Iterative Transient State Distribution Calculation in Semi-Markov Processes

Nicholas J. Dingle*

June 4, 2004

Abstract

This paper presents an iterative technique for the transient analysis of large structurally unrestricted semi-Markov processes (SMPs), which builds on our previous work on iterative passage time calculation. The method is based on the calculation and subsequent numerical inversion of Laplace transforms. We demonstrate our technique on a Markovian process algebra model of a web-server with 69 440 states.

1 Introduction

We describe an iterative algorithm for the calculation of transient state distributions in large structurally unrestricted semi-Markov processes (SMPs) presented in [BDHK04]. This is built on our iterative passage time [BDHK03, BDKW03] and is based on the calculation and subsequent numerical inversion of Laplace transforms. Our algorithm requires much less computational effort than previous techniques, particularly when computing the transient state distribution of a set of states rather than a single state.

The rest of this paper is organised as follows. In Section 2, we briefly detail the background theory behind semi-Markov processes and show how to derive first passage times and transient state distributions. Our iterative passage time procedure is described in Section 3 and the new iterative transient scheme is presented in Section 4. A process algebra model of a web-server with 69 440 states is described in Section 5 and an example transient state distribution is calculated. Section 6 concludes.

*Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ. Email: njd200@doc.ic.ac.uk

2 Definitions and Background Theory

2.1 Semi-Markov Processes

Consider a Markov renewal process $\{(\chi_n, T_n) : n \geq 0\}$ where T_n is the time of the nth transition ($T_0 = 0$) and $\chi_n \in \mathcal{S}$ is the state at the nth transition. Let the kernel of this process be:

$$R(n, i, j, t) = \mathbb{P}(\chi_{n+1} = j, T_{n+1} - T_n \leq t \mid \chi_n = i)$$

for $i, j \in \mathcal{S}$. The continuous time semi-Markov process, $\{Z(t), t \geq 0\}$, defined by the kernel R, is related to the Markov renewal process by:

$$Z(t) = \chi_{N(t)}$$

where $N(t) = \max\{n : T_n \leq t\}$, i.e. the number of state transitions that have taken place by time t. Thus $Z(t)$ represents the state of the system at time t. We consider only SMPs in which $R(n, i, j, t)$ is independent of any previous state except the last, and thus R becomes independent of n:

$$\begin{aligned} R(i, j, t) &= \mathbb{P}(\chi_{n+1} = j, T_{n+1} - T_n \leq t \mid \chi_n = i) \quad \text{for any } n \geq 0 \\ &= p_{ij} H_{ij}(t) \end{aligned}$$

where $p_{ij} = \mathbb{P}(\chi_{n+1} = j \mid \chi_n = i)$ is the state transition probability between states i and j and $H_{ij}(t) = \mathbb{P}(T_{n+1} - T_n \leq t \mid \chi_{n+1} = j, \chi_n = i)$, is the sojourn time distribution in state i when the next state is j.

2.2 First Passage Times

Consider a finite, irreducible, continuous-time semi-Markov process with N states $\{1, 2, \ldots, N\}$. Recalling that $Z(t)$ denotes the state of the SMP at time t ($t \geq 0$) and the $N(t)$ denotes the number of transitions which have occurred by time t, the first passage time from a source state i at time t into a non-empty set of target states $\vec{\jmath}$ is defined as:

$$P_{i\vec{\jmath}}(t) = \inf\{u > 0 : Z(t + u) \in \vec{\jmath}, N(t + u) > N(t), Z(t) = i\}$$

For a stationary time-homogeneous SMP, $P_{i\vec{\jmath}}(t)$ is independent of t:

$$P_{i\vec{\jmath}} = \inf\{u > 0 : Z(u) \in \vec{\jmath}, N(u) > 0, Z(0) = i\} \tag{1}$$

$P_{i\vec{\jmath}}$ is a random variable with an associated probability density function $f_{i\vec{\jmath}}(t)$ such that the passage time quantile is defined as:

$$\mathbb{P}(t_1 < P_{i\vec{\jmath}} < t_2) = \int_{t_1}^{t_2} f_{i\vec{\jmath}}(t)\ dt$$

