A Note on Typing Variables and References

by

Dennis Volpano
Geoffrey Smith

February 1996

Approved for public release; distribution is unlimited.

Prepared for: Naval Postgraduate School
Monterey, CA 93943
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
This report was prepared as part of the Advanced Type Systems Project at the Naval Postgraduate School, which is currently funded by the National Science Foundation under Agreement No. CCR-9400592 and CCR-9414421. Any opinions, findings, and conclusions or recommendations expressed in this report are those of the author and do not necessarily reflect the views of the National Science Foundation.

Reproduction of all or part of this report is authorized.

DENNIS M. VOLPANO
Assistant Professor
Department of Computer Science

Reviewed by:

YUTAKA KANAYAMA
Professor
Department of Computer Science

Released by:

TED LEWIS
Chairman
Department of Computer Science

GORDON SCHACHER
Acting Dean of Research
We consider the polymorphic typing of variables and references with C's address-of operator `&` in the context of nonweak types. A natural semantics and type system are given for a polymorphically-typed imperative language with first class functions. The type system is proved sound with respect to the natural semantics.
A Note on Typing Variables and References

Dennis Volpano
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
volpano@cs.nps.navy.mil

Geoffrey Smith
School of Computer Science
Florida International University
Miami, FL 33199
smithg@fiu.edu

February 8, 1996

Polymorphic typing of variables and references is considered in [1]. However, a treatment of the address-of operator ‘&’ in the context of nonweak types is not given. The operator is treated in [2] but only in the context of weak types, since every type in that system is weak. In this note, the semantics and subject reduction theorem of [1] are reformulated in order to accommodate ‘&’ in the presence of nonweak types.

The syntax of the language in [1] is extended as follows:

\[(Expressions)\quad e ::= &e | l.1\]
\[(Values)\quad v ::= l.0\]

Meta-variable \(l\) ranges over locations. We say \(l.1\) is a variable and \(l.0\) is a reference.

Unlike references, variables are not values. Variables and references replace variable locations and reference locations respectively in the syntax of [1].

Typing rules \((REFLOC)\) and \((VARLOC)\) of [1] are changed and a typing rule for ‘&’ is added—see Figure 1. The domain of a location typing is no longer partitioned into variable and reference locations.

Some changes are needed in the evaluation rules. These changes are reflected in the new rules given in Figure 2.

We now turn to subject reduction. First, we introduce some lemmas:

**Lemma 1** (Superfluousness) If \(\lambda; \gamma \vdash e : \tau \text{ and } l \notin \text{dom}(\lambda)\), then \(\lambda[l : \tau'] ; \gamma \vdash e : \tau\).

**Lemma 2** (Substitution) If \(\lambda; \gamma \vdash v : \sigma \text{ and } \lambda; \gamma[x : \sigma] \vdash e : \tau\), then \(\lambda; \gamma \vdash [v/x]e : \tau\).

Also, if \(\lambda; \gamma \vdash l.1 : \tau \text{ var }\) and \(\lambda; \gamma[x : \tau \text{ var}] \vdash e : \tau'\), then \(\lambda; \gamma \vdash [l.1/x]e : \tau'\).

\[(REFLOC)\quad \lambda; \gamma \vdash l.0 : \tau \text{ ref }\lambda(l) = \tau\]
\[(VARLOC)\quad \lambda; \gamma \vdash l.1 : \tau \text{ var }\lambda(l) = \tau\]
\[(ADDRESS)\quad \lambda; \gamma \vdash e : \tau \text{ var, } \tau \text{ is weak}\]
\[\lambda; \gamma \vdash &e : \tau \text{ ref}\]

Figure 1: New Rules of the Type System
(CONTENTS) \[ \mu \vdash l.1 \Rightarrow \mu(l), \mu \]

(BINDVAR) \[ \mu \vdash e_1 \Rightarrow v_1, \mu_1 \]
\[ l \not\in \text{dom}(\mu_1) \]
\[ \mu[l := v_1] \vdash [l.1/x]e_2 \Rightarrow v_2, \mu_2 \]
\[ \mu \vdash \text{letvar } x := e_1 \text{ in } e_2 \Rightarrow v_2, \mu_2 \]

