left(\frac{1}{\lambda}+\frac{1}{\delta'}\right)$ will be negative, and we end up with

$$\psi(hu_1) < 0 \qquad \text{for} \quad h \neq 0 \ \text{ small} \ . \tag{12}$$

On the other hand, we check directly that $\psi(0)=0$. This shows that 0 cannot be a minimizer of ψ, not even a local one. So $\overline{u} \neq 0$ and $\overline{u} \neq \Lambda_o^{-1}(0) = 0$. □

Corollary 1. *Assume H is C^2 and (a_∞, b_∞)-subquadratic at infinity. Let ξ_1, $\dots, \xi_N$ be the equilibria, that is, the solutions of $H'(\xi)=0$. Denote by ω_k the smallest eigenvalue of $H''\left(\xi_k\right)$, and set:*

$$\omega := \text{Min}\left\{\omega_1, \dots, \omega_k\right\} \ . \tag{13}$$

If:

$$\frac{T}{2\pi}b_\infty < -E\left[-\frac{T}{2\pi}a_\infty\right] < \frac{T}{2\pi}\omega \tag{14}$$

then minimization of ψ yields a non-constant T-periodic solution $\overline{x}$.

We recall once more that by the integer part $E[\alpha]$ of $\alpha \in \mathrm{I\!R}$, we mean the $a \in \mathbb{Z}$ such that $a < \alpha \le a+1$. For instance, if we take $a_\infty = 0$, Corollary 2 tells us that $\overline{x}$ exists and is non-constant provided that:

$$\frac{T}{2\pi}b_\infty < 1 < \frac{T}{2\pi} \tag{15}$$

or

$$T \in \left(\frac{2\pi}{\omega}, \frac{2\pi}{b_\infty}\right) \ . \tag{16}$$

Proof. The spectrum of Λ is $\frac{2\pi}{T}\mathbb{Z} + a_\infty$. The largest negative eigenvalue λ is given by $\frac{2\pi}{T}k_o + a_\infty$, where

$$\frac{2\pi}{T}k_o + a_\infty < 0 \le \frac{2\pi}{T}(k_o+1) + a_\infty \ . \tag{17}$$

Hence:

$$k_o = E\left[-\frac{T}{2\pi}a_\infty\right] \ . \tag{18}$$

The condition $\gamma < -\lambda < \delta$ now becomes:

$$b_\infty - a_\infty < -\frac{2\pi}{T}k_o - a_\infty < \omega - a_\infty \tag{19}$$

which is precisely condition (14). □

Lemma 1. *Assume that H is C^2 on $\mathrm{I\!R}^{2n}\backslash\{0\}$ and that $H''(x)$ is non-degenerate for any $x \ne 0$. Then any local minimizer $\widetilde{x}$ of ψ has minimal period T.*

Proof. We know that $\widetilde{x}$, or $\widetilde{x} + \xi$ for some constant $\xi \in \mathrm{I\!R}^{2n}$, is a T-periodic solution of the Hamiltonian system:

$$\dot{x} = JH'(x) \ . \tag{20}$$

There is no loss of generality in taking $\xi = 0$. So $\psi(x) \ge \psi(\widetilde{x})$ for all $\widetilde{x}$ in some neighbourhood of x in $W^{1,2}\left(\mathrm{I\!R}/T\mathbb{Z};\mathrm{I\!R}^{2n}\right)$.

But this index is precisely the index $i_T(\widetilde{x})$ of the T-periodic solution $\widetilde{x}$ over the interval $(0,T)$, as defined in Sect. 2.6. So

$$i_T(\widetilde{x}) = 0 \ . \tag{21}$$

Now if $\widetilde{x}$ has a lower period, T/k say, we would have, by Corollary 31:

$$i_T(\widetilde{x}) = i_{kT/k}(\widetilde{x}) \ge k i_{T/k}(\widetilde{x}) + k - 1 \ge k - 1 \ge 1 \ . \tag{22}$$

This would contradict (21), and thus cannot happen. □

Notes and Comments. The results in this section are a refined version of [**?**]; the minimality result of Proposition 14 was the first of its kind.

To understand the nontriviality conditions, such as the one in formula (16), one may think of a one-parameter family x_T, $T \in \left(2\pi\omega^{-1}, 2\pi b_\infty^{-1}\right)$ of periodic solutions, $x_T(0) = x_T(T)$, with x_T going away to infinity when $T \to 2\pi\omega^{-1}$, which is the period of the linearized system at 0.

