0.0.1 10. Hausaufgabe

Comment: „Monadic IO“

```
A) (Introduction)
B) Using Monads to do IO is better
   1. Ability to optimize a program at compile-time
   2. Static typechecking
   3. Ability to sequence IO actions
C) (Conclusion)
```

With the purely functional programming language Haskell becoming more widely known each day, the idea of using Monads to do Input/Output (IO) gains support. In the following I´ll show why using Monads to do IO is a better way to communicate with the real world than using „normal“ side-effectful functions which aren´t possible in a purely functional language.

Firstly, by using Monads, the compiler is able to optimize programs at compile-time. For example, the following Haskell code...

```
f :: IO String
f = haskell_compiler_version >>= \x -> return x
```

...can be substituted by...

```
f :: IO String
f = return "Glasgow Haskell Compiler 6.4"
```

...at compile time. This causes many perfomance optimizations which wouldn´t be possible to do without using Monads to do IO.

Secondly, by using Monads, the compiler can typecheck your code even when using IO. For example, in the following code, Haskell will throw an exception if the user does not enter an integer:

```
read_an_integer :: IO Int
read_an_integer = readLn

main :: IO ()
main = read_an_integer >>= \x -> putStrLn $ "Your input
was: " ++ x
```

As this is clearly not possible without using monadic IO, it´s a great benefit for the programmer not having to do all the typechecking on his own.

Lastly, monadic IO is the only way to sequence side-effectful actions in a purely functional language. It would, for example, be disatrous, if the following snippet deleted »file« before reading it:

```
f fh = readFromFH fh >>= \line -> unlink file
```

By using monadic IO, and especially by using the binding operator »(>>=)«, the compiler is able to properly sequence all side-effectful functions.

Monadic IO is a great way to do IO in purely functional languages, so I hope other languages will adapt this way of sequencing side-effectful functions one day.

Comment: „IM2000“

```
A) (Introduction)
B) IM2000 is better than SMTP
   1. No bounces
   2. Less traffic
   2. Less spam
C) (Conclusion)
```

One of the world´s most important Internet services today is the electronic mail, commonly transferred using the Simple Mail Transfer Protocol (SMTP) or one of its extensions. In the following I´ll show the advantages Internet Mail 2000 (IM2000) to regular SMTP.

Many mail server administrators know the problem as old as Internet mail itself – a SMTP server triggers sending a so-called „bounce“ back to sender if it isn´t able to further deliver the mail. As many statistics show, this is currently a big problem and will continue to become even more serious in the next few years. With IM2000, there are no bounces due to the different way the storing of mail messages is handled with IM2000 – while SMTP mails were simply sent to next reachable SMTP server, which (may) had to report a failure, the mails stay stored on the sender´s ISP. Such, if the destination mail address doesn´t exist, the mail is simply not fetched,

but there´s no need to send bounces. Therefore, IM2000 is better than SMTP.

Another reason why IM2000 is better than SMTP is, that the traffic actually needed is minimized when using IM2000. As described in the last paragraph, the actual mail message is not sent to the destination´s mail server until the receiver´s Mail User Agent (MUA) tells the sending server to do so. With millions of mails being sent each day, this will cause enormous savings of traffic. Because of this, IM2000 is better than SMTP.

The last reason why IM2000 is better than SMTP is, that unsolicited bulk mail, commonly referred to as „spam“, will cease to exist. Because the costs of sending a mail move from the receiver´s side to the sender, spammers can´t continue to use infected computers, usually connected in a central-managed „botnet“, as cheap mail servers – those computers would have to stay online 24/7/365 to be able to wait for a »MAIL˙XFER« request. Clearly, most personal computers are switched off at least once a day. Therefore, ordinary personal computers won´t be able to get abused as cheap mail relays, and this in turn will cause spam to cease. Therefore IM2000 is better than SMTP.

Because of all these advantages of IM2000 to SMTP, I´d vote for a quick adaption of IM2000 to make the Internet the friendly place it once used to be.

Comment: „IPv6“

```
A) (Introduction)
B) IPv6 is better than IPv4
   1. More addresses available
   2. Redundancy by using multicast
   3. Mobile IPv6 Extensions
C) (Conclusion)
```

With the accellerated growth of the Internet, the address space provided by IPv4 will soon be exhausted. In the following, I´ll show why IPv4´s designated successor, IPv6, is better than IPv4.

The first reason, why IPv6 is better than IPv4 is, that the address space available will be big enough for the next few generations. This is because IPv4 uses an unsinged 32 bit sized integer to address

each node of the Internet, while IPv6 uses 128 bit. Such, 2^{128} addresses will be available with IPv6, which will clearly be enough. Therefore, IPv6 is better than IPv4.

Secondly, IPv6 increases the redundancy by extensively using multicast. With IPv4, the breakdown of only one router on the path to the destination is sufficient to cause all connections to the destination host to terminate. IPv6 fixes this problem by providing multicast, i.e. one address is used to address multiple hosts. Therefore **all** routers on the path to the destination have to go offline in order to cause the destination host being unreachable. Because of this, IPv6 is better than IPv4.

Most importantly, IPv6 provides Mobile IPv6 Extensions. IPv4 lacks these extensions, and that´s the reason for easy roaming not working presently. By contrast, IPv6 has complete support for the Mobile IPv6 Extensions. Thus, you´ll be able to switch networks **transparently**. Of course, you´ll get a new address, but your old address will continue to work! Therefore, one doesn´t have to terminate all his connections only to be able to switch networks. Therefore, IPv6 is better than IPv4.

Because off all these advantages of IPv6 to IPv4, I´d like seeing an accelerated adaption of IPv6.