(UPDATE) \[ \mu \vdash e \Rightarrow v, \mu' \]
\[ \mu \vdash l.1 := e \Rightarrow \text{unit}, \mu'[l := v] \]
\[ \mu \vdash e_1 \Rightarrow l.0, \mu_1 \]
\[ \mu_1 \vdash e_2 \Rightarrow v, \mu_2 \]
\[ \mu \vdash *e_1 := e_2 \Rightarrow \text{unit}, \mu_2[l := v] \]

(ALLOC) \[ \mu \vdash &l.1 \Rightarrow l.0, \mu \]

\[ \mu \vdash e \Rightarrow l.0, \mu' \]
\[ \mu \vdash &*e \Rightarrow l.0, \mu' \]

(DEREF) \[ \mu \vdash e \Rightarrow l.0, \mu' \]
\[ \mu \vdash *e \Rightarrow \mu'(l), \mu' \]

Figure 2: The New Evaluation Rules
Lemma 3 (\forall\text{-intro}) If \( \lambda; \gamma \vdash e : \sigma \) and \( \alpha \) does not occur free in \( \lambda \) or in \( \gamma \), then \( \lambda; \gamma \vdash e : \forall \alpha . \sigma \).

Lemma 4 If \( \lambda[l : \tau]; \gamma \vdash e : \tau' \) and \( l \) does not occur in \( e \), then \( \lambda; \gamma \vdash e : \tau' \).

The preceding lemmas are straightforward variants of those in [1].

The subject reduction theorem now becomes:

**Theorem 5** Suppose that \( \mu \vdash e \Rightarrow v, \mu' \), \( \lambda \vdash e : \tau \), \( \mu : \lambda \), and \( \lambda(l) \) is weak if \( l.1 \) occurs in the range of \( \mu \) or in a \( \lambda \)-abstraction in \( e \), or \( l.0 \) occurs in the range of \( \mu \) or in \( e \). Then there exists a \( \lambda' \) such that \( \lambda \subseteq \lambda' \), \( \mu' : \lambda' \), \( \lambda' \vdash v : \tau \), and \( \lambda'(l) \) is weak if \( l.1 \) or \( l.0 \) occurs in the range of \( \mu' \) or in \( v \).

**Proof.** The proof is by induction on the structure of the derivation of \( \mu \vdash e \Rightarrow v, \mu' \).

For brevity, we present only the interesting cases: (BIND), when \( e_1 \) is not a value, and the evaluation rules of Figure 2.

(BIND). Suppose \( e_1 \) is not a value. Then the evaluation must end with

\[
\mu \vdash e_1 \Rightarrow v_1, \mu_1
\]

\[
\mu_1 \vdash [v_1/x]e_2 \Rightarrow v_2, \mu_2
\]

while the typing must end with

\[
\lambda \vdash e_1 : \tau_1
\]

\[
\lambda;[x \rightarrow \text{AppClose}_A(r_1)] \vdash e_2 : \tau_2
\]

\[
\lambda \vdash \text{let } x = e_1 \text{ in } e_2 : \tau_2
\]

Also, \( \mu : \lambda \) and \( \lambda(l) \) is weak if either \( l.1 \) occurs in the range of \( \mu \) or in a \( \lambda \)-abstraction in \( e_1 \) or \( e_2 \), or \( l.0 \) occurs in the range of \( \mu \) or in \( e_1 \) or \( e_2 \).

By induction, there exists a \( \lambda_1 \) such that \( \lambda \subseteq \lambda_1 \), \( \mu_1 : \lambda_1 \), \( \lambda_1 \vdash v_1 : \tau_1 \), and \( \lambda_1(l) \) is weak if \( l.1 \) or \( l.0 \) occurs in the range of \( \mu_1 \) or in \( v_1 \).