Table 1. This is the example table taken out of *The TEXbook,* p. 246

Year	World population
8000 B.C.	5,000,000
50 A.D.	200,000,000
1650 A.D.	500,000,000
1945 A.D.	2,300,000,000
1980 A.D.	4,400,000,000

Theorem 1 (Ghoussoub-Preiss). *Assume $H(t,x)$ is $(0,\varepsilon)$-subquadratic at infinity for all $\varepsilon > 0$, and T-periodic in t*

$$H(t,\cdot) \qquad \text{is convex} \quad \forall t \tag{23}$$

$$H(\cdot,x) \qquad \text{is} \quad T\text{−periodic} \quad \forall x \tag{24}$$

$$H(t,x) \geq n\left(\|x\|\right) \qquad \text{with} \quad n(s)s^{-1} \to \infty \quad \text{as} \quad s \to \infty \tag{25}$$

$$\forall \varepsilon > 0 \ , \quad \exists c \ : \ H(t,x) \leq \frac{\varepsilon}{2}\left\|x\right\|^2 + c \ . \tag{26}$$

Assume also that H is C^2, and $H''(t,x)$ is positive definite everywhere. Then there is a sequence x_k, $k \in \mathbb{N}$, of kT-periodic solutions of the system

$$\dot{x} = JH'(t,x) \tag{27}$$

such that, for every $k \in \mathbb{N}$, there is some $p_o \in \mathbb{N}$ with:

$$p \geq p_o \Rightarrow x_{pk} \neq x_k \ . \tag{28}$$

□

*Example 1 (*External forcing*).* Consider the system:

$$\dot{x} = JH'(x) + f(t) \tag{29}$$

where the Hamiltonian H is $(0, b_\infty)$-subquadratic, and the forcing term is a distribution on the circle:

$$f = \frac{d}{dt}F + f_o \qquad \text{with} \quad F \in L^2\left(\mathbb{R}/T\mathbb{Z}; \mathbb{R}^{2n}\right) \ , \tag{30}$$

where $f_o := T^{-1} \int_o^T f(t)dt$. For instance,

$$f(t) = \sum_{k \in \mathbb{N}} \delta_k \xi \ , \tag{31}$$

where δ_k is the Dirac mass at $t = k$ and $\xi \in \mathbb{R}^{2n}$ is a constant, fits the prescription. This means that the system $\dot{x} = JH'(x)$ is being excited by a series of identical shocks at interval T.

Definition 1. *Let $A_\infty(t)$ and $B_\infty(t)$ be symmetric operators in $\mathbb{R}^{2n}$, depending continuously on $t \in [0,T]$, such that $A_\infty(t) \le B_\infty(t)$ for all t.*

A Borelian function $H : [0,T] \times \mathbb{R}^{2n} \to \mathbb{R}$ is called (A_∞, B_∞)-subquadratic at infinity if there exists a function $N(t,x)$ such that:

$$H(t,x) = \frac{1}{2} \left(A_\infty(t)x, x\right) + N(t,x) \tag{32}$$

$$\forall t \ , \quad N(t,x) \quad \text{is convex with respect to } \ x \tag{33}$$

$$N(t,x) \ge n\left(\|x\|\right) \qquad \text{with } \ n(s)s^{-1} \to +\infty \ \text{ as } \ s \to +\infty \tag{34}$$

$$\exists c \in \mathbb{R} \ : \quad H(t,x) \le \frac{1}{2}\left(B_\infty(t)x, x\right) + c \quad \forall x \ . \tag{35}$$

If $A_\infty(t) = a_\infty I$ and $B_\infty(t) = b_\infty I$, with $a_\infty \le b_\infty \in \mathbb{R}$, we shall say that H is (a_∞, b_∞)-subquadratic at infinity. As an example, the function $\|x\|^\alpha$, with $1 \le \alpha < 2$, is $(0,\varepsilon)$-subquadratic at infinity for every $\varepsilon > 0$. Similarly, the Hamiltonian

$$H(t,x) = \frac{1}{2}k \left\|k\right\|^2 + \left\|x\right\|^\alpha \tag{36}$$

is $(k, k+\varepsilon)$-subquadratic for every $\varepsilon > 0$. Note that, if $k < 0$, it is not convex.

Notes and Comments. The first results on subharmonics were obtained by Rabinowitz in [**?**], who showed the existence of infinitely many subharmonics both in the subquadratic and superquadratic case, with suitable growth conditions on H'. Again the duality approach enabled Clarke and Ekeland in [**?**] to treat the same problem in the convex-subquadratic case, with growth conditions on H only.

Recently, Michalek and Tarantello (see [**?**] and [**?**]) have obtained lower bound on the number of subharmonics of period kT, based on symmetry considerations and on pinching estimates, as in Sect. 5.2 of this article.