The Laplace transform of $f_{i\vec{\jmath}}(t)$, $L_{i\vec{\jmath}}(s)$, can be computed by solving a set of N linear equations:

$$L_{i\vec{\jmath}}(s) = \sum_{k \notin \vec{\jmath}} r^*_{ik}(s) L_{k\vec{\jmath}}(s) + \sum_{k \in \vec{\jmath}} r^*_{ik}(s) \qquad : \text{for } 1 \leq i \leq N \tag{2}$$

where $r^*_{ik}(s)$ is the Laplace-Stieltjes transform (LST) of $R(i, k, t)$ from Section 2.1 and is defined by:

$$r^*_{ik}(s) = \int_0^\infty e^{-st} dR(i, k, t)$$

When there are multiple source states, denoted by the vector $\vec{\imath}$, the Laplace transform of the passage time distribution is:

$$L_{\vec{\imath}\vec{\jmath}}(s) = \sum_{k \in \vec{\imath}} \alpha_k L_{k\vec{\jmath}}(s)$$

where the weight α_k is the probability of being in state $k \in \vec{\imath}$ at the starting instant of the passage. If measuring the system from equilibrium then the α_ks are normalised steady-state probabilities. That is, if $\boldsymbol{\pi}$ denotes the steady-state vector of the embedded discrete-time Markov chain (DTMC), then α_k is given by:

$$\alpha_k = \begin{cases} \pi_k / \sum_{j \in \vec{\imath}} \pi_j & \text{if } k \in \vec{\imath} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The vector with components α_k is denoted by $\boldsymbol{\alpha}$.

The function $f_{i\vec{\jmath}}$ can then be retrieved from $L_{i\vec{\jmath}}(s)$ by using a number of numerical Laplace transform inversion techniques, for example the Euler method [AW95]. This requires the evaluation of $L_{i\vec{\jmath}}(s)$ at a number of values of s (typically, around 30) for each value of t at which the passage time density or quantile is required.

2.3 Transient state distributions

Another important modelling result is the transient state distribution $\pi_{ij}(t)$ of a stochastic process:

$$\pi_{ij}(t) = \mathbb{P}(Z(t) = j \mid Z(0) = i)$$

From Pyke's seminal paper on SMPs [Pyk61], we have the following relationship between passage time densities and transient state distributions, in Laplace form:

$$\pi^*_{ij}(s) = \frac{1}{s} \frac{1 - h^*_i(s)}{1 - L_{ii}(s)} \quad \text{if } i = j, \quad \pi^*_{ij}(s) = L_{ij}(s) \pi^*_{jj}(s) \quad \text{if } i \neq j \tag{4}$$

where $\pi^*_{ij}(s)$ is the Laplace transform of $\pi_{ij}(t)$ and $h^*_i(s) = \sum_k r^*_{ik}(s)$ is the LST of the sojourn time distribution in state i. For multiple target states, this becomes:

$$\pi^*_{i\vec{\jmath}}(s) = \sum_{k \in \vec{\jmath}} \pi^*_{ik}(s) \tag{5}$$

However, to construct $\pi^*_{i\vec{\jmath}}(s)$ directly using this translation is computationally expensive: for a vector of target states $\vec{\jmath}$, we need $2|\vec{\jmath}| - 1$ passage time quantities, $L_{ik}(s)$, which in turn require the solution of $|\vec{\jmath}|$ linear systems of the form of Eq. 2. This motivates our development of a new transient state distribution formula for multiple target states in semi-Markov processes which requires the solution of only one system of linear equations per s-value required.

From Pyke's formula for the transient state distribution between two states [Pyk61, Eq. (3.2)], we can derive:

$$\pi_{ij}(t) = \delta_{ij}\overline{F}_i(t) + \sum_{k=1}^{N} \int_0^t R(i,k,t-\tau)\, \pi_{kj}(\tau)\ d\tau$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise, and $\overline{F}_i(t)$ is the reliability function of the sojourn time distribution in state i, i.e. the probability that the system has not left state i after t time units. $R(i,k,t-\tau)$ represents the occurrence of a single transition out of state i to an adjacent state k in time $t - \tau$ and $\pi_{kj}(\tau)$ is the probability of being in state j having left state k after a further time τ.