Now to apply induction again we want to show that

\[
\lambda_1 \vdash [v_1/x]e_2 : \tau_2.
\]

By Lemma 1 we have

\[
\lambda_1;[x \rightarrow \text{AppClose}_A(r_1)] \vdash e_2 : \tau_2,
\]

so we can apply Lemma 2 to get what we want provided that we can show

\[
\lambda_1 \vdash v_1 : \text{AppClose}_A(r_1).
\]

Now, applying Lemma 3 to \( \lambda_1 \vdash v_1 : \tau_1 \) we can get \( \lambda_1 \vdash v_1 : \text{AppClose}_{\lambda_1}(r_1) \), but this is not good enough, because \( \lambda_1 \) may contain free strong type variables that are not free in \( \lambda \). To proceed, we exploit our knowledge about what locations can occur in \( v_1 \).

Let \( \lambda_1^* \) be formed by removing from \( \lambda_1 \) any typings \( l : \tau \) such that \( \tau \) is not weak. By the above use of induction, this process does not remove any typings of locations that occur in \( v_1 \), as all such locations have weak types. So by Lemma 4, \( \lambda_1^* \vdash v_1 : \tau_1 \). Hence, by Lemma 3, \( \lambda_1^* \vdash v_1 : \text{AppClose}_{\lambda_1}(r_1) \), since \( \lambda_1^* \) contains no strong type variables.
variables. Lemma 1 then gives $A \vdash v_1 : \text{AppClose}_\lambda(\tau_1)$, and finally by Lemma 2 we get $A \vdash [v_1/x]e_2 : \tau_2$.

By the use of induction above, $A(l)$ is weak if $l.1$ or $l.0$ occurs in the range of $\mu_1$. If a variable $l.1$ occurs in a $\lambda$-abstraction in $[v_1/x]e_2$, then either it occurs in $v_1$ or in a $\lambda$-abstraction in $e_2$. In the first case, $A(l)$ is weak by the above use of induction; in the second case, $A(l)$ is weak by the hypothesis, and so $A(l)$ is weak since $\lambda \subseteq \chi$.

Furthermore, if a reference $l.0$ occurs in $[v_1/x]e_2$, then either it occurs in $v_1$ or $e_2$. In the former case, $A(l)$ is weak by the above use of induction; in the latter, $A(l)$ is weak by the hypothesis, and so $A(l)$ is weak.

Hence we can use induction a second time to show that there exists a $A'$ such that $A \vdash \lambda A'$ and finally by Lemma 2 we get $A, A' \vdash v_2 : \tau_2$, and $A'(l)$ is weak if $l.1$ or $l.0$ occurs in the range of $\mu_2$ or in $v_2$. Since $\lambda \subseteq \lambda_1 \subseteq \lambda'$, we are done.

(INDV). The evaluation must end with

$$
\begin{array}{c}
\mu \vdash e_1 \Rightarrow v_1, \mu_1 \\
l \not\in \text{dom}(\mu_1) \\
\mu_1[l := v_1] \vdash [l.1/x]e_2 \Rightarrow v_2, \mu_2 \\
\mu \vdash \text{letvar } x := e_1 \text{ in } e_2 \Rightarrow v_2, \mu_2
\end{array}
$$

while the typing must end with

$$
\begin{array}{c}
\lambda \vdash e_1 : \tau_1 \\
\lambda_1[x : \tau_1 \text{ var}] \vdash e_2 : \tau_2 \\
\text{If } x \text{ occurs in a } \lambda\text{-abstraction in } e_2 \text{ then } \tau_1 \text{ is weak.}
\end{array}
$$

Also, $\mu : \lambda$ and $\lambda(l')$ is weak if either $l'.1$ occurs in the range of $\mu$ or in a $\lambda$-abstraction in $e_1$ or $e_2$, or $l'.0$ occurs in the range of $\mu$ or in $e_1$ or $e_2$.

By induction, there exists a $A'$ such that $\lambda \subseteq \lambda_1$, $\mu_1 : \lambda_1$, $\lambda_1 \vdash v_1 : \tau_1$, and $\lambda_1(l')$ is weak if $l'.1$ or $l'.0$ occurs in the range of $\mu_1$ or in $v_1$.