References

1. Bernadette Charron-Bost, Henri Debrat, and Stephan Merz. Formal verification of consensus algorithms tolerating malicious faults. In *Stabilization, Safety, and Security of Distributed Systems*, pages 120–134. Springer, 2011.
2. Bernadette Charron-Bost and André Schiper. The heard-of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, 2009.

3. Henri Debrat, Stephan Merz, et al. Verifying fault-tolerant distributed algorithms in the heard-of model. 2012.
4. Danny Dolev, Ruediger Reischuk, and H Raymond Strong. Early stopping in byzantine agreement. *Journal of the ACM (JACM)*, 37(4):720–741, 1990.
5. Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.
6. Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2):288–323, 1988.
7. Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.
8. Eli Gafni. Round-by-round fault detectors (extended abstract): unifying synchrony and asynchrony. In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 143–152. ACM, 1998.
9. Martijn Hendriks. Model checking the time to reach agreement. In *Formal Modeling and Analysis of Timed Systems*, pages 98–111. Springer, 2005.
10. Marta Kwiatkowska, Gethin Norman, and Roberto Segala. Automated verification of a randomized distributed consensus protocol using cadence smv and prism? In *Computer Aided Verification*, pages 194–206. Springer, 2001.
11. Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems (TOCS)*, 16(2):133–169, 1998.
12. Uwe Nestmann, Rachele Fuzzati, and Massimo Merro. Modeling consensus in a process calculus. In *CONCUR 2003-Concurrency Theory*, pages 399–414. Springer, 2003.
13. Anna Pogosyants, Roberto Segala, and Nancy Lynch. Verification of the randomized consensus algorithm of aspnes and herlihy: a case study. *Distributed Computing*, 13(3):155–186, 2000.
14. Nicola Santoro and Peter Widmayer. Time is not a healer. In *STACS 89*, pages 304–313. Springer, 1989.
15. Nicola Santoro and Peter Widmayer. Agreement in synchronous networks with ubiquitous faults. *Theoretical Computer Science*, 384(2):232–249, 2007.
16. Ulrich Schmid. How to model link failures: A perception-based fault model. In *Dependable Systems and Networks, 2001. DSN 2001. International Conference on*, pages 57–66. IEEE, 2001.
17. Toh Ne Win, Michael D Ernst, Stephen J Garland, Dilsun Kırlı, and Nancy A Lynch. Using simulated execution in verifying distributed algorithms. *International Journal on Software Tools for Technology Transfer*, 6(1):67–76, 2004.
18. Piotr Zielinski. Automatic verification and discovery of byzantine consensus protocols. In *Dependable Systems and Networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference on*, pages 72–81. IEEE, 2007.

Subject Index

Absorption 327

Absorption of radiation 289–292, 299, 300

Actinides 244

Aharonov-Bohm effect 142–146

Angular momentum 101–112

– algebraic treatment 391–396
Angular momentum addition 185–193
Angular momentum commutation relations 101
Angular momentum quantization 9–10, 104–106
Angular momentum states 107, 321, 391–396
Antiquark 83
α-rays 101–103
Atomic theory 8–10, 219–249, 327
Average value
(*see also* Expectation value) 15–16, 25, 34, 37, 357

Baker-Hausdorff formula 23
Balmer formula 8
Balmer series 125
Baryon 220, 224
Basis 98
Basis system 164, 376
Bell inequality 379–381, 382
Bessel functions 201, 313, 337
– spherical 304–306, 309, 313–314, 322
Bound state 73–74, 78–79, 116–118, 202, 267, 273, 306, 348, 351
Boundary conditions 59, 70
Bra 159
Breit-Wigner formula 80, 84, 332
Brillouin-Wigner perturbation theory 203

Cathode rays 8
Causality 357–359
Center-of-mass frame 232, 274, 338
Central potential 113–135, 303–314
Centrifugal potential 115–116, 323
Characteristic function 33
Clebsch-Gordan coefficients 191–193
Cold emission 88
Combination principle, Ritz's 124
Commutation relations 27, 44, 353, 391
Commutator 21–22, 27, 44, 344
Compatibility of measurements 99
Complete orthonormal set 31, 40, 160, 360
Complete orthonormal system, *see* Complete orthonormal set
Complete set of observables, *see* Complete set of operators

Eigenfunction 34, 46, 344–346
– radial 321
– – calculation 322–324
EPR argument 377–378
Exchange term 228, 231, 237, 241, 268, 272

f-sum rule 302
Fermi energy 223

H_2^+ molecule 26
Half-life 65
Holzwarth energies 68