Transforming this convolution into the Laplace domain and generalising to multiple target states, $\vec{\jmath}$, we obtain:

$$\pi^*_{i\vec{\jmath}}(s) = \delta_{i\in\vec{\jmath}}\overline{F}^*_i(s) + \sum_{k=1}^{N} r^*_{ik}(s)\pi^*_{k\vec{\jmath}}(s) \tag{6}$$

Here, $\delta_{i\in\vec{\jmath}} = 1$ if $i \in \vec{\jmath}$ and 0 otherwise. The Laplace transform of the reliability function $\overline{F}^*_i(s)$ is generated from $h^*_i(s)$ as:

$$\overline{F}^*_i(s) = \frac{1 - h^*_i(s)}{s}$$

Eq. 6 can be written in matrix–vector form; for example, when $\vec{\jmath} = \{1, 3\}$, we have:

$$\begin{pmatrix} 1 - r^*_{11}(s) & -r^*_{12}(s) & \cdots & -r^*_{1N}(s) \\ -r^*_{21}(s) & 1 - r^*_{22}(s) & \cdots & -r^*_{2N}(s) \\ -r^*_{31}(s) & -r^*_{32}(s) & \cdots & -r^*_{3N}(s) \\ \vdots & \vdots & \ddots & \vdots \\ -r^*_{N2}(s) & -r^*_{N2}(s) & \cdots & 1 - r^*_{NN}(s) \end{pmatrix} \begin{pmatrix} \pi^*_{1\vec{\jmath}}(s) \\ \pi^*_{2\vec{\jmath}}(s) \\ \pi^*_{3\vec{\jmath}}(s) \\ \vdots \\ \pi^*_{N\vec{\jmath}}(s) \end{pmatrix} = \begin{pmatrix} \overline{F}^*_1(s) \\ 0 \\ \overline{F}^*_3(s) \\ \vdots \\ 0 \end{pmatrix} \tag{7}$$

Again for multiple source states with initial distribution $\boldsymbol{\alpha}$, the Laplace transform of the transient function is:

$$\pi^*_{\vec{\imath}\vec{\jmath}}(s) = \sum_{k \in \vec{\imath}} \alpha_k \pi^*_{k\vec{\jmath}}(s)$$

3 Iterative Passage Time Analysis

In this section, we describe the iterative algorithm for generating passage time densities/quantiles previously presented in [BDHK03, BDKW03]. The technique computes the rth transition passage time of the system, $P^{(r)}_{i\vec{\jmath}}$. This is the conditional passage time of the system having reached any of the specified target states within r state-transitions. The unconditioned passage time random variable, $P_{i\vec{\jmath}}$, is then obtained in the limit as $r \to \infty$. The Laplace transform of $P^{(r)}_{i\vec{\jmath}}$, $L^{(r)}_{i\vec{\jmath}}(s)$, is calculated with a sufficiently high value of r to give an approximation to $L_{i\vec{\jmath}}(s)$ to within a specified degree of accuracy. $L_{i\vec{\jmath}}(s)$ is then numerically inverted to obtain the desired passage time density $f_{i\vec{\jmath}}(t)$.

Recall the semi-Markov process $Z(t)$ of Section 2.2, where $N(t)$ is the number of state transitions that have taken place by time t. Formally, we define the rth transition first passage time to be:

$$P^{(r)}_{i\vec{\jmath}} = \inf\{u > 0 : Z(u) \in \vec{\jmath}, 0 < N(u) \leq r, Z(0) = i\} \tag{8}$$

which is the time taken to enter a state in $\vec{\jmath}$ for the first time having started in state i at time 0 and having undergone up to r state transitions. $P^{(r)}_{i\vec{\jmath}}$ is a random variable with associated probability density function, $f^{(r)}_{i\vec{\jmath}}(t)$, which has Laplace transform $L^{(r)}_{i\vec{\jmath}}(s)$. $L^{(r)}_{i\vec{\jmath}}(s)$ is, in turn, the ith component of the vector

$$\mathbf{L}^{(r)}_{\vec{\jmath}}(s) = \left(L^{(r)}_{1\vec{\jmath}}(s), L^{(r)}_{2\vec{\jmath}}(s), \ldots, L^{(r)}_{N\vec{\jmath}}(s)\right)$$

representing the passage time for terminating in $\vec{\jmath}$ for each possible start state. This vector may be computed as:

$$\mathbf{L}^{(r)}_{\vec{\jmath}}(s) = \mathbf{U}\left(\mathbf{I} + \mathbf{U}' + \mathbf{U}'^2 + \cdots + \mathbf{U}'^{(r-1)}\right)\mathbf{e}_{\vec{\jmath}}$$

where $\mathbf{U}$ is a matrix with elements $u_{pq} = r^*_{pq}(s)$ and $\mathbf{U}'$ is a modified version of $\mathbf{U}$ with elements $u'_{pq} = \delta_{p \notin \vec{\jmath}}\, u_{pq}$, where states in $\vec{\jmath}$ have been made absorbing. The column vector $\mathbf{e}_{\vec{\jmath}}$ has entries $e_{k\vec{\jmath}} = \delta_{k \in \vec{\jmath}}$.

From Eqs. 1 and 8:

$$P_{i\vec{\jmath}} = P^{(\infty)}_{i\vec{\jmath}} \quad \text{and thus} \quad L_{i\vec{\jmath}}(s) = L^{(\infty)}_{i\vec{\jmath}}(s).$$

This approach can be generalised to multiple source states $\vec{\imath}$ using, for example, the normalised steady-state vector $\boldsymbol{\alpha}$ of Eq. 3:

$$\begin{aligned} L_{\vec{\imath}\vec{\jmath}}^{(r)}(s) &= \boldsymbol{\alpha}\mathbf{L}_{\vec{\jmath}}^{(r)}(s) \\ &= (\boldsymbol{\alpha}\mathbf{U} + \boldsymbol{\alpha}\mathbf{U}\mathbf{U}' + \boldsymbol{\alpha}\mathbf{U}\mathbf{U}'^2 + \cdots + \boldsymbol{\alpha}\mathbf{U}\mathbf{U}'^{(r-1)})\,\mathbf{e}_{\vec{\jmath}} \\ &= \sum_{k=0}^{r-1} \boldsymbol{\alpha}\mathbf{U}\mathbf{U}'^k\,\mathbf{e}_{\vec{\jmath}} \end{aligned} \tag{9}$$

Convergence of the sum in Eq. 9 is said to have occurred at a particular r, if, for a given s-point:

$$|\mathrm{Re}(L_{\vec{\imath}\vec{\jmath}}^{(r+1)}(s) - L_{\vec{\imath}\vec{\jmath}}^{(r)}(s))| < \varepsilon \quad \text{and} \quad |\mathrm{Im}(L_{\vec{\imath}\vec{\jmath}}^{(r+1)}(s) - L_{\vec{\imath}\vec{\jmath}}^{(r)}(s))| < \varepsilon$$

where ε is chosen to be a suitably small value (e.g. 10^{-8}).

4 Iterative Transient Method

Our iterative transient state distribution generation technique builds on the passage time computation technique of the previous section. We aim to calculate $\pi_{i\vec{\jmath}}(t)$, that is the probability of being in any of the states of $\vec{\jmath}$ at time t having started in state i at time $t = 0$. We approximate this transient state distribution by constructing $\pi_{i\vec{\jmath}}^{(r)}(s)$, which is the rth iterative approximation to the Laplace Transform of the transient state distribution function.

$\pi_{i\vec{\jmath}}^{(r)}(s)$ is, in turn, the ith component of the vector:

$$\tilde{\pi}_{\vec{\jmath}}^{(r)}(s) = \Big(\pi_{1\vec{\jmath}}^{(r)}(s), \pi_{2\vec{\jmath}}^{(r)}(s), \ldots, \pi_{N\vec{\jmath}}^{(r)}(s)\Big)$$

which may be computed as:

$$\tilde{\pi}_{\vec{\jmath}}^{(r)}(s) = \Big(\mathbf{I} + \mathbf{U} + \mathbf{U}^2 + \cdots + \mathbf{U}^r\Big)\,\mathbf{v} \tag{10}$$

where $\mathbf{v}$ is composed of the reliability functions for each of the target states in $\vec{\jmath}$, i.e. $v_i = \delta_{i\in\vec{\jmath}}\overline{F}_i^*(s)$.

Note that instead of using an absorbing transition matrix as in the passage time scheme, the transient method makes use of the unmodified transition matrix U which has elements $u_{pq} = r_{pq}^*(s)$. This reflects the fact that the transient state distribution accumulates probability from all the passages through the system and not just the first one.