Since $l \not\in \text{dom}(\lambda)$, $\lambda \subseteq \lambda_1[l := \tau_1]$.

Since $\lambda_1[l := \tau_1] \vdash l.1 : \tau_1 \text{ var}$ and (by Lemma 1) $\lambda_1[l := \tau_1][x : \tau_1 \text{ var}] \vdash e_2 : \tau_2$, we can apply Lemma 2 to get

$$
\lambda_1[l := \tau_1] \vdash [l.1/x]e_2 : \tau_2
$$

Also, $\mu_1[l := v_1] : \lambda_1[l := \tau_1]$ by Lemma 1.

Next, by the use of induction above, $\lambda_1(l')$ is weak if $l'.1$ or $l'.0$ occurs in the range of $\mu_1[l := v_1]$. Thus, $\lambda_1[l := \tau_1](l')$ is weak since $\lambda_1 \subseteq \lambda_1[l := \tau_1]$. Now suppose that a variable $l'.1$ occurs in a $\lambda$-abstraction in $[l.1/x]e_2$. Then either $l'.1$ occurs in a $\lambda$-abstraction in $e_2$, or else $l' = l$ and $x$ occurs in a $\lambda$-abstraction in $e_2$. In the first case, by the hypothesis, $\lambda(l')$ is weak and so $\lambda_1[l := \tau_1](l')$ is weak. In the second case, by the restriction on the (LETVAR) rule, $\tau_1$ is weak, and so $\lambda_1[l := \tau_1](l')$ is weak. Finally, if $l'.0$ occurs in $[l.1/x]e_2$ then it occurs in $e_2$. Thus, by the hypothesis, $\lambda(l')$ is weak and so $\lambda_1[l := \tau_1](l')$ is weak.

So by a second use of induction, there exists a $\lambda'$ such that $\lambda_1[l := \tau_1] \subseteq \lambda'$, $\mu_2 : \lambda'$, $\lambda' \vdash v_2 : \tau_2$, and $\lambda'(l')$ is weak if $l'.1$ or $l'.0$ occurs in the range of $\mu_2$ or in $v_2$. Since $\lambda \subseteq \lambda_1 \subseteq \lambda_1[l := \tau_1] \subseteq \lambda'$, we are done.

(ADDOF). Suppose the evaluation ends with

$$
\mu \vdash & l.1 \Rightarrow l.0, \mu
$$
while the typing ends with

\[
\lambda \vdash l.1 : \tau \text{ var, } \tau \text{ is weak} \\
\lambda \vdash \& l.1 : \tau \text{ ref}
\]

Also, \( \mu : \lambda \) and \( \lambda(l') \) is weak if either \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu \). Since \( \lambda \vdash l.1 : \tau \text{ var} \), we have \( \lambda(l) = \tau \) by rule (VARLOC). Thus, \( \lambda \vdash l.0 : \tau \text{ ref} \) by (REFLOC).

Furthermore, by the restriction on rule (ADDRESS), \( \tau \), or \( \lambda(l) \), is weak.

Now suppose the evaluation ends with

\[
\mu \vdash e \Rightarrow l.0, \mu'
\]

while the typing ends with

\[
\lambda \vdash e : \tau \text{ ref} \\
\lambda \vdash \star e : \tau \text{ var, } \tau \text{ is weak} \\
\lambda \vdash \& e : \tau \text{ ref}
\]

Also, \( \mu : \lambda \) and \( \lambda(l') \) is weak if \( l'.1 \) occurs in the range of \( \mu \) or in a \( \lambda \)-abstraction in \( e \), or \( l'.0 \) occurs in the range of \( \mu \) or in \( e \).

By induction, there is a \( \lambda' \) such that \( \lambda \subseteq \lambda' \), \( \mu' : \lambda' \), \( \lambda' \vdash l.0 : \tau \text{ ref} \), \( \lambda'(l) \) is weak, and \( \lambda'(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu' \). And, we're done.

(contents). The evaluation must end with

\[
\mu \vdash l.1 \Rightarrow \mu(l), \mu
\]

while the typing must end with

\[
\lambda \vdash l.1 : \tau \text{ var} \\
\lambda \vdash l.1 : \tau
\]

Also, \( \mu : \lambda \) and \( \lambda(l') \) is weak if either \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu \). From \( \mu : \lambda \), we have \( \lambda \vdash \mu(l) : \lambda(l) \). Since \( \lambda \vdash l.1 : \tau \text{ var} \), we have \( \lambda(l) = \tau \), so \( \lambda \vdash \mu(l) : \tau \).

(update). Suppose the evaluation ends with

\[
\mu \vdash e \Rightarrow v, \mu'
\]

while the typing ends with

\[
\lambda \vdash l.1 := e \Rightarrow \text{unit}, \mu'[l := v]
\]

Also, \( \mu : \lambda \) and \( \lambda(l') \) is weak if \( l'.1 \) occurs in the range of \( \mu \) or in a \( \lambda \)-abstraction in \( e \), or \( l'.0 \) occurs in the range of \( \mu \) or in \( e \).

By induction, there exists a \( \lambda' \) such that \( \lambda \subseteq \lambda' \), \( \mu' : \lambda' \), \( \lambda' \vdash v : \tau \), and \( \lambda'(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu' \) or in \( v \).

By rule (LIT), \( \lambda' \vdash \text{unit} : \text{unit} \). Since \( \lambda \vdash l.1 : \tau \text{ var} \), \( \lambda(l) = \tau \) by (VARLOC). So \( l \in \text{dom}(\lambda') \) since \( \lambda \subseteq \lambda' \), and thus \( \text{dom}(\mu'[l := v]) = \text{dom}(\lambda') \). If \( l' \) is a location such
that \( P' \neq l \), then \( \lambda' \vdash \mu'(l') : \lambda'(l') \) since \( \mu' : \lambda' \). If \( l' = l \) then \( \mu'[l := v](l') = v \). So \( \lambda' \vdash \mu'[l := v](l') : \tau \) since \( \lambda' \vdash v : \tau \). Thus, \( \mu'[l := v] : \lambda' \). Finally, by the above use of induction, \( \lambda'(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu'[l := v] \).

Now suppose the evaluation ends with

\[
\mu \vdash e_1 \Rightarrow 1.0, \mu_1 \\
\mu_1 \vdash e_2 \Rightarrow v, \mu_2 \\
\mu \vdash *e_1 := e_2 \Rightarrow \text{unit}, \mu_2[l := v]
\]

while the typing ends with

\[
\lambda \vdash *e_1 : \tau \text{ var} \\
\lambda \vdash e_2 : \tau \\
\lambda \vdash *e_1 := e_2 : \text{ unit}
\]

Also, \( \mu : \lambda \) and \( \lambda(l') \) is weak if \( l'.1 \) occurs in the range of \( \mu \) or in a \( \lambda \)-abstraction in \( e_1 \) or \( e_2 \), or \( l'.0 \) occurs in the range of \( \mu \) or in \( e_1 \) or \( e_2 \).

By rule \((L-VAL)\), \( \lambda \vdash e_1 : \tau \text{ ref} \). By induction, there exists a \( \lambda_1 \) such that \( \lambda \subseteq \lambda_1 \), \( \mu_1 : \lambda_1, \lambda_1 \vdash 1.0 : \tau \text{ ref}, \lambda_1(l) \) is weak, and \( \lambda_1(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu_1 \). By Lemma 1, \( \lambda_1 \vdash e_2 : \tau \). Suppose that a variable \( l'.1 \) occurs in a \( \lambda \)-abstraction in \( e_2 \). Then by the hypothesis, \( \lambda(l') \) is weak and so is \( \lambda_1(l') \) since \( \lambda \subseteq \lambda_1 \). Likewise, if \( l'.0 \) occurs in \( e_2 \), then \( \lambda(l') \) is weak and thus so is \( \lambda_1(l') \).