Finally, as before, the technique can be generalised to multiple start states by employing an initial vector $\boldsymbol{\alpha}$, where α_i is the probability of being in state i at time 0:

$$\pi_{\vec{\imath}\vec{\jmath}}^{(r)}(s) = \boldsymbol{\alpha}\Big(\mathbf{I} + \mathbf{U} + \mathbf{U}^2 + \cdots + \mathbf{U}^r\Big)\,\mathbf{v}$$

Having calculated $\pi^{(r)}_{i\vec{J}}(s)$ in this manner, the same numerical inversion techniques which are used in passage time analysis can be employed to compute $\pi_{i\vec{J}}(t)$.

5 Web-server Model

We demonstrate our technique by conducting transient analysis on the PEPA model of a high-availability web-server shown in Fig. 1 (originally described in [AKBD04]). This has an underlying Markov chain of 69 440 states. Although our iterative technique was developed mainly for use on SMPs, it can also be used to analyse Markovian systems.

Fig. 2 shows the transient state probability that all 5 servers have failed from the time when the system is first initialised. Also plotted is the corresponding steady-state probability, and we note that the transient probability converges onto this.

Of the 69 440 states in the model, 320 were target states (that is, there were 320 states in which all servers had failed). The graph is plotted for 100 t-points, and using the Euler inversion method this required the evaluation of $\pi^{*}_{i\vec{J}}(s)$ at 3 300 different values of s and thus 3 300 evaluations of Eq. 10. Had Pyke's method (described in Eqs. 4 and 5) been used, 1 056 000 evaluations of Eq. 9 would have had to have been performed. This illustrates the computational efficiency of our algorithm compared with previous techniques.

6 Conclusion

In this paper, we have presented an iterative algorithm for the calculation of transient state distributions in semi-Markov processes which requires significantly less computational effort than existing techniques. This builds on our prior work on the iterative calculation of passage time densities and quantiles. We have demonstrated our technique by calculating a transient state distribution in a process algebra model of a web-server with 69 440 states.

References

[AKBD04] A. Argent-Katwala, J.T. Bradley, and N.J. Dingle. Expressing performance requirements using regular expressions to specify stochastic probes over process algebra models. In *Proceedings of the 4th International Workshop on Software and Performance (WOSP'04)*, pages 49–58, Redwood Shores CA, USA, January 14th–16th 2004.

$$
\begin{aligned}
Server &\stackrel{def}{=} (ReadRequest, \top).ServerRead + (sFail, rsf).ServerFail \\
&+ (sWrite, rsw).Server \\
ServerRead &\stackrel{def}{=} (sReadLookup, rsrl).Server \\
ServerFail &\stackrel{def}{=} (sFailRecover, rsfr).Server + (sFailRecoverAll, \top).Server \\
&+ (sWrite, rsfsw).ServerFail \\
ServerGroup_0 &\stackrel{def}{=} (sFail, \top).ServerGroup_1 \\
ServerGroup_i &\stackrel{def}{=} (sFail, \top).ServerGroup_{i+1} + (sFailRecover, \top).ServerGroup_{i-1} \\
&\quad : 1 \leq i < 5 \\
ServerGroup_5 &\stackrel{def}{=} (sFailRecoverAll, rsgsfra).ServerGroup_0 \\
Servers &\stackrel{def}{=} (Server \underset{\mathcal{L}}{\bowtie} Server \underset{\mathcal{L}}{\bowtie} Server \underset{\mathcal{L}}{\bowtie} Server \underset{\mathcal{L}}{\bowtie} Server) \\
&\quad \underset{\mathcal{L}'}{\bowtie} ServerGroup0
\end{aligned}
$$

where $\mathcal{L} = \{sWrite, sFailRecoverAll\}$ and $\mathcal{L}' = \mathcal{L} \cup \{sFailRecover\}$.