So by a second use of induction, there is a \( \lambda' \) such that \( \lambda_1 \subseteq \lambda' \), \( \mu_2 : \lambda' \), \( \lambda' \vdash v : \tau \), and \( \lambda'(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu_2 \) or in \( v \). The proof is now similar to the first \((\text{UPDATE})\) case above.

\((\text{ALLOC})\). The evaluation must end with

\[
\mu \vdash e := v, \mu' \\
\mu' \notin \text{ dom}(\mu') \\
\mu \vdash \text{ ref } e \Rightarrow 1.0, \mu'[l := v]
\]

while the typing ends with

\[
\lambda \vdash e : \tau, \tau \text{ is weak} \\
\lambda \vdash \text{ ref } e : \tau \text{ ref}
\]

Also, \( \mu : \lambda \) and \( \lambda(l') \) is weak if \( l'.1 \) occurs in the range of \( \mu \) or in a \( \lambda \)-abstraction in \( e \), or \( l'.0 \) occurs in the range of \( \mu \) or in \( e \).

By induction, there exists a \( \lambda' \) such that \( \lambda \subseteq \lambda' \), \( \mu' : \lambda' \), \( \lambda' \vdash v : \tau \), and \( \lambda'(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu' \) or in \( v \).

Now \( \lambda' \subseteq \lambda'[l : \tau] \) since \( l \notin \text{ dom}(\mu') \).

By Lemma 1 and the above use of induction, \( \mu'[l := v] : \lambda'[l : \tau] \). Furthermore, \( \lambda'[l : \tau] \vdash 1.0 : \tau \text{ ref} \) by rule \((\text{REFLOC})\). Again by the above use of induction, \( \lambda'(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu'[l := v] \), and hence \( \lambda'[l : \tau](l') \) is weak since \( \lambda' \subseteq \lambda'[l : \tau] \). Finally, \( \lambda'[l : \tau](l) = \tau \) and \( \tau \) is weak by the restriction on rule \((\text{REF})\).

\((\text{DEREF})\). The evaluation must end with

\[
\mu \vdash e \Rightarrow 1.0, \mu' \\
\mu' \vdash *e \Rightarrow \mu'(l), \mu'
\]
while the typing ends with

\[
\begin{align*}
\lambda &\vdash e : \tau_{\text{ref}} \\
\lambda &\vdash se : \tau_{\text{var}} \\
\lambda &\vdash se : \tau
\end{align*}
\]

Also, \( \mu : \lambda \) and \( \lambda(l') \) is weak if \( l'.1 \) occurs in the range of \( \mu \) or in a \( \lambda \)-abstraction in \( e \), or \( l'.0 \) occurs in the range of \( \mu \) or in \( e \).

By induction, there exists a \( \lambda' \) such that \( \lambda \subseteq \lambda' \), \( \mu' : \lambda' \), \( \lambda' : t.0 : \tau_{\text{ref}} \), \( \lambda'(l) \) is weak, and \( \lambda'(l') \) is weak if \( l'.1 \) or \( l'.0 \) occurs in the range of \( \mu' \).

Since \( \lambda' : t.0 : \tau_{\text{ref}} \), \( \lambda'(l) = \tau \) by rule (REFLOC). Now \( \lambda' : t.0 : \lambda'(l) \), since \( \mu' : \lambda' \), so \( \lambda' : \mu'(l) : \tau \). \□

**References**


Distribution List

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314
2 copies

Library, Code 52
Naval Postgraduate School
Monterey, CA 93943
2 copies

Director of Research Administration
Code 012
Naval Postgraduate School
Monterey, CA 93943
1 copy

Prof. Luqi
Computer Science Department
Code CS/Lq
Naval Postgraduate School
Monterey, CA 93943
1 copy

Dr. Geoffrey Smith
School of Computer Science
Florida International University
University Park
Miami, FL 33199
10 copies

Dr. Dennis Volpano
Computer Science Department
Code CS/Vo
Naval Postgraduate School
Monterey, CA 93943
10 copies