$$
\begin{aligned}
WriteBuffer_0 &\stackrel{def}{=} (bWrite, \top).WriteBuffer_1 \\
WriteBuffer_i &\stackrel{def}{=} (bWrite, \top).WriteBuffer_{i+1} + (sWrite, \top).WriteBuffer_0 \\
&\quad : 1 \leq i < 4 \\
WriteBuffer_4 &\stackrel{def}{=} (sWrite, \top).WriteBuffer_0 \\
Writer &\stackrel{def}{=} (bWrite, rwbw).WriterWrit \\
WriterWrit &\stackrel{def}{=} (wReset, rwr).Writer \\
Writers &\stackrel{def}{=} Writer \underset{\emptyset}{\bowtie} Writer \underset{\emptyset}{\bowtie} Writer
\end{aligned}
$$

$$
\begin{aligned}
Reader &\stackrel{def}{=} (sReadRequest, rrsrr).(sReadLookup, \top).ReaderRead \\
ReaderRead &\stackrel{def}{=} (rReset, rrr).Reader \\
Readers &\stackrel{def}{=} Reader \underset{\emptyset}{\bowtie} Reader \underset{\emptyset}{\bowtie} Reader
\end{aligned}
$$

$$
\begin{aligned}
Environment &\stackrel{def}{=} Writers \underset{\emptyset}{\bowtie} Readers \\
WebSystem &\stackrel{def}{=} Servers \underset{\{sWrite\}}{\bowtie} WriteBuffer0 \\
System &\stackrel{def}{=} Environment \underset{\mathcal{N}}{\bowtie} WebSystem
\end{aligned}
$$

where $\mathcal{N} = \{bWrite, sReadRequest, sReadLookup\}$.

Figure 1: PEPA model of a high-availability web-server [AKBD04].

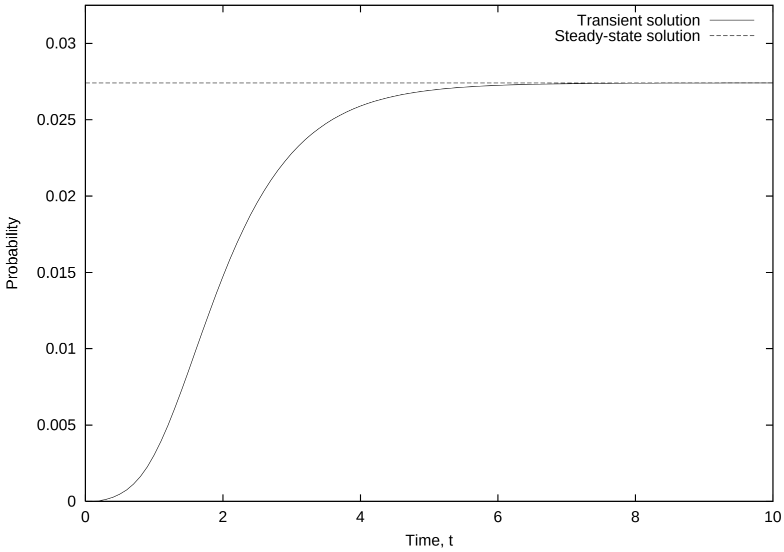

Figure 2: Transient state distribution of being in a total failure mode after start-up in the web-server model.

[AW95] J. Abate and W. Whitt. Numerical inversion of Laplace transforms of probability distributions. *ORSA Journal on Computing*, 7(1):36–43, 1995.

[BDHK03] J.T. Bradley, N.J. Dingle, P.G. Harrison, and W.J. Knottenbelt. Distributed computation of passage time quantiles and transient state distributions in large Semi-Markov models. In *Proceedings of the International Workshop on Performance Modeling, Evaluation and Optimization of Parallel and Distributed Systems (PMEO-PDS'03)*, Nice, April 26th 2003.

[BDHK04] J.T. Bradley, N.J. Dingle, P.G. Harrison, and W.J. Knottenbelt. Distributed computation of transient state distributions and passage time quantiles in large Semi-Markov models. *Future Generation Computer Systems*, 2004. (submitted).

[BDKW03] J.T. Bradley, N.J. Dingle, W.J. Knottenbelt, and H.J. Wilson. Hypergraph-based parallel computation of passage time densities in large semi-Markov models. In *Proceedings of the 4th International Conference on the Numerical Solution of Markov Chains (NSMC'03)*, pages 99–120, Urbana-Champaign IL, USA, September 2nd–5th 2003.

[Pyk61] R. Pyke. Markov renewal processes with finitely many states. *Annals of Mathematical Statistics*, 32(4):1243–1259, December 1961.